

# **CONTROL DE MOTORES POR MEDIO DE LA PLATAFORMA AER-ROBOT**

---

TRABAJO FIN DE PERIODO FORMATIVO DE UN PROGRAMA DE  
DOCTORADO (TFPFD)



Alejandra Armenta Valadez  
DIRECTOR: José Luis Muñoz Lozano



## Contenido

Capítulo 1: Introducción y objetivos .....	5
Introducción .....	5
Motivación .....	7
Objetivos .....	8
Estructura del trabajo .....	9
Capítulo 2: Tareas de Alcance y agarre para un robot de servicios .....	11
De la Robótica Industrial a la de Servicios.....	11
Alcance y Agarre.....	15
Alcance y Agarre en la Robótica de Servicios.....	16
Características de los Sistemas de Alcance y Agarre.....	18
Alcance .....	18
Agarre.....	20
Conclusiones .....	24
Capítulo 3: Fusión Multisensorial.....	26
Introducción .....	26
ARQUITECTURAS .....	29
ALGORITMOS.....	32
Conclusiones .....	36
Capítulo 4: Sistemas Bioinspirados .....	37
Comportamientos Reactivos.....	41
Arquitecturas de Control Reactivas.....	42
Principales Características.....	43
Enfoque basado en comportamientos.....	43
Características principales [20] [24]:.....	44
Inteligencia Artificial: Redes Neuronales .....	45

Redes Neuronales .....	45
NN Usadas en tareas de Alcance y Agarre .....	49
El paradigma AER (Address–EventRepresentation) .....	53
Introducción .....	53
Sistema AER.....	54
BUS AER.....	55
Estado del arte .....	55
Bloques de Control.....	56
Conclusiones .....	60
Capítulo 5 Implementacion practica: control de motores.....	62
La Plataforma AER-Robot.....	62
Componentes principales.....	63
Etapa de Control.....	63
FPGA .....	63
Microcontrolador .....	64
Etapa de Potencia.....	66
Puente H.....	66
Buffer.....	66
Control del Cabezal Lince mediante la Plataforma AER-Robot.....	67
Cabezal Lince .....	67
Conexiones .....	71
Recursos .....	72
VM-WARE.....	72
Xilinx .....	75
Matlab .....	80
Capítulo 6: Conclusiones y trabajos futuros.....	82
Anexo A: Especificaciones técnicas de la Plataforma AER-Robot .....	84
Anexo B: Funciones en Matlab.....	85

Init: .....	85
Bibliografía .....	88

## Capítulo 1: Introducción y objetivos

### Introducción

En los últimos años el desarrollo tecnológico ha sido tan abrumador, que no era extraño pensar que en unos años la ciencia ficción se vería superada por la realidad. La palabra robot fue usada por primera vez en el año 1921, cuando el escritor checo Karen Capek (1890-1938) estrena en el teatro nacional de Praga su obra Rossum's Universal Robot (R.U.R.) [1]. Su origen proviene de la palabra eslava robota, que se refiere al trabajo de forma forzada. El término hubiera quedado en desuso si no hubiera sido por los escritores del género literario de la ciencia ficción, el más reconocido como impulsor de la palabra robot es Isaac Asimov.

Pero antes que los robots dejaran de figurar solo en historias de ciencia ficción y fuesen una realidad en la industria; en 1948, R. C. Goertz desarrolló en el Argonne National Laboratory el primer manipulador, con el objetivo de maniobrar elementos radiactivos sin riesgo para el operador. El origen del concepto de robot fue la sustitución del operador por un programa de ordenador, el cual controlase los movimientos del manipulador.

En 1954 fue solicitada la primera patente de un dispositivo robótico por el británico C.E.Kenward. Sin embargo, fue George C. Devol, un ingeniero norteamericano, el que estableció las bases del robot industrial moderno. En 1956, Joseph F. Engelberger y Devol comenzaron a trabajar en la utilización industrial de sus máquinas, fundando la Consolidated Controls Corporation. Más tarde se convertiría en Unimation (Universal Automation) desarrollando los primeros proyectos de robots, e instalando su primera máquina Unimate en la General Motors de Trenton, Nueva Jersey, en 1960.

Para el año 1970, los laboratorios de las universidades de Stanford y del MIT se proponen controlar un robot mediante un computador. Sin embargo, el crecimiento de la robótica en Japón aventajó a los Estados Unidos gracias a Nissan, que formó la primera asociación robótica del mundo, la Asociación de Robótica Industrial de Japón (JIRA) en 1972. Mientras tanto en Europa, que tuvo un despertar más tardío, hasta 1973, no se construyó el primer robot con accionamiento totalmente eléctrico, a cargo de la firma sueca ASEA.

El Instituto de Robótica de América (RIA), se formó en 1974 (en 1984 cambió su nombre por el de Asociación de Industrias Robóticas, manteniendo las mismas siglas), aunque no fue hasta 1975, con la aparición del microprocesador, que la robótica volvería a tener un progreso relevante: se logró transformar la imagen y las características del robot, que hasta entonces eran grandes y toscos. En la década de los setenta, la robótica contaba ya con numerosos prototipos de elementos prostéticos y ortéticos (brazos, manos o piernas) de excelente mecánica, aunque con fuertes limitaciones en el control.

A partir de 1980, gracias al impulso sobre la informática aplicada y la experimentación de los sensores, se potencia la configuración del robot inteligente para adaptarse al ambiente y tomar decisiones en tiempo real y adecuarlas para cada situación. En este mismo año se fundó la Federación Internacional de Robótica. [2] La robótica en general y en especial en los sectores de aplicación no estrictamente industrial, ha experimentado progresos relevantes y ha ampliado notablemente su campo de actuación. Sin embargo estos avances han sido mucho más lentos que lo inicialmente esperado.

Como dice Alicia Casals en [4] los robots de servicios ofrecen una gran potencialidad para atender las crecientes expectativas sociales, especialmente en el mantenimiento y asistencia. La existencia de numerosos prototipos que consiguen niveles aceptables de movilidad para acceder al entorno de operación, así como suficiente destreza y capacidad de comunicación con un operador para efectuar las tareas que éste le encomienda, demuestra los avances conseguidos. A pesar de ello, actualmente no se dispone de robots de propósito general que puedan operar eficientemente en entornos diversos y en distintas aplicaciones, sino que hay que diseñar el robot específico para cada aplicación y entorno.

Para que un robot sea de propósito general necesitará efectuar constantemente las tareas de alcance y agarre, como parte de una manipulación de herramientas y objetos. Tales tareas aunque han sido muy estudiadas, aún no se logran efectuar en un sistema robótico de manera tan eficiente a como lo hacemos los seres humanos. Al contrario siguen presentes cantidad de problemas que además se ven incrementadas al hablar de entornos no controlados y para fines generales, que requiere la robótica de servicios.

El problema de desenvolverse en un entorno desconocido y dinámico, no es un problema nuevo en la robótica. La robótica móvil, se ha desarrollado en mayor parte, para hacer posible la realización de tareas de sistemas robóticos de manera autónoma, teniendo la habilidad de

moverse apropiadamente en un mundo real. En otras palabras, dichos robots deberían tener cierta percepción e inteligencia, para que puedan planificar sus propios movimientos. Dicha planeación incluye el modelado del entorno,

## **Motivación**

El progreso de la informática ha permitido dotar a los robots de mayor capacidad de control, mejores sistemas de percepción, haciendo posible el diseño de robots con comportamientos más inteligentes. En el sector de servicios y robots personales es donde probablemente el robot necesita más inteligencia, puesto que debe operar en entornos menos definidos y estructurados.

Desde los primeros proyectos en el sector de servicios, que se centraban en el campo asistencial o en el mantenimiento, el progreso ha sido lento y pocas veces la experimentación se ha llevado a cabo en condiciones de entorno reales. Ello se debe al hecho de que la robótica debe afrontar dificultades tecnológicas muy severas que pueden clasificarse en dos grandes grupos. La primera dificultad radica en la necesidad de conseguir una navegación segura cuando el robot debe desplazarse al lugar de actuación, lo que implica la disponibilidad de sistemas de posicionamiento. La segunda es la necesidad de percibir el entorno para navegar o para realizar el trabajo previsto. Los entornos de operación de los robots de servicios, según su campo de aplicación, difieren significativamente entre ellos y requieren a menudo arquitecturas especiales, tanto en lo que se refiere a su estructura física, como en su arquitectura de control.

Hacer una arquitectura de control para las tareas de Alcance y Agarre representa una oportunidad para centrarse en los problemas de control para manipular objetos; donde poder profundizar de forma eficiente; y aplicar diferentes conceptos utilizados y comprobados en otras ramas de la robótica, que puedan ser útiles en la propuesta de una arquitectura de control que sea eficiente para robots de servicios y que mejore características del control en tareas de Alcance y Agarre.

## Objetivos

Como se ha dicho anteriormente, las nuevas demandas de la robótica de servicios ha dejado al descubierto que actualmente no se dispone de robots de propósito general que puedan operar eficientemente en entornos diversos dinámicos y en distintas aplicaciones, sino que hay que diseñar el robot específico para cada aplicación y entorno. Dentro de la robótica móvil se han presentado soluciones en cuanto a la necesidad de navegación, percepción y autonomía de los robots. Sin embargo, en temas de control siempre quedan pendientes importantes problemas para resolver.

Por todo lo anterior, es importante identificar, analizar y proponer soluciones a los retos que surgen en el contexto de la robótica de servicio; dado que se requieren robots de propósito general que mimetice la interacción humana, (por ejemplo con un entorno dinámico y/o desconocido, humanos, otros robots , etcétera), estas soluciones deberían potenciar la capacidad de manipulación de los robots, mimetizando las capacidades y habilidades humanas; por lo tanto este trabajo se enfocará en elaborar un estudio de estos retos, identificar las arquitecturas y algoritmos existentes y analizarlos con el objetivo de identificar oportunidades de mejora.

Dentro de la tarea de manipulación diestra, las tareas de alcance y agarre

Nos proponemos:

- Analizar las principales características de las tareas de alcance y agarre dentro de la robótica de servicio y sus diferencias con la robótica industrial
- Estudio y análisis de la fusión multisensorial .
- Analizar los sistemas bioinspirados: comportamientos reactivos, inteligencia artificial, paradigma AER.

Así mismo, en la parte práctica del presente trabajo, pensar en una propuesta de un sistema robótico entero para este trabajo, era una opción inviable; ya que es demasiado ambicioso para un primer trabajo, como el presente. Por ello, una primera etapa de trabajo interesante para esta parte práctica para poder aplicar estudios sobre coordinación senso-motora y sobre integración de los canales sensoriales, es:

- el hacer un control de motores utilizando alguna de las técnicas utilizadas en la robótica móvil, ya que esta tarea es fundamental en la propuesta de una arquitectura de control completa, pues se vincula la entradas de los diferentes sensores como entradas de los parámetros de control de los motores de los diferentes componentes del sistema (articulaciones del brazo y de la mano) para finalizar con la memoria del procedimiento seguido y los recursos necesarios.

Como resultado de esta actividad, se espera poder proponer una nueva arquitectura de control usando algoritmos bioinspirados que permitan la mejora de estas tareas. Esto como parte de una tesis doctoral donde se parte con la hipótesis que el uso de sistemas bioinspirados y acciones reactivas en un sistema de control de alcance y agarre con fusión multisensorial, favorece las características de un sistema robótico que desarrolla estas tareas.

## **Estructura del trabajo**

Este trabajo consta de 5 capítulos. El primero es el actual capítulo donde se expone la motivación, objetivos y una pequeña introducción para el presente trabajo.

En el segundo se pretende dar una descripción de la evolución de la robótica industrial a la de servicios, con los diferencias en sus entornos que modifican la forma de resolver los diferentes problemas. Se describe las diferencias de las tareas de alcance y agarre en estos entornos tan diferentes, y porque vuelve a cobrar importancia estudiar estas tareas, sus características y principales problemas.

En el tercero capítulo se habla de la fusión sensorial como respuesta a la necesidad de los robots de servicios, a la percepción. Estos métodos se han usado en la robótica móvil, donde se intuye será de utilidad para la aplicación en esta robótica.

En el cuarto capítulo se dedica a métodos bioinspirados que se han aplicado en la robótica móvil e incluso en otros métodos de alcance y agarre como lo son las redes neuronales: estos métodos son comportamientos reactivos, redes neuronales y paradigma AER

Para el quinto capítulo se elegirá alguna técnica estudiada en el capítulo 4 para poner en práctica el controlar los motores para finalizar se incluirá la memoria del procedimiento seguido y los recursos necesarios.



## Capítulo 2: Tareas de Alcance y agarre para un robot de servicios

### De la Robótica Industrial a la de Servicios

El objeto de estudio actual sobre alcance y agarre solo puede entenderse en el contexto de la evolución que ha tenido la robótica en los últimos años. Gracias a la robótica de servicios, los estudios sobre estas tareas en los últimos años han vuelto a ser de gran interés para la comunidad científica. [1]

La primera empresa de robótica, Unimation, se fundó en la década de los 50's, cuando Joseph Engelberg junto con George Devol desarrollaron el primer robot de servicio exitoso. Se dice que este pionero de la robótica expreso: "*No puedo definir un robot, pero reconozco uno cuando lo veo*" [2]. Actualmente, existe una Federación Internacional de Robótica (IFR, por sus siglas en inglés) y en el año 2012 se publicó una regla ISO que define los términos utilizados en relación con los robots y dispositivos robóticos que operan tanto en entornos industriales y no industriales [ISO 8373:2012].

Según estos organismos (IFR e ISO) la robótica se define como la ciencia y la práctica de diseño, fabricación y aplicación de robots.

La robótica ha evolucionado con rapidez, recientemente ha comenzado a cobrar fuerza un nuevo tipo de robot, inicialmente denominado "robot de servicios", para diferenciarlo de su directo antecesor, el robot industrial. Sin entrar en definiciones formales, parece que existe suficiente consenso en la comunidad científica para considerar a dicho robot de servicios como aquel que representa un estadio intermedio en la evolución desde el más extendido robot industrial hasta el más complejo, denominado robot humanoide [1].

Según la IFR, un robot es un mecanismo programable accionado en dos o más ejes, con un grado de autonomía, moviéndose dentro de su entorno, para realizar tareas previstas. La autonomía en este contexto se entiende como la capacidad de realizar tareas previstas sin intervención humana. Un sistema robótico es un conjunto que comprende robot (s), efector

final (s) y cualquier tipo de maquinaria, equipos, dispositivos o sensores que apoyen desempeño de la tarea del robot.

Los robots pueden ser clasificados en diferentes categorías dependiendo de su función y las necesidades del mercado para las que fueron diseñados. Los que están mayormente identificados son los robots industriales y los de servicios, estos últimos también pueden ser subdivididos en robots de servicio personal o profesional, dependiendo la función para la cual sea usado.

Un Robot industrial de manipulación, para la IFR, es una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento.

Un robot de servicio es definido como un robot que opera semi o totalmente autónoma para realizar servicios útiles para el bienestar de los seres humanos y equipos, excluyendo las operaciones de fabricación.

Un robot de servicio profesional o un robot de servicio para uso profesional es un robot de servicio utilizado para una tarea comercial, generalmente manejado por un operador debidamente capacitado. Ejemplos de ellos robots de limpieza del robot los lugares públicos, robot de entrega en las oficinas u hospitales, robot de lucha contra incendios, robot para rehabilitación y robot de cirugía en los hospitales. En este contexto, un operador es una persona designada para iniciar, supervisar y detener el funcionamiento previsto de un robot o un sistema robótico.

Un robot de servicio personal o un robot de servicio para uso personal es un robot de servicio que se utiliza para una tarea no comercial, para personas no profesionales. Ejemplos de ello son robot doméstico, silla de ruedas automatizada, la movilidad personal con asistencia robótica y robot para hacer ejercicio.

La robótica industrial lleva en el mercado más de 4 décadas, sin embargo con la creciente demanda de los robots de servicios, la robótica vuelve a presentar un abanico de retos y problemas a resolver; esto debido a las características de diseño que difieren en estas 2.

Dentro de un ambiente controlado, como lo es una fábrica (robótica industrial), se pueden favorecer las situaciones para controlar al robot, por ejemplo la iluminación, evitar el paso de personal cuando el robot está operando, entre otros. Actualmente las aplicaciones donde el robot trabaja en entornos no estructurados se llevan a cabo, casi en su mayoría, mediante la teleoperación. La mayoría de estos robots tienen detección bastante primitiva y la percepción se limita a entornos estructurados 2D. [1]

En la automatización industrial se usan manipuladores rápidos, precisos y fáciles de programar, que realizan tareas repetitivas y tienen herramientas de proceso específicamente diseñadas para la tarea a realizar. Para realizar tareas de alcance y agarre cada robot tiene un efector final que en su mayoría ha sido diseñado para una tarea específica, por lo cual solo agarra un tipo de objeto o herramienta hasta que es reprogramado para manipular otro. [1] En cuanto a la generación de trayectorias de alcance; por lo general la pieza va al robot, por medio de cintas transportadoras hasta una posición inicial donde el robot inicia su rutina solo si un sensor le avisa que la pieza está colocada en la posición indicada.

Por el contrario, en las aplicaciones de la robótica de servicios, resulta imposible estructurar el entorno; por lo que se ha intentado dotar a estos robots de un mayor grado de inteligencia (*razonamiento*) y percepción. El razonamiento determina las acciones que se han de realizar en cada momento, según el estado del robot y su entorno para alcanzar las metas asignadas. Mientras que la percepción determina la relación del robot con su entorno de trabajo; mediante el uso de sensores del robot, le permiten localizar objetos, obstáculos, personas, etcétera y ubicarse en un entorno 3D desconocido y dinámico. [3] Los robots personales tienen que operar en las proximidades de los seres humanos de una manera segura, exigiendo una interacción hombre-robot que se traduce en desafíos significativos en las interfaces humano-robot. Con la aparición de los sistemas en red, integrados y la mayor presencia de las redes en los hogares y en las fábricas, los robots tendrán que trabajar con otros robots, aprender de diferentes tipos de sensores e interactuar con los diferentes usuarios humanos en función de su entorno inmediato. Esto se hace ya en los sistemas robóticos móviles, cuyos ambientes son dinámicos también. [4]

Dentro de la robótica de servicios para las tareas de alcance y agarre se requiere que la manipulación sea multitarea y la interacción física con el mundo real de los robots exige un sistema de control más complejo, en el cual, la mayoría de las veces para lograr hacer bien una

tarea de manera segura y exitosa, será conveniente integrar más de un tipo de sensado; por ejemplo visión en apoyo de la manipulación diestra.[4] Los efectores finales deben ser más versátiles con propósitos más generales, se tiende a utilizar actuadores más complejos, entre los que podemos establecer la distinción de dispositivos de agarre con origen en un modelo antropomorfo (inspirados en la mano humana) y aquellos que no siguen este modelo [3]

Para este trabajo puntualizaremos las demandas de la robótica de servicios para tareas de alcance y agarre. Para ello es necesario definir estos términos.

## Alcance y Agarre

El control sensoriomotor de los movimientos de la mano llevado a cabo por el Sistema Nervioso Central (SNC) se ha estudiado ampliamente con lo que se ha demostrado que el SNC separa el control de alcanzar y agarrar; por lo tanto, las configuraciones de brazos y manos dependerá de las propiedades del objeto (posición, orientación, geometría y tamaño), así como la próxima tarea a realizar. [5]

Partiendo de esta idea, que un robot sea capaz de localizar un cierto objeto, llegar hasta el evitando obstáculos intermedios y situar la muñeca (efector final de esta fase) en las proximidades del objeto, es a lo que nos referimos al hablar de una **tarea de ALCANCE**. Para ello se requiere de una planificación de trayectoria y el espacio con las diferentes configuraciones. [6]

A todo el proceso necesario para que el robot coloque el efector final en la posición más conveniente y aplique las fuerzas necesarias para sostener el objeto sin causar daño, es a lo que nos referimos con la **tarea de AGARRE**. [7]

Es importante diferenciar cuando se está hablando de la tarea de agarre y cuando hablamos simplemente de elagarre. El AGARRE se puede definir como un conjunto de lugares, puntos o regiones en la superficie del objeto donde se han de colocar los dedos de efector para agarrarlo. [6]

A continuación analizaremos las características que deben tener los sistemas de control para las tareas de alcance y agarre por separado; como bases para poder proponer un proceso conjunto de variables de control (ángulos, longitud del músculo y la fuerza muscular) inspirados en el funcionamiento del SNC, que permita encontrar una solución común al problema de la cinemática y la dinámica del movimiento, contrario a la robótica tradicional, donde son tratadas de manera separada.[7]

## Alcance y Agarre en la Robótica de Servicios

Como se ha dicho anteriormente las características que se buscan en un robot de servicios, dada su misma definición, son que tienen que operar semi o totalmente autónomamente en entornos no controlados y dinámicos, también como realizan diversos servicios o tareas, deben ser multitarea.

También se ha dicho que para lograr esta versatilidad que ayude en la multitarea, son necesarios efectores finales más complejos, por lo regular estos tienen mayor grados de libertad, lo que supone una complejidad adicional al decidir el agarre más adecuado a la tarea.

La biomimética o biomimesis es un campo de la investigación que trata de imitar a la naturaleza para lograr nuevos materiales y tecnologías.[8] En los últimos años la tendencia para crear sistemas basados en principios biomiméticos va en aumento, por lo que no es de extrañar que los dispositivos elegidos como efector final sean más parecidos a una mano humana. [1]

Hay que tomar en cuenta que una mano humana de cinco dedos cuenta con veinticinco grados de libertad (palma y la muñeca no incluidos), accionado por casi cuarenta músculos [5]. Los actuadores parecidos a una mano humana pueden llegar a tener más de 20 grados de libertad. Esto hace que sea un sistema sumamente complicado, ya que hay una gran cantidad de variables a controlar.

Para la mayoría de tareas motoras, existe un número extremadamente grande de maneras en las que éstas pueden ejecutarse. Por ejemplo, cuando se trata de alcanzar un objeto, la trayectoria de la mano, en principio, es arbitraria entre el punto inicial y el punto final del movimiento, y la velocidad con la que se ejecuta la trayectoria también es arbitraria. Además, debido al exceso de grados de libertad (GDL) en los efectores de movimiento en primates superiores y humanos, existe un número infinito de maneras en las que una trayectoria arbitraria puede implementarse a través de una sucesión de posturas del brazo. A nivel de biomecánica, se encuentra un mayor grado de redundancia ya que, en humanos, el número de músculos asociados a un solo GDL es grande. A pesar del alto nivel de redundancia existente, las investigaciones al nivel del comportamiento motor, han establecido una gran cantidad de

regularidades a través de la experimentación con distintos individuos e incluso con primates de distintas especies.[9]

Estudios psicofisiológicos sobre los movimientos llevados a cabo por los humanos en comportamientos como los de alcance de Morasso y Teulings en la década de los 80's, han demostrado que las trayectorias espaciales son más invariantes, que las rotaciones de las articulaciones o que los patrones fuerza-tiempo. [10]

Otro gran problema para los sistemas de alcance y agarre de la robótica de servicios, el de desenvolverse en un entorno 3D desconocido y dinámico. La tarea de la fusión multisensorial dentro de un sistema robótico pensado para el ámbito de servicios es fundamental, ya que un solo tipo de sensado es prácticamente impensable poder realizar todas las funciones que un robot de servicios necesita dar. Dentro de la robótica móvil se han hecho aplicaciones que han mejorado algunos de los problemas de navegación en entorno dinámicos las cuales son técnicas de fusión sensoria, pero también muchas son bioinspiradas como los comportamientos reactivos [14], las redes neuronales [6] y el paradigma AER [15]. Es importante estudiar estos métodos y de acuerdo con las características del alcance y agarre ver en la robótica de servicios si es viable su aplicación.

Por ejemplo para el agarre, es importante en el control de las fuerzas que se deben aplicar a los controles, poder proveer al sistema de información táctil. Es importante destacar que solo se puede hablar de sensado táctil cuando el actuador está en contacto con el objeto a agarrar; por lo que se entiende que un sistema táctil por si solo puede llegar a ser muy complejo, para lo cual se procura combinar con otro tipo de sensado, visión el más común, lo que ayuda a hacer esta tarea más sencilla. [1]

Para que el alcance se lleve a cabo con éxito el robot requiere ubicarse, localizar el objeto a manipular, llegar hasta él y evitar coaliciones; estas tareas no son nuevas dentro de la robótica; de hecho son comunes dentro de la robótica móvil; donde hay publicaciones con técnicas con diferentes métodos; pero donde se destaca mucho la bioinspiración como respuesta a estas demandas (insertar citas).

Analicemos a continuación las características de los sistemas de Alcance y Agarre.

## Características de los Sistemas de Alcance y Agarre

### Alcance

En el primer capítulo hablamos de que en la robótica de servicios se desarrolla en un ambiente controlado, donde se pueden favorecer las situaciones para controlar al robot; todo lo contrario a los robots de servicios, los cuales se desenvuelven en entornos desconocidos y no controlados, donde hay que evitar colisiones debido a cambios del entorno en tiempo real.

Imaginemos una simple tarea de alcanzar una pieza y agarrarla para una tarea posterior. Un robot industrial tendría zona de trabajo limpia de colisiones, y lo más seguro es que la trayectoria estuviera programada para repetirse una y otra vez. Para un robot de servicios es impensable, ya que en el entorno estarían circulando humanos, otros robots, mascotas etcétera. Además las condiciones de luz dependerían de la hora del día; todo esto pensando que el robot solo tiene que mover un manipulador para alcanzar el objeto deseado. Según la regla ISO un robot móvil se define como un robot que podrá viajar bajo su propio control, puede ser una plataforma móvil con o sin manipuladores. En la situación anterior hemos aclarado que el robot ha tenido que trasladarse, pero sin embargo el hecho de que tenga que operar en entornos dinámicos y que sea seguro para la interacción con humanos; hace que muchos problemas de la robótica industrial se comparta con los de la móvil. Ambos robots necesitan actuar ante la incertidumbre, manejar riesgos inherentes a una operación en ambientes dinámicos desconocidos en tiempo real, las aplicaciones frecuentemente requieren de experimentación y reconfiguración, por lo que necesitan brindarle al diseñador de cierta flexibilidad. [11]

En conclusión, se puede decir que el robot de servicios necesita características similares al sistema de percepción de un robot móvil para poder hacer un alcance libre de colisiones en un entorno cambiante. Ya que los dos deben planear trayectorias en entornos desconocidos; por lo cual los dos necesitan un sistema de control que les permita una respuesta rápida ante los cambios.

Otro gran problema al que nos enfrentaremos en la tarea de alcance es el de la redundancia en su configuración, al ser el brazo virtual antropomorfo una cadena cinemática abierta, el extremo de la cadena cinemática (muñeca) puede alcanzar una posición con diferentes configuraciones de las articulaciones del hombro y el codo. [7]

En definitiva, la tendencia actual de los sistemas de control en robótica, es la de emular el comportamiento biológico del sistema humano. Por tanto, El estudio del comportamiento humano permite trasladar a los sistemas artificiales dos conceptos de relevante importancia en este campo: Redundancia y Acomodación del movimiento del sistema motor. La primera de ellas supone la toma de decisión en tiempo real de todos los grados de libertad que intervienen en el proceso. La segunda permite diferenciar las diferentes fases que se producen en un movimiento de agarre o manipulación. [10]

Para [9] los principales problemas que hay en alcance se pueden resumir en la siguiente lista:

- Perfiles de velocidad acampanados y curvatura en las trayectorias de movimientos de alcance: las características de curvatura en el alcance, son producto y consecuencia del proceso de *aprendizaje motor*, proceso que inevitablemente, conlleva una serie de inexactitudes e imperfecciones que tienen como consecuencia la ejecución de trayectorias ligeramente curvadas (Bullock et al 1993, Guenther y MicciBarreca 1997). Esta última hipótesis, permite el desarrollo de una serie de modelos neuronales.
- Segmentación del Movimiento: Parece ineludible el que, para llevar a cabo un aprendizaje motor eficiente, los sistemas de movimiento planifiquen las tareas en niveles altos de abstracción por encima de los comandos motores individuales, ya que de otra manera la dimensionalidad del espacio de búsqueda durante la etapa de aprendizaje motor sería demasiado elevada a la hora de definir acciones motoras correctas para la ejecución de una tarea determinada. Las primitivas motoras, también llamadas unidades de acción, comportamientos básicos o gestos, pueden ofrecer esa clase de abstracción.
- El dilema Velocidad-Precisión : En el alcance rápido de un objetivo, el tiempo de movimiento ( $MT$ ) ha sido empíricamente establecido como una función de la distancia  $A$  (desde el punto de inicio hasta el punto de llegada) y del tamaño del objetivo  $W$  (equivalente a la precisión requerida en el alcance) a través la expresión logarítmica

$$MT = a + b \cdot \log_2 \left( \frac{2A}{W} \right) \quad (1)$$

donde  $a$  y  $b$  son constantes de proporcionalidad de la llamada Ley de Fitts o *dilema velocidad- precisión* (Fitts, 1954). Dado que la Ley de Fitts constituye un hecho contrastado en el control motor humano de brazos y manos, numerosos modelos computacionales la han utilizado a la hora de verificar su validez.

- El problema de la Redundancia o de la Equivalencia Motora: Como ya se ha mencionado, cuando se alcanza un objeto en el espacio tridimensional que nos rodea, el excesivo número de GDL existente en la estructura cinemática del cuerpo humano permite adoptar un número enorme de posturas para la misma posición de la mano (en coordenadas externas) a lo largo de la trayectoria de movimiento. A este hecho también se le conoce como problema de la equivalencia motora.
- Aprendizaje Motor: Dadas las continuas alteraciones en la longitud y propiedades biomecánicas de los distintos miembros del cuerpo durante su desarrollo, la capacidad de aprendizaje motor en un sistema biológico de control del movimiento es fundamental. Cuando se trata del control del movimiento del brazo y la mano, los primates muestran un elevado grado de flexibilidad a la hora de desarrollar nuevas estrategias motoras para solucionar nuevas tareas. Así pues, el aprendizaje es un elemento fundamental en un modelo computacional de control motor del movimiento del brazo y la mano.
- Acoplamiento Acción-Percepción: Las secciones presentadas hasta el momento, se centran primordialmente en estudios básicos de control motor y no tanto en cuestiones que involucran la coordinación percepción – acción motora. Sin embargo, la interacción entre la percepción y la acción revela muchas de las estructuras que el SNC emplea en el control motor. En el caso del aprendizaje motor relacionado con el control del movimiento de alcance y agarre, se ha establecido que los primates son capaces de aprender por sí mismos las transformaciones espaciales–motoras que posteriormente les permiten ejecutar tareas en el espacio 3D.

## **Agarre**

Para realizar exitosamente la tarea de agarre, hay un proceso que se debe hacer cuando se ha detectado el tamaño y forma del objeto deberá ejecutar el algoritmo de síntesis de agarres. El objetivo de este algoritmo es el cálculo ' del agarre óptimo ' para una determinada pieza [12]

En la actualidad existen grandes problemas cuando se quiere generar posibles agarres, ya que como [12]dice, no existe ningún método que permita generar de forma automática un conjunto de posibles agarres. Por ello, para poder desarrollar un método propone un triángulo con prioridades, el cual es:

- Tomar en cuenta complejidad del efector
- Necesidad de reconocer el tipo de objeto
- Seleccionar la mejor forma de agarrarlo dentro de todas las posibilidades.

El problema de seleccionar un conjunto de contactos sobre un objeto para agarrarlo con un efector, es la síntesis de agarres. Mientras que el análisis de agarres comprende el estudio y evaluación de un determinado agarre que responda a los requisitos de la tarea [13]asegurando la estabilidad durante el agarre del objeto.

Para modelar un agarre robótico, debe considerarse la cinemática y capacidades de la mano robótica; analizar las limitaciones de los requisitos de la tarea.[1]

Las características principales que se buscan en un agarre son [14] [15][12]:

- Estabilidad: Depende concretamente de la disposición de los puntos de contacto sobre la superficie del objeto e indica si las fuerzas y pares generados como resultado del desplazamiento se oponen o se suman a las que ocasionaron el mismo. En otras palabras, si el objeto volvería a su posición inicial después de un desplazamiento infinitesimal.
- Equilibrio: Las fuerzas ejercidas por la pinza a través de los puntos de contacto contrarrestan las debidas a la gravedad. Es decir, una supuesta situación estática, por lo que la suma de las fuerzas y los momentos que actúan sobre la pinza debe ser igual a cero.
- Viabilidad: Se refiere a la existencia de una configuración para los dedos de la pinza capaz de lograr que los extremos de los dedos se sitúen sobre los puntos de contacto.
- Alcanzabilidad: Posibilidad de encontrar un camino libre de obstáculos y evitar colisiones lo que permita una configuración de agarre partiendo de una configuración inicial.
- Rigidez: Relación entre la fuerza o par externos que es necesario aplicar al objeto para que este se desplace de su posición y la magnitud del desplazamiento considerando que el efector no se desliza por la superficie del objeto.
- Resistencia al deslizamiento: se mide como la máxima fuerza o par externos que es posible ejercer (bien en cualquier dirección o en ciertas direcciones relevantes) antes

de que algún dedo comience a deslizar. El origen de las perturbaciones externas puede ser cualquiera de los mencionados anteriormente.

- Manipulabilidad: se define como la capacidad de cambiar la posición de la pieza mediante movimientos de los dedos. Por lo general, un agarre firme proporciona escasa manipulabilidad y viceversa.
- Flexibilidad: Se dice que el sistema es flexible cuando permite un cierto ajuste de la pieza ante esfuerzos externos. Esta característica es de utilidad para corregir errores del sistema de posicionado (alcance).

En la literatura encontramos que hay dos tipos de agarre [16]:

Agarre Dactilar (pinchgrasp): Es aquel donde el contacto con el objeto se hace únicamente con los extremos o las yemas de los dedos.

Agarre envolvente (envolvinggrasp, powergrasp): Cuando el contacto con el objeto es a través de la parte interna de los dedos o lo que sería equivalente a la palma de la mano.

También se puede clasificar en función del objetivo del agarre, ya sea inmovilizar la pieza o reorientarla. [12]

Cuando se desea inmovilizar la pieza, los dedos se posicionan sobre ella y ejercen una presión constante durante todo el proceso, de forma que idealmente no se desplazan de su Agarre de piezas mediante robots posición; un ejemplo sería el transporte de una pieza desde una posición a otra. Cuando se desea reorientar la pieza, los puntos de contacto de los dedos varían, existiendo diversas estrategias: rodamiento de un dedo sobre un punto de contacto (rolling), deslizamiento (sliding) o liberación de uno de los dedos y colocación en una posición nueva (fingergaiting)

Existen distintas propiedades que permiten determinar si un agarre inmoviliza adecuadamente la pieza o no. Se trabaja siempre sobre el agarre definido como un conjunto de puntos de contacto; por tanto, contactos lineales y contactos superficiales deben ser descompuestos en contactos puntuales como se explicó anteriormente. [12]Las propiedades son:

Forceclosure, 4 o cierre de fuerza: un agarre cumple esta condición si es posible contrarrestar cualquier fuerza o par externo que actúe sobre el objeto aplicando fuerzas sobre los puntos de contacto. No se hace ninguna consideración sobre la magnitud de las fuerzas que sería necesario aplicar: en determinadas circunstancias, para contrarrestar una pequeña fuerza externa en una dirección concreta podría ser necesario aplicar fuerzas desmesuradamente grandes (inviabiles) en algún punto de contacto, pero el agarre seguiría cumpliendo la condición de forceclosure.

Formclosure o cierre de forma: se dice que un agarre cumple esta propiedad cuando la situación de los puntos de contacto restringe completamente todos los movimientos de la pieza (se restringen todos los grados de libertad de la pieza).

Immobility o inmovilidad: Se trata de una condición menos estricta que la condición de formclosure dado que toma en consideración la curvatura local en los puntos de contacto. La curvatura puede indicar la imposibilidad de movimiento de un objeto a pesar de que el agarre no cumpla la condición de formclosure.

Durante las 2 últimas décadas, ha estado en constante estudio el área de investigación referente a los agarres y a la manipulación robótica, abordando la problemática crucial en este ámbito y desarrollando una extensa base teórica. Todos estos trabajos se han clasificado en 3 categorías: análisis de agarres, medidas de calidad de agarre y síntesis de Agarre.

Aunque el control de ambas tareas se realice por separado, ambas tareas de control son uno de los principales retos actuales. [12]

En las secciones anteriores se ha analizado las tareas de alcance y agarre por separado. Sin embargo, también hemos dicho que el SNC hace estas dos tareas paralelamente, por lo que no se hace el alcance y después el agarre. Lo cual hace que la tarea de agarre, se empiece con el robot en movimiento lo que trae consigo problemas del tipo de movimientos de Agarre con Perturbación y efectos de la variación del tamaño del objeto (Variaciones sistemáticas en el tamaño del objeto y variaciones abruptas en el tamaño del objeto durante la ejecución del movimiento). [9]

Otro problema tratado en [17] de agarre que se ha estudiado recientemente, se trata que el agarre en un robot de servicios, por lo general se hace para ejecutar una tarea, por lo que se requiere de llevar a cabo un agarre efectivo que permita una manipulación posterior. En este artículo se toma como ejemplo una taza como objeto para agarrar, sin embargo, no es lo mismo agarrar una taza vacía, que una taza con agua; en esta segunda condición la estabilidad es primordial, pero además de esto, hay una restricción adicional, la cual es, que la mano del robot debe evitar la zona de apertura de la taza y es preferible que el agarre se haga por el asa. Para resolver este problema Hao Dang y Peter K. Allen proponen generar agarres semanticos (semantic grasp) [17]

## Conclusiones

En conclusión, no hay que perder de vista las dos dificultades principales a las que nos enfrentamos dentro de la robótica de servicios; las cuales son : entornos dinámicos, 3d desconocidos y con propósitos generales, por lo que los efectores finales tienden a ser más complejos.

Recapitulando, hemos visto, que la mecánica de los robots está altamente desarrollada, y en el mercado hay una gran variedad de sensores, etc. Los principales retos actuales dentro de la robótica de servicios están en el ámbito de control. Un robot de servicios contará con (extremidades) complejos que incluyen un alto número de grados de libertad. Por lo que sería útil que el sistema de control pudiera aprender posiciones, dinámica inversa, métodos para reducir el número de variables a controlar con tantos grados de libertad, un sistema inteligente. además de incrementar su nivel de percepción no solo con poner más sensores, sino el control que se hace para poder procesar toda la información sensorial.

Los robots de servicios deben funcionar en entornos dinámicos desconocidos al igual como lo hacen los

Otro gran problema para los sistemas de alcance y agarre de la robótica de servicios, es el desenvolverse en un entorno 3D desconocido y dinámico. La tarea de la fusión multisensorial dentro de un sistema robótico pensado para el ámbito de servicios es fundamental, ya que un solo tipo de sensor es prácticamente impensable poder realizar todas las funciones que un robot de servicios necesita dar. Y en cuanto a la navegación hay que fijar la mirada en los avances en materia de control bioinspirado que se han aplicado en la robótica móvil.

Por ejemplo para el agarre, es importante en el control de las fuerzas que se deben aplicar a los controles, poder proveer al sistema de información táctil. Es importante destacar que solo se puede hablar de sensor táctil cuando el actuador está en contacto con el objeto a agarrar; por lo que se entiende que un sistema táctil por sí solo puede llegar a ser muy complejo, para lo cual se procura combinar con otro tipo de sensor, visión el más común, lo que ayuda a hacer esta tarea más sencilla. [1]

Otro concepto que no se debe olvidar, es que aunque se hagan controles especiales para cada tarea de alcanzar y agarrar debe haber un proceso conjunto de variables de control (ángulos,

longitud del músculo y la fuerza muscular); por lo tanto, las configuraciones de brazos y manos dependerá de las propiedades del objeto (posición, orientación, geometría y tamaño), así como la próxima tarea a realizar. Por lo tanto estas 2 tareas serán codependientes.

Por último, debemos recordar la necesidad de tener varias fuentes de sensado para que nuestro robot pueda desenvolverse en un entorno 3D desconocido y dinámico, por lo cual analizar las técnicas de fusión multisensorial, es una tarea básica, para el desarrollo de una arquitecturas de control que integren algoritmos y conceptos biomiméticos.

La tarea de la fusión multisensorial dentro de un sistema robótico pensado para el ámbito de servicios es fundamental, ya que un solo tipo de sensado es prácticamente impensable poder realizar todas las funciones que un robot de servicios necesita dar. Por ello vamos a estudiar más a fondo este tema de control.

## Capítulo 3: Fusión Multisensorial

### Introducción

Los sensores son la forma en que un sistema recibe información útil sobre las características del entorno que le rodea. No es difícil intuir que, utilizando un solo sensor, solo se puede obtener datos parciales de este. Las ventajas potenciales de la integración de múltiples sensores, sobre sensores individuales, para hacer implementaciones en sistemas donde el ser humano interactúe cada vez menos son: que la información obtenida es más precisa y más completa, ya que, existen características que son imposibles de percibir con sensores individuales [23]

Los avances en la tecnología de los sensores, han ayudado a que estos sean más ágiles y sofisticados, motivando los estudios hacia la automatización o semiautomatización, es así como nace la necesidad de utilizar varios sensores, del mismo tipo o diferentes, para brindar diferentes datos y al integrarlos tener una mejor imagen del entorno, derivando en aplicaciones en múltiples campos: militar, industrial, médico, entre otros; y tomando cada vez más importancia para estas aplicaciones.

La fusión sensorial o multisensorial es un tema bastante nuevo y amplio, la terminología es amplia y confusa, ya que la mayoría de las veces no está generalizada. Es común encontrar varios autores refiriéndose a los mismos conceptos con diferentes nombres. Un claro ejemplo de esto es la evolución que ha tomado el nombre fusión sensorial hasta llegar a Fusión e Integración Multisensorial; donde cuando algunos autores plantearon el debate entre fusión vs integración[1.], ahora la tendencia es a referirse indiferentemente o incluso referirse como Sistemas de integración y fusión multisensorial[4]

Para el presente trabajo, pensando en un sistema robótico de servicios destinado a las tareas de alcance y agarre, se requiere de al menos un brazo manipulador, un efector final antropomorfo que le facilite ser multitarea, y para que pueda desenvolverse en un entorno dinámico desconocido, es importante que cuente con un sistema de visión estereoscópico, que le ayude al robot, tener una referencia en 3D del entorno.

Históricamente el primer trabajo que intentó unificar términos con el *Data Fusion Lexicon*, fue el modelo JDL (Joint Directors of Laboratories) [18], es genérico pero permitió poner las bases para un común entendimiento, pues se desarrolló con el fin de mejorar la comunicación entre investigadores militares y desarrolladores de sistemas en el año 1986. Además sentó las bases para futuros debates y publicaciones. El JDL define fusión de datos como un proceso de multinivel y multifacético, que permite la detección, asociación, correlación, estimación y la combinación automática de los datos e información de varias fuentes. Klein [19] generaliza esta definición indicando que los datos pueden ser proporcionados por una o varias fuentes. Ambas definiciones son generales y son modelos que se pueden aplicar a diferentes campos.

Con el mayor desarrollo en la tecnología de los sensores, nace también la gestión multisensorial, la cual se puede referir como un sistema o proceso que busca manejar o coordinar el uso de diversos sensores, gestionando los dispositivos dentro de un entorno dinámico e incierto, para mejorar el funcionamiento de la percepción y fusión de datos. Se suele distinguir entre fusión directa, fusión indirecta y una fusión mixta si se combinan la salida de los dos métodos anteriores. [1]

Se habla de fusión directa cuando los datos provienen de un conjunto de sensores y los valores históricos de los datos sensoriales; por supuesto, las fuentes de datos no forzosamente son sensores idénticos en los procesos de fusión, estos pueden ser heterogéneos, homogéneos, sensores virtuales (soft sensors), etcétera.

En la fusión indirecta se usa como fuente de información un conocimiento previo sobre el entorno o la intervención humana. [20]

En 1989, Ren C. Luo y Michael [18], tratan brevemente el tema fusión sensorial vs integración. Para ellos el término de fusión sensorial se refiere a cualquier estado en la integración del proceso donde hay una combinación o fusión de diferentes fuentes de información sensorial en un formato único; mientras que la integración multisensorial se define como el uso de información proveniente de múltiples sensores para asistir y completar la tarea de un sistema. Wald [21] afirma que la fusión sensorial es un sistema de procesamiento en el que están expresados los medios y las herramientas para la alianza de los datos procedentes de diferentes fuentes, buscando obtener la “mayor calidad” dependiendo de la aplicación. Wald considera como datos de diferentes fuentes aquellos datos de la misma fuente en instantes distintos.

Boudjemaa y Forbes [4] hicieron la siguiente clasificación para los tipos de fusión de acuerdo a que faceta en el sistema están fusionando:

- *Sensor Fusion*: en este caso un determinado número de sensores miden la misma propiedad que es fusionada para obtener información más precisa y fidedigna acerca del fenómeno en observación.
- *Attribute Fusion*: Varios sensores miden diferentes atributos de la misma situación experimental.
- *Fusion across domains*: Los sensores miden el mismo atributo sobre un puntual número de propiedades y rangos.
- *Fusion across times*: Para mayor precisión en la determinación de la información histórica del sistema, por ejemplo a partir de una calibración anterior, la fusión se realizará con las mediciones actuales.

Durrant-Whyte [9] clasifica un sistema de fusión sensorial de acuerdo a tres configuraciones u escenarios que pueden estar presentes simultáneamente en aplicaciones del mundo real:

- Fusión sensorial de tipo competitiva: El objetivo de esta configuración es reducir los efectos del ruido y mediciones erróneas. Es una clase de aplicación de fusión donde se combinan datos de la misma medida para aumentar la fiabilidad, precisión y reducir los conflictos.
- Fusión sensorial de tipo complementaria: Combina datos del sensor que no tienen dependencia entre sí para crear un modelo más completo. Los sensores no dependen directamente de los demás, al contrario, al fusionarse dan información más completa del fenómeno bajo observación.
- Fusión sensorial de tipo cooperativa: combina las observaciones de los diferentes sensores que dependen entre sí, para dar lugar a una medición de nivel superior.

En 2011 Ren C. Luo [5], utiliza el término de integración y fusión multisensorial (MFI, por sus siglas en inglés: Multisensor fusion and Integration) para aludir a la tecnología concerniente a la combinación de datos provenientes de varios sensores, mejorando la precisión y haciendo que la información sea más confiable que si se obtuviera con un solo sensor.

El mayor desarrollo de la fusión sensorial, como en la mayoría de la tecnología, se ha desarrollado para aplicaciones militares, no hay que olvidar que de ellos vino el primer modelo JDL, aunque actualmente hay grandes aplicaciones dentro del plan académico, comercial, industrial con la implementación de robots, control automático de sistemas de manufactura industrial, desarrollo de edificios inteligentes y con las aplicaciones médicas.

Es claro el avance de los últimos años relacionado con los sensores y la tecnología de los sistemas de control, sin embargo unas de las grandes barreras en la implementación efectiva de la fusión multisensorial sigue siendo, el intento de combinar datos precisos obtenidos en los sensores con variantes desconocidas de nuestro sistemas, señales de propagación y ambientes de ruido desafiantes [2.].

En conclusión, podríamos definir la **fusión multisensorial** como la combinación de los datos procedentes de lecturas de diferentes sensores en una única estructura de datos. Esta combinación se puede realizar con datos obtenidos directamente de sensores o estimados a partir de datos sensoriales, obtenidos todos ellos a partir de fuentes sensoriales diversas, de forma tal, que la información resultante de dicha combinación sea más precisa, más completa y más confiable que la que es posible conseguir a partir de las fuentes usadas de forma individual.

Una pregunta fundamental para el desarrollo de un sistema de fusión sensorial es: ¿En qué punto del proceso es conveniente combinar o hacer la fusión de datos? [00055sig]

Las arquitecturas dan respuesta a esta pregunta fundamental y por tanto tenemos varios tipos de arquitecturas, dependiendo de la forma en que se organizan y el momento de la fusión de datos sensoriales.

Un sistema de fusión sensorial tiene varios componentes funcionales que manejan y controlan el proceso. La forma en que están organizados se conoce como Arquitectura de fusión multisensorial, mientras que los algoritmos son cada una de las técnicas utilizadas para fusionar los datos que provienen de diferentes fuentes. [3]

## ARQUITECTURAS

No existe una sola clasificación de arquitecturas, varios autores presentan diferentes nomenglaturas y criterios.

Una forma de clasificar las arquitecturas es:

### **Centralizada**

En un sistema de fusión centralizada, la unidad de fusión se hace por medio de un procesador central o nodo que colecta la información de los diferentes sensores. Las decisiones se toman en esta unidad central, y se derivan a las salidas respectivas. Si todos los sensores están situados con precisión, teóricamente este es el método con mejor rendimiento [21]. No obstante, los errores de los sensores, por muy pequeños que parezcan, pueden hacer que el rendimiento del algoritmo de fusión sea muy bajo. Además de este, los principales inconvenientes de la arquitectura centralizada son:

- **Comunicación:** Se puede causar un cuello de botella ya que todos los sensores se comunican directamente con el nodo de fusión centralizada.

- Falta de flexibilidad: El sistema centralizado no puede adaptarse con facilidad a cambios en la aplicación.
- Vulnerabilidad: si hay un fallo en el procesador central, o la instalación central de comunicaciones, todo el sistema falla.
- No modularidad: Si se añaden sensores generalmente es necesario hacer modificaciones en la programación. Entre más sensores se requiere mejorar tanto la forma de comunicación (ancho de banda) y como el coste computacional, para que el sistema pueda procesar los datos adicionales.

### **Distributiva o descentralizada**

En este tipo de arquitectura [25], cada nodo puede funcionar como una unidad de fusión de datos local utilizando mediciones locales. El proceso de fusión puede ser iterativo y posteriormente converger en el resultado global como un sistema centralizado. Presenta las siguientes características:

- Escalabilidad: se usa principalmente en redes de gran tamaño.
- Eficiencia: Al estar dividido en varios nodos, se requiere menos transmisiones y el coste computacional se reparte en los mismos.
- Robustez y fiabilidad: Al contrario que los sistemas centralizados son más tolerantes a errores en la comunicación o sensores (ej. Retrasos en la transmisión de los datos)
- La autonomía: auto-adaptación y auto-organización.

La predicción del comportamiento global del sistema basado en las pequeñas unidades locales de fusión individuales es una tarea muy difícil. Por ello el principal inconveniente de este tipo de arquitecturas es la falta de estructura global que ayude a controlar el proceso de fusión.

### **Jerárquica**

La ejecución de las tareas de fusión se realiza en grados sucesivos descendentes. La carga computacional es reducida, por lo que puede ser muy eficiente. Su comportamiento está determinado, mayormente, por las condiciones iniciales [3.]. Sin embargo, algún cambio en el entorno, puede requerir de una nueva organización jerárquica de tareas, sobrecargando el sistema.

### **Federativa**

Es una arquitectura híbrida entre la distribuida y la jerárquica. Ofrece la autonomía de nodos de alta fusión de sensores que se encuentran en las arquitecturas distribuidas, pero la comunicación entre nodos intermedios está organizada de forma similar con el esquema jerárquico. De esta forma se mejora la robustez y flexibilidad del sistema de fusión; sin

embargo, no es compatible en entornos extremadamente dinámicos ya que no tiene la suficiente reestructuración dinámica necesaria [3].

### **Holonic**

Este tipo de arquitectura también es un híbrido. Está compuesto por *holons*, los cuales son unidades autónomas y autosuficientes que presentan las siguientes características:

- Autonomía: *holons* pueden crear y mantener la ejecución sus planes (funciones);
- Cooperación: proceso en el que los *holons* pueden desarrollar y ejecutar conjuntamente algunas tareas de fusión;
- Auto-organización: los *holons* son capaces de organizarse e intervenir para completar un determinado nivel de fusión en el sistema;
- Reconfigurabilidad: el plan de *holons* puede ser cambiado para que sea más oportuno y eficiente.

Los *holons* son recursivos, pueden contener otros *holons*.

Usando la arquitectura *Holonic*, las tareas de fusión complejas son resueltas por medio de la cooperación de varias *holons* juntas, siguiendo *holarchies*, que es la jerarquía de los *holons*.

Esta estructura jerárquica tiene la capacidad de generar vínculos dinámicos, por lo mismo es complicado predecir el rendimiento global y la tarea de organizar los *holons* puede ser difícil [3].

### **Market-based**

Es un enfoque nuevo y diferente para gestión de sensores y fusión aprovechando las técnicas de comercio electrónico [22]. La idea clave es que el campo de gestión de sensores se ha centrado tradicionalmente en la producción y procesamiento de la mayor cantidad de información posible, prestando poca o ninguna atención en el conocimiento del usuario o la demanda para el procesamiento de la información.

Esta arquitectura pone énfasis en la importancia de la intervención del ser humano en el bucle de fusión, adoptando un enfoque orientado en la asignación de recursos; mejorando su utilización pero aumentando la complejidad que implica la asignación de procesos.

## ALGORITMOS

En la literatura se encuentran dos principalmente dos nomenclaturas para clasificar los algoritmos de acuerdo al nivel de fusión.

Basándose en el modelo JHL, la clasificación es:

- Nivel bajo de fusión (Fusión de señales y píxeles) [28]

Los múltiples datos sensoriales; debido a las propiedades de la muestra de diferentes tipos de sensores, tales como frecuencia de muestreo, necesitan sincronizarse y adaptarse antes del proceso de fusión. Estas adaptaciones representan este nivel. Los métodos de estimación, tales como la media ponderada, el método de mínimos cuadrados, y filtros de Kalman se han utilizado con éxito para la fusión a nivel de señal de los sensores.

La fusión a nivel de píxeles se refiere al procesamiento de la imagen, a partir de la información original guardada de un pixel para generar una nueva imagen compuesta con mejor calidad y más características, ya que al tener más de un sensor se proporciona una mejor interpretación de la escena. Métodos de fusión de uso común en la fusión a nivel de píxeles incluyen la fusión de banda de racionamiento, el análisis de componentes principales (PCA), técnica de fusión transformada wavelet y el uso combinado de estos métodos.

- Nivel medio de fusión (Fusión de características)

Los datos fusionados a nivel de función son las características extraídas de señales e imágenes. Un procedimiento de fusión en general a nivel de funciones se puede descomponer en tres pasos: compatibilidad de característica de conjunto y la normalización; la reducción de características y concatenación; y por último la correspondencia de características. Hay muchos métodos que pueden ser utilizados para este nivel, algunos de ellos: máquinas de soporte vectorial (SVM), análisis de conglomerados K-means clustering, Kohonen featuremap y aprender de cuantificación vectorial (LVQ).

- Nivel alto de fusión (fusión de signos). [21]

La información fusionada a este nivel es una representación simbólica de los parámetros del proceso. Busca procesar la decisión local desde múltiples sensores para alcanzar una decisión conjunta, y esto se puede lograr por sistemas expertos, redes neuronales, adaptativos, y lógica difusa.

Considerando el tipo de proceso del que se derivan, se pueden nombrar como métodos de:

- **Estimación [21]**  
Se pueden clasificar en métodos recursivos y no recursivos. El algoritmo de fusión recursiva más famoso es el filtro de Kalman [8]. El filtro de Kalman es esencialmente un conjunto de ecuaciones matemáticas que aplican un tipo de estimador previsible. Proporciona una estimación óptima en un sentido estadístico si el sistema se puede describir con un modelo lineal, y tanto el sistema y el error del sensor puede ser modelado como ruido blanco gaussiano [9].  
Métodos de fusión no recursivos incluyen los promedios ponderados, los mínimos cuadrados y los métodos de estimación basados en covarianza.
- **Clasificación**  
Se refieren a los procesos de agrupación de datos de fuentes múltiples en conjuntos de datos clasificados. Una característica del espacio multidimensional es que se puede dividir en regiones distintas. La ubicación de la nueva función del vector se compara con ubicaciones preclasificadas en el espacio de características, permitiendo identificar qué clase de datos tiene los nuevos atributos de la entidad.  
Estos métodos se han utilizado masivamente para diversos campos de aplicación, tales como el procesamiento de imágenes, sensores remotos, procesamiento de señales, reconocimiento y seguimiento de objetos.
- **Inferencia**  
Inferencia bayesiana y el razonamiento Dempster-Shafer (DS) son dos de los algoritmos de identificación más populares, y cada uno tiene sus propias ventajas y limitaciones. Se muestra en la literatura [14] que la inferencia bayesiana puede abordar el problema actual de manera más eficiente, y al mismo tiempo, el razonamiento DS hace explícita cualquier falta de información relativa a la probabilidad de una proposición y puede abordar problemas que la probabilidad no puede.

Sin importar cuál sea su nomenclatura, ejemplos de algoritmos algunos algoritmos según las clasificaciones pasadas:

#### **Nivel bajo de fusión / Métodos de estimación**

- Recursivos: Filtro de Kalman, Filtro extendido de Kalman
- No recursivos: Promedio ponderado, mínimos cuadrados
- Covariance-based: Cross covariance, Covariance intersection, Covariance unión

### **Nivel medio de fusión / Métodos de clasificación**

- Parametric templates
- Cluster analysis
- K-means clustering
- Learning vector quantization
- Kohonen feature map
- Artificial neural network
- Support vector machines

### **Nivel alto de fusión / Métodos de inferencia**

- • Bayesian inference
- • Particle filters
- • Dempster-Shafer theory
- • Expert system
- • Fuzzy logic

Algunos autores, como Ren C. Luo, introducen además una cuarta clasificación: **métodos de inteligencia artificial**. [05486390]

Otra clasificación presentada en 05412296 es:

#### **A. Inferencia probabilística**

Los métodos probabilísticos se basan generalmente en la regla de Bayes, combinando los datos previos y las observaciones [8]. La fusión de datos Bayesiana se puede conseguir usando filtros de Kalman, y métodos secuenciales de Monte Carlo.

#### **B. Razonamiento Creencia probatorio: Evidential Belief Reasoning**

Esta categoría de algoritmos de fusión de datos se basan en Dempster-Shafer (DS), la teoría de la prueba [27]. Esta teoría introduce la noción de asignar creencias y plausibilidades a una posible hipótesis de la medición junto con la regla de combinación necesaria para fusionarlos. Se puede considerar como una generalización de la teoría bayesiana que se ocupa de las funciones de masa de probabilidad. En contraste con la inferencia bayesiana, que permite que cada fuente de información contribuya en diferentes niveles de detalle, por ejemplo, un sensor puede proporcionar información para distinguir las entidades individuales mientras que otros

sensores pueden proporcionar información para distinguir clases de entidades. Las probabilidades son asignadas solamente cuando la información de soporte está disponible.

### **C. Técnicas de Soft Computing**

Soft computing es una rama de algoritmos informáticos que ofrecen soluciones aproximadas a los problemas computacionalmente difíciles. Las principales subcampos dentro Soft Computing incluyen la lógica difusa, las redes neuronales, y la inteligencia de enjambre. La mayoría de estas técnicas están inspiradas en los procesos computacionales que se encuentran en los sistemas biológicos.

#### **D. Optimización basada en la fusión de datos: Optimization-based Data Fusion**

Estos algoritmos tratan de fusión de datos como una optimización de una función objetivo a menudo definido heurísticamente (costo). Se utilizan diversas técnicas de optimización.

La función objetivo se asocia generalmente a unos criterios de rendimiento. El proceso de optimización puede ser regularizado mediante la aplicación de ciertas restricciones basadas en el conocimiento previo sobre el fenómeno observado. Los métodos de fusión de datos teóricos de la información son una rama de los métodos de optimización basados en la función objetivo y se definen en términos de las medidas de información [15].

#### **E. Enfoque de fusión híbrida: *Hybrid fusion approach***

La idea principal en el desarrollo de algoritmos de fusión híbridos es que los diferentes métodos de fusión no deberían estar compitiendo, ya que cada uno aporta diferentes perspectivas. Combinando sus diferentes características se logra un mayor tratamiento integral de datos.

Basir [30] propuso un método de fusión basada en el razonamiento Dempster-Shafer difuso que sirve de marco para primeras evaluaciones. Los resultados experimentales muestran que el enfoque de fusión híbrida supera tanto la fusión razonamiento basado en la teoría difusa tradicional o DS; cuando se aplica a problemas de segmentación de la imagen.

## Conclusiones

Una pregunta fundamental para el desarrollo de un sistema de fusión sensorial es: ¿En qué punto del proceso es conveniente combinar o hacer la fusión de datos?[00055sig]

Las arquitecturas dan respuesta a esta pregunta fundamental y por tanto tenemos varios tipos de arquitecturas, dependiendo de la forma en que se organizan y el momento de la fusión de datos sensoriales.

Un sistema de fusión sensorial tiene varios componentes funcionales que manejan y controlan el proceso. La forma en que están organizados se conoce como Arquitectura de fusión multisensorial, mientras que los algoritmos son cada una de las técnicas utilizadas para fusionar los datos que provienen de diferentes fuentes

Hemos visto que dado nuestro sistema robótico requerido para llevar a cabo las tareas de alcance y agarre; este realizará integración y fusión de datos.

Hay algoritmos bioinspirados que ya se aplican para estas tareas como lo son las redes neuronales; hay otro tipo de arquitectura de la cual no se ha hablado en este capítulo, pero que resulta muy interesante de analizar dado los resultados en la robótica móvil, nos referimos a los comportamientos reactivos.

Dentro del control de robots, técnicas utilizadas en la robótica en general. Como se ha visto se han estudiado arquitecturas y algoritmos, pero los que más me han interesado son las técnicas bioinspiradas. En el siguiente capítulo estudiaremos tres técnicas bioinspiradas que han aportado importantes características a los robots móviles tales como aprendizaje, percepción del entorno real para poder evadir obstáculos de forma autónoma en tiempo real y un paradigma que ofrece al sistema una nueva forma de procesar la información de forma paralela, inmune al ruido y muy veloz.

Es importante

## Capítulo 4: Sistemas Bioinspirados

*“En las últimas décadas, la simulación y la síntesis de sistemas biológicos se han convertido en elementos imprescindibles para la comprensión y la explicación de la lógica y la complejidad de la vida, inscribiéndose en el núcleo mismo de la nueva biología” [Luisi, 2010; Solé y Goodwin, 2000; Stewart, 1999; Emmeche, 1998].*

*“A diferencia de todas las ciencias biológicas tradicionales, esta nueva ciencia, conocida como vida artificial, no se concentra de forma exclusiva en el estudio de la vida como ha sido (o como es) en nuestro planeta sino que intenta crearla (sintetizarla), además, en medios artificiales. Es decir, antes que un enfoque histórico, la vida artificial emplea un enfoque productivo” [Varela, 2000].*

La naturaleza siempre ha sido fuente de inspiración para la ingeniería debido a su adaptabilidad y forma de resolver los problemas. En tal sentido, a mediados de los años ochenta, comienzan a cobrar relevancia diferentes investigaciones en el campo. Así, se conforma lo que se denomina “una nueva tendencia en robótica” [18]. Esta tendencia se posiciona entre la planificación de alto nivel de la IA (Inteligencia Artificial) deliberativa y el control de bajo nivel, a la vez que toma inspiración en la naturaleza y pone especial acento en comportamiento y reacción rápida y no en conocimiento y planificación [19]

En los últimos años se ha pasado de crear modelos a partir de la manipulación de fenómenos físicos, para emular conductas biológicas moldeadas con la intención de obtener de ellas el mayor beneficio, creando sistemas que tienen la capacidad de aprender. [20]

En el campo de la robótica existen numerosas aplicaciones que pueden resolverse utilizando un sistema sensorial integrado y un lazo de realimentación de los estímulos percibidos. En definitiva, el futuro del control inteligente de robots está en dotar a los dispositivos robóticos antropomórficos de capacidades de actuación basadas en los modelos que se han estudiado a partir de experimentos psicofísicos sobre el comportamiento biológico en los humanos. Para ello es necesario conocer y aplicar los estudios sobre coordinación senso-motora y sobre integración de los canales sensoriales. [10]

Es claro el avance en los últimos años relacionado con los sensores y la tecnología de los sistemas de control, sin embargo una de las grandes barreras en la implementación efectiva de la fusión multisensorial sigue siendo, el intento de combinar datos precisos obtenidos en los sensores con variantes desconocidas de nuestro sistemas, señales de propagación y ambientes de ruido desafiantes. [3]

Con el objetivo de diseñar esquemas de control que permitan dotar de una cierta inteligencia a los sistemas robóticos, un número importante de investigaciones va dirigido a conseguir diseñar arquitecturas de control que posean las características del sistema humano en cuanto a capacidad de recepción de estímulos en tiempo real, capacidad de aprendizaje, adaptación a las variaciones del entorno, selección de comportamientos, y demás características de un entorno típico de la robótica de servicios. [10]

Otra parte de esta bioinspiración es el aprendizaje. Cuando se trata de estrategias de aprendizaje, podemos encontrar diferentes trabajos en la literatura donde el robot observa a un ser humano realizar una tarea y luego es capaz de realizar la tarea por sí mismo. Ejemplos de kinaesthetics, modelado y control de redes neuronales entre muchas otras. A pesar del considerable aumento del número de manos mecánicas de varios dedos disponibles en el mercado y un creciente interés paralelo en comprender cuestiones, hay pocas aplicaciones basadas en un control de comprensión biomimético (es decir, las estrategias de agarre de alto nivel inspiradas en la base neural de las habilidades del alcance humano). Una de las dificultades que surgen en el aprendizaje basado en el humano es como medir el desempeño humano. Por el contrario, hay muchos enfoques agarre que dependen totalmente de las limitaciones físicas y geométricas para la síntesis y / o estrategias de alcance. [6]

Hay trabajos donde se han propuesto una forma más sencilla de control, la cual esta bioinspirada en la mano humana. En estos trabajos se parte de que en nuestra mano humana no todos nuestras articulaciones son capaces de moverse de forma independiente, muchas veces los movimientos están unidos a los de las otras articulaciones a través de músculos o nervios. [17] [18]

Tradicionalmente, el software utilizado para controlar una mano robótica compleja siempre se ha enfocado en intentar tener en cuenta todos los grados de libertad de las articulaciones robotizadas, aunque esto resulta torpe a nivel informático y acaba ralentizando al robot. En

vez de eso, por medio de sinergías, se limita el movimiento de la mano del robot de la misma forma que la mano humana se ve limitada. Al unir las articulaciones de esta forma, es posible controlar una mano robótica compleja mediante el uso de algoritmos más eficientes y rápidos, sin perder por ello ningún grado de funcionalidad. Es posible aprender a partir de la biología a reducir los grados de libertad," Aunque tengas 20 grados de libertad, no necesitas usarlos todos". [22]

Esto es un claro ejemplo de que la bioinspiración es un camino claro para la resolución de problemas robóticos en todos los niveles: accionamiento, control, aprendizaje, etcétera; es una idea interesante integrar conceptos bioinspirados como esta "reducción de grados de libertad" [7].

Es necesario, una mejora del control de la cinemática y / o dinámica del movimiento. Los procesos de control bioinspirados podrían proporcionar nuevas respuestas a los problemas encontrados en robótica de alcance y agarre como lo son la optimización del costo computacional, tiempo de cálculo, necesidad de señales de retroalimentación, etcétera. [7]

La equivalencia motora es la habilidad de llevar a cabo una tarea utilizando diversos recursos motores. En la naturaleza, los animales y humanos, somos capaces de llevar a cabo una misma tarea mediante diferentes medios motores, un claro ejemplo las tareas de alcance y agarre de diferentes objetos. Por lo cual uno de los desafíos más interesantes en el campo de la neurociencia es el de comprender cómo un organismo vivo es capaz de llevar a cabo el control de sus miembros de forma flexible. [10]

A continuación explicaremos algunos sistemas bioinspirados, básicos a desarrollar en el presente trabajo, que son :

- los comportamientos reactivos: que están bioinspirados en los actos reflejo, en otras palabras, sistemas que reaccionan a cambios de su medio ambiente o a mensajes provenientes de otros agentes. No son capaces de razonar acerca de sus intenciones. Sus acciones se realizan como resultado de reglas que se disparan o de la ejecución de planes.
- Redes Neuronales Artificiales: emulan a las Redes Neuronales biológicas, basados en la eficiencia de los procesos llevados a cabo por el cerebro, e inspirados en su funcionamiento

- El paradigma AER (AddressEventRepresentation): que imitan la forma en que el sistema nervioso transmite la información y puede llevar a cabo procesamiento de los datos al mismo tiempo.

## Comportamientos Reactivos

La robótica autónoma constituye un área muy particular en el vasto campo de la robótica. En esta línea, puede afirmarse que para que un robot sea autónomo es necesario que el mismo sea capaz de reaccionar ante situaciones que no han sido consideradas en la programación de su control y sin ningún tipo de supervisión proveniente del exterior. [11]

Una nueva alternativa al paradigma modelado-planificación es el propuesto por Rodney Brooks (1986), el cual no se centra en la fusión de sensores, sino en la noción de fusión de conductas. Esta arquitectura provee una forma de combinar el control en tiempo-real distribuido con conductas disparadas por sensores. Las conductas son capas de sistemas de control que corren en paralelo cuando los sensores apropiados son disparados. Las conductas de más alto nivel tienen el poder de temporalmente suprimir a las conductas de más bajo nivel.[21]

Brooks fue uno de los líderes iniciadores en el paradigma de la robótica puramente reactiva. Su equipo empujó en enfoque con el desarrollo de la arquitectura Subsumption. El ha articulado el rumbo de la inteligencia artificial (IA) clásica y roto la vía del paradigma sensor-planear-actuar que dominó la IA de la década de los 70's.[19]

El enfoque reactivo se basa en modelos biológicos para explicar el comportamiento observado en distintos organismos vivos. Los sistemas reactivos se definen como sistemas de eventos discretos que mantienen una interacción constante con su entorno, reaccionan continuamente a los estímulos externos e internos [22].

El Control Reactivo es una estrategia basada en conductas que acoplainformación sensorial tiempo real con acciones motrices , sin la intervención de representaciones simbólicas para el modelado parcial o total del ambiente operativo de un robot. El control reactivo se ajusta bien en los ambientes dinámicos. [23]

A diferencia de los sistemas puramente transformacionales, que asignan entradas a salidas, los sistemas reactivos de comportamiento basado en eventos de exposiciones, se caracterizan

porque el tiempo de respuesta es muy rápido debido a que no utiliza un modelo del mundo para ejecutar las acciones del robot. [11] Los sensores del robot proveen la información necesaria para generar algún tipo de movimiento. La fusión de los diferentes tipos de sensores determina la acción a realizarse sobre los actuadores y por lo tanto determinará el comportamiento final del robot.

Este tipo de control permite una reacción ante los estímulos procedentes de los sensores en tiempo real. No exige un procesamiento complejo de la información sensorial ya que la fusión se produce en este caso en el nivel de los comportamientos y no de los sensores.

El control reactivo se adapta perfectamente a la programación de minirobots ya que son equipos que disponen de una limitada capacidad de procesamiento y almacenamiento. Además de que en muchos casos no es preciso un modelo del mundo para actuar en él con un comportamiento inteligente. Se basa en el concepto de comportamiento, que proporciona la relación de los pares estímulo-respuesta para un determinado estado del entorno, haciendo que de la interacción de diferentes comportamientos con el entorno emerja la inteligencia necesaria para realización de determinadas tareas. La actividad sensorial proporciona la información necesaria para satisfacer una respuesta a bajo nivel de un reflejo motor. [19]

## **Arquitecturas de Control Reactivas**

Las arquitecturas que siguen este tipo de comportamiento, a veces llamadas “puramente reactivas”, implementan estrategias de control como una colección de pares “condiciones – acciones” y no llevan a cabo procesos de planificación ya que no operan sobre un modelo interno del ambiente. [11]

Las acciones del robot se basan en un acoplamiento directo entre sensores y actuadores mediante bucles rápidos de realimentación. La respuesta a estímulos es reflexiva, no regulada por procesos de carácter deliberativos de ningún tipo y los movimientos del robot se guían sólo a partir de la información que se está presente en ese momento en los sensores.

## Principales Características

- Gran velocidad de respuesta; aunque esta está se ve limitada si el robot debe abordar tareas que requieren planificación.
- Alta velocidad en el procesamiento de información.
- Los modelos de procesamiento de información “masivamente paralelos” son la base de estas arquitecturas, como por ejemplo, las redes neuronales artificiales (RNA).
- La información percibida por el robot no se integra en ningún modelo del ambiente. El sistema percibe el mundo y ejecuta la acción que le parezca más adecuada.
- Como desventaja, la principal es falta de planificación del sistema entero.



**Figura 1: Arquitectura de control basada en el enfoque reactivo puro**

## Enfoque basado en comportamientos

Son una evolución del enfoque reactivo puro. Los diferentes módulos se configuran en forma simultánea y coordinada; como resultado son arquitecturas cognitivas con mayor grado de interacción entre los módulos.

Los comportamientos describen la forma en que reacciona el sistema ante un determinado estado de los sensores; mientras que los actuadores determinan las acciones a seguir por parte del robot de una manera un poco más elaborada que la consulta a una simple tabla de decisión “sensor – actuador”. Asimismo, estas arquitecturas funcionan en base a un conjunto de comportamientos paralelos ejecutándose en forma concurrente y donde cada uno de los distintos módulos de comportamiento se vinculan en forma directa con los sensores y actúan directamente sobre los actuadores. Algunos ejemplos de comportamientos serían: buscar alimentos, cerrar pinzas o evitar obstáculos. [23]

## **Características principales [20] [24]:**

- Las funciones de sensorización y de actuación de un sistema robot están vinculadas directamente mediante comportamientos.
- Su desarrollo es más rápido que las arquitecturas pertenecientes al enfoque basado en conocimiento, ya que requiere menos procesamiento de la información.
- Tienen capacidad para tratar situaciones no previstas, ya que depositan mayor confianza en el mundo como fuente de información y de determinación de sus acciones.
- Aún con la simplicidad del enfoque reactivo, estas arquitecturas escalan en complejidad buscando incorporar otras habilidades en los robots, tales como memoria, aprendizaje y comunicación.
- Dado que no intentan mantener representaciones objetivas del ambiente tienen menos dificultades de representación de los estímulos sensoriales.

## Inteligencia Artificial: Redes Neuronales

Las técnicas utilizadas en la fusión de alto nivel, también llamada inteligencia artificial, son las redes neuronales artificiales (NN, por sus siglas en inglés: Neural Networks) y técnicas de lógica difusa.[ia2]

La inteligencia artificial, ha venido desplazando sistemáticamente hacia lo que se conoce como nueva inteligencia artificial (Coppin, 2004), inteligencia computacional (Engelbrecht, 2007) o inteligencia artificial bio-inspirada (Floreano y Mattiussi, 2008), tres formas de referir la búsqueda de soluciones “inteligentes” mediante modelos inspirados biológicamente (por células, neuronas, respuesta inmune, evolución...).[ia1]

Los métodos de inteligencia artificial son inferencias de alto nivel que parten de un razonamiento humano, tales como el reconocimiento de patrones, la planificación, la inducción, la deducción y el aprendizaje. (ia3)

Las técnicas bioinspiradas de fusión poseen capacidades de extracción de información, la representación del conocimiento y la generación, la máquina de aprendizaje, la inferencia, y el procesamiento del lenguaje [19].

Empezaremos con definir que son las Redes Neuronales, su base biológica y estudiaremos métodos neuronales que se han propuesto para tareas de alcance y agarre y que pueden ser de utilidad para un robot de servicios; en la fusión multisensorial también es común usar este método para reconocimiento de imágenes, y creación de mapas de navegación en robots móviles, pero estas no se han incluido en este trabajo.

### Redes Neuronales

Las redes neuronales, son sistemas bioinspirados en el sistema nervioso humano que intentan imitar la forma en que el cerebro humano adquiere el conocimiento por medio del proceso de aprendizaje y donde la fuerza con que están interconectadas las neuronas (pesos sinápticos) se utilizan para almacenar la información. [rn1]

A diferencia de la mayoría de los sistemas que operan principalmente de forma secuencial, el sistema nervioso es masivamente paralelo. De hecho, el sistema nervioso humano contiene alrededor de cien mil millones de neuronas interconectadas entre sí. En una analogía al cerebro, una entidad compuesta de neuronas interconectadas, las redes neurales se componen de elementos de procesamiento interconectados llamados unidades, que responden en paralelo a un conjunto de señales de entrada dadas a cada uno. La unidad es el equivalente a su homólogo cerebro, la neurona.(rn4)

Una de sus principales funciones es procesar la información sensorial, sin embargo, el encargado de llevar el proceso automático es el bulbo raquídeo. Se estima que el cerebro humano contiene más de cien mil millones ( $10^{11}$ ) de neuronas y  $10^{15}$  sinapsis en el sistema nervioso. Los estudios realizados sobre la anatomía del cerebro humano concluyen que hay, en general, más de 1000 sinapsis por término medio a la entrada y a la salida de cada neurona.[rn3]

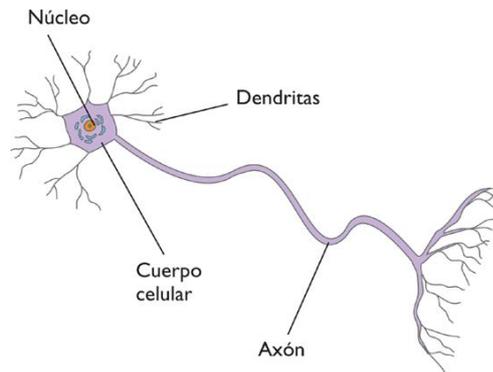
Aunque el tiempo de conmutación de las neuronas biológicas (unos pocos milisegundos) es casi un millón de veces mayor que en los actuales componentes de los ordenadores, las neuronas naturales tienen una conectividad miles de veces superior a la de las artificiales. Algunos científicos afirman que si un cerebro puede realizar un mayor cantidad de sinapsis puede desarrollar una mayor inteligencia que uno con menor desarrollo neuronal. [rn4].

Por ello las neuronas y las conexiones entre ellas (sinapsis) representan la clave para el procesado de la información. Pero ¿qué es una neurona?

Las neuronas son un tipo de células del sistema nervioso que tienen como característica básica la excitabilidad de su membrana plástica, su principal función es la recepción de estímulos y la conducción del impulso nervioso (en forma de potencial de acción) entre ellas o con otro tipo de células. [rn5].

Las neuronas constan de 3 partes (Figura 2):

1. El cuerpo de la neurona, contiene un núcleo controla las actividades de toda la célula
2. Las dendritas, reciben las entradas de los axones de otras células nerviosas, también se ramifican o extienden del cuerpo de las células.
3. El axón, se encarga de transmitir la salida de la neurona a las dendritas de otras neuronas.



**Figura 2**

Cada célula nerviosa está conectada a otras miles de células a través de sus axones y dendritas. La información es transmitida por medio de diferencias de potencial eléctrico. Las neuronas se encuentran organizadas en redes y sistemas. La comunicación entre ellas se realiza a través de contactos funcionales altamente especializados denominados sinapsis[[rn7](#)]es un espacio virtual y en la que distinguimos:

- Zona presináptica: integrada por los botones terminales del axón
- Hendidura sináptica, que es el espacio virtual
- Zona postsináptica: formada por las espinas dendríticas de la neurona siguiente.

Los botones terminales del axón están repletos de vesículas en donde se almacenan neurotransmisores. Al llegar el impulso nervioso a los botones terminales provoca que las vesículas viertan sus neurotransmisores en la hendidura sináptica. Los neurotransmisores se dirigen a la zona postsináptica permitiendo que el impulso nervioso pase a la siguiente neurona.

Las sinapsis pueden ser eléctricas o químicas utilizando sustancias transportadoras (neurotransmisores). Las de tipo eléctrico las observamos en especies inferiores. En la especie humana, las neuronas se excitan unas a otras mediante señales químicas que transmiten la información nerviosa a través de los denominados neurotransmisores.[[rn8](#)]

El primer modelo de red neuronal fue propuesto en 1943 por McCulloch y Pitts[[Figura 3](#)][rn2](#)] en términos de un modelo computacional de actividad nerviosa. Este modelo era un modelo binario, donde cada neurona tenía un escalón o umbral prefijado, y sirvió de base para los modelos posteriores.

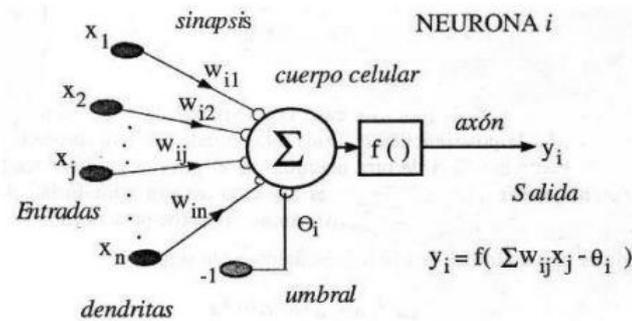


Figura 3

Las redes neuronales tienen un gran atractivo para muchos investigadores debido a su gran cercanía a la estructura del cerebro, una característica no compartida por los sistemas más tradicionales, el cual recibe la información de los receptores (ojos, oídos, nariz y nervios del tacto) y hace un cierto análisis de esta información para componer una reacción.

Aunque a menudo se considera un fenómeno relativamente nuevo, los intentos iniciales de las redes neuronales artificiales se remontan a principios del siglo 20. Aunque no fue hasta 1985, que el Instituto Americano de Física fue sede de la primera reunión anual de Redes Neuronales para la Computación y en 1987 el Instituto de Ingenieros Eléctricos ( IEEE) primera Conferencia Internacional sobre Redes Neuronales eléctricas . Este interés ha prevalecido hasta nuestros días y las redes neuronales artificiales ya han encontrado usos en todo, desde equipos de diagnóstico médico con el software de reconocimiento de voz . nnn1

Las redes neuronales presentan ventajas respecto a sistemas de computación clásicos, una de ellas es su capacidad de aprendizaje. Una red neuronal puede aprender y cambiar sus coeficientes sin necesidad de ser reprogramada. Además puede aplicarse a cualquier problema lineal o no lineal. Sin embargo presenta la desventaja que un gran tiempo de cálculo si existen muchas neuronas en la red. [7]

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada y este puede ser supervisado o no supervisado. En el aprendizaje supervisado se introducen a la red unos patrones de entrenamiento, que consisten en unas entradas ejemplo a la red y sus salidas correspondientes. La red entonces, con sus coeficientes iniciales producirá unas salidas, después evaluará el error cometido respecto a las salidas ejemplo del patrón de entrenamiento y modificará un poco los coeficientes de conexión entre

neuronas. Esta variación se hace según un algoritmo matemático llamado de gradiente descendente. [7]

La topología o arquitectura de las redes neuronales consiste en la organización y disposición de las neuronas en la red formando capas o agrupaciones de neuronas más o menos alejadas de la entrada y salida de la red. [7] Según esta característica las redes neuronales se clasifican en:

- Redes monocapa. Tienen una sola capa o nivel de neuronas.
- Redes multicapa. Tienen múltiples capas neuronales.

Además del tipo de aprendizaje y de su topología, existen 2 aspectos más que caracterizan a una red neuronal artificial [7]: tipo de asociación y forma de representación de la información.

Tipo de asociación entre capas de neuronas: Las redes neuronales son sistemas que almacenan cierta información aprendida; esta información se registra de forma distribuida en los pesos asociados a las conexiones entre neuronas. Comportándose la red como una memoria asociativa. Existen dos maneras de hacer esta asociación:

- Heteroasociación: La red aprende parejas de datos  $[(A_1, B_1), (A_2, B_2), \dots, (A_n, B_n)]$ , de forma que cuando se presente cierta información de entrada  $A_i$ , deberá responder generando la correspondiente salida asociada,  $B_i$ .
- Autoasociación: La red aprende ciertas informaciones:  $A_1, A_2, \dots, A_n$ . De forma que cuando se presenta una información a la entrada, realizando una autocorrelación, respondiendo con uno de los datos almacenados, el más parecido a la entrada.

Forma de representación de la información. Las redes se clasifican en:

- Redes continuas: La representación de la entrada es analógica, mientras que la información de salida es binaria.
- Redes híbridas: La información de entrada es analógica y la información de salida es binaria.
- Redes discretas: Tanto la información de entrada como la de salida es binaria.

## **NN Usadas en tareas de Alcance y Agarre**

### **Red neuronal de Kohonen [7]**

Es utilizada como primer paso en la tarea de Alcance. Las redes de Kohonen [26] son redes monocapa, de aprendizaje no supervisado. La idea principal de esta Red Neuronal es dividir el

espacio de trabajo del brazo virtual en regiones en las cuales las configuraciones articulares del brazo son similares para posteriormente, una red RBF calcule la cinemática inversa para posicionar la mano virtual en un punto determinado.

### **Red neuronal de base radial: [7]**

Sirve para el cálculo de la configuración articular del hombro y el codo, cinemática inversa, para posicionar la muñeca en las inmediaciones de la región que se trate. En cada región, a partir de la configuración articular aprendida de forma totalmente no supervisada, se procederá a alcanzar el objeto con pequeños incrementos a partir de esa configuración local.

Aprenden la configuración aproximada de la postura de la mano para el agarre. Dado un objeto situado en una posición y con una orientación determinada, calcula la preforma de la mano de manera aproximada, antes de contactar con el objeto.

Calcula la forma que deberá tener la mano para contactar con el objeto y posibilitar el agarre del mismo.

Broomhead y Lowe, 1988 [27], introducen las redes de base radial, redes RBF, para hacer ajustes a funciones no lineales, mediante una clasificación no lineal. Estas redes basadas en funciones de base radial construyen sus modelos con funciones de activación que son diferentes tanto en la capa oculta como la capa de salida. Esto es, una red RBF está diseñada con neuronas en la capa oculta activadas mediante funciones radiales de carácter no lineal con sus centros gravitacionales propios y en la capa de salida mediante funciones lineales. En general, una red RBF tiene un mejor desempeño con un mayor volumen de datos de entrenamiento. Una red RBF está formada por tres capas.

1. La capa de entrada que sirve para los ejemplos o patrones de entrenamiento y prueba.
2. La capa oculta completamente interconectada entre todos sus nodos con la capa de entrada y activada a través de la función radial (gaussiana).
3. La capa de salida, también completamente interconectada a la capa oculta y activada a través de una función lineal continua.

**Modelos neuronales de control senso-motor para la representación interna, generación de trayectorias y alcance de objetivos: VITE - *Vector Integration to Endpoint*- [Bullock 1989], AVITE [Gaudio 1991] y *DIRECT -Direction to Rotation Effector Control of Trajectories-* [Bullock 1993a] y [Guenther 1992]. [10]**

Estas redes están interconectadas para formar un sistema senso-motor, estos modelos describen en gran detalle el problema del alcance mediante una colección de redes neuronales cada una de las cuales realiza una subtarea, motivada por datos anatómicos, fisiológicos o psicológicos permitiendo el diseño de una arquitectura global, integrada y multisensorial (percepciones de visión y tacto, principalmente) para operaciones de alcance y agarre de precisión en entornos dinámicos:

- Representación neuronal de un objetivo percibido visualmente dentro de un sistema de coordenadas centrado en la cabeza, [Greve 1993].
- Transformación de esta representación a un sistema de referencia fijo en el cuerpo, [Guenther 1992] y [Grossberg 1993].
- Formación de una trayectoria espacial desde la posición actual de la mano hasta el objetivo, [Bullock 1993a].
- Transformación de esta trayectoria espacial en órdenes a los actuadores de brazo (cinemática inversa), [Bullock 1993a].
- Generación de comandos a las articulaciones de los brazos para mantener su posición a pesar de los cambios en la tensión de los músculos (dinámica inversa), [Bullock 1992a y 1992b].

**Modelo de aprendizaje direccional para el control de la redundancia postural basado en la red HRBF - *Hyperplane Radial Basis Function*- [Guenther 1997].**

La precisión requerida para el proceso de agarre centrado del objeto en el proceso de coordinación cutáneo-motora, puede ser obtenida mediante el diseño e implementación de una red neuronal no lineal adaptativa que optimice el aprendizaje de la información proporcionada por cada maniobra de agarre. son una modificación de las redes RBF -Radial Basis Function- Además permite el control adaptativo de la relajación postural de las articulaciones del sistema robot de alcance y agarre para resolver el problema de la equivalencia motora. [10]

### **Redes Neuronales Pulsantes**

Las Redes Neuronales Pulsantes (SNN por sus siglas en inglés: Spiking Neuronal Network) forman parte de una ciencia emergente conocida como Neurociencia Computacional, también son consideradas las RN de tercera generación y están basadas en el modelo de neurona pulsante de Hodgkin-Huxley (1952) [26].

En las redes neuronales pulsantes, las señales son transmitidas a través de potenciales de acción. La información es representada como patrones de actividad neuronal o pulsos, lo que crea una diferencia significativa entre las redes neuronales pulsantes y las redes neuronales clásicas. Desde que las redes neuronales pulsantes están basadas en los modelos de neuronas pulsantes, que son muy semejantes al modelo biológico de la neurona, muchos de los principios encontrados en la neurociencia biológica pueden ser usados en las redes. Los mecanismos de aprendizaje para este tipo de redes pueden ser aplicados para explicar el comportamiento de las redes en el cerebro y también pueden ser aplicados a sistemas de inteligencia artificial para procesar información compleja representada por estímulos biológicos. [27]

Las funcionalidades de las neuronas pulsantes pueden ser aplicadas para tratar con estímulos biológicos y explicar el comportamiento del cerebro. Una característica diferente de las redes neuronales pulsantes es que la información es codificada en patrones de actividad neuronal y las neuronas se comunican usando trenes de pulsos en lugar de valores individuales como sucede en las primeras dos generaciones. Las redes neuronales pulsantes trabajan con una gran cantidad de neuronas. Como una red de este tipo, a gran escala, requiere grandes recursos computacionales para ser simulada el modelo "Integrate-and-fire" y el modelo "Spike response" son usualmente considerados como una simplificación del modelo de Hodgkin-Huxley. Desde que los modelos de neuronas pulsantes son utilizados y la información es codificada usando los patrones de actividades neuronales, los mecanismos de aprendizaje para las redes neuronales son muy diferentes de cómo eran en las primeras dos generaciones de redes neuronales clásicas.[nnp1]

Es interesante la aplicación de la tecnología pulsante en redes neuronales, ya que comparte muchos fundamentos bioinspirados con el paradigma AER. El cual estudiaremos a continuación.

## El paradigma AER (Address–EventRepresentation)

### Introducción

La naturaleza siempre ha sido fuente de inspiración para la ingeniería debido a su adaptabilidad y forma de resolver los problemas. En tal sentido, a mediados de los años ochenta, comienzan a cobrar relevancia diferentes investigaciones en el campo. Así, se conforma lo que se denomina “una nueva tendencia en robótica”. Esta tendencia se posiciona entre la planificación de alto nivel de la IA (Inteligencia Artificial) deliberativa y el control de bajo nivel, a la vez que toma inspiración en la naturaleza y pone especial acento en comportamiento y reacción rápida y no en conocimiento y planificación. [18]

En los últimos años se ha pasado de crear modelos a partir de la manipulación de fenómenos físicos, para emular conductas biológicas moldeadas con la intención de obtener de ellas el mayor beneficio [22], un ejemplo de esta bioinspiración es el paradigma AER; el cual busca imitar la comunicación neuromórfica: Transmitir la información de manera similar a como lo harían los sistemas neuronales biológicos. El término neuromórfico fue ideado por Carver Mead del California Institute of Technology de 1989 para describir sistemas VLSI que contienen circuitos electrónicos analógicos que imitan las arquitecturas neurobiológicas presentes en el sistema nervioso. [25]

Inspirado en la naturaleza, una variable analógica que se pasa al paradigma AER, adquiere un valor representado por una frecuencia de pulso proporcional al mismo. Cuando se trata de un bus AER, cada vez que se manda un pulso (evento) se envía también una dirección, que nos indica de que variable o elemento se trata, y el tiempo en que se repite la dirección (evento), que muestra el valor de dicha variable o elemento. Por tanto a través de un mismo bus AER se pueden transmitir variables de diferentes elementos (multiplexación). [17]

El AER (Address-Event-Representation) es una técnica de adquisición sensorial y procesamiento bioinspirado basado en la codificación y transmisión de la información como un flujo continuo de pulsos eléctricos (eventos), de forma similar a como los sistemas neurobiológicos envían y procesan impulsos nerviosos. [25]

Este sistema, transmite la información modulada en frecuencia, de tal forma que la duración entre pulsos (spikes) es proporcional al valor de la señal a transmitir, obteniendo una inmunidad al ruido adecuada. El receptor de un determinado mensaje debe recoger los pulsos y reconstruir la señal original.

Un pulso o disparo es el proceso por el cual se activa una célula nerviosa, cuando existe un potencial eléctrico en las neuronas o a los nodos de una red hasta el punto de superar cierto umbral, se dice que la célula se puede disparar permitiéndole propagar el potencial eléctrico.

Se definen como "trenes de pulso" a patrones de disparos con secuencias de tres o más pulsos que se presentan sucesivamente en el tiempo, separados entre sí por tiempos interpulsos.

Los trenes de pulsos son los medios básicos de comunicación en el sistema nervioso.

En neurociencia nos encontramos con uno de los principales problemas llamado "romper el código neuronal". Debido a la gran cantidad de modelos e interpretaciones para los datos este se convierte en uno de los problemas más difíciles de resolver. Consiste en comprender exactamente cómo es que la manera en que se activa una neurona depende del número, la frecuencia, la correlación de los diferentes trenes de pulsos que llegan ella.

Las principales ventajas de la codificación pulsante son la simplicidad; baja cantidad de canales de comunicación necesarios; información continua en el tiempo, aunque solo se transmiten los spikes cuando son necesarios, evitando saturar los canales de comunicación; y alta inmunidad al ruido.

## **Sistema AER**

AER (Address–EventRepresentation) es un mecanismo por el cual celdas de computación de distintos chips se comunican asíncronamente entre ellas. Se entiende por celdas de computación a unidades computacionales, como puede ser una neurona de silicio, un píxel de una matriz o retina...[17]

Este paradigma nace en los laboratorios de Caltechentorno a 1991 como un simple protocolo asíncrono de comunicación inter-chip para replicar de manera continua en el tiempo el estado de un array de neuronas de un chip en otro chip. Cada neurona (o píxel) codifica su estado como la frecuencia de una secuencia de impulsos (eventos). Los impulsos generados

asíncronamente por un array de píxeles se arbitran y transmiten a un bus digital inter-chip de alta velocidad. [25]

Utilizando una codificación binaria para representar direcciones y eventos. Cada palabra codifica la dirección del nodo de envío. El uso más sencillo del AER consiste en usarlo para transmitir información, de forma que un receptor reconstruye la información del emisor sin cambios. [26]

## **BUS AER**

En el bus inter-chip AER cada evento asíncrono se transmite en pocos nano segundos (15ns en los chips más rápidos reportados actualmente). Los eventos generados por todos los píxeles se multiplexan en el tiempo en un bus digital. Además de permitir la comunicación de muchas neuronas, otra ventaja del AER es que se optimiza el ancho de banda y potencia de comunicación, ya que píxeles inactivos o poco activos apenas consumen ancho de banda y potencia de comunicación. Solamente los píxeles activos (con información relevante) transmiten pulsos. Así mismo, los píxeles más activos (más relevantes) envían su información antes. [26]

## **Estado del arte**

Los primeros trabajos sobre AER se centran en sensores AER. En general, los sensores AER se dotan de algún pre-procesamiento de la información para así concentrarse en información relevante. Posteriormente, surgió la idea de explotar el modo de eventos AER no sólo para comunicación interchip sino en toda la cadena de sensado, procesamiento y actuación tal como ocurren en los sistemas neuromotores biológicos. Recientemente, la incorporación de aprendizaje STDP en un robot con sistema de control adaptativo en modo del pulsos ha sido demostrada exitosamente. [25]

Actualmente se han desarrollado sistemas de control y procesamiento de señales basados en modelos neuronales pulsantes que han permitido controlar motores usando información visual y sonora.

Para procesar la información AER existen diversos modelos computacionales y elementos que los implementan. Un modelo muy usado es el modelo Integrate&Fire, del que podemos encontrar chips neuromórficos que contienen capas de neuronas implementando este modelo. También encontramos sistemas que realizan convoluciones basadas en pulsos o

eventos AER, existiendo una gran variedad de convolucionadores AER, ya sean analógicos o digitales. Además se pueden encontrar nuevos sistemas empotrados de procesamiento AER, aprovechando la potencia del co-diseño hardware/software de sistemas AER, así como propuestas de redes híbridas, en las que se proponen nuevas arquitecturas de redes celulares para integrarlas con sistemas AER. Como aplicaciones particulares de los convolucionadores, se han presentado dispositivos para el filtrado de visión AER. Otros modelos muy extendidos son, el de la sinapsis probabilística, mediante la que los spikes son procesados mediante el uso de técnicas probabilísticas, y el modelo Winner-Take-All, en la que sólo la neurona más activa de toda una capa dispara spikes, implementando de esta manera modelos atencionales. Al igual que los seres vivos tienen capacidad de aprendizaje, también existen implementaciones basadas en la representación AER pulsante capaces de ser entrenadas para aprender patrones, encontrando diversos modelos e implementaciones en circuitos AVLSI, así como propuestas para el aprendizaje de robots. Estos sistemas se han utilizado para las más diversas aplicaciones, como por ejemplo son el reconocimiento de voz, la eco-localización, la categorización de escenas, reconocimientos faciales y de caracteres. Además podemos encontrar una gran variedad de ejemplos en los que estos sistemas han sido combinados y fusionados para ser integrados en plataformas robóticas. El diseño, implementación, e integración de sistemas neuro-inspirados no sería posible sino no existieran herramientas especializadas para la depuración y el desarrollo de sistemas AER. Las llamadas AER-Tools [Paz05][Gomez06], son una serie de elementos que permiten la depuración e interfaz con un ordenador personal de los sistemas AER [Paz05][Paz06] [Berner07], para poder visualizar, reproducir, y analizar la información AER que circula por un sistema multi-chip neuro-inspirado basado en la representación AER. [27]

## **Bloques de Control**

aplicar los modelos de las neuronas pulsantes al control de elementos motrices de las plataformas robóticas e implementarlos en circuitos lógicos programables, como son las FPGA [Maxfield04].

Para verificar el comportamiento tanto de los componentes VHDL, como de su interacción con sistemas dinámicos (como son los motores de DC), hemos recurrido al simulador Simulink [SIMULINK], incluido en MATLAB [MATLAB], junto con una extensión de Xilinx para integrar componentes VHDL en Simulink, llamado Xilinx SystemGenerator [SYSGEN]. Para cada uno de los mecanismos de actuación sobre motores de DC basados en spikes, expuestos con anterioridad, se han desarrollado diversas simulaciones en diversos escenarios. En ellas podremos apreciar los diferentes comportamientos que muestran los motores de DC ante los

diversos parámetros configurables de estos mecanismos. Así su correlación aproximada con las ecuaciones teóricas. [27]

### **SpikeGenerator (Generador de spikes)**

Para poder simular los elementos de control que iremos exponiendo a lo largo de este trabajo, necesitamos un elemento capaz de proporcionarnos secuencias de spikes, en base a números discretos aplicados a su entrada, con el fin de excitar a todo el sistema. Para ello diseñaremos un generador sintético de spikes, de tal manera que la frecuencia de los spikes de salida sea proporcional al valor discreto proporcionado a su entrada; tal y como se muestra en la siguiente ecuación, donde  $x$  representa el valor de entrada, y  $k_{FreqModulation}$  la constante de modulación en frecuencia:

$$f(x)_{spikes} = k_{FreqModulation} * x$$

Existen diversas propuestas de generadores de spikes sintéticos basados en FPGAs, y orientados generalmente a la generación de imágenes [Linares02] [Linares03] [Gomez05]: 'Scan', 'Uniform', 'Random', 'Random-Square' y 'Exhaustive'. Dado que para aplicaciones de control necesitamos métodos que distribuyan de manera homogénea los spikes en el tiempo, nos vamos a centrar en los métodos exhaustivos [Gomez05][Linares02]. Estos métodos dividen el tiempo en frames, estando cada frame dividido a su vez por slices, los cuales duran un ciclo de reloj. De esta manera si disparamos los spikes en los slices adecuados obtendremos una distribución homogénea de los mismos. [27]

### **Control de Motores**

El control de motores es un problema que está ampliamente resuelto hoy en día usando sistemas digitales, analógicos o mezcla de ambos. Pero resulta interesante este tipo de trabajos para poder observar las características que adquiere un sistema con el paradigma AER y sobretodo en vista de hacer el control de un sistema robótico.

### **Control por Frecuencia y Amplitud**

La modulación por frecuencia de pulsos, PFM, resulta especialmente atractiva, ya que coincide con la empleada por los spikes y las neuronas pulsantes en la naturaleza. Una señal modulada mediante PFM sigue un esquema exactamente opuesto a una señal PWM. En la modulación PFM el tiempo en alto es siempre constante, pero la frecuencia, o separación temporal entre pulsos, es variable. [25]

El uso de la modulación PWM introduce un integrador, el cual integra a lo largo del período de la señal PWM. Es decir, se introduce un retraso desde que llega la información en spikes y se transmite al motor la señal PWM, equivalente a  $T_{pwm}$ . Sin embargo, introducir un integrador transforma una secuencia de spikes a un número discreto (valor integrado), y con un período de muestreo conocido (período de la señal PWM). Entre el valor integrado y el modulador PWM sería posible introducir una función de transferencia en el dominio Z, e implementar en su interior reguladores en Z, como por ejemplo un PID discreto. [27]

Sin embargo, usando la modulación PFM los spikes son aplicados al motor casi sin retraso alguno, eliminando la latencia introducida por el método PWM. Pero en su contra, al no tener período de muestreo (ni fijo, ni siquiera conocido), la implementación de controles basados en PFM no están extendidos, y pasan por el uso de otro tipo de técnicas, como pueden ser las probabilísticas. [25]

### **Modelo Inter-Spike-Interval Difference & Generate**

Este modelo se basa en medir el período de los spikes de entrada, procesándolos para obtener el período de los nuevos spikes de salida como un número discreto, y finalmente aplicando dicho número a un generador de spikes sintético exhaustivo bit-wise). Este modelo tiene un alto coste hardware, ya que hacen falta implementar divisores y multiplicaciones de un elevado número de bits.[27]

### **Modelo Hold&Fire.**

Manteniendo la filosofía de los modelos neuronales pulsantes, intentando solventar todos los problemas expuestos anteriormente que presentaba el modelo anterior, surge el modelo Hold&Fire[27]. Este modelo se basa en la retención (Hold) de los spikes, esperando a la evolución de las entradas, disparando (Fire) o anulando spikes, en base a la naturaleza de los spikes que se reciban. Es decir, cada vez que llega un nuevo spike este es retenido internamente, esperando a la evolución de las entradas durante un tiempo fijo, tiempo de hold. Si no llega ningún nuevo spike después del tiempo de hold, el spike retenido es disparado. Sin embargo, si llegan nuevos spikes, el spike retenido puede ser disparado, reteniendo un nuevo spike, o cancelado, sin producir ninguna salida y sin retener ningún spike. En definitiva, pensando en una realimentación negativa, la idea es que según vayan llegando los spikes se vayan cancelando mutuamente, produciendo así la substracción de las tasas de spikes de ambos puertos de entrada, y ofreciendo una salida con la tasa de spikes adecuada. [27]



## Conclusiones

La naturaleza siempre ha sido fuente de inspiración para la ingeniería debido a su adaptabilidad y forma de resolver los problemas. En tal sentido, a mediados de los años ochenta, comienzan a cobrar relevancia diferentes investigaciones en el campo. Así, se conforma lo que se denomina “una nueva tendencia en robótica”. Esta tendencia se posiciona entre la planificación de alto nivel de la IA (Inteligencia Artificial) deliberativa y el control de bajo nivel, a la vez que toma inspiración en la naturaleza y pone especial acento en comportamiento y reacción rápida y no en conocimiento y planificación.

Un ejemplo de sistema bioinspirado es el paradigma AER, Address-Event-Representation(AER), el cual es un mecanismo por el cual celdas de computación de distintos chips (unidades computacionales) se comunican asincrónicamente entre ellas.

Este sistema, transmite la información modulada en frecuencia, de tal forma que la duración entre pulsos (spikes) es proporcional al valor de la señal a transmitir, obteniendo una inmunidad al ruido adecuada. El receptor de un determinado mensaje debe recoger los pulsos y reconstruir la señal original.

En el bus AER es posible diferenciar entre las diferentes celdas, asignando a ellas una dirección propia, de tal forma que se puede establecer una comunicación punto a punto y permite un procesamiento extra de las señales mientras se transmiten las direcciones.

Las principales ventajas de la codificación pulsante son la simplicidad; baja cantidad de canales de comunicación necesarios; información continua en el tiempo, aunque solo se transmiten los spikes cuando son necesarios, evitando saturar los canales de comunicación; y alta inmunidad al ruido.

Mediante este paradigma se han desarrollado bloques de control implementado en VHDL de las operaciones básicas, pero para que se realicen bajo el paradigma AER. De esta manera se ha logrado hacer un control de motores por medio de esta tecnología utilizando la plataforma AER-Robot, que permite mandar y recibir datos desde el ordenador, usando Matlab.

Como se ha visto a lo largo de todo este capítulo, las técnicas bioinspiradas nos aportan

Como se ha dicho en el capítulo anterior, lo de paradigma aer para control d robots, para lo cual es necesario usar los bloques de control en lazo cerrado. Muchos de estos bloques nacieron

del sensado con AER, específicamente las retinas aer y el procesamiento de imágenes. Como los métodos para procesar las imágenes.

trabajos previos de tarjetas de control de motores, yo he comprobado el funcionamiento.

entornos dinámicos y la respuesta de

nn aprendizaje, posiciones, sinergias, movimiento, reconocimiento de patrones, etc etc

controlar un sistema con muchos grados de libertad, velocidad, paralelismo, inmunidad al ruido. Aer, se han hecho sistemas robóticos que procesan información de sensado visual para controlar el movimiento de los motores a altas velocidades, con una alta inmunidad al ruido y al cambio de las condiciones del entorno.

Interesante para pensar en un futuro y por eso se ha elegido como el método para la aplicación práctica del presente trabajo, ya que intentar implementar cualquiera de los otros dos era demasiado ambicioso y se hubiera tenido que dejar en simulación.

importancia de hacer una arquitectura conveniente a nuestros algoritmos

El método elegido para esta puesta practica es el paradigma AER.

## Capítulo 5 Implementación práctica: control de motores.

Se ha elegido comprobar el control de motores mediante el paradigma en AER

Se comprobará el control de motores mediante el paradigma AER, utilizando la plataforma AER-Robot para mover el cabezal Lince

Dado que la principal ventaja del paradigma AER es la alta velocidad, se ha hecho una comprobación del sistema AER utilizado para mover el cabezal Lince con la plataforma AER-Robot. El cual se explica a continuación.

### La Plataforma AER-Robot

La plataforma AER-Robot ha sido diseñada conceptualmente como un puente entre motores de DC, y el bus AER. La idea de esta plataforma es recibir información AER y actuar en consecuencia sobre motores de DC, además de sensarlos y enviar el valor de los sensores como información AER. Desde el punto de vista de una red neuronal, la plataforma AER-Robot representaría una capa de entrada/salida de una red AER, donde la entrada son los sensores del robot y la propia información AER, y las salidas, la actuación sobre cada motor, así como la información sensorial codificada como eventos AER. [27]

La plataforma AER-Robot es completamente reprogramable y escalable (se pueden poner tantas AER-Robots como se necesiten), estando basada en una arquitectura de FPGA más microcontrolador. Cada plataforma puede actuar sobre 4 motores DC, sensando estos motores mediante sensores analógicos y encoders ópticos de cuadratura. [27]

La plataforma AER-Robot cuenta con 2 entradas para 2 fuentes de alimentación; una correspondiente a la etapa de control, la cual funciona con 6 volts; y la segunda para la etapa de potencia; en nuestro caso, para el Cabezal Lince, utilizaremos 13 volts (Figura 4)

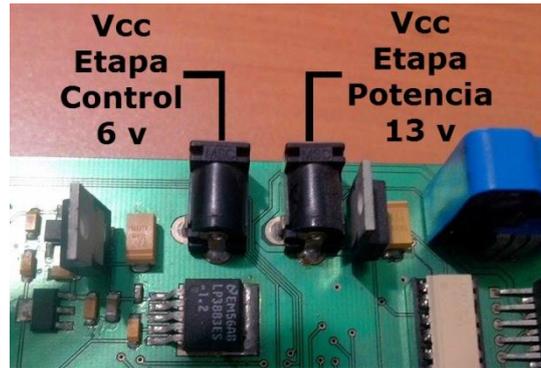


Figura 4: Entradas de alimentación de la plataforma AER-Robot

## Componentes principales

La plataforma AER-Robot tiene como principales componentes en la etapa de control: FPGA y microcontrolador. En la etapa de potencia podemos destacar la presencia de puentes H y Buffer LVC4245. A continuación se detalla las características de cada uno, así como su función y las diferentes configuraciones y herramientas necesarias.

## Etapa de Control

### FPGA

La plataforma AER-Robot cuenta realmente con 2 FPGA's, cuyos modelos son xc18v04 y la xc3s400. La primera (xc18v04), es una PROM para nuestro sistema de control. Por tal motivo solo se deberá programar cuando el programa creado está en su versión definitiva, ya que si se cargan diferentes programas continuamente se dañará.

Mientras tanto, para comprobar los programas que se van haciendo, se utiliza la segunda FPGA (xc3s400); por lo cual cada vez que se desconecte la fuente de voltaje de la plataforma AER-Robot y hasta que la primera FPGA tenga el programa definitivo, se borrará el programa habiendo que programarla siempre, de lo contrario no habrá ningún programa a ejecutar guardado en las FPGA.

Es necesario tener instalado en el ordenador el programa XILINXS ISE Design Suite para programar las FPGA se utiliza el Cable USB II de Xilinx el cual se conecta del puerto USB del ordenador a los pines pertinentes para programar la FPGA (interfaz FPGA/JTAG) en la plataforma AER-Robot.

Como se puede apreciar en el Anexo A, la interfaz FPGAJTAG de la plataforma AER-Robot tiene 6 pines de entrada. Por lo cual, al utilizar el Cable Usb II, que tiene 7 cables, no se usará el pin: HALT/INIT/WP, conectando según Figura 5: VREF (cable rojo), GND (cable negro), TCK (cable amarillo), TDO (cable lila), TDI (cable blanco) Y TMS (cable verde).

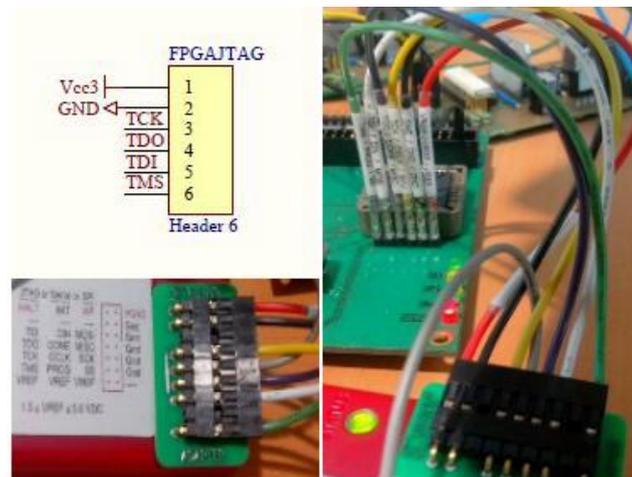


Figura 5

Lo normal es que el controlador se haya instalado automáticamente, pero de lo contrario se puede descargar gratuitamente en la siguiente dirección: <http://www.xilinx.com/products/design-tools/ise-design-suite/>.

El led indicador del estado del Cable USB II después de estar conectado, solo estará en verde, si el pin con el voltaje Vref está correctamente conectado.

## Microcontrolador

Pertenece a la familia de los 8051, es fabricado por SiliconLabs, y es el C8051F320 / 342 [C8051F320]. [27] La función principal del microcontrolador C8051F320 es la de brindarle a la plataforma AER-Robot conectividad por medio del puerto USB. De esta manera, será el puente para mandar la información entre la FPGA y el ordenador.

El C8051F320 funciona en sistemas operativos Windows 95/98/NT/ME/2000. Como actualmente la mayoría de los ordenadores tiene Windows 7, se ha creado una máquina virtual con el sistema operativo Windows 95; con esto se corrige cualquier problema de compatibilidad.



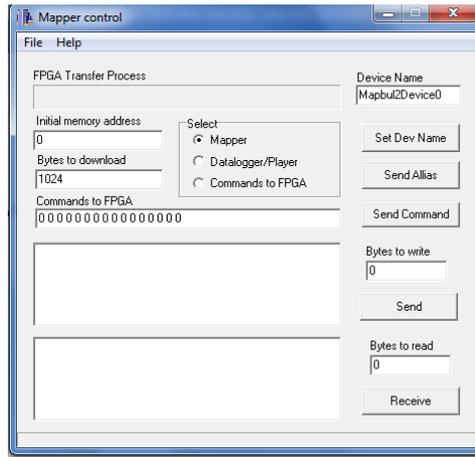


Figura 7

El devicename, debe coincidir con el que se usará en el programa Matlab:

```
Devname= ' Mapbul2Device0 '
```

## Etapa de Potencia

### Puente H

El puente H utilizado para controlar los motores DC es el L298KV. Y como se sabe, es el integrado electrónico que permite controlar los giros de los diferentes motores del cabezal.

### Buffer

LVC4245 Actúa como una barrera para proteger el puente h. Se utiliza para transferir una tensión de un primer circuito, que tiene un nivel de salida de alta impedancia, a un segundo circuito con un nivel de entrada de baja impedancia. El buffer impide que el segundo circuito cargue demasiado al primero, provocando un funcionamiento incorrecto. En un buffer ideal, la resistencia de entrada es infinita y la resistencia de salida es 0.

En un buffer de voltaje, la ganancia suele ser 1, el voltaje no varía. [28]

## Control del Cabezal Lince mediante la Plataforma AER-Robot

### Cabezal Lince

El cabezal Lince cuenta con tres grados de libertad, sin embargo para este trabajo solo se utilizaran dos de ellos: el movimiento horizontal y vertical en los ejes x,y.



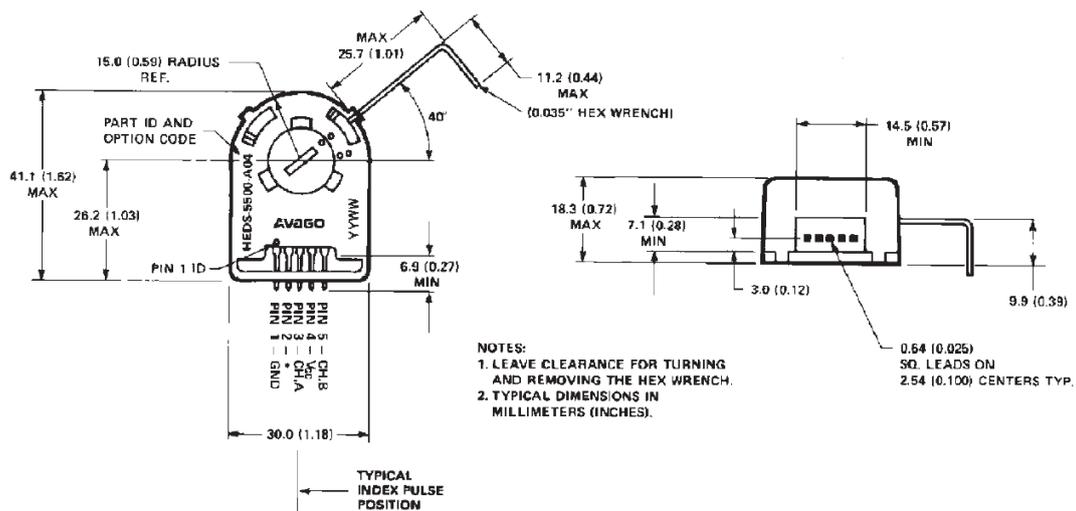
**Figura 8 Cabezal Lince**

Como interfaz para la tecnología AER se utiliza la plataforma AER-Robot , como ya hemos dicho se pueden controlar 4 motores. Para esta aplicación, se han utilizado 2 puentes h, con su respectivo buffer externos, ya que al hacer pruebas se sobre calentaron los puentes h incluidos en la tarjeta y se dañaron.\*\*\*



Figura 9 Plataforma AER-Robot y Puentes H auxiliares

HEDS-5500/5540, HEDM-5500/5540



\*Note: For the HEDS-5500 and HEDM-5500, Pin #2 is a No Connect. For the HEDS-5540 and HEDM-5540, Pin #2 is CH. I, the index output.

Figura 10 Especificaciones técnicas Encoder

**Absolute Maximum Ratings**

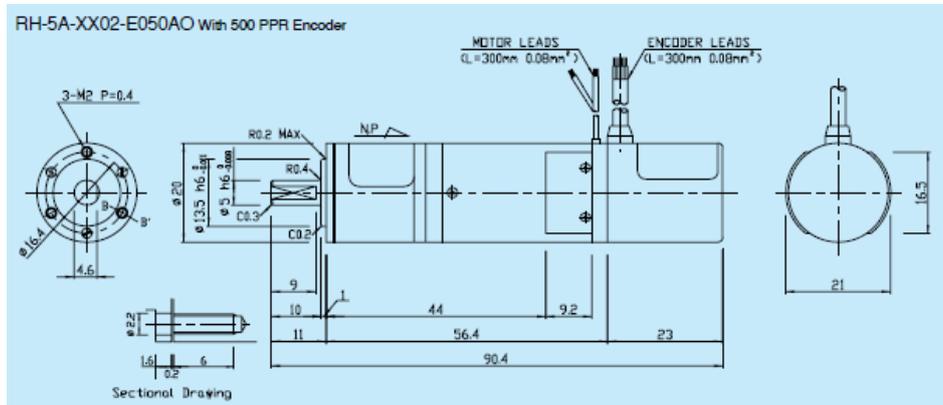
Parameter	HEDS-55XX/56XX
Storage Temperature, $T_S$	-40°C to 100°C
Operating Temperature, $T_A$	-40°C to 100°C
Supply Voltage, $V_{CC}$	-0.5 V to 7 V
Output Voltage, $V_O$	-0.5 V to $V_{CC}$
Output Current per Channel, $I_{OUT}$	-1.0 mA to 5 mA
Vibration	20 g, 5 to 1000 Hz
Shaft Axial Play	$\pm 0.25$ mm ( $\pm 0.010$ in.)
Shaft Eccentricity Plus Radial Play	0.1 mm (0.004 in.) TIR
Velocity	30,000 RPM
Acceleration	250,000 rad/sec <sup>2</sup>

**Recommended Operating Conditions**

Parameter	Sym.	Min.	Typ.	Max.	Units	Notes
Temperature HEDS Series	$T_A$	-40		100	°C	
Temperature HEDM Series						
5500/5600	$T_A$	-40		70	°C	non-condensing atmosphere
5540	$T_A$	-40		85	°C	
Supply Voltage	$V_{CC}$	4.5	5.0	5.5	Volts	Ripple < 100 mVp-p
Load Capacitance	$C_L$			100	pF	2.7 k $\Omega$ pull-up
Count Frequency	f			100	kHz	Velocity (rpm) x N/60
Shaft Perpendicularity Plus Axial Play (HEDS Series)				$\pm 0.25$ ( $\pm 0.010$ )	mm (in.)	6.9 mm (0.27 in.) from mounting surface
Shaft Eccentricity Plus Radial Play (HEDS Series)				0.04 (0.0015)	mm (in.) TIR	6.9 mm (0.27 in.) from mounting surface
Shaft Perpendicularity Plus Axial Play (HEDM Series)				$\pm 0.175$ ( $\pm 0.007$ )	mm (in.)	6.9 mm (0.27 in.) from mounting surface
Shaft Eccentricity Plus Radial Play (HEDM Series)				0.04 (0.0015)	mm (in.) TIR	6.9 mm (0.27 in.) from mounting surface

Note: The module performance is guaranteed to 100 kHz but can operate at higher frequencies. 2.7 k $\Omega$  pull-up resistors required for HEDS-5540/5640 and HEDM-5540.

**Figura 11 Características eléctricas Encoder**



Output voltage	3 V $\pm$ 10% / 1000 rpm
Ripple (RMS) <sup>1)</sup>	1% (200~5000 rpm)
Ripple (peak to peak) <sup>1)</sup>	3% (200~5000 rpm)
Linearity <sup>1)</sup>	1% (200~5000 rpm)
Direction deviation	1% (200~5000 rpm)
Temperature coefficient	< 0.02% / °C
Moment of Inertia <sup>2)</sup>	1.2 x 10 <sup>-6</sup>
Armature resistance	45 $\Omega$ $\pm$ 10% (at 20 °C)
Armature inductance	7 mH $\pm$ 20%
Maximum rotation speed	5000 rpm
Friction torque	< 4 x 10 <sup>-3</sup> Nm
Minimum load resistance	10 k $\Omega$
Insulation resistance	100 M $\Omega$ (DC 500 V Megger)
Breakdown voltage	AC 500 V / 1 minute
Weight	0.08 kg

Please note:

<sup>1)</sup> This value refers to the tach-generator only.

<sup>2)</sup> This value is referred to the motor shaft. For the moment of inertia referred to the actuator output shaft multiply this value by the square of the reduction ratio.

## RH Mini Series DC Servo Actuators

### Technical Data

Rating:	Continuous	Vibration:	2.5 g (5 ... 400 Hz)
Excitation device:	RE Permanent magnet	Shock:	< 30 g (11 ms)
Insulation:	Class B	Construction:	Totally enclosed
Insulation voltage:	AC 500V, one minute	Lubrication:	Grease (SK-2)
Insulation resistance:	100MΩ or more (DC 500V Megger)	Ambient temperature:	0 ~ 40°C
		Ambient humidity:	20 ~ 80% (non condensing)

Item	Actuator	RH-5A			RH-8D		RH-11D		RH-14D	
		8002	5502	4402	6006	3006	6001	3001	6002	3002
Rated Output Power <sup>1)</sup>	W	1.5	1.7	1.4	8.6	6.2	13.6	12.3	20.3	18.5
Rated Voltage <sup>1)</sup>	V	12	12	12	24	24	24	24	24	24
Rated Current <sup>1)</sup>	A	0.5	0.5	0.5	1.0	0.8	1.3	1.3	1.8	1.8
Rated Output Torque <sup>1)</sup> T <sub>N</sub>	in-lb	1.4	2.6	2.6	12	17	19	34	28	52
	Nm	0.16	0.29	0.29	1.4	2.0	2.2	3.9	3.2	5.9
Rated Output Speed <sup>1)</sup> n <sub>N</sub>	rpm	88	55	44	60	30	60	30	60	30
Max. Continuous Stall Torque <sup>1)2)</sup>	in-lb	2.1	3.5	3.8	13	20	22	39	48	69
	Nm	0.24	0.39	0.43	1.5	2.3	2.5	4.4	5.4	7.8
Peak Current <sup>1)2)</sup>	A	0.83	0.78	0.77	1.6	1.1	2.4	2.1	5.4	4.1
Maximum Output Torque <sup>1)2)</sup> T <sub>m</sub>	in-lb	3.5	5.2	6.1	24	31	43	69	122	174
	Nm	0.39	0.59	0.69	2.7	3.5	4.9	7.8	14	20
Maximum Output Speed <sup>1)</sup>	rpm	180	110	90	100	50	100	50	100	50
Torque Constant	in-lb/A	6	10	12	19	37	22	43	26	51
	Nm/A	0.69	1.11	1.38	2.10	4.20	2.46	4.91	2.92	5.76
Voltage Constant (B.E.M.F.)	V/rpm	0.06	0.12	0.15	0.22	0.44	0.26	0.50	0.30	0.60
Inertia at Output Shaft <sup>3)</sup>	in-lb-sec <sup>2</sup>	0.006	0.014	0.022	0.033	0.13	0.095	0.38	0.18	0.72
	kgm <sup>2</sup> x10 <sup>6</sup>	0.63	1.6	2.5	3.7	15.0	11.0	43.0	21.6	81.6
Mechanical Time Constant	msec	13.3	13.3	13.3	8.5	8.5	8.5	8.5	7.0	7.0
Rated Power Rate <sup>1)</sup>	kW/sec	0.039	0.055	0.034	0.51	0.26	0.43	0.36	0.51	0.42
Thermal Time Constant <sup>1)</sup>	min	5.2	5.2	5.2	9	9	10	10	11	11
Thermal Resistance <sup>1)</sup>	°C/W	11.4	11.4	11.4	4.2	4.2	3.3	3.3	2.8	2.8
Gear Ratio	1:R	1:50	1:80	1:100	50	100	50	100	50	100
Maximum Radial Load <sup>4)</sup>	lb	13	13	13	44	44	55	55	88	88
	N	59	59	59	196	196	245	245	392	392
Maximum Axial Load	lb	7	7	7	22	22	44	44	88	88
	N	29	29	29	98	98	196	196	392	392
Motor Rated Output <sup>1)5)</sup>	W	(2.6)	(2.6)	(2.6)	(10)	(10)	(20)	(20)	(30)	(30)
Motor Rated Speed <sup>1)</sup>	rpm	4500	4500	4500	3000	3000	3000	3000	3000	3000
Armature Resistance	Ω	8.6	8.6	8.6	10	10	4.7	4.7	2.7	2.7
Armature Inductance	mH	2.7	2.7	2.7	2.2	2.2	1.6	1.6	1.1	1.1
Electrical Time Constant	ms	0.31	0.31	0.31	0.22	0.22	0.34	0.34	0.41	0.41
No-Load Running Current <sup>4)</sup>	A	0.27	0.24	0.28	0.38	0.36	0.61	0.55	0.89	0.91
Actuator Accuracy	arc-min	4.5	4.5	4.5	2.5	2.5	2.0	2.0	2.0	2.0
Actuator Repeatability	arc-sec	±90	±90	±90	±60	±60	±60	±60	±60	±60
Servo Drive Combinations	DC 20V	DC:H55-09, DDP-090-09, DEP-090-09			-		-		-	
	DC 24V	-			DC:H55-09, DDP-090-09, DEP-090-09		DC:H55-09, DDP-090-09, DEP-090-09		DC:H55-09, DDP-090-09, DEP-090-09	
	AC 100V	HS-360-1A			HS-360-1B		HS-360-1C		HS-360-1D	

Table 2

#### Additional information

\* Actuator specifications show output characteristics, including gear efficiency.

\* All specifications are applicable for actuators mounted on aluminum heat sink of the following sizes:  
RH-5: 100 x 100 x 3 mm,  
RH-8, 11, 14: 150 x 150 x 6 mm.

#### Please Note:

<sup>1)</sup> The values are for saturated actuator temperature. Other values (not marked with <sup>1)</sup>) are for actuator temperature of 20°C.

<sup>2)</sup> The values given represent an upper limit and actual load values should be lower.

<sup>3)</sup> The tabulated value is the moment of inertia reflected to the output shaft resulting from the sum of the motor inertia and the gear inertia.

<sup>4)</sup> Values are for rated output speed.

<sup>5)</sup> Values are for reference only.

<sup>6)</sup> Cantilevered load applied at the midpoint of the shaft extension.

Figura 12 Especificaciones Servomotores Cabezal Lince

## Conexiones

Para este trabajo se ha utilizado una plataforma AER-Robot con 2 puentes h y buffer auxiliares extras, para lo cual se ha tenido que modificar el archivo .ucf del proyecto en XILINX, cambiando los pines que no se usan en los puentes h integrados en la plataforma-AER, por los auxiliares:

#MOTORES

NET "MOTOR\_RY<1>" LOC = "P165"; #0

NET "MOTOR\_RY<0>" LOC = "P166";

```
NET "MOTOR_LY<0>" LOC = "P167" ; #1  
NET "MOTOR_LY<1>" LOC = "P168" ;
```

#ANTIGUAS CONFIGURACIONES

```
#NET "MOTOR_RX<0>" LOC = "P169" ; #2  
#NET "MOTOR_RX<1>" LOC = "P171" ;  
#NET "MOTOR_LX<1>" LOC = "P172" ; #3  
#NET "MOTOR_LX<0>" LOC = "P175" ;
```

#MOTORES PUENTES H AUXILIARES (NUEVAS CONFIGURACIONES)

```
NET "MOTOR_RX<0>" LOC = "P148" ; #2  
NET "MOTOR_RX<1>" LOC = "P147" ;  
NET "MOTOR_LX<1>" LOC = "P123" ; #3  
NET "MOTOR_LX<0>" LOC = "P124" ;
```

Para revisar el resto de las conexiones, en el Anexo A se encuentra la hoja técnica de la plataforma AER-Robot elaborada por la Universidad de Sevilla.

## Recursos

Hasta este punto hemos explicado como configurar la máquina virtual con los programas necesarios, hemos programado el microcontrolador C8051F320; hemos realizado las diferentes conexiones entre el ordenador-Plataforma AER\_Robot y la plataforma AER\_Rbot-Cabezal Lince; tenemos listas nuestras correspondientes fuentes de voltaje. Así que ahora procederemos a configurar nuestra FPGA, para después empezar a mandar parámetros mediante Matlab a la plataforma AER-Robot para controlar finalmente los movimientos del cabezal Lince.

## VM-WARE

En el capítulo anterior explicamos los programas necesarios de instalar en nuestra máquina virtual y de que es necesario programar el C8051F320.

Para proseguir es necesario abrir la máquina virtual:



Figura 13

Una vez que ya estamos dentro de la máquina virtual de la figura 14 debemos verificar que tanto el C8051F320 como el Cable USB II estén conectados y sean vinculados con la máquina virtual.

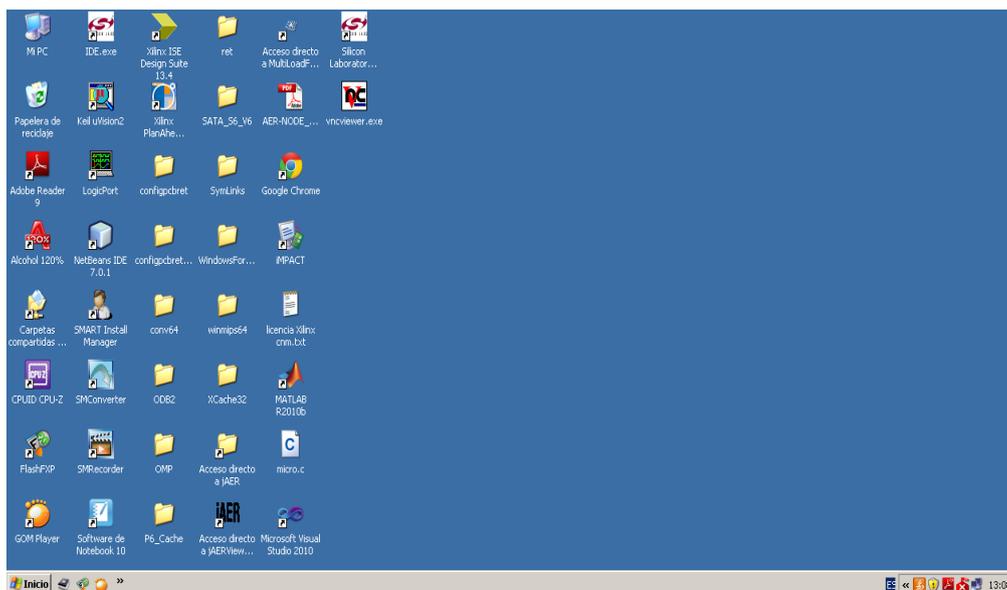


Figura 14

Para ello se siguen los pasos que se explican a continuación.

1. Verificar que el dispositivo C8051F320 es reconocido por el ordenador en general.  
(Figura 15)

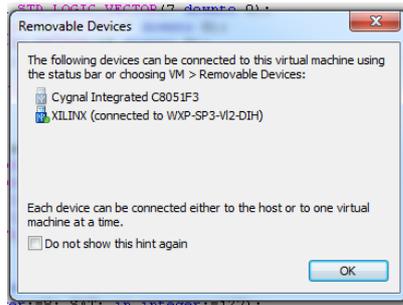


Figura 15

Si ya hemos hecho la configuración anteriormente, aparecerá automáticamente el dialogo de la Figura 16; al cual hay que dar en la opción OK.



Figura 16

Si no aparece el mensaje anterior, o conectamos posteriormente el dispositivo, se puede conectar de la siguiente manera:

1. Ir al menú de la máquina virtual: Player/Removable Devices/Cygnal Integrated C8051F320

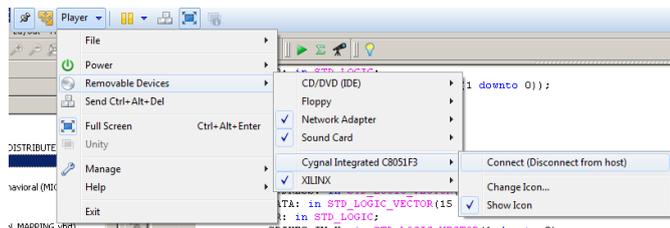


Figura 17

2. Dar OK al dialogo siguiente:



Figura 18

3. Finalmente verificar que aparezca una paloma que indique que está activado para la máquina virtual. Figura 19

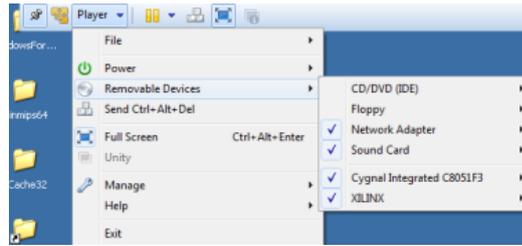


Figura 19

De igual forma se debe verificar que el dispositivo Xilinx esté conectado a nuestra máquina virtual.

Al conectarse en el ordenador la máquina virtual mostrará el mensaje para conectarlo a esta máquina y desconectar del host. **Figura 20.** Hay que dar aceptar.



Figura 20

En caso de que no salga este mensaje se puede verificar que el dispositivo este conectado en la máquina virtuellendo al menú superior de la máquina virtual: Player/RemovableDevices/XILINX/ Connect (Disconnectfrom host). (Figura 21)

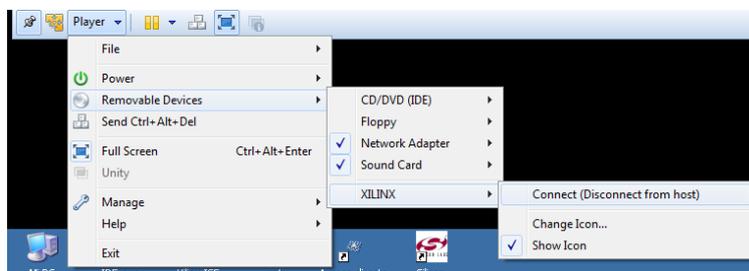


Figura 21

Ahora podemos proceder a programar la FPGA utilizando XILINX.

## Xilinx

Una vez que se haya cargado la máquina virtual, abrimos el programa Xilinx ISE Design Suite



13.4:

En este caso usaremos el proyecto desarrollado por la Universidad de Sevilla en lenguaje VHDL para el control de los motores de un cabezal con 6 grados de libertad.

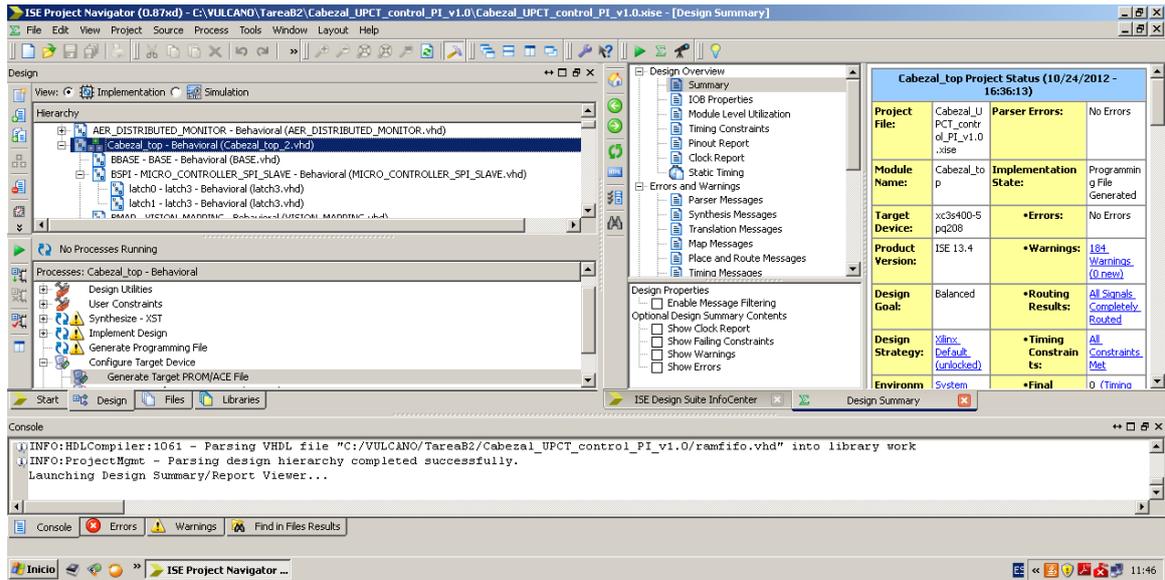


Figura 22

Para configurar los pines de entradas y salidas físicos de la plataforma AER-Robot para el programa en VHDL se da click en el fichero .ucf

En el Anexo A está las hojas de especificación de la Plataforma AER-Robot, para poder completar esta tarea. En la FIG. 23 se puede observar una parte del programa con terminación .ucf con los pines configurados .Deben coincidir con las conexiones reales en la plataforma AER-Robot.

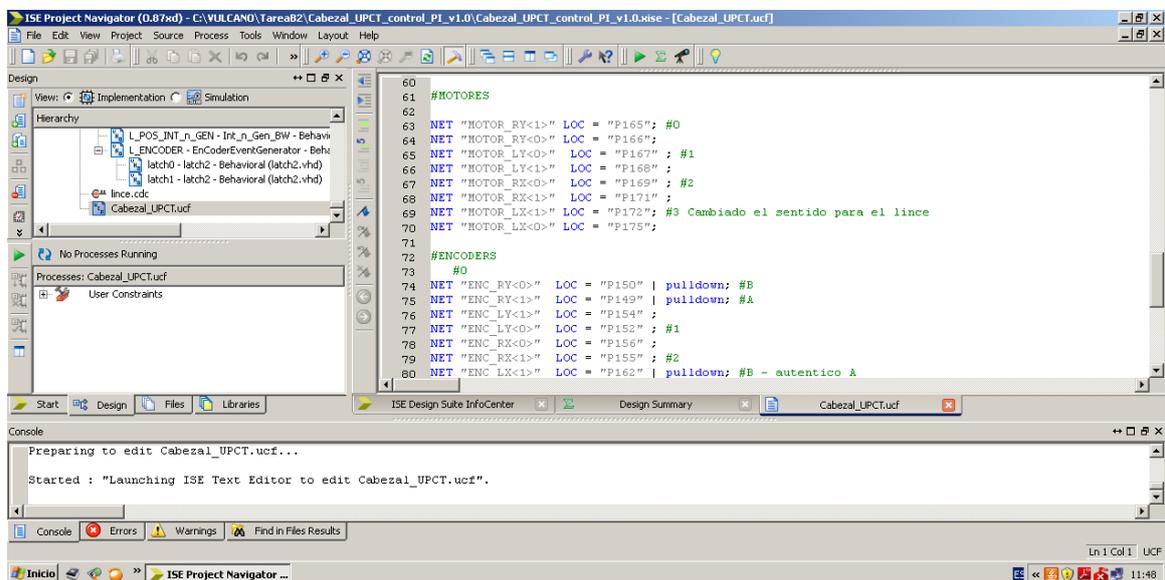


Figura 23

En el panel de Desig, encontraremos todos los archivos .VHD . Estos archivos corresponden a los Bloques de Construcción para el Procesamiento de Señales Spikes. Para cada motor existirá una serie de programas con los diferentes bloques necesarios para su control. En la ventana de la izquierda (FIG 24) se puede observar el editor de programas.

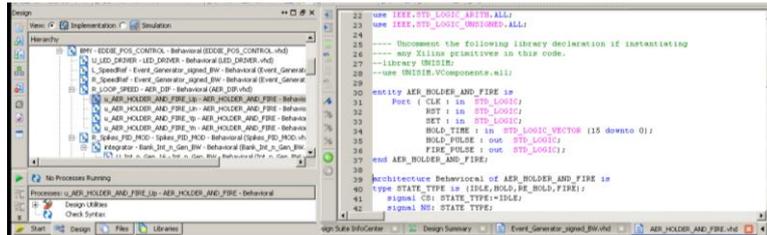


Figura 24

Para compilar el programa y crear el archivo que se cargará en la FPGA se debe verificar que en la pantalla de Design , este seleccionada la source donde están contenidos todos los archivos VHDL (Figura 25):

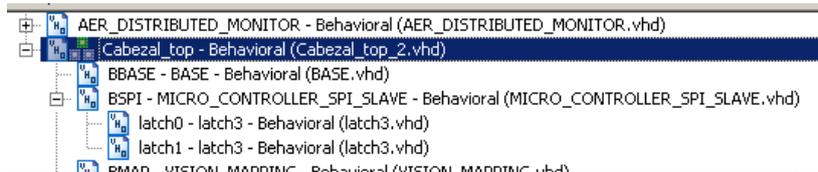


Figura 25

Después se deberá dar “Run” a los procesos Synthesize-XST; ImplementDesign y GenerateProgramming File:



Figura 26

Entonces se abrirá un diálogo que nos avisará que para seguir con el proceso hay que abrir el programa ISE iMPACT. Hay que dar Click en la opción OK. (Figura 27)

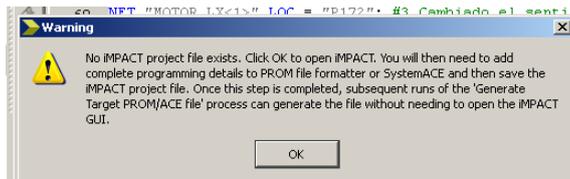


Figura 27

En la ventana de Flows se debe seleccionar BoundaryScan (figura 28), y después colocando el cursor en la pantalla derecha hay que dar click derecho del ratón para que salga el menú de la fig. 29. Se elige la opción: InitializeChain.

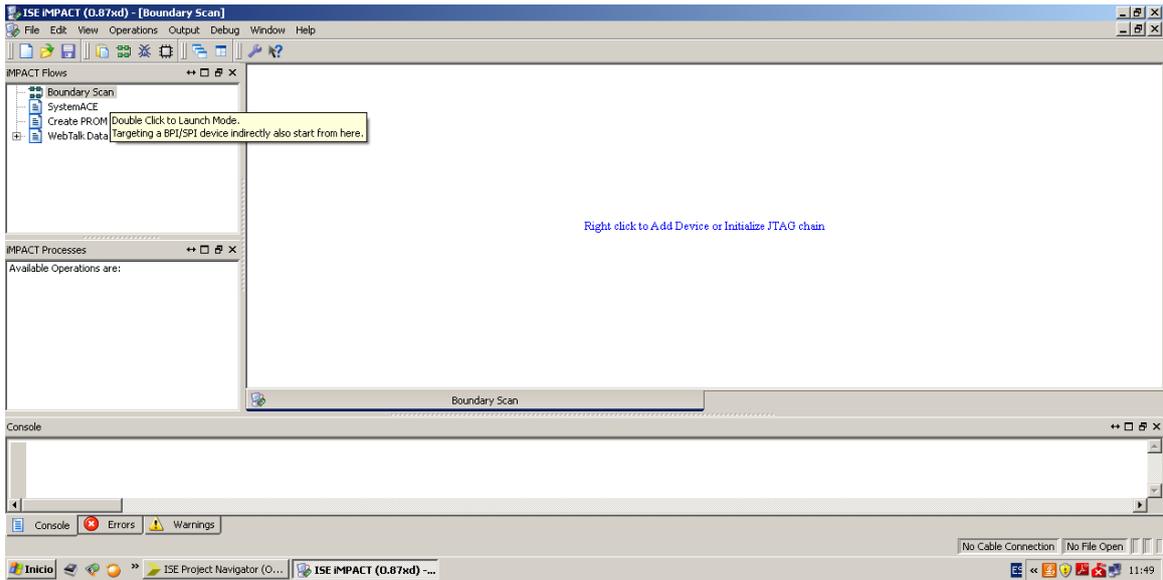


Figura 28

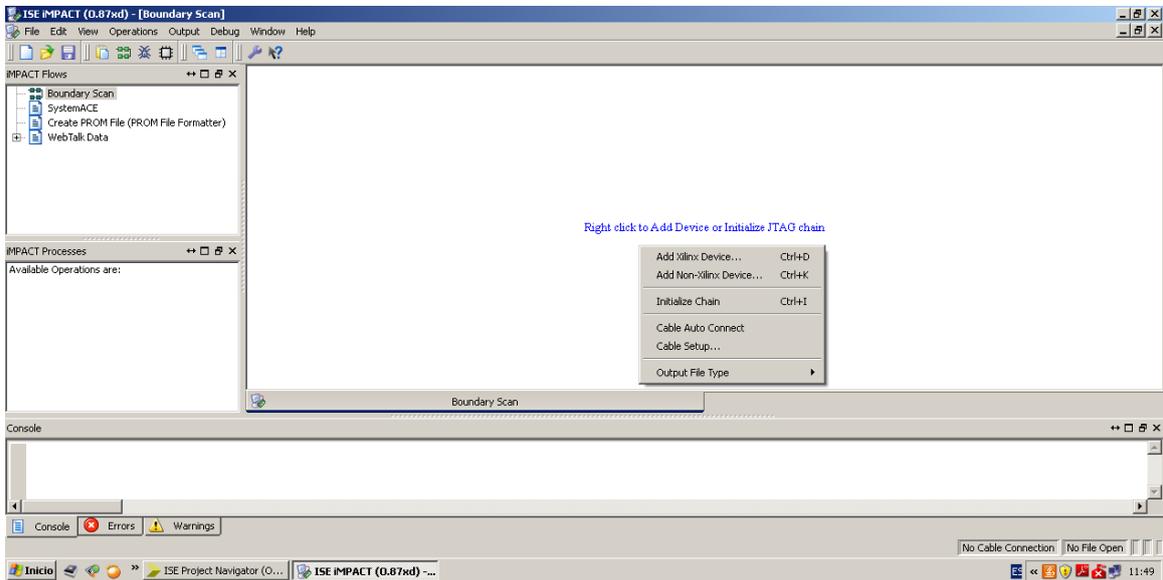


Figura 29

Entonces deberán aparecer las FPGA's de la tarjeta y aparece un cuadro de diálogo (figura 30), hay que seleccionar que no, porque cuando se cierre el diálogo, podemos seleccionar la segunda FPGA: xc3s400 que es la que queremos programar hasta que tengamos el programa definitivo (Figura 31).

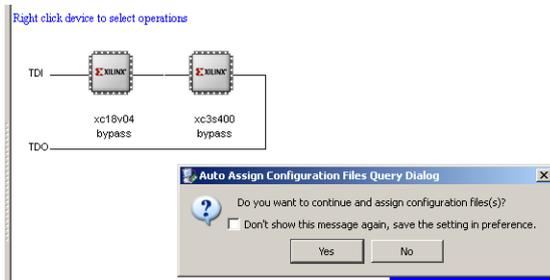
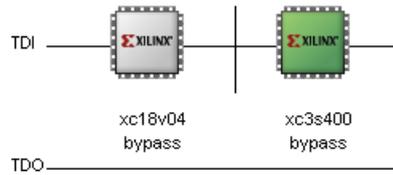
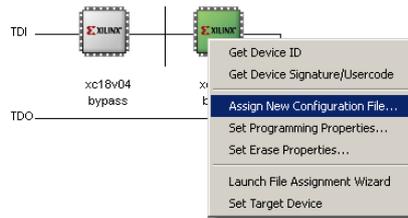


Figura 30



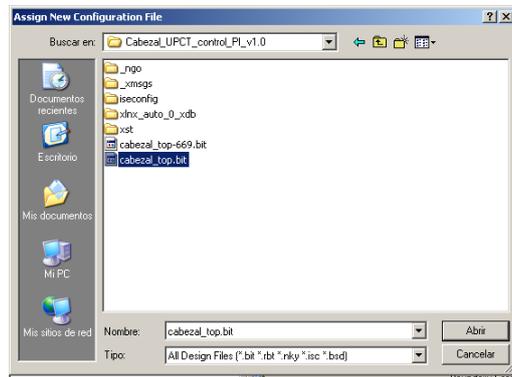
**Figura 31**

Para cargar el programa deberemos poner el cursor encima de la FPGA a programar y dar click derecho, para obtener el menú de la Figura 32 y elegir la opción: Assign New Configuration File:



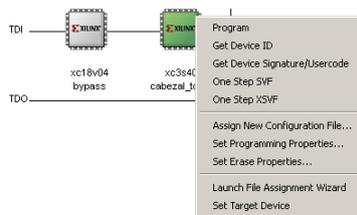
**Figura 32**

Nos desplegará un cuadro de dialogo donde deberemos elegir el fichero con la extensión .bit que creamos anteriormente y daremos click en abrir (Figura 33)



**Figura 33**

Ahora volveremos a dar click derecho encima del xc3s400, se desplegará un nuevo menú y elegiremos Program (Figura 34). Después se abrirá el dialogo de la fig. 35 al que daremos click en OK



**Figura 34**

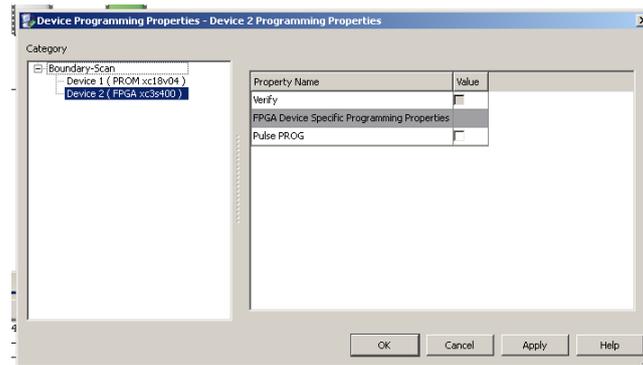


Figura 35

Debe salir el siguiente mensaje de programado con éxito. (Figura 36)

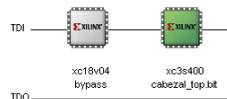


Figura 36

Una vez que nuestra FPGA está cargada procederemos a mandar los archivos de referencia por medio de Matlab. Recordemos que si apagamos la fuente de alimentación de la etapa de control será necesario volver a programar la FPGA. En caso de tener abierto el programa Xilinx solo será necesario dar otra vez la opción de programar, en caso contrario solo habrá que abrir Impact y hacer los pasos desde ahí.

## Matlab

Ahora abriremos Matlab. Desde Matlab podemos abrir y cerrar los diferentes dispositivos, por ejemplo si queremos hacer pruebas con los led de la plataforma AER-Robot, se utilizan instrucciones:

Para encender:

```
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 7]', 1)
```

Para apagar:

```
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 0]', 1)
```

Recordemos que devname es el que configuramos al principio (Figura 7).

AER\_ROBOT\_UPLOAD (Anexo B) es una función de Matlab, que sirve para abrir el dispositivo y establecer la comunicación por el puerto USB, así como mandar parámetros a los diferentes elementos controlados por a FPGA.

Para mover los motores, se debe correr el programa Init (Anexo B), con el cual se mandan los parámetros iniciales al cabezal. Hay que hacer pruebas para saber los parámetros óptimos para el motor con el que se está trabajando. Al principio esto se hace un poco arbitrariamente, pero una vez que se encuentra el parámetro se pueden hacer rutinas para mover los diferentes motores.

La rutina creada para mover el cabezal también se incluye en la parte final del Anexo B.

Esta plataforma se puede utilizar para hacer simulaciones de control por visión con spikes; pero esto no formo parte de este trabajo.

## Capítulo 6: Conclusiones y trabajos futuros

Para conseguir un progreso más significativo es preciso alcanzar niveles más altos de autonomía, lo que pasa necesariamente por disponer de sistemas de percepción más próximos al sistema de percepción humano para interpretar el entorno y la propia evolución de la tarea a realizar

Como se ha visto a lo largo de todo este capítulo, las técnicas bioinspiradas nos aportan Como se ha dicho en el capítulo anterior, lo de paradigma aer para control d robots, para lo cual es necesario usar los bloques de control en lazo cerrado. Muchos de estos bloques nacieron del sensado con AER, específicamente las retinas aer y el procesamiento de imágenes. Como los métodos para procesar las imágenes.

NN: aprendizaje, posiciones, sinergias, movimiento, reconocimiento de patrones, etc etc controlar un sistema con muchos grados de libertad, velocidad, paralelismo, inmunidad al ruido. AER, se han hecho sistemas robóticos que procesan información de sensado visual para controlar el movimiento de los motores a altas velocidades, con una alta inmunidad al ruido y al cambio de las condiciones del entorno.

Interesante para pensar en un futuro y por eso se ha elegido como el método para la aplicación práctica del presente trabajo, ya que intentar implementar cualquiera de los otros dos era demasiado ambicioso y se hubiera tenido que dejar en simulación.

importancia de hacer una arquitectura conveniente a nuestros algoritmos

Para conseguir un progreso más significativo es preciso alcanzar niveles más altos de autonomía, lo que pasa necesariamente por disponer de sistemas de percepción más próximos al sistema de percepción humano para interpretar el entorno y la propia evolución de la tarea a realizar

El trabajo desarrollado pretendía proporcionar una visión general de los retos existentes móvil que implica el paso de robótica industrial a servicios. No se ha profundizado en aspectos como la sensorica o los actuadores, porque estos ya están altamente desarrollados. Por ello, los grandes retos están en el área de control

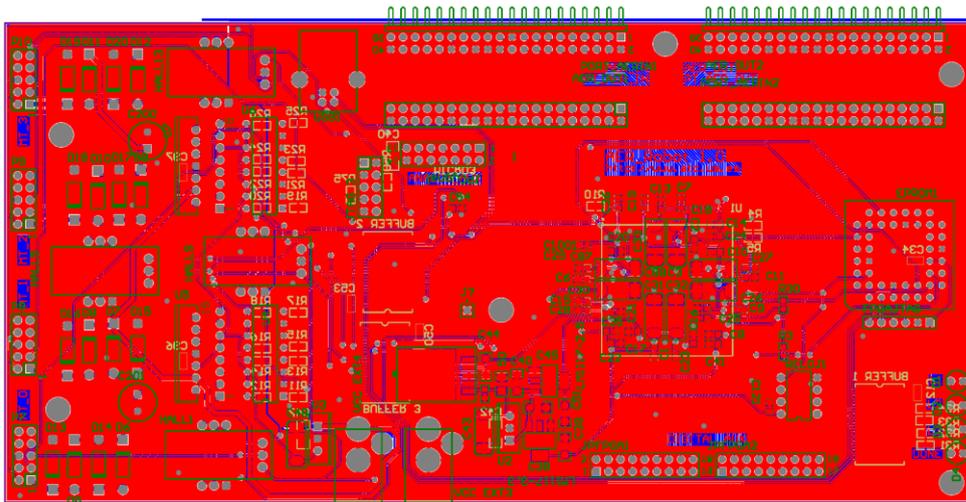
Dentro del control, existen técnicas utilizadas en la robótica móvil, que han solucionado problemas de percepción e inteligencia similares a los de la robótica de servicios. Estos son métodos de fusión sensorial y las técnicas bioinspiradas.

Por ello, el objetivo del trabajo se ha cumplido ya que se ha hecho un repaso por las características de estas técnicas, evaluando sio en el contexto de la robótica industrial, las complicaciones y ayudas que representa integrarlas en la arquitectura de control. Ahora es el momento de centrar los esfuerzos en vista a la tesis; de tomar estos elementos e intentar comprobar la hipótesis de la que parte este trabajo que el uso de sistemas bioinspirados y acciones reactivas en un sistema de control de alcance y agarre con fusión multisensorial, favorece las características de un sistema robótico que desarrolla estas tareas.

.Una vez se profundice de forma más efectiva en cada uno de esos temas se podrán ver las posibilidades existentes para la realización de una tesis doctoral

Está claro, que necesito adentrarme más en el mundo del diseño hardware y el VHDL, para poder aprovechar al máximo los beneficios de usar el paradigma AER. También se debe evaluar si es conveniente hacer todo el sistema en base a este paradigma o solo algunos algoritmos y partes de la arquitectura general. Y así con los otros elementos estudiados en el presente trabajo.

## Anexo A: Especificaciones técnicas de la Plataforma AER-Robot



## Anexo B: Funciones en Matlab

```
function AER_ROBOT_UPLOAD_CONFIG(alias,config,ec)
ifec==1
fprintf('\nAbriendo dispositivo %s\n',alias);
end
[micro,e]=usbaeropen(alias);

ifec==1
if (e==2)
fprintf('\tImposible abrir dispositivo\n');
return
end

fprintf('Leyendo configuración\n');
end
tmpconfig=[];
tmpconfig=config;
numBytes=size(tmpconfig);
ifec==1
fprintf('\tLeídos %d bytes\n',numBytes(1));
end
%for(i=1:1:64-mod(numBytes(1),64))
%   tmpconfig=[tmpconfig;hex2dec('8F')];
%end
numBytes=size(tmpconfig);
ifec==1
fprintf('\tEnviando configuración a %s\n',alias);
end
e=usbaersend(micro,tmpconfig(1:numBytes(1)));
ifec==1
if (e==0)
fprintf('\tEnviados un total de %d bytes\n',numBytes(1));
else
fprintf('ERROR: no se ha enviado la configuracion\n');
end
end
usbaerclose(micro);
ifec==1
fprintf('Cerrado dispositivo %s\n',alias);
end
```

### Init:

```
devname='Mapbul2Device0';
AER_ROBOT_UPLOAD_CONFIG(devname,[hex2dec('F0') 128 0] ',1)
%X axis motors
AER_ROBOT_UPLOAD_CONFIG(devname,[0 0 3 hex2dec('12') 2 0 128 0 4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname,[0 0 3 hex2dec('42') 2 0 128 0 4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname,[0 0 3 3 2 0 128 0 4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname,[0 0 3 4 2 0 128 0 4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname,[0 0 3 5 2 0 128 0 4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname,[0 0 3 6 2 0 128 0 4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname,[0 0 3 7 2 0 128 0 4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname,[0 0 3 hex2dec('30') 2 0 128 0 4] ',1)
```

```

AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 hex2dec('30')+1 2 0 128 0
4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 hex2dec('30')+2 2 0 128 0
4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 hex2dec('30')+3 2 0 128 0
4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 hex2dec('30')+4 2 0 128 0
4] ',1)
kk=0;
AER_ROBOT_UPLOAD_CONFIG(devname, [128+1 kk/256 mod(kk,256)] ',0)
AER_ROBOT_UPLOAD_CONFIG(devname, [128+2 kk/256 mod(kk,256)] ',0)
%Y axis motors
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 128+hex2dec('12') 2 0 128 0
4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 128+hex2dec('42') 2 0 128 0
4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 128+3 2 0 128 0 4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 128+4 2 0 128 0 4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 128+5 2 0 128 0 4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 128+6 2 0 128 0 4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 128+7 2 0 128 0 4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 128+hex2dec('30') 2 0 128 0
4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 128+hex2dec('30')+1 2 0 128 0
4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 128+hex2dec('30')+2 2 0 128 0
4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 128+hex2dec('30')+3 2 0 128 0
4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [0 0 3 128+hex2dec('30')+4 2 0 128 0
4] ',1)
AER_ROBOT_UPLOAD_CONFIG(devname, [1 kk/256 mod(kk,256)] ',0)
AER_ROBOT_UPLOAD_CONFIG(devname, [2 kk/256 mod(kk,256)] ',0)

```



## Bibliografía

- [1] D.L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6-23, Enero 1997.
- [2] (2013) IEEE Global History Network. [Online]. [http://www.ieeeahn.org/wiki/index.php/Joseph\\_Engelberger](http://www.ieeeahn.org/wiki/index.php/Joseph_Engelberger)
- [3] Tom Lozano-Pérez. Foreword, *Mobile Robots and Robotics*, Gordon Edition ed., i. j. and Wilfong, Ed.: Springer-Verlag, 1990.
- [4] George Bekey, Yuan Zheng Vijay Kumar, "INDUSTRIAL, PERSONAL, AND SERVICE ROBOTS," World Technology Evaluation Center (WTEC), International Research and Development in Robotics.
- [5] N. Daoud, J.-P. Gazeau, S. Zeghloul, M.A. Maier, S. Eskiizmirli, F. Touvet, "A biomimetic reach and grasp approach for mechanical hands," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 473-486.
- [6] Moreno Montero Ángeles Ma., *Procedimientos de obtención de configuración de agarre en manipulación robótica*. Salamanca, España: Tesis Doctoral. Universidad de Salamanca, 2013.
- [7] Marina Beltrán Blanco, "Estudio del empleo de redes neuronales artificiales para el alcance y agarre de objetos con un brazo y mano virtual antropomorfo."
- [8] Y. BARCOHEN, *Biomimetics-using nature to inspire human innovation, Bioinspiration & Biomimetics*, 1st ed. United Kingdom, 2006.
- [9] Javier Molina,.
- [10] Juan Luis Pedreño Molina, *ARQUITECTURA NEURONAL DE INSPIRACIÓN BIOLÓGICA PARA LA INTEGRACIÓN TÁCTIL EN SISTEMAS VISUO-MOTORES*. Cartagena, España: Tesis Doctoral. Universidad Politécnica de Cartagena, 2001.
- [11] A., Monte, G., Olivera, V, Hossian, "Análisis del Comportamiento de Robots Móviles con RNA. Un Acercamiento desde el Paradigma Reactivo," *Revista Latinoamericana de Ingeniería de Software*, pp. 18-24, Jan. 2013.
- [12] Fernández Peris César., *Aprendizaje mediante Árboles de Decisión para la Síntesis de Agarres Robóticos*..: PhD Thesis. Universidad Miguel Hernández, Enero 2005.
- [13] Antonio Bicchi and Vijay Kumar, "Robotic grasping and contact: a review," *IEEE*

- International Conference on Robotics and Automation*, vol. 1, p. 348\_353, 2000.
- [14] Mark R. Cutkosky, "Mechanical properties for the grasp of a robotic hand," Robotics Institute, Carnegie Mellon University, Technical Report CMU-RI-TR-84-24, Septiembre 1984.
- [15] Van-Duc Nguyen, "The synthesis of stable force-closure grasps," MIT Artificial Intelligence Laboratory, Technical Report AI-TR-905, 1986.
- [16] Zhixing Xue, J.M. Zoellner, and R. Dillmann, "Automatic optimal grasp planning based on found contact points," in *Advanced Intelligent Mechatronics. IEEE/ASME International Conference*, 2-5 July 2008, pp. 1053,1058.
- [17] Peter K. Allen Hao Dang, "Semantic grasping: planning task-specific stable robotic grasp," *Springer Science+Business Media New York*, Mayo 2014.
- [18] R.C. Luo and M.G. Kay, "Multisensor integration and fusion in intelligent systems," *Systems, Man and Cybernetics, IEEE Transactions*, vol. 19, no. 5, pp. 901-931, Octubre 1989.
- [19] Klein. L. A., "Sensor and data fusion concepts and applications.," *SPIE Optical Engineering Press*, pp. 14-131, 1993.
- [20] H.B. Mitchell., "Multi-Sensor Data Fusion: an introduction.," *Springer*, 2007.
- [21] Wald L., "Some terms of reference in data fusion. ," *IEEE Transactions on Geosciences and Remote Sensing*, vol. 37, no. 3, pp. 1190–1193, 1999.
- [22] N. Sharkey, "The New Wave in Robot Learning," *Robotics and Autonomous System*, vol. 22, pp. 179-185, 1991.
- [23] R. Hernández, "comparación e implementación de técnicas de control reactivo de robots móviles utilizando sensores de rango (sonar e infrarrojos)," PFC-UMH 2005, 2005.
- [24] N Sharkey, "The New Wave in Robot Learning," *Robotics and Autonomous System*, vol. 22, pp. 179-185, 1991.
- [25] Alejandro Linares Barranco, "Estudio y Evaluación de Interfaces para conexión de Sistemas Neuromórficos mediante Address-Event-Representation.," Sevilla, 2003.
- [26] M. Flanders and J. F. Soetching M. Santello, "Postural hand synergies for tool use," *J. Neuroscience*, vol. 18, no. 23, pp. 10105-10115, Diciembre 1998.
- [27] Corey Goldfeder, Peter Allen Matei Ciocarlie, "Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem," *Robotics: Science and Systems Manipulation Workshop-Sensing and Adapting to the Real World*, 2007.

- [28] Rodney. Brooks, "A. Robust Layered Control System For A Mobile Robot ," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 1, Marzo 1986.
- [29] D. Harel and A. Pnueli., ""On the development of reactive systems, Logics and models of concurrent systems", " *Springer-Verlag New York, Inc.*, vol. 13, pp. 477-498, 1989.
- [30] Panos E. Trahanias. Amalia F. Foka, "Predictive autonomous robotnavigation. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems," in *In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002, pp. 490-495.
- [31] J.L., Flynn, A.M. Jones, *Mobile Robots: Inspiration to Implementation*,. Wellesley, MA: A.K. Peters Ltd., 1993.
- [32] W. Maas, "Networks of spiking neurons: the third generation of neuronal network models," *Neuronal Networks*, vol. 10, pp. 1659-1671, 1997.
- [33] W., Kistler, W. M. Gerstner, "Spiking neuron models," *Cambridge University Press*, 2002.
- [34] Rafael Paz Vicente, "Una aportación al procesamiento de la información visual mediante técnicas bioinspiradas.," Sevilla , 2008.
- [35] Gabriel Jimenez-Moreno, Alejandro Linares-Barranco, Manuel J. Dominguez-Morales, Rafael Paz-Vicente and Anton Civit-Balcells Angel Jimenez-Fernandez, "A Neuro-Inspired Spike-Based PID Motor Controller for Multi-Motor Robots with Low Cost FPGAs," *sensors*, vol. 12, pp. 3831-3856, 26 March 2012.
- [36] A. Linares-Barranco, R. Paz-Vicente, G. Jimenez, A. Civit. A. Jimenez-Fernandez, "BuildingBlocks for Spike-based Signal Processing. ," *IJCNN*, 2010. Barcelona-Spain.
- [37] Ángel Francisco Jiménez Fernández, *Diseño y evaluación de sistemas de control y procesamiento de señales basadas en modelos neuronales pulsantes*. Sevilla: Tesis Doctoral. Universidad de Sevilla. Escuela Técnica Superior de Ingeniería Informática., 2010.
- [38] M. Jeannerod, "The timing of natural prehension movements," *J. Motors Behavior*, vol. 16, pp. 135-154., 1984.
- [39] D. Harel and A. Pnueli., "Logics and models of concurrent systems," *Computer And Systems Sciences*, *Springer-Verlag New York, Inc*, vol. 13, pp. 477-498.
- [40] D.L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6-23, Enero 1997.