

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA DE TECNOLOGÍAS Y
SERVICIOS DE TELECOMUNICACIÓN**

TRABAJO FIN DE GRADO

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
REALIMENTACIÓN HÁPTICA PARA FORMACIÓN
EN CIRUGÍA LAPAROSCÓPICA**

DELIA SEPÚLVEDA MUÑOZ

2017

UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

Reunido el tribunal examinador en el día de la fecha, constituido por

Presidente: Dra D. María Teresa Arredondo Waldmeyer

Vocal: Dra D. Ana Pilar González Marcos

Secretario: D. María José Melcón de Giles

Suplente: Dr D. José Ramón Tapia Merino

para juzgar el Trabajo Fin de Grado titulado:

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
REALIMENTACIÓN HÁPTICA PARA FORMACIÓN EN
CIRUGÍA LAPAROSCÓPICA

del alumno D. Delia Sepúlveda Muñoz
dirigido por D. Álvaro Gutiérrez Martín

Acuerdan otorgar la calificación de: _____

Y, para que conste, se extiende firmada por los componentes del tribunal, la presente diligencia

Madrid, a 26 de Enero de 2017

El Presidente

El Vocal

El Secretario

Fdo: _____ Fdo: _____ Fdo: _____

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA DE TECNOLOGÍAS Y
SERVICIOS DE TELECOMUNICACIÓN**

TRABAJO FIN DE GRADO

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
REALIMENTACIÓN HÁPTICA PARA FORMACIÓN
EN CIRUGÍA LAPAROSCÓPICA**

DELIA SEPÚLVEDA MUÑOZ

2017

Resumen

El presente Trabajo de Fin de Grado consiste en el diseño y desarrollo de un sistema de realimentación háptica y simulación virtual a implementar en un entrenador de cirugía laparoscópica. Para ello se utilizará un software de representación de objetos en 3D y se hará uso de los 8 motores disponibles en un entrenador laparoscópico comercial.

Los motores se activarán gracias a las instrucciones procedentes del programa de simulación, que proporciona un vector de fuerza debido a la interacción de los dos instrumentos laparoscópicos con los objetos de simulación virtual. Este vector de fuerza deberá ser procesado para enviar el voltaje correspondiente a los motores encargados de la realimentación de los brazos laparoscópicos.

Para la comunicación es necesario codificar estos datos en el PC para ser transmitidos a la tarjeta de desarrollo (Arduino Due). El Arduino Due se encarga de procesar los distintos ciclos de trabajo de las señales que alimentan los motores y genera las señales de Modulación por Anchura de Pulsos (PWM, de sus siglas en inglés) necesarias en cada caso.

Gracias a la realización de lo descrito anteriormente, el usuario podrá recibir los diferentes estímulos hápticos mediante una realimentación de fuerza en lazo cerrado. Se tendrán especialmente en cuenta las limitaciones de tiempo real para que el usuario tenga una experiencia háptica lo más realista posible.

Abstract

The aim of this project is to design and develop a haptic feedback and virtual simulation system, which will be included in a laparoscopic surgery trainer. For this purpose, a 3D object representation software and 8 motors from a commercial laparoscopic trainer will be used.

The motors will be activated responding to the instructions from the simulation program, which provides a force vector due to the interaction of the two laparoscopic instruments with the virtual object. This force vector will be processed to send the voltage to the corresponding motors, which provide the feedback to the laparoscopic arms.

In the communication, it is necessary to encode these data in the PC before transmitting them to the development board (Arduino Due). Arduino Due processes

the different duty cycle signals which power the motors and generates the PWM signals necessary in each case.

Therefore, the user can receive the different haptic stimuli by a closed loop force feedback. The real time limitation will be taken into account for achieving a realistic haptic user experience.

PALABRAS CLAVE: háptico, motor DC, laparoscopia, realimentación de fuerza, entorno simulado.

KEYWORDS: haptic, DC motor, laparoscopy, force feedback, virtual environment.

Agradecimientos

Querría darle las gracias a todo el personal de Robolabo, en especial a mi tutor Álvaro Gutierrez por el apoyo y la ayuda recibidas, que me han guiado en el presente Trabajo de Fin de Grado.

A mis amigos, por dedicarme su tiempo cuando más se necesita.

A mi familia, especialmente a mis padres, por estar siempre presentes, apoyarme en todo momento y por haberme enseñado que con trabajo y constancia todo se consigue.

Índice general

Resumen	IV
Agradecimientos	VI
Índice General	VII
Índice de Figuras	IX
Índice de Tablas	X
Lista de acrónimos	XII
1. Introducción y Objetivos	1
1.1. Introducción a la laparoscopia	1
1.1.1. Tipos de entrenadores laparoscópicos comerciales	1
1.2. Introducción a los sistemas de realimentación háptica	4
1.3. Motivación	6
1.4. Objetivos	6
2. Análisis cinemático del entrenador laparoscópico	7
2.1. Estudio de los grados de libertad y rango de movimiento	7
2.2. Motores DC	10
2.2.1. Modelado teórico del motor DC	10
2.2.2. Diseño de un controlador de posición	12
2.3. Estudio cinemático de la caja entrenadora	15
2.3.1. Cinemática directa	15
2.3.2. Cinemática inversa	17
2.3.3. Pares de fuerza	18
3. Realimentación háptica	21
3.1. Sistema de bloques	21
3.2. Hardware	22
3.2.1. Arduino Due	22
3.2.2. Tarjeta de expansión de controlador de motores DC	22
3.2.3. Adaptación de las señales de los motores	24
3.3. Sistema en tiempo real	25
3.3.1. Requisitos de tiempo real	25
3.3.2. Puertos USB disponibles en Arduino Due	25
3.3.3. Medida de tiempos	26

3.4. Software	27
3.4.1. CHAI 3D	27
3.4.2. Comunicación	28
3.4.3. Implementación en Arduino	29
3.4.4. Experimentos	32
4. Conclusiones y líneas futuras	35
4.1. Conclusiones	35
4.2. Líneas futuras	35
Apéndice	38
A. Especificaciones del Motor	39
Bibliografía	41

Índice de figuras

1.1. Simulador T5 LARGE	2
1.2. Simulador LapTrainer	3
1.3. Simulador eoSim	3
1.4. Simulador Laparoscopic Surgical Workstation	4
1.5. PHANTOM Omni® Haptic Device	5
1.6. HapticMaster	6
2.1. Entrenador para la formación de cirujanos laparoscópicos (Gregorio et al., 2011).	7
2.2. Grados de libertad del entrenador laparoscópico.	8
2.3. Movimiento sobre el eje Z.	8
2.4. Movimiento sobre el eje Y.	9
2.5. Movimiento sobre el eje X.	9
2.6. Esquema de bloques de un controlador P para un sistema continuo de segundo orden y tipo uno.	12
2.7. Distintos tipos de salida para un controlador P según ζ	13
2.8. Respuesta de un controlador proporcional de posición a la entrada escalón	14
2.9. Coordenadas generalizadas	16
3.1. Sistema de bloques del sistema	21
3.2. X-NUCLEO-IHM04A1	23
3.3. Zona Z^+	23
3.4. Zona Z^-	24
3.5. Placa de conexionado	24
3.6. Visualización de las pinzas laparoscópicas y la membrana en CHAI 3D	28
3.7. Señales con distintos tipos de ciclo de trabajo	30
3.8. Señal PWM Unipolar	31
3.9. Diferentes radios de los nodos que conforman la membrana virtual . .	32
3.10. Diferentes tamaños de la matriz de nodos	33
3.11. Diferentes tamaños de la matriz de nodos	34

Índice de tablas

2.1. Características del motor DC	10
3.1. Tiempo de envío de un carácter	26
3.2. Tiempo de envío de ocho caracteres	26
3.3. Tiempo de ejecución de las diferentes tareas	27
3.4. Identificadores de cada motor	29
3.5. Movimiento de los motores según el signo del ciclo de trabajo	29
3.6. Pines de Arduino Due correspondientes al dispositivo izquierdo	30
3.7. Pines de Arduino Due correspondientes al dispositivo derecho	31

Lista de Acrónimos

PWM: Pulse Width Modulation.

CMI: Cirugía Mínimamente Invasiva.

DC: Direct current.

CVS: Coulomb plus Viscous plus Static friction.

USB: Universal Serial Bus.

UART: Universal Asynchronous Receiver-Transmitter.

DMOS: Double-diffused Metal Oxide Semiconductor.

Capítulo 1

Introducción y Objetivos

1.1. Introducción a la laparoscopia

La cirugía laparoscópica es un tipo de Cirugía Mínimamente Invasiva (CMI) en la que el instrumental quirúrgico y el endoscopio (cámara) se introducen en la cavidad abdominal del paciente a través de pequeñas incisiones. El cirujano manipula el instrumental siguiendo la intervención a través de un monitor donde se visualizan las imágenes captadas por el endoscopio. El cirujano laparoscópico no interactúa directamente con los órganos ya que no tiene una visión directa de los mismos, por lo que se produce una disminución de la percepción sensorial. La información visual pasa a ser monoscópica, es decir, se pierde la sensación de profundidad aumentando los problemas de coordinación ojo-mano. Esto hace que el aprendizaje y el propio desarrollo de la técnica se conviertan en una difícil tarea. (A Park, 2002)

Este tipo de cirugía ha revolucionado la práctica quirúrgica habitual en los últimos años, debido a la reducción del trauma tisular, del tiempo de recuperación y de las complicaciones y dolor postoperatorios entre otras ventajas. Sin embargo, esta técnica quirúrgica requiere la supervisión de un experto personal altamente cualificado y una formación y aprendizaje extensos, que suele incluir una etapa de formación experimental en animales. (J Usón, 1999)

Por todo esto, la cirugía laparoscópica representa una nueva era de intervenciones quirúrgicas dependientes de la tecnología y, en cierta medida, su futuro progreso depende del crecimiento de tecnologías y dispositivos de intervención. La cirugía laparoscópica ha tenido un impacto significativo en todas las disciplinas quirúrgicas y ahora está firmemente arraigada en la práctica quirúrgica rutinaria. (Cuschieri, 2005)

1.1.1. Tipos de entrenadores laparoscópicos comerciales

A diferencia de la cirugía abierta, para la práctica quirúrgica laparoscópica es necesario la adquisición de nuevas habilidades debido a la atenuación del sentido del tacto, el alejamiento del campo operatorio, la disociación mano-ojo y a la falta de la tercera dimensión, por lo que es necesario una gran preparación y aprendizaje previos.

Tradicionalmente, el entrenamiento quirúrgico estándar ha sido de aprendizaje, en el que el cirujano practicante aprende a realizar la cirugía bajo la supervisión de un cirujano entrenado. Esta práctica es costosa, lenta y de efectividad variable. La cirugía laparoscópica incluye el uso de instrumentos empleando un acceso mínimamente

invasivo, y en general se considera más difícil de realizar que la cirugía abierta. Actualmente el entrenamiento mediante un simulador de realidad virtual (simulación computarizada) es una opción para complementar el entrenamiento en cirugía laparoscópica estándar. El entrenamiento en realidad virtual mejora las aptitudes técnicas de los cirujanos practicantes. (M Nagendran, K Gurusamy et al., 2013)

Los simuladores son sistemas que permiten dominar el instrumental y ejercitar gestos quirúrgicos (suturas, nudos, etc.) mediante cajas de entrenamiento o programas de realidad virtual. (Gomis, 2003)

Actualmente existen diferentes tipos de simuladores de cirugía laparoscópica comerciales que se pueden clasificar según el público al que están destinados ya que éstos pueden ser para enseñanza y servir de aprendizaje a futuros cirujanos o también pueden estar destinados a cirujanos ya formados que quieran seguir mejorando su técnica. La mayoría de estos dispositivos cuentan también con simulación física o virtual de diferentes tejidos o diferentes funcionalidades para mejorar su técnica con el laparoscopio.

A continuación se exponen algunos de estos entrenadores laparoscópicos ya desarrollados.

- T5 LARGE

Desarrollado por la empresa 3-Dmed, es un sistema versátil que puede ser usado tanto por estudiantes como por cirujanos con experiencia. Cuenta con un monitor que muestra las imágenes obtenidas del endoscopio de navegación. Cuenta con diferentes tipos de tejidos físicos artificiales para la práctica. Contiene 14 puertos laparoscópicos con arandelas intercambiables de diferentes tamaños (5mm y 10mm) e iluminación interior. (3-Dmed) (ver Figura 1.1)



Figura 1.1: Simulador T5 LARGE

- LapTrainer

Desarrollado por SIMULAB Corporation, es un entrenador de cirugía laparoscópica destinado a la formación de estudiantes. Desarrolla la coordinación ojo-mano, así como el manejo de herramientas y la precisión con diversas

técnicas quirúrgicas. Cuenta con dos puertos laparoscópicos y un endoscopio del cuál se transmiten las imágenes a un PC o portátil, cuyo cabezal es ajustable y se pueden conseguir ángulos de 0°, 30° y 45°. Los tejidos utilizados al igual que en el anterior son físicos y artificiales.(Simulab) (ver Figura 1.2)



Figura 1.2: Simulador LapTrainer

- eoSim SurgTrac

Este dispositivo se presenta en varias versiones dependiendo del público al que está dirigido. Nos centraremos en su versión Start ya que es la destinada al aprendizaje de los estudiantes que es el tema que nos concierne. Desarrollado por la compañía eoSurgical, permite adquirir a los estudiantes destrezas psicomotoras básicas mediante la práctica en el hogar, ya que solo se necesita una tablet o smartphone para obtener las imágenes de las pinzas laparoscópicas que provienen de la propia cámara del dispositivo. (eoSurgical) (ver Figura 1.3)



Figura 1.3: Simulador eoSim

- Laparoscopic Surgical Workstation

Desarrollado por la empresa Immersion Corporation, simula la interacción háptica del cirujano. Este dispositivo provee una realimentación de fuerza al

usuario, el cual puede sentir la consistencia de los órganos virtuales. Cuenta con dos brazos laparoscópicos que poseen 4 grados de libertad cada uno. Y será el entrenador que se utilizará en este Trabajo de Fin de Grado. (Tzovaras, 2008) (ver Figura 1.4)



Figura 1.4: Simulador Laparoscopic Surgical Workstation

1.2. Introducción a los sistemas de realimentación háptica

Los interfaces hápticos introducen los robots en entornos virtuales para aumentar la sensación de presencia, reproduciendo fuerzas de contacto, mejoras de interacción, sentido de presencia y destreza en el manejo de objetos virtuales

Los dispositivos hápticos se pueden clasificar en:

- Dispositivos hápticos pasivos. No se produce realimentación, ya que son sistemas de lazo abierto. Se programa la disipación como una función del espacio o del tiempo.
- Dispositivos hápticos activos. En este caso sí que hay realimentación, por lo tanto son sistemas de lazo cerrado. Estos a su vez se clasifican en:
 - Isotónicos: Los actuadores funcionan como una fuente de fuerza (variación de esfuerzo) y se mide la posición. El usuario mueve el dispositivo, que reacciona con una fuerza. Es como especificar una impedancia para producir una simulación y corresponde al control de impedancia:
 - Se controla la fuerza de interacción del dispositivo con el operador.
 - El dispositivo actúa como una impedancia y el operador como una admitancia.
 - No es imprescindible que el dispositivo tenga sensores de fuerza. Se puede realizar con modelos virtuales.
 - Un robot de impedancia es uno de baja inercia y fricción (la mayoría de los dispositivos hápticos). Se les considera “fuentes de fuerza”.

Control de impedancia: $v \rightarrow F$ (Monasterio-Huelin)

Ejemplo: Phantom, dispositivo háptico desarrollado por MIT Artificial Intelligence Laboratory. Posee 6 grados de libertad. Puede medir de forma precisa la posición espacial 3D (a lo largo de los ejes X, Y y Z) y la orientación (giro, inclinación y dirección) del lápiz de mano. Utiliza motores para crear las fuerzas de retorno en la mano del usuario para simular el tacto y la interacción con objetos virtuales. (Geomagic) (ver Figura 1.5)



Figura 1.5: PHANTOM Omni® Haptic Device

- Isométricos: Los actuadores funcionan como una fuente de posición y se mide la fuerza. El dispositivo mide las fuerzas que el usuario efectúa sobre él, y reacciona con un movimiento (posición, velocidad, aceleración). Corresponde al control de admitancia:
 - Se controla el movimiento del dispositivo.
 - El dispositivo actúa como una admitancia y el operador como una impedancia.
 - Un robot de admitancia es uno de elevadas inercia y fricción (robot industrial). Se les considera “fuentes de velocidad”

Control de admitancia: $F \rightarrow v$ (Monasterio-Huelin)

Ejemplo: HapticMaster es un dispositivo háptico comercial de 3 grados de libertad desarrollado por Moog FCS Robotics. Es un dispositivo de admitancia (controlado por la fuerza) y por lo tanto es capaz de proporcionar alta rigidez y fuerzas con una fricción mínima. (DelfHapticLab) (ver Figura 1.6)

El sistema ha desarrollado en este Trabajo de Fin de Grado se trata de un sistema de realimentación háptica de fuerza que permite recibir una sensación de impacto y vibración cuando se aplican fuerzas virtuales exteriores, que además nos permite discernir geometría y textura. Produce la sensación de estar tocando realmente un mundo virtual o remoto. El objetivo final es que el operador no distinga entre lo real y lo virtual, aunque depende de las limitaciones del hardware.



Figura 1.6: HapticMaster

1.3. Motivación

Pocos de los dispositivos comerciales de entrenadores laparoscópicos, cuentan con simulación virtual de tejidos, aún siendo éstos una herramienta fundamental para la formación y evaluación de cirujanos. Sin duda, en un futuro, la realidad virtual será el medio utilizado para formar y evaluar la capacitación del personal médico, resolviendo muchas cuestiones hoy planteadas relacionadas con la seguridad de los pacientes y con problemas éticos y económicos (Gomis, 2003). La necesidad de un entrenamiento más completo para la formación de futuros cirujanos laparoscópicos, es deseable y con sistema de realidad virtual se pueden obtener unos mejores resultados de aprendizaje en un menor tiempo. Éstas han sido las motivaciones para la realización del presente Trabajo de Fin de Grado.

1.4. Objetivos

Los objetivos de este Trabajo de Fin de Grado son:

- Estudiar las distintas velocidades que se pueden conseguir en la transmisión de datos entre Arduino Due y el PC.
- Estudiar el hardware del que se dispone en el entrenador laparoscópico y comprender el uso de cada uno de sus elementos.
- Desarrollar e implementar el hardware necesario para el correcto funcionamiento de los motores.
- Desarrollar el software de control de los motores del entrenador laparoscópico.
- Implementar el módulo software que permite la realimentación háptica de fuerza de lazo cerrado.

Capítulo 2

Análisis cinemático del entrenador laparoscópico

2.1. Estudio de los grados de libertad y rango de movimiento

El entrenador de cirugía laparoscópica dispone de dos pinzas laparoscópicas unidas a la caja entrenadora mediante un anclaje (ver Figura 2.1). Debido a esta unión, cada pinza dispone de 4 grados de libertad, 3 movimientos de rotación (sobre los ejes X, Y, Z) y uno de traslación (eje Z) (ver Figura 2.2). Para el desarrollo de este Trabajo de Fin de Grado se tendrán en cuenta solo 3 grados de libertad, despreciando el movimiento de rotación sobre el eje Z en lo que respecta a la realimentación háptica, ya que se trata de un movimiento de orientación y por tanto la posición de las pinzas laparoscópicas no varía.

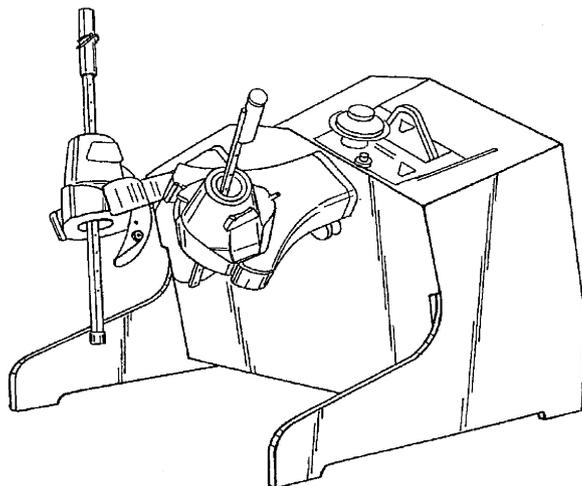


Figura 2.1: Entrenador para la formación de cirujanos laparoscópicos (Gregorio et al., 2011).

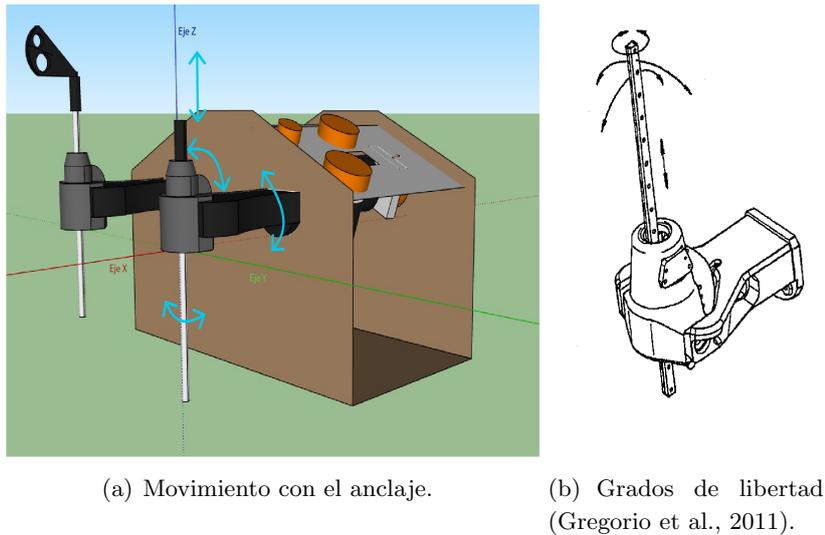


Figura 2.2: Grados de libertad del entrenador laparoscópico.

Descripción de los movimientos que se pueden realizar con el laparoscopio:

- Movimiento sobre el eje Z. Se trata de un movimiento de traslación sobre el eje Z. El rango de movimiento es de 35cm, siendo el desplazamiento que se puede realizar desde el anclaje hasta el final de la parte superior de las pinzas laparoscópicas de 17.5cm y del anclaje hacia el extremo inferior de 17.5cm. (ver Figura 2.3)

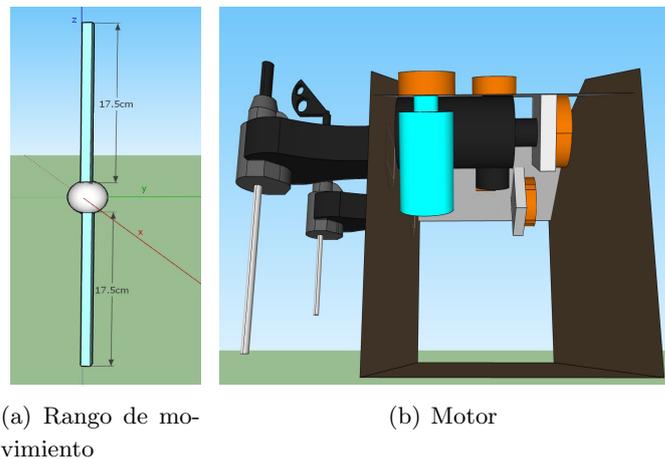


Figura 2.3: Movimiento sobre el eje Z.

- Movimiento sobre el eje Y. Es un movimiento de rotación sobre el eje Y. Gracias a este movimiento se pueden mover las pinzas laparoscópicas a lo largo de los ejes X y Z siendo la posición en el eje Y constante. Partiendo de la posición inicial en la que las pinzas laparoscópicas están perpendiculares a la superficie y

paralelas a la caja entrenadora, se pueden realizar movimientos de 46° cuando se mueven hacia la caja entrenadora y de 55.5° cuando las alejamos de ella. (ver Figura 2.4)

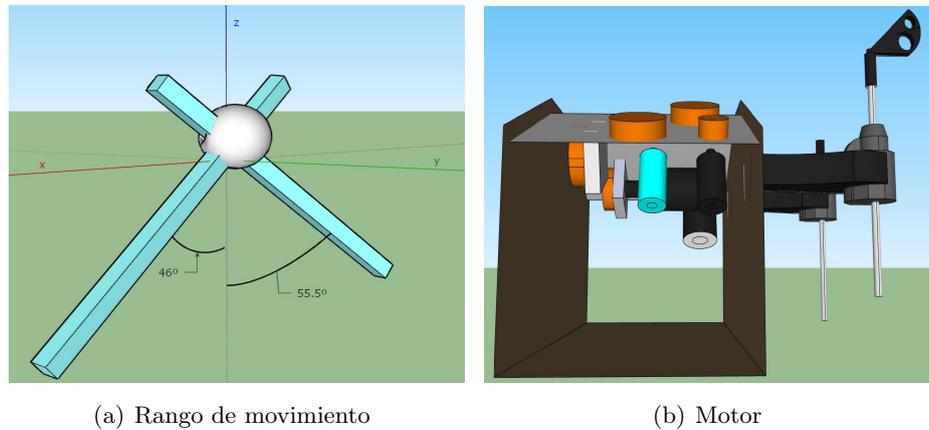


Figura 2.4: Movimiento sobre el eje Y.

- Movimiento sobre el eje X. Es un movimiento de rotación sobre el eje X, este movimiento permite desplazar las pinzas laparoscópicas en los ejes Y y Z mientras que la posición en el eje X se mantiene constante. Partiendo de la misma posición inicial que en el caso anterior, el rango de movimiento permitido es de 26° , cuando el movimiento se realiza hacia el centro de la caja entrenadora y de 67° cuando es hacia los extremos. (ver Figura 2.5)

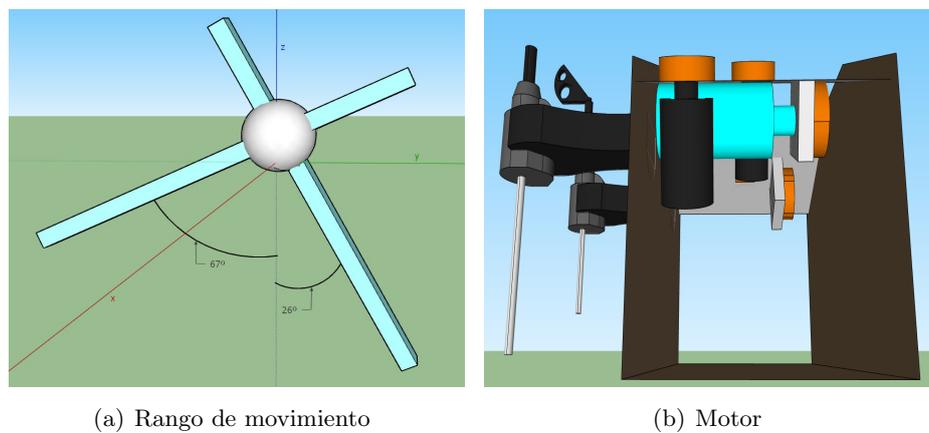


Figura 2.5: Movimiento sobre el eje X.

2.2. Motores DC

Cada grado de libertad de la pinza laparoscópica, tiene asociado un motor de Corriente Continua (DC). Un motor eléctrico de Corriente Continua es esencialmente una máquina que convierte energía eléctrica en movimiento o trabajo mecánico, a través de medios electromagnéticos. Los motores disponibles en el entrenador de cirugía laparoscópica son motores DC de la compañía Maxon, modelo 118783, cuyas características se detallan en la Tabla 2.1. Los datos se han obtenido a partir de la hoja de características distribuída por el fabricante (especificaciones presentadas en el Apéndice A).

Parámetro	Valor	Unidades
U_N , Tensión nominal	48	V
R_m , Resistencia terminal	11,5	Ω
L_m , Inductancia del rotor	$3,16 \times 10^{-3}$	H
J_m , Inercia del rotor	$6,28 \times 10^{-6}$	kgm^2
t_m , Constante de tiempo mecánica	5×10^{-3}	s
t_e , Constante de tiempo eléctrica	$2,747 \times 10^{-4}$	s
k_b , Constante de la fuerza contraelectromotriz	$1,186 \times 10^{-1}$	Vs/rad
k_m , Constante de par	$1,186 \times 10^{-1}$	Nm/A
I_o , Corriente del motor sin carga	$3,4 \times 10^{-2}$	A
n_o , Velocidad del motor sin carga	398,982	rad/s

Tabla 2.1: Características del motor DC

2.2.1. Modelado teórico del motor DC

La finalidad de este desarrollo es obtener la función de transferencia de cada uno de los motores, que tendrá la forma:

$$G_{\dot{\theta}_m}(s) = \frac{K_m}{s + p_m} \quad (2.1)$$

siendo la función de transferencia de velocidad angular del motor, expuesta de forma simplificada. A continuación se detalla cada uno de sus términos y la manera en la que se ha obtenido cada uno de ellos.

El motor puede ser definido mediante la Ecuación eléctrica 2.2:

$$u_m(t) = R_m i(t) + L_m \frac{di(t)}{dt} + e_b(t) \quad (2.2)$$

donde $u_m(t)$ representa la tensión de entrada al motor, $i(t)$ la corriente eléctrica del motor, $e_b(t)$ la fuerza contraelectromotriz, R_m la resistencia terminal y L_m la inductancia del rotor. Y también mediante la Ecuación mecánica 2.3

$$\tau_m(t) = J_m \ddot{\theta}_m(t) + \tau_l(t) + \tau_f(t) : \quad (2.3)$$

donde $\tau_m(t)$ representa el par del motor, $\ddot{\theta}_m(t)$ la aceleración angular del motor, J_m la inercia del rotor, $\tau_l(t)$ el par de la carga visto desde el eje del motor y $\tau_f(t)$ el par de fricción (Monasterio-Huelin, 2016c).

Siguiendo el modelo clásico de fricción o modelo CVS que consta de tres componentes, el par de fricción de Coulomb τ_{fC} , el par de fricción viscosa τ_{fV} y el par de fricción estática τ_{fS} , teniendo en cuenta en este caso que el par de fricción viscosa es el más importante y se define como $\tau_{fV}(t) = B_m \dot{\theta}_m(t)$ la ecuación mecánica del motor puede escribirse como:

$$\tau_m(t) = J_m \ddot{\theta}_m(t) + B_m \dot{\theta}_m(t) + \tau_c(t) \quad (2.4)$$

donde B_m es la constante de fricción viscosa, $\dot{\theta}_m(t)$ es la velocidad angular del motor y $\tau_c(t)$ el par que incorpora la carga y la fricción, a excepción del par de fricción viscosa. Para estimar el coeficiente de fricción viscosa, B_m , se va a utilizar el método de la constante de tiempo mecánica, $t_m = \frac{R_m J_m}{R_m B_m + k_b k_m}$, de donde se obtiene un resultado aproximado de B_m :

$$B_m = \frac{J_m}{t_m} - \frac{k_b k_m}{R_m} \quad (2.5)$$

Se considerará un motor DC que satisface las siguientes relaciones de acoplo electromecánico:

$$e_b = k_b \dot{\theta}_m(t) \quad (2.6)$$

$$\tau_m = k_m i(t) \quad (2.7)$$

Con las ecuaciones anteriormente presentadas y considerando el motor sin carga, se pueden reescribir las siguientes ecuaciones eléctrica y mecánica del motor:

$$u_m(t) = R_m i(t) + L_m \frac{di(t)}{dt} + k_b \dot{\theta}_m(t) \quad (2.8)$$

$$k_m i(t) = J_m \ddot{\theta}_m(t) + B_m \dot{\theta}_m(t) \quad (2.9)$$

Considerando la función de transferencia de velocidad angular del motor como aquella en la que la entrada es $u_m(t)$ y la salida es $\dot{\theta}_m(t)$

$$G_{\dot{\theta}_m}(t) = \frac{\dot{\theta}_m(t)}{u_m(t)} \quad (2.10)$$

y realizando su transformada de Laplace bajo condiciones iniciales nulas, y eliminando la variable de la corriente eléctrica $I(s)$ se obtiene:

$$G_{\dot{\theta}_m}(s) = \frac{k_m}{(J_m s + B_m)(L_m s + R_m) + k_b k_m} \quad (2.11)$$

Siendo la ganancia a bajas frecuencias del motor:

$$G_{\dot{\theta}_m}(0) = \frac{k_m}{B_m R_m + k_b k_m} \quad (2.12)$$

esta ecuación tiene dos polos reales que pueden representarse como

$$p_{1,2} = -\frac{1}{2} \left(\frac{1}{t_e} + \frac{1}{t'_m} \right) \pm \frac{1}{2} \sqrt{\left(\frac{1}{t_e} - \frac{1}{t'_m} \right)^2 - 4 \frac{k_m k_b}{J_m L_m}} \quad (2.13)$$

donde $t'_m = \frac{J_m}{B_m}$ se obtiene eliminando el acoplo eléctrico en las ecuaciones del motor. Utilizando el valor de B_m obtenido por el método de la constante de tiempo mecánica, el valor de los polos es:

$$p_1 = -3433,6$$

$$p_2 = -211,98$$

El polo dominante es p_2 , que además cumple que $|p_2| \ll |p_1|$, por lo que la función de transferencia ahora puede escribirse como:

$$G_{\dot{\theta}_m}(s) = \frac{K'_m}{(s - p_1)(s - p_2)} \quad (2.14)$$

calculando K'_m con la expresión dada por $K'_m = \frac{k_m}{J_m L_m}$.

Los motores DC normalmente cumplen que $t_e \ll t_m$ además de que $p_1 \ll p_2$, como es este caso. Esto sugiere dos métodos distintos de simplificación de las ecuaciones del motor: eliminar la constante de tiempo eléctrica t_e o eliminar el polo no dominante p_1 . Los resultados serán distintos, aunque similares, pero ambos métodos nos llevan a obtener una función de transferencia dada por la Ecuación 2.1 que es lo que se desea.

Utilizando el método de simplificación de eliminación de la constante eléctrica del motor, teniendo en cuenta que la constante eléctrica del motor es mucho menor que la constante mecánica, lo que se traduce en despreciar el factor de la inductancia L_m en la ecuación eléctrica, se obtiene $p_m = -\frac{1}{t_m}$ y $K_m = p_m G_{\dot{\theta}_m}(0)$

Obteniéndose finalmente el modelo simplificado del motor:

$$G_{\dot{\theta}_m}(s) = \frac{1642}{s + 200} \quad (2.15)$$

2.2.2. Diseño de un controlador de posición

Aunque en el apartado anterior se ha obtenido la función de transferencia de velocidad angular del motor, en este caso se va a hacer uso de la función de transferencia de posición, que se consigue fácilmente ya que $G_{\theta_m}(s) = \frac{1}{s} G_{\dot{\theta}_m}(s)$.

Para probar el correcto funcionamiento de los motores se ha diseñado un controlador de posición de tipo P (ver Figura 2.6) (Monasterio-Huelin, 2016d).

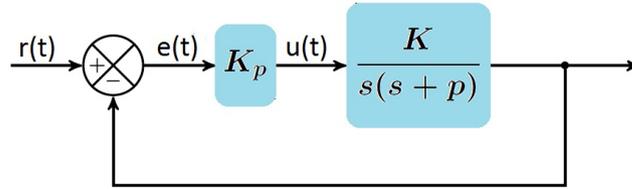


Figura 2.6: Esquema de bloques de un controlador P para un sistema continuo de segundo orden y tipo uno.

En la Figura 2.6 la constante K_p es la constante proporcional del controlador de posición y el siguiente bloque es la función de transferencia de posición del motor. Por lo tanto, lo que se desea obtener es la constante K_p .

La función de transferencia de lazo cerrado $H_P(s)$ tiene la forma:

$$H_P(s) = \frac{K_P K}{s^2 + ps + K_P K} \quad (2.16)$$

Esta función coincide con la primera forma canónica de los sistemas de segundo orden dada por:

$$G_1(s) = \frac{Y(S)}{U(S)} = \frac{\omega_n^2}{s^2 + 2\alpha s + \omega_n^2} \quad (2.17)$$

Esta ecuación se obtiene realizando la Transformada de Laplace a la ecuación diferencial que se corresponde con la respuesta de un sistema de segundo orden a la entrada escalón unidad,

$$\ddot{y}(t) + 2\alpha\dot{y}(t) + \omega_n^2 y(t) = \omega_n^2 u(t) \quad (2.18)$$

donde $\omega_n \in \mathfrak{R}^+$ es la frecuencia natural, $\alpha \in \mathfrak{R}^+$ es el coeficiente de atenuación, $u(t) \in \mathfrak{R}$ la entrada e $y(t) \in \mathfrak{R}$ la salida del sistema continuo.

Si vemos la analogía entre las ecuaciones 2.16 y 2.17 se puede apreciar que la frecuencia natural y el coeficiente de amortiguamiento vienen dados por

$$\omega_n = \sqrt{K_P K} \quad (2.19)$$

$$\zeta = \frac{p}{2\sqrt{K_P K}} = \frac{p}{2\omega_n} \quad (2.20)$$

Para el diseño del controlador P conviene elegir el parámetro ζ , ya que este nos indica el tipo de salida obtenida, ésta puede ser (ver Figura 2.7):

- Sistema de segundo orden subamortiguado: $\zeta < 1$. Los polos serán complejos y conjugados y se encuentran en el semiplano negativo del dominio complejo. La respuesta impulsional para un sistema subamortiguado es una combinación de una exponencial monótonamente decreciente con el tiempo y un armónico de frecuencia $w_d = \omega_n \sqrt{1 - \zeta^2}$
- Sistema de segundo orden críticamente amortiguado: $\zeta = 1$. Las dos raíces son iguales y negativas.
- Sistema de segundo orden sobreamortiguado: $\zeta > 1$. Los polos del sistema serán reales, negativos y distintos, la respuesta temporal es la combinación de dos sistemas de primer orden.

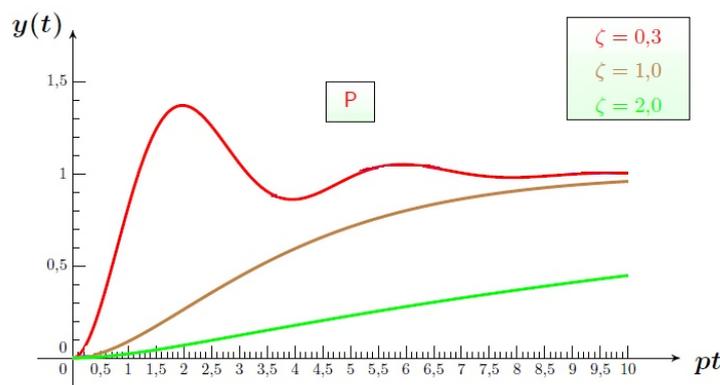


Figura 2.7: Distintos tipos de salida para un controlador P según ζ (Monasterio-Huelin, 2016d)

Las características de régimen transitorio de los sistemas continuos de segundo

orden son:

$$M_p = e^{-\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\right)\pi}$$

$$t_p = \frac{\pi}{\omega_d}$$

$$t_r = \frac{\pi - \varphi_0}{\omega_d}$$

$$t_s = \frac{\ln\left(\frac{1}{\nu\sqrt{1-\zeta^2}}\right)}{\zeta\omega_n}$$

donde ν es la tolerancia que se le permite a la salida respecto a la entrada en régimen permanente, y

$$\text{sen } \varphi_0 = \sqrt{1 - \zeta^2}$$

$$\text{cos } \varphi_0 = \zeta$$

donde M_p es la sobreelongación máxima, t_p es el tiempo de pico, es decir, el tiempo en el que la señal de salida consigue esa sobreelongación, t_r es el tiempo de subida en el que la salida alcanza el valor de la señal de entrada en este caso el valor unidad de la señal escalón, y t_s es el tiempo de establecimiento que marca una frontera entre el régimen permanente práctico y el régimen transitorio.

Los parámetros de diseño escogidos han sido $M_p = 15\%$ de la señal de referencia, ya que se desea que la señal de salida del controlador sea subamortiguada y no posea mucha oscilación. Fijando el valor de M_p obtenemos el coeficiente de amortiguamiento obteniéndose un valor de $\zeta = 0,5169$ que es menor de 1 y por tanto el sistema es subamortiguado. Ahora se obtiene ω_n a partir de la Ecuación 2.20, y finalmente ya se puede obtener K_p , que es el valor necesario para la implementación del controlador proporcional de posición. Mediante la Ecuación 2.19, se obtiene un valor de $K_p = 22,7909$ y su respuesta teórica simulada en Matlab se muestra en la Figura 2.8:

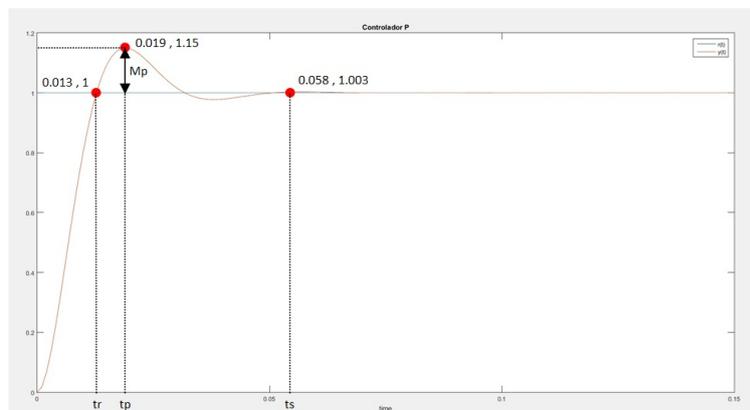


Figura 2.8: Respuesta de un controlador proporcional de posición a la entrada escalón

2.3. Estudio cinemático de la caja entrenadora

2.3.1. Cinemática directa

Al problema de obtener la posición y orientación en el espacio cartesiano del extremo final de las pinzas laparoscópicas, a partir de las coordenadas generalizadas, se le denomina problema cinemático directo. Se van a obtener las coordenadas x, y, z en función de los ángulos conocidos de las articulaciones y los parámetros geométricos de los elementos del laparoscopio (Mark W. Spong, Seth Hutchinson, 2006) (ver Figura 2.9).

El movimiento hacia el centro de la pared de la caja entrenadora o hacia los laterales que puede realizar el brazo laparoscópico se corresponde con un cambio del ángulo q_1 (ver Figura 2.9(a)) y supone una rotación respecto al eje x , siendo su matriz de rotación:

$$R_x(q_1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos q_1 & -\text{sen } q_1 \\ 0 & \text{sen } q_1 & \cos q_1 \end{pmatrix} \quad (2.21)$$

El movimiento que puede realizar el brazo laparoscópico hacia la pared de la caja entrenadora o alejándose de la misma se corresponde con un cambio del ángulo q_2 (ver Figura 2.9(b)) y supone una rotación respecto al eje y , siendo su matriz de rotación:

$$R_y(q_2) = \begin{pmatrix} \cos q_2 & 0 & \text{sen } q_2 \\ 0 & 1 & 0 \\ -\text{sen } q_2 & 0 & \cos q_2 \end{pmatrix} \quad (2.22)$$

El movimiento de subida y bajada que puede realizar el brazo laparoscópico se corresponde con una traslación de la distancia l respecto al eje z de las pinzas laparoscópicas (ver Figura 2.9(c)), siendo su matriz de traslación:

$$T_z(l) = \begin{pmatrix} 0 \\ 0 \\ l \end{pmatrix} \quad (2.23)$$

A partir de estas matrices se va a obtener es una ecuación matricial del tipo:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = X(q) \quad (2.24)$$

siendo $X(q) = R_x(q_1) * R_y(q_2) * T_z(l)$

Tras la multiplicación de estas tres matrices se obtiene la matriz resultante:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} l \text{sen } q_2 \\ -l \text{sen } q_1 \cos q_2 \\ l \cos q_1 \cos q_2 \end{pmatrix} \quad (2.25)$$

lo que representa las tres coordenadas cartesianas especificadas según las coordenadas generalizadas del robot.

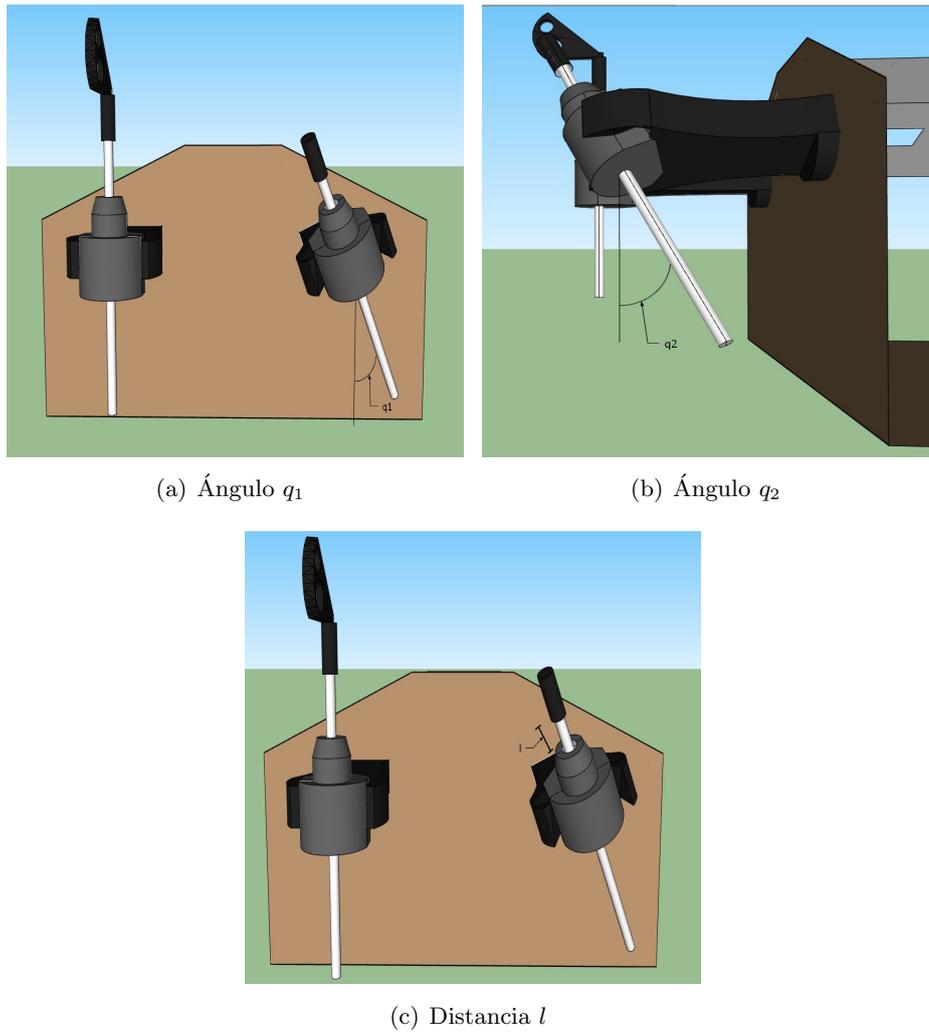


Figura 2.9: Coordenadas generalizadas

2.3.2. Cinemática inversa

El problema cinemático inverso de posición consiste en obtener las coordenadas generalizadas a partir de las coordenadas de un punto fijo en el extremo de las pinzas laparoscópicas, es decir, determinar la configuración que debe adoptar el robot para una posición y orientación del extremo conocidas (Monasterio-Huelin, 2016a). En este caso no se va a tener en cuenta la orientación ya que esta funcionalidad no va a ser implementada en lo relativo a la realimentación háptica.

Para obtener las coordenadas generalizadas (q_1, q_2, l) a partir de las coordenadas cartesianas (x, y, z) se debe resolver la Ecuación matricial 2.25.

Si se divide la coordenada y entre la coordenada z

$$\frac{y}{z} = -\tan q_1 \quad (2.26)$$

se obtiene la coordenada q_1

$$q_1 = \arctan\left(-\frac{y}{z}\right) \quad (2.27)$$

Elevando al cuadrado las coordenadas y y z

$$\begin{aligned} y^2 &= l^2 \sin^2 q_1 \cos^2 q_2 \\ z^2 &= l^2 \cos^2 q_1 \cos^2 q_2 \end{aligned}$$

sumando ambas ecuaciones y realizando su raíz cuadrada, se obtiene

$$\sqrt{y^2 + z^2} = l \cos q_2 \quad (2.28)$$

tomando la coordenada x y la Ecuación 2.28 y dividiendo ambas ecuaciones se llega a

$$\frac{x}{\sqrt{y^2 + z^2}} = \tan q_2 \quad (2.29)$$

obteniendo que la coordenada q_2 es

$$q_2 = \arctan\left(\frac{x}{\sqrt{y^2 + z^2}}\right) \quad (2.30)$$

Finalmente se eleva al cuadrado la coordenada x y la Ecuación 2.28

$$\begin{aligned} x^2 &= l^2 \sin^2 q_2 \\ y^2 + z^2 &= l^2 \cos^2 q_2 \end{aligned}$$

y se suman ambas ecuaciones

$$x^2 + y^2 + z^2 = l^2 \quad (2.31)$$

permitiendo obtener la coordenada l , como:

$$l = \sqrt{x^2 + y^2 + z^2} \quad (2.32)$$

En resumen, las ecuaciones de cinemática inversa son:

$$\begin{cases} q_1 = \arctan\left(\frac{-y}{z}\right) \\ q_2 = \arctan\left(\frac{x}{\sqrt{y^2 + z^2}}\right) \\ l = \sqrt{x^2 + y^2 + z^2} \end{cases}$$

2.3.3. Pares de fuerza

Ahora se va a incorporar una fuerza externa virtual al modelo del motor, esta fuerza virtual, F_v , es aplicada en el extremo de las pinzas laparoscópicas cuando éstas entran en contacto con el objeto virtual simulado (Monasterio-Huelin, 2016b). Entonces, aparecen los pares de fuerza que hay que aplicar a cada motor, que tienen la forma,

$$\tau_v(t) = J^T(q(t))F_v(t) \quad (2.33)$$

donde J representa el jacobiano del robot, el cual depende de las coordenadas generalizadas y se obtiene de hacer las derivadas parciales de las coordenadas x, y, z respecto del tiempo

$$J(q(t)) = \frac{dX(t)}{dt} = \frac{dX(q(t))}{dq(t)} \frac{dq(t)}{dt} \quad (2.34)$$

Si se deriva, se obtiene:

$$J^T(q(t)) = \begin{pmatrix} 0 & -l \cos q_1 \cos q_2 & -l \sin q_1 \cos q_2 \\ l \cos q_2 & l \sin q_1 \sin q_2 & -l \cos q_1 \sin q_2 \\ \sin q_2 & -\sin q_1 \cos q_2 & \cos q_1 \cos q_2 \end{pmatrix} \quad (2.35)$$

$$F_v(t) = \begin{pmatrix} f_{vx}(t) \\ f_{vy}(t) \\ f_{vz}(t) \end{pmatrix} \quad (2.36)$$

La demostración de la relación 2.33 deriva del principio de los trabajos virtuales de D'Alembert, cuya idea es considerar el caso de una situación estática del mecanismo, sobre el que se realiza un desplazamiento infinitesimal o virtual por la acción de la fuerza externa, y compatible con las ligaduras existentes en el mecanismo, las cuales, por la hipótesis de rigidez, no realizan trabajo (Monasterio-Huelin, 2016b).

De esta manera los pares de fuerza de cada movimiento obtenidos son:

$$\tau_v(t) = \begin{pmatrix} 0 & -l \cos q_1 \cos q_2 & -l \sin q_1 \cos q_2 \\ l \cos q_2 & l \sin q_1 \sin q_2 & -l \cos q_1 \sin q_2 \\ \sin q_2 & -\sin q_1 \cos q_2 & \cos q_1 \cos q_2 \end{pmatrix} \begin{pmatrix} f_{vx}(t) \\ f_{vy}(t) \\ f_{vz}(t) \end{pmatrix} \quad (2.37)$$

q_1, q_2 y l se corresponden con un cambio en la posición de los brazos laparoscópicos respecto a los ejes de coordenadas cartesianas centrados en la articulación. Para transportar estos cambios a los ejes de los motores se debe llevar a cabo una transformación de reducción:

$$\begin{aligned} \theta_1(t) &= \frac{R_1}{r} q_1 \\ \theta_2(t) &= \frac{R_2}{r} q_2 \\ \theta_3(t) &= \frac{l}{r} \end{aligned}$$

donde r es la medida del radio del motor, R_1 es el radio de la circunferencia que forma el movimiento del ángulo q_1 , R_2 es el radio de la circunferencia que forma el movimiento del ángulo q_2 y l es el desplazamiento que se produce cuando se realiza el movimiento del brazo de subida o bajada.

Por lo tanto, los pares de fuerza en el motor quedan de la siguiente manera:

$$\begin{cases} \tau_{\theta_1} = \frac{r}{R_1}(-l \cos q_1 \cos q_2) f_{vy} - (l \sin q_1 \cos q_2) f_{vz} \\ \tau_{\theta_2} = \frac{r}{R_2}((l \cos q_2) f_{vx} + (l \sin q_1 \sin q_2) f_{vy} - (l \cos q_1 \sin q_2) f_{vz}) \\ \tau_{\theta_3} = \frac{r}{l}((\sin q_2) f_{vx} - (\sin q_1 \cos q_2) f_{vy} + (\cos q_1 \cos q_2) f_{vz}) \end{cases}$$

Una vez obtenido los pares de fuerza de cada movimiento, se debe obtener la tensión eléctrica de entrada a los motores para que se produzca el movimiento de los mismos según la fuerza recibida del sistema virtual de simulación. Ésto se obtiene mediante la utilización de las ecuaciones eléctrica 2.8 y mecánica 2.9 del motor, despreciando el factor de la inductancia L_m en la ecuación eléctrica debido a que $t_e \ll t_m$ y añadiendo el par de carga visto desde el eje del motor a la ecuación mecánica:

$$u_m(t) = R_m i(t) + k_b \dot{\theta}_m(t) \quad (2.38)$$

$$k_m i(t) = J_m \ddot{\theta}_m(t) + B_m \dot{\theta}_m(t) + \tau_l(t) \quad (2.39)$$

Despejando la corriente eléctrica $i(t)$ de la ecuación mecánica 2.39 y sustituyendo en la ecuación eléctrica 2.38, se obtiene la ecuación simplificada del motor DC,

$$u_m(t) = \frac{R_m J_m}{k_m} \ddot{\theta}_m(t) + \left(\frac{R_m B_m}{k_m} + k_b \right) \dot{\theta}_m(t) + \frac{R_m}{k_m} \tau_l(t) \quad (2.40)$$

Se van a despreciar los términos relacionados con la aceleración y la velocidad del motor debido a la complejidad de cálculo para obtenerlos y la suposición de un modelo ideal. En consecuencia, se va a considerar que el par de carga que afecta al motor solo es el par de fuerza virtual, quedando la ecuación simplificada:

$$u_m(t) = \frac{R_m}{k_m} \tau_v(t) \quad (2.41)$$

siendo R_m y k_m la resistencia terminal y la constante del par del motor respectivamente.

Capítulo 3

Realimentación háptica

3.1. Sistema de bloques

Para el correcto funcionamiento de todo el sistema y percibir una realimentación de fuerza en la caja entrenadora se deben realizar ciertas tareas:

- Enviar la trama con los cambios de posición realizados para la correcta visualización de las pinzas laparoscópicas en el entorno virtual.
- Procesar la interacción de fuerza que se produce en el entorno virtual entre las pinzas laparoscópicas y la membrana simulada.
- Generar la trama con el ciclo de trabajo correspondiente con el movimiento de cada motor.
- Decodificar la trama recibida para proceder a la alimentación de los motores.

Para cubrir estas necesidades se va a desarrollar un sistema que consta de tres bloques que conforman el lazo cerrado de realimentación háptica (ver Figura 3.1).

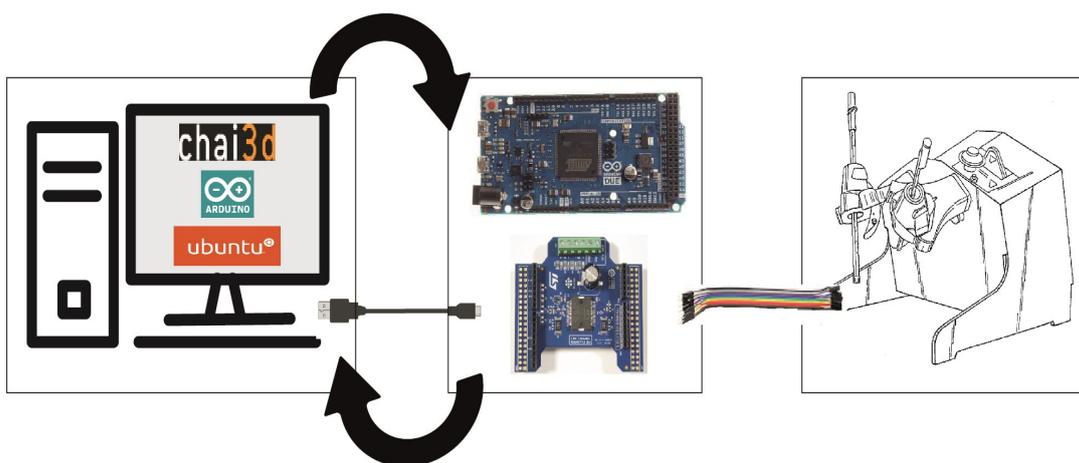


Figura 3.1: Sistema de bloques del sistema

Estos bloques están contituidos por los siguientes elementos software y hardware específicos cuya funcionalidad se explicará en detalle en secciones posteriores:

- CHAI 3D, programa necesario para la visualización del movimiento realizado por las pinzas laparoscópicas y procesar la fuerza de interacción.
- Tarjeta Arduino Due encargada de enviar por el puerto serie la información recogida por los encoders y de leer la trama proveniente del programa CHAI 3D.
- Núcleo X-NUCLEO-IHM04A1, tarjeta de expansión que se encarga del control de los motores.
- Caja entrenadora laparoscópica, eje central del Trabajo de Fin de Grado.

Se necesitará un PC con el sistema operativo Ubuntu y los programas CHAI 3D y Arduino instalados. La conexión entre el PC y la placa Arduino Due se realiza mediante USB, mientras que la conexión entre la tarjeta de expansión de control de los motores y la caja entrenadora se realizará mediante cables jumpers. La comunicación debe ser bidireccional para que se produzca un lazo cerrado de información que fluye del PC al Arduino y en sentido contrario.

3.2. Hardware

3.2.1. Arduino Due

Arduino Due es una tarjeta de desarrollo basada en el microcontrolador Atmel SAM3X8E ARM Cortex-M3 de 32bit, dispone de 54 pines de entrada/salida digitales y un reloj de 84MHz (Arduino, 2016).

Para el desarrollo que se va a llevar a cabo dispone de pines suficientes, aunque será necesario subir la frecuencia de los mismos para el control de los motores como se explicará más adelante. Los pines a utilizar son 16 para la lectura de los encoders que funcionarán como entradas y 24 actuando como señales de salida para el control de los motores.

3.2.2. Tarjeta de expansión de controlador de motores DC

Para realizar el control y alimentación de los motores se ha utilizado el X-NUCLEO-IHM04A1, una tarjeta de expansión para el control dual de motores DC (ver Figura 3.2).

- Posee 4 salidas (A^+ , A^- , B^+ , B^-) para el control bidireccional de dos motores.
- Tiene 2 entradas de alimentación GND y VCC, esta última permite una tensión de alimentación de la etapa de potencia de 8V a 50V.
- Dispone de 8 entradas para conexionado con Arduino, dos de ellas GND, las 6 restantes se utilizan para el control de los motores y son las señales de entrada de la etapa de potencia y se corresponden con 2 señales de Enable, que se activan cuando se encuentran a nivel alto y 4 señales PWM que activan los motores y controlan su fuerza y sentido de giro.



Figura 3.2: X-NUCLEO-IHM04A1
(STMicroelectronics, 2015)

Esta tarjeta de expansión está basada en el L6206 (DMOS dual full bridge driver), se trata de un puente en H que actúa como etapa de potencia y permite a un motor DC girar en ambos sentidos. En las figuras 3.3 y 3.4 se muestra uno de los dos puentes en H con los que cuenta la tarjeta de expansión. Llamaremos Zona Z^+ al periodo de tiempo en que los DMOS's A1 y A2 se encuentren en su región óhmica y los DMOS's B1 y B2 se encuentren en su región de corte, y Zona Z^- al periodo de tiempo en que los DMOS's B1 y B2 se encuentren en su región óhmica y los DMOS's A1 y A2 se encuentren en su región de corte. De esta manera se crea un voltaje positivo (U_m^+) y negativo (U_m^-) en los bornes del motor situado entre las salidas $OUT1_A$ y $OUT2_A$ permitiéndole girar en ambos sentidos.

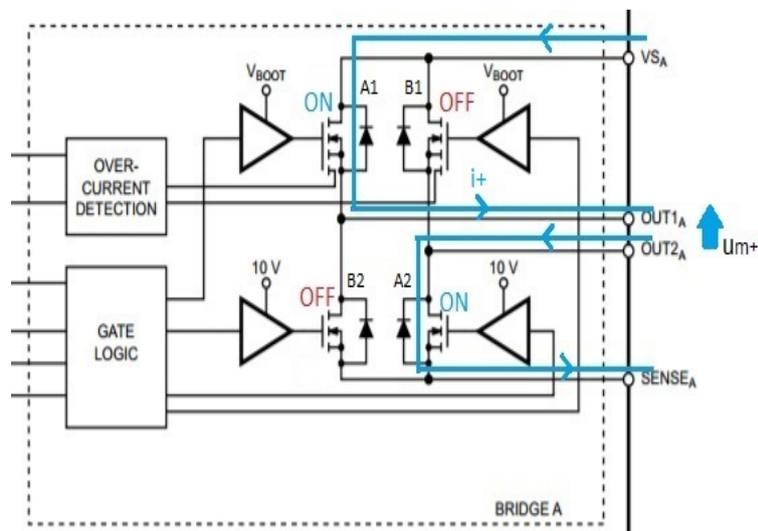
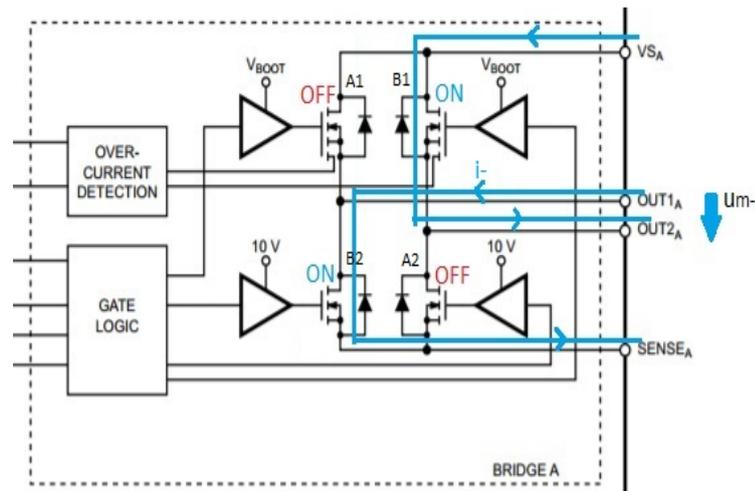


Figura 3.3: Zona Z^+

Figura 3.4: Zona Z^-

3.2.3. Adaptación de las señales de los motores

Para conectar los motores a las tarjetas de expansión, ha sido necesario crear una placa de acondicionamiento para estructurar las señales y facilitar su conexión (ver Figura 3.5).

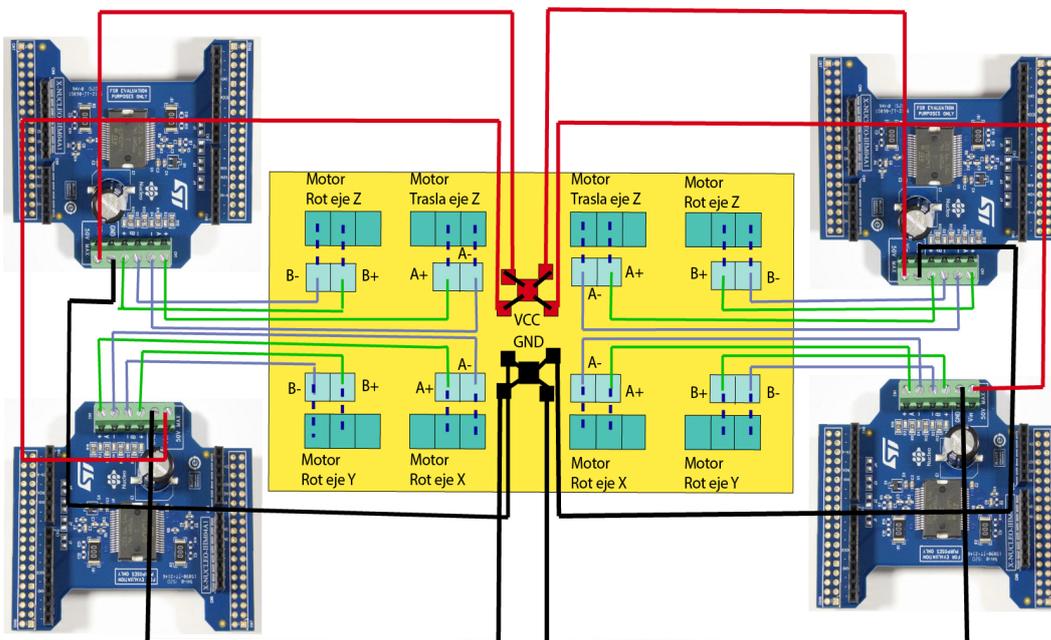


Figura 3.5: Placa de conexión

En esta tarjeta se encuentran conectados los 8 motores correspondientes a las dos pinzas laparoscópicas. Éstos están conectados a las señales de salida provenientes de los 4 X-NUCLEO-IHM04A1, cada uno de ellos se encuentra conectado a las dos señales de salida de la etapa de potencia. También se encuentran las señales de alimentación

VCC y GND provenientes de la fuente de alimentación y distribuidas a las 4 tarjetas de expansión.

3.3. Sistema en tiempo real

Para que el sistema funcione de forma correcta y transparente para el usuario es necesario que se cumplan ciertos requisitos de tiempo en el proceso de realimentación háptica. El operador humano del dispositivo háptico debe percibir el mundo virtual sobre el que actúa con la máxima fidelidad posible. A esta exigencia se le denomina problema de la transparencia háptica (Monasterio-Huelin, 2016b).

3.3.1. Requisitos de tiempo real

Hay que tener en cuenta que la percepción háptica requiere una frecuencia de muestreo muy elevada para que se perciba una continuidad en los movimientos. En visión hacen falta unas treinta imágenes/s (30Hz), en lo relacionado a lo háptico hacen falta señales de entre 500 a 1000Hz.

En cualquier caso, siempre se producirán errores entre la posición exacta del extremo del dispositivo háptico y las coordenadas en el espacio virtual. Esto es debido a los errores numéricos que resultan de resolver el problema cinemático directo y a la discretización que se lleva a cabo al obtener la posición real a partir de la información de los encoders y transmitirla al mundo virtual. Estos hechos plantean un problema relacionado con la realimentación háptica, denominado problema de la estabilidad háptica. La inestabilidad háptica es debida, normalmente, a la selección de un tiempo de muestreo excesivamente lento, y suele percibirse como oscilaciones en el extremo del dispositivo háptico cuando se produce la interacción con un objeto virtual (Monasterio-Huelin).

3.3.2. Puertos USB disponibles en Arduino Due

La comunicación entre el PC y la placa Arduino Due se realiza mediante conexión USB. Arduino Due dispone de dos tipos de conexión USB. El puerto Native USB que está conectado directamente al microcontrolador SAM3X y el puerto de Programación conectado al ATMEL 16U2 (Arduino, 2016).

- El puerto USB de Programación es el convertidor USB-Serial, está conectado a la primera UART del SAM3X. Éste es el puerto que se utiliza por defecto para subir los sketches y realizar la comunicación con Arduino Due.
- El conector USB del puerto Native está conectado directamente a los pines del host USB del SAM3X. El uso del puerto Native permite utilizar el dispositivo Due como cliente USB o como un dispositivo host USB para que los dispositivos puedan conectarse a Arduino Due.

Al utilizar el puerto Native no es necesario fijar ninguna tasa de baudios, ya que el Arduino Due nos ofrecerá la mayor tasa disponible, lo que nos permitirá transmitir los datos a una mayor velocidad. Un inconveniente de utilizar este puerto USB es que la memoria flash del microcontrolador SAM3X no se borra

al subir un nuevo sketch, por lo tanto, hay que realizar el borrado de manera manual pulsando el botón de reset disponible en la placa.

3.3.3. Medida de tiempos

Se han realizado pruebas de transmisión de distintos caracteres alfanuméricos, transmitiendo datos en forma de String. Se han utilizado ambos puertos USB, Native y de Programación, para probar si se obtiene una mejora en la velocidad de transmisión.

Para medir el tiempo de ejecución se ha utilizado la función de Arduino *micros()*, esta función devuelve el número de microsegundos desde que el microcontrolador comenzó a ejecutar el programa (ver Tablas 3.1 y 3.2).

USB	Función	Tiempo de ejecución
Native	SerialUSB.println()	7 μ s
Programación	Serial.println()	10 μ s

Tabla 3.1: Tiempo de envío de un carácter

USB	Función	Tiempo de ejecución
Native	SerialUSB.println()	8 μ s
Programación	Serial.println()	40 μ s

Tabla 3.2: Tiempo de envío de ocho caracteres

Se observa que a media que aumentamos el número de caracteres también aumenta el tiempo de transmisión utilizando el USB de Programación, en cambio en el tiempo de transmisión con el USB Native se produce un incremento mínimo.

Por lo tanto, para conseguir una elevada frecuencia que evite el problema de la estabilidad háptica es necesario utilizar el puerto USB Native, ya que si utilizamos el puerto recomendado, el de Programación, la frecuencia de transmisión obtenida no es suficiente para percibir una continuidad en la realimentación háptica con frecuencia de 1kHz.

Una vez fijado el uso de USB Native para la transmisión de datos, se va a proceder a medir los tiempos de cómputo que se requieren en el proceso de realimentación háptica. Las medidas realizadas en la tarjeta se realizan nuevamente con la función *micros()* y las realizadas en el PC del programa CHAI 3D se realizan con la función *clock_gettime()* mediante el reloj *CLOCK_MONOTONIC*. Este reloj representa el tiempo absoluto transcurrido desde un punto arbitrario y fijo en el pasado, y no se ve afectado por los cambios en el reloj del tiempo del sistema. Se han tomado un conjunto de 10 medidas y el resultado mostrado es el mayor obtenido (ver Tabla 3.3).

Observando la Tabla 3.3 vemos que el tiempo empleado en la parte visual es holgado para realizar un muestreo con una frecuencia de 30Hz sin que se solapen los procesos. En cambio, en los procesos relacionados con la parte háptica, aunque

Entorno	Tarea	Tiempo de ejecución
Arduino	Enviar trama de los encoders a CHAI 3D	275 μ s
Arduino	Recibir trama del ciclo de trabajo de CHAI 3D	1263 μ s
CHAI 3D	Posiciones y gráfica	1949 μ s
CHAI 3D	Procesado de trama de los encoders	2196 μ s
CHAI 3D	Cómputo de fuerzas	1156 μ s

Tabla 3.3: Tiempo de ejecución de las diferentes tareas

conlleven un menor tiempo de cómputo, éste es demasiado elevado para conseguir una frecuencia de muestreo de 1KHz. Por lo tanto, se ha fijado un tiempo de muestreo de 1,33ms, $\frac{1}{1,33ms} = 751,88Hz \cong 750Hz$.

3.4. Software

3.4.1. CHAI 3D

Para la simulación virtual del tejido que interactúa con el entrenador laparoscópico se ha obtenido por el programa CHAI 3D ¹, ya que es de fácil manejo y uso. CHAI3D consiste en un conjunto de bibliotecas de código abierto en C++ para la manipulación informática, la visualización y la simulación interactiva en tiempo real. CHAI3D soporta una serie de dispositivos hápticos disponibles comercialmente de diferentes grados de libertad, y facilita el soporte de nuevos dispositivos personalizados de retroalimentación de fuerza (CHAI3D, 2016). Para realizar la comunicación entre este software y la tarjeta Arduino Due es necesario que se realice la apertura del puerto serie que conecta el PC con Arduino, esto permitirá tanto la lectura como la escritura de las tramas.

Este software devuelve el vector de fuerzas externas de interacción en el sistema de coordenadas cartesianas ($F_v(t)$), esta fuerza externa se produce cuando las pinzas laparoscópicas interactúan con el tejido virtual simulado. Para ello es necesario crear dos dispositivos hápticos, uno para cada pinza, que se sitúan en la parte final de las mismas (implementados en color rojo). Una vez obtenida la fuerza externa se generan los pares de fuerza correspondientes a cada motor y se calcula la tensión eléctrica de entrada a los motores, a partir de la cual se genera el ciclo de trabajo de los motores. De esta manera es posible realizar la realimentación de fuerza necesaria.

El tejido simulado es una membrana generada gracias al módulo GEL disponible en CHAI 3D. Este módulo se utiliza para simular cuerpos rígidos y deformables en tiempo real. Se trata de una red de nodos que reaccionan cuando el dispositivo háptico se encuentra en su interior, y pueden fijar las características de la membrana generada como la rigidez o la viscosidad. El proceso de realimentación háptica se realiza de forma continua, mediante la ejecución de una hebra. En este bucle se recorre el array de nodos de dos dimensiones y se comprueba mediante la función *computeForce()* si se produce un choque entre las pinzas laparoscópicas y la membrana (ver Figura 3.6).

Se ha implementado la funcionalidad de poder desplazar la cámara integrada en

¹CHAI 3D: <http://www.chai3d.org/>

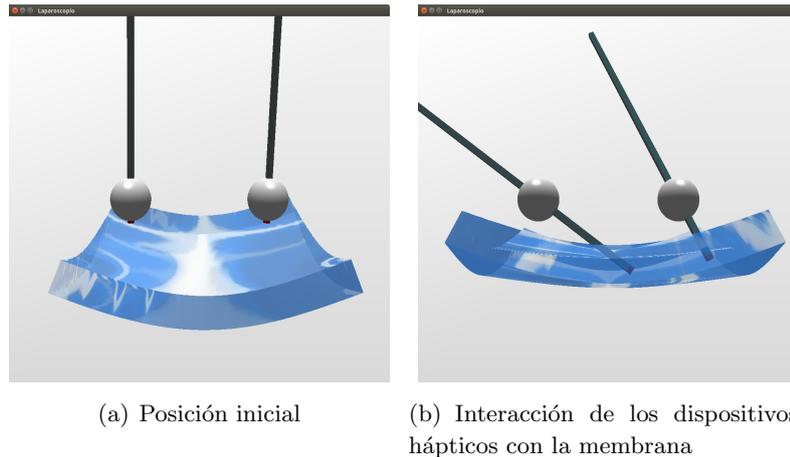


Figura 3.6: Visualización de las pinzas laparoscópicas y la membrana en CHAI 3D

el software para visualizar cómo interactúan las pinzas laparoscópicas desde todas las perspectivas.

3.4.2. Comunicación

La comunicación se realiza en ambos sentidos, del PC al Arduino Due y viceversa, ya que se tiene que producir una realimentación de fuerza en lazo cerrado.

- **Comunicación de Arduino al PC.** En este sentido se realiza el envío de la posición de los instrumentos al program CHAI3D, recogiendo la información proporcionada por los encoders instalados en cada motor.
- **Comunicación del PC a Arduino.** En este caso se envía el ciclo de trabajo correspondiente a cada motor.

Se va a desarrollar en profundidad la comunicación del PC con Arduino, ya que la comunicación en el sentido contrario ha sido implementada en el Trabajo Fin de Grado de Teresa Gómez (Gómez, 2016). En este Trabajo también se recoge cómo se ha realizado la simulación virtual de las pinzas laparoscópicas y la visualización de su movimiento en tiempo real.

El programa desarrollado en CHAI3D se encarga de enviar las tramas al Arduino Due por el puerto serie. Esta trama se envía en formato String, lo que nos permite enviar caracteres alfanuméricos. La trama está formada por un conjunto de caracteres que identifican de forma inequívoca cada motor, cada identificador va seguido por el ciclo de trabajo correspondiente.

- Los identificadores utilizados son letras que van de la «i» a la «p» (ver Tabla 3.4).
- El ciclo de trabajo de las señales de activación de cada motor tienen un rango de -100 a +100. El signo positivo o negativo del ciclo de trabajo indica el sentido de giro del motor (ver Tabla 3.5).

Identificador	Instrumento	Movimiento
i	Derecho	Rotación
j	Derecho	Dentro/Fuera
k	Derecho	Arriba/Abajo
l	Derecho	Izquierda/Derecha
m	Izquierdo	Rotación
n	Izquierdo	Dentro/Fuera
o	Izquierdo	Arriba/Abajo
p	Izquierdo	Izquierda/Derecha

Tabla 3.4: Identificadores de cada motor

Motores	Signo	Movimiento	Signo	Movimiento
j, n	Positivo	<i>Fuera</i> \rightarrow <i>Dentro</i>	Negativo	<i>Dentro</i> \rightarrow <i>Fuera</i>
k, o	Positivo	<i>Abajo</i> \rightarrow <i>Arriba</i>	Negativo	<i>Arriba</i> \rightarrow <i>Abajo</i>
l, m	Positivo	<i>Izquierda</i> \rightarrow <i>Derecha</i>	Negativo	<i>Derecha</i> \rightarrow <i>Izquierda</i>

Tabla 3.5: Movimiento de los motores según el signo del ciclo de trabajo

3.4.3. Implementación en Arduino

Para realizar el movimiento de los motores es necesario generar señales PWM. Éstas mantienen constante la frecuencia, es decir, el tiempo entre disparo de pulsos, mientras que se hace variar la anchura del pulso. La proporción de tiempo que está encendida la señal, respecto al total del ciclo, se denomina ciclo de trabajo, y se expresa en tanto por ciento. De esta manera se consigue variar la velocidad de un motor DC. En este caso tenemos señales PWM activas a nivel alto y bajo, lo que quiere decir que están totalmente activas cuando su valor promedio es VCC y cero respectivamente (ver Figura 3.7).

Para que los motores puedan realizar el movimiento deseado sin notar vibraciones o ruidos es necesario que las señales PWM que los activan sean de frecuencia elevada, en este caso se ha fijado un valor de 20KHz. Se necesitan un total de 16 señales PWM, dos para cada motor, ya que éste debe girar en ambos sentidos. Para ello será necesario utilizar dos tipos de señales PWM de alta frecuencia que nos ofrece Arduino Due, ya que las señales PWM por defecto tienen una frecuencia de aproximadamente 1KHz. Estos dos tipos de señales son las señales PWMC y las señales provenientes de los Timer.

- Las señales PWMC son señales PWM Hardware generadas directamente en el microcontrolador SAM3X. Arduino Due consta de 8 canales de PWMC numerados del 0 al 7 accesibles mediante los pines de salida 35, 36, 39, 41, 9, 8, 7, 6 (ver Tabla 3.6). Los canales disponen de dos señales activas a nivel alto y a nivel bajo siendo ambas señales de salida complementarias. Es necesario realizar la configuración de los canales activandolos y fijando el preescalado. En este caso se ha seleccionado un preescalado de 2100 para obtener la frecuencia deseada, ya que disponemos de una frecuencia de reloj accesible de 42MHz. Por

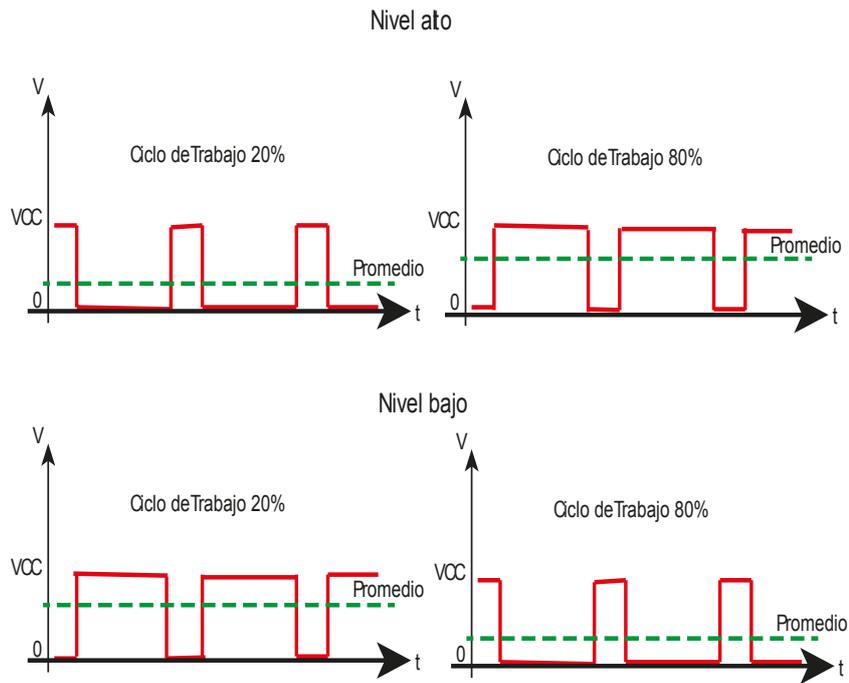


Figura 3.7: Señales con distintos tipos de ciclo de trabajo

lo tanto, $42MHz/2100 = 20KHz$. Estas señales se utilizarán para realizar el movimiento de la pinza laparoscópica izquierda, ya que solo es posible obtener 8 señales PWM independientes mediante este sistema.

Pines Arduino	Movimiento
35, 36	Rotación
39, 41	Dentro/Fuera
8, 9	Arriba/Abajo
6, 7	Izquierda/Derecha

Tabla 3.6: Pines de Arduino Due correspondientes al dispositivo izquierdo

- Las señales PWM que emplean los Timer generan una onda de salida utilizando un registro de comparación por cada salida. Cuando el tiempo alcanza el valor del registro de comparación, la salida invierte su valor. Cada salida conectada a un mismo temporizador comparte la misma frecuencia, aunque pueden tener distintos ciclos de trabajo, dependiendo del valor de su registro de comparación. Todas ellas activas a nivel bajo.

Se dispone de 3 Timers/Counters (TC0, TC1, TC2) cada uno formado por 3 canales, todos ellos asociados a dos líneas de salida de pines del Arduino Due, excepto TC1 que no dispone de pines de salida. Tenemos por lo tanto 12 salidas de PWM de las cuales se utilizarán solo 8 correspondientes a los pines de salida 2, 3, 4, 5, 10, 11, 12, 13 (ver Tabla 3.7). Estas salidas se encargan de activar los motores del dispositivo derecho. También se ha fijado un preescalado de 2100

para obtener la frecuencia deseada de 20KHz.

Pines Arduino	Movimiento
2, 3	Rotación
4, 5	Dentro/Fuera
10, 11	Arriba/Abajo
12, 13	Izquierda/Derecha

Tabla 3.7: Pines de Arduino Due correspondientes al dispositivo derecho

Ambas señales PWM son Unipolares, es decir, tienen un valor promedio que puede ir de cero a VCC (ver Figura 3.8).

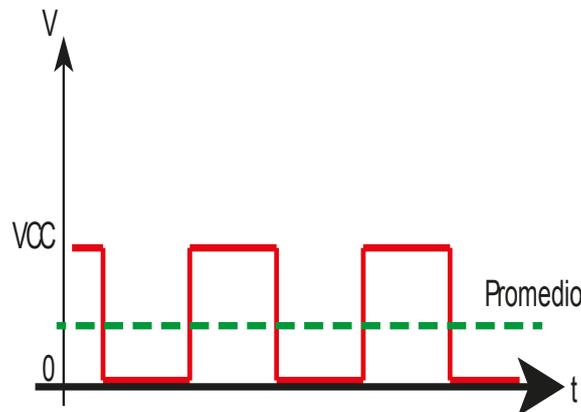


Figura 3.8: Señal PWM Unipolar

Los Timers anteriormente mencionados que no disponían de pines de salida serán utilizados para generar dos rutinas de atención a las interrupciones. Según la frecuencia de activación deseada se dispone de diferentes relojes. En este caso se utilizará el TIMERCLOCK1 que tiene una frecuencia de $84MHz/2 = 42MHz$.

- **Rutina encargada de enviar la trama de posicionamiento.** Esta interrupción se encarga de enviar el cambio de posición que se ha producido en las pinzas laparoscópicas según la información recogida por los encoders. En este caso el periodo de muestreo no es muy exigente, ya que se trata de la parte visual del sistema y como se ha comentado anteriormente con una frecuencia de 30Hz sería suficiente para no percibir un parpadeo en el refresco de las imágenes. Esta rutina se generará con una frecuencia de 30Hz y se utiliza un preescalado de 1400000 , $42MHz/1400000 = 30Hz$.
- **Rutina encargada de decodificar la trama de movimiento.** En esta rutina de atención a la interrupción se leen e interpretan los ciclos de trabajo de cada motor, es decir, se procede a la realimentación háptica de fuerza. En este caso la frecuencia de muestreo debe ser más elevada, entre 500-1000Hz. Debido a las

limitaciones de la placa de desarrollo y del software de simulación solo ha sido posible obtener una frecuencia de realimentación de 750Hz, que se obtiene con un preescalado de $56000, 42MHz/56000 = 750Hz$.

3.4.4. Experimentos

Para realizar la comprobación del correcto funcionamiento del todo el sistema se han llevado a cabo los siguientes experimentos:

- **Comprobación de la unión de los nodos de la membrana virtual.** El tamaño del radio de los nodos es un parámetro importante, ya que si estos son mucho más grandes que el dispositivo háptico, éste puede atravesar el tejido sin percibir ninguna sensación háptica, ya que se crean huecos entre los nodos por los que las pinzas laparoscópicas pueden atravesar la membrana sin recibir realimentación de fuerza. Por lo tanto, es necesario que los nodos tengan un tamaño similar o menor al dispositivo háptico para que la realimentación háptica sea lo más realista posible (ver Figura 3.9).

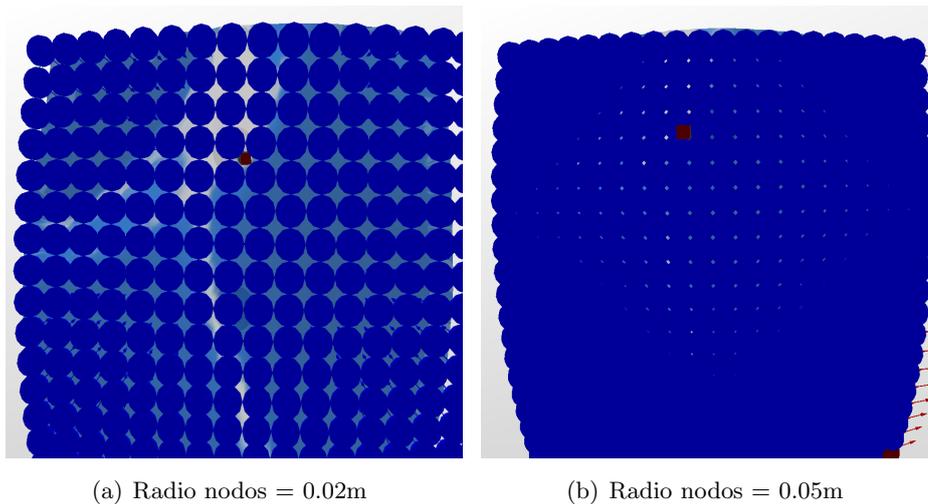


Figura 3.9: Diferentes radios de los nodos que conforman la membrana virtual

- **Realización de distintos tamaños de la membrana virtual.** Esto implica variar el número de nodos que forman la membrana. Se ha llevado a cabo esta prueba para que ambos dispositivos hápticos puedan tener acceso a la membrana a partir de la realización de cualquier movimiento. Un mayor tamaño de membrana implica un mayor tiempo de computación ya que el array de nodos a recorrer es mayor, lo que conlleva que no se cumplan las exigencias de tiempo del problema de estabilidad háptica. Finalmente se ha obtenido por una matriz de 20x20 nodos ya que es un tamaño suficiente para que ambos dispositivos puedan acceder a la membrana y se cumplan los requisitos de tiempo (ver Figura 3.10).
- **Dureza de la membrana virtual.** Una vez fijadas las características de los nodos que forman la membrana se han realizado pruebas con distintos valores

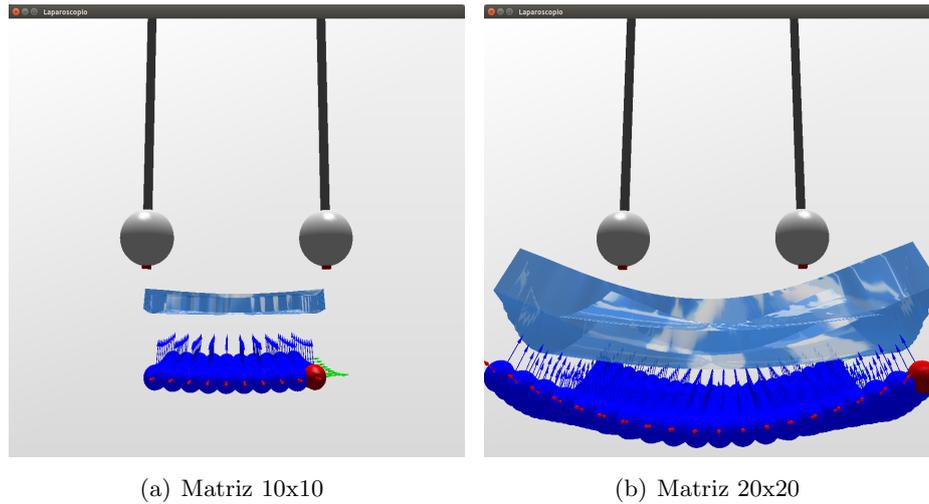
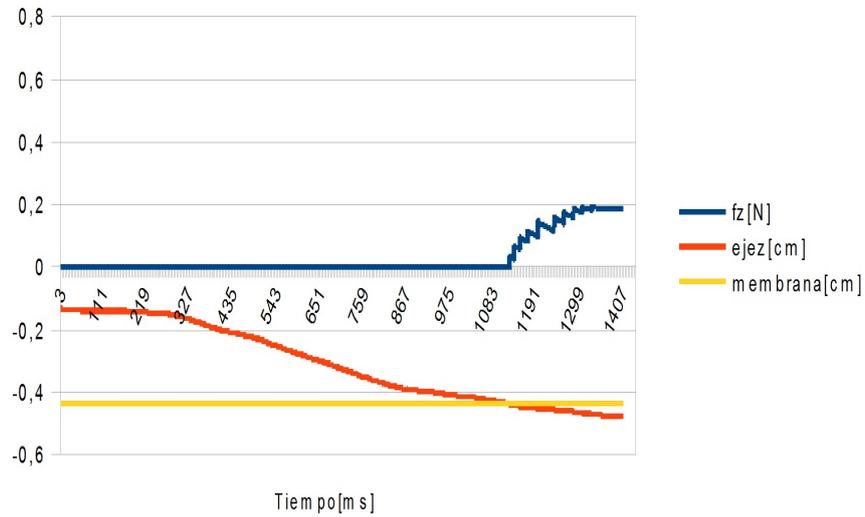
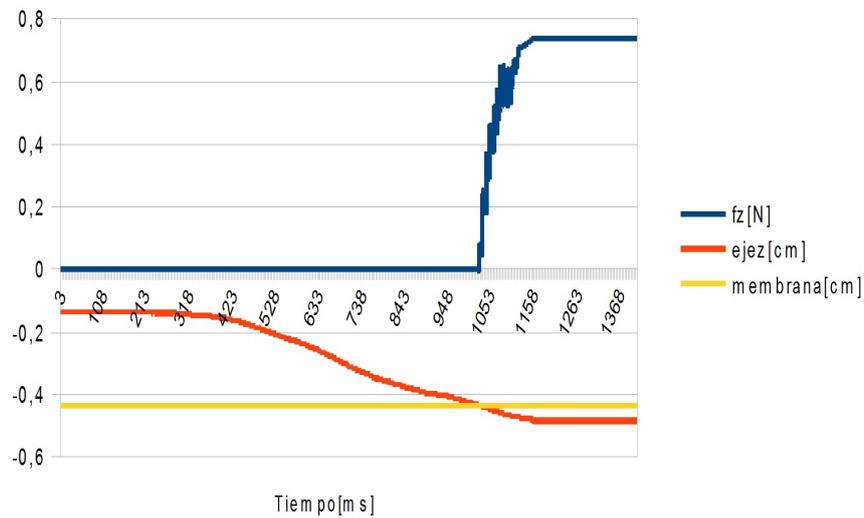


Figura 3.10: Diferentes tamaños de la matriz de nodos

del parámetro *stiffness*, que fija la rigidez de la membrana. Se ha comprobado que cuanto más pequeño es su valor más fácil es de atravesar la membrana, en cambio, si su valor es elevado resulta difícil atravesarla. Esto se ha realizado para fijar una proporcionalidad entre la fuerza de interacción recibida y el voltaje de entrada al motor (ver Figura 3.11).



(a) Stiffness = 20



(b) Stiffness = 80

Figura 3.11: Diferentes tamaños de la matriz de nodos

Capítulo 4

Conclusiones y líneas futuras

4.1. Conclusiones

En el presente Trabajo de Fin de Grado se ha realizado un sistema de realimentación de fuerza logrando los objetivos propuestos de tiempo real y sensación háptica. Para una fácil reproducción del sistema creado, los programas utilizados son de código abierto y fácil acceso, y el hardware es, exceptuando los motores, libre y de bajo coste.

El uso del programa CHAI 3D permite variar distintos parámetros del entorno virtual para lograr una realimentación háptica lo más realista posible. El entorno de desarrollo Arduino permite un fácil acceso de lectura y escritura al puerto USB, consiguiendo una comunicación bidireccional para crear una realimentación de lazo cerrado.

El hardware utilizado cumple con las expectativas, ya que es de fácil manejo y nos permite realizar el control de los motores de los que dispone la caja entrenadora. En lo relativo a la caja entrenadora no se ha hecho ninguna consideración, ya que éste no es el objetivo del Trabajo de Fin de Grado y se ha utilizado tal y como se encontraba el dispositivo.

La gran limitación que se ha encontrado es el tiempo de computación necesario para la realización de la realimentación háptica, aún así, se ha conseguido una frecuencia que permite superar los problemas de inestabilidad háptica.

Se ha creado un sistema que puede ser utilizado por estudiantes de cirugía laparoscópica y considerarse una primera aproximación a la realización de una cirugía real, aún habiendo mejoras que realizar para lograr un mayor realismo en los tejidos simulados.

4.2. Líneas futuras

Como líneas futuras para la mejora del actual sistema se realizan las siguientes propuestas:

- **Incremento de la frecuencia de muestreo** Reduciendo el número de caracteres a enviar, facilitando la decodificación de la trama recibida y minimizando el proceso de cómputo de fuerzas externas.
- **Realización de una tarjeta hardware de diseño propio.** Utilizando microprocesadores con una mayor potencia de procesamiento se superarían las

limitaciones de tiempo de computación del proceso de realimentación háptica, permitiendo una mayor frecuencia de muestreo. De esta manera la sensación háptica será mas realista y se permitirá la inclusión de objetos virtuales más complejos.

- **Mejora de la mecánica de la caja entrenadora.** Actualmente la caja entrenadora cuenta con una pinza en el brazo laparoscópico izquierdo, por lo que habría que arreglar la del brazo derecho y desarrollar la funcionalidad en ambos brazos. También es urgente mejorar los arrollamientos de los motores que se encuentran en mal estado, ya que algunos de ellos han tenido que ser repuestos en el transcurso de este Trabajo de Fin de Grado.
- **Desarrollo de nuevos modelos de tejidos.** Simular de manera compleja los tejidos humanos que intervienen en una cirugía laparoscópica. Así, el aprendizaje de los futuros cirujanos laparoscópicos se puede realizar de una manera más realista y compleja.
- **Diseñar un entorno complejo de simulación.** De manera que se reproduzca en detalle la caja entrenadora en el entorno virtual, se incorpore la funcionalidad de apertura y cierre de las pinzas y la correspondiente realimentación háptica de las mismas cuando éstas entren en contacto con el tejido simulado. Añadir una interfaz gráfica que permita seleccionar al usuario algunas de las características de la membrana virtual para tener más opciones de práctica.
- **Desarrollar un modelo complejo de la ecuación del motor.** Así poder obtener un modelo más complejo, incluyendo todos los términos que influyen en el cálculo del voltaje de entrada al motor y obteniendo una realimentación háptica más realista.

Apéndice

Apéndice A

Especificaciones del Motor

RE 35 Ø35 mm, Graphite Brushes, 90 Watt

maxon DC motor

M 1:2

Stock program
 Standard program
 Special program (on request)

Order Number

		118776	118777	118778	118779	118780	118781	118782	118783	118784	118785	118786	118787	118788	118789	118790
--	--	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

Motor Data

1 Assigned power rating	W	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
2 Nominal voltage	Volt	15.0	30.0	42.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0
3 No load speed	rpm	7070	7220	7530	7270	6690	6970	4750	3810	3140	2570	2100	1620	1290	1060	856
4 Stall torque	mNm	872	949	1070	966	878	766	613	483	394	320	253	194	155	125	99.5
5 Speed / torque gradient	rpm / mNm	8.45	7.77	7.17	7.63	7.88	7.89	7.86	7.84	8.09	8.19	8.47	8.55	8.54	8.80	8.94
6 No load current	mA	245	124	93	77	69	50	45	34	27	22	17	13	10	8	7
7 Starting current	A	44.9	24.4	20.3	15.5	12.9	10.1	6.43	4.16	2.74	1.83	1.16	0.704	0.448	0.298	0.193
8 Terminal resistance	Ohm	0.334	1.23	2.07	3.09	3.72	4.76	7.46	11.5	17.5	26.2	40.5	68.2	107	161	248
9 Max. permissible speed	rpm	8200	8200	8200	8200	8200	8200	8200	8200	8200	8200	8200	8200	8200	8200	8200
10 Max. continuous current	A	4.00	2.74	2.15	1.78	1.63	1.45	1.17	0.944	0.768	0.630	0.508	0.392	0.313	0.256	0.206
11 Max. continuous torque	mNm	77.7	107	113	111	111	110	111	112	111	110	109	108	108	107	106
12 Max. power output at nominal voltage	W	152	175	206	181	150	118	75.0	48.4	31.8	21.2	13.7	8.07	5.10	3.36	2.13
13 Max. efficiency	%	81	84	86	85	85	84	83	82	80	79	77	74	72	69	66
14 Torque constant	mNm / A	19.4	38.9	62.5	62.2	68.0	75.8	95.2	119	144	175	214	276	340	418	513
15 Speed constant	rpm / V	491	246	182	154	140	126	100.0	80.8	66.4	54.6	44.7	34.6	27.8	22.9	18.8
16 Mechanical time constant	ms	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
17 Rotor inertia	gcm ²	65.5	65.5	69.6	65.0	64.5	62.7	62.8	62.8	60.7	59.9	57.8	57.2	57.2	55.5	54.5
18 Terminal inductance	mH	0.09	0.34	0.62	0.87	1.04	1.29	2.04	3.16	4.65	6.69	10.30	17.10	26.90	39.30	59.70
19 Thermal resistance housing-ambient	K / W	6.2	6.2	6.2	6.2	6.2	6.2	6.2	6.2	6.2	6.2	6.2	6.2	6.2	6.2	6.2
20 Thermal resistance rotor-housing	K / W	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
21 Thermal time constant winding	s	27	27	29	27	27	26	26	26	25	25	24	24	24	23	23

Specifications

- Axial play 0.05 - 0.15 mm
- Max. ball bearing loads (dynamic):
 - axial: 5.6 N
 - not preloaded: 2.4 N
 - radial (5 mm from flange): 26 N
 - Force for press fits (static) (static, shaft supported): 1200 N
- Radial play ball bearing 0.025 mm
- Ambient temperature range -20 ... +100°C
- Max. rotor temperature +155°C
- Number of commutator segments 13
- Weight of motor 340 g
- Values listed in the table are nominal. For applicable tolerances see page 43. For additional details please use the maxon website or contact your local maxon office.

Operating Range

Comments

- Recommended operating range
- Continuous operation: In observation of above listed thermal resistances (line 19 and 20) the maximum permissible rotor temperature will be reached during continuous operation at 25°C ambient. = Thermal limit
- Short term operation: The motor may be briefly overloaded (recurring).

118783 Motor with high resistance winding
118776 Motor with low resistance winding

Bibliografía

- 3-Dmed. T5 large. www.3-dmed.com/product/t5-large-laparoscopic-trainer. [Online; accedido 09/12/2016].
- A Park, D. W. (2002). Training and educational approaches to minimally invasive surgery: state of the art. In *Semin. Laparosc Surg.*
- Arduino (2016). Getting started with the arduino due. www.arduino.cc/en/Guide/ArduinoDue. [Online; accedido 21/12/2016].
- CHAI3D (2016). Library documentation. www.chai3d.org/download/doc/html/wrapper-overview.html.
- Cuschieri, A. (2005). Laparoscopic surgery: current status, issues and future developments. *Journal of the Royal Colleges of Surgeons of Edinburgh and Ireland*, 3(3):125–135.
- DelfHapticLab. Hapticmaster. www.delfthapticslab.nl/device/hapticmaster/. [Online; accedido 09/12/2016].
- eoSurgical. Surgtrac. www.eosurgical.com/collections/simulators-individual/products/eosim-essential. [Online; accedido 09/12/2016].
- Geomagic. 3dsystem. www.geomagic.com/files/5714/4552/8169/Haptic_Device_ES_Web.pdf. [Online; accedido 09/12/2016].
- Gómez, T. (2016). Diseño e implementación de un sistema de visualización de datos en tiempo real de un entrenador laparoscópico.
- Gomis, F. D. (2003). “la enseñanza de la cirugía laparoscópica”. *Cirugía española*, (58):608–617.
- Gregorio, P., Olien, N., Bailey, D., and Vassallo, S. (2011). Medical simulation interface apparatus and method. US Patent 8,007,282.
- J Usón, S. P. (1999). *Pautas para el aprendizaje en suturas laparoscópicas*. Consejería de Educación, Ciencia y Tecnología, Extremadura, España.
- M Nagendran, K Gurusamy et al. (2013). Entrenamiento en realidad virtual para complementar el entrenamiento estándar en cirujanos practicantes con experiencia laparoscópica previa limitada. www.cochrane.org/es/CD006575. [Online; accedido 09/12/2016].

- Mark W. Spong, Seth Hutchinson, M. V. (2006). *Robot Modeling and Control*. Malloy Inc., United States.
- Monasterio-Huelin, F. Dispositivos hápticos y cirugía robótica. www.robolabo.etsit.upm.es/asignaturas/shcr/apuntes/haptico-2.pdf. [Online; accedido 09/12/2016].
- Monasterio-Huelin, F. (2016a). Cinemática. www.robolabo.etsit.upm.es/asignaturas/rob/apuntes/Cinematica-v1.pdf. [Online; accedido 13/12/2016].
- Monasterio-Huelin, F. (2016b). Introducción a los sistemas hápticos. www.robolabo.etsit.upm.es/asignaturas/rob/apuntes/SistemasHapticos-v1.pdf. [Online; accedido 14/12/2016].
- Monasterio-Huelin, F. (2016c). Motor dc, etapa de potencia y pwm. www.robolabo.etsit.upm.es/asignaturas/seco/apuntes/aplicacion-MotorDC.pdf. [Online; accedido 11/12/2016].
- Monasterio-Huelin, F. (2016d). Sistemas de segundo orden: especificaciones de diseño del régimen transitorio. controladores p, pd y p-d. www.robolabo.etsit.upm.es/asignaturas/seco/apuntes/SistemaOrden2-P-PD.pdf. [Online; accedido 12/12/2016].
- Simulab. Labtrainer. www.simulab.com/products/laptrainer%E2%84%A2-simuvision%C2%AE-student-skill-set-package. [Online; accedido 09/12/2016].
- STMicroelectronics (2015). X-nucleo-ihm04a1. www.st.com/en/ecosystems/x-nucleo-ihm04a1.html. [Online; accedido 26/12/2016].
- Tzovaras, D. (2008). *Multimodal User Interfaces*. Springer-Verlag Berlin Heidelberg, Grecia.