



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



TESINA DE MÁSTER

Título:

DISEÑO E IMPLEMENTACIÓN DE UNA PINZA
MODULAR DESTINADA A LA MANIPULACIÓN
MULTI-PICK DE ALIMENTOS; DESARROLLO Y
PROGRAMACIÓN DE LA APLICACIÓN DE
CONTROL.

Autor: Adrián Ochoa Soler

Director: Dr. Martín Mellado Arteché

Índice

1.	INTRODUCCIÓN.....	4
1.1	Presentación de la tesina	4
1.2	Introducción a los sistemas de manipulación	4
1.2.1	Sistemas Pick & place.	7
1.2.2	Sistemas Multi-pick	8
1.3	Descripción de la problemática a resolver	9
1.4	Motivación	12
1.4.1	Motivación profesional.....	12
1.4.2	Motivación innovadora	14
1.4.3	Motivación personal.....	14
1.5	Objetivos.....	14
2.	ESTADO DEL ARTE	16
2.1	Herramientas Multi-pick Fijas.....	16
2.1.1	<i>Multi-pick handling unit</i>	16
2.1.2	<i>Gripper FXP</i>	17
2.1.3	<i>Robotic EOAT</i>	18
2.1.4	<i>Palletiser for small boxes</i>	19
2.1.5	<i>Robots de paletización y embalaje RP4 y RCP</i>	20
2.2	Herramientas Multi-pick dinámicas.....	20
2.2.1	Herramientas de manipulación por apilado de productos	21
2.2.1.1	<i>MultiPick Case cardboard gripper</i>	21
2.2.1.2	<i>Robocollect y SG47</i>	22
2.2.2	Herramienta Multipick Salami-Picker	23
2.2.3	Mitsubishi Electric Single and Multi Pick Robot	24
2.2.4	Multi gripper system	25
2.3	Dispositivos de posicionamiento.	26
2.3.1	Sistemas de posicionamiento lineal.....	26
2.3.1.1	<i>Guías lineales de un solo eje</i>	27
2.3.1.2	<i>Guías lineales de dos o más ejes</i>	28
2.3.2	Sistemas de posicionamiento angular.	30
2.3.2.1	<i>Motores</i>	30
2.3.2.2	<i>Mesas giratorias</i>	32

3.	EXPLICACIÓN DEL PROBLEMA	34
3.1	Alternativas para el diseño de la herramienta.	37
3.1.1	Herramienta multi-pick doble.....	37
3.1.2	Herramienta multi-pick triple.	39
3.1.3	Herramienta multi-pick múltiple.....	41
3.2	Situación de partida	44
3.3	Elección de la herramienta multi-pick.	46
3.4	Herramientas Hardware a utilizar.....	46
3.4.1	Robot industrial ABB IRB-360.	46
3.4.2	Cámara de visión	47
3.4.3	Guía lineal SLN-27	48
3.4.4	Driver de control DRV8825.....	49
3.4.5	Arduino Mega 2560.....	51
3.4.6	Motor Brushless.	53
3.4.7	Driver de control del motor Brushless.	54
3.4.8	Fuente de alimentación.....	55
3.5	Herramientas Software a utilizar	56
3.5.1	Software de Arduino.	56
3.5.2	Software de visión (Pickmaster).	57
3.5.2.1	<i>Algoritmo de control de visión por computador para multi-pick robotizado.....</i>	<i>58</i>
3.5.3	Otros.....	60
4.	DESARROLLO E IMPLEMENTACIÓN DE LA HERRAMIENTA	61
4.1	Herramienta multi-pick doble (TCP fijo + TCP móvil).	61
4.1.1	Inicio de operación.....	63
4.1.2	Recepción de coordenadas	64
4.1.3	Operación Pick	65
4.1.3.1	<i>Determinación de la posición mediante el encoder.....</i>	<i>66</i>
4.1.3.2	<i>Control del motor PaP con rampa de velocidad lineal.</i>	<i>68</i>
4.1.4	Operación Place	72
4.2	Herramienta multi-pick triple (2 TCP móvil + 1 TCP fijo)	75
4.2.1	Inicio de operación.....	77
4.2.2	Recepción de coordenadas.....	77
4.2.3	Operación Pick	79
4.2.3.1	<i>Control del motor brushless.....</i>	<i>79</i>

4.2.4	Operación Place	85
5.	MONTAJE Y PROGRAMACIÓN DE LA HERRAMIENTA	87
5.1	Montaje y programación de la herramienta doble.	87
5.2	Montaje y programación de la herramienta triple.	98
6.	VERIFICACIÓN Y RESULTADOS OBTENIDOS.....	113
6.1	Resultados en el posicionamiento del motor PaP	113
6.2	Resultados en el posicionamiento del motor Brushless	115
6.3	Resultados de la herramienta doble.	117
7.	CONCLUSIONES Y TRABAJOS FUTUROS	118
7.1	Conclusiones.	118
7.2	Trabajos futuros.	118
7.2.1	Mejora de la programación de la aplicación de control.	118
7.2.2	Ampliación del área efectiva de trabajo mediante detección de colisiones.	119
7.2.3	Incorporación de una interfaz gráfica para el control de la herramienta.	120
8.	REFERENCIAS.....	121

1. INTRODUCCIÓN

1.1 Presentación de la tesina

El trabajo descrito en esta tesina corresponde a un desarrollo tecnológico e innovador nacido en Octubre de 2013 y que tiene como objetivo diseñar y desarrollar un prototipo de pinza mecatrónica dinámica y modular, que permita la manipulación de alimentos en procesos automatizados multi-pick. El proyecto está siendo desarrollado en el Instituto de automática e informática industrial (AI2) localizado en la Ciudad Politécnica de la Innovación (CPI) de la Universidad Politécnica de Valencia (UPV). El grupo de trabajo se encuentra compuesto por:

- Adrián Ochoa Soler:

Ingeniero técnico industrial especializado en electrónica industrial por la UPV (Julio de 2011) y miembro del AI2 en calidad de becario de colaboración. Actualmente se encuentra cursando el último curso del Máster en Automática e Informática Industrial (MAII).

Es el autor del presente documento.

- Martín Mellado Arteché:

Doctor en Ingeniería informática, Profesor Titular de la Universidad Politécnica de Valencia y miembro investigador del AI2.

Actúa como director de la Tesina.

1.2 Introducción a los sistemas de manipulación

El término manipulación se refiere a aquello que por su actividad tiene contacto directo con los alimentos durante su preparación, fabricación, transformación, elaboración, envasado, almacenamiento, transporte, distribución, venta, suministro y servicio.

Tanto la manipulación como el agarre de objetos son conceptos de vital importancia en el campo de la robótica industrial. En los últimos años ha habido una gran dedicación tanto al diseño como a la fabricación de sistemas mecatrónicos dedicados a tareas de agarre para aplicaciones de ensamblaje, elevación o transporte de objetos.

Hoy en día podemos encontrar numerosas técnicas de manipulación de objetos que consiguen llevar un elemento desde una posición inicial hasta la posición final deseada. Las herramientas que incluyen los robots actuales pueden conseguirlo en la mayoría de los casos y satisfacer las necesidades para automatizar el proceso, pero no obstante, hay casos en los que los procesos requieren una herramienta especial para cumplir las especificaciones deseadas. Generalmente, cuando se realiza una manipulación de objetos, se debe cumplir como mínimo alguno de los parámetros siguientes:

- **Orientación:** Consiste en situar el objeto en una orientación y posición relativa respecto a otro. Esta operación requiere hasta un máximo de 6 grados de libertad.
- **Agarre:** Se produce cuando se realiza el acto de tomar, coger, o mantener firmemente algo con una herramienta.
- **Transporte:** Se refiere al desplazamiento del objeto desde un punto de origen a un punto de destino.

El caso más complejo de manipulación de objetos requerirá los tres parámetros anteriores, pero hay que tener en cuenta que la dificultad y complejidad del diseño del mecanismo aumentará necesariamente al incorporar más fases, ya que es mucho más fácil manipular objetos con el único objetivo de transportarlo, que necesitar también el agarre y la orientación del mismo.

Otro factor a tener en cuenta en cuanto a la complejidad de la manipulación es el mismo objeto a manipular, por ejemplo, el tamaño, la forma, el material, y el centro de gravedad son, entre otros, parámetros importantes que pueden complicar la manipulación.

Para realizar una correcta manipulación, se requiere precisión y repetitividad, de manera que se pueda asegurar la calidad exigida. Para ello, es necesario recurrir a diversas tecnologías como sensores de presencia, o cámaras de visión artificial, que sean capaces de indicar la posición y orientación del objeto a manipular. Actualmente, se utilizan técnicas de visión por computador capaces de reconocer formas y colores para que, procesando el dato, el sistema sea capaz de conocer la orientación y posición del objeto y lograr así su agarre.

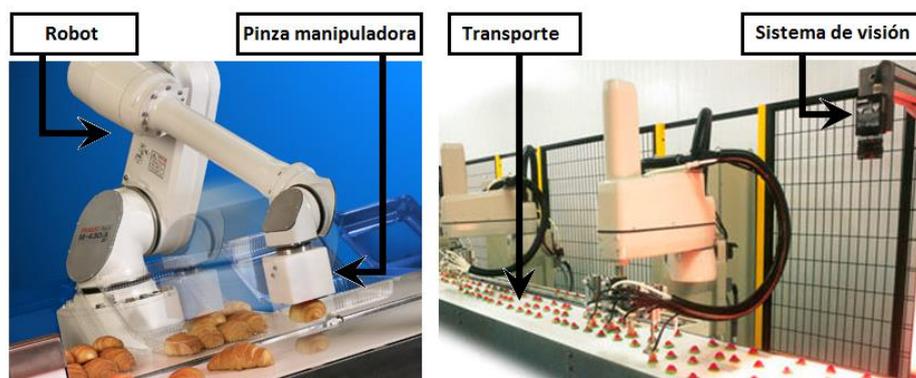


Figura 1.1. Ejemplos de manipulación pick and place de alimentos mediante robots industriales.

Para el diseño de herramientas manipuladoras existen diversas técnicas que utilizan distintas tecnologías. Las más estudiadas, se podrían clasificar en los siguientes tipos, según la fuente de energía utilizada:

- Eléctricas y electrónicas.

- Mecánicas.
- Neumáticas.
- Electrostáticas.
- Adhesivas.

Además, estas técnicas se podrían combinar entre ellas, por ejemplo, se puede utilizar una técnica de tipo eléctrica para el posicionamiento del objeto, y una técnica neumática para el agarre del mismo. De cada una de las técnicas mencionadas anteriormente derivan distintos tipos de agarre del objeto (subgrupos), G. Seliger, por ejemplo, emplea 4 grupos según el principio y la fuente de energía a emplear, y los divide en los siguientes subgrupos según el tipo de agarre:

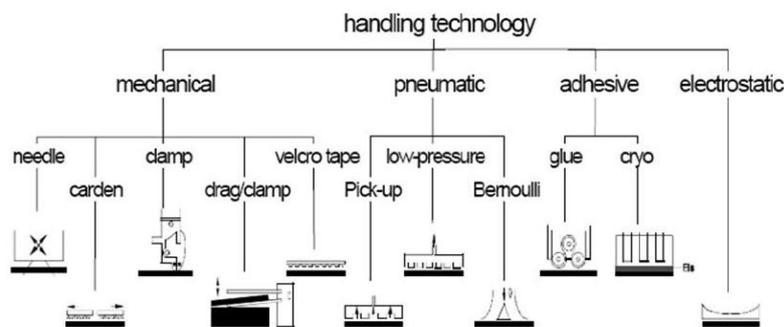


Figura 1.2. Grupos y subgrupos de tecnologías para el agarre de productos.

C.Blanes y M.Mellado, han realizado una tabla (tabla 1.1) considerando los distintos métodos anteriores y analizando nuevas posibilidades. Teniendo en cuenta los conceptos anteriores necesarios a realizar durante la manipulación robotizada y según las estrategias a usar, muestran los métodos posibles y las capacidades de estos en función de su habilidad para transportar, orientar y agarrar el producto. La estrategia constituye la base sobre la cual se pretende manipular el objeto. Como se ha citado anteriormente, el conjunto del proceso puede emplear una o más estrategias. Una vez seleccionada se debe considerar el método con el cual se pretende realizar manipulación. El conjunto de estrategias y métodos debe lograr satisfacer las necesidades creadas de transporte, orientación y agarre buscados:

Estrategia	Método		Capacidades		
			Transporta	Orienta	Agarra
Aire	Aspiración	Ventosas	No	No	Si
		Conductos	Si	No	No
	Impulsión	Bernoulli	Baja	No	Si
		Soplado	Si	Baja	No
Fuerzas de contacto	Garras	Eléctricas	No	Si	Si
		Neumáticas	No	Si	Si
		Hidráulicas	No	Si	Si
		De goma (globos y dedos)	No	No	Si
		Manos articuladas	No	Si	Si
	Mecanismos articulados	No	Baja	Si	
	Ganchos y velcros	No	No	Si	
Clipado	No	No	Si		
Penetración en objeto	Agujas	No	No	Si	
Unión química	Soldadura	No	No	Si	
	Adhesivos (sistema LAAG)	No	No	Si	
Magnetismo	Atracción	Baja	Si	Si	
	Repulsión	Si	Baja	No	
Carga eléctrica	Fuerzas electrostáticas entre piezas	No	No	Si	
Fluido	Congelación de vapor Cryo	No	No	Si	
	Magneto-reológico	No	Baja	Si	
Propiedades físicas del objeto	Gravedad o caída	Si	Baja	No	
	Apilar/Empujar	Si	Baja	No	
	Dinámica	Si	Baja	No	
	Vibración orientada	Si	Baja	No	
	Abrazar	No	Baja	Si	
Levitación	Aerodinámica	Si	No	No	
	Electrostática	Si	No	No	
	Óptica	Si	No	No	
	Acústica	Si	No	No	
	Magnética	Si	No	No	
	Superconductor	Si	No	No	

Tabla 1.1. Métodos y capacidades para la manipulación de alimentos

1.2.1 Sistemas Pick & place.

Los sistemas pick & place se pueden utilizar en la industria para una gran variedad de aplicaciones de entre las que destacan la manipulación de alimentos y el empaquetado o paletizado de productos. Básicamente la función que realizan estos robots es coger un producto o un conjunto de productos que provienen de una cinta transportadora, un contenedor, o una tolva y transportarlos a un lugar concreto en donde, en general, se realizará la fase de empaquetado o incluso la preparación del producto para otra fase de elaboración.

Desde el punto de vista de los movimientos cíclicos que puede ejecutar un sistema pick & place, se pueden distinguir tres operaciones básicas y repetitivas:

- Coger una pieza con la pinza (gripper): **pick-up**.
- Trasladar la pieza a un lugar establecido por el usuario: **transfer**.
- Abrir la pinza para colocar la pieza en el sitio de destino: **place**.

Una de las características principales que deben poseer los robots que realizan aplicaciones de pick & place es la velocidad. El modelo siempre dependerá del producto a manipular, su volumen, su peso y también el área de trabajo para tener en cuenta el alcance.

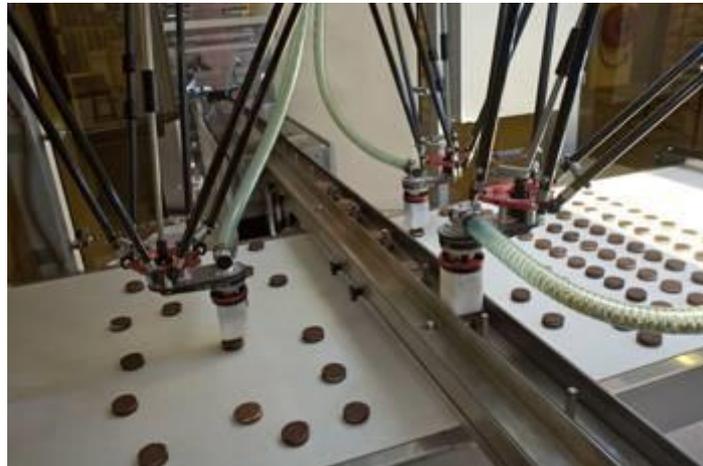


Figura 1.3. Robot Delta ejecutando tarea de pick and place

1.2.2 Sistemas Multi-pick

El concepto de un sistema Multi-pick se basa en la posibilidad de coger varios productos antes de realizar su traslado (transfer) al objetivo establecido. Al poder coger un número determinado de productos, se pueden satisfacer necesidades en la producción tales como ahorro de tiempo de fabricación, y ahorro de coste del producto.

El diseño de sistemas y garras Multi-pick está pensado para trabajar en sistemas de manipulación pick & place de manera que el agarre de varios productos se haría en la operación de *pick-up* con el propósito de depositarlos en el objetivo realizando la operación *place*.

Desde el punto de vista de la manipulación de productos, los sistemas Multi-pick podrían diseñarse para que la pinza o garra manipuladora sea capaz de coger un número determinado de productos a la vez. Por otra parte existe la posibilidad de que la pinza se haya diseñado con el propósito de recoger N productos de tal manera que se realicen N movimientos *pick-up*. Durante las operaciones *pick-up* los productos se pueden ir apilando, y así, cuando se realice la operación *place*, depositarlos en el lugar establecido de manera simultánea.

En definitiva, se puede decir que dentro de los sistemas de manipulación Pick & place, se pueden utilizar dispositivos Multi-pick, con la posibilidad de realizar la operación de *pick* de dos maneras diferentes:

- Con una garra Multi-pick fija. (figura 1.4).
- Con una garra Multi-pick dinámica. (figura 1.5).



Figura 1.4. Sistema Multi-pick fijo.



Figura 1.5. Sistema Multi-pick de tipo pila

La utilización de herramientas Multi-pick permite reducir de forma notable el efecto de los desplazamientos de los productos en los sistemas pick & place, no obstante, estas garras suelen ser de compleja operativa y mayor peso que las garras convencionales. Además, su empleo está en parte condicionado por la disposición de los objetos antes del agarre. En el diseño de este tipo de herramientas se tiene en cuenta que la distribución de los objetos sea adecuada para la simplificación de la operativa de la garra. La orientación de inicio y destino de los objetos puede ser la misma, o pueden reorientarse durante el desplazamiento al destino.

Un factor a tener en cuenta, es que el aumento de la complejidad de la garra, incrementa el peso de la misma ya que el número de componentes se ve incrementado, esto repercute en el manipulador necesario, la distribución de todos los periféricos, un mayor dimensionado y necesidades de aire, y un incremento de la necesidad de mantenimiento.

1.3 Descripción de la problemática a resolver

El sector de la robótica se está adaptando a las necesidades generales de las empresas agroalimentarias, y está desarrollando sistemas de empaquetado y manipulación a alta velocidad para las líneas de producción. Los avances de las tecnologías a un menor coste y el desarrollo de sistemas flexibles de robótica modular hacen que cada vez resulte más rentable automatizar los actuales procesos manuales.

Actualmente, se sigue investigando por ejemplo en la evolución de las superficies de contacto y la presión ejercida entre un dispositivo de agarre y un determinado objeto, ya que ha sido siempre un parámetro de control muy perseguido en robótica de manipulación. Además, la gran

ventaja que supone el poder realimentar el sistema de agarre mediante sensores y poder realizar controles en lazo cerrado de posición y fuerza supone, también, un aumento de la electrónica de adquisición y, por supuesto, de control del sistema.

Partiendo de las ventajas que nos ofrecen el control y los nuevos dispositivos de agarre, se puede seguir evolucionando en el campo de la robótica de manipulación realizando mejoras en los procesos de producción y empaquetado.

En la industria, la mayoría de los procesos automáticos de empaquetado, disponen de una cinta transportadora donde se van depositando los productos y se conducen hasta el área de trabajo de cada robot, los cuales, ejecutando las tareas de *pick & place* irán depositando los productos en su lugar correspondiente de empaquetado. Para llevar a cabo este proceso, se utiliza un sistema de visión que identifica los productos provenientes y le envía la información a los diferentes robots (figura 1.6). En este caso, los productos pueden ir de manera desordenada ya que cada robot posee una pinza que agarrará un único producto. Sin embargo, es inevitable que se produzcan retardos en la operación, así como descartes de objetos que irremediablemente no da tiempo a agarrar.

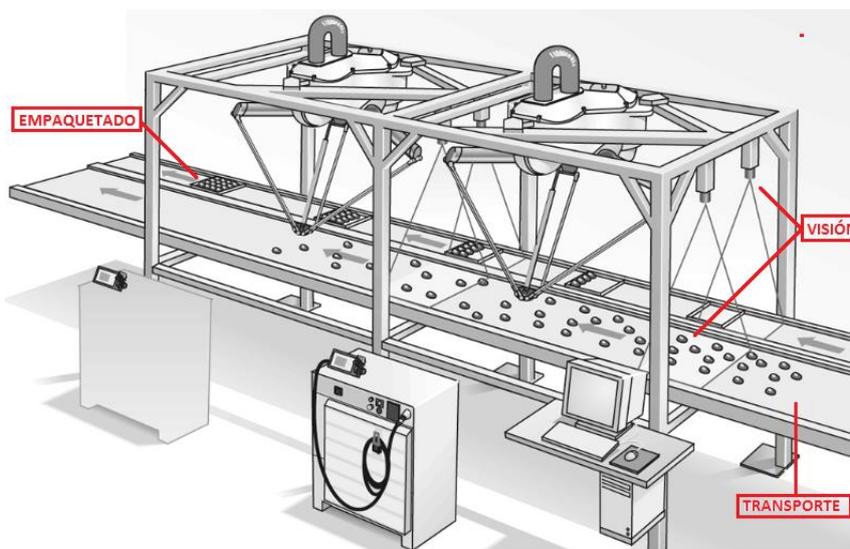


Figura 1.6. Fases de empaquetado de productos mediante sistema pick & place.

Cuando se realiza la fase de empaquetado en sistemas en los que los robots industriales disponen de herramientas de agarre Multi-pick, nos podemos encontrar con los siguientes problemas:

- Si se tiene un sistema de empaquetado en el que los robots disponen de una pinza Multi-pick fija (figura 1.4), es necesario que los productos estén ordenados antes de que lleguen a la fase de visión y empaquetado para que se produzca el correcto agarre de

los mismos (pre-procesado). Esto supone un problema, debido a que necesariamente se requiere de mecanismos adicionales que realicen dicha operación.

- Esta adversidad se puede resolver colocando una pinza dinámica que sea capaz de realizar varios movimientos *pick*. Como se ha visto en el apartado 1.2.2, existe la posibilidad de utilizar una pinza que vaya apilando los N productos establecidos y que posteriormente realice la operación *place* (figura 1.5). Pero esta solución también tiene sus desventajas. El método de apilar solo sirve para productos que puedan estar en contacto unos con otros y que posean formas geométricas sencillas y similares. No se podría utilizar por ejemplo, con productos alimentarios que se compongan de productos líquidos o viscosos tales como el chocolate o la mermelada. Además, el hecho de realizar numerosas operaciones de *pick*, hace que el proceso de empaquetado pueda llegar a ser más lento en determinados procesos.

La solución pasaría por utilizar una pinza Multi-pick dinámica capaz de posicionar cada uno de sus dispositivos de agarre a los productos establecidos por el sistema de visión antes de realizar la operación de *pick*. Así, el robot será capaz de realizar el agarre de los productos a la vez, y evitar así el pre-procesado o alineación de los objetos y el retardo o descarte de los mismos que se producen en los sistemas de manipulación *pick and place* actuales. Seguidamente, en el transcurso de *pick* a *place*, se debería realizar el posicionado de cada uno de los dispositivos de agarre de la pinza a la posición establecida de *place*, de tal manera que el empaquetado se realice de forma correcta.

Una gran ventaja de utilizar este tipo de pinza Multi-pick sería la posibilidad de añadirle módulos con la misma disposición, de tal manera que cada módulo añadido sería un producto más que podríamos coger de manera simultánea.

Para realizar el diseño de una herramienta Multi-pick es necesario resolver problemas como la correcta elección de los dispositivos que realizan los movimientos de posicionado de *pick* y *place*. Entre ellos, se encuentran los motores de accionamiento, los mecanismos de posicionado y su distribución, y la utilización de sensores que permitan la realimentación del lazo de control del sistema. La buena elección de los dispositivos descritos permitirá un diseño de la herramienta capaz de reunir las siguientes características y mejorar así las especificaciones que actualmente disponen los sistemas de manipulación y empaquetado:

- Aumentar la productividad del proceso.
- Aportar flexibilidad ante posibles cambios en la logística de la producción.
- Reducir los costes.

1.4 Motivación

Actualmente ha habido numerosos avances en la automatización de sistemas y procesos en entornos industriales. Concretamente en la industria alimentaria, tanto la robótica como la mecatrónica desempeñan un papel importante siendo las piezas clave, ya que han abierto nuevas oportunidades para la automatización de procesos alimentarios. La utilización de robots en este sector, aporta flexibilidad, seguridad y protección, una mayor higiene del proceso, repetitividad del tratamiento, efectividad y elevada producción. Además, el hecho de eliminar el contacto humano con los productos en la cadena de producción y envasado, los sistemas robotizados ofrecen la posibilidad de reducir el riesgo de contaminación.

En la industria alimentaria se emplean todo tipo de robots y herramientas de manipulación para las labores más variadas. Así, podemos encontrarlos manipulando productos delicados como pan tostado, galletas, de bollería y confitería. Actualmente, los expertos del sector aseguran que estamos viviendo una reducción del ciclo de vida de los productos. Se ha hecho muy necesario sorprender, reinventar y ampliar las gamas de productos, un hecho que ha obligado a los fabricantes a ser más flexibles en la producción y más ágiles a la hora de lanzar nuevos productos al mercado.

Los robots están diseñados para hacer la vida más fácil a aquellas empresas que los utilizan. En determinados procesos como, por ejemplo, el envasado de pequeños bombones o caramelos en una caja, se requiere un robot que sea rápido, preciso y que agilice el proceso de producción. Además, se busca que sea eficiente respecto al consumo energético y sostenible con el medio ambiente.

1.4.1 Motivación profesional

Según el Ministerio de agricultura, alimentación y medio ambiente la industria de alimentación y bebidas es el primer sector industrial en facturación y un importante generador de empleo en nuestro país. Este sector ha logrado convertirse y mantenerse como el primer sector industrial en España. En los últimos años, el volumen que factura el sector supera los 90 mil millones de euros, lo que supone el 20,5% del total de las ventas netas de la industria nacional. Asimismo, aporta el 16,8% del Valor Añadido Bruto de la industria.

Las principales aplicaciones de la robótica y otros procesos de automatización en la industria alimentaria están aún por desarrollar. Si se consiguen reducir los costes y mejorar los niveles de calidad y seguridad alimentaria la robótica tendrá mucho peso en la industria alimentaria.

Según la Asociación Británica de Automatización Robótica la tecnología robótica ha tardado en penetrar en la industria alimentaria porque los sistemas de robótica y automatización no suelen incluir las características necesarias para las plantas de procesamiento de alimentos, como es la capacidad de lavado y eliminación de residuos. Además, los sistemas robotizados trabajan mejor con productos de tamaños y formas uniformes, y los alimentos no suelen reunir estas características.

El sector de la robótica se está adaptando a las necesidades generales de las empresas agroalimentarias, y está desarrollando sistemas de empaquetado y manipulación a alta velocidad para las líneas de procesamiento. Los avances de las tecnologías a un menor coste y el desarrollo de sistemas flexibles de robótica modular harán que cada vez resulte más rentable automatizar los actuales procesos manuales.

En el reportaje técnico “Robótica en la industria alimentaria” realizado por los institutos de investigación AZTI y FRATONIK de Tecnalia, la industria alimentaria se sitúa en un mercado sumamente competitivo y cada vez más global. Además la flexibilidad y la eficacia del proceso vienen dadas por la adaptabilidad del operario. Debido a la subida de los precios se debe potenciar la competitividad de la industria alimentaria europea, buscando soluciones para reducir los costes de producción y poder mantener el margen comercial, además de desarrollar nuevos productos alimenticios de calidad y seguros. Se requiere por tanto de una innovación en el diseño y control de los procesos, donde los robots presentan una buena alternativa para la automatización de los mismos. A pesar de las ventajas que pueden ofrecer los robots (alta flexibilidad, elevada productividad y baja tasa de errores, capacidad de trabajar en ambientes hostiles, etc.) existen algunas barreras económicas y técnicas asociadas a la variada naturaleza de los productos (frágiles y de geometría variable). Sin embargo, estas barreras se están eliminando gracias a la reducción de los precios de los robots, el aumento de coste de personal y al desarrollo de robots y herramientas de manipulación especializadas para aplicaciones alimentarias, que tienen en cuenta la variabilidad de productos a procesar. Además, se están haciendo colaboraciones entre tecnólogos de alimentos y diseñadores de robots, para facilitar la implementación de éstos en el sector de la alimentación.

Que esto es una realidad lo demuestran los datos del último informe del departamento de estadística de la federación internacional de robótica, donde se refleja que durante 2007 el incremento de venta de robots en el sector de la alimentación ha sido del 19% con respecto a 2006, estimando además que en los posteriores años las ventas de robots han ido creciendo.

Los puntos clave que presenta la automatización del sector alimentario serían los siguientes:

- **E**l sector alimentario en Europa debe mejorar su competitividad.
- **R**obots, la clave para la automatización de procesos alimentarios.
- **L**os robots aportan flexibilidad, seguridad, efectividad y elevada producción.
- **H**ay que romper con las barreras económicas y técnicas a la hora de incorporar robots.
- **H**ay que investigar en el desarrollo de efectores finales, programación y control.

Todos estos motivos expuestos anteriormente han dado lugar al desarrollo del presente trabajo.

1.4.2 Motivación innovadora

Haciendo hincapié en el último punto clave del apartado 1.4.1 hay que investigar en el desarrollo de efectores finales, programación y control. El presente proyecto trata de diseñar una herramienta modular la cual se puede construir con distintos dispositivos cuyas características pueden ser muy variadas. Este hecho da pie a buscar información sobre los distintos elementos a utilizar para el diseño de la pinza. Además, el control automático de la misma se puede realizar de varias maneras ofreciendo la posibilidad de utilizar diferentes dispositivos de control así como diferentes lenguajes de programación.

Las tareas a desarrollar se alejan mucho de la típica automatización de un proceso industrial común basado en un autómatas programable. Esto se debe tanto a la propia naturaleza de la aplicación, como al hecho de que actualmente no se ha desarrollado ningún sistema similar para entornos industriales.

1.4.3 Motivación personal

Mi motivación personal como ingeniero industrial, la oportunidad de trabajar en una temática puntera supone un empuje especial. El hecho de haber realizado este proyecto en el Instituto de investigación de automática e informática industrial (AI2) de la Universitat Politècnica de València, es una motivación importante de cara a mi futuro como profesional del campo de la automática, la robótica y el control de sistemas.

1.5 Objetivos.

Los sistemas diseñados para realizar tareas de manipulación y agarre, suelen estar implementados en procesos automatizados, partiendo de un entorno de trabajo fijo y perfectamente conocido. En este tipo de sistemas, cualquier cambio en las condiciones de trabajo, supone el poner en marcha un proceso de calibración para adaptarlos a las nuevas condiciones de trabajo.

El objetivo del presente trabajo es diseñar y desarrollar un pinza mecatrónica dinámica y modular, que permita la manipulación de alimentos en procesos automatizados multi-pick. Para el diseño se emplearán mecanismos y dispositivos industriales de control como guías lineales y motores paso a paso. La pinza se conectará a un robot industrial que se encargará de aproximar el sistema de manipulación a una zona determinada.

El grupo de la pinza dispondrá de una serie de ventosas para manipular varios productos. Una ventosa maestra se posicionará sobre un producto concreto. El resto de ventosas tendrán al menos dos grados de libertad, en configuración X-Y o polares $r-\alpha$ que permitirá a la misma situarse sobre otro producto libremente. La pinza alineará los productos y los depositará en una zona destinada a ellos.

Se dispone de un sistema de visión por computador ya implementado, a través del cual se realizará la elección del posicionado sobre los alimentos a manipular. Mediante un sistema de control se ejecutarán las acciones correspondientes que permitirán a la pinza mecatrónica recoger varios alimentos que provienen desordenados a través de una cinta transportadora y disponerlos de forma regular al depositarlos.

2. ESTADO DEL ARTE

2.1 Herramientas Multi-pick Fijas.

En la actualidad, este tipo de herramientas multi-pick es el que presentan mayor número de diseños e implementaciones en el mercado industrial a diferencia de las herramientas multi-pick dinámicas. La complejidad de su diseño se limita a dimensionar la pinza al número de objetos que se desee agarrar debido a que dichos productos se aproximarán al área de trabajo siempre con la misma disposición (esto es, con la misma distancia fija entre ellos). Dependiendo del dispositivo de agarre utilizado (agujas, pinzas de apertura y cierre, ventosas, etc) tendrá más o menos complejidad en su diseño, pero el control automático de la herramienta se realizará de manera similar a las herramientas de manipulación convencionales de un solo producto. Es obvio que a mayor número de productos a manipular de forma simultánea, más compleja será la herramienta y tendrá un mayor peso que presentará inconvenientes que deberemos tener en cuenta a la hora de elegir el robot.

2.1.1 *Multi-pick handling unit.*

Tecno pack es una empresa Italiana que se dedica a realizar proyectos, construir y suministrar máquinas envasadoras *flowpack* e instalaciones automáticas de envasado. En alguno de sus proyectos incorporan herramientas multi-pick fijas, como es el caso de la pinza *Multi-pick handling unit*.

La herramienta se compone de una hilera de periféricos cada uno de ellos con un sistema de agarre por agujas. Esta pinza (figura 2.1) se encarga de tomar madalenas recién producidas que provienen de una cinta transportadora, y depositarlas de manera organizada y en fila individual en una cinta transportadora para su posterior envasado (flow-pack).

Esta pinza está diseñada para acoplarla a un robot antropomorfo de 6 ejes y se utiliza principalmente para aplicaciones de envoltura, o en general de pick and place; sobre todo en los casos donde se solicite manipular más elementos a la vez. Concretamente la empresa ha decidido utilizar el robot antropomorfo IRB 2400 de ABB, disponible en tres versiones, con capacidad de controlar desde 5 hasta 16 kg de peso.

Para realizar el control se utiliza ABB S4C+ y su versatilidad se nota aún más utilizando la función *conveyor tracking*. Esta función permite al robot operar sobre elementos en movimiento, permitiendo mayores libertades en la fase de estudio y diseño de la instalación.

Mediante el control ABB S4C+, IRB 2400 puede introducirse en las líneas de envasado entrando en comunicación con los demás dispositivos presentes y se puede interconectar con el sistema de visión ABB, es decir el PickMaster.

Gracias al software PickMaster, el robot está en condición de tomar los productos del sistema de transporte en movimiento colocándolos en el interior de bandejas o blister, en movimiento, y controlando en tiempo real la distribución de la carga entre más de un robot.



Figura 2.1. Multi-pick handling unit de la empresa Tecno pack

2.1.2 Gripper FXP.

La empresa Alemana Schmalz es líder en el mundo de técnicas de automatización, de manipulación y de fijación. De entre sus números productos destaca una herramienta de manipulación multi-pick que utiliza un Sistema de ventosas de vacío para superficies con generador de vacío externo y válvula esférica de flujo. Mediante dispositivos de características innovadoras tales como el refuerzo de vacío para la optimización del flujo, válvulas de retención sin fugas y eyectores poderosos han conseguido diseñar una herramienta compacta y segura capaz de adaptarse a productos de diferentes características (Figura 2.2).

El producto innovador es ideal para procesos en los que la posición de recogida del producto a agarrar no está precisamente definido o en el que existe la necesidad de coger múltiples productos al mismo tiempo en un ciclo. El FXP es ligero (entre 2,6 y 8,4 kilos, dependiendo de la longitud) y lo hace perfecto para su uso con robots. Las ventosas individuales se pueden cambiar rápidamente gracias a una función de push-in. Están hechas de silicona con RoHS-FDA y son, por tanto, adecuadas para su uso en la industria alimentaria y de bebidas.

Este tipo de herramienta se puede acoplar tanto a robots antropomorfos como en los robots delta, siempre que soporten el peso de entre 2,6 y 8,4 Kg.



Figura 2.2. Gripper FXP de la empresa Schmalz

2.1.3 Robotic EOAT.

SAS Automation es una empresa Norteamericana que se dedica al diseño y fabricación de herramientas de manipulación modulares y sistemas de agarre para robots industriales. Actualmente fabrica herramientas de manipulación y agarre para las principales marcas de robots como KUKA, Yaskawa Motoman, ABB, Fanuc y otros.

En agosto de 2012 lanzó al mercado una pinza multi-pick para *muffins* y pasteles de diferentes formas y tamaños.

Esta herramienta contiene numerosas pinzas de agarre formadas por 2 agujas de acero inoxidable. Este sistema flexible es capaz de manejar 144 magdalenas y desplazarlas a los moldes para hornear, con la velocidad de 77760 por hora; y 1,4 millones de magdalenas por día.

Al ser una pinza pesada se debe acoplar a robots antropomorfos de 6 ejes como por ejemplo el robot Fanuc S420F que puede soportar una carga de 120Kg (figura 2.3).

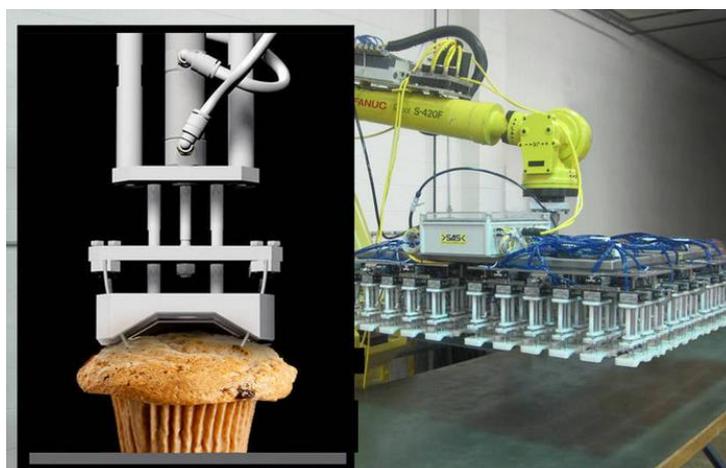


Figura 2.3. Herramienta multi-pik EOAT de la empresa SAS Automation

2.1.4 *Palletiser for small boxes.*

RoTeg mbH fue fundada como una oficina de consultoría de ingeniería en Dortmund en 1993 y se ha desarrollado hasta convertirse en una empresa establecida en el campo de la construcción de instalaciones y máquinas. Hoy en día, roTeg mbH es una empresa sólidamente establecida con muchas referencias en el dominio de la robótica y la ingeniería especial.

Como especialista en tecnología de paletización, roTeg ofrece soluciones de automatización completamente desarrolladas y con la tecnología de agarre correspondiente.

Entre sus diseños se encuentra la herramienta *Palletiser for small boxes* (figura 2.4) cuyas características se detallan a continuación:

- Garra de vacío de forma predeterminada con 2-8 ventosas.
- Ajustable debido a su estructura modular.
- Ajuste manual después del cambio de producto.
- Cierre rápido con posiciones de anclaje.
- Construcción robusta.

El concepto de control y por lo tanto la facilidad de uso del control son los criterios decisivos más importantes para un robot. En 1999, se desarrolló el sistema de control basado en PC PARO, que hoy se ha convertido en un sistema poderoso, y fácil de usar. Una navegación sencilla, y un sofisticado auto-diagnóstico permiten realizar la automatización del proceso.



Figura 2.4. *Palletiser for small boxes* de la empresa Roteg

2.1.5 Robots de paletización y embalaje RP4 y RCP.

La empresa Apsol es una compañía italiana especializada en el diseño y fabricación de maquinaria para el "fin de línea" de los procesos de envasado. Abarca una gran variedad de sectores como la alimentación y bebidas, el sector químico, aceites y condimentos, etc.

Los robots de paletización de la serie "RP4" y "RPC" completan la gama de los finales de línea Apsol. Están disponibles para sistemas de paletización en línea o múltiples con herramientas de agarre multi-pick para cajas de cartón, sacos, botellas de plásticos, etc. (figura 2.5).

El RP4 es un robot articulado característico por su gran flexibilidad, fiabilidad y simplicidad.

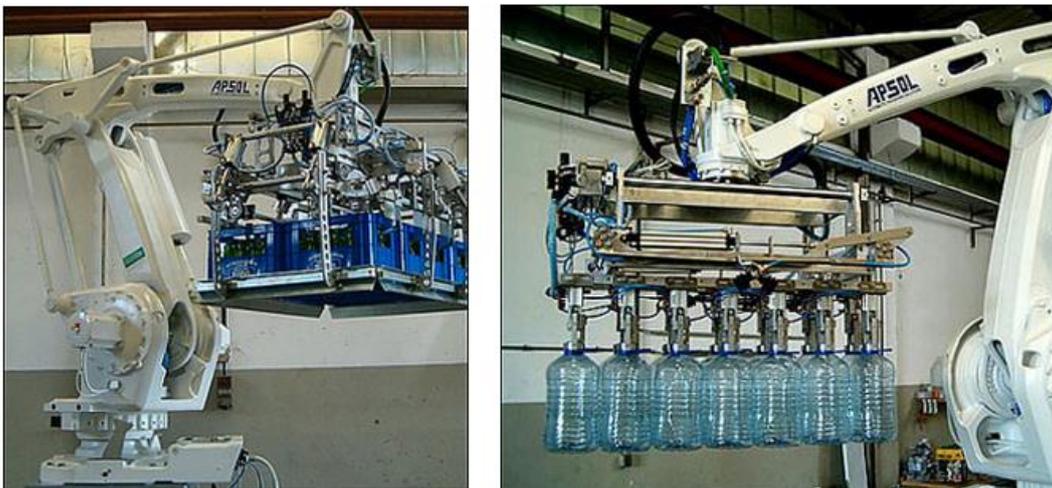


Figura 2.5. RP4 de la empresa Italiana Apsol.

2.2 Herramientas Multi-pick dinámicas.

A diferencia de los sistemas de manipulación multi-pick fijos, las herramientas multi-pick dinámicas están todavía en plena evolución. Como se ha mencionado en el apartado 1.3 el hecho de diseñar una pinza dinámica implica la inclusión de dispositivos adicionales que permitan orientarse a la posición de los productos en cada operación de *pick* y *place*. Esto hace que la herramienta sea más compleja y tenga un peso mayor, además del mantenimiento que requiere.

La herramienta dinámica debe ser rápida y en ocasiones precisa, por lo que también deben serlo sus dispositivos de posicionado. Esto hace que aumente el coste de la pinza.

A continuación se mostrarán ejemplos de diseños fabricados por empresas de robótica y automatización, debido a que no se han encontrado artículos de investigación para este tipo de pinzas.

2.2.1 Herramientas de manipulación por apilado de productos

Este tipo de herramientas de manipulación disponen de un dispositivo de almacenamiento donde se van apilando los productos a medida que se van ejecutando las operaciones de *pick*. Una vez se realiza el agarre de los productos estipulados en el control, se realizará la operación de *place* tal y como se realizan las operaciones de almacenaje LIFO (Último en entrar, primero en salir).

Estas herramientas son muy útiles para procesos de empaquetado y paletizado, ya que tienen la ventaja de realizar varias operaciones de *pick* y una sola operación *place* en cada ciclo, lo que permite un ahorro de tiempo notable.

2.2.1.1 *MultiPick Case cardboard gripper*

La empresa Finlandesa Cimcorp es el proveedor líder de sistemas de pódico robótico de gran superficie para la intra-logística. Suministran soluciones robóticas para la gestión de los flujos en los entornos de producción y distribución en la industria de los neumáticos, industria al por menor, de alimentos y bebidas, y del sector postal.

En septiembre de 2010, desarrolló una solución de picking para satisfacer la necesidad de cumplir necesidades de orden y rapidez en el almacenamiento y organización de productos, con costes reducidos, mayor capacidad, mayor precisión y una mejor trazabilidad y logística en aplicaciones del sector alimentario.

Para la recolección de productos, el robot se va moviendo y posicionando hacia los elementos a apilar. La herramienta y el robot pueden trabajar en diferentes modos de operación alternativos tales como "pre-selección" y "soltar", con el objetivo final establecido por él. *Multipick case* también puede tener en cuenta reglas de apilamiento especiales exigidas. Por ejemplo, los minoristas pueden especificar que los productos más pesados se coloquen en el fondo de la pila de almacenamiento para facilitar su reposición en tiendas.

La herramienta es controlada por el software de control de depósito (WCS) que supervisa las funciones de almacenamiento requeridas, incluyendo el control de la preparación de pedidos y de flujo de materiales. El sistema WCS (que cuenta con una interfaz gráfica de usuario intuitiva) siempre opera bajo el sistema de host del cliente, de la que recibe todos los datos pertinentes, tales como productos, órdenes de línea, los volúmenes, las prioridades y los plazos de entrega.

Esta herramienta en concreto presenta un gran volumen y por ello, se suele acoplar a sistemas gantry como son los puentes grúa (figura 2.6) que son capaces de soportar grandes pesos. Concretamente la *Multipick case* es capaz de soportar 120kg de peso, que unidos al peso de la herramienta suman una cantidad de peso considerable.



Figura 2.6. Multipick case de Cimcorp.

2.2.1.2 Robocollect y SG47

Otra pinza por apilado de productos pero en este caso para productos pequeños y de poco peso es la que se diseñó en 2007 por la empresa Alemana Robomotion GmbH. Esta contiene una pinza para el apilado de productos y otra pinza mecánica (figura 2.7a). La herramienta puede agregar los productos a la pila, de manera que la posición y orientación de los productos pueden provenir de manera aleatoria, por lo que en combinación con un robot, los productos pueden ser incluidos en la pinza desde cualquier posición. Además, al realizar la operación place los productos pueden ser depositados con una flexibilidad elegida o dejarlos caer de manera apilada.

El procedimiento de trabajo es el mismo que el explicado en el apartado 2.2.1.1 con la diferencia de que para esta pinza de manipulación se utiliza un robot delta en lugar de un puente grúa. En este caso una cámara detecta las posiciones de los productos sobre la cinta transportadora y pasa los datos al robot, que mediante la rotación del cuarto eje realiza el giro adecuado para orientarse hacia la posición del producto a manipular.

De la misma forma trabaja la herramienta multi-pick SG47 diseñada en conjunto con la empresa alemana Schunk (Figura 2.7b). Esta herramienta se utiliza para productos alimenticios y está realizada mediante plásticos y aceros aptos para este tipo de productos. La apertura y cierre de la pinza se realiza con tiempos de ciclo cortos (0.05 seg) permitiendo ejecutar 145 ciclos por minuto. Algunas de sus características se detallan a continuación:

- Fuerza de agarre de 6 Newtons (fuerza elástica).
- Rango de presión de 5.4 a 5.6 bar.
- Ángulo de apertura 12 ° y 35 °
- Temperatura de funcionamiento 0 ° a 80 ° C
- Peso de 0,375 kg

- Control neumático



Figura 2.7a. Robocollect



Figura 2.7b. SG47

2.2.2 Herramienta Multipick Salami-Picker

En el año 2006 la compañía Alemana Robomotion GmbH junto con la empresa Schunk, consiguieron solucionar un problema de automatización del sector alimentario de salchichas alemanas. Diseñaron una herramienta multi-pick apta para trabajar con robots delta IRB 340 FlexPicker de ABB. La pinza (figura 2.8) dispone de una tecnología capaz de tomar seis tipos diferentes de salchichas de la cinta transportadora e insertarlas en el destino establecido. La tecnología de agarre mecánico ayudó a aumentar la fiabilidad del proceso considerablemente, aumentando así el volumen de producción por robot.

El proceso de automatización para el envasado de las salchichas consiste en, mediante un sistema de visión por computador se procesan los datos de posicionamiento y localización de los productos que provienen a través de una cinta transportadora. La herramienta multi-pick insertada en el robot está compuesta por una pinza triple capaz de recoger tres salchichas que provienen con posición y orientación aleatoria. Las salchichas se agarran en el orden especificado por el sistema de visión y se colocan de manera simultánea en otra cinta transportadora que depositará las salchichas en la máquina de envasado. Los robots delta IRB 340 FlexPickers operan a alta velocidad en la línea de envasado. Mediante el software desarrollado se permite un suministro óptimo de los datos de posicionamiento del procesamiento de imágenes para estos robots, por lo que se pueden ejecutar operaciones con la tasa de utilización óptima.

La herramienta dispone de tres cilindros neumáticos con una pinza de agarre para cada salchicha, y dependiendo del orden establecido actúan uno u otro. El orden de agarre del producto presenta una vital importancia, ya que debido a la orientación y posición de las salchichas éste puede variar en cada ciclo de operación, por lo que el software de procesamiento de datos debe ser rápido y efectivo.



Figura 2.8. Multipick Salami-Picker

2.2.3 Mitsubishi Electric Single and Multi Pick Robot

Recientemente, con equipos de fabricación de primera clase y un departamento de ingeniería especializada, desde 1977, la I + D Custom Automation de la compañía Japonesa Mitsubishi Electric ha proporcionado soluciones de estado del arte en integración de sistemas de automatización y modernas instalaciones de tecnología de vanguardia y de primera línea. Uno de los proyectos realizados se basó en incorporar la tecnología de herramientas pick and place en sus robots y diseñó un sistema automatizado complejo conocido como 'RoboPick'.

El sistema combina la rigidez y la flexibilidad mediante sus últimos robots SCARA que ayudan a minimizar el movimiento de trayectoria y logrando un mayor rendimiento para productos que tiene orientaciones aleatorias.

El sistema de visión Cognex In-Sight 7000 detecta las posiciones (coordenadas X e Y) y la orientación (ángulo) de los productos que proceden de una cinta transportadora. Un robot SCARA RH-3SDHR se mueve en sincronía con la cinta transportadora agarrando los productos de uno con el fin de colocarlos en la cinta transportadora de salida. Los productos se trasladan hacia otro robot SCARA RH-3FH35 que dispone de una pinza multi-pick (figura 2.9) que recoge tres productos de uno en uno orientando sus dispositivos de agarre y, seguidamente coloca de nuevo los productos simultáneamente al azar en la primera cinta transportadora.

El sistema de automatización Mitsubishi Electric RoboPick contiene también una herramienta de calibración para la visión artificial (Cognex In-Sight) y una interfaz de explorador. (EasyBuilder).

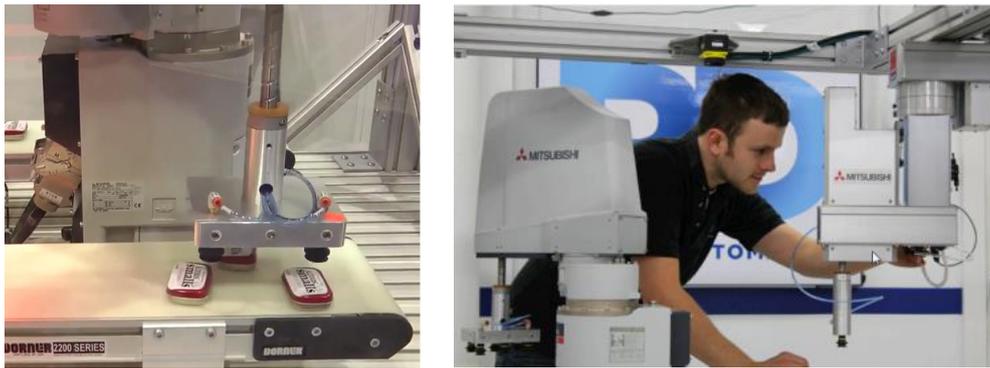


Figura 2.9. Sistema de automatización Robo-pick

2.2.4 Multi gripper system

La descarga automática de piezas en bruto es un proceso importante en la producción de insertado de piezas y otros productos que requieren precisión.

Para este caso la compañía Alemana *Metzner* tiene desarrolladas varias herramientas multi-pick. La primera de ellas (figura 2.10a) consiste en una herramienta con 4 cilindros neumáticos alineados que agarran piezas ubicadas en distintas ubicaciones para depositarlas de manera alineada en una cinta transportadora. Esta pinza se inserta en un robot SCARA de 4 ejes que, guiado por un sistema de visión, será el encargado de realizar los movimientos de recarga de piezas.



Figura 2.10a.

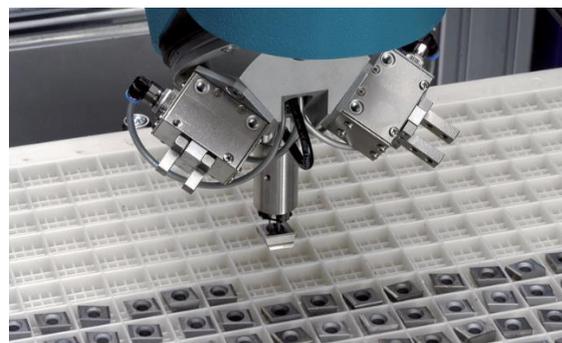


Figura 2.10b.

La segunda herramienta (figura 2.10b) tiene la misma finalidad que la anterior, solo que dispone de un dispositivo de agarre menos y su distribución es diferente. Posee una pinza de agarre para coger piezas de un tipo y otras dos de características idénticas para poder coger piezas de disposición geométrica similar. Al igual que la herramienta anterior, esta está preparada para ser insertada en un robot SCARA de 4 ejes.

2.3 Dispositivos de posicionamiento.

Como se ha mencionado en los apartados anteriores, el hecho de diseñar una herramienta multi-pick dinámica y modular requiere incorporar sistemas y dispositivos que permitan posicionarse en las coordenadas (lineales y angulares) del producto, que como se ha dicho, pueden acercarse a la herramienta de manipulación a través de la cinta transportadora con una disposición aleatoria.

Para realizar el diseño de la herramienta se ha estudiado la posibilidad de utilizar diferentes sistemas de posicionamiento capaces de, mediante la combinación de estos, poder elaborar un sistema único que conforme la herramienta multi-pick.

Los sistemas de posicionamiento más conocidos y utilizados en la industria son los dispositivos de posicionamiento lineal, y los angulares o rotacionales. Además, para asegurar la correcta posición de éstos, se deben utilizar sensores que permitan cerrar el lazo de control y asegurar así que la posición objetivo es la adecuada en cada momento.

Otro de los aspectos importantes a la hora de elegir un mecanismo de posicionamiento es la carga que se debe trasladar, debido a que dependiendo de ella, los dispositivos a utilizar variarán en tamaño y especificaciones.

2.3.1 Sistemas de posicionamiento lineal.

Los sistemas de posicionamiento lineal son dispositivos de acción o actuadores inherentemente mecánicos. La función que tienen estos sistemas es proporcionar una fuerza para mover otro dispositivo mecánico de manera lineal. Dependiendo del origen de la fuerza de estos dispositivos se pueden denominar:

- Neumáticos
- Hidráulicos
- Eléctricos

Para una Herramienta de manipulación multi-pick que tiene como objetivo ser modular, es decir, que se le puedan añadir dispositivos de agarre a medida que se requieran tomar más productos, es necesario que esté compuesta por actuadores de poco peso y, dependiendo de la aplicación, se requerirá también alta velocidad y alta precisión.

Tras realizar una amplia investigación sobre dispositivos lineales industriales disponibles en el mercado, los más innovadores y sofisticados se presentarán en los apartados siguientes.

2.3.1.1 Guías lineales de un solo eje.

Se pueden encontrar actuadores lineales tanto eléctricos como neumáticos capaces de realizar movimientos precisos a alta velocidad. Los actuadores eléctricos disponen de un husillo o unos rodamientos que acoplados a un motor, realizan la traslación del objeto a posicionar.

En la figura 2.11 se puede observar una guía lineal eléctrica fabricada por la empresa *Newmark systems*. Estos actuadores disponen de una guía de aluminio donde se incorpora un carro que, a través de un motor eléctrico, se posiciona en la ubicación objetivo mediante una tarjeta controladora. Las características más destacadas son:

- Rangos de posición desde 50mm hasta 300mm.
- Una resolución de hasta 1.3 μm , y una precisión de 0.0012mm
- Velocidad de entre 20mm/segundo y 200mm/segundo.
- Capaz de soportar una carga de 4.5Kg
- Peso de 0.58Kg.



Figura 2.11. Guía lineal eléctrica MSL de Newmark Systems

Los actuadores lineales neumáticos los podemos encontrar de muchos tipos, ya sean cilindros con vástago o sin vástago, pistones de simple o doble efecto, etc.

Aunque en la actualidad en el mundo de la neumática se está consiguiendo un control de posición bastante efectivo, por ejemplo, mediante reguladores (sensores) de flujo, cabe destacar que la utilización de este tipo de actuadores requiere de mayor investigación para los objetivos de este proyecto. No obstante, en la figura 2.12 se puede observar un actuador neumático de la compañía SMC donde algunas de sus características se detallan a continuación.

- Presión de trabajo de 0.2 a 0.55 MPa.
- Temperatura ambiente entre -10 y 60°C.
- Tiempo de respuesta 1.2ms.
- Rango de trabajo desde 50mm hasta 400mm.
- Peso máximo de carga admisible de 2Kg.



Figura 2.12. Guía lineal Neumática MXY6 de SMC.

El mayor inconveniente de los actuadores mencionados anteriormente es el peso, debido a que para conseguir posiciones en dos dimensiones deberíamos insertar otra guía lineal acoplada a la primera, cosa que incrementaría el peso de la herramienta completa, además para cada objeto a coger se requeriría otros dos sistemas más, y así sucesivamente. Al final, se necesitaría un robot más robusto y con mayor capacidad de carga, lo que limita las velocidades que se necesitan en una aplicación de pick & place.

2.3.1.2 Guías lineales de dos o más ejes.

Aunque las guías mostradas en el apartado anterior pueden ser acopladas con la posibilidad de conseguir un sistema de mayor dimensión (figura 2.13), existen en la industria sistemas lineales de dos y tres dimensiones que permitirían ser acoplados mediante un sistema mecánico al eje de herramienta de un robot.

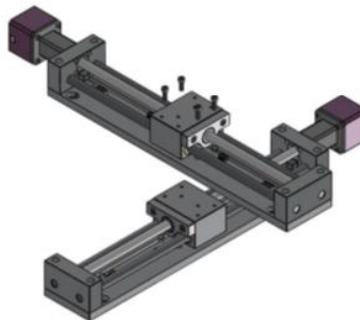


Figura 2.13. Ejemplo de acople de dos guías para realizar un sistema lineal en dos dimensiones.

La empresa alemana Festo dispone de sistemas cartesianos de dos y tres dimensiones (figura 2.14). El dispositivo está compuesto por un pórtico compacto de gran funcionalidad, para el montaje en espacios muy reducidos. Además, el diseño de los actuadores minimiza la masa móvil. Dispone también de un controlador electrónico donde se logra una perfecta compatibilidad entre actuador y controlador. El posicionado del carro se consigue mediante el accionamiento de dos motores paso a paso con encoder óptico incorporado (permitiendo la

realimentación en bucle cerrado) y un controlador de dos ejes, adaptado al sistema. Se puede utilizar en dos modos de funcionamiento, el primero de manera directa a través de buses Ethernet y CAN, y el segundo mediante la selección de registros de E/S digitales.

Éste sistema está equipado para procesos industriales de alimentación, prensado, unión de piezas, dosificación de líquidos, montaje de componentes electrónicos, etc.



Figura 2.14. Pórticos en 2D y 3D de la empresa Festo.

Las especificaciones más destacadas de éste sistema son:

- Áreas efectivas de trabajo desde 10 hasta 252cm²
- Cargas útiles hasta 3Kg.
- Precisión de repetición desde 0.1 hasta 0.05mm.
- Velocidad máxima de trabajo de 0.5m/s.
- Aceleración máxima de 10m/s².
- Peso total de la herramienta desde 2.7Kg hasta 5.7Kg dependiendo del área de trabajo.

En la figura 2.15, se puede observar otro mecanismo de posicionamiento en 2 dimensiones capaz de trabajar a altas velocidades. Este dispositivo diseñado por la empresa alemana *A-drive* es activado a través de motores lineales, lo cual permite que su composición sea de menor tamaño comparado con otros sistemas.

El dispositivo funciona acompañado de un controlador electrónico que permite repuestas rápida respuesta y movimientos precisos. Con un codificador de alta calidad se consiguen resoluciones de hasta 0.005 micras con movimientos silenciosos. Algunas de sus características se detallan en los siguientes puntos:

- Áreas efectivas entre 12 y 145cm².
- Peso entre 3.6Kg y 25Kg dependiendo del área de trabajo.
- Cargas útiles desde 1.8 hasta 20Kg.



Figura 2.15. Sistema lineal de dos dimensiones con motores lineales de la empresa A-drive.

2.3.2 Sistemas de posicionamiento angular.

Los mecanismos de posicionamiento angular al igual que los lineales son dispositivos de acción. La función que tienen estos sistemas es proporcionar una fuerza para mover otro dispositivo mecánico de manera rotacional. Dependiendo del origen de la fuerza de estos dispositivos también se pueden denominar hidráulicos, eléctricos y neumáticos.

Se puede considerar que los dispositivos rotacionales más utilizados en la industria son los motores, de entre los cuales se pueden encontrar diferentes tipos. No obstante existen mecanismos de rotación angular que se utilizan para posicionar elementos en una orientación deseada.

2.3.2.1 Motores.

El uso de motores como sistemas de posicionamiento requiere un análisis especial ya que su selección y uso no es aleatorio, sino que el tipo de motor que se desee para el dispositivo depende de la función que se vaya a ejecutar, lo mismo con el control del motor, este dependerá de diversos factores uno de ellos es donde se desempeñe el sistema completo.

El propósito de diseño de la pinza modular es que los dispositivos de posicionamiento de los que disponga sean de tamaño y peso reducido, con lo cual, para este trabajo es importante hacer un especial hincapié en los motores eléctricos utilizados hoy en día en la industria robótica y en concreto en las herramientas de manipulación que se acoplan al robot.

Para utilizar los motores como dispositivos de posicionamiento angular en estas herramientas, se destacan como motores paso a paso, los motores brushless y los servomotores.

- **Motores Paso a paso:** Convierte una serie de impulsos eléctricos en desplazamientos angulares discretos. El motor paso a paso se comporta de la misma manera que un convertidor digital-analógico (D/A) y puede ser gobernado por impulsos procedentes de sistemas lógicos. Este motor presenta las ventajas de tener alta precisión y repetitividad en cuanto al posicionamiento.

Los motores paso a paso, se controlan por el cambio de dirección del flujo de corriente a través de las bobinas que lo forman. Se controla el desplazamiento del rotor en función de las tensiones que se aplican a las bobinas, con lo que se pueden conseguir desplazamientos en ambos sentidos de rotación. Además, también se puede controlar el número de pasos por vuelta, así como la velocidad del motor.



Figura 2.16. Motor paso a paso y tabla de la verdad típica de control.

Estudios recientes han demostrado que se pueden realizar controles de posición y velocidad mediante controladores avanzados y teorías de control modernas. Un ejemplo de control es el realizado por R. Marino, S. Peresada y P. Tomei en el artículo “*Nonlinear adaptive control of permanent magnet step motors*” donde demuestran que se puede realizar un control adaptativo no lineal que garantiza el seguimiento asintótico de una señal de referencia angular, realimentando el conjunto de mediciones de estado (posición, velocidad y corrientes), cuando los únicos parámetros conocidos son el par de la carga y la resistencia del estator. Las simulaciones realizadas en esta investigación muestran el rendimiento del algoritmo adaptativo y la comparan con el control PID convencional.

- Servomotores:** Son un dispositivos similares a los motores de corriente continua que tienes la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición. El servomotor espera un tren de pulsos que se corresponde con el movimiento a realizar. Están generalmente formados por un amplificador, un motor, un sistema reductor formado por ruedas dentadas y un circuito de realimentación, todo en una misma caja de pequeñas dimensiones, por lo que es muy utilizado en aplicaciones de robótica.

Al igual que los motores paso a paso, ofrecen muchas ventajas para ser controlados y se han publicado numerosos artículos de investigación que utilizan teorías de control moderno y controladores sofisticados para conseguir controles de posición y velocidad efectivos. Uno de ellos es el publicado por Chung Yuan y Chung Li denominado “*Adaptive fuzzy sliding-mode control for PM synchronous servo motor drives*”. En él, se diseñó un sistema de control adaptativo con lógica difusa y en modo deslizante, para realizar un control de posición de un servomotor síncrono de imanes permanentes.



Figura 2.17. Ejemplos de servomotores.

- Motores brushless:** Este tipo de motores no disponen de escobillas ni colector y tampoco de delgas; por lo que el elemento que controla que el rotor gire sea cual sea su posición es un variador electrónico. Este variador analiza la posición en la que se encuentra el rotor en cada momento, para hacer que la corriente que le llegue sea la adecuada y provocar así el movimiento de rotación que le corresponde. Este tipo de motores se utiliza cada vez más en dispositivos electrónicos y en robótica, ya que poseen características importantes como alto par y altas velocidades (dependiendo del motor). Existen numerosas técnicas de control para este tipo de motores, de entre las cuales se destacan las técnicas de control trapezoidal, sinusoidal y vectorial. En el artículo “*Técnicas de control para motores Brushless*” de Roger Juanpere Tolrà, se realiza una comparativa entre estas tres técnicas de control analizando su principio de funcionamiento así como sus prestaciones y/o limitaciones.



Figura 2.18. Ejemplos de motores brushless.

2.3.2.2 Mesas giratorias.

Otro de los dispositivos de posicionamiento angular que se usa en la automatización industrial son las mesas giratorias (rotary table). Estos mecanismos son ampliamente utilizados para realizar mecanizados y se colocan en máquinas de torneado, fresado, rectificado, etc. No obstante, se están utilizando también en sistemas robotizados.

Estas mesas están compuestas por un motor y por engranajes helicoidales con holgura reducida, lo que permite actuaciones sin impactos a alta velocidad y repetitividad de posicionamiento. Están preparadas para trabajar en rangos desde 0 a 360°, aunque la mayoría de ellas trabajan en un rango de ángulos inferior.

En la figura 2.19 se puede ver un ejemplo de una mesa rotacional eléctrica de la empresa *SMC*, que se utiliza para fines de automatización industrial, y cuyas características son las siguientes:

- Par máx. de giro entre 0.2 y 10 N·m.
- Ángulo de giro hasta 360°.
- Peso de 0.5Kg.
- Repetitividad de $\pm 0.01^\circ$.
- Servomotor de 24V.
- Velocidad angular desde 72 a 420°/s.

Esta mesa dispone de un controlador encargado de asegurar el correcto posicionamiento del mecanismo. El controlador se comunica por un puerto serie RS485 (según protocolo Modbus), dispone de control de bloqueo y sistema refrigerador así como de 11 entradas y 13 salidas en paralelo.



Figura 2.19 Rotary table LER10 de SMC.

3. EXPLICACIÓN DEL PROBLEMA

El problema parte desde una cinta transportadora, donde se irán depositando de manera aleatoria y desordenada los objetos a manipular. Un sistema de visión por computador identificará los productos y mediante un algoritmo de procesamiento de datos se determinarán de manera óptima los elementos a recoger por la pinza multi-pick.

Dependiendo de los N módulos de agarre que disponga la pinza, se podrán agarrar los N productos determinados por el algoritmo implementado en el sistema de visión. Las técnicas empleadas en visión artificial determinarán las coordenadas de los productos a recoger. Estos productos serán objetos con el mismo tamaño y de forma geométrica circular. Es importante conocer la geometría de los productos a agarrar, ya que dependiendo de ella cambiarán las características y especificaciones de la herramienta.



Figura 3.1. Izquierda: Ejemplo de transporte de galletas con cinta transportadora. Derecha: Sistema industrial de visión por computador

Una vez determinadas las coordenadas de posición de los elementos, éstas se enviarán a un robot industrial donde le ha sido colocada la herramienta multi-pick. El siguiente paso será la realización de la operación *pick* donde se recogerán los N objetos determinados. Seguidamente, los productos deberán ser transportados a otra cinta transportadora donde posteriormente se realizará el envasado del producto. En el transcurso del transporte del producto desde la cinta transportadora principal a la cinta transportadora objetivo, se deberá realizar la reubicación de los productos para poder ejecutar la operación de dejado *place*. La reubicación consistirá en colocar los productos de manera ordenada y con una distribución predeterminada, de tal manera que se depositen en el envase de empaquetado.

Una vez se haya ejecutado la operación *place*, se procedería a realizar de nuevo la operación de *pick* donde previamente se habrán procesado los productos a agarrar, y de forma cíclica se repetirían cada uno de los pasos explicados.



Figura 3.2. Ejemplo de operación pick and place con pinza multi-pick fija.

Como se ha mencionado anteriormente, la herramienta multi-pick estará formada por N módulos que podrán coger los N productos establecidos. Todo el sistema al completo se debe acoplar a la muñeca del robot donde se encuentra el punto central de la herramienta de éste, (TCP, *Tool Center Point*). Cada módulo incorporado en la herramienta multi-pick deberá tener un TCP, el cual será controlado independientemente para que se ubique en la posición de cada producto a agarrar.

A cada punto de control TCP, se le insertará una ventosa neumática que será la encargada de agarrar el producto. Además, cada TCP se debe desplazar realizando un movimiento determinado por dos coordenadas que bien pueden ser cartesianas (x, y) o polares (r, α) dependiendo de cómo se diseñe la pinza. Estos desplazamientos describirán un área de trabajo, siendo la máxima área efectiva la que determinarán los dispositivos de posicionamiento utilizados.

Para hacerse una idea del problema que se ha presentado, se puede observar el siguiente caso a modo de ejemplo en el que se pretenden coger 4 elementos a la vez, mediante 4 módulos cartesianos sin tener en cuenta el área efectiva y el peso de la herramienta total.

- **Ejemplo de aplicación con una herramienta multi-pick de 4x4:**

Cada módulo tendrá la distribución mostrada en la siguiente figura, donde se presenta el área efectiva en amarillo y el centro del punto de control (TCP) en rojo:

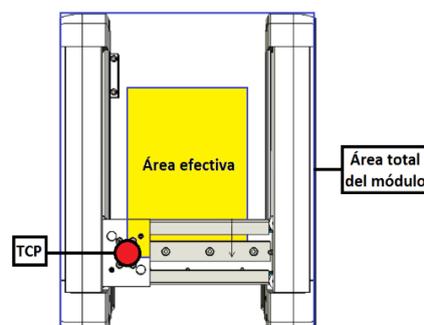


Figura 3.3. Módulo de posicionamiento x-y.

Como se observa, el módulo está compuesto por dos guías lineales las cuales están montadas una sobre la otra. Por ejemplo, la primera guía doble sería la responsable de conseguir el desplazamiento en el eje Y, mientras que la segunda guía simple, montada sobre la primera, consigue el desplazamiento en X. Sobre esta última guía va montado el soporte donde se ubicaría la ventosa que a su vez, sería el centro del punto de control.

Gracias al desplazamiento del TCP, podemos hacer coincidir el TCP de cada módulo con cada objeto permitiendo así que la herramienta sea dinámica y de gran versatilidad.

Como se van a disponer cuatro módulos, la herramienta completa 4x4 se detalla a continuación donde se indica el centro TCP del robot:

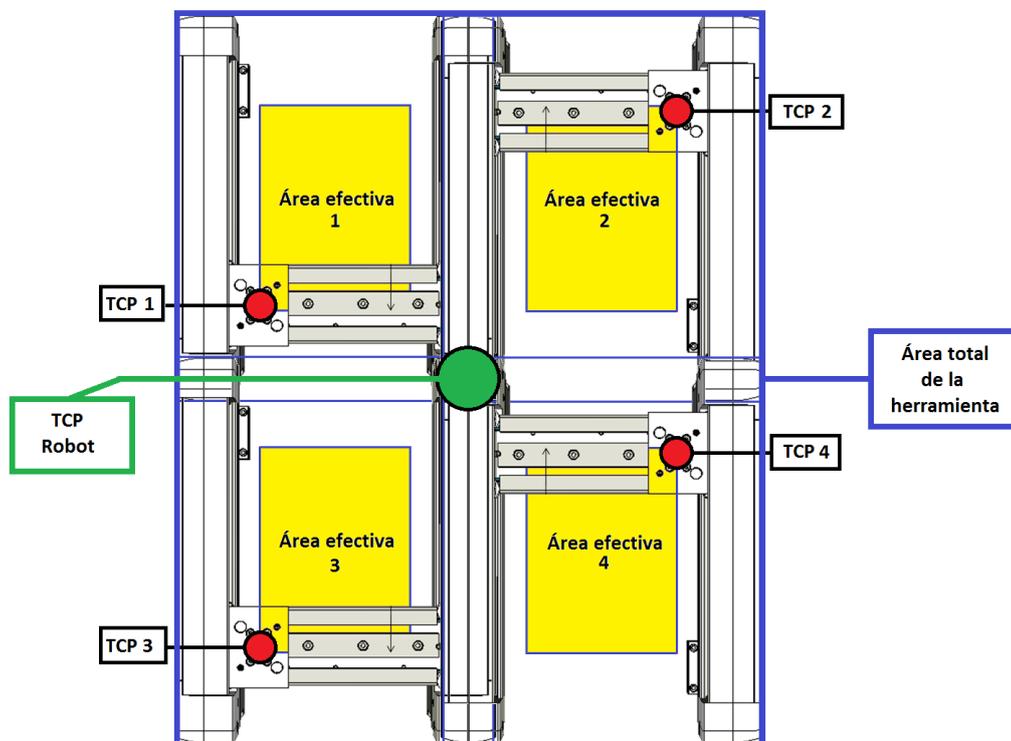


Figura 3.4. Ejemplo de herramienta con 4 módulos de posicionamiento x-y.

En la figura anterior se observan 4 áreas de trabajo, una por cada módulo. Con esta distribución, se podrían seleccionar 4 objetos, cada uno ubicado en un área de trabajo distinta, y mover los TCP para que coincidan con el centro de cada objeto. Así, se procedería a realizar el multi-pick de los 4 objetos simultáneamente.

Es entonces cuando, una vez agarrados los 4 objetos, el robot ejecutaría el movimiento para transportar los elementos hasta la posición deseada mientras de manera simultánea se reubican los objetos a la posición establecida de dejada, ejecutando a posteriori la operación *place*.

En la siguiente imagen se ha realizado una secuencia del proceso descrito, donde se describen los pasos a realizar de manera visual. En la imagen se observan los objetos redondos de color naranja, y los seleccionados por el sistema de visión se han marcado en verde. En un primer instante (paso 1) los objetos se acercan a la herramienta multi-pick por la cinta transportadora. En el paso 2 se puede apreciar como los TCP de cada módulo se posicionan en las coordenadas de los objetos seleccionados y posteriormente en el paso 3 se reorientan a la posición establecida de dejada. Ya en el paso 4 se puede ver como se depositan los elementos en la cinta transportadora donde se empaqueta el producto.

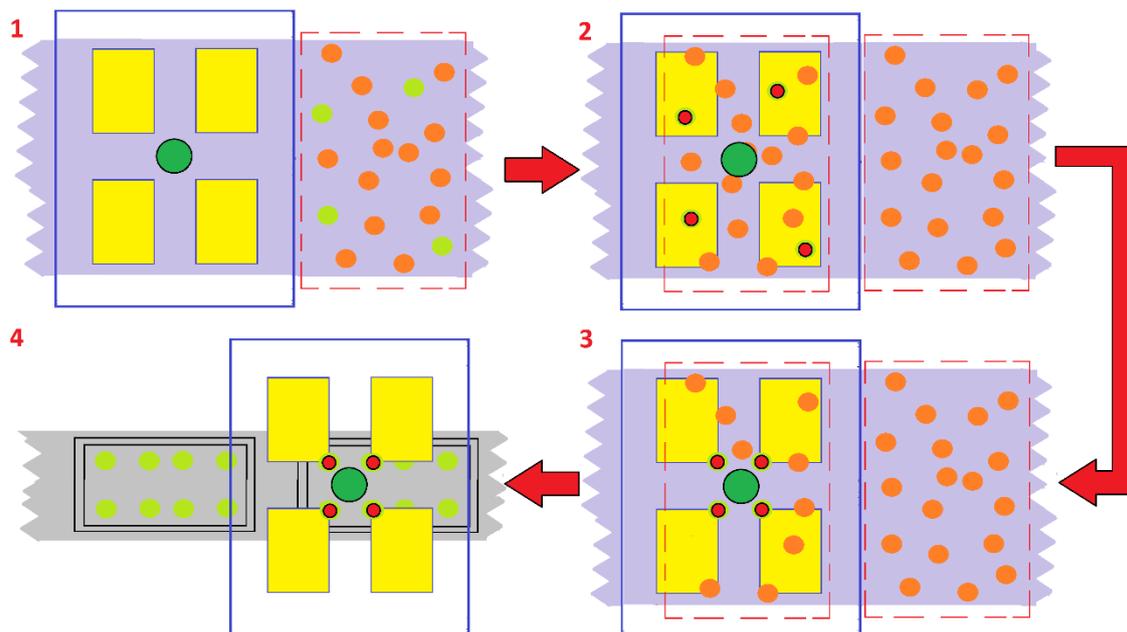


Figura 3.5. Pasos de ejecución del proceso pick and place con 4 módulos de posicionamiento x-y

Más adelante en el punto 3.2 se analizarán las diferentes alternativas de distribución de estos módulos, así como la solución adoptada para cada uno de ellos.

3.1 Alternativas para el diseño de la herramienta.

3.1.1 Herramienta multi-pick doble.

Una de las alternativas para el diseño de la pinza más simple que se puede considerar es la posibilidad de disponer de un solo módulo de posicionamiento. Este módulo se puede acoplar de forma paralela a la muñeca (TCP) del robot de manera que se obtenga una configuración donde se podrían coger dos productos simultáneamente. El primero de ellos se cogería

mediante la ventosa colocada al TCP del robot industrial y sería un TCP fijo. El segundo se capturaría mediante la ventosa conectada al TCP móvil del módulo.

Esta configuración tiene dos variantes, la primera sería la posibilidad de abarcar un área cuadrada o rectangular mediante coordenadas cartesianas. Para la segunda, se aprovecharía el eje actuador del robot para hacer girar la herramienta, abarcando así un área de trabajo circular mediante coordenadas polares.

A continuación, se muestra mediante un esbozo la configuración cartesiana:

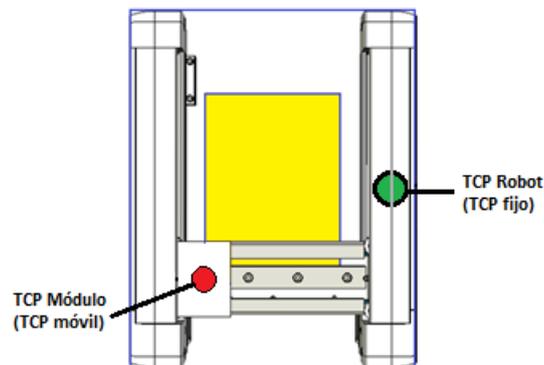


Figura 3.6. Módulo de posicionamiento x-y conectado al actuador final del robot.

En la figura anterior se dispone de un módulo x-y que contiene el TCP móvil. Mediante un elemento mecánico se colocaría el módulo cartesiano acoplado al eje actuador del robot, donde una posible configuración podría ser la mostrada en la figura. Basándose en el TCP fijo, la aplicación de visión por computador procesaría las imágenes captadas por la cámara determinando los dos objetos a coger por la herramienta.

Este diseño permite coger dos elementos simultáneamente siendo una solución simple, y con un bajo coste. No obstante, se observa una pequeña separación entre el área efectiva del módulo y el punto fijo que establece el actuador del robot. Esto implica una limitación en cuanto a la operación *place* del proceso ya que la distancia entre objetos tendría una separación mínima que dependería de las medidas y especificaciones del sistema cartesiano.

Una alternativa a la configuración anterior sería la mostrada en la figura 3.7. Para este caso se utilizaría una simple guía lineal que podría mover el TCP móvil a través de un motor eléctrico. La guía se acoplaría al robot mediante un elemento mecánico. A diferencia de la configuración cartesiana esta alternativa permitiría, a través de coordenadas polares, abarcar un área circular o semicircular dependiendo de su diseño.

Al igual que la configuración anterior se observa una pequeña separación entre el área efectiva del módulo y el punto fijo. Sin embargo, el hecho de utilizar una sola guía lineal simplifica el control, abarata el coste y lo más importante, disminuye el peso total de la herramienta.

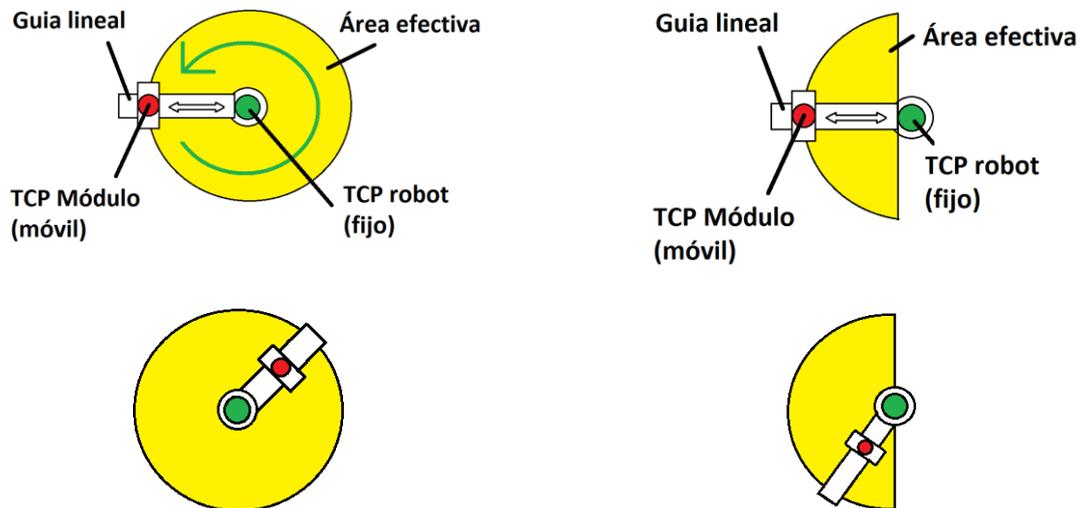


Figura 3.7. Herramienta multi-pick doble con coordenadas polares.

Para esta configuración, el hecho de utilizar el eje actuador del robot para hacer girar la guía lineal presenta una limitación importante. Como se ha indicado en el punto anterior, la construcción de la herramienta en este proyecto está destinada para manipular productos alimentarios de geometría circular y con el mismo tamaño, como podrían ser por ejemplo galletas. Para este tipo de objetos no presentaría inconveniente, no obstante si se quisiera manipular elementos de constitución geométrica no circular y además distinta, la orientación de éstos podría variar ya que los productos se sitúan en la cinta transportadora de manera aleatoria y, por lo tanto, no se podría utilizar el eje actuador del robot para hacer girar la guía lineal.

La solución pasaría por incorporar un motor adicional que haría el giro de la guía a la posición angular indicada, aunque esta solución incrementaría la distancia entre el punto fijo y el punto móvil del robot, perdiendo efectividad en el empaquetado del producto.

3.1.2 Herramienta multi-pick triple.

Otra de las alternativas de diseño para la herramienta consiste en incorporar otro módulo de posicionamiento móvil a la herramienta simple descrita en el punto 3.1.1, de tal manera que se puedan coger 3 objetos simultáneamente.

Al igual que en la herramienta doble, es posible realizar el diseño mediante dos configuraciones diferentes; cartesiana o polar.

Si se decide realizar el diseño de la herramienta utilizando áreas cuadradas o rectangulares mediante el posicionamiento de los TCP móviles con coordenadas cartesianas, se podría utilizar una distribución como la que se presenta en la figura 3.8, donde se colocan dos módulos x-y consecutivos. Estos dos módulos se acoplarían al eje actuador del robot consiguiendo obtener 3 puntos de agarre, los dos TCP móviles y el TCP fijo del robot.

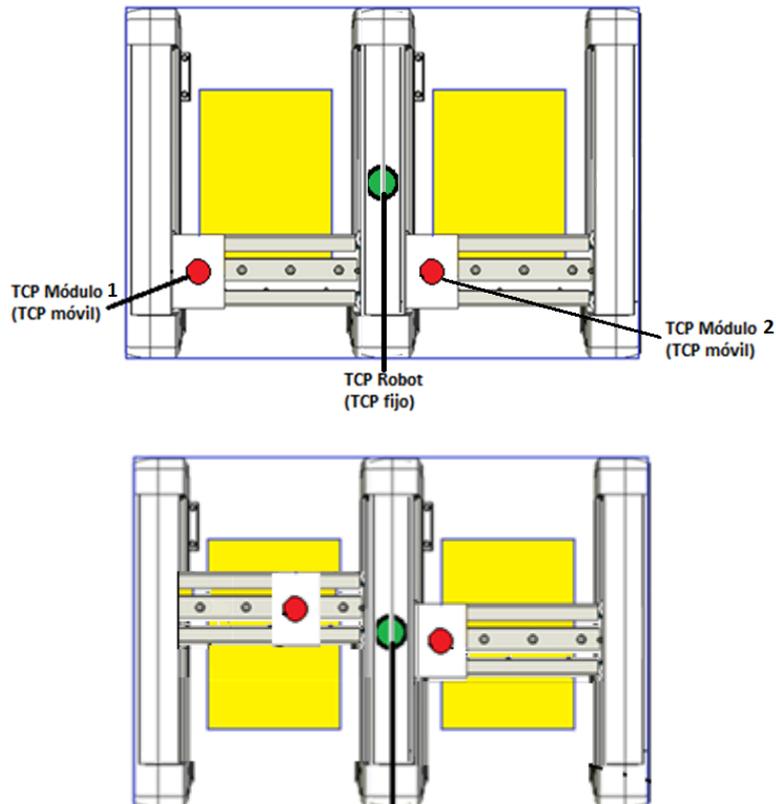


Figura 3.8. Herramienta multi-pick triple con dos módulos x-y.

Para este caso, la aplicación de visión por computador utilizaría un algoritmo que determinaría en un primer instante un objeto a coger mediante el TCP fijo del robot. A partir de ese momento se buscarían los objetos más cercanos que se encuentren dentro de las áreas efectivas y a su vez que sean más óptimos para poder realizar la operación *pick* de manera efectiva. Acto seguido se enviarían las coordenadas al robot y se ejecutará el proceso completo.

Al utilizar coordenadas polares para realizar la manipulación mediante la alternativa triple, no solo bastaría con incorporar otra guía lineal idéntica, si no que se deberían incorporar dos nuevos dispositivos.

En la figura 3.9, se observa un esbozo con una posible distribución de los módulos. En este caso se incorporan dos motores uno para cada guía lineal. La función de estos motores es la de realizar el giro de la guía lineal y posicionarla en la coordenada angular especificada.

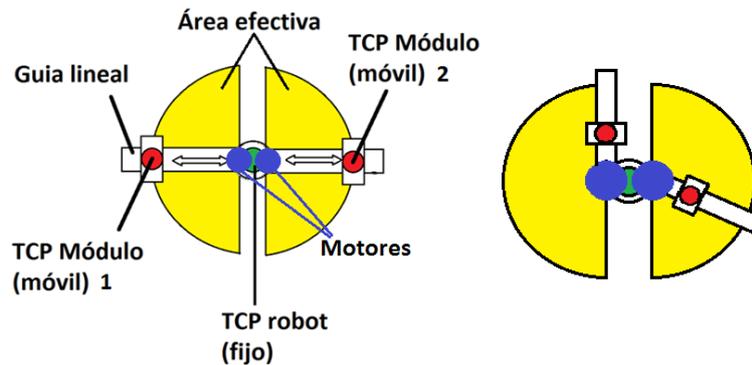


Figura 3.8. Herramienta multi-pick triple con coordenadas polares.

El hecho de incorporar una guía lineal más a la herramienta impide utilizar el eje actuador del robot para realizar el giro, ya que de esta forma se produciría el giro de las dos guías innecesariamente y, aunque se podría solucionar el problema añadiendo un offset en el algoritmo de control de posición, esto complicaría el diseño del software e incrementaría el tiempo del proceso.

Con la incorporación de los motores para el posicionamiento angular de cada guía, se consigue una herramienta triple con dispositivos de posicionamiento de poco peso, presentando una ventaja respecto a la configuración cartesiana de la herramienta. No obstante el área efectiva total podría ser menor dependiendo de los actuadores elegidos.

3.1.3 Herramienta multi-pick múltiple.

Viendo las alternativas de diseño expuestas en los puntos anteriores, es fácil intuir que a medida que vayamos añadiendo módulos a la herramienta incrementaremos el número de objetos a manipular en la línea de automatización. Así, si por ejemplo se añade otro módulo al diseño de la herramienta triple, se podrían coger 4 objetos simultáneamente (3 TCP móviles + 1 TCP fijo). Añadiendo otro módulo, se tendría un total de 4 módulos permitiendo el agarre de 5 productos simultáneamente (figura 3.10, 4 TCP móviles + 1 TCP fijo) y así, consecutivamente.

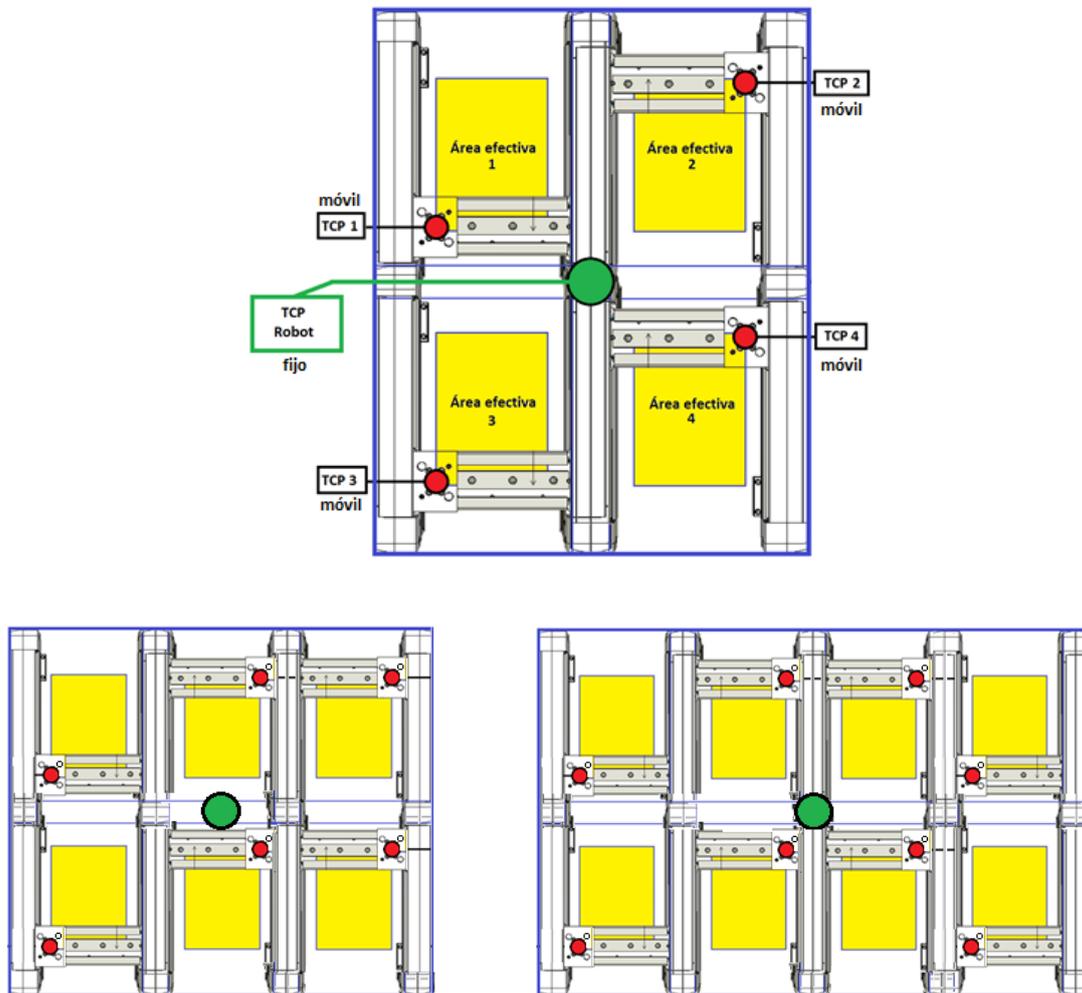


Figura 3.10. Herramienta multi-pick múltiple con módulos x-y.

Las soluciones mostradas en la figura anterior, cumplen con las condiciones establecidas en este proyecto, en el que se pretende que la herramienta multi-pick sea modular. Como se observa en la imagen, se pueden ir añadiendo módulos dependiendo del número de objetos que se pretenda manipular. Además, también se podría estudiar la posición en la que se debe situar el TCP fijo del robot, para realizar el depositado de los productos en el envase o la caja de empaquetado.

No obstante, la distribución modular de los diferentes actuadores de posicionamiento cambiaría si se desea utilizar la configuración que utiliza coordenadas polares para el posicionado.

En este caso, se pueden hacer múltiples distribuciones que permitirían a su vez abarcar diferentes áreas de trabajo.

Si por ejemplo se desean emplear 4 módulos de posicionamiento circular, una posible distribución de los mismos como la mostrada en la figura 3.11, permitiría abarcar 4 áreas efectivas que tendrían forma de semicircunferencia.

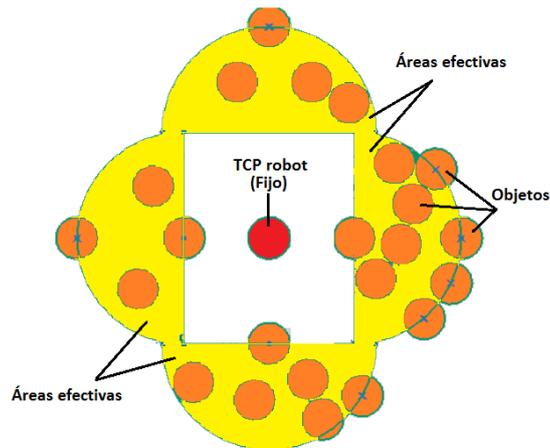


Figura 3.11. Herramienta multi-pick múltiple con coordenadas polares.

En la imagen anterior se han dibujado las 4 áreas de trabajo que se conseguiría formar distribuyendo las guías lineales en cruz. Además, se han introducido objetos para tener una visión de las posibles posiciones en las que se podrían encontrar éstos sobre la cinta transportadora.

Como se observa, para conseguir esta distribución se han tenido que separar las guías lineales del eje actuador del robot, lo que produce un área rectangular no efectiva en el centro de la herramienta. Una solución que permitiría disminuir el área no efectiva, sería la mostrada en la figura 3.12, donde se ha realizado una separación mínima entre las guías lineales y el eje actuador.

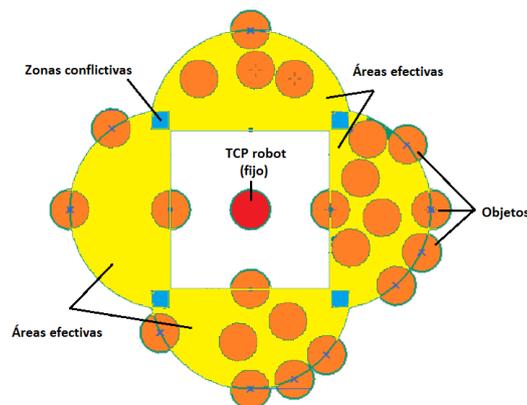


Figura 3.12. Herramienta multi-pick múltiple con coordenadas polares y sistema de detección de colisiones.

En este caso, se ha conseguido disminuir el área no efectiva que se producía en la distribución anterior. No obstante, se observa en azul una zona de área que correspondería al área efectiva de dos guías lineales solapadas entre sí. Esto produciría un conflicto en el control de posición de las guías y sería necesario emplear técnicas de detección de colisiones para evitar que las guías choquen entre sí.

Si se diseña un control de posición que disponga de técnicas de detección de colisiones, se pueden distribuir las guías lineales de manera que se minimice el área no efectiva y se maximice el área de trabajo. A continuación se muestran posibles distribuciones que permitirían minimizar el área no efectiva, pero que requerirían de técnicas de evitación de obstáculos en el control.

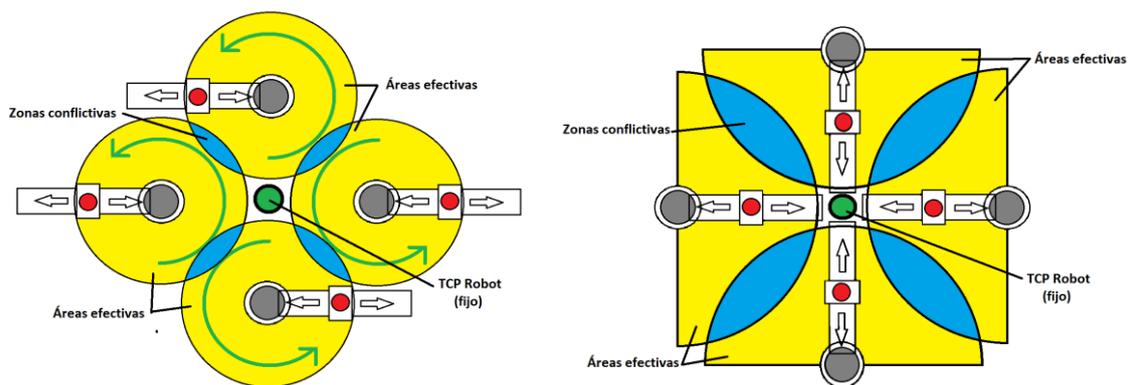


Figura 3.13. Configuraciones de multi-pick múltiple con coordenadas polares y control con evitación de obstáculos.

3.2 Situación de partida

La aplicación a realizar se desarrollará en el laboratorio de automatización para el sector de alimentación (L.A.S.A) del Instituto de Automática e Informática industrial (AI2) de la Universidad Politécnica de Valencia.



Figura 3.14. Línea de automatización en el laboratorio L.A.S.A del AI2.

En dicho laboratorio se dispone de una línea automatizada de producción que contiene los siguientes elementos:

- Cinta transportadora industrial donde se colocan los productos para realizar ensayos de prueba.
- Un Pc convencional donde están instalados los programas necesarios para llevar a cabo el procesamiento de datos de la parte de visión por computador, así como las comunicaciones con el controlador y el robot industrial.
- Robot industrial delta ABB IRB 360 Flexpicker, así como su controlador industrial IRC5.
- Cámara de visión para realizar la captura de imágenes y procesarlas en Pickmaster.



Figura 3.15. Robot delta IRB 360 de ABB, controlador y sistema de visión del laboratorio L.A.S.A del AI2.

A partir de estos dispositivos de automatización se desarrollará la herramienta multi-pick. En un primer instante y una vez determinados los demás dispositivos para el diseño de ésta, se realizará un primer proyecto básico en el que se tengan en cuenta aspectos como número de módulos, forma de las áreas de trabajo, etc. Todas las dimensiones de los diferentes elementos que dispone la herramienta deben ser parametrizables, así, de esta manera se podrán añadir los módulos que se requieran posteriormente. Es por ello que, los algoritmos de control también se deberán diseñar para que tengan en cuenta las futuras ampliaciones.

3.3 Elección de la herramienta multi-pick.

Una vez estudiadas las alternativas propuestas en el punto 3.1, se ha decidido optar por realizar un primer diseño básico utilizando la configuración por coordenadas polares que permitirá describir un área de trabajo circular.

En una primera prueba, se utilizará una guía lineal que irá conectada en su extremo al eje actuador del robot delta ABB disponible en el laboratorio. Esta conexión permitirá manipular dos objetos simultáneamente.

Se realizará el control de posición de la herramienta y se hará una primera prueba verificando los resultados esperados. Seguidamente, se incorporará otra guía para poder manipular un objeto más (3 en total), colocando los motores a cada una de las guías para realizar los respectivos giros. Una vez montada la herramienta completa se modificará el algoritmo de control incorporando los cambios necesarios para su correcto funcionamiento.

Las razones por la que se ha decidido utilizar esta configuración se detallan a continuación:

- Se utilizarán guías lineales y motores que a diferencia de los módulos x-y, presentan un menor coste de fabricación.
- A priori, el movimiento de rotación puede llegar a ser más rápido que el movimiento lineal, debido a que las guías lineales utilizan elementos mecánicos como husillos y rodamientos que limitan la velocidad y requieren de mantenimiento.

3.4 Herramientas Hardware a utilizar.

A continuación se explicarán cada una de las herramientas hardware utilizadas para el diseño de la herramienta multi-pick.

3.4.1 Robot industrial ABB IRB-360.

El robot industrial IRB-360 de ABB, es un robot tipo delta como el mostrado en la figura 3.18. Los robots delta se conforman básicamente en 3 brazos conectados a una base común mediante paralelogramos. Su principal diferencia con el resto de robots es su gran velocidad y precisión moviendo cargas pequeñas. Por lo que se utilizan mucho en el sector alimentario para tareas en sistemas de manipulación *pick and place* por su gran flexibilidad y velocidad.

El IRB 360 es extremadamente potente y presenta una aceleración de hasta 10G y una capacidad de manejo de hasta 3 kg. Gracias a las cadenas de accionamiento optimizadas y a las funciones

QuickMove patentadas por ABB, se trata del robot más rápido de su clase, con hasta 200 operaciones de selección por minuto (en función del ciclo y la carga).

Concretamente el robot que se utilizará para el desarrollo de la herramienta es el IRB 360-1 que está equipado con el controlador IRC5 y el software de control de robots RobotWare. Este software admite todos los aspectos del sistema de robot, como el control del movimiento, el desarrollo y la ejecución de programas de aplicación, la comunicación, etc.

Algunas de sus especificaciones técnicas se detallan a continuación:

- Capacidad de carga de 1Kg.
- Repetitividad de posición 0.1mm.
- Repetitividad angular entre 0.4 y 1.5°.
- La repetitividad cuando trabaja a una velocidad de 200mm/s es de 1mm.

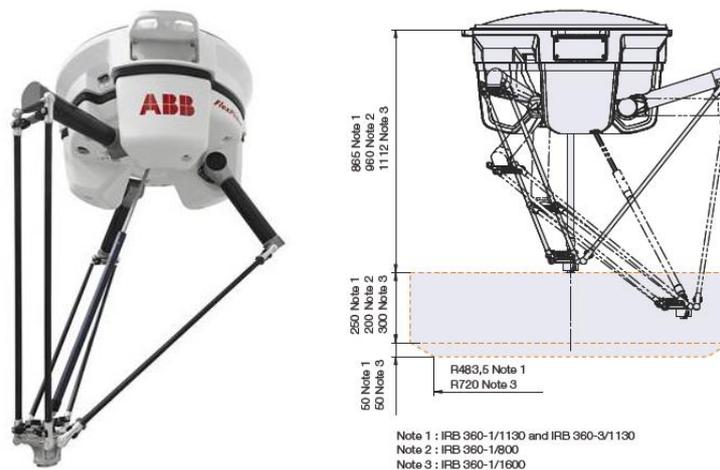


Figura 3.16. Robot delta IRB-360 de ABB y área de trabajo.

3.4.2 Cámara de visión

Para la captura de imágenes en la parte de visión por computador, se ha utilizado una cámara de escaneo de área especializada para este tipo de procesos.

Se dispone de una cámara Basler Scout sca1300-32g, que posee un sensor de la marca Sony, y cuyas características técnicas se detallan a continuación:

- Resolución: 1294x264 px.
- Ratio de imágenes: 32 frames por segundo.
- Color y blanco y negro.
- Sincronización por software o disparo externo.
- Tipo de sensor: CCD.
- Peso: 160g.

- Formato de video soportado: Mono 8, Bayer BG 8, Bayer BG 16, Bayer BG 12 Packed, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed, RGB 8 Packed.

3.4.3 Guía lineal SLN-27

Para el posicionado del TCP móvil de la herramienta se requiere un dispositivo de posicionado con las siguientes características:

- Debe ser rápido y de respuesta eficaz.
- Poco peso. Debido a que el robot utilizado es capaz de aguantar un peso de hasta 1.6Kg y además, se debe tener en cuenta que la herramienta debe ser modular (posibilidad de insertar más dispositivos iguales).
- Por la misma razón de peso y modularidad, los dispositivos deben ser de tamaño reducido.
- Bajo mantenimiento y precio económico.

Se han barajado las posibles opciones para la elección del dispositivo teniendo en cuenta entre ellos los mencionados en el apartado 2.3.1. Después de valorar las ventajas y desventajas de cada uno de ellos, se ha tomado la decisión de utilizar la guía lineal SLN-27 de la empresa alemana IGUS.

Esta guía es un actuador lineal en miniatura concebido para tareas sencillas de manipulación, que puede adquirirse con un motor paso a paso para realizar el movimiento del carro.

Los materiales de que consta y su diseño convierten a este eje lineal en un sistema deslizante económico y flexible. El rail guiado es de aluminio anodizado de larga duración y lo hace apto para aplicaciones en espacios muy reducidos. Los cojinetes deslizantes de plástico en los soportes del husillo y en el patín de guiado hacen que el sistema funcione sin lubricación y no precise de mantenimiento alguno.



Figura 3.17. Guía lineal SLN-27 con motor paso a paso.

La guía lineal contiene un motor paso a paso cuya función será mover el patín de posicionamiento. Seguidamente se detallan las especificaciones técnicas del eje lineal.

- **Eje lineal:**

- Carrera efectiva de 100mm.
- Peso 100 g.
- Máxima velocidad 300rpm.
- Par máximo 0.1Nm.
- Husillo de acero inoxidable y paso de métrica 5 x 0.8 mm.

El motor paso a paso que incluye la guía lineal es un motor bipolar de tamaño estándar NEMA 11 (*Asociación Nacional de Fabricantes Eléctricos*). Este incluye un encoder incremental que servirá para verificar la posición en la que se encuentra el motor cerrando así el lazo de control. Las características más destacadas de ambos se pueden ver a continuación:

- **Motor paso a paso:**

- Voltaje máximo 60 VDC, y nominal entre 24-48 VDC.
- Corriente nominal de 1 A.
- Par máximo 0.13 Nm.
- Ángulo de paso 1.8°.

- **Encoder:**

- Incremental.
- Voltaje nominal 5 VDC.
- Control mediante protocolo rs422.

Cabe destacar que al utilizar un encoder incremental, es necesaria la colocación de un dispositivo que sea capaz de asignar al encoder la posición de inicio (home), de tal manera que cuando se desconecte la herramienta de la alimentación, cuando ésta se vuelva a conectar, se realice un posicionado inicial para que el encoder empiece a contar las posiciones desde cero. Para ello, se utilizará un simple final de carrera que se colocará al principio de la guía lineal.

3.4.4 Driver de control DRV8825

El motor paso a paso necesita un driver de control para regular la posición deseada de éste. Al igual que el dispositivo de posicionamiento lineal, es necesario buscar un controlador que se adapte a las necesidades requeridas para el diseño de la garra:

- Que soporte la corriente y el voltaje nominal del motor paso a paso.
- Apto para motores paso a paso bipolares.
- Debe tener la posibilidad de cambiar la dirección de giro.
- Económico y de poco peso.
- Sin necesidad de incorporar mucho cableado.

- Compatible con Arduino.

Un controlador que ha cumplido todas las especificaciones ha sido el DRV8825. Este es un circuito controlador de motores paso a paso que proporciona una corriente de 1,5 amperios de forma continua y hasta 2,2 A con refrigeración adicional.

Este driver está basado en el circuito DRV8825 de Texas Instruments y se utiliza para controlar motores paso a paso bipolares de entre 8,2 y 45 V con una precisión de entre 1 y 1/32 pasos en seis velocidades diferentes. El controlador tiene un potenciómetro que permite ajustar el consumo de corriente para evitar daños en el motor y tener un movimiento más suave y continuo del mismo.

Al circuito del driver se le pasan como entradas una señal PWM y un señal de bit que indicará la dirección de giro del motor. Además, dispone de los siguientes bits de configuración:

- **MODE 0, 1, 2:** Permite diferentes configuraciones para determinar la precisión de paso del motor (resolución). Dependiendo del valor lógico de los 3 bits, se configurará una resolución desde 1 paso completo (full step) hasta 32 micro pasos por paso (1/32).
- **SLEEP:** Configuración para que el driver entre en bajo consumo mientras no esté trabajado.
- **RESET:** Reestablece las salidas y entradas del controlador.
- **ENABLE:** Activa o desactiva el driver.

El bloque controlador de salida consta de un circuito de potencia diseñado mediante transistores MOSFET de canal N configurado como puente H, que será el encargado de proporcionar la corriente a los devanados del motor.

A continuación se muestra una imagen del driver DRV8825 con un ejemplo de conexión (figura 3.18) así como el circuito electrónico del driver, con sus componentes de potencia y protección (figura 3.19).

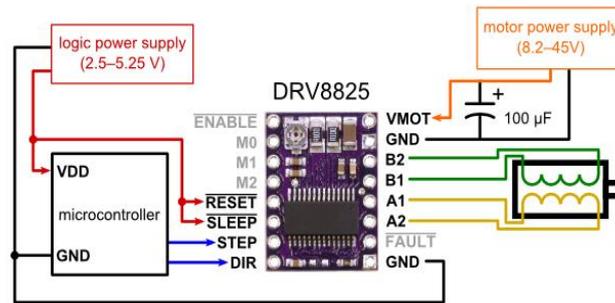


Figura 3.18. Diagrama de cableado para la conexión de un microcontrolador al driver DRV8825 (modo Full Step).

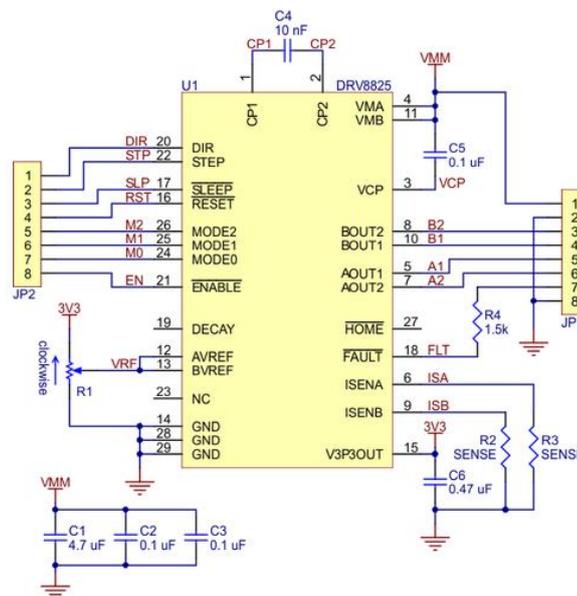


Figura 3.19. Esquemático del DRV8825 de Texas Instruments.

3.4.5 Arduino Mega 2560

Para realizar el control de posición y velocidad de la herramienta multi-pick, así como realizar la comunicación con el robot industrial y el PC, es necesario un cerebro capaz de reunir las características requeridas. Para este proyecto en el que se pretende realizar un prototipo de herramienta, se ha pensado utilizar un microcontrolador económico, de gran capacidad de procesado, con gran cantidad de entradas/salidas digitales y analógicas, etc. A la hora de escoger el microcontrolador a emplear hay que tener en cuenta multitud de factores, como la documentación y herramientas de desarrollo disponibles y su precio, la cantidad de fabricantes que lo producen y por supuesto las características del microcontrolador (tipo de memoria de programa, número de temporizadores, interrupciones, etc.)

Para la realización de este prototipo de herramienta multi-pick se requiere una placa programable basada en microcontrolador que disponga de las especificaciones mencionadas

anteriormente, y principalmente basándose en el coste de ésta, se ha decidido emplear la placa programable de la plataforma de hardware libre Arduino, concretamente, el Arduino Mega 2560.

Arduino Mega 2560 es una placa electrónica basada en el microcontrolador ATmega1280 de Atmel AVR, y un entorno de desarrollo. Está diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares.

Cuenta con 54 pines digitales de entrada / salida (de los cuales 14 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UARTs (puertos serie), y un oscilador de 16MHz. Entre las especificaciones técnicas más destacadas figuran las detalladas a continuación:

- Voltaje nominal: 5V.
- Voltaje de entrada recomendado: 7-12V.
- Voltaje de entrada límite: 6-20V.
- Corriente por pines de E/S: 40mA.
- Memoria Flash: 128 Kb de los cuales 4Kb se utilizan para la carga de inicialización.
- SRAM: 8Kb
- EEPROM: 4Kb

El Arduino Mega tiene una serie de facilidades para establecer la comunicación con un ordenador, otro Arduino, u otros microcontroladores. El ATmega1280 ofrece cuatro UART hardware para TTL (5V) de comunicación serial. Un FT232RL FTDI insertado en la placa donde uno de ellos transmite la información a través de USB y drivers FTDI (incluido con el software de Arduino). El software de Arduino incluye un monitor serie que permite enviar datos sencillos hacia y desde la placa Arduino. Además el ATmega1280 también soporta I2C (TWI) y la comunicación SPI.



Figura 3.20. Arduino Mega 2560.

La placa se puede programar mediante el software de desarrollo que implementa el lenguaje de programación propio basado en el lenguaje de programación de alto nivel Processing. Los ATmega1280 vienen precargados con un gestor de arranque que le permite cargar el código sin

el uso de un programador de hardware externo. Se comunica utilizando el protocolo original STK500 (referencia, archivos de cabecera C).

Una de las ventajas del uso de Arduino es la posibilidad de utilizar otros lenguajes de programación y aplicaciones populares. Esto es debido a que Arduino usa una transmisión serie de datos que es soportada por la mayoría de los lenguajes mencionados. Para los que no soportan el formato serie de forma nativa, es posible utilizar software intermediario que traduzca los mensajes enviados por ambas partes para permitir una comunicación fluida.

Algunos ejemplos de lenguaje de programación pueden ser: *Java*, *.NET*, *C*, *C++* (mediante *libSerial* o *Windows*), *C#*, *Matlab*, *Python*, o *Flash* (*ActionScript*).

3.4.6 Motor Brushless.

En el caso de utilizar dos guías lineales, es necesario la incorporación de dos motores que se encarguen de realizar el posicionado angular de las guías.

Al igual que para el posicionado lineal, por razones de que la herramienta debe ser modular, (apartado 3.4.3) el dispositivo a emplear debe reunir las características de rápida respuesta, poco peso, tamaño reducido, bajo mantenimiento y precio económico.

Una posible alternativa es la utilización de un servomotor. El servomotor es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición. Puede ser controlado tanto en velocidad como en posición y los hay de numerosos tamaños y pesos. No obstante, están compuestos a parte del motor de corriente continua, por una circuitería de control y un conjunto de engranajes mecánicos que actúan como reductores, todo ello encapsulado en el mismo motor. Por estos motivos el sistema completo se haría más complejo y aparatoso, lo que ha dado lugar a prescindir de estos motores para el diseño de este prototipo.

Otra alternativa son los motores brushless (apartado 2.3.2.1). Existen motores brushless que son construidos especialmente para ser colocados en soportes de cámaras de fotos y vídeo, y realizan el posicionado angular de las mismas. Estos motores también llamados “Gimbal” cumplen los requisitos mencionados y además, su forma, tamaño, peso y simplicidad lo han hecho ideales para este proyecto.

Por todos estos motivos, se ha decidido emplear el modelo mostrado en la figura 3.21.



Figura 3.21. Motor Brushless Gimbal.

El funcionamiento de este tipo de motores es similar al del motor síncrono trifásico, ya que es accionado mediante la aplicación de una tensión sinusoidal de 3 fases desfasadas 120° . Entonces, al motor se le aplicaran tres voltajes con amplitudes $(U_a, U_b, U_c) = U_0 [\cos \theta, \cos (\theta - 2 \cdot \pi / 3), \cos (\theta - 4 \cdot \pi / 3)]$, donde θ determina la posición donde está el motor, y al mismo tiempo la variable de control del mismo.

Como se ha explicado, el motor está diseñado para el control angular de las cámaras con la posibilidad de realizar giros de 360° . En concreto el motor elegido para la herramienta (GMB5208H-180T) puede soportar un peso de hasta 1500 gramos. Sus características más destacadas son:

- Resistencia: 17.2ohm.
- Peso: 180 g.
- Tamaño: 35x28x12 cm.

Para el control de posición del motor se utilizará un potenciómetro que conectado a una entrada analógica del microcontrolador proporcionará la información de posicionado en la que se encuentra el motor cerrando así el bucle de control.

3.4.7 Driver de control del motor Brushless.

El motor brushless necesita un driver que amplifique las señales PWM que necesita y que controle las corrientes que circulan por ella mediante un puente en H para cada una de ellas. Además, debe cumplir con las necesidades requeridas como son:

- Que soporte la corriente y el voltaje nominal del motor.
- Debe tener la posibilidad de cambiar la dirección de giro.
- Económico y de poco peso.
- Sin necesidad de incorporar mucho cableado.
- Compatible con Arduino.

Después de analizar los controladores existentes en el mercado, se ha decidido optar por utilizar el driver trifásico L634D (figura 3.22).

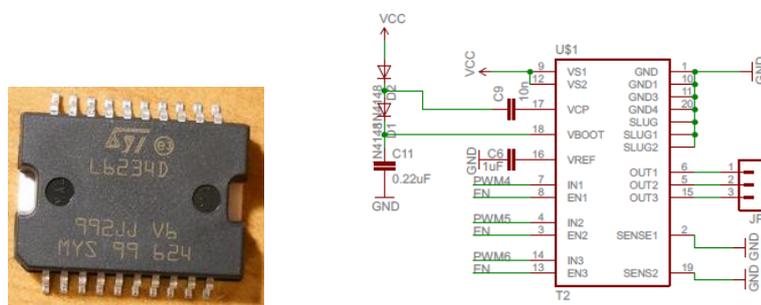


Figura 3.21. Driver L634D para el motor Brushless.

Este driver es un triple puente en H específico para motores brushles. Los transistores DCMOS de los puentes en H son capaces de proporcionar una elevada corriente, por lo que son muy utilizados ya que pueden controlar una gran variedad de motores. Funciona mediante un voltaje de alimentación de entre 7 y 58V y es capaz de soportar hasta 5 amperios de corriente de pico.

Los pines utilizados para esta aplicación han sido los siguientes:

- ENABLE: Para activar y desactivar el driver.
- PWM1, PWM2, PW3: Donde se conectarán las tres señales PWM provenientes del microcontrolador Arduino.
- OUT1, OUT2, OUT3: Serán las señales de salida del driver que se conectarán al motor brushless.

3.4.8 Fuente de alimentación.

Para alimentar los motores y proporcionar así la corriente deseada se utilizará una fuente de alimentación de laboratorio de corriente continua regulable, con limitación de corriente y protección contra cortocircuitos, cuyas especificaciones técnicas se muestran a continuación:

- Modelo: df1731sb.
- Salida dual.
- Voltaje: 0 - 60V.
- Corriente: 0 – 6A.

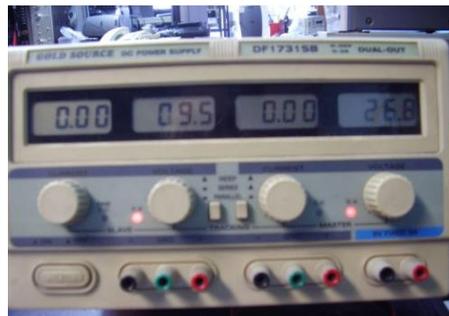


Figura 3.23. Fuente de alimentación de laboratorio.

3.5 Herramientas Software a utilizar

3.5.1 Software de Arduino.

Para el desarrollo de la aplicación de control se utilizará, como se ha mencionado anteriormente, una placa electrónica de Arduino. Esta dispone de un software de programación cuya versión descargada con el fin de programar la aplicación ha sido la 1.0.5-r2.

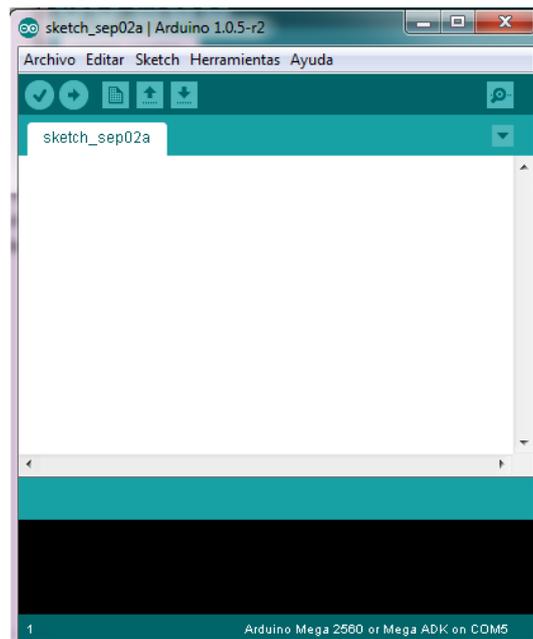


Figura 3.24. Sketch de Arduino.

El entorno de desarrollo contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para las funciones comunes, y una serie de menús. Se conecta con el hardware Arduino para cargar los programas y comunicarse con ellos.

El código a realizar se programa en editor de textos llamado *Sketch*, y se guarda en formato *.pde*.

Las funciones básicas y los operadores que usa esta plataforma están basados en el lenguaje C, y C++. Algunas de las estructuras y sintaxis de programación se muestran a continuación:

- **Sintaxis Básica:** Delimitadores: `;`, `{}`. Comentarios: `//`, `/* */`. Cabeceras: `#define`, `#include`. Operadores aritméticos: `+`, `-`, `*`, `/`, `%`. Operadores de comparación: `==`, `!=`, `<`, `>`, `<=`, `>=`. Operadores de bits: `&`, `|`, `^`, `~`, `<<`, `>>`. Etc.
- **Estructuras de control:** Condicionales: `if`, `else`, `switch case`. Bucles: `for`, `while`, `do... while`. Etc.

- **Constantes:** HIGH/LOW: que representan los niveles alto y bajo de las señales de entrada y salida. INPUT/OUTPUT: entrada o salida. False (falso), true (verdadero).
- **Tipo de datos:** *void, boolean, char, unsigned char, byte, int, unsigned int, word, long, unsigned long, float, double, string, array.*
- **Conversión entre tipos:** *char(), byte(), int(), word(), long(), float().*
- **Calificadores y ámbito de las variables:** *static, volatile, const.*
- **E/S Digital:** *pinMode(pin, modo), digitalWrite(pin, valor), int digitalRead(pin).*
- **E/S Analógica:** *analogReference(tipo), int analogRead(pin), analogWrite(pin, valor).*
- **Tiempo:** *unsigned long millis(), unsigned long micros(), delay(ms), delayMicroseconds(microsegundos).*
- **Matemáticas y trigonométricas.**
- **Interrupciones externas:** *attachInterrupt(interruptión, función, modo), detachInterrupt(interruptión).*

3.5.2 Software de visión (Pickmaster).

PickMaster es un software especialmente diseñado para aplicaciones *pick and place* de alta velocidad. El software proporciona integración completa del robot IRB360 con el controlador IRC5, el sistema de visión y las cintas transportadoras en la celda de recogida. Dispone de asistentes de configuración de los componentes hardware que se puedan conectar, tales como una cámara de visión. Además, como el programa permite una programación de tipo GUI fácil y rápida para nuevos tipos de productos, con seguridad será considerado como un sueño para los operarios. En realidad, una vez que se está conectado y en marcha, se dispone de un control total de la aplicación y la información se presenta en una sola pantalla.

La aplicación PickMaster se implementa mediante tres módulos principales. El Administrador del sistema es la aplicación principal, donde se definen las líneas de producción y los proyectos. Para la detección de la visión se utiliza el Servidor Vision y por último está el Servidor de movimiento, que se encarga de la comunicación con el controlador del robot.

Estas relaciones se muestran en la siguiente figura:

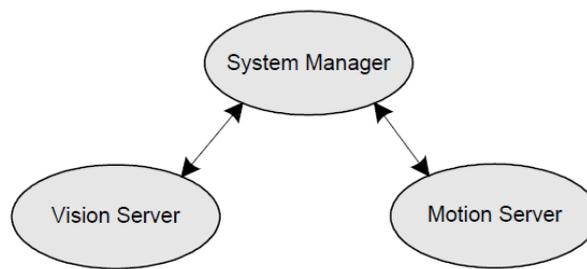


Figura 3.25. Módulos principales de Pickmaster.

El servidor Visión se implementa con Cognex Vision Library y por lo tanto, sólo es compatible con las cámaras de Cognex. Los objetos de la cámara se configuran y calibran en una línea PickMaster y no se suelen cambiar muy a menudo. En los proyectos de PickMaster, se definen modelos de visión para encontrar patrones particulares.

Cuando se ejecuta un proyecto de la cámara se activa desde el controlador de robot. Las imágenes se analizan en el servidor encontrado las posiciones y orientaciones de los objetos, que serán reportados al administrador del sistema.

Finalmente, se utiliza una señal estroboscópica generada por el tablero visión cuando se conecta una imagen con el controlador del robot, para sincronizar las posiciones con las cintas transportadoras.

3.5.2.1 Algoritmo de control de visión por computador para multi-pick robotizado.

Para situaciones en las que no se dispone de los modelos de visión, o no son suficientes para resolver necesidades específicas, se pueden implementar modelos especializados de visión con el SDK de PickMaster. Es el caso del algoritmo de control de visión por computador que ha realizado como proyecto final de carrera el alumno de Ingeniería informática de la Universidad Politécnica de Valencia: Sergio Navarro, junto con su director Martín Mellado.

El proyecto consiste en el desarrollo de la aplicación de visión por computador para herramientas multi-pick robotizadas. Esta aplicación reconoce los objetos que se desplazan por una cinta mediante visión por computador. Dependiendo del número de objetos a coger, que se habrán determinado previamente, se obtiene su posición y orientación de cada uno en base a un sistema de coordenadas.

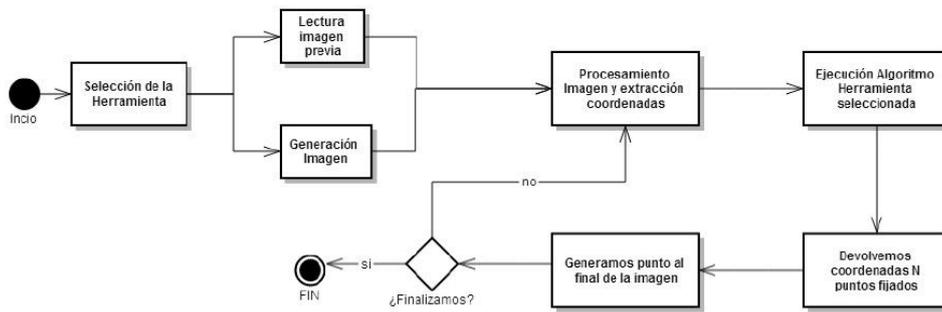


Figura 3.26. Diagrama de flujo del algoritmo de control de visión por computador para la herramienta multi-pick

En la siguiente figura se muestra una imagen en el que el algoritmo ha determinado (se observa en color rojo) las coordenadas correspondientes a un objeto al que se le llamará objeto líder, y corresponderá al objeto que deberá agarrar el TCP fijo del robot. A partir de ahí se buscarán los objetos cercanos al TCP fijo que no interfieran en su área de trabajo y, a su vez, que se encuentren dentro del área de trabajo del TCP móvil.

Una vez se determina los objetos óptimos a recoger, se le envían los datos a Pickmaster para realizar el procesado de datos y la comunicación con el robot.

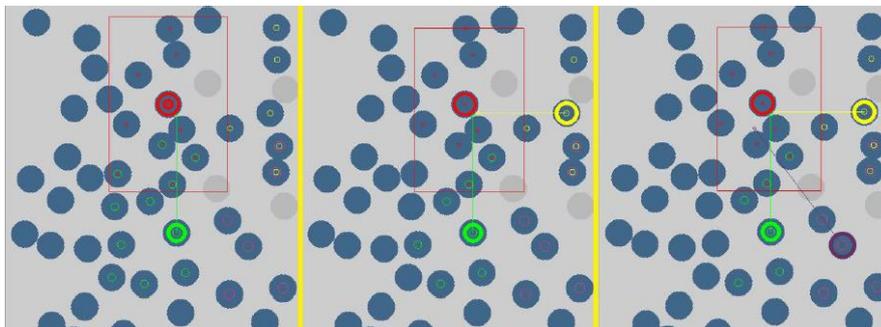


Figura 3.27. Ejemplo visual del algoritmo de control de elección de objetos por visión artificial.

Para comunicar las funcionalidades de esta aplicación con PickMaster, en dicho proyecto se ha creado una librería dinámica DLL en C++, con posibilidad de usarse en otros lenguajes de programación. Esto es debido a que para comunicarse con PickMaster, hace falta crear un proyecto en el lenguaje de programación C# e incluir en él todas las funcionalidades creadas.

3.5.3 Otros

Todo el software descrito anteriormente, estará funcionando sobre el sistema operativo Windows 7 professional. Además, se han utilizado los programas CAD *Siemens NX* y *SolidWorks* para realizar planos y diseño de piezas.

4. DESARROLLO E IMPLEMENTACIÓN DE LA HERRAMIENTA

4.1 Herramienta multi-pick doble (TCP fijo + TCP móvil).

Tal y como se ha comentado en el apartado 3.3, se ha decidido desarrollar la herramienta multi-pick circular manejada por coordenadas polares (r - α) empezando por el diseño de la pinza doble.

Para ello, se parte de los siguientes elementos eléctricos, electrónicos y mecánicos, explicados cada uno de ellos en el punto 3.4:

- Guía lineal SLN-27 de 100mm de carrera.
- Motor paso a paso Nema 11 de 45V y 1 A.
- Encoder incremental.
- Final de carrera.

Además, los compañeros del instituto de automática e informática industrial (AI2), mediante el programa CAD *SolidWorks*, han realizado el diseño de las piezas mecánicas para el acople de la guía y el motor al robot, quedando la herramienta multi-pick como se muestra en las figura 4.1:

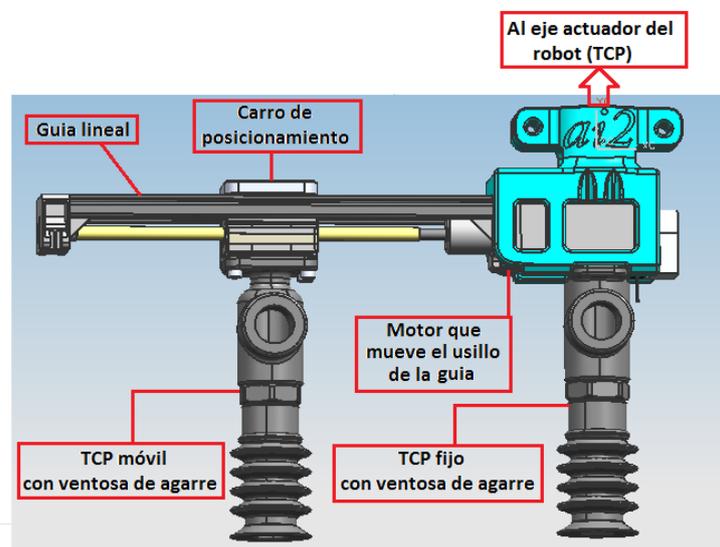


Figura 4.1. Herramienta Multi-pick circular con una guía lineal.

Como se puede observar también en la figura anterior, para el agarre de los productos se utilizarán ventosas de succión por vacío que irán conectadas a un eyector. El eyector será el encargado de realizar el vacío para succionar los productos y realizar así las operaciones de *pick* y *place*, y será activado o desactivado por una electroválvula. A su vez, la electroválvula, estará

conectada al controlador IR5 del robot industrial a una señal digital de 24V y determinará la conexión y desconexión de la electroválvula.

Las ventosas de succión son dispositivos muy utilizados en este tipo de procesos. Para esta herramienta, se utilizarán ventosas circulares con un fuelle de 2.5cm que permitirán el agarre de objetos redondos de un diámetro mínimo de 3cm.

Se aprecia como una de las ventosas está conectada al TCP fijo del robot, y la otra al TCP móvil del patín de la guía lineal.

Tal y como está montada la herramienta multi-pick en esta configuración, se ha realizado un plano mediante el programa de diseño *Siemens NX*, donde se muestran las diferentes áreas de trabajo que es posible abarcar para realizar la manipulación de objetos.

En la figura 4.2, se muestra una posible configuración donde es posible manipular objetos dentro de un área de trabajo de 393cm². De esta manera se podrán manipular objetos redondos desde 3 hasta 6,5 cm de diámetro:

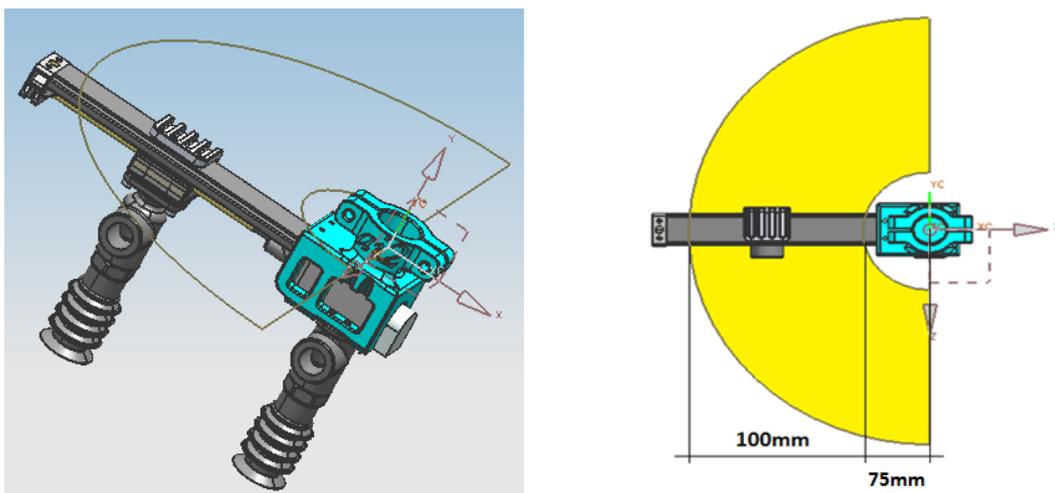


Figura 4.2. Área de trabajo en semicírculo de la herramienta multi-pick doble.

$$\text{Área efectiva} = \frac{\pi \cdot (R^2 - r^2)}{2} = \frac{\pi \cdot (175^2 - 75^2)}{2} = 39269\text{mm}^2 \cong 393\text{cm}^2$$

No obstante, también es posible realizar la configuración de la figura 4.3, donde se aprovecha el giro completo de 360° del eje actuador del robot para ampliar el área de trabajo de la herramienta. Esta configuración tiene la ventaja de poder manipular objetos dentro de un área mayor, pero tiene la desventaja de que debido a esto, puede tardarse más tiempo en realizarse la recogida del producto, dependiendo donde esté situada la guía móvil.

Cabe adelantar que para los ensayos prácticos se realizarán las pruebas utilizando la configuración en semicírculo, esto es, mediante la configuración de la figura 4.2.

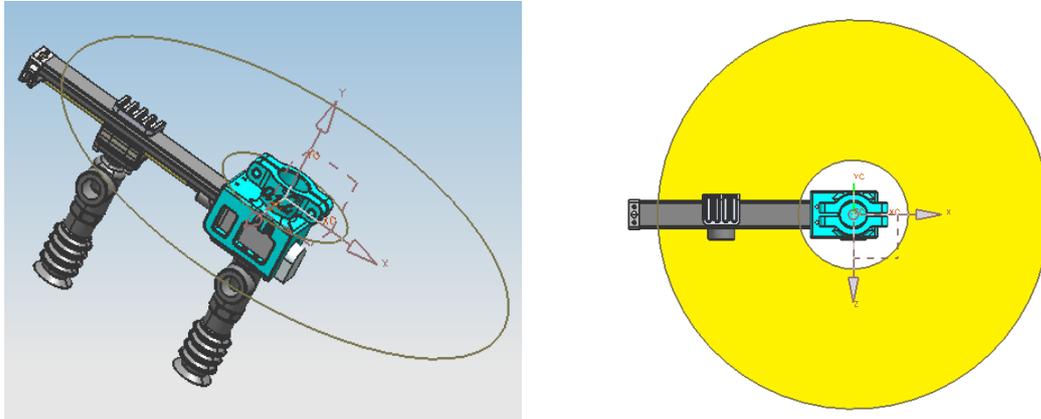


Figura 4.3. Área de trabajo en círculo completo de la herramienta multi-pick doble.

$$\text{Área efectiva} = \pi \cdot (R^2 - r^2) = \pi \cdot (175^2 - 75^2) = 78539\text{mm}^2 \cong 785\text{cm}^2$$

4.1.1 Inicio de operación.

El primer paso que deba realizar la herramienta multi-pick será la de alcanzar una posición de inicio. Esta posición determinará que la herramienta ya está preparada para trabajar y por lo tanto puede iniciarse el proceso de manipulación. Además, servirá también para indicarle al encoder incorporado en el motor paso a paso que se ha llegado a la posición 0, ya que es un encoder incremental y necesita obtener la información de inicio de cuenta.

El programa arrancará moviendo el carro de la guía lineal hacia la posición marcada como “posición de inicio” donde se encuentra colocado un final de carrera. Este final de carrera está conectado a una entrada digital del microcontrolador, de tal manera que cuando se active la señal digital el carro se detendrá, y el encoder se reiniciará a la posición 0. Es entonces cuando la herramienta puede empezar a trabajar.

El proceso de manipulación de objetos, y por lo tanto, los pasos que debe realizar la aplicación de control se pueden dividir en 3 grandes bloques donde más adelante se explicará en detalle el funcionamiento de cada uno de ellos. El primer bloque consiste en la obtención de las coordenadas del TCP móvil, que procederán a través del puerto serie del PC donde se encuentra instalado el programa de Pickmaster y la aplicación de visión por computador. Una vez se obtengan las coordenadas, se procederá a posicionar la herramienta en la ubicación donde se encuentre el objeto a coger con el TCP móvil y seguidamente se realizará la operación *pick*. Una vez ejecutada la operación de recogida, la herramienta pasará a hacer la operación *place* para

depositar el objeto en su destino. Cuando haya terminado esta operación, se repetirá de forma cíclica todo lo explicado hasta finalizar el proceso completo de manipulación.

Un sencillo ejemplo de la aplicación de control se puede ver en la figura 4.4 donde se ha realizado un flujograma principal para entender el procedimiento de la misma.

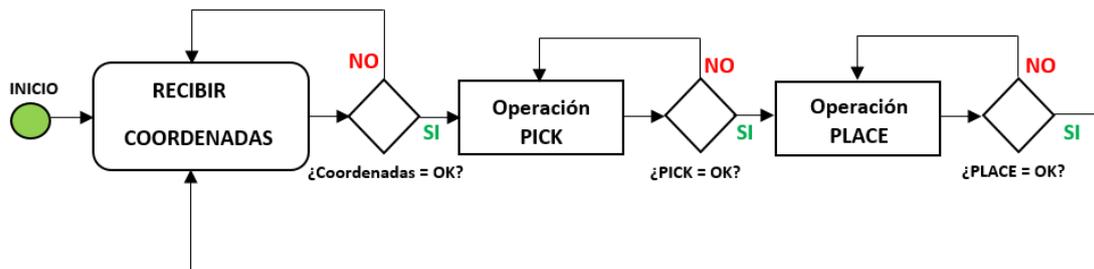


Figura 4.4. Flujograma principal de la aplicación de control del proceso de manipulación.

4.1.2 Recepción de coordenadas

Una vez que el TCP móvil del robot esté en la posición de inicio a la que se ha llamado “home position”, la aplicación de control entrará en modo de espera, en el que el sistema estará detenido hasta que le llegue un dato por el puerto serie.

En Pickmaster se habrán determinado dos objetos a tomar, uno para el TCP fijo y otro para el TCP móvil.

Este programa tiene la ventaja de indicarle al eje actuador del robot la orientación del objeto, de tal manera que se realice un giro antes de la operación *pick* para hacer coincidir la pinza manipuladora con dicha orientación. Como el objetivo de este proyecto es manipular objetos redondos, se podría prescindir de la orientación del mismo. Pero si se hace coincidir la orientación del objeto fijo con la coordenada angular (α) del objeto a coger con el TCP móvil, se podrá hacer girar la guía lineal mediante el eje actuador del robot.

Es por tanto que, para esta configuración de herramienta, la coordenada angular (α) la procesará Pickmaster con el controlador del robot, mientras que la coordenada lineal que determinará la distancia a recorrer por el carro de la guía será la recibida por el controlador Arduino y procesada por el mismo.

Tenemos pues dos coordenadas:

- **Coordenada angular (α):** La procesa Pickmaster junto con el controlador IRC5 para ser ejecutada por el robot delta.
- **Coordenada lineal (r):** Se le envía a través del puerto serie del PC al microcontrolador Arduino para que sea procesada.

El envío de datos al microcontrolador Arduino se realizará a través del puerto serie USB del ordenador donde se encuentra instalado Pickmaster. El algoritmo de visión habrá determinado la coordenada y la habrá convertido a un valor binario, por lo que se dispondrá de 1 byte en el que esté codificada dicha coordenada.

Una vez se recibe el dato (byte) por el puerto serie, se convertirá a decimal y se deberá realizar un escalado para ser interpretado correctamente por el encoder. El encoder tiene una precisión de 400 pasos/mm, y por lo tanto, la distancia recibida deberá ser multiplicada por la precisión. Un ejemplo de escalado podría ser el mostrado a continuación, donde se muestra un escalado a la resolución del encoder, cuando se recibe una coordenada con valor de 25mm de distancia a recorrer:

Distancia = 25mm, Factor de escala = 400/mm:

$$\text{Recorrido} = 25 \cdot 400 = 10000$$

Para este caso, la posición que debe alcanzar el TCP móvil será la posición 10.000 establecida por el encoder.

Antes de ejecutar los movimientos para posteriormente realizar la operación pick, se debe conocer la dirección (avance o retroceso) que debe tomar el TCP móvil. La distancia a recorrer se debe comparar con la posición actual en la que se encuentre el TCP móvil y entonces determinar la dirección que debe tomar éste. Esta es otra de las tareas que se ejecutará una vez recibida la coordenada, donde de manera detallada se ha realizado un flujograma en el que se observan los pasos a realizar en esta etapa (figura 4.5).

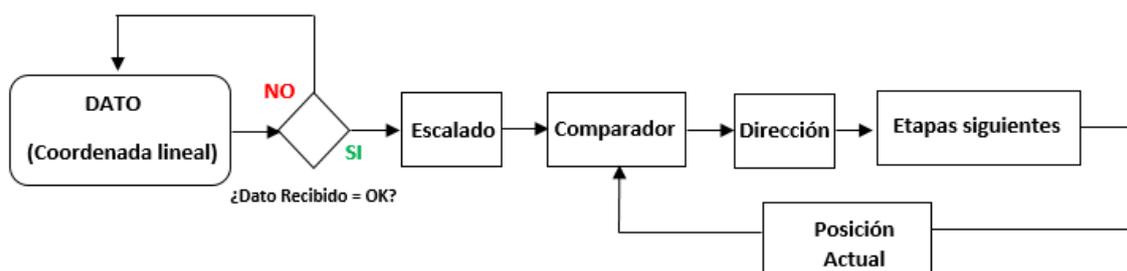


Figura 4.5. Flujograma de la etapa de recepción de coordenadas.

4.1.3 Operación Pick

Como se ha repetido en numerosas ocasiones en los apartados anteriores, la operación *pick* consiste en el agarre o recogida del producto. Para ello, antes de ejecutar esta operación, los

dispositivos de agarre (TCP fijo y móvil) deberán estar posicionados en las coordenadas pertenecientes a los objetos establecidos. Esta será la función que deberá realizar esta etapa.

Hasta ahora, se tiene registrado el recorrido que debe realizar el TCP móvil para posicionarse en el objetivo, así como la dirección de avance o retroceso.

El encargado de mover el carro será el motor paso a paso, y el encoder determinará cuándo se ha alcanzado la posición objetivo ya que irá contando cada uno de los pasos del motor.

Se tiene pues, una referencia (posición objetivo), un actuador (motor paso a paso), y un sensor (encoder) que cerrará el lazo de control indicando en cada instante la posición en la que se encuentra el TCP móvil.

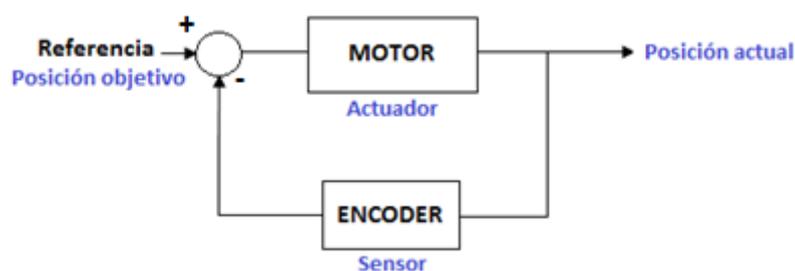


Figura 4.6. Control de posición del TCP móvil.

4.1.3.1 Determinación de la posición mediante el encoder.

El sensor encargado de medir la posición en cada instante será el encoder incremental del que dispone el motor paso a paso.

El encoder es un codificador que utiliza dos sensores ópticos posicionados con un desplazamiento de un cuarto de ranura el uno del otro. Mediante la señal de alimentación de 5V, genera dos señales de pulsos digitales que están desfasadas en 90° o en cuadratura. A estas señales de salida, se les llama comúnmente A y B. Mediante ellas es posible suministrar los datos de posición, velocidad y dirección de rotación del eje.

Para saber la dirección en la que está rotando el motor (figura 4.7), se utiliza una comparación de las señales A y B en la que generalmente, si la señal A adelanta a la señal B el eje estará rotando en sentido horario, mientras que si por el contrario la señal B adelanta a la señal A, el sentido será anti horario.



Figura 4.7. Señales de pulso generadas por encoder incremental.

Con esta información, es posible asociar cada detección de giro a un contador, que se incrementará o decrementará según si la dirección es en sentido horario o anti horario, respectivamente.

Mediante el uso de interrupciones del microcontrolador, es posible realizar la cuenta y por tanto conocer la posición en la que se encuentre el motor y a su vez el TCP móvil.

Es por tanto que, para medir la posición en la que se encuentre el motor y por lo tanto el TCP móvil, se debe realizar un código en el software del microcontrolador donde el procedimiento a seguir será el siguiente:

1. Se utilizará una interrupción externa para cada señal del encoder (A y B). En un primer instante estas interrupciones se activaran de forma paralela cuando se produzca un flanco de subida (esto es A = "1 lógico" o B = "1 lógico").
2. Cuando se active una interrupción se compararán las dos señales para determinar la dirección de giro y se incrementará o decrementará un contador dependiendo de la situación de ésta. Seguidamente se cambiará la activación de la interrupción de manera que se active cuando se produzca un flanco de bajada (esto es A = "0 lógico" o B = "0 lógico").
3. Se realizará el mismo procedimiento que el paso 2, con la diferencia de que una vez determinada la dirección, se volverá a cambiar la activación de la interrupción a flanco de subida.

Una vez realizado el primer paso, los pasos 2 y 3 se ejecutarán de forma cíclica y a la velocidad de rotación del motor. De manera visual, se puede observar el procedimiento en la siguiente figura:

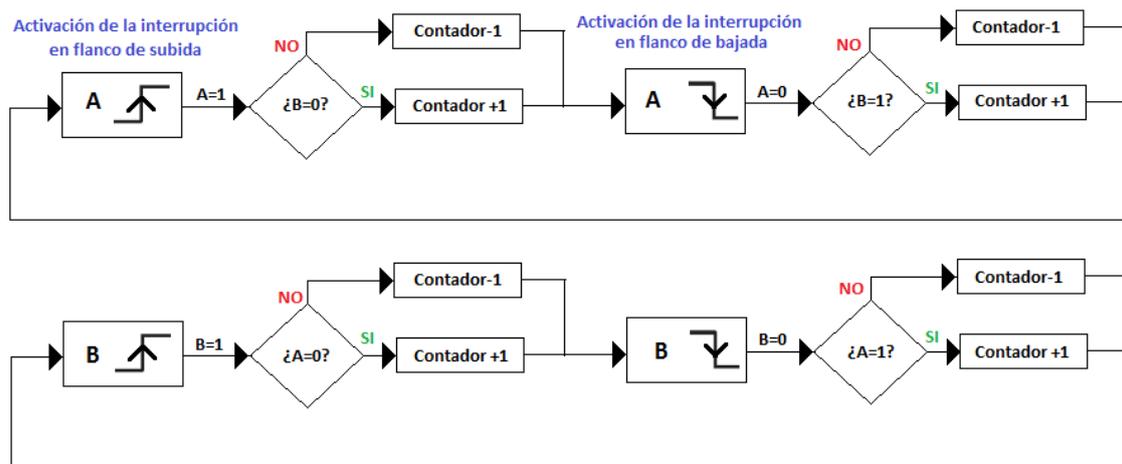


Figura 4.8. Flujo de manejo de interrupciones para determinar la posición del motor.

4.1.3.2 Control del motor PaP con rampa de velocidad lineal.

Ahora que se conoce la posición en la que se encuentra el motor en cada instante, se debe pasar a la parte de control.

Se dispone del driver DRV8825 al cual le debemos pasar a través del microcontrolador la dirección y una señal PWM que será la encargada de indicarle al controlador la velocidad y precisión con la que debe girar el motor.

Arduino contiene un módulo y una librería en la que a través del uso de registros del microcontrolador (timers, contadores, etc) es capaz de generar una señal por modulación de ancho de pulso PWM a la que se le puede modificar la frecuencia de trabajo y el ancho de pulso.

Como las operaciones de manipulación *pick and place* requieren de rapidez en su ejecución, es conveniente que el motor trabaje a la máxima velocidad permitida. Es por ello que la frecuencia de trabajo de la señal PWM debe ser la máxima que permita el motor paso a paso, siendo esta de 3KHz.

El hecho de introducirle a un motor paso a paso la máxima frecuencia en el arranque, presenta un gran inconveniente. El motor no se puede arrancar a la velocidad máxima de trabajo debido a que éste no tendrá el par suficiente para arrastrar la carga con las inercias que supone arrancar a alta velocidad. Además, es probable que el rotor y el estator del motor no se puedan sincronizar.

La solución al problema pasa por arrancar el motor a una razón de pasos baja y posteriormente acelerarlo paulatinamente hasta alcanzar la velocidad máxima. Para ello, se realizará una rampa de velocidad lineal que conseguirá una aceleración constante.

La rampa (ver figura 4.9) consiste en controlar el tiempo de retardo entre los impulsos del motor e ir incrementando poco a poco la frecuencia de trabajo de la señal PWM. Este tiempo de retardo debe de ser calculado con el fin de conseguir que la velocidad siga la rampa de velocidad lo más exactamente posible. Los pasos discretos controlan el movimiento del motor paso a paso, y la resolución de retardo de tiempo entre pasos está dado por la frecuencia del temporizador.

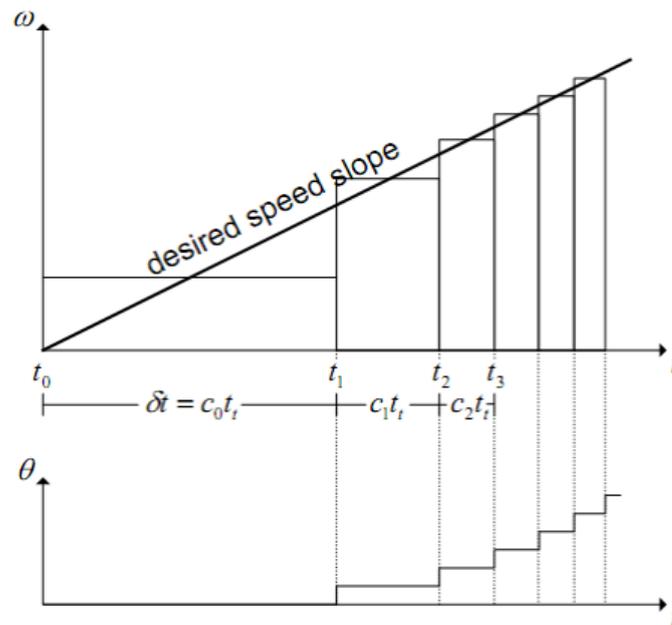


Figura 4.9. Rampa de velocidad lineal, Pulsos/velocidad paso a paso.

En la figura anterior δt es el retardo del contador empleado para realizar la señal PWM, C_0 es el primer retardo ejecutado por el contador, y C_n son los sucesivos retardos.

Después de realizar varios ensayos experimentales con el motor y la guía lineal conectados al microcontrolador y al driver de control, se ha encontrado una rampa de aceleración con la que se produce un arranque del motor óptimo. Se empezará con una frecuencia de 700 Hz, y se incrementará 350 Hz por cada iteración hasta alcanzar la frecuencia máxima de trabajo. Se tendrán pues 7 iteraciones con los valores siguientes:

- Frecuencia de arranque = 700 Hz
- 1ª iteración = 1050 Hz
- 2ª iteración = 1400 Hz
- 3ª iteración = 1750 Hz
- 4ª iteración = 2100 Hz
- 5ª iteración = 2450 Hz
- 6ª iteración = 2800 Hz
- 7ª iteración = 3000 Hz

Una vez determinados los tiempos para generar la rampa de arranque del motor, ya se puede realizar el control de posición de éste.

La primera etapa (recepción de coordenadas) nos proporciona la distancia que debe recorrer el TCP móvil así como la dirección de giro que debe tomar. Una vez se tienen estos dos datos, se seguirán los siguientes pasos:

1. Activar el driver de control del motor.
2. Asignarle al driver la dirección de giro.

3. Activar el PWM ejecutando la rampa de arranque del motor.
4. Comprobar continuamente la posición hasta alcanzar el objetivo.
5. Paro del motor. Se desactiva el driver y a su vez la señal PWM.

Si se cambia la carga y por tanto el par a generar, puede ser que se deban cambiar las condiciones de retardo y pulsos diseñados para realizar la rampa, ya que el cambio de estas características puede influir en el arranque del motor.

Si la rampa no se adapta a las adversidades, cabe la posibilidad que el motor se quede enclavado y no funcione. Por esto, es conveniente realizar también un control que tenga en cuenta el fallo de la rampa de aceleración, de tal manera que si se produce un error, se vuelva a realizar de nuevo la rampa hasta que el motor pueda arrancar y llegar a su velocidad constante.

Para ello, comprobar en el paso 4 que la posición varía en cada instante mientras no se alcance la posición deseada. Si no se alcanza el objetivo y además la posición no ha variado, esto significará que el motor está parado. Es entonces cuando se debe realizar de nuevo la rampa de arranque. Si el fallo persiste, debe saltar una alarma que indique que se ha producido una avería.

Otro de los factores a tener en cuenta es el cambio de dirección. Puede ocurrir que por circunstancias del tiempo de cómputo del microcontrolador, el motor se pase de la posición de objetivo. Si por ejemplo se debe ir a la posición 1000, y la interrupción del encoder ha saltado en la posición 999 y luego en la 1001, se debería cambiar la dirección de giro del motor para regresar a la posición de destino, que es la 1000.

Este posible fallo, se ha corregido mediante el software de programación, de manera que se introducido una variable que determina si se ha producido un error de medida, y si es así, se corrija el fallo cambiando la dirección de giro.

En la figura 4.10 se muestra el flujograma de la programación de control, en el que se pueden observar las diferentes tareas y condiciones que se deben realizar y tener en cuenta para esta etapa. Una vez se ejecuten todos los pasos, se procederá a ejecutar la etapa 3.

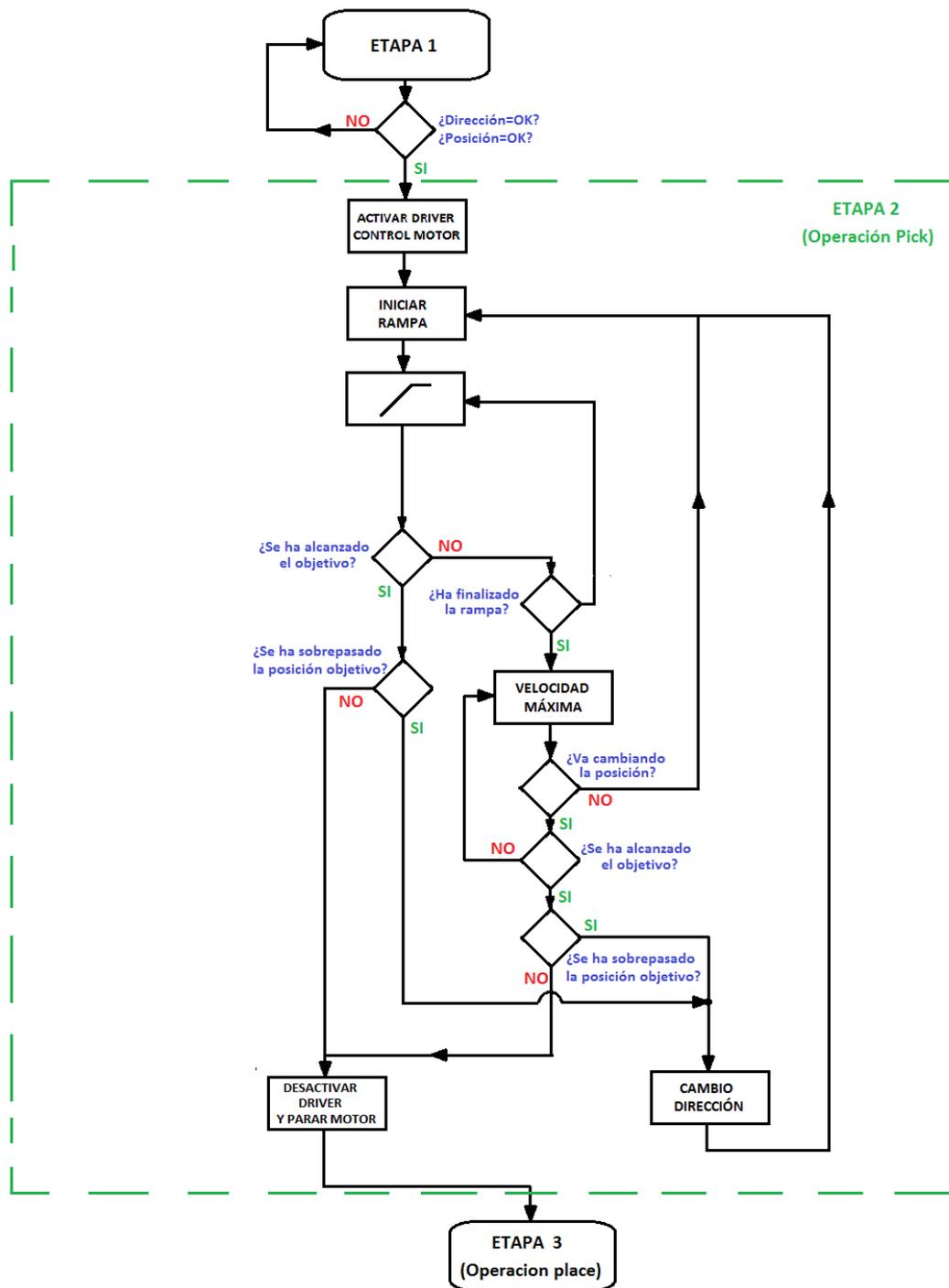


Figura 4.10. Flujograma de control para la etapa de operación Pick.

4.1.4 Operación Place

Una vez se recoge el producto de la cinta transportadora, este se debe depositar en un lugar establecido para su posterior empaquetado. Normalmente se suele depositar el producto en cajas dispuestas de manera ordenada que se encuentran en otra cinta transportadora de entrada a una máquina empaquetadora denominada “*flowpack*”, en la que una vez rellenas las cajas pasaran a empaquetarse. Es por tanto que, esta etapa tendrá la función de posicionar los objetos en la posición establecida de dejada.

Como se dispone de una guía lineal de una carrera efectiva de 100mm, se ha pensado asignar la posición de *place* de tal manera que el objeto quede a mitad de la guía (50mm). De esta forma, la distancia que deberá recorrer el TCP móvil tanto para esta operación, como para la siguiente recogida (siguientes operaciones *pick*) siempre será menor, y por lo tanto se optimizarán los tiempos de ciclo. No obstante, no habrá ningún problema en cambiar la posición asignada ya que se utilizará una variable parametrizable.

En definitiva, la operación *place* consistirá en organizar el producto tal y como se observa en la figura 4.11:

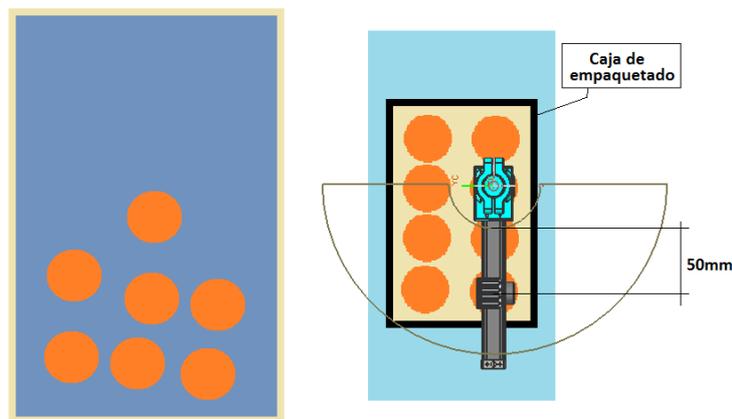


Figura 4.11. Esquema de la operación place con la distribución elegida.

El control de posición de la herramienta para la etapa de operación *place* se realizará de la misma manera que la etapa anterior explicada en el apartado 4.1.3. La única diferencia será, que no es necesario solicitar en cada iteración las coordenadas de los TCP fijo y móvil del robot y la herramienta respectivamente, ya que la posición de cada uno será siempre la misma (50mm para el carro de TCP móvil, y 270° para el TCP fijo). Las coordenadas de dejada en la caja de empaquetado para cada uno de los objetos serán determinadas por el programa de software Pickmaster.

Cuando se realiza el vacío entre las ventosas y el producto (*pick*) significa que éste ha sido agarrado. Es entonces cuando el controlador IRC5 de ABB activa una señal digital indicando

dicha situación. Se aprovechará esta señal digital para indicar a la aplicación de control de la herramienta que se debe realizar el posicionado del TCP móvil de la guía, el traslado del producto al lugar de destino y la dejada del mismo.

Para iniciar la operación de dejada, se utilizará un *flag* dentro de una interrupción externa del microcontrolador de Arduino, de tal manera que al activarse ésta se ejecute el control de los dispositivos, siguiendo la siguiente secuencia de pasos:

1. Comprobar la dirección de giro y asignarla al driver de control del motor paso a paso.
2. Activar el PWM ejecutando la rampa de arranque del motor.
3. Comprobar continuamente la posición hasta alcanzar la posición configurada como Place.
4. Paro del motor. Se desactiva el driver y a su vez la señal PWM.
5. Retorno a la primera etapa.

Al igual que en la etapa anterior, se ha realizado un flujograma (figura 4.12) en el que se observa la aplicación de control para esta etapa.

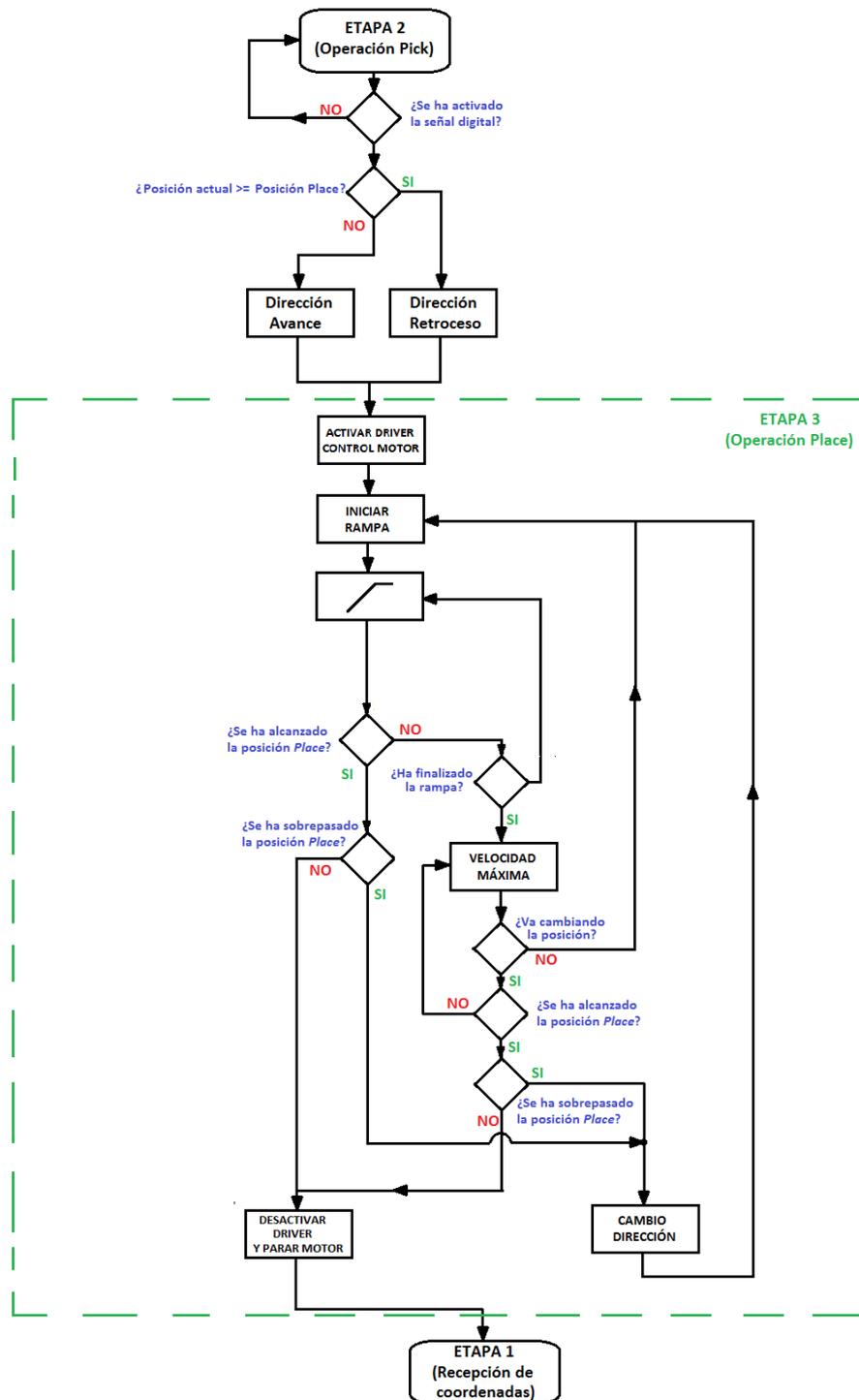


Figura 4.12. Flujograma de control para la etapa de operación Place.

4.2 Herramienta multi-pick triple (2 TCP móvil + 1 TCP fijo)

El desarrollo de esta herramienta se ha realizado partiendo de la herramienta multi-pick doble para aprovechar los dispositivos utilizados en ella, con la particularidad de utilizar dos motores brushless para realizar el giro de las guías lineales.

Se parte pues, de los siguientes elementos eléctricos, electrónicos y mecánicos, explicados cada uno de ellos en el punto 3.4:

- 2 Guías lineal SLN-27 de 100mm de carrera.
- 2 Motores paso a paso Nema 11 de 45V y 1 A.
- 2 Encoder incremental.
- 2 Finales de carrera.
- 2 Motores brushless "Gimbal".
- 2 Potenciómetros de 5K Ω .

Al igual que para la herramienta anterior, los compañeros del instituto de automática e informática industrial (AI2), mediante el programa CAD SolidWorks, han realizado el diseño de las piezas mecánicas para el acople de las guías y los motor al robot, quedando la herramienta multi-pick como se muestra en las figura 4.13:

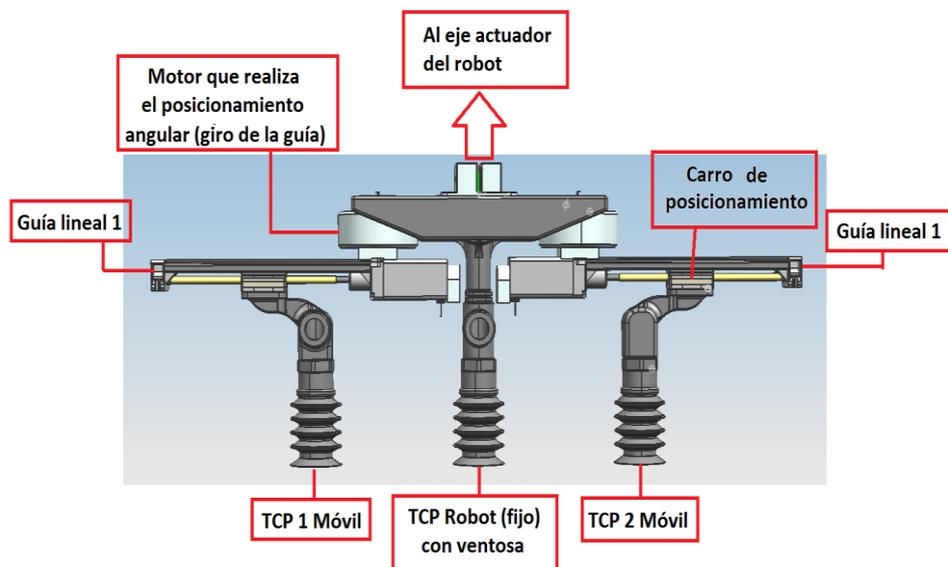


Figura 4.13. Herramienta Multi-pick circular con dos guías lineales.

Las ventosas utilizadas en este caso también serán las mismas que en la herramienta anterior (apartado 4.1).

Tal y como está montada la herramienta multi-pick en esta configuración, se ha realizado un plano mediante el programa de diseño *Siemens NX*, donde se muestran las diferentes áreas de trabajo que es posible abarcar para realizar la manipulación de objetos.

En la figura 4.14, se muestra una posible configuración donde es posible manipular objetos dentro de dos áreas de trabajo de 393cm^2 cada una. De todas formas, se podría llegar a ampliar el área efectiva de trabajo si se dimensiona el ángulo de trabajo de cada uno de los motores se puede conseguir un área de trabajo mayor (figura 4.15).

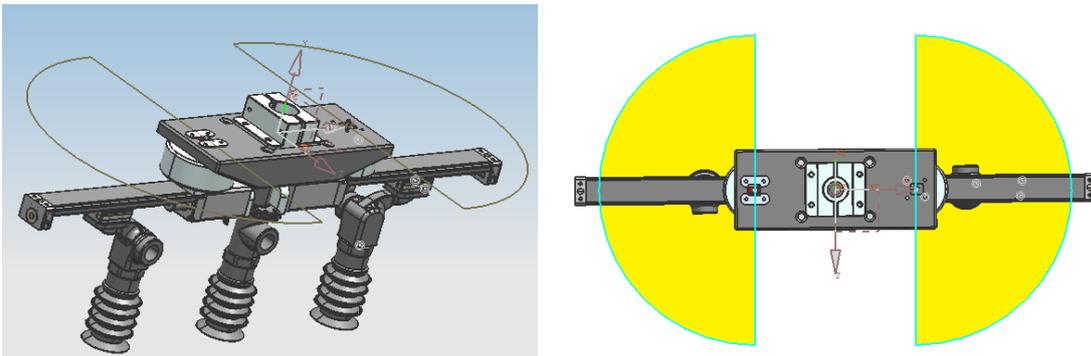


Figura 4.14. Área efectiva de la herramienta Multi-pick circular con dos guías lineales.

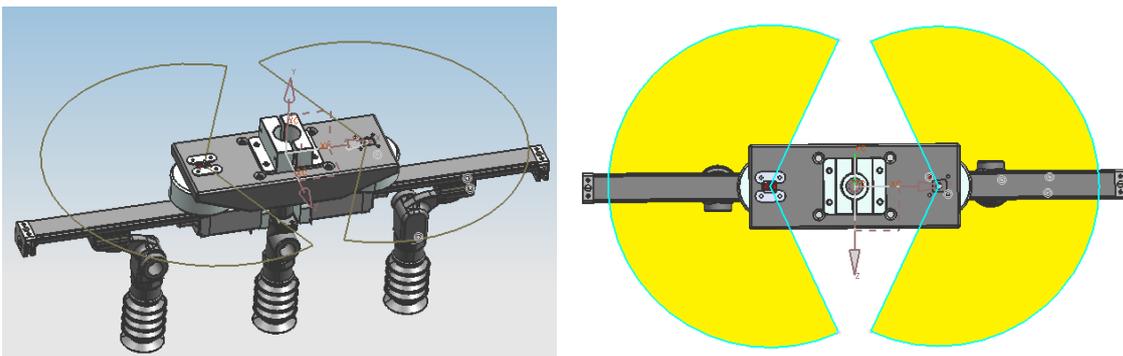


Figura 4.15. Área efectiva ampliada de la herramienta Multi-pick circular con dos guías lineales.

4.2.1 Inicio de operación.

El inicio de operación de la herramienta tiene el mismo fundamento que la herramienta doble (apartado 4.1.1), con la excepción de que también se debe de realizar un posicionado inicial del ángulo en el que cada guía lineal se posicionará para comenzar a trabajar. Es por ello que el primer paso que deba realizar la herramienta multi-pick será la de alcanzar una posición de inicio tanto del TCP móvil, como del ángulo de posición de cada guía. Esta posición determinará que la herramienta ya está preparada para trabajar y por lo tanto puede iniciarse el proceso de manipulación.

El programa arrancará moviendo el carro de cada guía lineal hacia la posición marcada como “posición de inicio” donde se encuentra colocado un final de carrera. Seguidamente se procederá a posicionar las guías mediante el motor brushless que dispone cada una. Para ello, se comprobará mediante la lectura del valor que marca un potenciómetro instalado en el eje de cada motor brushless. Cuando se llegue a la posición establecida, (por ejemplo 90°) es entonces cuando la herramienta puede empezar a trabajar.

4.2.2 Recepción de coordenadas.

Una vez que se alcanza la posición inicial de los dispositivos de posicionamiento y agarre de objetos “home position”, la aplicación de control entrará en modo de espera, en el que el sistema estará detenido hasta que le lleguen los por el puerto serie.

En Pickmaster se habrán determinado tres objetos a tomar, uno para el TCP fijo y los otros dos para los TCP’s móvil.

A diferencia de la herramienta doble, en la herramienta triple se debe anular la coordenada angular del cuarto eje del robot, ya que su TCP no requiere de orientación, pero a su vez se deberán recibir dos coordenadas (α -r) para cada módulo. Ya que los TCP móviles sí que requieren de orientación.

Se recibirán a través del puerto serie las 4 coordenadas que se habrán procesado en Pickmaster con el algoritmo de visión:

- **Coordenada angular 1 (α):** Corresponderá a la coordenada angular del módulo 1, y se le envía a través del puerto serie del PC al microcontrolador Arduino para que sea procesada.
- **Coordenada lineal 1 (r):** Corresponderá a la coordenada lineal del módulo 1, y se le envía a través del puerto serie del PC al microcontrolador Arduino para que sea procesada.
- **Coordenada angular 2 (α):** Corresponderá a la coordenada angular del módulo 2, y se le envía a través del puerto serie del PC al microcontrolador Arduino para que sea procesada.

- **Coordenada lineal 2 (r):** Corresponderá a la coordenada lineal del módulo 2, y se le envía a través del puerto serie del PC al microcontrolador Arduino para que sea procesada.

Una vez se recibe cada dato (byte) por el puerto serie, se convertirá a decimal y se deberá realizar el escalado para ser interpretado correctamente por cada encoder (apartado 4.1.2) así como el escalado de cada coordenada angular para ser interpretadas por cada potenciómetro.

Antes de ejecutar los movimientos para posteriormente realizar la operación pick, se debe conocer la dirección (avance o retroceso) que debe tomar cada TCP móvil, así como la dirección en la que debe girar el motor brushless para realizar el giro adecuadamente. Tanto las distancias a recorrer como los ángulos a girar se deben comparar con las posiciones y ángulos actuales en la que se encuentren los TCP móvil y los motores brushles y entonces determinar la dirección que debe tomar éste. Esta es otra de las tareas que se ejecutará una vez recibida la coordenada, donde los pasos a realizar son los mismos que se observan en la figura 4.5 del apartado 4.1.2, salvo que en lugar de recibir una coordenada se reciben 4.

Para recibir las coordenadas se dispondrá de un vector que guardará en sus posiciones las coordenadas recibidas por el puerto serie. Una vez recibidas, se deberá realizar el escalado de las mismas para poder compararlas con los valores de los potenciómetros y los encoders.

En el caso del escalado de las coordenadas lineales se realizará de la misma manera que en la herramienta doble, pero en el caso de las coordenadas angulares se procederá a realizar el escalado de la siguiente manera:

- El potenciómetro disponible es un potenciómetro convencional sin topes de 5KΩ, que mediante su lectura a través del pin analógico del Arduino al que está conectado nos proporcionará una resolución de 10 bits para valores comprendidos entre 0 y 360°, por lo que el valor del ángulo recibido deberá ser multiplicado por:

Valores del potenciómetro: [0 – 1023]

Medida en grados: [0 - 360°]

$$\text{Escalado} = \frac{1023}{360} = 2.841$$

Por ejemplo, un ángulo de 10° tendrá un valor de:

$$\text{Valor [10°]} = 10 * 2.841 = 284.1$$

Que al utilizar variables *int* se corresponderá con el valor 284.

Cabe destacar que para el prototipo de herramienta a elaborar en el presente proyecto no se requiere precisión en la aplicación, debido a que los objetos a manipular son de geometría

circular, por lo que no se tienen en cuenta los pequeños errores que se pueden producir en el escalado de variables, ya que estos no ocasionarán inconveniente alguno a la hora de realizar el agarre del producto.

4.2.3 Operación Pick

La operación pick tiene el mismo principio de funcionamiento que el explicado en el apartado 4.1.3, sólo que para esta herramienta, se controlarán dos módulos (guía lineal + encoder + motor brushless) en los que se determinará la posición de dos TCP móviles.

Tenemos pues, dos guías lineales amarradas cada una a su motor brushless, donde el posicionamiento lineal (coordenadas lineales) de cada TCP móvil se realizará de la misma manera que en los apartados 4.1.3.1 y 4.1.3.2 una vez recibidas las coordenadas lineales pertenecientes a cada módulo.

Además, se han obtenido también las coordenadas angulares de cada módulo, y la función de los motores brushless será la de posicionar cada guía en la coordenada establecida.

4.2.3.1 Control del motor brushless.

El primer paso a realizar para controlar este tipo de motores deberá ser proporcionarle las tres señales adecuadas que le permita tener una precisión de 1° por paso, con un total de 360° por vuelta.

Para ello, se deberán generar a través de Arduino tres señales PWM. Cada PWM deberá ser capaz de generar una señal sinusoidal, formando así tres señales sinusoidales. Además, estas estarán desfasadas 120° entre sí, tal y como se muestra en la siguiente figura:

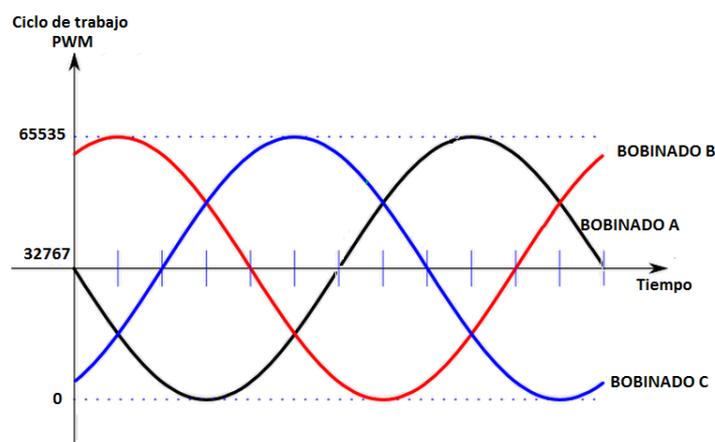


Figura 4.16. Señales senoidales para el control del motor Brushless.

Observando la figura 4.16 se tiene, que mediante tres señales PWM de 16 bits de precisión, se van generando paso a paso cada una de las señales sinusoidales desfasadas.

La manera de realizar estas tres señales se detalla a continuación:

1. Mediante un programa de cálculo se ha obtenido un vector que contiene cada uno de los valores para generar la señal senoidal con la función seno. (Para este caso se ha elegido concatenar 360 valores).
2. Una interrupción interna del microcontrolador ha sido programada para que ejecute una función a una frecuencia establecida. Al disponer de un vector que contiene la señal sinusoidal de funcionamiento del motor, la frecuencia de rotación de éste será igual a la frecuencia de la interrupción interna multiplicada por el número de posiciones que se recorren del vector establecido.
3. En dicha función se ejecutarán las tres señales PWM. Para cada una, se le asignará cada vez que se ejecute la interrupción un valor de ciclo trabajo diferente, que corresponderá a la posición en la que se encuentre el vector calculado en el paso 1.
4. Como el vector dispone de 360 posiciones, la primera vez que se ejecute el programa y a la vez la interrupción, a la primera PWM (PWM1) se le asignará la primera posición del vector, a la PWM2 la posición 120 y a la PWM3 la posición 240, consiguiendo así que estén desfasadas entre sí 120°.

Para determinar la frecuencia de la interrupción se deberá hacer uso del *timer* asociado a ella. El microcontrolador ATmega 2560 de Arduino contiene 6 *timers*, de los cuales tres de ellos (3, 4, y 5) contienen un temporizador de 16 bits. Este temporizador contiene un contador interno que puede funcionar a la frecuencia que marca un reloj.

Funciona mediante un aumento del "*counter register*" que según como se configure, su contaje será a una frecuencia mayor o menor y una vez finalice el contaje (desbordamiento) para el que lo hemos configurado, empezará de nuevo. Debido a que el timer depende de una fuente de reloj, tal y como se ha dicho al principio, la unidad más pequeña medible será el periodo:

$$T = \frac{1}{f} = \frac{1}{16MHz} = 62.5nS$$

Para poder usar los *Timers* o contadores, deberemos modificar los registros:

- TCCRnA: Donde se elegirá el modo de operación (PWM, CTC...). Para este caso se elegirá el modo contador.

- TCCRnB: Donde se elige la resolución del *timer* mediante un pre-escalado. Para este caso se ha elegido un $CLK_{I/O}=1024$.
- TMSK: Se configura el valor con el que se comparará el contador.

Por ejemplo, si se quiere configurar el *timer* a una frecuencia de 2KHz, se deberá poner en el contador CTC el siguiente valor:

Se tiene un pre-escalado de 1024, por lo que la resolución del *timer* será:

$$Resolución = \frac{1}{f/prescalado} = \frac{1}{16MHz/1024} = \frac{1024}{16MHz} = 64\mu s$$

Y por lo tanto el valor del CTC:

$$CTC = \frac{Tiempo\ a\ establecer}{Resolución} - 1 = \frac{2000Hz}{64\mu s} = \frac{1}{64\mu s \cdot 2000Hz} - 1 = 6.81$$

Este valor se almacenará en el registro OCR1A, el cual se comparará constantemente con el *Timer*.

Otro aspecto a tener en cuenta, es la precisión que dispone el motor brushless por su composición electromecánica. Este motor está constituido por 12 bobinas eléctricas y 14 polos magnéticos (figura 4.17), siendo capaz de posicionarse en 36 posiciones diferentes, es decir, una precisión de 10° . Para aumentar la precisión es necesario controlar los campos magnéticos del motor para conseguir así mayor suavidad en la rotación y por lo tanto mayor precisión. Es por ello que se han establecido las 360 posiciones del vector con el que se asignarán los ciclos de trabajo de las PWM.

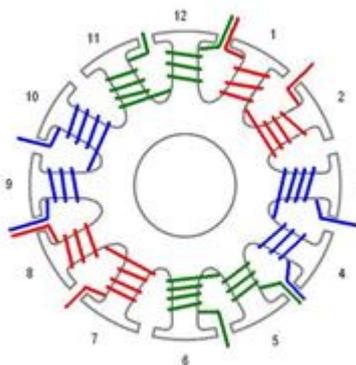


Figura 4.17. Bobinas eléctricas del motor brushless.

Mediante la realización de pruebas y experimentos en el laboratorio, se han encontrado los valores para determinar la precisión del motor (ver apartado 7), llegando a la conclusión de que para realizar una vuelta completa del motor, se deben recorrer 1265 posiciones del vector (recorrerlo 3,5 veces). Por lo tanto se deberán recorrer aproximadamente 3 posiciones para alcanzar la posición de 1° .

Las señales PWM deben pasar por el controlador L634D para ser amplificadas y adecuar así las corrientes al motor. Además, este driver dispone de una señal de habilitación (Enable) que se utilizará para activar y desactivar el motor a través de software. Los motores serán alimentados mediante una fuente de alimentación, cuyo voltaje será de 15V en corriente continua.

Hasta ahora, se tiene la acción de control que se le debe aplicar al motor para llegar a una posición angular, no obstante, se debe verificar que se ha llegado a una posición angular objetivo. Para ello se ha situado un potenciómetro en el eje del motor que mediante la lectura del mismo con el microcontrolador de Arduino es capaz de proporcionarnos una precisión de 10 bits (0-1023 posiciones). Mediante la comparación continua del valor obtenido con el potenciómetro se podrá determinar si se ha producido algún error de posicionado y actuar en consecuencia de ello, al igual que se hace mediante el encoder y el motor paso a paso de la guía lineal.

Para alcanzar la posición deseada, se utilizará un control de velocidad en el que se determinará la frecuencia de rotación del motor dependiendo de la posición de referencia y del error calculado a través del potenciómetro.

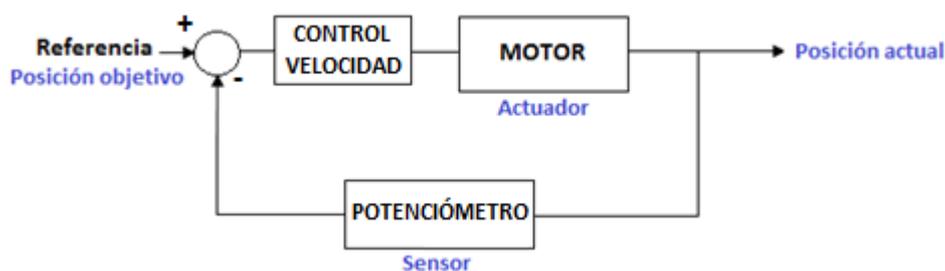


Figura 4.18. Control de posición del Brushless.

La primera etapa (recepción de coordenadas) nos proporciona las distancias que deben recorrer los TCP móviles, las posiciones angulares que deben alcanzar los motores brushless, así como la dirección de giro que debe tomar cada motor. Una vez se tienen estos dos datos, se seguirán los siguientes pasos:

1. Asignarle a los drivers de los motores paso a paso la dirección de giro

2. Activar el driver de control del motor de la guía lineal 1 y activar el PWM ejecutando la rampa de arranque del motor.
3. Comprobar continuamente la posición hasta alcanzar el objetivo (coordenada lineal del TCP móvil 1).
4. Paro del motor paso a paso 1. Se desactiva el driver y a su vez la señal PWM.
5. Activar el driver de control del motor de la guía lineal 2 y activar el PWM ejecutando la rampa de arranque del motor.
6. Comprobar continuamente la posición hasta alcanzar el objetivo (coordenada lineal del TCP móvil 2).
7. Paro del motor paso a paso 2. Se desactiva el driver y a su vez la señal PWM.
8. Activar el driver de control del motor brushless 1. Comprobar continuamente la posición angular hasta alcanzar el objetivo (posición angular del TCP móvil 1).
9. Desactivar driver.
10. Activar el driver de control del motor brushless 2. Comprobar continuamente la posición angular hasta alcanzar el objetivo (posición angular del TCP móvil 2).
11. Desactivar driver.

En la figura 4.19 se muestra el flujograma de la programación de control, en el que se pueden observar las diferentes tareas y condiciones que se deben realizar y tener en cuenta para esta etapa. Una vez se ejecuten todos los pasos, se procederá a ejecutar la etapa 3.

El posicionamiento lineal de los TCP móviles se realizará de la misma manera que la herramienta doble y de forma secuencial, de tal manera que primero se posicionará el TCP de la herramienta 1 y acto seguido el de la herramienta 2. Una vez se posicionen los TCP de las guías, se procederá a posicionarlas en la coordenada angular que le corresponde a cada una mediante los brushless.

En la figura 4.19 se ha querido remarcar el control de velocidad en rojo, para seguidamente explicar su funcionamiento, el cual se puede observar de manera gráfica en el siguiente grafo:

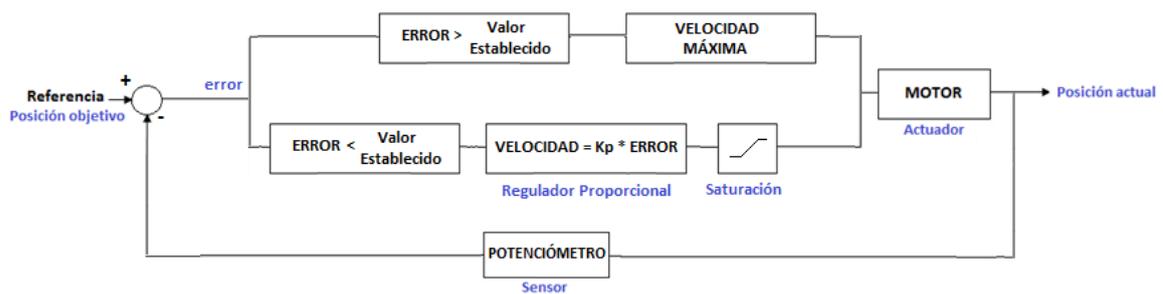


Figura 4.20. Control de velocidad del motor Brushless.

Además, para cada medida del error se comprobará si dicho error es igual al anterior y si es así, se incrementará un contador. Si el contador se llega a incrementar 3 veces, significará que el motor se ha quedado enclavado produciéndose así un fallo en la operación. Se deberá encender un led indicando dicha situación.

4.2.4 Operación Place

El control de posición de la herramienta para la etapa de operación place se realizará de la misma manera que se ha explicado en el apartado 4.1.4, y el proceso se ejecutará con los mismos objetivos especificados en el apartado anterior (4.2.3). La diferencia será, que no es necesario solicitar en cada iteración las coordenadas de los TCP móviles de la herramienta, ya que la posición de cada uno será siempre la misma (50mm para los carros de TCP móvil, y 90° para los motores brushless). Las coordenadas de dejada en la caja de empaquetado para cada uno de los objetos serán determinadas por el programa de software Pickmaster, de tal manera que los objetos queden distribuidos como se ha representado en la figura 4.21.

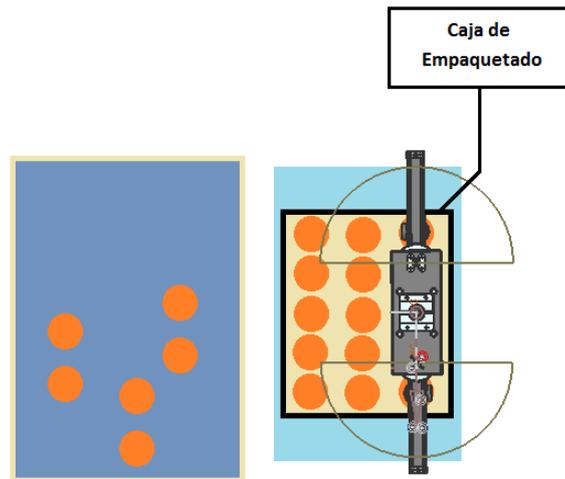


Figura 4.21. Esquema de la operación place con la distribución elegida.

Al igual que en la herramienta doble, para iniciar la operación de dejada, se utilizará un flag dentro de una interrupción externa del microcontrolador de Arduino, de tal manera que al activarse ésta se ejecute el control de los dispositivos, siguiendo la siguiente secuencia de pasos.

1. Asignarle a los drivers de los motores paso a paso la dirección de giro
2. Activar el driver de control del motor de la guía lineal 1 y activar el PWM ejecutando la rampa de arranque del motor.
3. Comprobar continuamente la posición hasta alcanzar el objetivo (posición place del TCP móvil 1).
4. Paro del motor paso a paso 1. Se desactiva el driver y a su vez la señal PWM.
5. Activar el driver de control del motor de la guía lineal 2 y activar el PWM ejecutando la rampa de arranque del motor.
6. Comprobar continuamente la posición hasta alcanzar el objetivo (posición place del TCP móvil 2).
7. Paro del motor paso a paso 2. Se desactiva el driver y a su vez la señal PWM.
8. Activar el driver de control del motor brushless 1. Comprobar continuamente la posición angular hasta alcanzar el objetivo (posición place del TCP móvil 1).
9. Desactivar driver.
10. Activar el driver de control del motor brushless 1 . Comprobar continuamente la posición angular hasta alcanzar el objetivo (posición place del TCP móvil 2).
11. Desactivar driver.

5. MONTAJE Y PROGRAMACIÓN DE LA HERRAMIENTA

5.1 Montaje y programación de la herramienta doble.

Para realizar el montaje mecánico de la herramienta multi-pick doble se dispone de los siguientes elementos:

- **Guía lineal con motor paso a paso y encoder incremental:**



Figura 5.1. Guía lineal SLN-27 con motor y encoder.

- **Soporte de acople TCP robot – Guía lineal:**

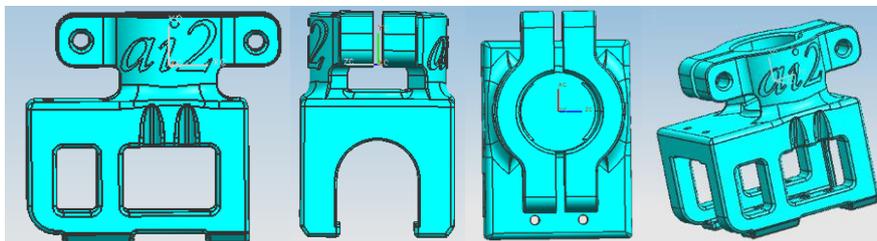


Figura 5.2. Elemento de acople del robot a la guía.

- **Elemento de acople ventosas – TCPs con conexiones para neumática:**

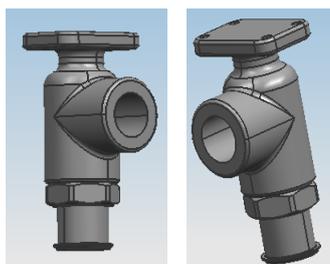


Figura 5.3a. Elemento de acople ventosas y TCP móvil.

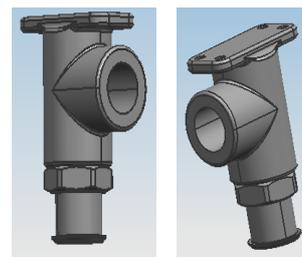


Figura 5.3b. Elemento de acople ventosas y TCP fijo

- Final de carrera, ventosas y pieza tope para final de carrera:



Figura 5.4. Final de carrera, ventosa y pieza tope para final de carrera.

Los soportes de acople del TCP robot a la guía lineal, así como los elementos de acople de las ventosas y la pieza tope para el final de carrera, se han diseñado específicamente para esta herramienta en el instituto de automática e informática industrial (AI2) y fabricado por sinterizado láser mediante una impresora 3D. Todos ellos se han montado en conjunto para completar la herramienta, la cual se puede observar en la siguiente figura, donde se ha realizado su montaje real sobre el robot:



Figura 5.5. Montaje real de la herramienta multi-pick doble acoplada sobre el robot delta IRB360 ABB.

Una vez montada la herramienta y colocada sobre el robot, se deben realizar las conexiones a los diferentes dispositivos de control. En la figura 5.6, se observa el esquema completo de conexiones entre los elementos que componen todo el sistema automático de manipulación.

En el centro del esquema se encuentra la herramienta de manipulación que irá conectada al robot, al Arduino y al driver de control del motor. El robot irá conectado al controlador IRC5 donde se encuentra la alimentación así como los buses de comunicación y demás elementos de control. Además, el controlador IRC5 se conecta a través de una red de Ethernet, al PC donde estará instalados el sistema operativo Windows 7 y los programas Pickmaster y Arduino 1.0.5.

El sistema de visión también está conectado al PC, vía Ethernet. El driver de control DRV8825, alimentado a 45 V a través de la fuente de alimentación, se conecta al Arduino para recibir la información procedente del microcontrolador.

Este último, se conecta al PC a través del puerto serie USB, para recibir la alimentación de 5V necesaria para funcionar, así como para recibir los datos de posicionamiento además de para cargar el software de control implementado.

Se han conectado también un pulsador de inicio con un indicador led para saber que el sistema está en proceso, y otro led que indicará si se ha producido un fallo.

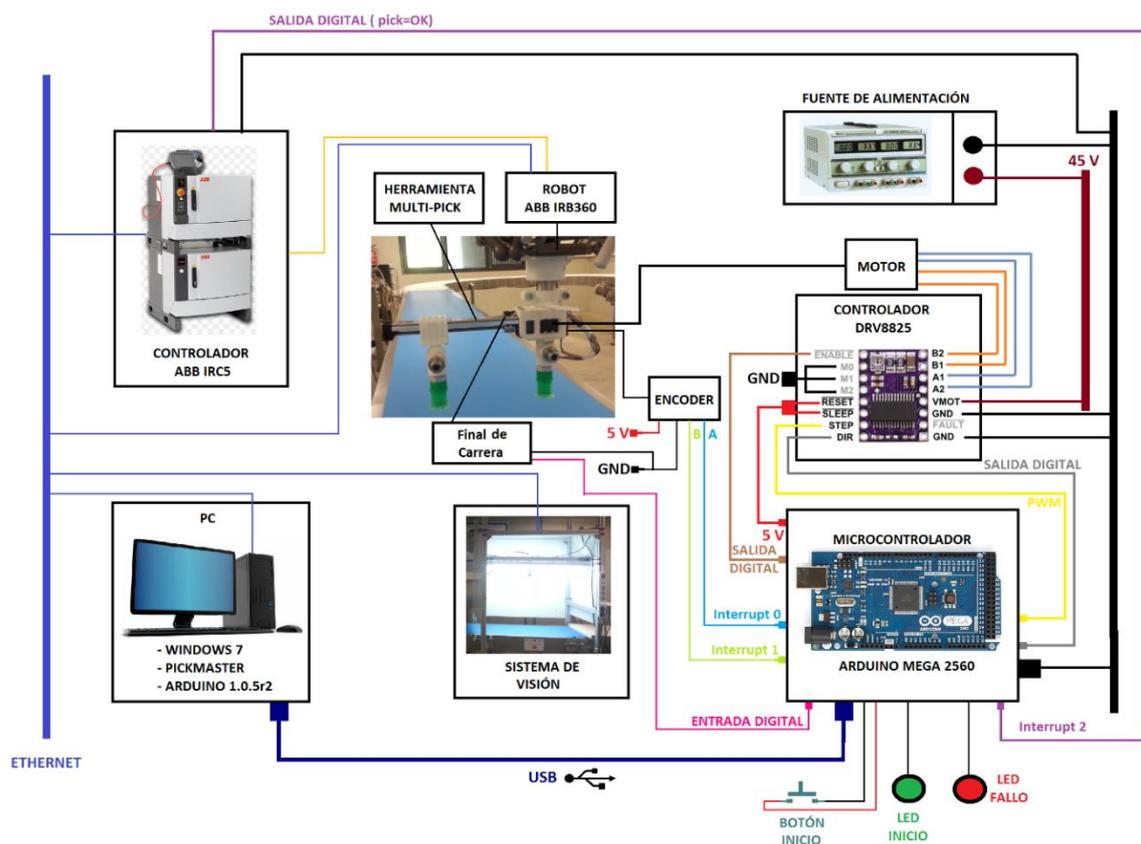


Figura 5.6. Esquema de conexiones del sistema de manipulación completo.

A continuación, se muestra una tabla con las entradas y salidas utilizadas en el arduino para cada elemento conectado.

Conexiones al Arduino Mega 2560		
Tipo de entrada/salida	Número	Dispositivo conectado
Entrada digital	27	Pulsador de inicio
Entrada digital	28	Final de carrera
Salida digital	13	Led indicador inicio programa
Salida digital	14	Led indicador fallo en la guía
Salida digital	30	Habilitación del DRV8825 (Enable)
Salida digital	31	Bit de dirección de giro DRV8825
PWM	8	PWM para el DRV8825 (Step)
Interrupción externa	2	Interrupción 0, Señal encoder A
Interrupción externa	3	Interrupción 1, Señal encoder B
Interrupción externa	21	Interrupción 2, Señal digital fin de pick (IRC5)

Tabla 5.1. Conexiones al Arduino Mega.

Seguidamente, se procede a explicar el código de programación para la aplicación de control de la herramienta que se ha implementado en el entorno de desarrollo IDE de Arduino 1.0.5-r2.

El primer paso es incluir las librerías que usa el código y definir la numeración de las entradas y salidas para mayor comodidad en la programación:

```

//***** LIBRERÍAS Y DEFINICIONES *****/

#include <PWM.h> //Librería
#include <avr.interrupt.h>
#define DIR 31 // Bit de dirección de giro
#define STEP 8 // PWM motor paso a paso (DRV8825)
#define ENABLE 30 //Habilitación del DRV8825
#define FIN_CARRERA 28 // Final de carrera
#define BOTON_PRINCIPAL 27 // Pulsador de inicio
#define LED_INICIO 13
#define LED_FALLO 14
    
```

Se declaran también las variables globales a utilizar:

```

//*****DECLARACIÓN DE VARIABLES GLOBALES*****/

long posicion=0, distancia, A=0, B=1, recorrido=0, posicion_pick, escalado=400;
long posicion_ant=0;
int posicion_place=20000, cont=0, arranque;
int estado=0, i, int2 = 0, inicio=0, direccionpick, direccion, arranquehome=1, PRINCIPAL=1;
float f;
float fh=500, fhmax=1000;

//rampa motor PaP

float fmax1=1999;
float fmax2=2499;
float fmax3=2999;
    
```

Antes de la ejecución del programa, se deben configurar los pines de las entradas, salidas, interrupciones, etc, así como los estados en los que se deben inicializar los mismos. Para ello Arduino dispone de la función **void setup ()**. Esta función se ejecuta una única vez al iniciarse el programa y en ella se ha implementado el siguiente código:

```

//***** PREPARACIÓN Y ARRANQUE DE PROGRAMA *****//
void setup()
{
  InitTimersSafe(); // Inicialización para poder usar PWM
  Serial.begin(115200); //Inicio de comunicación puerto serie

  pinMode(BOTON_PRINCIPAL, INPUT_PULLUP); //Declaramos pulsador como entrada

  //Declaramos led inicio y fallo como salidas y las ponemos a nivel bajo (apagados):

  pinMode(LED_INICIO,OUTPUT);
  digitalWrite(LED_INICIO,LOW);

  pinMode(LED_FALLLO,OUTPUT);
  digitalWrite(LED_FALLLO,LOW);

  //DRIVER MOTOR PASO A PASO
  // asignamos las salidas digitales para el driver DRV8825:

  pinMode(DIR, OUTPUT);
  pinMode(STEP, OUTPUT);
  pinMode(ENABLE, OUTPUT);
  digitalWrite(ENABLE, HIGH);

  //Posición inicio (home position)

  pinMode(FIN_CARRERA, INPUT); // Asignamos el final de carrera a una entrada digital
  digitalWrite(FIN_CARRERA, HIGH); //pullup

  //INTERRUPCIONES ENCODER

  attachInterrupt(0, A_UP, RISING); //Interrupción señal A, activado a flanco de subida
  attachInterrupt(1, B_UP, RISING); //Interrupción señal B, activado a flanco de subida

  //INTERRUPCIONES ROBOT

  pinMode(21,INPUT_PULLUP);
  attachInterrupt(2, PLACE, FALLING); // Se activa la interrupción por flanco de bajada.
}
    
```

A continuación se muestran las diferentes funciones que se han implementado y se utilizarán en programa principal.

- **Función de posición inicial (Home position):**

Será el primer paso que ejecutará la aplicación y consistirá en posicionar el TCP móvil en la posición de inicio (extremo de la guía lineal).

Esta función se encarga de mover el TCP móvil en la dirección de retroceso (LOW) hasta que se produzca contacto con el final de carrera. Cuando el final de carrera cambie de estado (de HIGH a LOW), se detendrá el motor y se inicializará la variable *posición* a 0. Para realizar el movimiento se aplicará una pequeña rampa de aceleración hasta llegar a una frecuencia de 1KHz.

```
//-----POSICION HOME INICIO-----//

void homeposition()
{
  if(estado==HIGH)//Estado del final de carrera
  {
    digitalWrite(ENABLE,LOW); // Habilitación del driver
    digitalWrite(DIR,LOW); // Dirección de giro (retroceso)
    if(fh<=fhmax) // Rampa de aceleración para arranque del motor
    {
      SetPinFrequency(STEP,fh);
      pwmWriteHR(STEP, 32768);
      delayMicroseconds(50);
      fh=fh+1;
    }
    else
    {
      SetPinFrequency(STEP,1000); //Velocidad motor 1000Hz
      pwmWriteHR(STEP, 32768);
    }
  }
  else
  {
    digitalWrite(STEP, LOW);
    digitalWrite(ENABLE, HIGH); // Se deshabilita el driver
    arranquehome=0; //Cambio el valor de variable para que no se repita el home position
    posicion=0; // Inicialización de posición
  }
}
```

- **Funciones lectura de posición encoder:**

Se muestra el código implementado donde dependiendo en el estado que se encuentre cada impulso de señal de los sensores ópticos del encoder, saltará una de las dos interrupciones externas y ejecutará su función asociada. En el apartado 4.1.3.1 ya se ha explicado el funcionamiento de las interrupciones así como la lectura del encoder.

```

//-----INTERRUPCIONES ENCODER-----//
void A_UP()
{
    detachInterrupt(0); //deshabilitación de interrupción 0
    A=1;
    if(B==0)
        posicion++; // AVANCE (incremento posición)
    if(B==1)
        posicion--; // RETROCESO (decremento posición)
    attachInterrupt(0, A_DOWN, FALLING); //Activación de interrupción 0 a flanco bajada
}

void A_DOWN()
{
    detachInterrupt(0); //deshabilitación de interrupción 0
    A=0;
    if(B==1)
        posicion++; // AVANCE
    if(B==0)
        posicion--; // RETROCESO
    attachInterrupt(0, A_UP, RISING); //Activación de interrupción 0 a flanco Subida
}

void B_UP()
{
    detachInterrupt(1); //deshabilitación de interrupción 1
    B=1;
    if(A==1)
        posicion++; // AVANCE
    if(A==0)
        posicion--; // RETROCESO
    attachInterrupt(1, B_DOWN, FALLING); //Activación de interrupción 1 a flanco bajada
}

void B_DOWN()
{
    detachInterrupt(1); //deshabilitación de interrupción 1
    B=0;
    if(A==0)
        posicion++; // AVANCE
    if(A==1)
        posicion--; // RETROCESO
    attachInterrupt(1, B_UP, RISING); //Activación de interrupción 1 a flanco Subida
}
    
```

- **Movimiento *pick* con rampa de velocidad:**

Esta función será la encargada de realizar la rampa de aceleración con velocidad lineal, así como mantener la velocidad máxima del motor una vez haya finalizado la rampa y hasta que se alcance la posición objetivo.

Se tienen 3 variables $f_{max1}=1999$, $f_{max2}=2499$ y $f_{max3}=2999$, las cuales delimitan las condiciones de la rampa. Como se observa en el código siguiente, cada condición está implementada con un bucle *while* que comprobará:

- Si se ha llegado a la frecuencia establecida.
- Si se ha alcanzado la posición.

- Si se ha sobrepasado la posición y a su vez se ha realizado un cambio de dirección.

Una variable $f=700$ se irá incrementando mientras se encuentre dentro de dichas condiciones y hasta alcanzar la frecuencia máxima de 3KHz.

Si se produce un fallo y el motor no se mueve se incrementará un contador. Cuando esta situación se produzca 3 veces seguidas, se reiniciará la rampa de nuevo y se incrementará otro contador (*cont_fallo*). Es entonces cuando se realizarán 3 intentos de reinicio de rampa para recuperar el movimiento del motor y, de no ser así, se desactivará el driver y se encenderá el led de alarma indicando que se ha producido un fallo.

```
//-----MOVIMIENTO PICK CON RAMPA DE VELOCIDAD-----//
void rampick()
{
  while ((f<=fmax1)&&(((posicion >= recorrido) && (direccionpick == -1)) || ((posicion <= recorrido) && (direccionpick == 1))))
  {
    SetPinFrequency(STEP,f);
    pwmWriteHR(STEP, 32768);
    f=f+350;
  }

  while ((f<=fmax2)&&(((posicion >= recorrido) && (direccionpick == -1)) || ((posicion <= recorrido) && (direccionpick == 1))))
  {
    SetPinFrequency(STEP,f);
    pwmWriteHR(STEP, 32768);
    f=f+350;
  }

  while ((f<=fmax3)&&(((posicion >= recorrido) && (direccionpick == -1)) || ((posicion <= recorrido) && (direccionpick == 1))))
  {
    SetPinFrequency(STEP,f);
    pwmWriteHR(STEP, 32768);
    f=f+350;
  }

  while ((f>fmax3)&&(((posicion >= recorrido) && (direccionpick == -1)) || ((posicion <= recorrido) && (direccionpick == 1))))
  {
    SetPinFrequency(STEP,3000);
    pwmWriteHR(STEP, 32768);

    if (fallo>=3)
    {
      LED_FALLO=HIGH; // Indicación de que se ha producido un fallo
      digitalWrite(ENABLE, HIGH); // Desactivamos driver
    }

    if (posicion_ant==posicion)// Si no se mueve el motor
    {
      if (cont>=3)
      {
        f=700; //Iniciamos rampa de nuevo
        fallo++;
      }
      else
      {
        cont++;
      }
    }

    else
    {
      posicion_ant=posicion;
      cont=0;
      cont_fallo=0;
    }
  }
}
```

Esta función se ha implementado de esta manera ya que al tener una condición para cada variable $f_{max}(1,2,3)$, se puede parametrizar y cambiar la velocidad de la rampa si se desea, de esta forma si se cambian algunas características de la herramienta y los objetos a agarrar, solamente habrá que modificar los valores.

- **Movimiento *place* con rampa de velocidad:**

El funcionamiento es exactamente igual que la función de movimiento pick con rampa de velocidad, solo que se para este caso, en las condiciones se incluye la posición de dejada en lugar de la posición de agarre del producto.

```

//-----MOVIMIENTO PLACE CON RAMPA DE VELOCIDAD-----//
void ramplace()
{
  while ((f<=fmax1)&&((posicion >= posicion_place) && (direccion == -1)) || ((posicion <= posicion_place) && (direccion == 1)))
  {
    SetPinFrequency(STEP,f);
    pwmWriteHR(STEP, 32768);
    f=f+350;
  }
  while ((f<=fmax2)&&((posicion >= posicion_place) && (direccion == -1)) || ((posicion <= posicion_place) && (direccion == 1)))
  {
    SetPinFrequency(STEP,f);
    pwmWriteHR(STEP, 32768);
    f=f+350;
  }
  while ((f<=fmax3)&&((posicion >= posicion_place) && (direccion == -1)) || ((posicion <= posicion_place) && (direccion == 1)))
  {
    SetPinFrequency(STEP,f);
    pwmWriteHR(STEP, 32768);
    f=f+350;
  }
  while ((f>fmax3)&&((posicion >= posicion_place) && (direccion == -1)) || ((posicion <= posicion_place) && (direccion == 1)))
  {
    SetPinFrequency(STEP,3000);
    pwmWriteHR(STEP, 32768);

    if (fallo>=3)
    {
      LED_FALL0=HIGH; // Indicación de que se ha producido un fallo
      digitalWrite(ENABLE, HIGH); // Desactivamos driver
    }

    if (posicion_ant==posicion)// Si no se mueve el motor
    {
      if (cont>=3)
      {
        f=700; //Iniciamos rampa de nuevo
        fallo++;
      }
      else
      {
        cont++;
      }
    }

    else
    {
      cont++;
    }
  }

  else
  {
    posicion_ant=posicion;
    cont=0;
    cont_fallo=0;
  }
}
}

```

- **Interrupción *place*:**

Esta sencilla función se activa cuando se produce la interrupción de la señal digital que envía el controlador del robot cuando se produce el agarre del producto. Simplemente es un *flag*, de tal manera que se ejecute la función cambiará el estado de la variable *int2*, permitiendo así que se ejecute la operación *place* en el programa principal.

```
//-----INTERRUPCION PLACE-----//
void PLACE()
{
    int2=1;
}
```

Explicadas todas las funciones, se muestra a continuación el programa principal de la herramienta.

Este código se implementa dentro de la función ***loop()*** del entorno de Arduino. Todo lo que esté implementado dentro de esta función se ejecutará de manera secuencial y de forma cíclica y no se detendrá hasta que no se le dé una orden, o al desconectar el microcontrolador de la alimentación.

Al iniciarse el programa, éste esperará a que el pulsador de inicio sea pulsado. De ser así, la primera función que se ejecutará será la de posicionamiento inicial (*homeposition*).

Seguidamente se procederá a realizar la manipulación de objetos mediante las etapas explicadas en el apartado 4.1.

Una condición *while* determinará si la operación a realizar es la de *pick* o la de *place*. Si el puerto serie está disponible porque se ha recibido algún dato, se realizará la lectura de coordenadas y seguidamente la función *pick*. Si por el contrario no se ha recibido ningún dato, entonces se realizará la función *place* cuando esta haya sido requerida por la interrupción.

```
//-----PROGRAMA PRINCIPAL-----//
void loop()
{
    //-----INICIO-----//
    while (PRINCIPAL==1)//Pulsador de inicio.
    {
        if (digitalRead(BOTOM_PRINCIPAL)==HIGH)
        {
            digitalWrite(LED_INICIO,LOW);
            PRINCIPAL=1;
        }
        else if (digitalRead(BOTOM_PRINCIPAL)==LOW)
        {
            digitalWrite(LED_INICIO,HIGH);
            PRINCIPAL=0;
        }
        delay(200);
    }
}
```

```

//.....POSICIÓN HOME.....//

if (arranquehome==1)
{
    estado=digitalRead(FIN_CARRERA);
    homeposition();
}

//.....MOVIMIENTO PICK&PLACE.....//
else
{
    while (Serial.available() == 0)
    {
        digitalWrite(ENABLE, HIGH);
        if (int2 == 1)//Si se activa la interrupción 2 --> FUNCIÓN PLACE
        {
            int2=0;
            digitalWrite(ENABLE, LOW);

            if (posicion>posicion_place) //elegimos dirección de retroceso
            {
                digitalWrite(DIR,LOW);
                direccion = -1;
            }

            else if (posicion<posicion_place)//elegimos dirección de avance
            {
                digitalWrite(DIR,HIGH);
                direccion = 1;
            }

            f=700;

            while (((posicion >= posicion_place) && (direccion == -1)) || ((posicion <= posicion_place) && (direccion == 1)))
            {

                ramplace(); //Llamamos a la función que realiza la rampa y movimiento motor en el Place

            }

            digitalWrite(ENABLE, HIGH);
            digitalWrite(STEP, LOW);

        }

    }

    distancia = Serial.read(); //Si recibimos dato (distancia) por el puerto serie --> PICK

    if ((distancia==100)|| (distancia==99))//Seguridad para que no de golpes el carro contra el tope guía
    {
        distancia=distancia-2;
        recorrido = distancia * escalado;
    }
    else
    {
        recorrido = distancia * escalado;
    }

    if (recorrido>posicion)//elegimos dirección de retroceso
    {
        digitalWrite(DIR,HIGH);
        direccionpick = 1;
    }

    else if (recorrido<posicion)//elegimos dirección de Avance
    {
        digitalWrite(DIR,LOW);
        direccionpick = -1;
    }

    digitalWrite(ENABLE, LOW);

```

```

f=700;

while ((posicion >= recorrido) && (direccionpick == -1) || ((posicion <= recorrido) && (direccionpick == 1)))
{
    rampick();//Llamamos a la función que realiza la rampa y movimiento motor en el Pick
}

digitalWrite(ENABLE, HIGH); //Se detiene el DRIVER DRV8825
digitalWrite(STEP, LOW);
}
}

```

5.2 Montaje y programación de la herramienta triple.

Para realizar el montaje mecánico de la herramienta multi-pick triple se dispone de los siguientes elementos:

- **2 Guías lineales con motor paso a paso y encoder incremental (ver apartado 5.1).**
- **Soporte de acople TCP robot – Guías lineales – Motores brushless:**

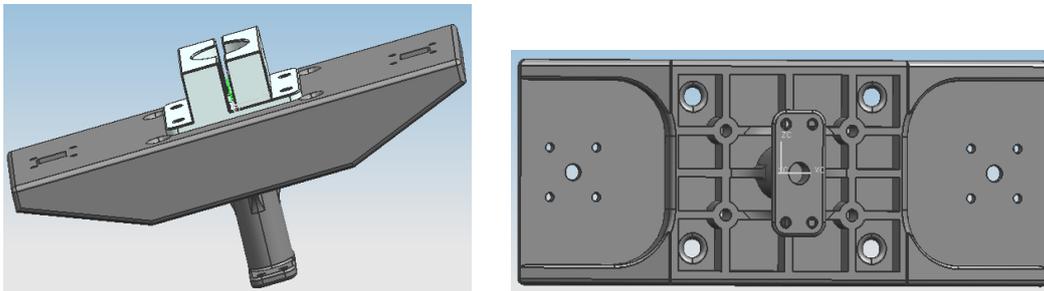


Figura 5.7. Elemento de acople del robot a las guías y a los brushless.

- **Elementos de acople ventosas – TCPs con conexiones para neumática (ver apartado 5.1).**
- **2 Finales de carrera, ventosas y pieza tope para final de carrera (ver apartado 5.1).**

- **Elementos de sujeción de los motores brushless, y potenciómetros:**

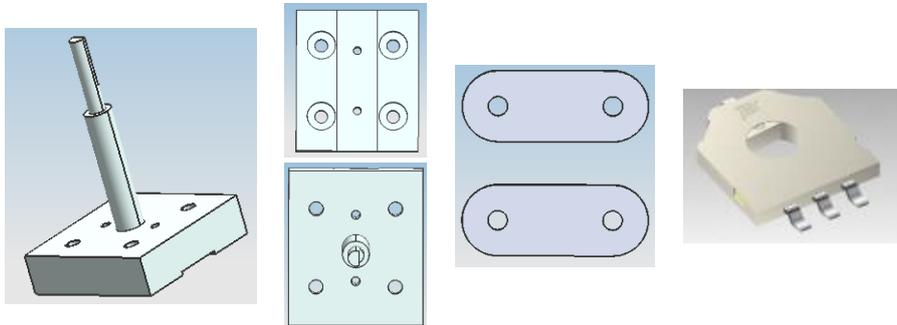


Figura 5.8. Elementos de acople a los brushless y al potenciómetro.

- **2 Motores Brushless.**



Figura 5.9. Motor brushless tipo gimbal.

Los soportes de acople del TCP robot a las guías lineal y a los motores brushless, así como los elementos de acople de las ventosas y la pieza tope para el final de carrera, se han diseñado específicamente para esta herramienta en el instituto de automática e informática industrial (AI2) y fabricado por sinterizado láser mediante una impresora 3D, excepto la pieza de fijación al motor brushless que es de aluminio. Todos ellos se han montado en conjunto para completar la herramienta, la cual se puede observar en la siguiente figura, donde se ha realizado su montaje real sobre el robot:

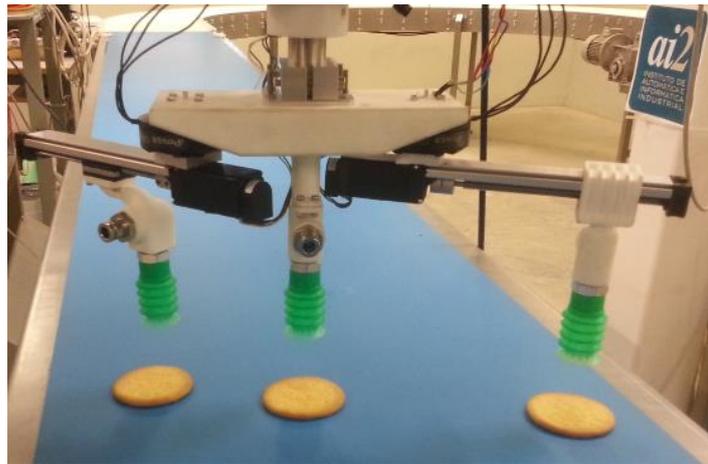


Figura 5.10. Montaje real de la herramienta multi-pick triple acoplada sobre el robot delta IRB360 ABB.

Una vez montada la herramienta y colocada sobre el robot, se deben realizar las conexiones a los diferentes dispositivos de control. El esquema de conexiones para esta herramienta será similar al representado en la figura 5.6 del apartado 5.1, salvo que en este caso se realizarán las siguientes modificaciones:

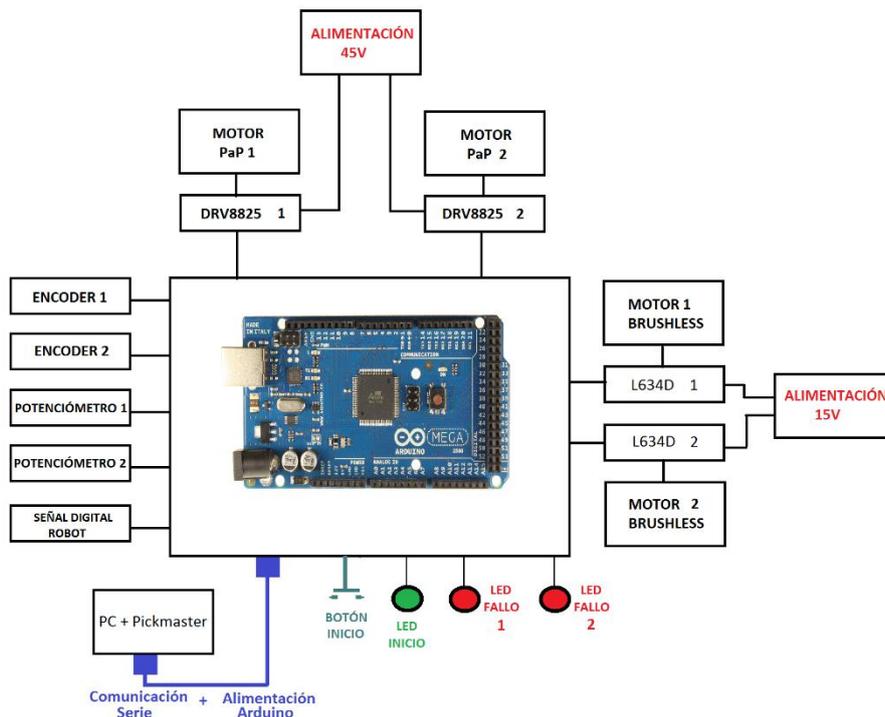


Figura 5.11. Esquema conexiones Herramienta triple.

A continuación, se muestra una tabla con las entradas y salidas utilizadas en el Arduino para cada elemento conectado.

Conexiones al Arduino Mega 2560		
Tipo de entrada/salida	Número	Dispositivo conectado
Entrada digital	27	Pulsador de inicio
Entrada digital	28	Final de carrera
Entrada digital	29	Final de carrera 2
Salida digital	13	Led indicador inicio programa
Salida digital	14	Led indicador fallo en alguna guía
Salida digital	52	Led indicador fallo en algún Brushless
Salida digital	30	Habilitación del DRV8825 (Enable)
Salida digital	32	Habilitación del DRV8825 (Enable 2)
Salida digital	31	Bit de dirección de giro DRV8825
Salida digital	24	Bit de dirección de giro DRV8825 2
Salida digital	22	Habilitación controlador Brushless 1
Salida digital	23	Habilitación controlador Brushless 2
PWM	8	PWM para el DRV8825 (Step)
PWM	9	PWM para el DRV8825 (Step 2)
PWM1	8	PWM para Brushless 1 fase A
PWM	8	PWM para Brushless 1 fase B
PWM	8	PWM para Brushless 1 fase C
PWM	8	PWM para Brushless 2 fase A
PWM	8	PWM para Brushless 2 fase B
PWM	8	PWM para Brushless 2 fase C
Interrupción externa	2	Interrupción 0, Señal encoder A
Interrupción externa	3	Interrupción 1, Señal encoder B
Interrupción externa	20	Interrupción 3, Señal encoder C
Interrupción externa	19	Interrupción 4, Señal encoder D
Interrupción externa	21	Interrupción 2, Señal digital fin de pick (IRC5)

Tabla 5.2. Conexiones al Arduino Mega.

Seguidamente, se procede a explicar el código de programación para la aplicación de control de la herramienta que se ha implementado en el entorno de desarrollo IDE de Arduino 1.0.5-r2. El código mostrado contiene las ampliaciones necesarias que se han añadido al programa de la herramienta multi-pick doble (apartado 5.1).

En primer lugar, se añaden las librerías y definiciones:

```

#define DIR_2 24 // Bit de dirección de giro
#define STEP_2 9// PWM motor paso a paso (DRV8825 2)
#define ENABLE_2 32 //Habilitación del DRV8825 2
#define FIN_CARRERA_2 29 // Final de carrera2
#define ENABLE_G1 22 //Habilitación controlador brushless 1
#define ENABLE_G2 23 //Habilitación controlador brushless 1

#define pwm1 5 // Pwm brushless fase 1
#define pwm2 11 // Pwm brushless fase 2
#define pwm3 12 // Pwm brushless fase 3

#define pwm1_2 44 // Pwm brushless fase 1
#define pwm2_2 45 // Pwm brushless fase 2
#define pwm3_2 46 // Pwm brushless fase 3

#define LED_FALL0_2 52 //Fallo de algún motor Brushless
    
```

Y se declaran también las variables globales a utilizar:

```

int frecuencia_timer=2500;
int frecuencia_timer_2=2500;
int frecuencia_timer_home=1000;

int activar_gimbal=0;
int activar_gimbal_2=0;

int pot, angulo, increment=0;
int pot_2, angulo_2, increment_2=0;

float esc_ang=2.84;
int angmin=235, angmax=800, angulo_place=490;
int error=0, error_place=0, error_ant=0, error_abs=0, cont_G1=0, cont_G2=0;

int sen_gimbal[]={32768,31624,30482,29343,28208,27078,25955,24841,23736,22642,21561,20493,
19440,18403,17384,16384,15404,14444,13507,12594,11705,10842,10005,9197,8417,7666,6946,6258,
5602,4979,4390,3836,3316,2833,2386,1976,1604,1269,973,716,498,319,180,80,20,0,20,80,180,319,
498,716,973,1269,1604,1976,2386,2833,3316,3836,4390,4979,5602,6258,6946,7666,8417,9197,10005,
10842,11705,12594,13507,14444,15404,16384,17384,18403,19440,20493,21561,22642,23736,24841,
25955,27078,28208,29343,30482,31624,32768,33912,35054,36193,37328,38458,39581,40695,41800,
42894,43975,45043,46096,47133,48152,49152,50132,51092,52029,52942,53831,54694,55531,56339,
57119,57870,58590,59278,59934,60557,61146,61700,62220,62703,63150,63560,63932,64267,64563,
64820,65038,65217,65356,65456,65516,65536,65516,65456,65356,65217,65038,64820,64563,64267,
63932,63560,63150,62703,62220,61700,61146,60557,59934,59278,58590,57870,57119,56339,55531,
54694,53831,52942,52029,51092,50132,49152,48152,47133,46096,45043,43975,42894,41800,40695,
39581,38458,37328,36193,35054,33912,32768,31624,30482,29343,28208,27078,25955,24841,23736,
22642,21561,20493,19440,18403,17384,16384,15404,14444,13507,12594,11705,10842,10005,9197,
8417,7666,6946,6258,5602,4979,4390,3836,3316,2833,2386,1976,1604,1269,973,716,498,319,180,
80,20,0,20,80,180,319,498,716,973,1269,1604,1976,2386,2833,3316,3836,4390,4979,5602,6258,
6946,7666,8417,9197,10005,10842,11705,12594,13507,14444,15404,16384,17384,18403,19440,20493,
21561,22642,23736,24841,25955,27078,28208,29343,30482,31624,32768,33912,35054,36193,37328,
38458,39581,40695,41800,42894,43975,45043,46096,47133,48152,49152,50132,51092,52029,52942,
53831,54694,55531,56339,57119,57870,58590,59278,59934,60557,61146,61700,62220,62703,63150,
63560,63932,64267,64563,64820,65038,65217,65356,65456,65516,65536,65516,65456,65356,65217,
65038,64820,64563,64267,63932,63560,63150,62703,62220,61700,61146,60557,59934,59278,58590,
57870,57119,56339,55531,54694,53831,52942,52029,51092,50132,49152,48152,47133,46096,45043,
43975,42894,41800,40695,39581,38458,37328,36193,35054,33912,32768};
// Vector con los valores de la función seno para 360 posiciones y resolución de 16 bits
    
```

```
// Posiciones iniciales en el vector de los Brushless 1 y 2;
int pos_dutyA=0;
int pos_dutyB=120;//Desfase señal Senoidal 120 grados
int pos_dutyC=240;////Desfase señal Senoidal 240 grados

int pos_dutyA_2=0;
int pos_dutyB_2=120;//Desfase señal Senoidal 120 grados
int pos_dutyC_2=240;////Desfase señal Senoidal 240 grados
byte i;
int dato[5]; //vector i contador para guardar los datos recibidos por puerto serie.
long posicion_2=0, distancia_2, C=0, D=1, recorrido_2=0, posicion_pick_2;
long posicion_ant_2=0, fallo_2=0;
int posicion_place_2=20000, cont_2=0;
int estado_2=0, direccionpick_2, direccion_2, arranquehome=1, PRINCIPAL=1;
float f_2;
```

La función **void setup ()** para este caso, estará configurada de la siguiente manera:

```
***** PREPARACION Y ARRANQUE DE PROGRAMA *****//
void setup()
{
  InitTimersSafe(); // Inicialización para poder usar PWM
  Serial.begin(115200); //Inicio de comunicación puerto serie

  pinMode(BOTON_PRINCIPAL, INPUT_PULLUP); //Declaramos pulsador como entrada

  //Declaramos led inicio y fallo como salidas y las ponemos a nivel bajo (apagados):

  pinMode(LED_INICIO,OUTPUT);
  digitalWrite(LED_INICIO,LOW);

  pinMode(LED_FALLO,OUTPUT);
  digitalWrite(LED_FALLO,LOW);

  pinMode(LED_FALLO_2,OUTPUT);
  digitalWrite(LED_FALLO_2,LOW);

  //Controladores brushless:

  pinMode(ENABLE_G1, OUTPUT);
  digitalWrite(ENABLE_G1,LOW);
  pinMode(pwm1, OUTPUT);
  pinMode(pwm2, OUTPUT);
  pinMode(pwm3, OUTPUT);
  pinMode(ENABLE_G2, OUTPUT);
  digitalWrite(ENABLE_G2,LOW);
```

```

pinMode(pwm1_2, OUTPUT);
pinMode(pwm2_2, OUTPUT);
pinMode(pwm3_2, OUTPUT);

//DRIVER MOTOR PASO A PASO
// asignamos las salidas digitales para el driver DRV8825:

pinMode(DIR, OUTPUT);
pinMode(STEP, OUTPUT);
pinMode(ENABLE, OUTPUT);
digitalWrite(ENABLE, HIGH);
pinMode(DIR_2, OUTPUT);
pinMode(STEP_2, OUTPUT);
pinMode(ENABLE_2, OUTPUT);
digitalWrite(ENABLE_2, HIGH);

//Posición inicio (home position)

pinMode(FIN_CARRERA, INPUT); // Asignamos el final de carrera a una entrada digital
digitalWrite(FIN_CARRERA, HIGH); //pullup

pinMode(FIN_CARRERA_2, INPUT); // Asignamos el final de carrera a una entrada digital
digitalWrite(FIN_CARRERA_2, HIGH); //pullup

//Interrupción interna Brushless1:

TCCR4A=0;
TCCR4B=0;
OCR4A=15625/frecuencia_timer; // La interrupción se ejecuta a la frecuencia de rotación del motor
TCCR4B |= (1<<WGM42);
TCCR4B |= (1<<CS40);
TCCR4B |= (1<<CS42);
TIMSK4=(1<<OCIE4A);

//Interrupción interna Brushless2:

TCCR3A=0;
TCCR3B=0;
OCR3A=15625/frecuencia_timer_2; // La interrupción se ejecuta a la frecuencia de rotación del motor
TCCR3B |= (1<<WGM32);
TCCR3B |= (1<<CS30);
TCCR3B |= (1<<CS32);
TIMSK3=(1<<OCIE3A);

//INTERRUPCIONES ENCODER 1

attachInterrupt(0, A_UP, RISING); //Interrupción señal A, activado a flanco de subida
attachInterrupt(1, B_UP, RISING); //Interrupción señal B, activado a flanco de subida

//INTERRUPCIONES ENCODER 2

attachInterrupt(3, C_UP, RISING); //Interrupción señal C, activado a flanco de subida
attachInterrupt(4, D_UP, RISING); //Interrupción señal D, activado a flanco de subida

//INTERRUPCIONES ROBOT

pinMode(21, INPUT_PULLUP);
attachInterrupt(2, PLACE, FALLING); // Se activa la interrupción por flanco de bajada.
}

```

A continuación se muestran las diferentes funciones que se han implementado y se utilizarán en programa principal.

- **Función de posición inicial (Home position):**

En este caso la función Homeposition, deberá situar cada módulo en su posición de inicio, esto es, los TCP móviles en el extremo de la guía y los motores brushless deben situarse en la posición angular de 0°. Se observa como la posición angular 0 corresponde al valor 235 del potenciómetro incorporado en cada motor (Variable global *angmin*). El posicionamiento de cada uno de los dispositivos se realizará de forma secuencial.

En el caso de los motores PaP, se ha establecido una frecuencia de 1000Hz, y para los motores brushless, se ha configurado el *Timer* también a 1000Hz=1ms.

```
//-----POSICION HOME INICIO-----//

void homeposition()
{
  while(estado==HIGH)//Estado del final de carrera
  {
    digitalWrite(ENABLE,LOW); // Habilitación del driver
    digitalWrite(DIR,LOW); // Dirección de giro (retroceso)
    if(fh<=fhmax) // Rampa de aceleración para arranque del motor
    {
      SetPinFrequency(STEP,fh);
      pwmWriteHR(STEP, 32768);
      delayMicroseconds(50);
      fh=fh+1;
    }
    else
    {
      SetPinFrequency(STEP,1000); //Velocidad motor 1000Hz
      pwmWriteHR(STEP, 32768);
    }
  }

  digitalWrite(STEP, LOW);
  digitalWrite(ENABLE, HIGH); // Se deshabilita el driver
  posicion=0; // Inicialización de posición
  fh=0; //Reiniciamos el valor del contador de la rampa home, para el motor2.
  while(estado_2==HIGH)//Estado del final de carrera (guía 2)
  {
    digitalWrite(ENABLE_2,LOW); // Habilitación del driver
    digitalWrite(DIR_2,LOW); // Dirección de giro (retroceso)
    if(fh<=fhmax) // Rampa de aceleración para arranque del motor
    {
      SetPinFrequency(STEP_2,fh);
      pwmWriteHR(STEP_2, 32768);
      delayMicroseconds(50);
      fh=fh+1;
    }
    else
    {
      SetPinFrequency(STEP_2,1000); //Velocidad motor 1000Hz
      pwmWriteHR(STEP_2, 32768);
    }
  }

  digitalWrite(STEP_2, LOW);
  digitalWrite(ENABLE_2, HIGH); // Se deshabilita el driver
  posicion_2=0; // Inicialización de posición

  pot = analogRead(0); //Leemos sensor de posición del motor gimbal (potenciómetro)
  OCR4A=15625/frecuencia_timer_home;//establecemos la velocidad del motor para el home

  while((pot>=angmin)|| (pot<=angmin)) //Mientras no lleguemos a la posición de inicio...
  {
```

```

    if (pot<angmin)
    {
        pot = analogRead(0); //Vamos leyendo el sensor
        increment=1; //Elegimos la dirección de giro
        activar_gimbal=1; // Activamos la ejecución de la onda senoidal
        digitalWrite(ENABLE_G1,HIGH); //activamos el controlador brushless
    }
    else if (pot>angmin)
    {
        pot = analogRead(0);
        increment=-1; //Elegimos la dirección de giro
        activar_gimbal=1; // Activamos la ejecución de la onda senoidal
        digitalWrite(ENABLE_G1,HIGH); //activamos el controlador brushless
    }

    pot=analogRead(0); //verificamos si se ha producido un error
}

activar_gimbal=0; //Deshabilitamos interrupción
digitalWrite(ENABLE_G1,LOW); //Deshabilitamos driver

pot_2 = analogRead(1); //Leemos sensor de posición del motor gimbal 2 (potenciómetro 2)
OCR3A=15625/frecuencia_timer_home; //establecemos la velocidad del motor para el home
while((pot_2>=angmin)|| (pot_2<=angmin)) //Mientras no lleguemos a la posición de inicio...
{
    if (pot_2<angmin)
    {
        pot_2 = analogRead(1); //Vamos leyendo el sensor
        increment_2=1; //Elegimos la dirección de giro
        activar_gimbal_2=1; // Activamos la ejecución de la onda senoidal
        digitalWrite(ENABLE_G2,HIGH); //activamos el controlador brushless
    }
    else if (pot_2>angmin)
    {
        pot_2 = analogRead(1);
        increment_2=-1; //Elegimos la dirección de giro
        activar_gimbal_2=1; // Activamos la ejecución de la onda senoidal
        digitalWrite(ENABLE_G2,HIGH); //activamos el controlador brushless
    }

    pot_2=analogRead(1); // verificamos si se ha producido un error
}

activar_gimbal_2=0; //Deshabilitamos interrupción
digitalWrite(ENABLE_G2,LOW); //Deshabilitamos driver

arranquehome=0; //Cambio el valor de variable para que no se repita el home position
}

```

- **Funciones lectura de posición encoder:**

Las funciones de lectura de posición serán idénticas a las de la herramienta doble, con la particularidad de que se incorporarán 4 funciones más para el encoder de la segunda guía lineal. Estas funciones se las ha llamado *C_UP()*, *C_DOWN()*, *D_UP()*, y *D_DOWN()*.

- **Movimiento *pick* con rampa de velocidad:**

Su funcionamiento será el mismo que para la herramienta doble, añadiendo otra función de movimiento *pick* para la guía lineal 2 a la que se le ha llamado *rampick_2()* .

- **Movimiento *place* con rampa de velocidad:**

El funcionamiento es exactamente igual que la función de movimiento *pick* con rampa de velocidad, solo que se para este caso, en las condiciones se incluye la posición de dejada en lugar de la posición de agarre del producto.

- **Interrupción *place*:**

Como se ha explicado en el apartado 5.1, esta función se activa cuando se produce la interrupción de la señal digital que envía el controlador del robot cuando se produce el agarre del producto permitiendo así que se ejecute la operación *place* en el programa principal.

```

//-----INTERRUPCION PLACE-----//
void PLACE()
{
  int2=1;
}
  
```

- **Funciones de los *timers* 3 y 4:**

Estas funciones se ejecutarán una vez se inicialice el programa y, de forma cíclica se irán ejecutando a la frecuencia configurada. Para evitar que mientras no se estén utilizando los motores brushless vayan cambiando las posiciones del vector, se ha incorporado una variable que se activará o desactivará dependiendo de cuando se requiera el uso del vector para hacer funcionar el motor correspondiente.

```
//-----INTERRUPCION INTERNA GIMBAL Brushless 1-----
ISR(TIMER4_COMPA_vect){

if (activar_gimbal==1)

{

pos_dutyA = pos_dutyA + increment;
if( pos_dutyA > 360) pos_dutyA = 0;
if( pos_dutyA<0) pos_dutyA =360;

pos_dutyB = pos_dutyB + increment;
if( pos_dutyB > 360) pos_dutyB = 0;
if( pos_dutyB<0) pos_dutyB =360;

pos_dutyC = pos_dutyC + increment;
if( pos_dutyC > 360) pos_dutyC = 0;
if( pos_dutyC<0) pos_dutyC =360;

SetPinFrequency(pwm1,100000);
pwmWriteHR(pwm1, sen_gimbal[pos_dutyA]);
SetPinFrequency(pwm2,100000);
pwmWriteHR(pwm2, sen_gimbal[pos_dutyB]);
SetPinFrequency(pwm3,100000);
pwmWriteHR(pwm3, sen_gimbal[pos_dutyC]);

}

//-----INTERRUPCION INTERNA GIMBAL Brushless 2-----
ISR(TIMER3_COMPA_vect){

if (activar_gimbal_2==1)

{

pos_dutyA_2 = pos_dutyA_2 + increment_2;
if( pos_dutyA_2 > 360) pos_dutyA_2 = 0;
if( pos_dutyA_2<0) pos_dutyA_2 =360;

pos_dutyB_2 = pos_dutyB_2 + increment_2;
if( pos_dutyB_2 > 360) pos_dutyB_2 = 0;
if( pos_dutyB_2<0) pos_dutyB_2 =360;

pos_dutyC_2 = pos_dutyC_2 + increment_2;
if( pos_dutyC_2 > 360) pos_dutyC_2 = 0;
if( pos_dutyC_2<0) pos_dutyC_2 =360;

SetPinFrequency(pwm1,100000);
pwmWriteHR(pwm1_2, sen_gimbal[pos_dutyA_2]);
SetPinFrequency(pwm2,100000);
pwmWriteHR(pwm2_2, sen_gimbal[pos_dutyB_2]);
SetPinFrequency(pwm3,100000);
pwmWriteHR(pwm3_2, sen_gimbal[pos_dutyC_2]);

}

}
```

- **Funciones de control de los motores brushless:**

Cuando se requiera realizar el movimiento de los motores brushless, se hará uso de estas funciones que serán las encargadas de realizar el control de velocidad de los mismos. En un primer paso, se comprobará que el motor brushless no se ha quedado enclavado ni averiado mediante la comparación con el error de posición obtenido en el programa principal. Si se produce un fallo tres veces seguidas, se procederá a desconectar el motor y encender el led de fallo de motores. De no ser así, se aplicará el control de velocidad. Dicho control comprueba el

valor del error absoluto y dependiendo de este, se cambiará la frecuencia del *timer* para que a su vez, se cambie la velocidad de rotación del motor. Se ha establecido una constante de proporcionalidad (de valor 125) que realizará un control proporcional al error absoluto obtenido. Además, se ha incorporado una saturación para que no se sobrepase la frecuencia máxima de rotación del motor (obtenida experimentalmente) así como una frecuencia mínima para evitar retrasos en la secuencia de control:

```

//-----MOVIMIENTO PICK Y PLACE GIMBAL_1-----//

void gimbal_1()
{
  if (cont_G1>=3)
  {
    digitalWrite(LED_FALLO_2,HIGH);//Si se produce un fallo se enciende el led;
    activar_gimbal=0;
    digitalWrite(ENABLE_G1,LOW);
  }

  else if (error_ant==error)
  {
    cont_G1++;
  }

  else
  {
    if(error_abs>5)
    {
      frecuencia_timer=2500;
      OCR4A=15625/frecuencia_timer;
    }
    else
    {
      frecuencia_timer=125*error_abs;
      if (frecuencia_timer>2500)
      {
        frecuencia_timer=2500;
      }
      else if (frecuencia_timer<500)
      {
        frecuencia_timer=500;
      }
      OCR4A=15625/frecuencia_timer;
    }
  }
  activar_gimbal=1;
  digitalWrite(ENABLE_G1,HIGH);
  error_ant=error;
  cont_G1=0;
}

//-----MOVIMIENTO PICK Y PLACE GIMBAL_2-----//

void gimbal_2()
{
  if (cont_G2>=3)
  {
    digitalWrite(LED_FALLO_2,HIGH);//Si se produce un fallo se enciende el led;
    activar_gimbal_2=0;
    digitalWrite(ENABLE_G2,LOW);
  }

  else if (error_ant==error)

```

```

    cont_G2++;
  }

  else
  {
    if(error_abs>10)
    {
      frecuencia_timer_2=2500;
      OCR3A=15625/frecuencia_timer_2;
    }
    else
    {
      frecuencia_timer_2=125*error_abs;
      if (frecuencia_timer_2>2500)
      {
        frecuencia_timer_2=2500;
      }
      else if (frecuencia_timer_2<500)
      {
        frecuencia_timer_2=500;
      }
      OCR3A=15625/frecuencia_timer_2;
    }
  }
  activar_gimbal_2=1;
  digitalWrite(ENABLE_G2,HIGH);
  error_ant=error;
  cont_G1=0;
}

```

Explicadas todas las funciones, se muestra a continuación el programa principal de la herramienta.

Este código se implementa dentro de la función **loop()** del entorno de Arduino. Todo lo que esté implementado dentro de esta función se ejecutará de manera secuencial y de forma cíclica y no se detendrá hasta que no se le dé una orden, o al desconectar el microcontrolador de la alimentación.

Al iniciarse el programa, éste esperará a que el pulsador de inicio sea pulsado. De ser así, la primera función que se ejecutará será la de posicionamiento inicial (homeposition).

Seguidamente se procederá a realizar la manipulación de objetos mediante las etapas explicadas en el apartado 4.2.

Una condición *while* determinará si la operación a realizar es la de *pick* o la de *place*. Si el puerto serie está disponible porque se ha recibido algún dato, se realizará la lectura de coordenadas y seguidamente la función *pick*. Si por el contrario no se ha recibido ningún dato, entonces se realizará la función *place* cuando esta haya sido requerida por la interrupción.

Cuando se reciban las coordenadas, se realizará el posicionado lineal de las guías 1 y 2, y seguidamente el posicionado angular de las mismas mediante el giro de los motores brushless. Una vez se active la interrupción que habilita la operación *place*, el proceso será a la inversa, es decir, primero se posicionarán los motores brushless 1 y 2, y seguidamente los TCPs de las guías.

6. VERIFICACIÓN Y RESULTADOS OBTENIDOS

6.1 Resultados en el posicionamiento del motor PaP

Como se ha explicado en el apartado 4.1.3.2, se requiere generar una rampa de velocidad para alcanzar así la máxima velocidad del motor paso a paso sin tener problemas en el arranque. Esta velocidad corresponde a 500rpm con una frecuencia máxima de 3KHz. Para ello, se han creado en el código de programación 3 límites de frecuencia en los que para cada uno de ellos se irá incrementando unos valores dependiendo de la condición en la que se encuentren.

De manera experimental y con la ayuda de la función *millis()* de la herramienta de desarrollo IDE de Arduino, la cual permite conocer el tiempo de ejecución desde que se arranca el programa, se han realizado pruebas en las que se ha verificado el tiempo que tarda en recorrer el TCP móvil desde un extremo de la guía hasta el otro, determinando así la rampa de velocidad que más se ajusta a las necesidades.

La frecuencia inicial del motor será de 700Hz. Se dispone de 3 límites de frecuencias Fmax1, Fmax2 y Fmax3 en las cuales se irá aumentando la frecuencia con unos incrementos establecidos. Estos incrementos serán los que determinarán la rampa de velocidad.

Se tiene pues:

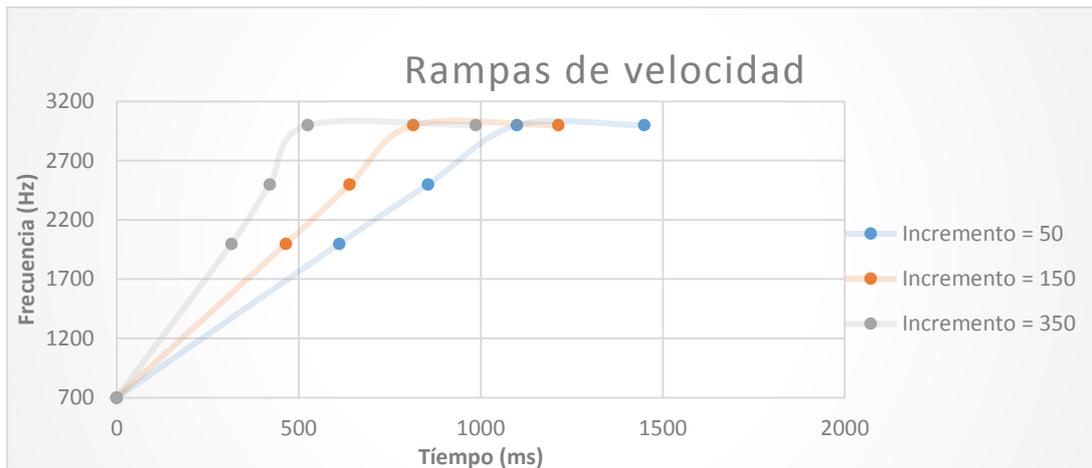
Final= 700 Hz, Fmax1= 1999 Hz, Fmax2=2499 Hz y Fmax3=2999 Hz.

Los resultados obtenidos para los incrementos de 50, 150, 350 han sido los siguientes:

	Frecuencia	Incremento = 50	Incremento = 150	Incremento = 350
F inicial	700 Hz	0	0	0
Fmax1	1999 Hz	611 ms	465 ms	315 ms
Fmax2	2499 Hz	855 ms	640 ms	420 ms
Fmax3	2999 Hz	1099 ms	815 ms	525 ms
F máxima	3000 Hz	1449 ms	1213 ms	986 ms

Tabla 6.1. Resultados experimentales motor paso a paso.

Después de realizar numerosas pruebas con incrementos diferentes, se ha decidido mostrar los 3 experimentos mostrados en la tabla 6.1. De los cuales se ha podido verificar que mediante un incremento de la frecuencia de 350 Hz en cada iteración de los bucles de condición, el motor es capaz de posicionar el TCP móvil de la guía en el peor caso (100mm de recorrido) en alrededor de 1 segundo. En la siguiente gráfica se observa la rampa de velocidad lineal (desde 700Hz hasta 2999 Hz) de cada una de las pruebas realizadas.



Gráfica 6.1. Rampas de velocidad lineal para distintos incrementos de la frecuencia.

Una vez determinada la rampa de velocidad, se ha procedido a comprobar la precisión en el posicionamiento del TCP móvil. Para ello se han establecido 3 posiciones diferentes y se han ejecutado las órdenes de posicionado 10 veces seguidas, obteniendo los siguientes resultados:

Nº de repeticiones	Recorrido (mm)		
	33mm	66mm	99mm
1	33	65	100
2	33	65	98
3	32	66	98
4	33	66	99
5	33	66	99
6	32	66	99
7	32	66	99
8	34	66	99
9	34	65	99
10	32	67	99
Media:	32.8	65.8	98.9
Desviación:	0.748	0.6	0.538

Tabla 6.2. Precisión del motor paso a paso.

Cabe decir que para hallar la precisión así como repetibilidad efectiva de la guía lineal se debería realizar un test de fatiga para observar el comportamiento, por ejemplo, mediante una jornada entera de trabajo.

6.2 Resultados en el posicionamiento del motor Brushless

Tal y como se ha explicado en el apartado 4.2.3.1, se debe buscar la manera de hacer girar el motor de manera suave para conseguir una precisión de 1° y poder obtener así una relación entre cada una de las posiciones del vector que crea la señal sinusoidal y el ciclo de trabajo de cada señal PWM que se le aplica al motor en cada instante de tiempo.

De manera experimental, se ha introducido un contador en el código que realiza el barrido de las posiciones del vector en cada PWM, de tal manera que se pueda obtener el número de posiciones que se recorren cuando se realiza un giro completo de 360° . De esta forma, se puede predecir cuantas posiciones se deben recorrer para obtener una posición de 1° .

Aplicando un valor al *Timer* que ejecuta la interrupción interna de 1ms, se ha obtenido el número de posiciones que se recorren para realizar un giro completo, el resultado se ha comparado con el valor del potenciómetro para verificar la posición, obteniendo el siguiente resultado:

Frecuencia del Timer (interrupción interna) = 1000Hz = 1ms.

Ángulo recorrido	Nº de posiciones recorridas
360°	1265
180°	633
90°	316
45°	158
10°	35
1°	3

Tabla 6.3. Resultados experimentales de la relación del ángulo recorrido con las posiciones del motor.

Una vez se determina la suavidad de giro del motor y la relación entre las posiciones del vector a recorrer con el ángulo de rotación del motor, se ha procedido a determinar la frecuencia mínima y máxima con la que se puede configurar el *Timer* que activa la interrupción sin que afecte a la rotación del motor.

Mediante la realización de pruebas en el laboratorio, se han obtenido las frecuencias máximas y mínimas para el *Timer*. Además, mediante la función *millis()* en Arduino, se han obtenido los tiempos que se tarda en alcanzar una posición concreta. Los resultados se pueden observar en la siguiente tabla.

	Tiempo (s)			
	360°	180°	90°	45°
F mínima = 200Hz	6.32	3.16	1.58	0.79
F = 500Hz	2.53	1.27	0.63	0.32
F = 1000Hz	1.26	0.63	0.31	0.16
F = 1500Hz	0.84	0.42	0.21	0.10
F = 2000 Hz	0.63	0.31	0.16	0.08
F = 2500 Hz	0.50	0.25	0.12	0.06

Tabla 6.4. Tiempos obtenidos. (Frecuencia vs posición angular)

A partir de los resultados anteriores (tabla 6.4), se ha determinado que para la implementación del control del motor brushless es suficiente aplicar frecuencias comprendidas entre 500 y 2,5 KHz en el *Timer* de la interrupción externa, obteniendo así tiempos de operación considerablemente buenos para el posicionado de los TCP móviles de la herramienta.

Para el control de velocidad, se han establecido los límites máximo y mínimo de frecuencia, de tal manera que para un error de posición mayor a 10° se aplique la máxima frecuencia al timer (2.5KHz), y para un error menor a 10° se aplique una regulador proporcional.

De manera experimental se han probado diferentes constantes de proporcionalidad, estableciendo finalmente un valor de $K_p = 125$. Los resultados obtenidos con esta constante de proporcionalidad aplicada a un error de posición establecido han sido los siguientes:

	Tiempo (ms)						
	0 - 180°	0 - 90°	0 - 45°	0 - 20°	0 - 10°	0 - 5°	0 - 3°
Control de velocidad ($K_p = 125$)	330	193	88	35	18	34	26

Tabla 6.4. Tiempos obtenidos. (Frecuencia vs posición angular)

Con estos resultados se verifican unos tiempos adecuados para la operación de posicionado, ya que en el caso más desfavorable (en el que la guía debe desplazarse mediante un giro de 180°), el dispositivo se posicionará en 330ms. Tiempo más que suficiente para realizar la operación completa.

6.3 Resultados de la herramienta doble.

Una vez montada la Herramienta sobre el robot, se ha realizado una prueba en la que se establece una velocidad de trabajo de 600mm/s en el robot Flexpicker de ABB.

Se ha lanzado una operación pick and place en la que se han recogido 4 galletas de geometría redonda en dos secuencias. A la primera secuencia le corresponde una coordenada de 30mm de recorrido y a la segunda una coordenada de 76mm de recorrido, la operación place se ha configurado para que se dejen las 4 galletas apiladas de dos en dos a una distancia de 50mm entre ellas. La operación pick and place completa se ha realizado en un total de 5 segundos y los resultados han sido satisfactorios.

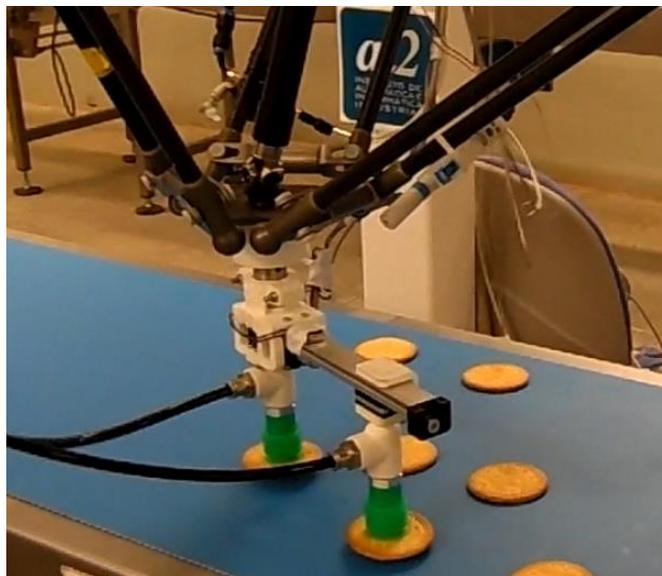


Tabla 6.1. Resultados experimentales con la herramienta doble.

7. CONCLUSIONES Y TRABAJOS FUTUROS

7.1 Conclusiones.

En el transcurso de este proyecto se han diseñado, y construido dos prototipos de herramientas multi-pick para la manipulación de productos con geometría circular, capaces de funcionar bajo condiciones de velocidad moderada y realizando la manipulación pick and place de dos o tres productos simultáneamente, logrando así aumentar la productividad del proceso.

Mediante la programación y el control, se ha conseguido obtener una herramienta dinámica, capaz de mover sus efectores finales a cualquier posición dentro de su área de trabajo, optimizando al máximo el tiempo de respuesta de éstos. Además, las herramientas aportan flexibilidad ante posibles cambios en la logística de la producción.

Los dispositivos de posicionamiento y actuadores utilizados han sido capaces de resolver las especificaciones propuestas para las herramientas, consiguiendo así obtener un sistema de manipulación económico y con bajo mantenimiento. Por otra parte, al utilizar un microcontrolador económico, ha permitido disminuir los costes de la herramienta rompiendo así las barreras económicas que supone incorporar este tipo de sistemas en la industria.

Este desarrollo abre nuevas puertas hacia la creación de sistemas de automatización inteligentes robustos y fiables.

7.2 Trabajos futuros.

Con los resultados obtenidos, así como con la experiencia que se ha conseguido al elaborar los prototipos de ambas herramientas multi-pick, se han determinado una serie de futuros trabajos que aportarían mejoras y robustez a las herramientas desarrolladas. Seguidamente se numeran las propuestas que se han determinado a corto plazo.

7.2.1 Mejora de la programación de la aplicación de control.

Uno de los inconvenientes de utilizar el microcontrolador ATmega con Arduino, es que las operaciones de control se deben realizar de forma secuencial. Esto puede llevar a grandes problemas cuando se desea incorporar más módulos en la herramienta multi-pick, ya que se necesita mayor velocidad, así como un control de tiempos de las actuaciones de los diferentes dispositivos de posicionamiento.

Una solución a estos problemas podría ser la utilización de microcontroladores preparados para funcionar mediante herramientas de desarrollo o sistemas operativos en tiempo real.

De esta manera, implementando un código de control mediante el uso de hilos en la programación, se podrían ejecutar de forma paralela las tareas de posicionado de los dispositivos, consiguiendo así mejores tiempos de reacción, así como una herramienta multi-pick robusta y efectiva, ya que mediante la planificación de tareas se podría adaptar a situaciones en los que se requieren cambios como por ejemplo, el robot.

Un ejemplo sería el uso de microcontroladores industriales Cortex M3 con núcleo ARM de 32 bits. Estos microcontroladores son más potentes y tienen una capacidad de cálculo superior que los microcontroladores convencionales. Además se pueden programar mediante herramientas de desarrollo de código abierto y en tiempo real, como FreeRtos.



Figura 7.1. Microcontrolador Cortex de 32 bits y Sistema operativo FreeRtos.

7.2.2 Ampliación del área efectiva de trabajo mediante detección de colisiones.

Otro de los trabajos futuros que se puede desempeñar consiste en ampliar el área de trabajo efectiva cuando se dispone de mayor número de módulos. Sobre todo, cuando se utiliza coordenadas polares para la manipulación de los objetos (ver figura 7.2).

El problema a resolver parte de que al ampliar el área de trabajo o querer disminuir lo máximo posible los espacios vacíos dentro del área de agarre de productos, se obtienen zonas conflictivas en las que pueden producirse colisiones entre los distintos módulos.

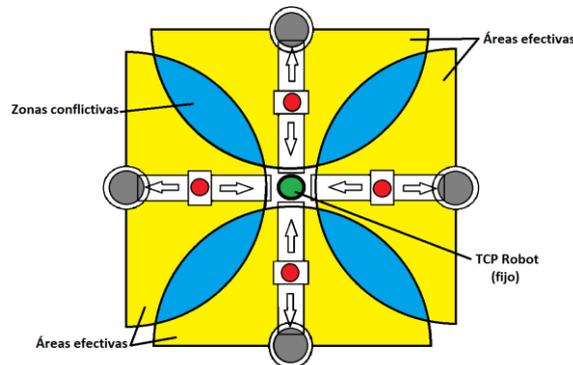


Figura 7.2. Herramienta de cuatro módulos.

La solución pasaría por la implementación de un nuevo código de control en el que se comparta información de coordenadas y trayectorias entre todos los módulos, de tal manera que se puedan organizar y orientar cada dispositivo de posicionamiento de manera ordenada (por ejemplo, controlando los tiempos de posicionado de cada módulo).

Otra solución efectiva pero al vez más cara, sería incluir sensores entre los módulos para tener información acerca de ellos, y actuar en consecuencia. Así, dependiendo de la posición en la que se encuentre un módulo en cada instante, se pueden calcular las trayectorias a realizar.

7.2.3 Incorporación de una interfaz gráfica para el control de la herramienta.

Cuando se tienen numerosos dispositivos de control, así como actuadores de posicionamiento, como es el caso de una herramienta multi-pick, es posible que se produzcan fallos en el transcurso de las operaciones de manipulación. Para tener un control de los fallos que puedan ocurrir, se podría realizar una interfaz gráfica o SCADA en la que se informara de los fallos producidos, así como del transcurso de posicionado de cada dispositivo (Distancias, posiciones angulares, etc.)

8. REFERENCIAS

- [1] G. Seliger, *Robotics in operations*, SHIMON Y. NOF: Handbook of industrial robotics, 2ª Edición, 1999, pág 1037-1045.
- [2] M. Mellado, C. Blanes, A. Valera, C. Ortiz, “Technologies for robot grippers in pick and place operations for fresh fruits and vegetables”. Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria (INIA). Spanish Journal of Agricultural Research 2011.
- [3] C. Blanes, “Técnicas de manipulación hortofrutícola robotizada en procesos pick & place”. Tesina de Máster de ingeniería mecánica y materiales. Universidad Politécnica de Valencia, 2009.
- [4] A. Wolf, R. Steinmann, H. Schunk, “Grippers in motion”. Editorial Springer, 2005. ISBN 3-540-25657-1.
- [5] R. Rodríguez, A. Lasa, J. Azpiazu, I. Martínez de Marañón. Repotaje técnico “*Robótica en la industria alimentaria*”. AZTI-Tecnalia, FATRONIK-Tecnalia, 2008.
- [6] R. Marino, S. Peresada, P. Tomei , “*Nonlinear adaptive control of permanent magnet step motors*”. Automatica (Journal of IFAC), Volume 31, pages 1595-1604. Noviembre 1995.
- [7] Chung Yuan, Chung Li, “*Adaptive fuzzy sliding-mode control for PM synchronous servo motor drives*”. Control Theory and applications, IEE Proceedings. Volume 145. Pages 63-72. Enero 1998.
- [8] Roger Juanpere Tolrà, “*Técnicas de control para motores Brushless*”. ingenia-cat – Motion Control Department. Barcelona.
- [9] S. Navarro Pérez, “*Desarrollo de una aplicación de visión por computador para multipick robotizado en aplicaciones de empaquetado*”. Proyecto final de carrera. Ingeniería Informática. Universidad Politécnica de Valencia. Julio 2014.

Empresas con proyectos de herramientas multi-pick:

- Garra *Multi-pick handling unit*, Tecnopack (fecha: Julio 2014): <http://www.tecnopackspa.it/>
- *Gripper FXP*, Schmalz (fecha: Julio 2014) : <http://us.schmalz.com/>
- Garra multi-pick *Robotic EOAT*, SAS Automation (fecha: Agosto 2014): <http://www.sas-automation.com/en/>
- Garra *Palletiser for small boxes*, Roteg (fecha: Julio 2014): <http://www.roteg.de/en/robot/gripping-systems/multi-pick/>
- Robots de paletización y embalaje *RP4 y RCP*, Apsol (fecha: Agosto 2014): <http://www.apsol.it/>
- *Multipick Cardboard gripper*, Cimcorp (fecha: Agosto 2014): <http://www.cimcorp.com/Welcome>
- Garras *Robocollect, Salami picker y SG47*, Robomotion GmbH (fecha: Agosto 2014): <http://www.robomotion.de/>
- *Robo-pick*, Mitsubishi Automation (fecha: Junio 2014): https://www.meau.com/eprise/main/sites/public/About_Us/-Home
- *Multigripper System*, Metzner (fecha: Julio 2014): <http://www.metzner.com/>

Empresas de dispositivos de control de movimiento:

- <http://www.newmarksystems.com/>
- https://www.smc.eu/portal_ssl/WebContent/main/index.jsp?is_main=yes
- <http://www.a-drive.de/>
- <http://www.festo.com/net/startpage/>
- <http://www.igus.es/>

- <http://www.gimatic.com/>
- <http://www.intelligentactuator.com/>

Información en la web:

- Ministerio de agricultura, alimentación y medio ambiente: <http://www.magrama.gob.es/es/alimentacion/temas/industria-agroalimentaria/marco-estrategico/default.aspx>
- Futuro prometedor para la robótica en la industria alimentaria: <http://www.alimentatec.com/futuro-prometedor-para-la-automatizacion-y-la-robotica-en-la-industria-alimentaria/?print=pdf>
- Información sobre el microcontrolador Arduino: <http://arduino.cc/>
- Driver de control de motor paso a paso: <http://www.pololu.com/product/2132>
- Motores gimbal brushless: <http://www.olliw.eu/2013/brushless-gimbal-direct-drive/>
- Robot IRB 360 Flexpicker de ABB: <http://new.abb.com/products/robotics/industrial-robots/irb-360>
- Pickmaster ABB: <http://www.abb.com/product/seitp327/6d52bccc823a4833c1256fdc00282a12.aspx>