

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Electrónica

Monitoreo y control remoto del brazo robótico Staubli RX-90.

**Informe de Proyecto de Graduación para optar por el título de Ingeniero
en Electrónica con el grado académico de Licenciatura**

Gabriel Guillén Rojas

Cartago, Septiembre de 2006


INSTITUTO TECNOLÓGICO DE COSTA RICA
ESCUELA DE INGENIERIA ELECTRONICA
PROYECTO DE GRADUACIÓN
TRIBUNAL EVALUADOR

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

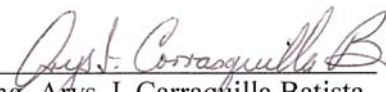
Miembros del Tribunal


Ing. Faustino Montes de Oca Murillo

Profesor lector


Ing. Marvin Hernández Cisneros

Profesor lector


Ing. Arys I. Carraquilla Batista

Profesora asesora



Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

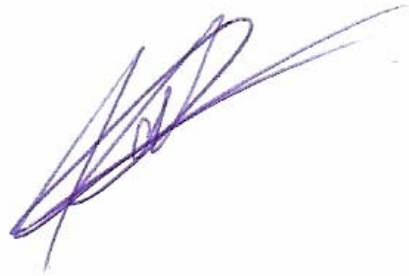
Cartago, 20 de septiembre de 2006

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, Costa Rica.
20 de septiembre de 2006

A handwritten signature in blue ink, consisting of several overlapping loops and a long horizontal stroke extending to the right.

Gabriel Guillen Rojas

Cédula: 1 - 1172 - 0880

RESUMEN

Con este proyecto se logra el control local y remoto del brazo robótico Staubli RX-90 en la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica. El proyecto consiste en un servidor, que publica una página Web para el control y monitoreo del brazo robótico; desde esta página se pueden cambiar los parámetros y mover el brazo robótico desde cualquier computadora en la Red de la Escuela de Ingeniería Electrónica por medio de un navegador de Internet.

Así mismo se incorpora una aplicación, desarrollada previamente, de reconocimiento de imágenes en dos dimensiones, para la ubicación de objetos por parte del brazo robótico para ser ejecutada de manera remota.

La importancia de este proyecto es su posible aplicación en procesos industriales, que requieren de una supervisión humana constante, de manera que esta podrá ser realizada de manera remota y automática. Así mismo se estaría protegiendo la integridad física de aquellas personas que estén expuestas a condiciones de alto riesgo en su lugar de trabajo, ya que el trabajo sería realizado por el brazo robótico.

El acceso remoto se realiza por medio de una red TCP/IP, teniendo una computadora local en el brazo robótico que funcione como Servidor Web, el usuario o cliente puede controlar remotamente las funciones a realizar por el brazo robótico, así como supervisar su funcionamiento.

Palabras Claves: Acceso Remoto, Brazo robótico Staubli RX-90, Servidor Web, Matlab Web Server, IIS, Cliente TCP/IP

Abstract

The remote access to a robotic arm Staubli RX-90 in the Electronic Engineering School of the Instituto Tecnológico de Costa Rica is achieved with this project. The project consists in a Web Server that publishes a Web page to supervise and control the robotic arm from any computer on the Electronic Engineering School Network.

A previously two dimension figures recognition application is incorporated in to the project, so the robotic arm can find objects.

The importance of this project is its application on Industrial that requires a constant human supervision, so this can be done from a far away place at any time. This way the health of any person involve with a highly risky process is being protected because they don't need to go to that point.

The remote access is done trough a TCP/IP network, having a local computer that works as a web server with the robotic arm. The client can supervise and control the functions to do with the arm.

Keywords: Remote access, Staubli RX-90 Robotic Arm, Web Server, Matlab Web Server, IIS, TCP/IP Client

ÍNDICE GENERAL

Capítulo 1: Introducción	9
1.1 Problema existente e importancia de su solución.....	9
1.2 Solución Seleccionada.....	9
Capítulo 2: Meta y Objetivos	11
2.1 Meta.....	11
2.2 Objetivo general.....	11
2.3 Objetivos específicos.....	11
Capítulo 3: Marco teórico	12
3.1 Descripción del sistema por mejorar.....	12
3.2 Antecedentes Bibliográficos.....	16
3.3 Descripción de los principales principios electrónicos relacionados con la solución del problema	17
3.3.1 Brazo robótico.....	17
3.3.2 Servidor Web.....	18
3.3.3 <i>MATLAB</i> Web Server.....	19
Capítulo 4: Procedimiento metodológico	22
4.1 Reconocimiento y definición del problema.....	22
4.2 Obtención y análisis de información.....	22
4.3 Evaluación de las alternativas y síntesis de una solución.....	22
4.4 Implementación de la solución.....	23
Capítulo 5: Descripción detallada de la solución	24
5.1 Análisis y selección de la solución.....	25
5.2 Descripción del software.....	25
5.2.1 Programación de Software.....	26
5.2.2 Configuración de Software.....	31
5.2.3 Configuración de la Red.....	32
5.2.4 Acceso desde Internet.....	33
5.3 Desarrollo de la cinemática directa.....	34
5.4 Restricciones de movimiento.....	35
Capítulo 6: Análisis de Resultados	37
6.1 Resultados.....	37
6.2 Análisis.....	41
Capítulo 7: Conclusiones y recomendaciones	45
7.1 Conclusiones.....	45
7.2 Recomendaciones.....	46
Bibliografía	47
Apéndices	50
Apéndice A.1 Glosario, abreviaturas y simbología.....	50
Apéndice A.2 Manuales de usuario.....	54
Apéndice A.2.1 Configuración de IIS.....	54
Apéndice A.2.2 Configuración de <i>MATLAB</i> Web Server.....	59
Apéndice A.2.3 Configuración de YAWCAM.....	61
Apéndice A.2.4 Manual de usuario de las páginas web.....	62
Apéndice A.3 Información sobre la institución.....	63
Apéndice A.3.1 Descripción de la institución.....	63
Apéndice A.3.2 Descripción del departamento en el que se realizó el proyecto.....	63
Apéndice A.4 Rutinas de programación.....	64
Apéndice A.4.1 Programas de <i>MATLAB</i>	64
Apéndice A.4.1.1 Activar.m:.....	64
Apéndice A.4.1.2 Desactivar.m:.....	64
Apéndice A.4.1.3 Join.m:.....	65
Apéndice A.4.1.4 Guarda.m:.....	66
Apéndice A.4.1.5 Ira.m:.....	66
Apéndice A.4.1.6 Maxvel.m:.....	67
Apéndice A.4.1.7 Encontrar.m:.....	68
Apéndice A.4.2 Código HTML y ASP.....	71
Apéndice A.4.2.1 Index.asp:.....	71

Apéndice A.4.2.2	Inicio.asp:.....	73
Apéndice A.4.2.3	Join1.asp:	75
Apéndice A.4.2.4	Join.asp:	76
Apéndice A.4.2.5	Captura1.asp:	83
Apéndice A.4.2.6	Imcaptura1.asp:	84
Apéndice A.4.2.7	Status.html:	86
Apéndice A.4.2.8	Bajo.html:.....	86
Apéndice A.5	Resolución del problema cinemática directo para el robot Staubli RX-90 [10]	87
Apéndice A.5.1	Numeración de las articulaciones y eslabones	87
Apéndice A.5.2	Obtención de los parámetro D-H para el brazo robótico Stauli RX-90...	87
Apéndice A.5.3	Obtención de las matrices de transformación ${}^{i-1}T_i$	89
Apéndice A.5.4	Obtención de la matriz de transformación T	90
Anexos	91
Anexo B.1	Programa en MATLAB para el reconocimiento de Figuras en 2D	91

ÍNDICE DE FIGURAS

Figura 1.1	Solución seleccionada	10
Figura 3.1	Brazo robótico [3]	13
Figura 5.1	Diagrama General de la Solución.....	24
Figura 5.2	Diagrama Específico de la Solución	25
Figura 5.3	Diagrama de bloques del Software.....	27
Figura 5.4	Posibilidades de ejecutar programas según el Nivel de usuario	29
Figura 5.5	Diagrama de bloques de interoperabilidad de software.....	32
Figura 5.6	Angulo de giro de cada articulación y referencia del brazo robótico	35
Figura 5.7	Restricciones de movimiento	36
Figura A.2.1	Instalación de IIS.....	54
Figura A.2.2	Inicialización del “Default website”	55
Figura A.2.3	Página de inicio de IIS	56
Figura A.2.4	Configuración del Home Directory de IIS	57
Figura A.2.5	Configuración del “Documents” de IIS.....	58
Figura A.2.1	Inicio.asp	62
Figura A.4.1	Index.asp	71
Figura A.4.2	Inicio.asp	73
Figura A.4.3	Join1.asp.....	75
Figura A.4.4	Captura1.asp.....	83
Figura A.5.1.	Articulaciones del Brazo robótico Staubli RX-90.....	87

ÍNDICE DE TABLAS

Tabla 3.1	Características mecánicas del brazo robótico [20]	12
Tabla 3.2	Área de trabajo y velocidad nominal de cada articulación [20].....	13
Tabla 6.1	Pruebas realizadas en diferentes computadoras antes de la configuración del switch de capa 3	39
Tabla 6.2	Pruebas realizadas en diferentes computadoras después de la configuración del switch de capa 3.....	39
Tabla 6.3	Pruebas de accesibilidad con diferentes usuarios	40
Tabla A.5.1	Parámetros D-H para el robot Staubli RX-90.....	87

Capítulo 1: Introducción

1.1 Problema existente e importancia de su solución

El problema consiste en monitorear y controlar de manera remota el brazo robótico ubicado en la escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica.

La importancia de haber solucionado este problema es su aplicación en la industria, poniendo como ejemplo lugares de alto riesgo, donde se manejen químicos peligrosos o condiciones peligrosas. De esta manera, si los procesos o trabajos que se deben realizar en estas áreas son llevados a cabo por personas en la misma zona, estas no tengan que exponerse a estos riesgos, ya que el brazo robótico hará el trabajo sin la necesidad de tener a ninguna persona cerca y siendo controlado desde una posición segura para el trabajador.

1.2 Solución Seleccionada

El problema se solucionó implementando un servidor Web para el acceso remoto y del programa MATLAB para el procesamiento de datos y manejo del brazo robótico.

Como se observa en la Figura 1.1, se accede de manera remota al brazo robótico desde cualquier computadora en la Red, por medio de un navegador de Internet (Internet Explorer de Microsoft®, Mozilla, etc). Aquí el usuario debe ingresar al sitio Web <http://ierobot/> y validar su sesión para poder acceder a la página Web para el control y monitoreo del brazo robótico de manera remota.

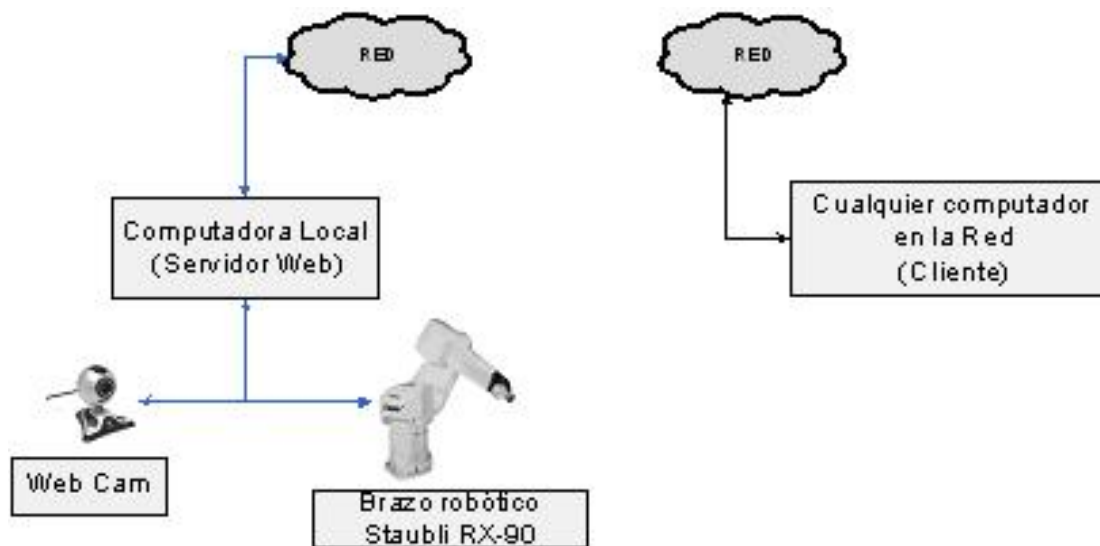


Figura 1.1 Solución seleccionada

Capítulo 2: Meta y Objetivos

2.1 Meta

Aplicar en la industria las funciones desarrolladas en el brazo robótico para llevar a cabo una tarea que requiera de una interacción humana constante.

2.2 Objetivo general

Operar y monitorear el brazo robótico Staubli RX-90 de manera remota

2.3 Objetivos específicos

- Formular un modelo de la cinemática directa del brazo robótico.
- Utilizar la Internet para acceder de manera remota al brazo robótico.
- Crear la aplicación, para el control y monitoreo de manera remota, del brazo robótico.

Capítulo 3: Marco teórico

3.1 Descripción del sistema por mejorar

Se cuenta con un brazo robótico con las características mecánicas mostradas en la tabla 3.1, tiene 6 articulaciones y eslabones (Figura 3.1), las articulaciones están enumeradas de 1 a 6 con su respectivo sentido de giro y los eslabones de 'A' a 'F', donde 'A' corresponde a la base del brazo robótico, en la tabla 3.2 se muestra el área de trabajo de cada articulación, lo que corresponde a la máxima cantidad absoluta que puede girar cada articulación, por ejemplo, la articulación 1 tiene un área de trabajo de 320°, lo que quiere decir que desde su posición 0, la articulación 1 puede girar un ángulo de -160° a 160° con una velocidad máxima de 236°/s

Tabla 3.1 Características mecánicas del brazo robótico [20]

Capacidad nominal de carga	6 Kg.
Máxima capacidad de carga	11 Kg.
Precisión en la repetición	0,02 mm
Alcance	985 mm
Montaje	Suelo, Techo.

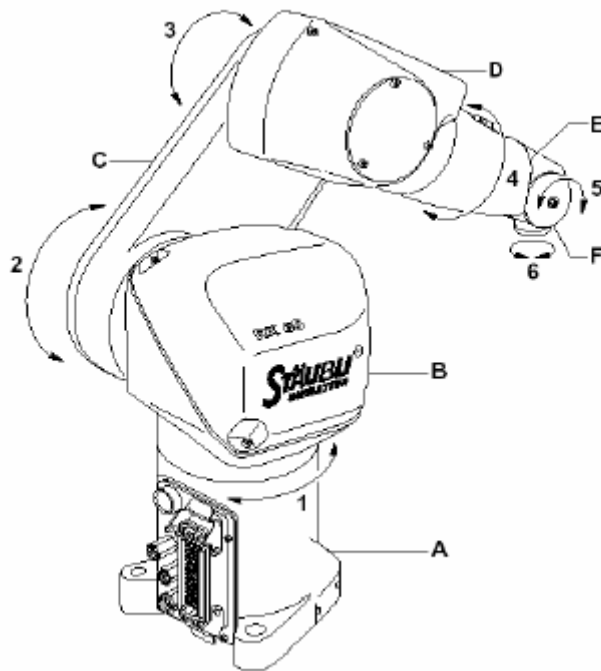


Figura 3.1 Brazo robótico [3]

Tabla 3.2 Área de trabajo y velocidad nominal de cada articulación [20]

Articulación	Área de trabajo	Velocidad nominal
1	320°	236°/s
2	275°	200°/s
3	285°	286°/s
4	540°	401°/s,
5	225°	320°/s
6	540°	580°/s

Este brazo robótico mide en total 1.405 m de alto, sus dimensiones y la de cada eslabón se muestran en la Figura 3.2.

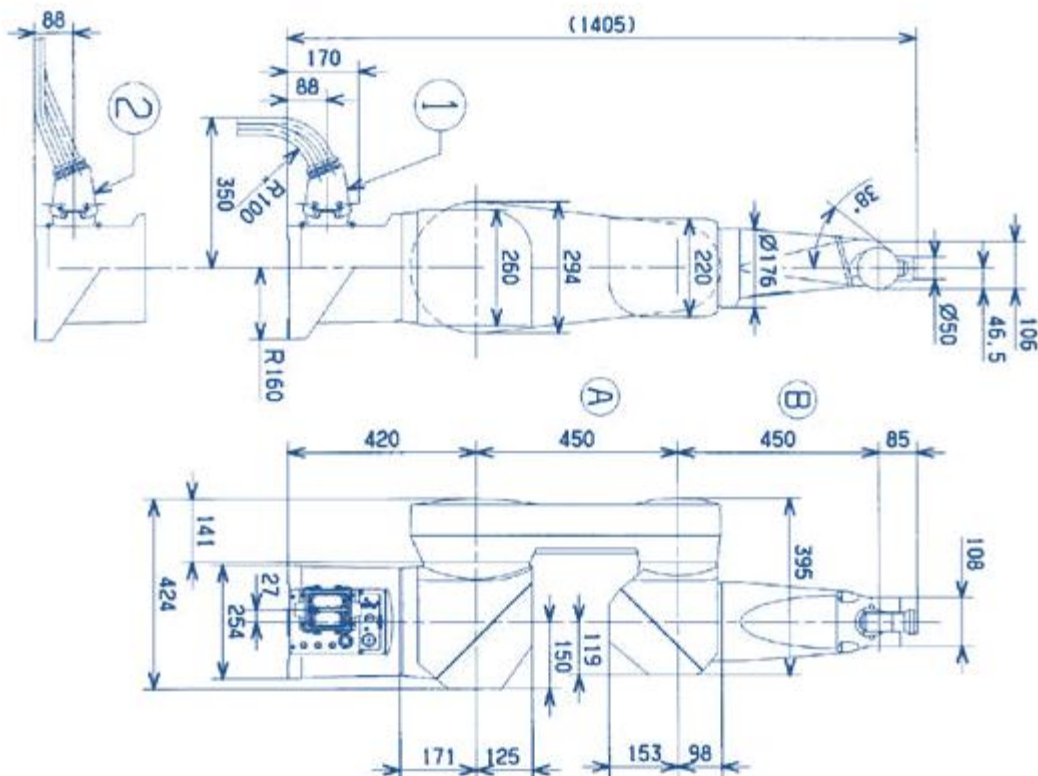


Figura 3.2 Dimensiones del brazo robótico Staubli RX90 [16]

El brazo robótico se encuentra ubicado en el Laboratorio de electromecánica de la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica, cuenta con una computadora para el control local del mismo, con las siguientes características:

Pentium 4, 1.60 GHz

256 Mb de memoria RAM

Sistema Operativo Windows XP, SP2

Disco Duro 40 Gb

Esta computadora tiene asignada la dirección IP 172.21.104.65 dentro de la red de la escuela de Ingeniería Electrónica. También se cuenta con una Webcam Logitech, que se conecta a un puerto USB de la computadora.

La comunicación entre la computadora y el brazo robótico se hace por medio del puerto serial, utilizando el lenguaje de programación llamado V+ que cuenta con diversas funciones para controlar el brazo robótico, se cuenta, entre la

bibliografía con la guía de referencia de programación con lenguaje V+, Adept
Technology.[1]

3.2 Antecedentes Bibliográficos

Para la obtención de información se utilizó el manual de usuario realizado por estudiantes del ITCR [3], se obtuvo información de Internet en sitio relacionados con robótica, redes de computadoras y uso del lenguaje de programación V+.

También se cuenta con un programa de *MATLAB* (Anexo B.1) suministrado por la Escuela de Ingeniería Electrónica, para el reconocimiento de Figuras en dos dimensiones por medio de Procesamiento Digital de una Imagen. Con una Webcam conectada a la PC, el usuario le solicita al programa que encuentre alguna figura (triángulo, rectángulo, cuadrado o círculo), el programa toma la imagen de la Webcam, la procesa, le indica al usuario si encuentra la figura solicitada y mueve el brazo robótico hacia la posición donde se encuentra la figura. Este programa sólo puede procesar como máximo 4 figuras a la vez, pueden ser todas iguales o distintas.

3.3 Descripción de los principales principios electrónicos relacionados con la solución del problema

3.3.1 Brazo robótico

“Descripción general del brazo robótico: El brazo consiste de los segmentos o los miembros interconectados por las articulaciones. Cada articulación abarca un eje alrededor de el cual dos miembros giran. Los movimientos de las coyunturas del robot son generados por los motores acoplados a los actuadores. Cada uno de estos motores tiene asociado un freno propio.

El brazo es suficientemente flexible y puede realizar una gran variedad de funciones.” [3]

“Dado que un robot puede considerarse como una cadena cinemática formada por objetos rígidos o eslabones unidos entre sí mediante articulaciones, se puede establecer un sistema de referencia fijo situado en la base del robot y describir la localización de cada uno de los eslabones con respecto a dicho sistema de referencia. De esta forma, el problema cinemático directo se reduce a encontrar una matriz homogénea de transformación T que relacione la posición y orientación del extremo del robot respecto del sistema de referencia fijo situado en la base del mismo. Esta matriz T será función de las coordenadas articulares” [7]

Para desarrollar la cinemática directa del brazo robótico se utiliza el algoritmo de Denavit-Hartenberg, para lo cual se cuenta en la bibliografía con el libro de Fundamentos de robótica de Antonio Barrientos [10].

El desarrollo de la cinemática directa, se utiliza para obtener las ecuaciones de ubicación espacial del extremo del brazo robótico con respecto a algún punto de referencia en función de la coordenadas articulares, de esta manera se pueden obtener las restricciones de movimiento del brazo robótico para impedir que este vaya a golpear algún objeto o a sí mismo.

3.3.2 Servidor Web

El acceso remoto se puede hacer por medio de una red de área local (LAN). En la que se sigue el modelo de servidor / cliente: “La definición del modelo tiene tres componentes básicos: un cliente (un dispositivo con el cual se accede a los servicios del servidor), un servidor (donde está o se "sirve" lo que uno necesita acceder") y la red que se utiliza para conectar estos dos componentes. Este modelo ha permitido el desarrollo amplio a cada uno de los extremos, es decir en el cliente, y en el servidor, sin tener que depender el uno del otro para poder avanzar en su desarrollo.” [6]

Para este proyecto, el cliente consiste en cualquier máquina en la red interna de la escuela, el servidor es la computadora local conectada al brazo robótico y la red de área Local de la escuela de ingeniería electrónica. Para poder interpretar o enviar datos, el servidor debe hablar un protocolo que pueda entender cualquier computadora en la red.

“Un servidor Web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.” [11]

Existen varios servicios, como Apache o IIS de Microsoft® entre los más usados hoy en día, para que una computadora interprete el protocolo HTTP y pueda proveer los servicios necesarios para funcionar como un servidor Web. Apache es de más difícil configuración que IIS y se utiliza principalmente en sistemas operativos Linux, en cambio IIS se utiliza principalmente en sistemas operativos Windows, este se distribuye con el Windows XP, la última versión de IIS en el año 2006, para este sistema operativo es la 5.1 . El IIS hace que la computadora trabaje como un servidor Web, por lo que si se tiene instalado se pueden publicar páginas Web. Al recibir una solicitud http por un cliente el servidor http responde con lo que se le solicita. Ya sea remitir un código para ejecutar una aplicación del lado del cliente o devolver el resultado de ejecutar la aplicación en el lado del servidor, en este caso el servidor ejecuta la aplicación

y genera un código HTML para ser enviado por medio de HTTP, a esta tecnología se le llama CGI Common Gateway Interface (interfaz Común de Pasarela). CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor Web y una aplicación externa. El servidor Web pasa las solicitudes hechas por parte del cliente a un programa externo en el lado del servidor. El resultado dado por este programa es enviado al cliente. De esta manera se puede hacer solicitudes del lado del cliente para que el servidor ejecute una aplicación.

Para establecer la comunicación entre el cliente y servidor se utilizan los puertos de red, los cuales están divididos en tres grupos: puertos bien conocidos (de 0 a 1023), puertos registrados (de 1024 a 49151) y los dinámicos y / o privados (de 49152 a 65535). Los puertos son asignados a un protocolo por la IANA (por sus siglas en inglés Agencia de Asignación de Números Internet) [25], el protocolo HTTP utiliza el puerto de red 80. Cualquier aplicación o protocolo que provea un servicio de red requiere de un puerto de red, si esta aplicación o protocolo es a nivel Interno de una Red, se puede utilizar cualquier puerto de red disponible

3.3.3 *MATLAB* Web Server

MATLAB Web Server funciona como CGI, que recibe los datos del cliente por medio de una página Web, ejecuta una aplicación *MATLAB* en el servidor y envía el resultado al cliente. *MATLAB* Web Server consiste en un grupo de programas que habilita a los programadores de *MATLAB* crear aplicaciones y ejecutarlas desde Internet ver figura 3.3:

- *MATLAB* server: es un servidor TCP/IP, ejecuta el programa de *MATLAB* (archivo *.m) especificado en el documento HTML en el campo escondido llamado *mfile*. *MATLAB* Server invoca a *matweb.m*, que ejecuta el archivo *.m especificado. Puede ser configurado para escuchar cualquier puerto TCP/IP disponible editando el archivo *MATLAB server.conf*.

- Matweb: es el cliente TCP/IP de *MATLAB Server*, este programa usa el CGI para extraer datos de documentos HTML y transferirlos a *MATLAB Server*
- Matweb.m: Llama al archivo *.m que se quiere ejecutar.

Se usan dos archivos de configuración de *MATLAB Web Server*:

- Matweb.conf: archivo que necesita matweb para conectarse con *MATLAB Server*, las aplicaciones deben de estar enlistadas aquí
- *MATLAB server.conf*: archivo donde se configuran los atributos TCP/IP de *MATLAB Web Server*

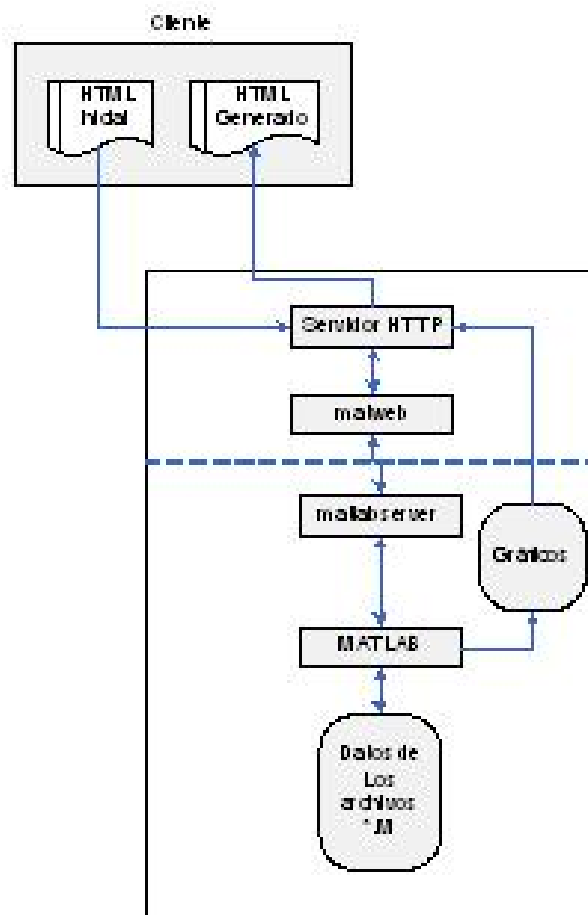


Figura 3.3 *MATLAB Web Server*

Los requerimientos recomendados para la instalación de Matlab Web Server, son los mismos que para la instalación de Matlab, según el fabricante en su sitio Web [23]:

- Sistema operativo Windows XP
- Procesador Pentium III
- 460 Mb espacio libre en Disco Duro
- 1024 Mb de RAM

Capítulo 4: Procedimiento metodológico

4.1 Reconocimiento y definición del problema

El problema ha sido planteado por parte de la asesora del proyecto, la Ing. Arys Carrasquilla, la idea es aplicar las funciones y aplicaciones a desarrolladas en la industria.

4.2 Obtención y análisis de información

La información más relevante se ha obtenido por medio de Internet, manuales de usuario del brazo robótico y guía de referencia de programación del lenguaje, V+, entre otros.

4.3 Evaluación de las alternativas y síntesis de una solución

Para la implementación del servidor HTTP se evaluó la posibilidad de utilizar Apache o IIS, que para efectos de solucionar el problema ambos daban los mismos resultados, con la diferencia de que IIS es de muy simple configuración y viene incluido en el paquete de Microsoft Windows XP.

Se requería que la solución fuera compatible con *MATLAB*, debido a que existe un proyecto previo, para reconocimiento de imágenes en 2D, además de que *MATLAB* es una herramienta de cálculo muy versátil y popular hoy en día. Luego de una investigación en Internet y en el sitio Web de *MATLAB* <http://www.mathworks.com/> se plantea como parte de la solución utilizar el programa “*MATLAB* Web Server”, el cual recibe datos por medio de una página Web para ejecutar una aplicación de *MATLAB* y enviar el resultado por una página Web.

4.4 Implementación de la solución

Para implementar la solución se siguió el siguiente procedimiento:

- Lograr el control de manera local del brazo robótico.
- Pruebas del programa de *MATLAB* suministrado para el reconocimiento de imágenes en 2D
- Crear el servidor Web en la computadora local del brazo robótico, para esto se ha utilizado el IIS de Microsoft®.
- Configurar *IIS* para que funcione con *MATLAB Web Server*
- Configurar *MATLAB Web Server* para que funcione en la computadora local
- Realizar pruebas con *MATLAB Web Server* y *IIS*, ejecutando aplicaciones de *MATLAB*
- Hacer cambio en el Switch de capa 3 de la Escuela de Ingeniería Electrónica para que las páginas Web publicadas por el servidor Web de la computadora local del brazo robótico sean accesibles en toda la red
- Crear aplicaciones de *MATLAB* para el control del brazo robótico
- Crear paginas HTML para publicar en el servidor
- Configuración de programa para visualizar la *webcam* en una página Web
- Crear el Script ASP (Active Server Pages) para que solo usuarios autenticados puedan acceder la página Web.
- Modificar los programas de *MATLAB* para que, dependiendo del nivel de usuario autenticado este pueda operar el brazo robótico.
- Pruebas de acceso remoto desde cualquier computadora habilitada dentro de la red de la Escuela de Ingeniería Electrónica al brazo robótico, en las cuales se movió el brazo y se visualizaron todos estos movimientos
- Estudio y desarrollo de la cinemática directa del brazo robótico

Capítulo 5: Descripción detallada de la solución

Para lograr el acceso remoto al brazo robótico se recurre a la red interna de la Escuela de Ingeniería Electrónica, como se muestra en la Figura 5.1, donde cualquier computadora en al red es un cliente y la computadora conectada al brazo robótico es el servidor, el cual procesa las solicitudes recibidas de algún cliente para manejar el brazo robótico o desplegar la imagen de la webcam.

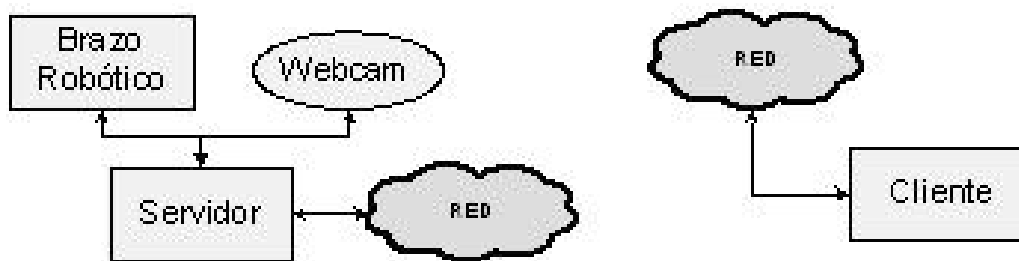


Figura 5.1 Diagrama General de la Solución

Para el manejo de los servicios de red del servidor se utiliza el programa IIS, el cual se encarga de recibir las solicitudes de los clientes y procesarlas según sea necesario. Como observa en la figura 5.2, IIS recibe las solicitudes de la Red, y según sea necesario envía los datos a *MATLAB* Web Server para que este los envíe a *MATLAB* para su procesamiento. *MATLAB* recibe también la imagen de la webcam para el procesamiento digital de la misma, *MATLAB* se encarga de controlar el brazo robótico y realizar cálculos, los resultados son enviados a *MATLAB* Web Server que este a su vez los publica en una página Web por medio de IIS.

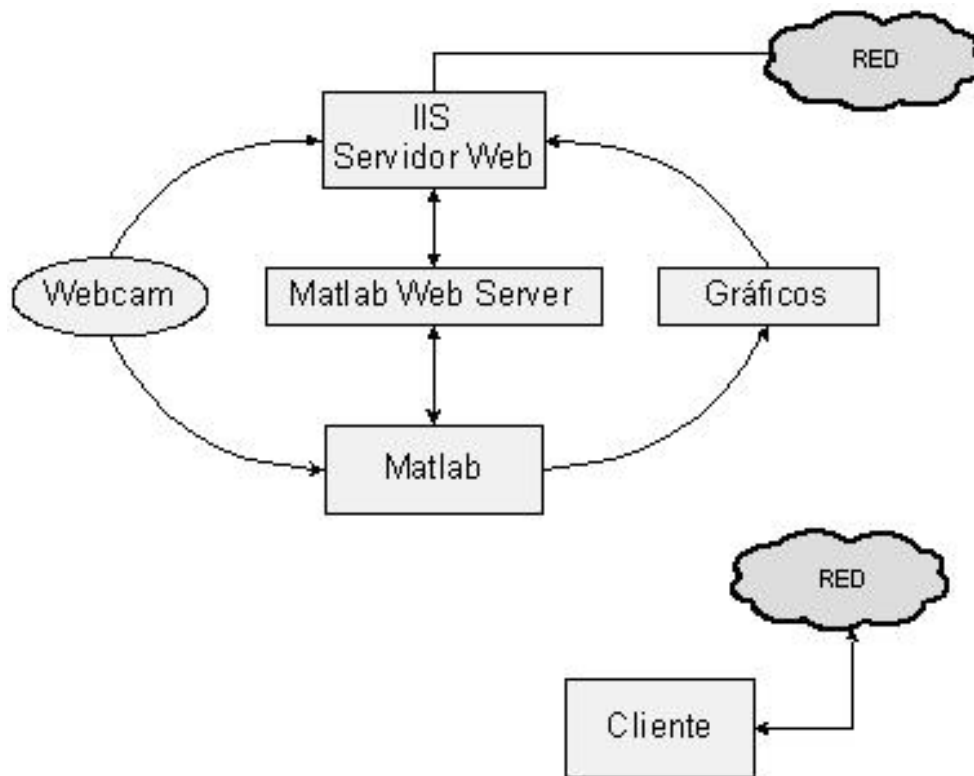


Figura 5.2 Diagrama Específico de la Solución

5.1 Análisis y selección de la solución

Se optó por esta solución debido a la flexibilidad que presenta *MATLAB*. Teniendo como servidor Web el IIS de Microsoft®, que es de simple configuración, se puede desarrollar cualquier aplicación que requiera de cálculos complejos y manejo del brazo robótico de forma remota por medio de *MATLAB*.

5.2 Descripción del software

Para la interfaz de datos con el usuario, se utilizaron los lenguajes asp y html, con estos se crearon las páginas Web para la adquisición y despliegue de datos, el asp se utiliza principalmente para la autenticación de usuarios en el sitio Web.

Los cálculos y comunicación con el brazo robótico se hacen por medio de *MATLAB*, utilizando el lenguaje de programación de *MATLAB*.

Para una mejor explicación del software desarrollado, se divide este apartado en dos partes: Programación de Software y Configuración de Software, donde la primera parte consiste de la explicación de todo el software que se tuvo que programar y la segunda parte de la configuración del software necesario para la solución del problema.

5.2.1 Programación de Software

En la figura 5.3 se observa el diagrama de bloques del software realizado, en donde se observa cuales son los *Script* de *MATLAB* que utiliza cada página Web, todos revisan cual es el usuario que se ha validado para ejecutar la función solicitada, si este tiene los derechos suficientes procede con la solicitud, de lo contrario, envía un mensaje indicando que no se tiene suficientes derechos. A continuación se explica cada uno de ellos:

- *Join.m* (Apéndice A.4.1.3): Mueve una articulación del brazo robótico, luego de haber verificado que el brazo no va a realizar un movimiento inseguro, con los datos de articulación, ángulo y velocidad, adquiridos de la página Web. Para verificar que el brazo robótico no se va a golpear, primero se obtiene la posición actual del brazo robótico (comando '*where*' en v+) y aplica la ecuación 5.2 (ver sección 5.4)
- *Homing.m* (Apéndice A.4.1.8): Lleva el brazo robótico a la posición de Home (comando '*do ready*' en v+).
- *Guarda.m* (Apéndice A.4.1.4): Guarda la posición actual del brazo robótico (comando '*here p*' en v+) en la memoria indicada.
- *Ira.m* (Apéndice A.4.1.5): Lleva el brazo robótico a la posición especificada (comando '*do move p*' en v+)
- *Maxvel.m* (Apéndice A.4.1.6): Cambia la velocidad máxima a la que se puede mover el brazo robótico a la especificada por el usuario en la página Web (comando '*speed*' en v+).

- *Encontrar.m* (Apéndice A.4.1.7): Se utiliza el programa de reconocimiento de imágenes en 2D (Anexo B.1) provisto por la Escuela de Ingeniería Electrónica. Primero toma la imagen del archivo c:/temp/out.jpg, este archivo corresponde a la imagen más reciente que ha capturado el programa YAWCAM de lo que está viendo la Webcam, realiza un procesamiento digital de esta, para luego encontrar la figura especificada por el usuario en la página Web y mover el brazo robótico hacia la posición donde se encuentra esta figura.
- *Activar.m* (Apéndice A.4.1.1): Activa el brazo robótico
- *Desactivar.m* (Apéndice A.4.1.2): Desactiva el brazo robótico

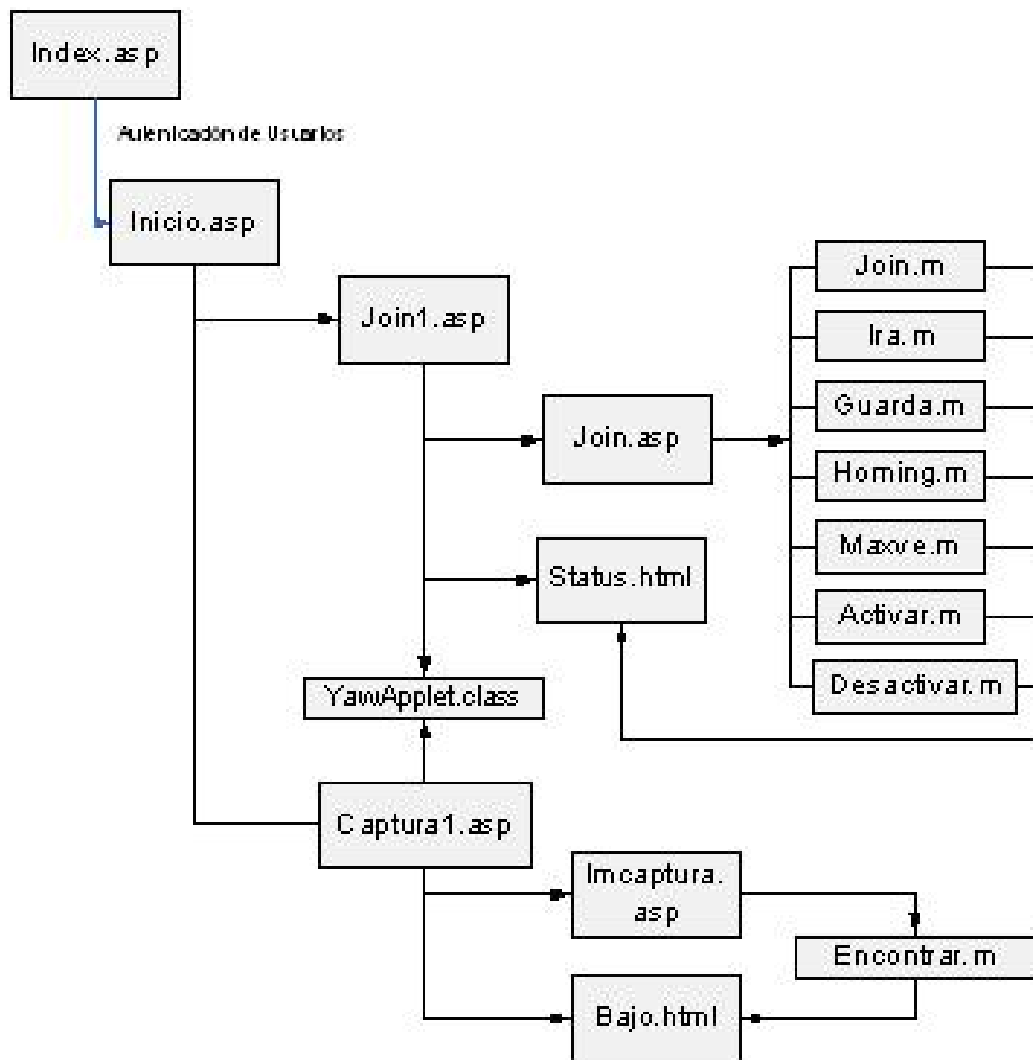


Figura 5.3 Diagrama de bloques del Software

Todas las páginas Web utilizan un *Script* de *ASP* para la validación del usuario, se muestra a continuación:

```
<%  
'este IF es el que verifica que se haya pasado por el formulario de  
login  
'se usa el mismo IF en todas las páginas restringidas.  
if session("autorizado") = "" then response.redirect "index.asp"%>
```

De esta manera se verifica que solo usuarios validados puedan acceder a las páginas y los que no han sido validados son redirigidos a la página *index.asp* (Apéndice A.4.2.1). El nivel del usuario que se ha autenticado se almacena en la variable *usuario* (`usuario <%= session("autorizado")%>`), esta variable se envía a *MATLAB* Web Server ya que en todos los códigos de programa se verifica primero cual es el nivel de usuario para saber si tiene derecho o no a ejecutar el programa.

Existen 5 niveles de usuarios, con diferentes derechos o privilegios para manejar el brazo robótico, en los programas de *MATLAB* se verifica primero cual es el nivel del usuario que ha iniciado sesión para luego proceder con la solicitud hecha por el usuario. Los derechos que tiene cada nivel de usuario son (ver figura 5.4):

Nivel 1: Es un “súper-usuario”, tiene derecho a todo.

Nivel 2: Tiene derecho a encontrar figuras con el brazo robótico (página Web *captura.asp*), habilitar, deshabilitar y llevar a posición de home el brazo robótico.

Nivel 3: Tiene derecho a mover el brazo robótico en la página *joint.asp*, así como de habilitar, deshabilitar y llevar a posición de home el brazo robótico.

Nivel 4: Tiene los mismos derechos del usuario nivel 3 además de que puede cambiar las posiciones predefinidas de 1 a 6 en el brazo robótico

Nivel 5: Únicamente tiene derecho de acceder al sitio Web y ver la *Web cam*

El nombre y contraseña de los usuarios se edita en el archivo *Index.asp* (Apéndice A.4.2.1), el campo donde se editan estos parámetros es únicamente

visible en el servidor, se debe de cambiar el texto entre las comillas para el nivel de usuario que se quiere cambiar (*user1* y *pass1* corresponden al nombre de usuario y contraseña respectivamente, del usuario de Nivel 1 y así respectivamente para los usuarios de cada nivel)

```
user1 = "user1"  
pass1 = "123"  
user2 = "user2"  
pass2 = "123"  
user3 = "user3"  
pass3 = "123"  
user4 = "user4"  
pass4 = "123"  
user5 = "user5"  
pass5 = "123"
```



Figura 5.4 Posibilidades de ejecutar programas según el Nivel de usuario

Para el acceso remoto del brazo robótico se utilizan las siguientes páginas Web:

- *Index.asp*: Página de validación de usuarios, también se llega a esta página si se intenta acceder alguna de las otras páginas sin antes haberse autenticado.
- *Inicio.asp*: Página de inicio, de aquí se va a las páginas de encontrar figuras y mover brazo robótico.
- *Join1.asp*: En esta página se reciben las ordenes de mover el brazo robótico, ir a una posición predefinida o guardar una posición.
- *Captura.asp*: Página para ordenar el reconocimiento de figuras y mover el brazo robótico para que encuentre una figura.

Para la visualización de la *webcam* se utiliza el siguiente *Script* de *Java*:

```
<APPLET CODE=YawApplet.class WIDTH="320" HEIGHT="240">
  <param name="Host" value="172.21.104.65">
  <param name="Port" value="1711">
</APPLET>
```

El *Script* de *Java* ejecuta el archivo *YawApplet.class*, que por un puerto de red disponible, despliega el vídeo en tiempo real, se utilizó el puerto de red 1711, pero pudo haber sido cualquier otro libre en la red de la Escuela de Ingeniería Electrónica. También se debe de indicar la *dirección IP* del servidor, en este caso la 172.21.104.65. Del lado del servidor se ejecuta el programa *YAWCAM*, que es el responsable de publicar la imagen de la *webcam* en el puerto 1711 así como de la obtención de la imagen para el procesamiento y ubicación de figuras. Este software (así como el archivo *YawApplet.class*) se obtuvo del sitio Web www.yawcam.com, es gratuito y de uso libre. El *Script* de *java* se ejecuta del lado del cliente por *JRE (Java Runtime Environment)* y utiliza el código en archivo *YawApplet.class* para el despliegue del video en tiempo real.

5.2.2 Configuración de Software

En la figura 5.4 se muestra la interoperabilidad entre el software utilizado, a continuación se explica cada uno de ellos y su configuración.

- *IIS*: Software de Microsoft®, viene Incluido con el sistema Operativo Windows XP en su versión 5.1. Este es el servidor Web por medio del cual el cliente puede acceder a las páginas Web en el servidor. Se configura para que se ejecute como página predeterminada *inicio.asp* y tenga en su directorio predeterminado todas las páginas Web desarrolladas. Para la configuración refiérase al Apéndice A.2.1
- *MATLAB*: Debe de estar instalado en la computadora donde se monte el servidor Web.
- *MATLAB Web Server*: Software de Mathworks, vienen incluido en un *Toolbox* de *MATLAB*. Se encarga de recibir la solicitud de IIS de ejecutar un Script determinado y regresar el resultado y Se configura de tal manera que el puerto para *MATLAB* Server sea el 1710, que según la IANA esta asignado al protocolo *Impera* [26], que no se utiliza en la Red de de la Escuela de Ingeniería electrónica. Se debe configurar cual es directorio para almacenar imágenes, en este caso *c:\MATLAB_files* y el nombre de la computadora sobre la que se implementa el servidor. Para su configuración refiérase al Apéndice A.2.2
- *YAWCAM*: Software libre de licencia, se utiliza para hacer el stream de la webcam al puerto de red 1711, el cual según la IANA se utiliza para *pptconference*, que no se utiliza en la Red de de la Escuela de Ingeniería electrónica. También se configura para que tome una imagen del lo que esta viendo la cámara cada 5

segundos en un mismo archivo (c:/temp/out.jpg), esta imagen es la que se utiliza para ser procesada por *MATLAB* para el reconocimiento de imágenes en 2D. Para su configuración refiérase al Apéndice A.2.3

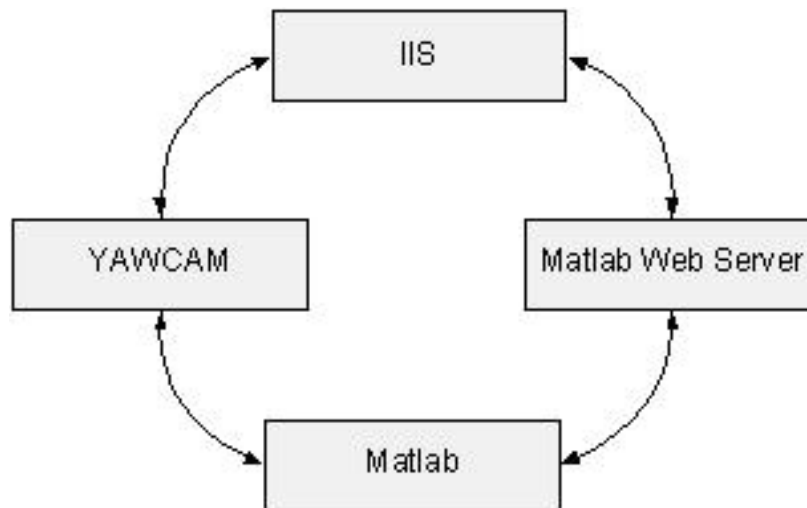


Figura 5.5 Diagrama de bloques de interoperabilidad de software

5.2.3 Configuración de la Red

Para que las computadoras en la red de la Escuela de Ingeniería Electrónica puedan acceder al servidor Web por medio de la VLAN (red de área privada virtual) se debe habilitar el tráfico de paquetes TCP/IP entre la computadora local del brazo robótico (ip: 172.21.104.65) y las redes deseadas, en este caso la red de profesores, ya que este se tiene bloqueado por razones de seguridad. También para acceder al sitio Web únicamente introduciendo el nombre *//ierobot/* se debe configurar el servidor DNS (Sistema de nombre de dominio) para que cuando una computadora solicite la página *//ierobot/* se redirija a la ip 172.21.104.65.

Para esto se recurrió al encargado de mantener la Red de la Escuela de Ingeniería Electrónica y se le solicitó que realizara este cambio, para lo cual se cambió la configuración del Switch de capa 3 de manera tal que la

computadora con la ip 172.21.104.65 pueda enviar paquetes a la red de profesores, también se agregó esta dirección ip en la en la base de datos del servidor para que responda a las peticiones hechas a //ierobot/

5.2.4 Acceso desde Internet

Para lograr el acceso remoto al brazo robótico a través de Internet se requiere de un IP pública, la cual se ha solicitado al Centro de Cómputo del Instituto Tecnológico de Costa Rica y se esta a la espera de la misma. Cuando se disponga de la IP pública se debe solicitar un NAT (por sus siglas en inglés Traducción de Dirección de Red) “...se puede utilizar para dar salida a redes públicas a computadores que se encuentran con direccionamiento privado” [27]

El NAT se debe de hacer entre la IP pública provista y la IP privada de la computadora local del brazo robótico (ip 172.21.104.65).

También se deben tomar en cuenta, por parte de los encargados del Centro de Cómputo, consideraciones de seguridad y ancho de banda necesarios para el desempeño óptimo del sistema.

5.3 Desarrollo de la cinemática directa

En el Apéndice A.5 se muestra el desarrollo de la cinemática directa del brazo robótico, esto con el objetivo de determinar las restricciones de movimiento del brazo robótico para evitar movimientos que vayan a golpear un obstáculo, sin embargo en los alrededores del brazo robótico no se presenta ningún obstáculo por lo que el desarrollo de la cinemática directa se hace únicamente con el fin de dejar la base para futuros proyectos donde se requiera colocar algún objeto en el rango de movimiento del brazo robótico (por ejemplo una banda transportadora), así mismo el brazo robótico posee una función de calibración, donde se configura para evitar que golpee algún objeto. Se obtuvieron las siguientes ecuaciones para la posición del extremo del brazo robótico Staubli RX-90 con respecto a la base del mismo en función de las coordenadas de cada articulación en grados ($\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5, \Theta_6$, ver figura 5.5) donde en la posición por defecto los valores del ángulo de cada articulación son: $\Theta_1 = 0^\circ, \Theta_2 = -90^\circ, \Theta_3 = 90^\circ, \Theta_4 = 0^\circ, \Theta_5 = 0^\circ, \Theta_6 = 0^\circ$

$$\begin{aligned}P_x (mm) &= C_1 (l_2 \cdot C_2 - l_4 \cdot S_{23}) \\P_y (mm) &= S_1 (l_2 \cdot C_2 - l_4 \cdot S_{23}) \\P_z (mm) &= l_2 \cdot C_2 + l_4 \cdot S_{23}\end{aligned}\tag{5.1}$$

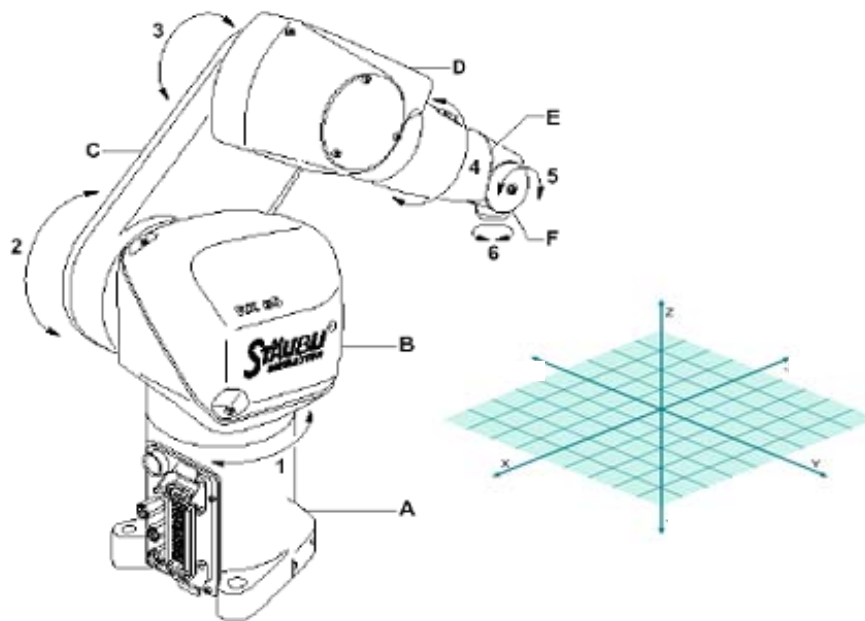


Figura 5.6 Angulo de giro de cada articulación y referencia del brazo robótico

donde

$$C_1 = \cos \theta_1 \quad l_2 = 450$$

$$S_1 = \sin \theta_1 \quad l_4 = 735$$

$$C_{23} = C_2 C_3 - S_2 S_3$$

$$S_{23} = C_2 S_3 + S_2 C_3$$

5.4 Restricciones de movimiento



Como no existe la presencia de ningún obstáculo en las inmediaciones del brazo robótico, el único peligro que existe es que el brazo robótico se golpee a sí mismo, por lo que se utiliza trigonometría para determinar las coordenadas articulares no permitidas. En la figura 5.6 se representa el eslabón 'C' y la suma de los eslabones 'D', 'E' y 'F' como un triángulo, donde se representan los ángulos de las articulaciones 2 y 3 (Refiérase a la figura 5.5 para ver los eslabones y articulaciones del brazo robótico).

Como 'C' = 'D+E+F' se tiene que $\Theta_a = \Theta_b$ y $\Theta_c = 180 - \Theta_3$

$$\Theta_a = (180 - \Theta_c)/2$$

$$\Theta_a = (\Theta_3)/2$$

Como $\Theta_a + \Theta_2$ debe de ser menor que 180°

$$(\Theta_3)/2 + \Theta_2 < 180^\circ \quad (5.2)$$

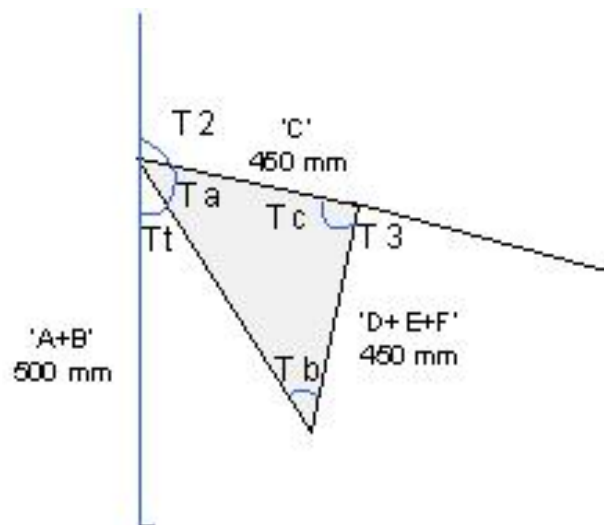


Figura 5.7 Restricciones de movimiento

Capítulo 6: Análisis de Resultados

6.1 Resultados

Se hicieron pruebas del acceso remoto al brazo robótico desde distintas computadoras conectadas a la red de la Escuela de Ingeniería Electrónica en distintos puntos habilitados para esto, en las primera pruebas no se logró el monitoreo por medio de la webcam en las computadoras que no se encontraban en la misma Red que la computadora local del brazo robótico tabla 6.1, en esta tabla se observa como no se logró la visualización en tiempo real para las computadoras fuera de la red de Laboratorio de control, sin embargo si se logra el reconocimiento en figuras en 2D debido a que esta función se hace completamente de forma local en el servidor. Luego de solicitar un cambio en el switch de capa 3 de la Escuela de Ingeniería Electrónica se logró ver en tiempo real el brazo robótico (tabla 6.2).

La computadora que controla el brazo robótico se encuentra directamente en la misma VLAN que las computadoras del Laboratorio de control automático, las pruebas de acceso remoto realizadas en este laboratorio fueron exitosas: autenticación de usuarios, se visualizó la Webcam, se movió el brazo robótico y se probó también el reconocimiento de figuras por medio de la Webcam logrando obtener los resultados esperados en cuanto a accesibilidad y monitoreo.

Para probar el acceso remoto desde otras computadoras se recurrió a una computadora dentro de la red de profesores, la cuál se encuentra en una diferente VLAN (red de área Local virtual). Se probó exitosamente la autenticación de usuarios de diferentes niveles, el monitoreo del brazo robótico por medio de la Webcam, se movió el brazo robótico y se probó también el reconocimiento de figuras de 2D.

También se utilizó una computadora conectada a la red inalámbrica de la Escuela de Ingeniería Electrónica, esta red, al igual que la red de profesores, se encuentra en una misma diferente VLAN que la de la computadora local del brazo robótico, se realizaron las mismas pruebas que para el caso anterior y todos los resultados obtenidos fueron exitosos.

En cuanto a la autenticación de usuarios se comprobó que todos los niveles de usuario si pueden realizar todas las funciones según se obtuvieron los resultados mostrados en la tabla 6.3.

Tabla 6.1 Pruebas realizadas en diferentes computadoras antes de la configuración del switch de capa 3

Computadora	Lab. Control	Profesores	Inalámbrica
Acceso a la página Web	SI	SI	SI
Visualización en tiempo real de la webcam	SI	No	No
Autenticación de usuarios	SI	SI	SI
Movimiento del robot	SI	SI	SI
Reconocimiento de figuras en 2D	SI	SI	SI

Tabla 6.2 Pruebas realizadas en diferentes computadoras después de la configuración del switch de capa 3

Computadora	Lab. Control	Profesores	Inalámbrica
Acceso a la página Web	SI	SI	SI
Visualización en tiempo real de la webcam	SI	SI	SI
Autenticación de usuarios	SI	SI	SI
Movimiento del robot	SI	SI	SI
Reconocimiento de figuras en 2D	SI	SI	SI

Tabla 6.3 Pruebas de accesibilidad con diferentes usuarios

	Nivel de usuario				
Función	<i>Nivel 1</i>	<i>Nivel 2</i>	<i>Nivel 3</i>	<i>Nivel 4</i>	<i>Nivel 5</i>
Acceder al sitio Web	Si	Si	Si	Si	Si
Habilitar brazo robótico	Si	Si	Si	Si	No
Deshabilita brazo robótico	Si	Si	Si	Si	No
Llevar a <i>home</i> brazo robótico	Si	Si	Si	Si	No
Encontrar figuras de 2D	Si	Si	No	No	No
Mover eje	Si	No	Si	Si	No
Guardar posición	Si	No	No	Si	No
Ira a posición	Si	No	No	Si	No

La computadora utilizada de manera local en el brazo robótico (servidor Web) tiene las siguientes características:

- Pentium 4, 1.60 GHz
- 256 Mb de memoria RAM
- Sistema Operativo Windows XP, SP2
- Disco Duro 40 Gb

Con lo cual en ciertos momentos su desempeño no es el más óptimo, su respuesta es relativamente lenta (de 3 a 5 segundos).

6.2 Análisis

En la metodología seguida para implementar la solución se siguió el procedimiento mostrado en la sección 4.4, primero que todo se logró el control y operación de forma local del brazo robótico, esto por medio de la computadora, también se verificó el funcionamiento del programa suministrado por la Escuela de Ingeniería Electrónica para el reconocimiento de figuras en 2D (Anexo B.1), luego de esto se procedió a habilitar la computadora local del brazo robótico como un servidor Web, para lo que se instalaron los servicios de información de Internet de Microsoft (IIS) (Apéndice A.2.1) y se configuró Matlab Web Server para que funcionara con IIS (Apéndice A.2.2) de esta manera se logró ejecutar programas simples de Matlab desde cualquier otra computadora en la red, por lo que se verificó la funcionalidad de IIS para proveer los servicios necesarios (http) para convertir una computadora en la red en un servidor Web y la ejecución de programas en Matlab de manera remota.

Uno de los principales aportes en este proyecto ha sido la implementación de la herramienta Matlab Web Server, ya que con esta se logra ejecutar programas de Matlab a través de Internet. Matlab es un programa ampliamente usado para aplicaciones de ingeniería, por lo que se podrán crear aplicaciones que puedan ser ejecutadas desde Internet.

Ya teniendo el servidor web listo se procedió a crear los Script de Matlab para ejecutar las funciones deseadas y se procedió a probar desde distintas computadoras en la red su funcionalidad, estos son Habilitar.m, Deshabilitar.m, Homming.m y Join.m, para probar estos lo que se hizo fue ir a una computadora en el Laboratorio de control automático y acceder a la página <http://172.21.104.65/join1.html>, ordenar el movimiento de algún eje y regresar a verificar si se movió el Brazo Robótico, esto se repitió para una computadora en la red de profesores y los resultados fueron los mismos, de esta manera se cumplió y verificó el objetivo del acceso remoto para el control del brazo robótico.

Luego de esto se investigó sobre el envío de la imagen de una Webcam en tiempo real y se encontró la aplicación llamada YAWCAM, en la cual se configuró el puerto de red 1711 para hacer el *Stream* de datos desde el servidor hasta el cliente, en las pruebas realizadas, se logró ver la imagen de la webcam únicamente en las computadoras del laboratorio de control automático, pero en las de la red de profesores no (tabla 6.1), investigando sobre el problema se llegó a la conclusión de que las computadoras de la red de profesores no tenían habilitado el tráfico de paquetes informáticos por el puerto 1711, por lo que se solicitó la habilitación de este puerto en el Switch de capa 3 y se agregó otro elemento a la base de datos del servidor DNS, para poder acceder la página Web, introduciendo únicamente el nombre *//ierobot/* en el explorador de Internet. Luego de hacer esto se logró visualizar la imagen de la Webcam desde las computadoras en la red de profesores en tiempo real, para probar esto se utilizó la página *http://ierobot/join1.html*, se le ordenó al brazo robótico que hiciera el movimiento de alguna articulación y se observaba en la misma página como se movía el brazo robótico, por lo que con esto se cumple con el objetivo de un acceso remoto para el monitoreo del brazo robótico. (Tabla 6.2). Así se logró crear un interfaz más amigable para el control y monitoreo del brazo robótico que la ya existente (enviando los comandos directamente al puerto serial por HyperTerminal).

De esta forma se alcanza el objetivo general de este proyecto “Operar y monitorear el brazo robótico Staubli RX-90 de manera remota” utilizando la infraestructura existente en la Escuela de Ingeniería Electrónica, se obtuvieron algunos retardos de tiempo a la hora de ejecutar los programas de Matlab, pero esto es debido a que la computadora local, no cumple con las especificaciones recomendadas por el fabricante de Matlab para el funcionamiento óptimo del sistema, por lo que se recomienda incrementar la memoria RAM de la computadora de 256 Mb a 1024 Mb.

La utilización de IIS como proveedor de los servicios de información de Internet provee una plataforma eficaz para un servidor Web, ya que en las pruebas realizadas se obtuvieron resultados positivos y se pudo acceder al brazo robótico de manera remota desde distintas computadoras en la red sin tener problemas de conectividad.

Luego haber cumplido con el objetivo general, se procedió a hacer el Script necesario para la autenticación de usuarios, esto se hizo por medio de un Script ASP, el cuál funcionó correctamente, permitiendo únicamente el acceso a las páginas Web a los usuarios autenticados y de permitir el cambio de parámetros o ejecución de movimientos según el nivel de seguridad de cada usuario (Tabla 6.3), Tomemos como ejemplo el usuario Nivel 5, si se pudo acceder al sitio web, pero cuando se intentó mover el brazo robótico, habilitarlo, deshabilitarlo, llevarlo a *home* no fue permitido, se verificó como también si un usuario intenta ingresar directamente a una página Web digitando el nombre en el explorador, sin antes haber iniciado sesión, no se le permite y se redirige a la página *index.asp*, por lo que se concluye que un Script ASP de inicio de sesión es un manera efectiva de controlar los usuarios que acceden a un sitio web. Sin embargo no se garantiza la seguridad del sitio por este medio, existen otros medios de autenticación de usuarios, por lo que se recomienda, para futuros proyectos investigar sobre estos y hacer mejoras en cuanto a la seguridad para el acceso al sitio Web.

Otro de los objetivos del proyecto es formular un modelo matemático del brazo robótico para conocer las restricciones de movimiento del brazo robótico de manera que este no se vaya a golpear a sí mismo o con algún obstáculo, como en este proyecto se trabajó únicamente con el brazo robótico, sin ningún obstáculo alrededor, la única posibilidad de golpear el brazo robótico es que se golpee a si mismo y la única forma de ocurra eso es que la suma del valor absoluto la posición de los ángulos 2 y 3 sea más de 180° , ya que estas dos articulaciones, y la 5 son los que giran sobre el eje Y (ver figura 5.5), pero la articulación 5 no se toma en cuenta ya que el tamaño del eslabón 'F', no va a provocar que se golpee el brazo robótico, esta restricción se incluyó en el programa *join.m*, de manera tal que primero se obtiene la posición actual de los ángulos J2' y J3' (Articulación 2 y 3) por medio del comando V+ '*where*' y se suman J2' o J3' al ángulo que ha introducido el usuario, si se cumple la ecuación 5.2 , se hace el movimiento, de lo contrario no se realiza el movimiento del brazo robótico y se envía un mensaje diciendo "Movimiento no permitido, cambie el ángulo de la articulación 2 o 3". Se realizaron pruebas en las cuales se movió el brazo robótico en todas las posiciones y cuando se

intentó hacer algún movimiento que fuera a provocar un golpe con la base del brazo robótico, el movimiento no se permitió, por lo que de esta manera se obtuvieron las restricciones de movimiento del brazo robótico por trigonometría.

Sin embargo, si hubiese algún obstáculo alrededor del brazo robótico sería necesario obtener la posición de este en el espacio con respecto a la base del brazo robótico en coordenadas XYZ y saber, con las coordenadas articulares, antes de hacer un movimiento si el brazo robótico va a pasar por este punto, para hacer esto se requiere obtener la ecuación de cinemática directa del brazo robótico (ecuación 5.1), siguiendo el algoritmo de Denavit-Hartenberg, que da las coordenadas XYZ, tomando como referencia la base del brazo robótico. Caso contrario, si se quiere saber cuales deben ser las coordenadas articulares para llegar a un dado punto XYZ, se debe obtener la cinemática inversa del brazo robótico.

Capítulo 7: Conclusiones y recomendaciones

7.1 Conclusiones

1. Se obtiene la cinemática directa del brazo robótico siguiendo el algoritmo de Denavit-Hartenberg.
2. Se consiguió crear la aplicación para el control y monitoreo de manera remota del brazo robótico utilizando lenguaje HTML y ASP.
3. Se logró el acceso remoto a la computadora local del brazo robótico desde cualquier computadora en la Red implementando un Servidor HTTP
4. Se logró la ejecución de programas en el lado del servidor mediante la herramienta MATLAB Web Server
5. Se consigue la autenticación de usuarios pero no se garantiza la seguridad en cuanto al acceso al servidor Web mediante un Script ASP

7.2 Recomendaciones

- 1.** Incrementar la memoria RAM de la computadora local del brazo robótico de 256 Mb a 1024 Mb, debido a que el fabricante de Matlab recomienda este valor para un mejor desenvolvimiento del programa.
- 2.** Investigar sobre como mejorar y garantizar la seguridad en cuanto al acceso al sitio Web, debido a que actualmente se hace mediante un Script ASP con el cual no se ha comprobado si presenta alguna vulnerabilidad para la seguridad.

Bibliografía

- [1] V+ Language Quick Reference (2001)[en línea] <
<http://www.adept.com/main/KE/Data/V%20PLUS/V%20Language%20Reference/vlangqr.html> > [Consulta 31 Marzo 2006]
- [2] ADEPT TECHNOLOGY INC, "v + LANGUAGE REFERENCE GUIDE, VERSION 12.1", 1997.
- [3] Vargas, David. Lester, Velenzuela; compiladores. "Manual de usuario del brazo robótico Straubuli RX-90", Escuela de Ingeniería Electrónica, ITCR.
- [4] Una introducción a la Robótica [en línea]
<<http://www.ilustrados.com/publicaciones/EpIVFuFpZVkcZBNuBt.php>>
[Consulta 14 Marzo 2006]
- [5] Instituto tecnológico de Costa Rica [en línea] <www.itcr.ac.cr> [Consulta 15 Marzo 2006]
- [6] [Modelo cliente servidor \[en línea\]](http://www.gestiopolis.com/delta/prof/PRO260.html)<<http://www.gestiopolis.com/delta/prof/PRO260.html>> [Consulta 15 Abril 2006]
- [7] Cinemática Directa [en línea] <es.wikipedia.org/wiki/Cinemática_Directa>
[Consulta 15 Abril 2006]
- [8] Cinemática Directa [en línea] <es.wikipedia.org/wiki/Cinemática_Directa>
[Consulta 15 Abril 2006]
- [9] Red Privada Virtual [en línea] <http://es.wikipedia.org/wiki/Red_privada_virtual> [Consulta 15 Abril 2006]
- [10] Barrientos, Antonio. "Fundamestos de Robótica", Editorial Mc Graw Hill, 1997

- [11] Servidor Web [en línea] < http://es.wikipedia.org/wiki/Servidor_web> [Consulta 06 Septiembre 2006]
- [12] IIS [en línea] < <http://es.wikipedia.org/wiki/IIS>> [Consulta 06 Setiembre 2006]
- [13] *MATLAB* [en línea] < <http://www.mathworks.com>> [Consulta 06 Setiembre 2006]
- [14] Stream [en línea] < http://nic-nac-project.de/~zeles/ant/trivio/glosa_inf#stream> [Consulta 06 Setiembre 2006]
- [15] ASP [en línea] < <http://www.soloasp.com.ar>> [Consulta 06 Setiembre 2006]
- [16] Staubli RX90 [en línea] < <http://www.machinery-export.com/de/roboer/staubli/staubli-RX-90-ES.html>> [Consulta 06 Setiembre 2006]
- [17] Cooperative Manipulation Testbed Development – Kinematics [en línea] < fcrar.ucf.edu/papers/fp4_dc_unf.pdf> [Consulta 06 Setiembre 2006]
- [18] Switching [en línea] < <http://neutron.ing.ucv.ve/revista-e/No4/articulo.htm>> [Consulta 06 Setiembre 2006]
- [19] http [en línea] < http://es.wikipedia.org/wiki/HTTP#Primeros_Servidores> [Consulta 06 Setiembre 2006]
- [20] Staubli RX-90 [en línea] < <http://www.machinery-export.com/de/roboer/staubli/staubli-RX-90-ES.html>> [Consulta 06 Setiembre 2006]

- [21] Apache [en línea] < http://es.wikipedia.org/wiki/Servidor_HTTP_Apache> [Consulta 06 Setiembre 2006]
- [22] Matlab Web Server [en línea] < http://attila.sdsu.edu/matlab/pdf_doc/webserver/webserver.pdf> [Consulta 06 Setiembre 2006]
- [23] Requerimientos de Matlab [en línea] < <http://www.mathworks.com/support/sysreq/r2006a/windows.html> > [Consulta 06 Setiembre 2006]
- [24] Ip [en línea] < http://es.wikipedia.org/wiki/Direccion_IP > [Consulta 06 Setiembre 2006]
- [25] IANA [en línea] < <http://es.wikipedia.org/wiki/IANA>> [Consulta 06 Setiembre 2006]
- [26] Numeros de puertos [en línea] < <http://www.iana.org/assignments/port-numbers> > [Consulta 06 Setiembre 2006]
- [27] NAT [en línea] < <http://es.wikipedia.org/wiki/NAT>> [Consulta 06 Setiembre 2006]
- [28] RAM [en línea] < http://es.wikipedia.org/wiki/Memoria_RAM> [Consulta 06 Setiembre 2006]

Apéndices

Apéndice A.1 Glosario, abreviaturas y simbología

- **ITCR:** Instituto Tecnológico de Costa Rica
- **Control automático:** Área de la electrónica que automatiza sistemas para que se regulen y tomen decisiones por sí mismos.
- **Servidor:** computadora o dispositivo en red que puede ser accedido por cualquier otra computadora autorizada en una red, esta procesa datos y envía la información solicitada.
- **Cliente:** computadora que acceda a un servidor por medio de una red
- **Protocolo:** conjunto de reglas que controlan la secuencia de mensajes que ocurren durante una comunicación entre entidades que forman una red. En este contexto, las entidades de las cuales se habla son programas de computadora o automatismos de otro tipo, tales y como dispositivos electrónicos capaces de interactuar en una red.
- **IIS:** Servicios de información de Internet, en un conjunto de servicios para servidores basados en Internet usando el sistema operativo Windows [12]
- **CGI:** Interfaz común de pasarela, un mecanismo de comunicación entre el servidor Web y una aplicación externa.
- **MATLAB:** Es un programa de matemáticas creado por *The MathWorks* en 1984. Se pueden ampliar sus capacidades con Toolboxes [22]

- **Webcam:** es una pequeña cámara digital conectada a un ordenador, la cual puede capturar imágenes y transmitir las a través de Internet en directo, ya sea a una página Web o a otra u otras computadoras en forma privada
- **HTML:** acrónimo inglés de **HyperText Markup Language** (lenguaje de marcas hipertextuales), es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.
- **Script:** es un conjunto de instrucciones, sentencias de control, variables y demás elementos de programación generalmente almacenadas en un archivo de texto (pueden considerarse como un archivo de instrucciones o como un programa).
- **Switch de capa 3:** Este tipo de *switches* integran *routing* y *switching* para producir altas velocidades (medidas en millones de paquetes por segundo). Este nuevo tipo de dispositivos es el resultado de un proceso de evolución natural de las redes de área local, ya que, combinan las funciones de los *switches* capa 2 con las capacidades de los routers [18]
- **ASP:** es una tecnología del lado servidor de Microsoft® para páginas Web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS). [15]
- **Java:** es un lenguaje de programación orientado a objetos desarrollado por James Gosling y sus compañeros de Sun Microsystems al inicio de la década de 1990.
- **JRE:** o **Java Runtime Environment** (Entorno de Ejecución Java) proporciona únicamente un subconjunto del lenguaje de programación

Java sólo para ejecución. El usuario final normalmente utiliza JRE en paquetes y añadidos.

- **Stream:** es un término que describe una estrategia sobre demanda para la distribución de contenido multimedia a través del Internet.
- **Dirección IP:** es un número que identifica de manera lógica y jerárquicamente a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol) [24]
- **DNS:** (Domain Name System) es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar distintos tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP
- **Home:** posición por defecto del brazo robótico o posición predeterminada
- **YAWCAM:** software de uso libre para el manejo de una Webcam, incluye servidor web, ftp, streaming y archivo, www.yawcam.com
- **IANA:** es el acrónimo de Internet Assigned Number Authority. La Agencia de Asignación de Números Internet era el antiguo registro central de los protocolos Internet, como puertos, números de protocolo y empresa, opciones y códigos. Fue sustituido en 1998 por ICANN. [25]
- **NAT:** (Network Address Translation - Traducción de Dirección de Red) es un estándar creado por la Internet Engineering Task Force (IETF) el cual utiliza una o más direcciones IP para conectar varios computadores a otra red (normalmente a Internet), los cuales tienen una dirección IP completamente distinta (normalmente una IP no válida de Internet

definida por el RFC 1918). Por lo tanto, se puede utilizar para dar salida a redes públicas a computadores que se encuentran con direccionamiento privado o para proteger máquinas públicas [27]

- **Memoria RAM:** es el acrónimo inglés de *Random Access Memory* (memoria de acceso aleatorio). Se trata de una memoria en la que se puede tanto leer como escribir información. Es una memoria volátil, es decir, pierde su contenido al desconectar la energía eléctrica. Se utiliza normalmente como memoria temporal para almacenar resultados intermedios y datos similares no permanentes.

Apéndice A.2 Manuales de usuario

Apéndice A.2.1 Configuración de IIS

1. Instalación, Inserte el Disco de Instalación de Windows XP

1. Ire a **Start-->Settings-->Control Panel-->Add/Remove Programs**, ver figura A.2.1.
2. Click en el botón **Add/Remove Windows Components**
3. Doble click en "Internet Information Services (IIS)"
4. Al final de la lista hacer clic en **World Wide Web Server**.

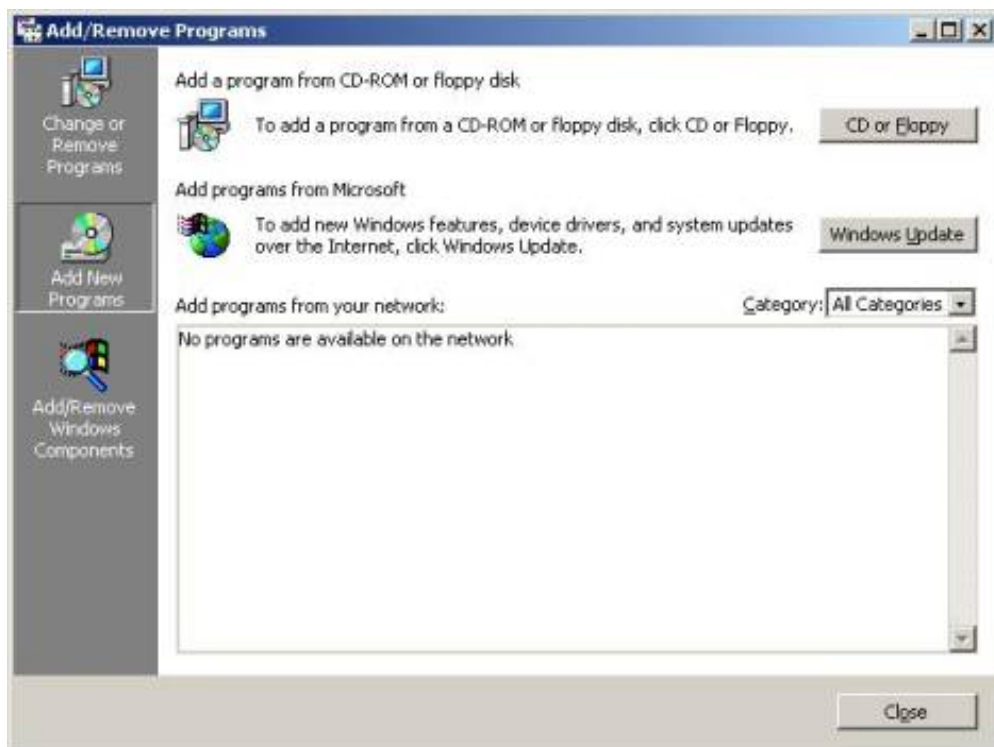


Figura A.2.1 Instalación de IIS

2. Asegúrese de que "Default website" esté corriendo:
 - a. Ira a **Start-->Settings-->Control Panel-->Administrative Tools-->Internet Services Manager**.
 - b. Asegúrese de que "Default website" esté "corriendo " ver figura A.2.2.
 - c. De lo contrario haga clic en **start**.

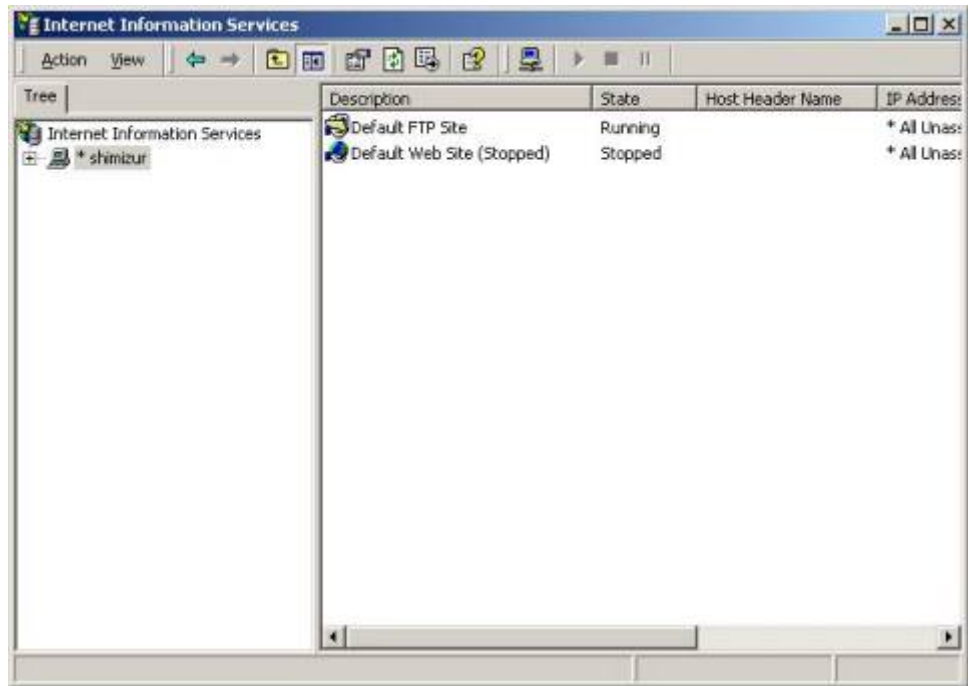


Figura A.2.2 Inicialización del "Default website"

3. Asegúrese de que se ha instalado correctamente el IIS, abriendo el siguiente sitio Web `http://localhost`.
4. Si se ha instalado correctamente aparecerá la página que se muestra en la figura A.2.3.



Figura A.2.3 Página de inicio de IIS

5. Ahora cambia las propiedades del "Default Website":

- . Abrir la consola IIS MMC en **Start-->Settings-->Control Panel-->Administrative Tools-->Internet Services Manager**.
- a. clic derecho en **Default website** y elija **Properties**.
- b. Ir a la pestaña de "Home Directory"
- c. Cambiar "Local Path" Al directorio donde se han copiado las páginas Web.
- d. Debe verse como se muestra en la figura A.2.4:



Figura A.2.4 Configuración del Home Directory de IIS

6. clic en la pestaña "Documents" y agregar el archive Inicio.asp y subirlo hasta la primer posición ver figura A.2.5.

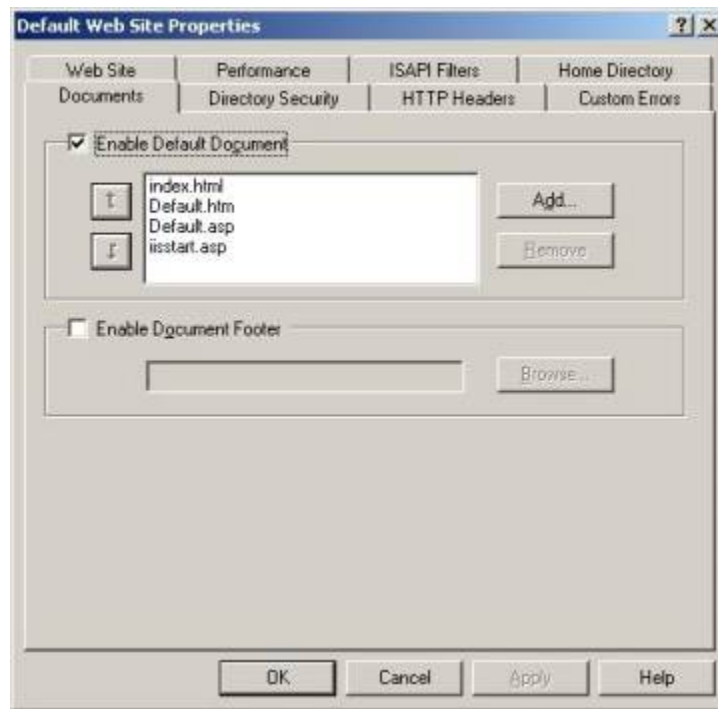


Figura A.2.5 Configuración del “Documents” de IIS

7. clic **OK** para cerrar la ventana
8. Cree 2 directorios virtuales o alias : `cgi-bin` y `icons` y póngalos a apuntar donde se han copiado las páginas Web
 - . clic derecho en **Default Website** en la consola de IIS.
 - a. clic en **New**; y luego en **Virtual Directory** Esto abrirá el “Virtual Directory Wizard”.
 - b. Siga las instrucciones escribiendo `icons` para el nombre y apuntándolo hacia `c:\MATLAB_files\` (se debe crear este directorio en C:\) Deje el resto como default
 - c. Repita lo mismo para `cgi-bin`, apuntándolo hacia el directorio donde está `Matweb.exe`, asegúrese de que los permisos estén establecidos como **ejecutable** en el último paso del *wizard*

Apéndice A.2.2 Configuración de *MATLAB* Web Server

Sustituya `$MATLAB$` con el directorio donde se ha instalado *MATLAB*

1. Configuración del puerto en el *MATLAB server.conf*: se ubica en el directorio de `$MATLAB$/webserver/` debe de incluir en un nuevo renglón el siguiente texto “-p 1710”
2. Configuración del puerto en el *matweb.conf*: se ubica en el directorio de `$MATLAB$/toolbox/webserver/wsdemos` debe de incluir todas las aplicaciones .m y el puerto de *MATLAB* Web Server de la siguiente manera:

```
[Activar]
mlserver=f3-08-01
mlport=1710
```

```
[ira]
mlserver=f3-08-01
mlport=1710
```

```
[guarda]
mlserver=f3-08-01
mlport=1710
```

```
[Maxvel]
mlserver=f3-08-01
mlport=1710
```

```
[Desactivar]
mlserver=f3-08-01
mlport=1710
```

```
[Homing]
mlserver=f3-08-01
mlport=1710
```

```
[join]
```

```
mlserver=f3-08-01
```

```
mlport=1710
```

```
[encontrar]
```

```
mlport=1710
```

```
mlserver=f3-08-01
```

```
mldir= C:/MATLAB_files
```

donde f3-08-01 es el nombre de la computadora donde esta instalado *MATLAB*, este se obtiene escribiendo en el “command prompt” *hostname*.

3. Asegúrese de que el servicio *MATLAB server* este configurado para que inicie de manera automática en **Start-->Settings-->Control Panel-->Administrative Tools-->Services**.

Apéndice A.2.3 Configuración de YAWCAM

1. Instale el Software: Si no dispone del archivo instalador, descárguelo del sitio www.yawcam.com.
2. Ejecute el programa yawcam.exe y en la pestaña de configuración elija editar configuración. Debe de haber abierto la ventana de “configuración”.
3. Para configurar la captura de la imagen que va a ser procesada por *MATLAB* elija la categoría *Salida/Archivo* y verifique que el archivo de salida sea `c:\temp\out.jpg` y que el intervalo sea de 5 segundos
4. Para configurar el puerto de red de salida de la webcam, en la categoría *Salida/Stream* Escriba el puerto “1711”
5. Para que la fecha y hora no aparezca a la hora de tomar la imagen para ser procesada, en la categoría *Superposición* borre el texto del campo que dice “#1” y deshabilite la opción de usar fondo.
6. Para que inicie la captura de la imagen y el stream de la webcam, en la categoría *Inicio* habilite únicamente las opciones “Iniciar Archivo” e “Iniciar Stream” y deshabilite la opción de “Comprobar Nuevas versiones”
7. Pulse el botón aceptar para cerrar la ventana de configuración
8. Para que el programa no se muestre en la barra de programas, en la pestaña de *ver* se debe habilitar la opción de “ocultar al iniciar”
9. Para hacer que el programa inicie automáticamente al iniciar Windows copie un acceso directo del archivo yawcam.exe en “C:\Documents and Settings\All Users\Start Menu”

Apéndice A.2.4 Manual de usuario de las páginas web

Al ingresar //ierobot/ en el explorador de Internet en una computadora de la red de escuela ingeniería electrónica se accede a la página para iniciar sesión, introduzca un nombre de usuario y contraseña válidos, luego es enviado a la pantalla de inicio (Figura A.2.1), aquí, si desea mover el brazo robótico, memorizar posiciones del brazo robótico, habilitar, deshabilitar o llevar el brazo robótico a la posición de *home* debe hacer clic en “Mover brazo robótico”, si desea encontrar figuras en 2 Dimensiones haga clic en “Encontrar Figuras”



Figura A.2.1 Inicio.asp

Apéndice A.3 Información sobre la institución

Apéndice A.3.1 Descripción de la institución

El proyecto se lleva a cabo en el Instituto Tecnológico de Costa Rica, el cual es una institución nacional autónoma de educación superior universitaria, dedicada a la docencia, la investigación y la extensión de la tecnología y ciencias conexas para el desarrollo de Costa Rica. Fue creado mediante Ley No. 4777 del 10 de junio de 1971. El Instituto Tecnológico de Costa Rica, tiene como misión la de: "Contribuir al desarrollo integral del país, mediante la formación de recursos humanos, la investigación y la extensión; manteniendo el liderazgo científico-tecnológico y técnico, la excelencia académica y el estricto apego a las normas éticas, humanistas y ambientales, desde una perspectiva universitaria estatal de calidad y competitividad a nivel nacional e internacional" [5]

Apéndice A.3.2 Descripción del departamento en el que se realizó el proyecto

El proyecto se lleva a cabo en la Escuela de Ingeniería en Electrónica, cuyo objetivo es formar profesionales en la ingeniería en electrónica.

Teniendo como ramas principales de estudio:

- Sistemas digitales
- Comunicaciones Eléctricas
- Electrónica de potencia
- Control automático
- Telemática

Por la calidad de sus egresados la Escuela de Ingeniería en Electrónica es reconocida como la mejor en Costa Rica en su campo.

Apéndice A.4 Rutinas de programación

Apéndice A.4.1 Programas de *MATLAB*

Apéndice A.4.1.1 Activar.m:

```
function retstr = Activar(h)
if ( h.user == 'Nivel1' | h.user == 'Nivel2' | h.user == 'Nivel3' |
h.user == 'Nivel4')
P=serial('COM1','Terminator',{'LF/CR','LF/CR'},'BaudRate',9600,'DataBi
ts',8);
fopen(P)
fprintf(P,'en po');

fclose(P)
salida = ['Brazo robótico activado '];

else
salida = ['Usuario ' h.user ', Unicamente los usuarios de Nivel 4 o
menor tiene derecho a cambiar este parámetro '];
end
outstruct.status = salida;
templatefile = which('status.html');
if (nargin == 1)
retstr = htmlrep(outstruct, templatefile);
elseif (nargin == 2)
retstr = htmlrep(outstruct, templatefile, outfile);
end
```

Apéndice A.4.1.2 Desactivar.m:

```
function retstr = Desactivar(h)

if ( h.user == 'Nivel1' | h.user == 'Nivel2' | h.user == 'Nivel3' |
h.user == 'Nivel4')

P=serial('COM1','Terminator',{'LF/CR','LF/CR'},'BaudRate',9600,'
DataBits',8);
fopen(P)
fprintf(P,'dis po');

fclose(P)
salida = ['Brazo robótico desactivado '];
else
salida = ['Usuario ' h.user ', Unicamente los usuarios de Nivel 4 o
menor tiene derecho a cambiar este parámetro '];
end
outstruct.status = salida;

templatefile = which('status.html');
if (nargin == 1)
retstr = htmlrep(outstruct, templatefile);
elseif (nargin == 2)
retstr = htmlrep(outstruct, templatefile, outfile);
end
```

Apéndice A.4.1.3 Join.m:

```
function retstr = join(h)

jo = h.joint;
angle = h.AZ;
ang = str2double(h.AZ);
j = str2double(h.joint);
speed = h.veloc;

command = ['do drive ' h.joint ', ' h.AZ ', ' h.veloc];

if ( h.user == 'Nivel1' | h.user == 'Nivel3' | h.user == 'Nivel4')
    P=serial('COM1','Terminator',{ 'LF/CR','LF/CR'}, 'BaudRate',9600, '
DataBits',8);
    fopen(P);

    fprintf(P, 'where');
    Ax = fscanf(P,'%c',275);
    js2=[Ax(225) Ax(226) Ax(227) Ax(228)];
    js3=[Ax(235) Ax(236) Ax(237) Ax(238)];

    j2 = str2double(js2)+90;
    j3 = str2double(js3)-90;

    jf23 = j2 + j3+ ang;
    jf2 = j2+ ang;
    jf3 = j3+ ang;

    if( (abs(jf23) >= 180 | abs(jf2) >= 115) & ((j==2) | (j==3)))
        fclose(P);
        salida = ['Movimiento no permitido, peligro de golpear el
brazo robótico <br/> Cambie el angulo a mover en la articulacion 2 o
3'];
    else
        fprintf(P,command);
        fclose(P);
        salida = ['Articulación ' h.joint ' ha sido movida ' h.AZ '
grados '];
    end

else
    salida = ['Usuario ' h.user ', Unicamente los usuarios de Nivel
1,3 o 4 tiene derecho a utilizar este parámetro '];
end

outstruct.status = salida;

templatefile = which('status.html');
if (nargin == 1)
    retstr = htmlrep(outstruct, templatefile);
elseif (nargin == 2)
    retstr = htmlrep(outstruct, templatefile, outfile);
end
```

Apéndice A.4.1.4 Guarda.m:

```
function retstr = guarda(h)

posi = h.pos;

command = ['here P' posi];

if ( h.user == 'Nivell' | h.user == 'Nivel4')

P=serial('COM1','Terminator',{ 'LF/CR','LF/CR'}, 'BaudRate',9600,'DataBi
ts',8);
fopen(P);
fprintf(P,command);
pause(3);
fprintf(P,'');
pause(3);
fprintf(P,'do ready');
pause(3);
fprintf(P,['do move P' posi])
pause(3);
fprintf(P,'do ready');
fclose(P);

salida = ['Se ha salvado la actual posición del brazo en ' posi];

else
salida = ['Usuario ' h.user ', Unicamente los usuarios de Nivel 1 o 4
tiene derecho a modificar este parámetro '];
end

outstruct.status = salida;

templatefile = which('status.html');
if (nargin == 1)
    retstr = htmlrep(outstruct, templatefile);
elseif (nargin == 2)
    retstr = htmlrep(outstruct, templatefile, outfile);
end
```

Apéndice A.4.1.5 Ira.m:

```
function retstr = ira(h)

posi = h.pos;
```

```

command = ['do move P' posi];

if ( h.user == 'Nivel1' | h.user == 'Nivel3' | h.user == 'Nivel4')
P=serial('COM1','Terminator',{ 'LF/CR','LF/CR'}, 'BaudRate',9600,'DataBi
ts',8);
fopen(P);
fprintf(P,command);
fclose(P);
salida = ['Se ha movido el brazo robótico a la posición ' posi];

else
salida = ['Usuario ' h.user ', Unicamente los usuarios de Nivel 1,3 o
4 tiene derecho a utilizar este parámetro '];
end

outstruct.status = salida;

templatefile = which('status.html');
if (nargin == 1)
    retstr = htmlrep(outstruct, templatefile);
elseif (nargin == 2)
    retstr = htmlrep(outstruct, templatefile, outfile);
end

```

Apéndice A.4.1.6 Maxvel.m:

```

function retstr = Maxvel(h)

maxveloc = h.MV;

command = ['speed ' h.MV];
%salida = ' '

if ( h.user == 'Nivel1' | h.user == 'Nivel1')
P=serial('COM1','Terminator',{ 'LF/CR','LF/CR'}, 'BaudRate',9600,'DataBi
ts',8);
fopen(P);
fprintf(P,command);
fclose(P);
salida = ['Velocidad cambiada a ' h.MV];
else
salida = ['Usuario ' h.user ', Unicamente los usuarios de Nivel 1
tiene derecho a cambiar este parámetro '];
end

outstruct.status = salida;

templatefile = which('status.html');
if (nargin == 1)
    retstr = htmlrep(outstruct, templatefile);

```

```

elseif (nargin == 2)
    retstr = htmlrep(outstruct, templatefile, outfile);
end

```

Apéndice A.4.1.7 Encontrar.m:

```

function rs = encontrar(h)

if ( h.user == 'Nivel1' | h.user == 'Nivel2')

% Get unique identifier (to form file name)
mlid = getfield(h, 'mlid');

% Set directory path for storage of graphic files.
cd(h.mldir);

% Cleanup jpegs older than 1 hour.
wscleanup('ml*capture.jpeg', 1);

P=serial('COM1','Terminator',{ 'LF/CR','LF/CR'}, 'BaudRate',9600, 'DataBi
ts',8);

user = getenv('UserName')
disp(user)

F = h.Figura

st= 0
st(1)=0
st(2)= 0
st(3)=0
st(4)= 0

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5

im1=im2bw(imread('c:/temp/out.jpg', 'jpg'));
im2=double(im1);
im3=edge(im2);
im4 = imfill(im3,'holes');
[B,L,N,A] = bwboundaries(im3);
    imshow(im4);

for cnt = 1:N
hold on;
boundary = B{cnt};
plot(boundary(:,2), boundary(:,1), 'r');
hold on;
text(mean(boundary(:,2)),
mean(boundary(:,1)),num2str(cnt), 'FontSize',14, 'Color', 'red');
end

```

```

for cnt = 1:N
boundary = B{cnt};
[th, r]=cart2pol(boundary(:,2)-mean(boundary(:,2)), ...
boundary(:,1)-mean(boundary(:,1)));
st(cnt)=std(r/max(r));%Arreglo de desviaciones estandar
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5

switch F
case 'Circulo'
vmin=0;
vmax=0.05;
case 'Cuadrado'
disp('Un cuadrado?')
vmin=0.06;
vmax=0.095; %Colocar los valores de vmin y
vmax
case 'Triangulo'
disp('Un triangulo?')
vmin=0.13;
vmax=0.16;
case 'Rectangulo'
disp('Un rectangulo?')
vmin=0.2;
vmax=0.3;
otherwise
disp(['La forma ' F ' es desconocida']);
vmin=-10;
vmax=-2;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%555

status = ' ';

for cuenta = 1:4
pos=num2str(cuenta);
if( (vmin < st(cuenta))&& (st(cuenta)< vmax))
fopen(P);
fprintf(P,'en po');
fprintf(P,['do move P' pos]);%Ubicacion del objeto
status=( [status ' La posicion ' pos ' SI contiene un ' F
'</p>'] );
pause(3);%Se detiene 6 seg
fprintf(P,'do ready');
pause(3);
fclose(P);
else
status=( [status ' La posicion ' pos ' NO contiene un ' F
'</p>'] );
end

end
end

```

```
sal=status;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%Render jpeg and write to file.
```

```
drawnow;
```

```
s.GraphFileName = sprintf('%scapture.jpeg', mlid);
```

```
imwrite(im4, s.GraphFileName, 'jpeg' );
```

```
s.GraphFileName = sprintf('/icons/%scapture.jpeg', mlid);
```

```
close all;
```

```
else
```

```
sal = ['Usuario ' h.user ', Unicamente los usuarios de Nivel 1 o 2  
tienen derecho a acceder este parámetro '];
```

```
end
```

```
s.salida = sal;
```

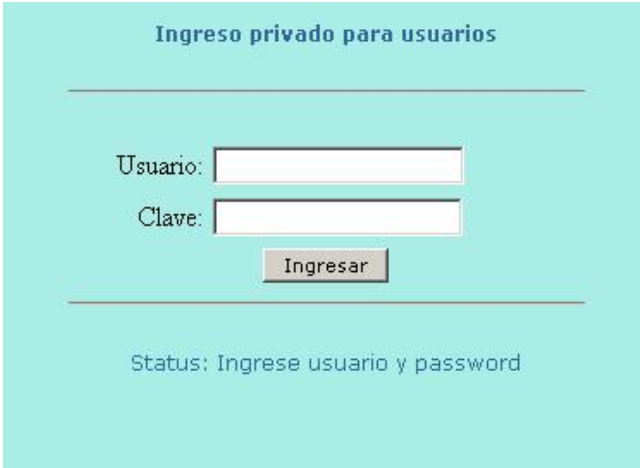
```
% Put name of graphic file into HTML template file.
```

```
templatefile = which('captura2.html');
```

```
rs = htmlrep(s, templatefile);
```

Apéndice A.4.2 Código HTML y ASP

Apéndice A.4.2.1 Index.asp:



Ingreso privado para usuarios

Usuario:

Clave:

Ingresar

Status: Ingrese usuario y password

Figura A.4.1 Index.asp

```
<%
if request.form("usuario") <> "" and request.form("clave") <> ""then
    'Los contenidos de las siguientes variables son los aceptados
    por el script
    'Cambiarlos para modificar el usuario y password válidos...
    user1 = "user1"
    pass1 = "123"
    user2 = "user2"
    pass2 = "123"
    user3 = "user3"
    pass3 = "123"
    user4 = "user4"
    pass4 = "123"
    user5 = "user5"
    pass5 = "123"

    if (Lcase(request.form("usuario")) = LCase(user1)) AND
(LCase(request.form("clave")) = Lcase(pass1)) then
        session("autorizado") = "Nivel1"
        response.redirect "inicio.asp"
    else
        if (Lcase(request.form("usuario")) = LCase(user2)) AND
(LCase(request.form("clave")) = Lcase(pass2)) then
            session("autorizado") = "Nivel2"
            response.redirect "inicio.asp"
        else
            if (Lcase(request.form("usuario")) = LCase(user3)) AND
(LCase(request.form("clave")) = Lcase(pass3)) then
                session("autorizado") = "Nivel3"
```



```

        response.redirect "inicio.asp"
    else
        if (LCase(request.form("usuario")) = LCase(user4)) AND
(LCase(request.form("clave")) = LCase(pass4)) then
            session("autorizado") = "Nivel4"
            response.redirect "inicio.asp"
        else
            if (LCase(request.form("usuario")) = LCase(user5)) AND
(LCase(request.form("clave")) = LCase(pass5)) then
                session("autorizado") = "Nivel5"
                response.redirect "inicio.asp"
            else

                mensaje = "Error de logueo"
            end if
        end if
    end if
end if

mensaje = "Ingrese usuario y password"
end if
%>
<HTML>
<BODY BGCOLOR="#AAECE6">
<P ALIGN="center" STYLE="font-family:Verdana;font-
size:12;color:#336699;"><B>Ingreso privado para usuarios</B></P>
<HR WIDTH="300"><BR>
<TABLE WIDTH="300" ALIGN="center" BORDER="0" CELLPADDING="2"
CELLSPACING="2">
<FORM METHOD="post" ACTION="<%=
Request.ServerVariables("SCRIPT_NAME")%>">
<TR><TD ALIGN="right">Usuario:</TD><TD><INPUT TYPE="Text"
NAME="usuario"></TD></TR>
<TR><TD ALIGN="right">Clave:</TD><TD><INPUT TYPE="Password"
NAME="clave"></TD></TR>
<TR><TD ALIGN="center" COLSPAN="2"><INPUT STYLE="font-family:Verdana;
font-size:11px" TYPE="Submit" VALUE="Ingresar"></TD></TR>
</FORM>
</TABLE>
<HR WIDTH="300">
<P ALIGN="center" STYLE="font-family:Verdana;font-
size:12;color:#336699;">Status: <%=mensaje%></P>
</BODY>
</HTML>

```

Apéndice A.4.2.2 Inicio.asp:



Figura A.4.2 Inicio.asp

```
<%
'este IF es el que verifica que se haya pasado por el formulario de
login
if session("autorizado") = "" then response.redirect "index.asp"%>

<HTML>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
<meta name="GENERATOR" content="Microsoft® FrontPage 2.0">
<title>Acceso remoto al brazo robótico Straubli RX-90</title>
</head>

<body bgcolor="#FFFFFF">
<P ALIGN="center" STYLE="font-family:Verdana;font-
size:12;color:#336699;">Bienvenido <B><%= session("autorizado")%></B>
</P>

<p><font color="#000000" size="4" face="Arial"><i></i></font> <font
size="5"><strong><b>Acceso remoto al brazo robótico Straubli RX-
90</b></strong></font> </p>

<ul>
<li><a href="join1.asp">Mover brazo robótico</a>
Mueve una articulación del brazo robótico
a una velocidad y posición definida.</li>
</ul>
```

```
<p> </p>
```

```
<ul>
```

```
  <li><a href="captura1.asp">Encontrar Figuras</a>  
    Encuentra una figura definida en alguna posición  
definida.</li>
```

```
</ul>
```

```
<p> </p>
```

```
<p> </p>
```

```
<p><font color="#0000FF" size="2" face="Arial">©2006 Escuela de  
Ingeniería Electrónica, Instituto Tecnológico de Costa Rica.</font>  
</p>  
</body>  
</HTML>
```

Apéndice A.4.2.3 Join1.asp:

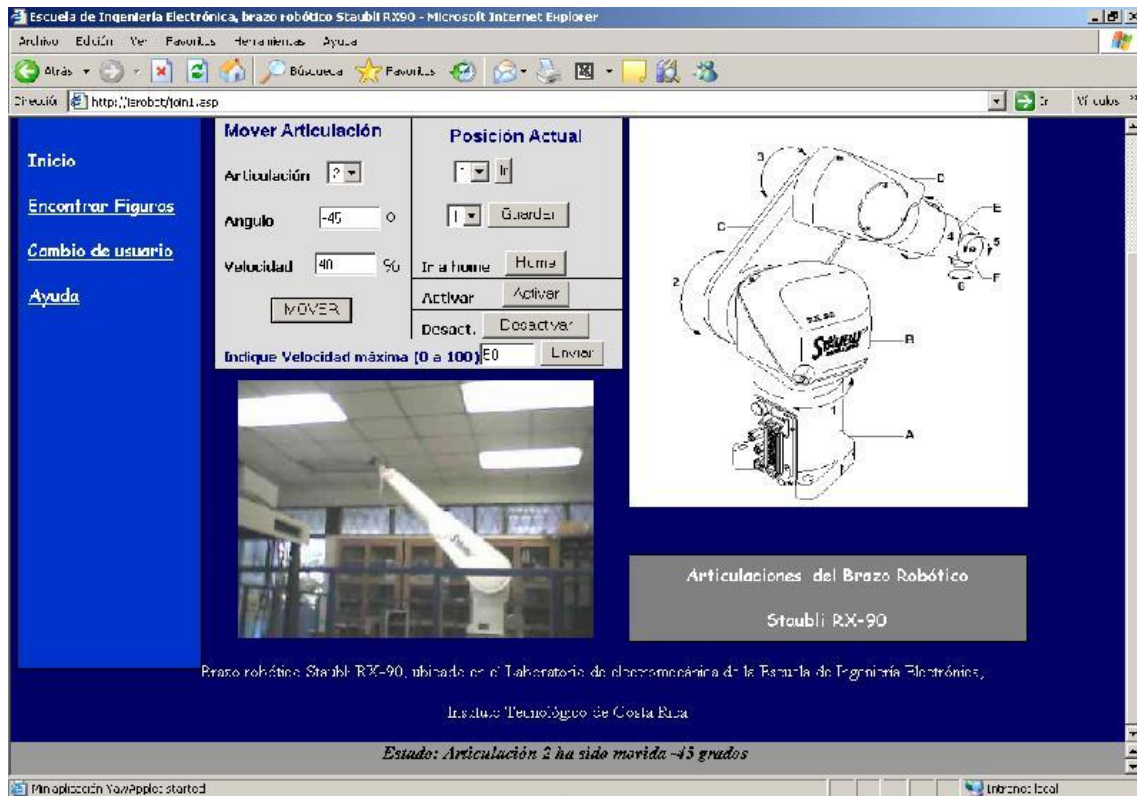


Figura A.4.3 Join1.asp

```

<%
'este IF es el que verifica que se haya pasado por el formulario de
login
'poner el mismo IF en todas las páginas restringidas.
if session("autorizado") = "" then response.redirect "index.asp"%>

<!-- $Revision: 1.2 $ -->
<HTML>
<HEAD>
<TITLE>Escuela de Ingeniería Electrónica, brazo robótico Staubli RX90
</TITLE>
</HEAD>
<frameset rows="529,*" border=0>
<p align="right"><b><font face="Batang" color="#FFFFFF">ESCUELA DE
INGENIERÍA
ELECTRÓNICA</font></b></p>

<frame scrolling=auto
src="join.asp"
name="topframe"
marginwidth=10
marginheight=0
frameborder=0>

<frame scrolling=auto
src="status.html"

```

```

marginwidth=0 marginheight=0
name="outputwindow" frameborder=0>

</frameset>

</body>
</HTML>

```

Apéndice A.4.2.4 Join.asp:

```

<%
'este IF es el que verifica que se haya pasado por el formulario de
login
if session("autorizado") = "" then response.redirect "index.asp"%>

<!-- $Revision: 1.2 $ -->
<html xmlns:v="urn:schemas-Microsoft@-com:vml" xmlns:o="urn:schemas-
Microsoft@-com:office:office" xmlns="http://www.w3.org/TR/REC-html40">
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
<link rel="File-List" href="Join_archivos/filelist.xml">

<!--[if !mso]>
<style>
v\:*      { behavior: url(#default#VML) }
o\:*      { behavior: url(#default#VML) }
.shape    { behavior: url(#default#VML) }
</style>
<![endif]--><!--[if gte mso 9]>
<xml><o:shapedefaults v:ext="edit" spidmax="1027"/>
</xml><![endif]-->

<body bgcolor="#000066">

<p align="right"><!--[if gte vml 1]><v:shapetype id="_x0000_t202"
  coordsize="21600,21600" o:spt="202" path="m,l,21600r21600,l21600,xe">
  <v:stroke jointstyle="miter"/>
  <v:path gradientshapeok="t" o:connecttype="rect"/>
</v:shapetype><v:shape id="_x0000_sl044" type="#_x0000_t202"
style='position:absolute;
  left:4.5pt;top:104.25pt;width:124.5pt;height:388.5pt;z-index:1'
fillcolor="#03c pure">
  <v:textbox>
<table cellpadding="0" cellspacing="0" width="100%" height="100%">
  <tr>
    <td align="center">
      <p align="left"><font face="Comic Sans MS"><a href="inicio.asp">
        <font color="#FFFFFF">Inicio</font></a></font></p>
      <p align="left"><font face="Comic Sans MS"><a href="captural.asp">
        <font color="#FFFFFF">Encontrar Figuras</font></a></font></p>
      <p align="left"><font face="Comic Sans MS"><a href="index.asp">
        <font color="#FFFFFF">Cambio de usuario</font></a></font></p>
      <p align="left"><font face="Comic Sans MS" color="#FFFFFF">
        <a href="Ayda.asp"><font
color="#FFFFFF">Ayuda</font></a></font></p>
      <p align="left">&nbsp;</p>
      <p>&nbsp;</p>
      <p>&nbsp;</p>

```



```

top:166.5pt;width:117pt;height:21.75pt;z-index:2' fillcolor="#ddd"
stroked="f">
<v:textbox inset=",0,,0">
<table cellspacing="0" cellpadding="0" width="100%" height="100%">
  <tr>
    <form action="/cgi-bin/matweb.exe" method="POST"
target="outputwindow">
      <input type="hidden" name="mlmfile" value="Guarda">
      <input type="hidden" size="1" name="user" value="<%=
session("autorizado")%>">
      <td align="center"><font face="Tahoma"><b>
<p><select name="pos">
<option selected>1
<option>2
<option>3
<option>4
<option>5
<option>6</select></b></font>&nbsp;<input type="submit"
value="Guardar">&nbsp;&nbsp;&nbsp;</p>
      <p align="left">&nbsp;</td>
    </form>
  </tr>
</table>
</v:textbox>
</v:shape><![endif]--><![if !vml]><span style='mso-
ignore:vglayout;position:
absolute;z-index:2;left:376px;top:222px;width:160px;height:33px'></span><![endif]><!--[if gte vml 1]><v:shape
id="_x0000_s1036" type="#_x0000_t202" alt=""
style='position:absolute;left:270.75pt;
top:241.5pt;width:142.5pt;height:21.75pt;z-index:2' fillcolor="#ddd">
<v:textbox inset=",0,,0">
<table cellspacing="0" cellpadding="0" width="100%" height="100%">
  <tr>
    <form action="/cgi-bin/matweb.exe" method="POST"
target="outputwindow">
      <input type="hidden" name="mlmfile" value="Desactivar">
      <input type="hidden" size="1" name="user" value="<%=
session("autorizado")%>">
      <td align="center">
      <p align="left"><b><font face="Tahoma"><font
size="2">Desact.</font>
      </font></b><font face="Tahoma"><b><input type="submit"
value="Desactivar"></td>
    </tr>
  </form>
</table>
</v:textbox>
</v:shape><![endif]--><![if !vml]><span style='mso-
ignore:vglayout;position:
absolute;z-index:2;left:360px;top:321px;width:196px;height:35px'></span><![endif]><!--[if gte vml 1]><v:shape
id="_x0000_s1037" type="#_x0000_t202" alt=""
style='position:absolute;left:137.25pt;
top:261.75pt;width:276pt;height:21.75pt;z-index:2' fillcolor="#ddd">
<v:textbox inset=",0,,0">
<table cellspacing="0" cellpadding="0" width="100%" height="100%">
  <tr>

```



```

<HTML>
<HEAD>
<TITLE>Captura de imagen</TITLE>
</HEAD>

<frameset rows="350, *" border=0>

  <frame scrolling=auto
    src="imcaptura.asp"
    name="topframe"
    marginwidth=10
    marginheight=1
    frameborder=0>

  <frame scrolling=auto
    src="bajo.html"
    marginwidth=0 marginheight=0
    name="outputwindow" frameborder=0>

</frameset>
</body>

</HTML>

```

Apéndice A.4.2.6 Imcaptura1.asp:

```

<link rel="File-List" href="imcaptura_archivos/filelist.xml">
<!--[if !mso]>
<style>
v\:*      { behavior: url(#default#VML) }
o\:*      { behavior: url(#default#VML) }
.shape    { behavior: url(#default#VML) }
</style>
<![endif]-->
<%
'este IF es el que verifica que se haya pasado por el formulario de
login
'poner el mismo IF en todas las páginas restringidas.
if session("autorizado") = "" then response.redirect "index.asp"%>

<html xmlns:v="urn:schemas-Microsoft@-com:vml" xmlns:o="urn:schemas-
Microsoft@-com:office:office" xmlns="http://www.w3.org/TR/REC-html40">
<!--[if gte mso 9]>
<xml><o:shapedefaults v:ext="edit" spidmax="1027"/>
</xml><![endif]-->

<body bgcolor="#000080">
<P ALIGN="center" STYLE="font-family:Verdana;font-
size:12;color:#336699;">
<font color="#FFFFFF">Bienvenido <B><%=
session("autorizado")%></B></font> </P>

<center>
<table border=0>

```

```

<td bgcolor=#DDDDDb valign=top>
<center><b><font size="5">Buscar Figura</font></b></center>

    <APPLET CODE=YawApplet.class WIDTH="320" HEIGHT="240">
      <param name="Host" value="172.21.104.65">
      <param name="Port" value="1711">
    </APPLET>

<form action="/cgi-bin/matweb.exe" method="POST"
target="outputwindow">
  <input type="hidden" name="mlmfile" value="encontrar">

<center>

<select name=Figura>
<option selected>Circulo
<option>Cuadrado
<option>Triangulo
<option>Rectangulo
</select>
<input type="hidden" size="1" name="user" value=<%=
session("autorizado")%>>
<input type="submit" value="Encontrar">
</dir>

</center>
</form>
<p><!--[[if gte vml 1]]><v:shapetype id="_x0000_t202"
  coordsize="21600,21600" o:spt="202" path="m,l,21600r21600,121600,xe">
  <v:stroke joinstyle="miter"/>
  <v:path gradientshapeok="t" o:connecttype="rect"/>
</v:shapetype><v:shape id="_x0000_s1027" type="#_x0000_t202"
style='position:absolute;
  left:519.75pt;top:36.75pt;width:177.75pt;height:117pt;z-index:1'
fillcolor="#03c pure">
  <v:textbox>
<table cellpadding="0" cellspacing="0" width="100%" height="100%">
  <tr>
    <td align="center">
      <p align="left">&nbsp;</p>
      <p align="justify">&nbsp;<font color="#FFFFFF" size="5">Elija la
figura que desea
      busque el brazo robótico y pulse encontrar</font></p>
      <p>&nbsp;</p>
      <p>&nbsp;</p>
      <p>&nbsp;</p>
      <p>&nbsp;</p>
      <p>&nbsp;</p>
      <p>&nbsp;</p>
      <p>&nbsp;</p>
    </td>
  </tr>
</table>
  </v:textbox>
</v:shape><![endif]--><![if !vml]><span style='mso-
ignore:vglayout;position:
absolute;z-index:1;left:692px;top:48px;width:243px;height:162px'><img

```


Apéndice A.5 Resolución del problema cinemática directo para el robot Staubli RX-90 [10]

Apéndice A.5.1 Numeración de las articulaciones y eslabones

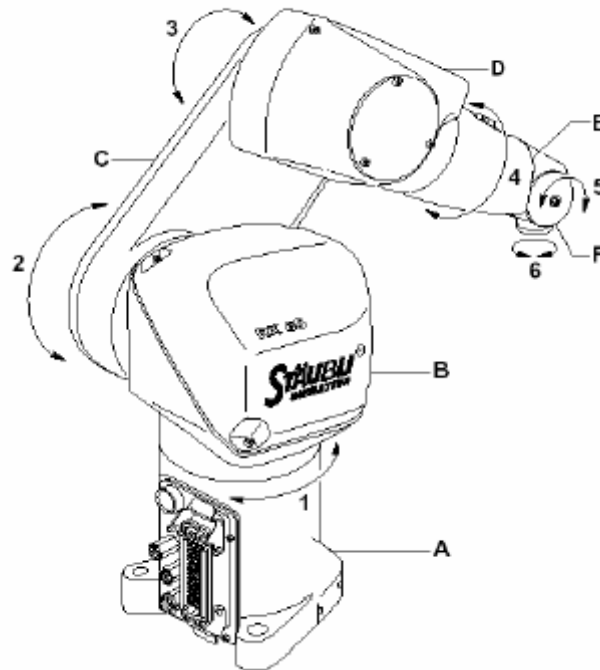


Figura A.5.1. Articulaciones del Brazo robótico Staubli RX-90

Apéndice A.5.2 Obtención de los parámetro D-H para el brazo robótico Stauli RX-90

Tabla A.5.1 Parámetros D-H para el robot Staubli RX-90

Articulación	θ	d	a	α
1	Θ_1	0	0	0
2	Θ_2	0	0	-90
3	Θ_3	l_3	l_2	0
4	Θ_4	l_4	0	-90
5	Θ_5	0	0	90
6	Θ_6	0	0	-90

Valor de los parámetros (mm)

$l_2 = 450$

$$l_3 = 0$$

$$l_4 = 735$$

Apéndice A.5.3 Obtención de las matrices de transformación ${}^{i-1}T_i$

Expresión general de Denavit Hartenberg

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.5.1})$$

Donde $c\theta_i = \cos \theta_i = c_i$ y $s\theta_i = \sin \theta_i = s_i$

Con los datos obtenidos en la tabla A.5.1 se calculan las matrices T , sustituyendo en la expresión general del algoritmo de Denavit-Hartenberg [Ecuación A.5.1] se obtiene lo siguiente (tomado de [17], pag. 7)

$$\begin{aligned} {}^0_1T &= \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^3_4T &= \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ -s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^1_2T &= \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^4_5T &= \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ {}^2_3T &= \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_2 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^5_6T &= \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Apéndice A.5.4 Obtención de la matriz de transformación T

Así se puede calcular la matriz **T** que indica la localización del extremo del brazo robótico con respecto a la base del mismo.

$$T = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.5.2})$$

Ahora se obtiene la matriz 0T_6 multiplicando cada una de las matrices mostradas anteriormente (tomado de [17], pag. 7) :

$$\begin{aligned} P_x &= C_1 (l_2 \cdot C_2 - l_4 \cdot S_{23}) \\ P_y &= S_1 (l_2 \cdot C_2 - l_4 \cdot S_{23}) \\ P_z &= l_2 \cdot C_2 + l_4 \cdot S_{23} \end{aligned}$$

Con estas ecuaciones se obtiene la posición del extremo del brazo robótico Staubli RX-90 en función de las coordenadas de cada articulación ($\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5, \Theta_6$) donde:

$$\cos \theta_1 = C_1$$

$$\sin \theta_1 = S_1$$

$$C_{23} = C_2 C_3 - S_2 S_3$$

$$S_{23} = C_2 S_3 + S_2 C_3$$

Anexos

Anexo B.1 Programa en *MATLAB* para el reconocimiento de Figuras en 2D

```
function varargout = Finall(varargin)

vid = videoinput('winvideo');
vidRes = get(vid, 'VideoResolution');
nBands = get(vid, 'NumberOfBands');
hImage = image( zeros(vidRes(2), vidRes(1), nBands) );

video = handles.Finall;
frame = getsnapshot(video);
axes(handles.axes2);
imshow(frame);
im1=im2bw(frame);
im2=double(im1);
im3=edge(im2);
axes(handles.axes3);
%imshow(im3);
im4 = imfill(im3,'holes');
[B,L,N,A] = bwboundaries(im3);
imshow(im4);

for cnt = 1:N
hold on;
boundary = B{cnt};
plot(boundary(:,2), boundary(:,1), 'r');
hold on;
text(mean(boundary(:,2)),
mean(boundary(:,1)), num2str(cnt), 'FontSize',14, 'Color', 'red');
end

for cnt = 1:N
boundary = B{cnt};
[th, r]=cart2pol(boundary(:,2)-mean(boundary(:,2)), ...
boundary(:,1)-mean(boundary(:,1)));
st(cnt)=std(r/max(r));%Arreglo de desviaciones estandar
end

handles.pushbutton2 =st;
guidata(hObject,handles)

switch F
case 'circulo'
disp('Un circulo?')
vmin=0;
vmax=0.05;
case 'cuadrado'
disp('Un cuadrado?')
vmin=0.06;
vmax=0.095; % vmax
```

```

case 'triangulo'
    disp('Un triangulo?')
    vmin=0.13;
    vmax=0.16;
case 'rectangulo'
    disp('Un rectangulo?')
    vmin=0.2;
    vmax=0.3;
otherwise
    disp(['La forma ' F ' es desconocida']);
    vmin=-10;
    vmax=-2;
end

for cuenta = 1:4
    pos=num2str(cuenta);
    if( (vmin < st(cuenta))&& (st(cuenta)< vmax))

        %fprintf(P,'en po');
        fprintf(P,['do move P' pos]);%Ubicacion del objeto
        disp([' La posicion ' pos ' SI contiene un ' F]);
        pause(6);%Se detiene 6 seg
        fprintf(P,'do ready');
        pause(6);
    else
        disp([' La posicion ' pos ' NO contiene un ' F]);

    end

end

end

```