



Industrial Data

ISSN: 1560-9146

iifi@unmsm.edu.pe

Universidad Nacional Mayor de San

Marcos

Perú

Quintero-Rosas, Verónica; Saldaña Higareda, Diego; Villarreal Rodríguez, Héctor

Control de seguridad industrial no invasivo

Industrial Data, vol. 18, núm. 1, 2015, pp. 91-97

Universidad Nacional Mayor de San Marcos

Lima, Perú

Disponible en: <http://www.redalyc.org/articulo.oa?id=81642256010>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Control de seguridad industrial no invasivo

RECIBIDO: 24/11/14 ACEPTADO: 03/01/15

VERÓNICA QUINTERO-ROSAS*
 DIEGO SALDAÑA HIGAREDA**
 HÉCTOR VILLARREAL RODRÍGUEZ***

RESUMEN

La industria en la actualidad cuenta con una gran gama de procesos que funcionan con una precisión y exactitud lograda tras años de desarrollo en tecnología, comúnmente los algoritmos que controlan la maquinaria industrial buscan emular la eficacia de la mano de obra humana pero a su vez aumentar la rapidez y exactitud de los procesos, un algoritmo o un robot pese a ser programado y creado en esencia por el ser humano no posee conciencia propia, sus actos están definidos por lineamientos establecidos por el operador, dentro de esto se incluye todo lo relacionado con seguridad, debido a la gran penetración que tienen los equipos automatizados en diferentes campos laborales, principalmente en la asistencia para realizar tareas que resultan de alto riesgo para una persona y requieren de la destreza humana para llevarse a cabo, la integridad de los operadores de este tipo de equipos se considera un punto de mucha importancia a desarrollar. Aplicar técnicas de seguridad en elementos automatizados utilizando sensores de reconocimiento corporal nos acerca un poco más a algo que parece imposible, concientizar a las máquinas.

Palabras clave: tecnología, seguridad, conciencia, automatización, industria, sensores

INDUSTRIAL SAFETY CONTROL, NON INVASIVE

ABSTRACT

The industry currently has a wide range of processes that operate with precision and accuracy achieved after years of development on technology, commonly algorithms that control industrial machinery seeking to emulate the efficiency of human labor but in turn increase the speed and accuracy of processes, an algorithm or a robot despite being scheduled and essentially by man has no conscience, his actions are defined by guidelines established by the operator, within this includes everything related to security, due to the high penetration with automated equipment in various industrial fields, especially in assistance to perform tasks that are at high risk for a person and require human skill to take place, the integrity of the operators of this type of equipment is considered a very important point to develop. Apply security techniques in automated elements using body sensor recognition brings us a little closer to something that seems impossible, to raise awareness machines.

Keywords: technology, safety, awareness, automation, industry, sensors

1. INTRODUCCIÓN

En México la industria automotriz fue la primera en hacer uso de robots para aumentar la producción y calidad de sus procesos de manufactura, desde entonces el uso de las maquinas dentro de la industria ha ido en potencial aumento, en un principio los robots solamente estaban destinados a sectores industriales, mecánicos y químicos, pero con los últimos avances en diferentes tecnologías de información, más áreas laborales han adoptado el uso de maquinaria para asistir la obra de mano humana o en todo caso sustituirla en su totalidad.

Los beneficios obtenidos a partir de estas prácticas van desde aumentar la velocidad en producción, mejorar la calidad de los productos producidos en masa y sobre todo reducción.

En costos de proceso, en contraste con los beneficios obtenidos por la utilización de robots en la industria, para operarlos es necesario contar con gente capacitada dentro de áreas específicas.

Apesar de que se trabaja con maquinaria precisa y cara, no debemos olvidar que esta implicada la fuerza mecánica de actuadores que pueden llegar a representar un peligro para los empleados, el apartado de seguridad en la industria automatizada requiere de especial atención en el desarrollo de técnicas que permitan a un operador realizar de forma segura trabajos que resultan peligrosos pero demandan una exactitud y precisión humana.

El total de accidentes en corporaciones es relativamente alto, desde el año 2015 al transcurso del 2015 se han registrado miles de accidentes dentro de la industria a nivel mundial [1].

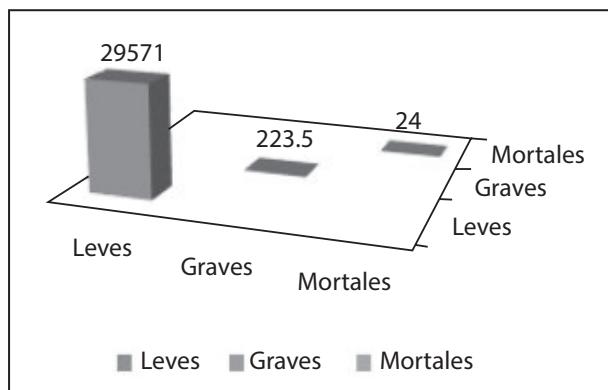
Cuando se decide sustituir a una persona por una máquina para realizar un determinado proceso, consecuentemente se pierde uno de los hechos en seguridad más importante; la toma de decisiones desde la perspectiva humana, para asegurar las ventajas de la utilización de sistemas robotizados sin perder puntos en seguridad. Podemos optar por la creación de algoritmos que utilicen el cuerpo humano de referencia en su operación como una alternativa eficiente para poder concientizar el funcionamiento de este y evitar peligros gracias a que se obtiene una máquina funcional operada por un ser lucido.

* Instituto Tecnológico de Mexicali, México Departamento de Sistemas y Computación Av. Tecnológico s/n Col. Elías Calles Mexicali, B.C., México, 21376.
 E-mail: veronicaquintero@itmexicali.edu.mx,

** E-mail: 11490759@itmexicali.edu.mx

*** E-mail: a11490773@itmexicali.edu.mx

Figura 1. Gráfica estadística sobre los accidentes de trabajo a nivel mundial, divididos por su gravedad.



2. ANTECEDENTES

Existen brazos robóticos utilizados en líneas de ensamblaje conocidos como robots SCARA, estos robots generalmente son programados para realizar una tarea repetitiva, pero igual que a un brazo robótico común, se le pueden integrar sensores de movimiento como el utilizado en este proyecto (Kinect) para obtener un sistema embebido en búsqueda de una respuesta en tiempo real [2]. Gracias al relativamente compacto tamaño de estos sensores pueden ser fácilmente empotrados en un dispositivo que pueda moverse, como una de las características buscadas en los robots móviles, es que puedan desplazarse de una manera segura por su ambiente; se puede utilizar un sensor para dotar de "vista" a un robot móvil [3] permitiéndole detectar obstáculos, la autonomía en los robots es otro campo de desarrollo en el que se puede implementar sensores utilizados en este proyecto, dado que este es capaz de crear campos de profundidad gracias a la serie de puntos infrarrojos que proyecta para estimar la proximidad de los objetos, Kinect resulta un aditamento barato y bastante preciso para añadirlo a un robot autónomo [4].

3. MARCO TEÓRICO

Kinect es un dispositivo ofrecido por Microsoft como aditamento para su consola casera de

videojuegos, mediante él se podía ofrecer una experiencia de juego, al estilo de realidad aumentada (Microsoft [5]), donde se utiliza el cuerpo completo y sus gestos como sustitución al clásico control, en un principio, como es típico de Microsoft, su funcionamiento era tratado como secreto de la compañía, Microsoft decidió ofrecer el código de sus sensor al ver que la comunidad de software libre podía sacarle provecho a su dispositivo, ofreciendo incluso ejemplos y muestras de cómo utilizar las capacidades de este, hasta la fecha existen infinidad de trabajos realizados por estudiantes utilizando Kinect, los sensores y micrófonos que posee el dispositivo de Microsoft son capaces de llevar un reconocimiento preciso de los gestos del cuerpo humano y sus extremidades, la voz y la profundidad, Microsoft también libero un kit de desarrollo para que los usuarios pudieran sacar provecho de su dispositivo (CodePlex [6]), brindando las herramientas necesarias para realizar aplicaciones capaces de utilizar reconocimiento de gestos y voz, arduino por otro lado se ha convertido en la plataforma de desarrollo por excelencia e insignia del software libre, gracias a la capacidad que posee de integración con otros distintos IDE para programar objetos interactivos lo hacen ideal para trabajar con sensores como Kinect, mediante la integración de arduino. (Processing [7]).

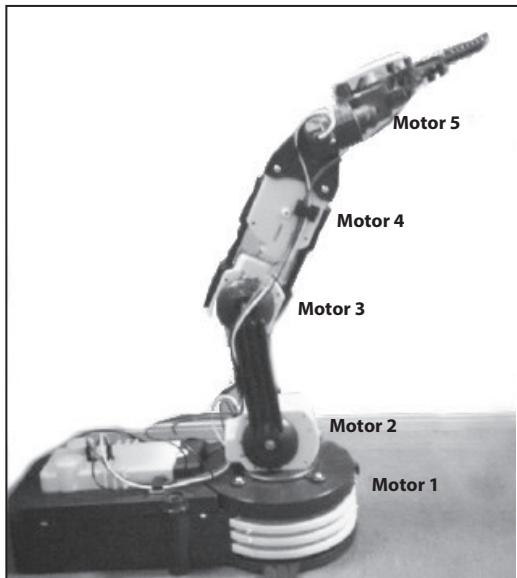
El objetivo principal del proyecto es utilizar Kinect y Arduino para poder mover un brazo robótico, el brazo utilizado fue adquirido en una tienda de electrónicos popular, para abaratar costos se optó por un brazo que utilizara motores en vez de servos.

Brazo robótico

El brazo está compuesto de 5 motores como se muestra en la fig. 1.1 utiliza independientemente como fuente de poder 4 pilas tamaño D, la pinza puede abrirse hasta 1.77 pulgadas, la base tiene 270 grados de libertad, el motor 2 cuenta con 180 grados de libertad, el motor 4 tiene 300 grados de libertad y por último el motor 5 tiene 120 grados de libertad.

Kinect y processing

Kinect, el sensor de Microsoft cuenta con una cámara RGB, un sensor de profundidad y un micrófono para captar comandos de voz [8]. Mediante Kinect captaremos los ángulos del cuerpo. Para poder interpretar esa información, utilizaremos Processing; un IDE compatible con las librerías del Kinect, en él se programa el cálculo de ángulos y las banderas que arduino recibirá para saltar entre casos.

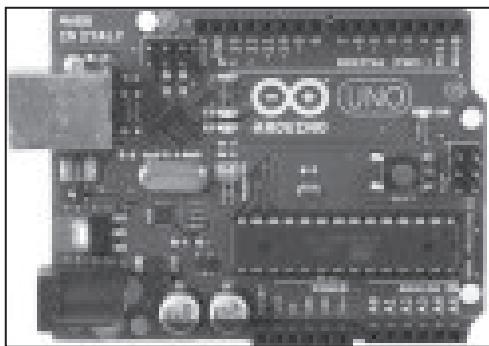
Figura 1.1. Brazo Robótico.

Fuente, I.T. Mexicali 2015.

Arduino Uno

La tarjeta arduino cuenta con las siguientes especificaciones [9]:

- 14 pines digitales
- 5 pines analógicos
- 1 Puerto USB.

Figura 1.3. Arduino Uno.

Fuente: Fotografía Arduino Uno.

Para conectar el brazo utilizaremos 3 pines para cada motor, pasando por alto las conexiones al puente H que utiliza 8 salidas del arduino.

La programación del algoritmo se realiza directamente sobre el chip de Arduino, para que sea posible conectar el Kinect a una aplicación que interprete lo datos que este envía; gracias

a processing podemos dibujar un esqueleto que servirá de guía para el brazo, los ángulos que formemos con nuestro cuerpo serán los comandos que indiquen cuando iniciar o terminar un movimiento, así como el tipo de movimiento deseado por el operador.

4. DESARROLLO

El software de programación processing es una parte vital dentro del funcionamiento de este proyecto, processing nos permite hacer uso completo del sensor óptico, programar el cálculo de los ángulos a medir y las extremidades que queremos que tome en cuenta; es encargado también de controlar los tiempos de respuesta, por lo que la rapidez del sistema depende totalmente de él.

En la fig. 1.4 se muestran las imágenes que procesa el sensor de movimiento Kinect, el esqueleto marcado en rojo es el que se toma en cuenta para realizar las mediciones.

Figura 1.4. Escaneo de imágenes con el sensor para medir los ángulos.

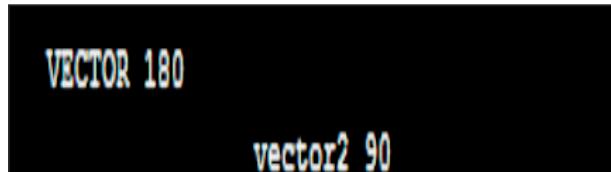
Fuente: I.T. Mexicali 2015.

Como se puede observar en la fig. 1.4 no se dibuja todo el esqueleto del sujeto, solamente utilizaremos los ángulos formados por los brazos, antebrazos hombros y cuello, mediante diferentes posiciones de manos generaremos los eventos para los movimientos del robot.

Los datos que se muestran en fig. 1.5, son los obtenidos por el software Processing, mismo que utiliza el sensor óptico para realizar las mediciones en los ángulos dibujados, los eventos se activan utilizando como referencia los ángulos leídos, dichos eventos se encuentran programados en arduino, todos los pines de arduino se utilizar

como salidas excepto por la entrada USB que se utilizara como puerto serial para recibir los datos de Processing.

Figura 1.5. Salida de datos que muestra los ángulos calculados por processing, la imagen es un recorte de la consola en el programa.



Fuente: I.T. Mexicali 2015.

Figura 2. Pines de salida de arduino, encargados del funcionamiento de los motores asistidos por el puente H.

```
void setup() {
    pinMode(E1, OUTPUT);
    pinMode(E2, OUTPUT);
    pinMode(E3, OUTPUT);
    pinMode(E4, OUTPUT);
    pinMode(I1, OUTPUT);
    pinMode(I2, OUTPUT);
    pinMode(I3, OUTPUT);
    pinMode(I4, OUTPUT);
    pinMode(I5, OUTPUT);
    pinMode(I6, OUTPUT);
    pinMode(I7, OUTPUT);
    pinMode(I8, OUTPUT);
    Serial.begin(9600);
}
```

Se utilizó un puente H [9] para permitirnos mover los motores en ambos sentidos, en cada lado del puente H se encuentran 3 elementos importantes a utilizar, ENABLE, INPUT y OUTPUT. ENABLE nos permite habilitar las entradas y salidas, INPUT permite al puente H recibir señales y OUTPUT funciona como salida.

Cuando Processing evalué el ángulo correcto, este pasara a utilizarse como bandera, en arduino tenemos programados diferentes casos para cada bandera leída (ángulos), tras leer los datos de Processing e interpretar cual caso es el correcto a ejecutar, arduino se encargara de activar los pines correctos en el puente H permitiendo el movimiento de los motores del brazo.

Figura 2.1. Evento de Arduino y lectura al puerto serial.

```
if(Serial.available() > 1)
{
    valor = Serial.read();

    switch(valor){

        case 151: //digitalWrite(E1, HIGH);
            digitalWrite(E2, HIGH);
            digitalWrite(I1, HIGH);
            digitalWrite(I2, LOW);
            digitalWrite(I3, HIGH);
            digitalWrite(I4, LOW);

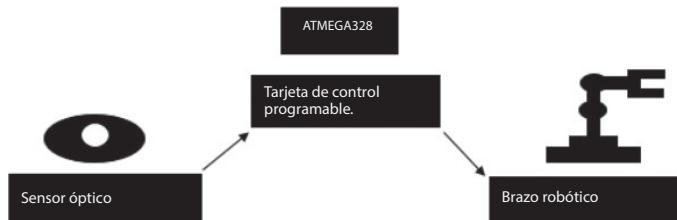
            delay(500);
            break;
    }
}
```

Fuente: I.T. Mexicali 2015.

5. MODELOS

Este es un esquema básico que muestra la interacción de las diferentes partes del sistema.

Figura 2.2. Modelo ARIMA.



Fuente: I.T. Mexicali 2015.

El sensor óptico que utilizaremos es el ya bastante mencionado Kinect de Microsoft, gracias a la tarjeta de control programable, arduino, seremos capaces de interpretar los datos de Kinect para convertir simple información en eventos que generaran movimiento en nuestro brazo robótico, todos los elementos aquí mostrados son cruciales así como su interacción entre sí.

6. TRABAJOS FUTUROS

Este tipo de sistemas tiene muchas posibles implementaciones, si estamos hablando de un sistema entero, lo podemos utilizar en cualquier industria que realice trabajos de pintura a mano, en vez de usar mucho personal podemos emplear una sola persona que se encargue de manejar este

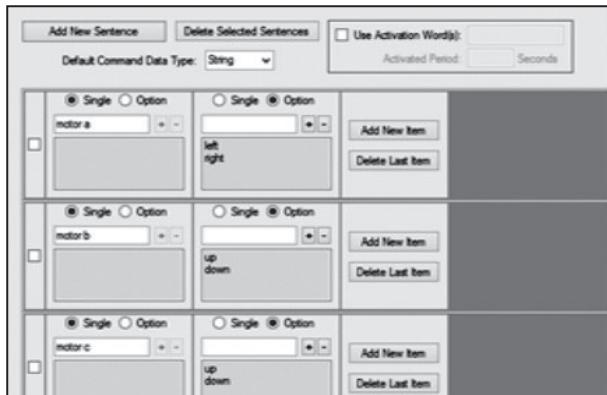
prototipo para agilizar el tiempo y disminuir costos y riesgos en la salud.

El mismo sistema se puede utilizar en fábricas que manejen materiales tóxicos o dañinos para las personas, en vez de poner en peligro a una persona con un traje aislante se puede utilizar este prototipo, así el empleado puede estar seguro desde una cámara o desde cualquier lugar fuera del peligro, además de que se puede hacer más rápido el trabajo.

El prototipo también se puede emplear en la medicina, para realizar cirugías, se le puede programar una secuencia diferente de cirugía dependiendo del problema y como el brazo es más exacto serviría de asistencia al doctor.

Si nos enfocamos en cada parte del prototipo este se puede utilizar en muchas áreas y para varios proyectos, por ejemplo, el módulo de control de voz, se puede utilizar para desarrollar vehículos que se controlen con comandos de audio, tanto para moverse, como para darle órdenes y que este se mueva automáticamente a tu posición, también se puede utilizar en conjunto con el internet de las cosas; por medio del celular dar órdenes a cualquier dispositivo de tu hogar. De hecho en esta investigación ya se tienen resultados utilizando BitVoicer. Bitvoicer es un software que permite el reconocimiento de voz para así permitir utilizar micro-controladores, dicho software permite el control de los procesamientos y el análisis de voz de PC. Las funciones que proporciona bitvoicer hacen que sea mucho más fácil y sobre todo rápido de diseñar distintos comandos de voz como se muestra en la figura:

Figura 2.3. Interfaz de creación de los comandos de voz



Fuente: Elaboración propia.

En la fig. 2.3 se muestra la interfaz de creación de los comandos, en esta interfaz para agregar una nueva secuencia se selecciona la opción de "add new sentence", a continuación se agregara uno de los cuadros que ya se encuentran en la interfaz, "default command data type" nosotros seleccionamos el tipo de dato que será el comando que se envié al controlador, dentro del cuadro de la secuencia, la opción de "single" es para decir que este comando será único y no se utilizará otro, normalmente se utiliza para definir la primera parte de un comando entero el cual tendrá varias opciones para la primer palabra nunca cambiara.

En nuestro caso agregamos otra parte a la secuencia, para ello ponemos la opción de "add new item", el cual está al lado derecho de la palabra ya definida (el cuadro donde escribimos el comando), en este caso ahora elegimos "option", este sirve para poder definir varias palabras que se combinaran con la palabra ya definida.

Como se puede observar nosotros definimos si será una sola palabra o serán varias, así podemos realizar una combinación de frases con diferentes funciones cada una. Todas las frases hechas se muestran en la parte inferior de la pantalla, como se muestra a continuación:

Figura 2.4. Pantalla donde se muestran los comandos creados, el tipo de dato que representan y el valor que se le asigna.

| Sentence | Data Type | Command |
|----------|-----------|---------|
| motor a | String | up |
| motor a | String | down |
| motor b | String | up |
| motor b | String | down |
| motor c | String | up |
| motor c | String | down |
| motor d | String | up |
| motor d | String | down |

Fuente: Elaboración propia.

En la fig. 2.4 se muestran todas las secuencias creadas, en la parte izquierda están las frases que el equipo detectará; en medio están el tipo de dato que será el comando y en la parte derecha está el comando o valor que representa.

En Arduino, se encuentran programadas las condiciones, que harán que el brazo ejecute el comando de voz, "Up" es la palabra que bitvoicer

reconoce como comando de voz, el disparador del evento en arduino; esta palabra es un valor que se asigna al momento de hacer las secuencias, como se asignó comandos de valor string estas deben de ser palabras y no números.

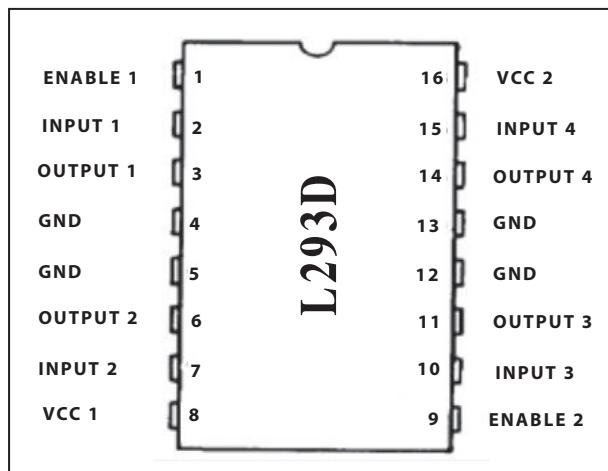
Figura 2.5. Condición para indicar que movimiento va a realizar el motor del brazo.

```
if (bvSerial.strData == "aup")
{
    //digitalWrite(E1, HIGH);
    digitalWrite(E2, HIGH);

    //digitalWrite(I1, HIGH);
    //digitalWrite(I2, LOW);
    digitalWrite(I3, HIGH);
    digitalWrite(I4, LOW);
```

Utilizamos un puente H para permitirnos mover los motores del brazo robótico [10] en ambos sentidos, en cada lado del puente H se encuentran 3 elementos importantes a utilizar, ENABLE, INPUT y OUTPUT. ENABLE nos permite habilitar las entradas y salidas, INPUT permite al puente H recibir señales y OUTPUT funciona como salida.

Figura 2.6. Diagrama del Puente H.



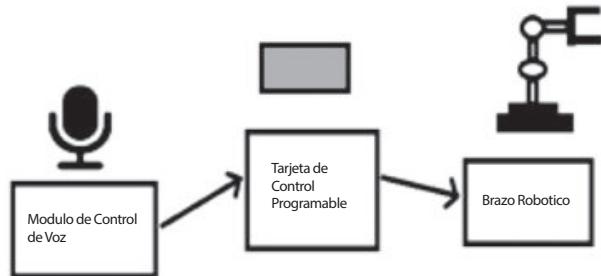
Fuente: <http://cdn.instructables.com/F7S/F6RQ/H4AGJ4GA/>

El programa que utilizamos para detectar los comandos de voz, utiliza el micrófono que viene

incluido en la computadora, si no se cuenta con uno, se puede utilizar un headset o cualquier dispositivo con un micrófono incluido, el usuario dará indicaciones al brazo por medio de este micrófono.

Al detectar las órdenes de voz dadas por el usuario, el controlador arduino realiza comparaciones, y dependiendo de cada instrucción realizará una acción diferente, todos estos datos los utilizan los motores del brazo quien realizará las acciones correspondientes.

Figura 2.7. Modelo del prototipo.



Fuente: Elaboración propia.

Otro punto como trabajo futuro el cual no está implementado aun es el módulo de control visual; se puede utilizar en los simuladores aéreos, en vez de tener a un piloto, es posible manejarlo desde la seguridad del hangar sin sufrir ningún daño en caso de un accidente, se puede utilizar para manejar maquinaria pesada, sin que el obrero esté en la máquina.

Como idea final, el prototipo entero se puede utilizar para desarrollar un vehículo de rescate en caso de desastres naturales, en vez de mandar a los voluntarios y estos corran un gran riesgo salvando a los demás, se puede utilizar un vehículo manejado por uno de los rescatistas desde la distancia, así el vehículo podría levantar escombros sin ningún problema y moverse por cualquier lugar para rescatar heridos en lugar de un ser humano.

7. CONCLUSIONES

El acelerado crecimiento de las redes de comunicación y los recientes avances en automatización de procesos nos lleva a preguntarnos qué impide que cada vez más empresas adopten el uso de robots y sistemas

inteligentes, uno de los principales factores en contra de la completa adopción de maquinaria automatizada en la industria es el costo, los costos no solo radican en la adquisición propia del equipamiento, sino en el mantenimiento de estos y peor aún en reparación en caso de daños, al entorno en el que opera o a un ente físico que podría ser desafortunadamente una persona, esto nos lleva a concluir que la seguridad es un factor delicado a considerar en cualquier institución, el peligro que puede representar un sistema automatizado debería ser mínimo, si las personas que intervienen en él se encuentran suficientemente capacitados.

Por más conocimientos que se tengan sobre una herramienta de este tipo, el control sobre ella no es absoluto debido a que están hechas para agilizar tareas y esto implica eliminar los aspectos ineficientes de la mano de obra humana, para contrarrestar tales factores es necesario que exista un equilibrio entre eficiencia, exactitud y complejidad, convergiendo entonces en un sistema capaz de llevar a cabo procesos que resulten peligrosos para un operador humano, pero que requieran de la destreza y toma de decisiones desde la perspectiva de este.

8. REFERENCIAS BIBLIOGRÁFICAS

- [1] <http://www.empleo.gob.es/estadisticas/eat/welcome.htm>
- [2] Pillajo C. "Human Machine Interface HMI Using Kinect Sensor to control a SCARA Robot." 2013
- [3] J.R. Ruiz-Sarmiento. "Navegación Reactiva de un Robot Móvil usando Kinect" España.
- [4] Ortiz Correa D. S. "Mobile Robots Navigation in indoor Environments Using Kinect Sensor" 2012.
- [5] Microsoft.(2010) Patente n ° S011320. EUA.
- [6] <https://kinectforwindows.codeplex.com/>
- [7] <https://processing.org/>
- [8] http://research.microsoft.com/en-us/events/fs2011/jancke_kinect_programming.pdf
- [9] arduino.cc
- [10] <http://pdf.datasheetcatalog.com/datasheet/SGSThomsonMicroelectronics/mXrszwq.pdf>