

# MANIPULACIÓN DE UN DISPOSITIVO HÁPTICO A DISTANCIA

***Jorge Gudiño Lau***

Facultad de Ingeniería Electromecánica, Universidad de Colima  
*jglau@ucol.mx*

***Felipe de Jesús Ríos Cortes***

Facultad de Ingeniería Electromecánica, Universidad de Colima  
*frcortez@ucol.mx*

***Miguel Duran Fonseca***

Facultad de Ingeniería Electromecánica, Universidad de Colima  
*mduran@ucol.mx*

***Saida Miriam Charre Ibarra***

Facultad de Ingeniería Electromecánica, Universidad de Colima  
*scharre@ucol.mx*

## Resumen

Este artículo describe un sistema de teleoperación formado por dos dispositivos hápticos, que son manipulados por un control no lineal proporcional derivativo con compensador de gravedad (PD+g). También aborda de forma detallada la configuración de la red LAN para la comunicación entre los dispositivos. La programación se desarrolló en Matlab, tanto para la comunicación como el control. Además, los resultados experimentales de la teleoperación son mostrados en este artículo, para demostrar que la comunicación funciona correctamente, así como el control.

**Palabra(s) Clave(s):** Control no lineal, dispositivo háptico, robots manipuladores, Teleoperación.

## 1. Introducción

Un sistema de teleoperación por lo general consiste de dos robot manipuladores que están conectados de tal manera que se permite que el operador humano manipule uno directamente, que se llama maestro, al generar señales que se asignan a la distancia al manipulador, al cual se le llama esclavo. [1]

La figura 1 muestra el esquema general de un sistema de teleoperación. Se entiende por teleoperación el caso en el que un operador humano, remotamente ubicado respecto al robot, lo manipula. Esta teleoperación resulta incompatible con una autonomía robótica, entendida como el caso en que el control y toma de decisiones se realizan por el robot mismo [2].

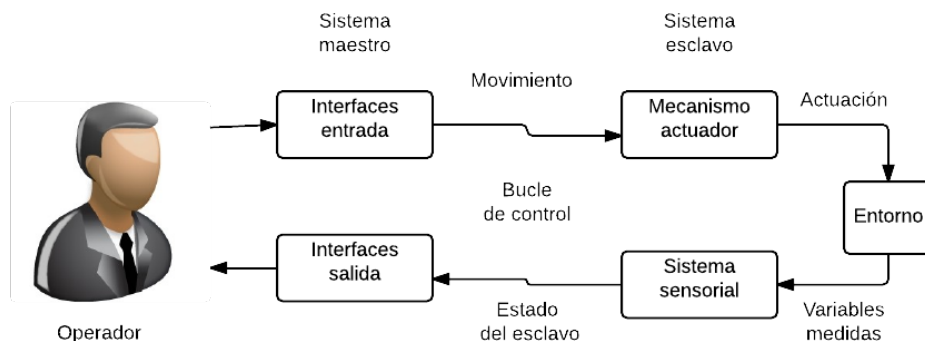


Figura 1 Esquema general de un sistema de teleoperación.

Los robots móviles teleoperados son ampliamente utilizados para llevar a cabo tareas complejas en ambientes peligrosos: algunos ejemplos conocidos son la inspección de estructuras subacuáticas, limpieza de plantas nucleares, o la remoción de minas. [3]

El diseño de la interfaz hombre-máquina suele ser crítico. Por lo cual normalmente rehace en el operador las tareas de toma de decisiones con base a la información sensorial, experiencia y habilidad. [4]

En 1947 nacen las primeras investigaciones referentes a teleoperación. En 1948 se obtuvo el desarrollo del primer manipulador teleoperado mecánico, denominado M1. Ambos manipuladores eran prácticamente iguales, como se muestra en la figura 2 [5].



Figura 2 Primer telemanipulador maestro-esclavo mecánico.

En los años 50 se obtuvieron las primeras investigaciones dirigidas a motorizar ambos manipuladores, maestro y esclavo. En 1954 Goertz presenta el primer manipulador maestro-esclavo con accionamiento eléctrico y servocontrol en ambos manipuladores conocido como E1, tal como se observa en la figura 3 [5].

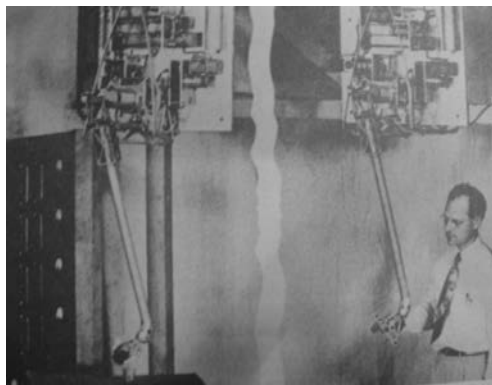


Figura 3 Primer telemanipulador maestro-esclavo eléctrico.

En los años sesenta se presentan las primeras investigaciones en el campo de las aplicaciones submarinas. A finales de los años sesenta y principios de los setenta, la tecnología de la teleoperación alcanza su mayor auge con la utilización de aplicaciones espaciales [6].

Actualmente se han tenido avances importantes en la aplicación de las tecnologías de la teleoperación al sector médico [7]. Desde los primeros desarrollos de prótesis o dispositivos de asistencia a discapacitados hasta la más

novedosa de la telecirugía, o el telediagnóstico, aunque este no pertenezca estrictamente al sector de la teleoperación.

El trabajo de Fung [9], con una teleoperación con control bilateral. La técnica remota consiste en un Robot PUMA 560 montado sobre un robot móvil Nomadic XR4000, Wireless LAN, un ordenador y una cámara de video; y la técnica local es una GUI que da realimentación visual al usuario, el usuario manipula el robot mediante un joystick con realimentación de fuerza.

El trabajo de Alvarado [10], “Diseño de Brazo Manipulador Háptico Teleoperado”, presenta el diseño de un sistema robótico que permite la retroalimentación háptica de un brazo manipulador teleoperado.

## 2. Desarrollo

El sistema de teleoperación utilizado en este trabajo consta de dos dispositivos hápticos Omni Phantom, uno se denomina maestro y otro como esclavo. El Omni maestro envía su posición angular, a través de internet bajo el protocolo TCP/IP desde la computadora cliente, y el Omni esclavo la recibe desde la computadora servidor y la reproduce. La conexión, envío y recibo de datos vía internet se llevó a cabo en Matlab y Simulink. En la figura 4 se puede observar el sistema de teleoperación.

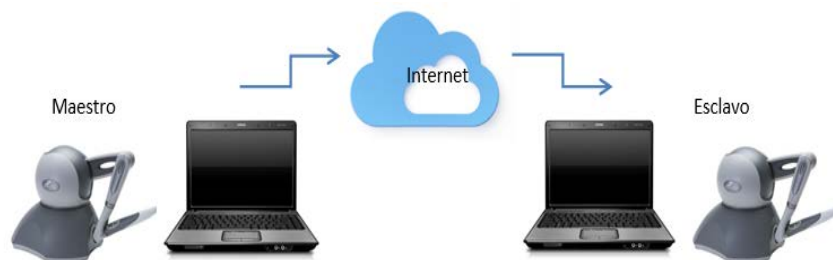


Figura 4 Sistema de teleoperación.

### Descripción del dispositivo háptico Omni Phantom

El dispositivo que se utiliza en esta investigación pertenece a los dispositivos hápticos Geomagic ®. Los dispositivos omni phantom (figura 5) mide de forma precisa la posición espacial 3D (a lo largo de los ejes x, y y z) y la orientación

(inclinación, giro y dirección) del lápiz de mano. Los dispositivos utilizan motores para crear las fuerzas de retorno en la mano del usuario para simular el tacto y la interacción con objetos virtuales. [8] El omni phantom consta con 6 grados de libertad, de los cuales solo 3 de ellos son actuados, es decir, son los que se pueden controlar mediante un algoritmo, mientras que los otros 3 son subactuados.



Figura 5 Phantom Omni.

## Software

Para este trabajo de investigación se necesita tener instalado lo siguiente:

- Sistema operativo windows7 o versiones más actuales a esta.
- Matlab 2014 o versiones más actualizadas, siempre y cuando sean para 32 bits.
- Visual Studio.
- Driver del adaptador USB del cable de red del Omni (Driver\_USB ).
- Controladores del Omni: -Geomagic\_Touch\_Device\_Driver\_2014
- OpenHaptics\_Academic\_3.1.
- Librería “Haptik library” (puede descargarse en la página web.[www.haptiklibrary.org](http://www.haptiklibrary.org))

## Comunicación

Para la comunicación en la teleoperación se utiliza el protocolo TCP/IP. Los estándares del protocolo TCP/IP son abiertos y ampliamente soportados por todo tipo de sistemas, es decir, se puede disponer libremente de ellos y son desarrollados independientemente del hardware de los ordenadores o de los sistemas operativos.

Para llevar a cabo la teleoperación se utilizan dos computadoras, una que cumpliera con el rol de cliente y otra como servidor. Para ello se crearon en el entorno de Matlab los programas que establecen la conexión TCP/IP, el envío y recibo de datos (posición del Omni Phantom).

## **Conexión**

En los programas para establecer la conexión entre cliente y servidor, se utilizaron las funciones `tcpip( )` y `fopen( )`:

- **Tcpip( )**. Esta función crea un objeto TCPIP y además se declara:
  - ✓ La IP de la computadora remota
  - ✓ El puerto remoto
  - ✓ “NetworkRole”, que activa el soporte para sockets de servidor, usando dos valores, “client” o “server” para establecer una conexión como cliente o servidor.
- **Fopen()**. Esta función conecta el objeto TCPIP al host.

## **Servidor**

Para el programa del servidor se utiliza la función `tcpip( )` configurándose de la siguiente manera:

```
conexión02=tcpip('148.213.117.20', 4013,'NetworkRole','server');
```

De esta forma, se crea un objeto TCPIP, `conexión02` (declarado como variable global), como una interfaz de servidor para poder establecer un medio para la comunicación, específicamente esta función habilita el puerto 4013 de la dirección IP ‘148.213.117.20’ y también asigna el “server” para establecer la conexión como servidor.

Para conectar el objeto “conexión02” con el cliente con IP ‘148.213.117.20’ establecido en la función `tcpip( )`, se utilizó la función `fopen( )` de la siguiente manera:

```
% Configuración y conexión  
disp('Receiver started');
```

```
conexion02=tcPIP('148.213.117.21', 4013,'NetworkRole','server');  
% Espera la conexión  
disp('Waiting for connection');  
fopen(conexion02);  
disp('Connection OK');
```

## **Cliente**

Para el programa del cliente se utiliza la función `tcPIP( )` configurándose de la siguiente manera:

```
conexión = tcPIP('148.213.117.21',4013);
```

Esta instrucción crea un objeto TCPIP, `conexión` (declarado como variable global), como una interfaz de cliente, específicamente esta función habilita el puerto 4013 de la dirección IP '148.213.117.21'. Para conectar el objeto "conexión" con el servidor con IP '148.213.117.21' establecido en la función `tcPIP( )`, se utiliza la función `fopen( )` como:

```
% Configuración y conexión  
conexion = tcPIP('148.213.117.21',4013);  
% Apertura del socket y envío de información  
fopen(conexion);  
disp('Connection OK');
```

## **Lectura y escritura de posición del dispositivo háptico**

La lectura y escritura de posición del Omni Phantom se llevó a cabo con la ayuda de la librería Haptik Library. Esta librería proporciona las posiciones del Omni o permite que estas se puedan manipular, para que el omni Phantom se dirija a la posición que se desee. En la figura 6 muestra la lectura de posición angular en simulink del programa "Envío" y la figura 7 muestra la escritura del programa "Recibo", en ambos casos se utiliza el bloque `haptik_simulink` (de la librería Haptik).

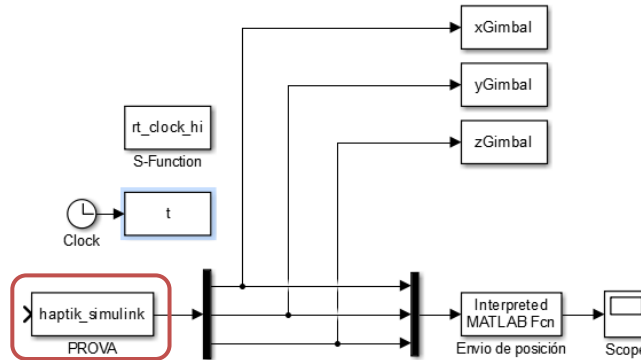


Figura 6 Lectura de posición, programa Envío.

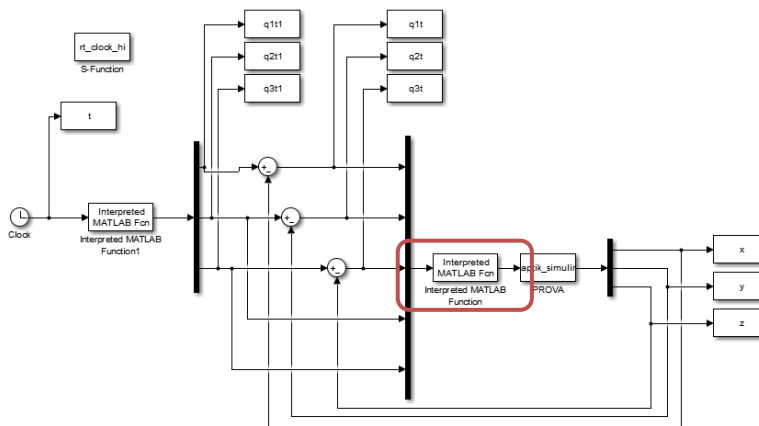


Figura 7 Escritura de posición, programa Recibo.

## Envío y recibo de datos

Para el envío y recibo de posiciones (datos), se crean dos programas en simulink. El programa para el envío se ejecuta desde la computadora cliente y el que recibe datos desde el servidor. Los dos llaman a través del bloque “Interpreted MATLAB function” a su respectiva función de envío o recibo de datos.

## Envío de datos

La figura 8 muestra el programa que realiza el envío de posición del dispositivo háptico, se observa que el bloque “Interpreted MATLAB function”, recibe los datos de posición desde el bloque “haptik\_simulink”, los cuales se envían con la función “envios02”.



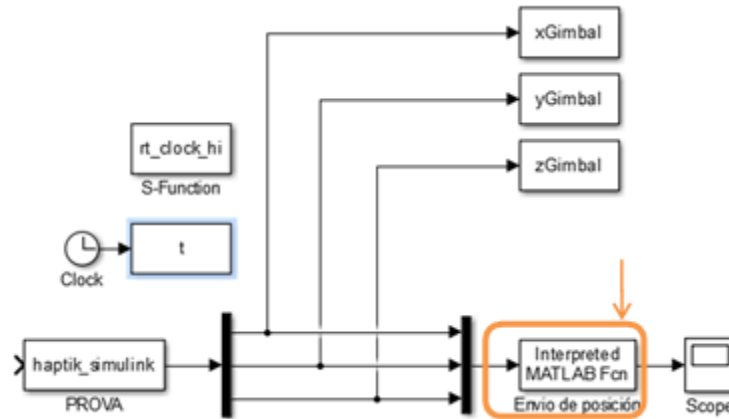


Figura 8 Envío de datos.

### **Función envios02**

Se programa una función llamada envios02 que es ejecutada por el bloque “Interpreted MATLAB function”. Ésta realiza el envío de posiciones de la siguiente manera:

- Recibe la posición del Omni proporcionada por el bloque haptik\_simulink
- Verifica si es la primera vez que se envíen los datos
- Si es así, envía la posición al servidor.
- Si no, evalúa si hubo un cambio de posición.
  - ✓ Si hubo cambios, envía la nueva posición.
  - ✓ Si no, mantiene la misma posición.

Para el envío de posición se utiliza la función de Matlab fwrite( ), la cual escribe los datos en el objeto tcpip, de la siguiente manera.

```
fwrite(conexión,tt,'double');
```

Siendo **conexión**: el objeto tcpip al que se escriben los datos,

**tt**: el vector de posición ,

**double**: el tipo de dato con el que se envía el valor tt.

### **Recibo de datos**

La figura 9 muestra el programa que realiza el recibo de datos de posición. El bloque “Recibo de posición”, con la función y “recibidos02”, recibe datos de

posición, y estos se escriben al dispositivo háptico a través del controlador y el bloque `haptik_simulink`.

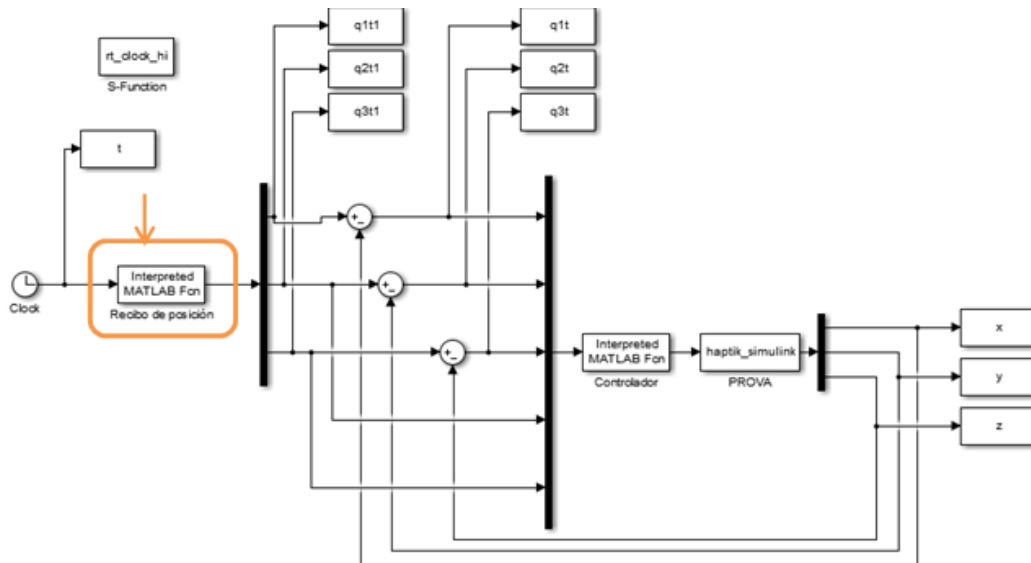


Figura 9 Recibo de datos.

**Función recibidos02.** La función recibidos02 es ejecutada por el bloque “Recibo de posición”. Esta recibe las posiciones con la función `fread( )`, de la siguiente manera:

$$Info = fread(conexión02, [3 1], 'double')$$

Siendo **conexión02**: el objeto `tcpip` desde donde se leen los datos:

- **[3 1]**: tamaño del vector de datos que recibe,
- **double**: el tipo de datos del vector que recibe.

Asignándose el valor de la función `fread( )` a la variable `info`.

### Control del sistema de teleoperación

De las ecuaciones de movimiento del omni phantom, el modelo dinámico del robot contiene no linealidades que se deben principalmente a las fuerzas de fricción, coriolis, gravedad y fuerzas centrífugas, por lo que no es posible aplicar directamente las técnicas de control clásico, por lo tanto se aplica un control PD

con compensador de gravedad, que contrarresten las no linealidades que aparecen en el modelo dinámico, en este caso el vector de gravedad, se aplica este controlador ya que es uno de los más sencillos y además garantiza el control de los movimientos del robot para que siga una trayectoria determinada por el omni maestro.

Para mover el omni Phantom esclavo a la posición deseada  $q_{1t1}$ ,  $q_{2t1}$  y  $q_{3t1}$ , enviados por el omni maestro, se utiliza un controlador PID con compensador de gravedad. En la figura 10 muestra el programa "Recibo" con el bloque Controlador. En el esquema se observa que la posición recibida por el omni maestro ( $q_{1t1}$ ,  $q_{2t1}$ ,  $q_{3t1}$ ) es el set point del controlador; se hace una diferencia la posición actual ( $q_1$ ,  $q_2$ ,  $q_3$ ), que es leída por el bloque `haptik_simulink`, para obtener el error de posición, este entra al controlador el cual corrige la posición hasta disminuir el error a cero.

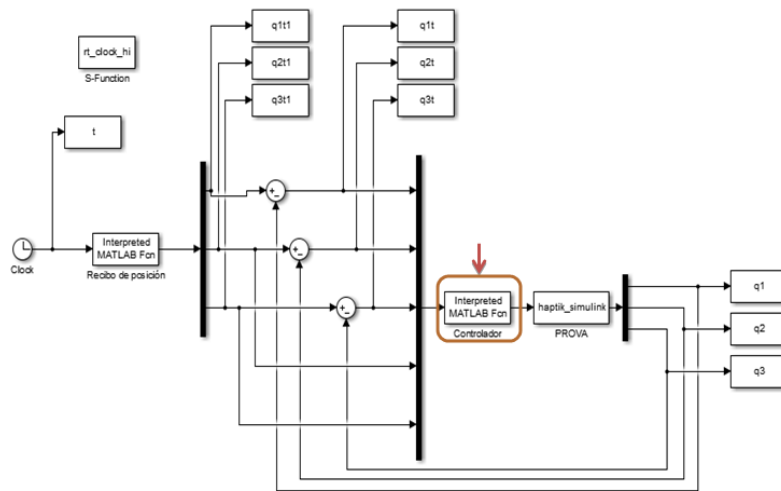


Figura 10 Control PD+G.

### 3. Resultados experimentales

#### Descripción del equipo experimental

El equipo experimental consiste en un manipulador Omni Phantom (dispositivo háptico) en arquitectura abierta, es decir puede implementarse cualquier algoritmo de control a este dispositivo háptico. También dos computadoras con Windows 7

instalado; con software especializado como visual studio 2008 C++, Matlab R2014a, tal como se muestra en la figura 11.



Figura 11 Equipo experimental.

### Resultados experimentales

Los resultados experimentales de la teleoperación, realizando trayectorias de forma manual y aleatoria con el Omni maestro, cada uno con diferente duración (en un tiempo de 15 a 35 segundos), las cuales fueron enviadas a través de internet siendo recibidas y repetidas por el Omni esclavo. De esta forma se obtuvieron las gráficas de error de posición y de la posición deseada (enviada por el Omni maestro) contra la posición real realizada por el esclavo.

La figura 12 muestra las gráficas del primer, segundo y tercer eslabón respectivamente (de arriba hacia abajo) del primer experimento.

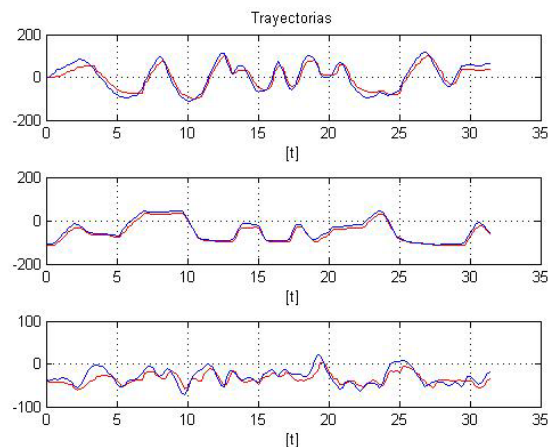


Figura 12 Trayectoria deseada vs real del primer experimento.

Se muestra en color azul el set point, que es la trayectoria enviada por el Omni maestro y en color rojo la trayectoria realizada por el Omni esclavo (para cada eslabón). Como se puede observar, se obtuvieron buenos resultados ya que hay una diferencia muy pequeña entre las trayectorias deseadas y las reales. Esto se puede ver en la gráfica siguiente, donde se muestra el error de ésta trayectoria.

El segundo experimento es similar al primero, pero los movimientos son más lentos tal como se muestra en la figura 13, muestra las gráficas del primer, segundo y tercer eslabón respectivamente (de arriba hacia abajo) del primer experimento.

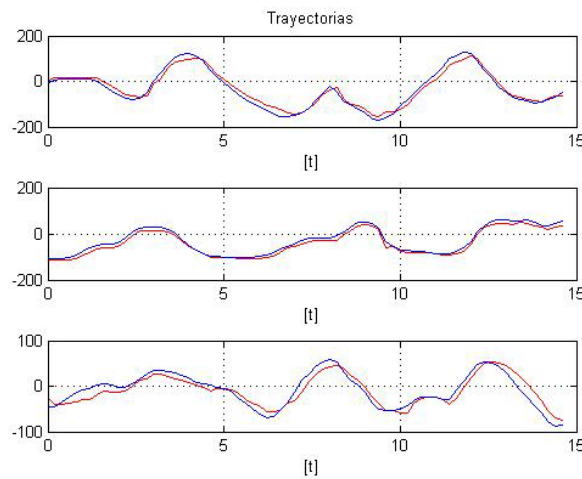


Figura 13 Trayectoria deseada vs real del segundo experimento.

#### 4. Conclusiones

En el presente trabajo de investigación se logró construir un sistema de teleoperación con dos dispositivos hápticos Omni Phantom utilizando las herramientas de Matlab, Simulink y la librería Haptik.

La comunicación se consiguió bajo el protocolo TCP/IP con ayuda de los objetos TCPIP de Matlab y con las funciones `fwrite()` y `fread()`, que permiten la escritura y lectura de datos (posiciones o trayectorias) vía internet de manera eficiente.

Para seguir la trayectoria y minimizar los problemas que causan los retardos en la comunicación se implementa un controlador PD con compensador de gravedad teniendo un resultado satisfactorio en el seguimiento de trayectorias por parte del Omni esclavo

Este trabajo de investigación queda como base para futuros trabajos de teleoperación en los que se aplique controladores no lineales más complejos que incluyan parte del modelo matemático de los dispositivos hápticos, complementándolo por ejemplo con entornos virtuales o llevándolo a una aplicación en específico como lo es el diagnóstico de deficiencias motrices, entrenamiento de cirugías a distancia, etc.

## **5. Bibliografía y Referencias**

- [1] C. Jianhong, S. Tosunoglu, C. Moore and D. Repperger, "A review of teleoperation system control". Florida Conference on Recent Advances in Robotics (FCRAR). May 2003.
- [2] D. Alvarado Juárez, M. A. Butrón Castañeda, D. Díaz Loyo y L. C. González García, "Teleoperación Háptica de Brazo Manipulador". Congreso científico tecnológico de IME. 2013. Pp. 1-6.
- [3] E. Nuño Ortega and L. Basañez Villaluenga, Teleoperación: técnicas, aplicaciones, entorno sensorial y teleoperación inteligente. 2004. Universitat Politècnica de Catalunya. Barcelona, España. Pp 1-15.
- [4] A. Ayala, L. García. G. Reyna, Desarrollo de un Sistema Robótico Háptico para Aplicaciones Biomédicas. 2008. Instituto Politécnico Nacional. CDMX. Pp 1-85.
- [5] J. Vertut, P. Coiffet, Les Robots; Téléopération, évolution des technologies. 1985th Edition. 1985. París: Editions Hermes. Francia.
- [6] T. B. Sheridan, "Space Teleoperation through Time Delay. Review and Prognosis". IEEE Transactions on Robotics and Automation. Vol. 9. Issue 5. Oct 1993. Pages: 592-606.
- [7] P. Dario, A. Menciassi, "Robotics for Surgery". 24<sup>th</sup> Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference. 2002.
- [8] M. Ghodoussi, S. Butner, W. Yulun, "Robotic Surgery - the Transatlantic Case". ICRA '02. IEEE International Conference on Robotics and Automation. Vol. 2 August 2002.

- [9] W. K. Fung, N. Xi, W. T. Lo, Y. H. Liu, "Adaptive real-time Internet-based teleoperation systems for efficiency improvement using network QoS". IEEE/ASME International Conference on Advanced Intelligent Mechatronics. July 2003.
- [10] D. A. Alvarado Juárez, M. A. Butrón Castañeda, D. Diaz Loyo, L. C. González García, "Diseño de Brazo Manipulador Háptico Teleoperado. La Mecatrónica en México". Vol. 2. Núm. 2. 2013. Pp. 67-76.

## **6. Autores**

Jorge Gudiño Lau es Doctor en Ingeniería Eléctrica en el área de control con especialidad en Robótica por la Universidad Nacional Autónoma de México en el 2005. Actualmente es Profesor-Investigador de Tiempo Completo y Líder del Cuerpo Académico UCOL-CA 21 Sistemas Mecatrónicos y Eléctricos de la Facultad de Ingeniería Electromecánica de la Universidad de Colima.

Felipe de Jesús Ríos Cortes es Ingeniero en Comunicaciones y Electrónica por la Universidad de Colima. Actualmente es catedrático de la Facultad de Ingeniería Electromecánica de la Universidad de Colima. Cursa los estudios de posgrado en la Maestría en Ingeniería en la misma facultad.

Dr. Miguel Ángel Duran Fonseca Doctor en Ingeniería Eléctrica en el Centro Nacional de Investigación y Desarrollo Tecnológico CENIDET en el 2010. Actualmente es Profesor-Investigador de Tiempo Completo, Coordinador Académico e integrante del Cuerpo Académico UCOL-CA 21 Sistemas Mecatrónicos y Eléctricos de la Facultad de Ingeniería Electromecánica de la Universidad de Colima.

M.I Saida Miriam Charre Ibarra es Maestra en Ingeniería en el área de automatización y control por el Instituto Tecnológico de Ciudad Guzmán en el 2004. Actualmente es Profesor-Investigador de Tiempo Completo y pertenece al Cuerpo Académico UCOL-CA 21 Sistemas Mecatrónicos y Eléctricos de la Facultad de Ingeniería Electromecánica de la Universidad de Colima.