



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA - CONTROL

CONTROL DE UN SISTEMA BILATERAL
TELEOPERADO CON RETARDO EN MOVIMIENTO
RESTRINGIDO

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
MAURO GILBERTO LÓPEZ RODRÍGUEZ

TUTOR PRINCIPAL
DR. MARCO ANTONIO ARTEAGA PÉREZ, FI-UNAM

CIUDAD DE MÉXICO, NOVIEMBRE, 2016

JURADO ASIGNADO:

Presidente: Dr. Maya Ortiz Paul Rolando
Secretario: Dr. Tang Xu Yu
1er Vocal: Dr. Arteaga Pérez Marco Antonio
2o. Vocal: Dr. Nuño Ortega Emmanuel
3er. Vocal: Dr. Rodríguez Angeles Alejandro

La tesis se realizó en el Posgrado de Ingeniería, UNAM.

TUTOR DE TESIS:

Dr. MARCO ANTONIO ARTEAGA PÉREZ, FI-UNAM

.

*A la Facultad de Ingeniería y a la Universidad Nacional Autónoma de México, por la
formación que me han dado.*

Es gracias a ustedes que es posible el presente trabajo.

En verdad, gracias.

A mi alma mater.

Reconocimientos

A mi familia por su apoyo incondicional durante esta etapa de mi vida.

Al Dr. Marco Antonio Arteaga Pérez por la confianza brindada para trabajar en el laboratorio de Robótica.

Por sus conocimientos y enseñanzas que me fueron útiles durante la realización de este trabajo.

A los profesores del Programa de Maestría y Doctorado en Ingeniería del campo de Control por brindar los fundamentos y encaminarme a la investigación.

A todos mis compañeros del Laboratorio de Robótica con quienes trabajé y quienes aportaron valiosos consejos y comentarios .

Finalmente quiero agradecer al proyecto “Control de robots coordinados por medio de sistemas maestro-esclavo”. Ref. IN116314. DGAPA-UNAM por el apoyo que me otorgó durante mis estudios de maestría y en el desarrollo de este trabajo.

Declaración de autenticidad

Por la presente declaro que, salvo cuando se haga referencia específica al trabajo de otras personas, el contenido de esta tesis es original y no se ha presentado total o parcialmente para su consideración para cualquier otro título o grado en esta o cualquier otra Universidad. Esta tesis es resultado de mi propio trabajo y no incluye nada que sea el resultado de algún trabajo realizado en colaboración, salvo que se indique específicamente en el texto.

Mauro Gilberto López Rodríguez. Ciudad de México, Noviembre, 2016

Resumen

El estilo de vida de hoy en día ha llevado al ser humano a expandir sus fronteras y buscar recursos en ambientes remotos que pueden tornarse complejos, peligrosos o incluso inaccesibles. La respuesta ante estas situaciones, son el uso de sistemas teleoperados bilaterales local-remoto, que toman más importancia cuando están en riesgo vidas humanas. Estos sistemas están diseñados principalmente para cumplir los siguientes objetivos: ser robustos, proveer la sensación al operador humano de presencia en el ambiente remoto y que demuestren buen desempeño en la tarea y tener transparencia. Estos objetivos dependen evidentemente de la fiabilidad que puedan tener cuando intercambian la información mutuamente el sistema local-remoto, esta información pudiese ser posición y/o velocidad, fuerza, visión etc. Sin embargo esta fiabilidad depende en gran medida del canal de comunicación empleado. Entre otros factores, la distancia entre el robot local y el remoto puede imponer un retardo sustancial en este intercambio de información. Este retardo provoca que los objetivos ahora sean ideales, ya que por la naturaleza del retardo, ya es imposible lograr un seguimiento de posición exacto, mientras que la transparencia se ve muy comprometida.

En muchos de los trabajos reportados anteriormente resuelven el problema de regulación, el problema de consenso con o sin líder, o la sincronización de un conjunto de robots induciendo trayectorias de posición periódicas, sin embargo, muchos de estos trabajos toman como un hecho el tener disponibles las velocidades. En la presente tesis se propone un esquema de teleoperación control-observador que soluciona el problema de no poder medir las velocidades. Este esquema tiene una propiedad importante en presencia del humano. Cuando un operador humano mueve el robot local en movimiento libre, el robot remoto tenderá a seguir un perfil de referencia retardado creado por la posición del robot local. Adicionalmente, en la situación más interesante cuando se está en movimiento restringido, este esquema garantiza que la persona tendrá la sensación de telepresencia.

La demostración del esquema propuesto es llevado a cabo con diferentes experimentos bajo diferentes circunstancias.

Índice general

Índice de figuras	XI
Índice de tablas	XIII
1. Introducción	1
1.1. Motivación	3
1.2. Estado del arte	3
1.2.1. Control de posición/fuerza en sistemas teleoperados	3
1.3. Planteamiento del problema	5
1.4. Contribución	6
1.5. Organización de la tesis	6
2. Marco teórico	9
2.1. Notación adoptada	9
2.2. Modelo dinámico	10
2.2.1. Modelo dinámico de un sistema de teleoperación	10
2.3. Control de posición/fuerza híbrido	10
2.3.1. Descomposición ortogonal	11
3. Esquema control observador propuesto	15
3.1. Diseño del observador	15
3.2. Diseño control local-remoto	16
3.2.1. Dinámica propuesta del humano	17
3.2.2. Resultado principal	18
4. Resultados experimentales	27
4.1. Movimiento restringido sin retardo caso ideal	28
4.2. Movimiento restringido con retardo	36
4.2.1. Retardo asimétrico	36
4.2.2. Retardo simétrico	48
4.2.3. Sin cancelación del par gravitacional	59

ÍNDICE GENERAL

5. Conclusiones	71
5.1. Trabajo futuro	72
A. Código/Manuales/Publicaciones	73
A.1. Robot <i>Geomagic Touch de 3D Systems</i>	73
A.1.1. Modelo Cinemático	74
A.1.1.1. Cinemática Directa	74
A.1.1.2. Cinemática Inversa	74
A.2. Pares de gravedad	76
A.3. Código Visual Studio C++	77
Bibliografía	97

Índice de figuras

1.1. Elementos de un sistema de teleoperación bilateral	2
2.1. Principio de Ortogonalización	12
4.1. Sistema de teleoperación con dos <i>Geomagic Touch</i>	27
4.2. Posiciones sin retardo $q_1(t)$ (—) vs $q_r(t)$ (- - - -) [°].	29
4.3. Errores de seguimiento sin presencia de retardo $e(t) = q_1(t) - q_r(t)$	30
4.4. Errores de observación sin presencia de retardo $z(t) = q_1(t) - \hat{q}_1(t)$	31
4.5. Errores de observación sin presencia de retardo $z(\dot{t}) = q_r(t) - \hat{q}_r(t)$	32
4.6. Velocidades estimadas sin retardo $\hat{q}_1(t)$ (—) vs $\hat{q}_r(t)$ (- - - -) [°/s].	33
4.7. Superficie virtual $p_1(t)$ (—) vs superficie física $p_r(t)$ (- - - -) sin presencia de retardo	34
4.8. Posiciones con retardo asimétrico $q_1(t)$ (—) vs $q_r(t - 0.7s)$ (- - - -) [°].	37
4.9. Posiciones con retardo asimétrico $q_r(t)$ (- - - -) vs $q_r(t - 0.3s)$ (—) [°]. .	38
4.10. Errores de seguimiento con presencia de retardo asimétrico $e(t) = q_1(t) - q_r(t - 0.7s)$	39
4.11. Errores de seguimiento con presencia de retardo asimétrico $e(t) = q_r(t) - q_1(t - 0.3s)$	40
4.12. Errores de observación con presencia de retardo asimétrico $z_1 = q_1(t) - \hat{q}_1(t)$	41
4.13. Errores de observación con presencia de retardo asimétrico $z_r = q_r(t) - \hat{q}_r(t)$	42
4.14. Velocidad observada $\hat{q}_1(t)$ (—) vs $\hat{q}_r(t - 0.7s)$ (- - - -) [°].	43
4.15. Velocidad observada $\hat{q}_r(t)$ (- - - -) vs $\hat{q}_1(t - 0.3s)$ (—) [°].	44
4.16. Superficie real $p_1(t)$ (—) vs superficie virtual $p_r(t)$ (—) con presencia de retardo asimétrico	45
4.17. Posiciones medidas $q_1(t)$ (—) vs posiciones estimadas $\hat{q}_r(t)$ (- - - -) [°].	46
4.18. Posiciones medidas $q_r(t)$ (—) vs posiciones estimadas $\hat{q}_1(t)$ (- - - -) [°].	47
4.19. Posiciones con retardo simétrico $q_1(t)$ (—) vs $q_r(t - 0.5s)$ (- - - -) [°]. .	48
4.20. Posiciones con retardo simétrico $q_r(t)$ (- - - -) vs $q_r(t - 0.5s)$ (—) [°]. .	49
4.21. Errores de seguimiento con presencia de retardo simétrico $e(t) = q_1(t) - q_r(t - 0.5s)$	50
4.22. Errores de seguimiento con presencia de retardo simétrico $e(t) = q_r(t) - q_1(t - 0.5s)$	51
4.23. Errores de observación con presencia de retardo asimétrico $z_1 = q_1(t) - \hat{q}_1(t)$	52

ÍNDICE DE FIGURAS

4.24. Errores de observación con presencia de retardo asimétrico $z_r = q_r(t) - \hat{q}_r(t)$	53
4.25. Velocidad observada $\dot{\hat{q}}_l(t)$ (—) vs $\dot{\hat{q}}_r(t - 0.5s)$ (- - - -) [°].	54
4.26. Velocidad observada $\dot{\hat{q}}_r(t)$ (- - - -) vs $\dot{\hat{q}}_l(t - 0.5s)$ (—) [°].	55
4.27. Superficie virtual $p_l(t)$ (—) vs superficie real $p_r(t)$ (- - - -) con presencia de retardo simétrico	56
4.28. Posiciones medidas $q_l(t)$ (—) vs posiciones estimadas $\hat{q}_r(t)$ (- - - -) [°].	57
4.29. Posiciones medidas $q_r(t)$ (—) vs posiciones estimadas $\hat{q}_l(t)$ (- - - -) [°].	58
4.30. Posiciones sin cancelación del par gravitacional y con presencia de retardo asimétrico $q_l(t)$ (—) vs $q_r(t - 0.7s)$ (- - - -) [°].	59
4.31. Posiciones sin cancelación del par gravitacional y con presencia de retardo asimétrico $q_r(t)$ (- - - -) vs $q_l(t - 0.3s)$ (—) [°].	60
4.32. Errores de seguimiento sin cancelación del par gravitacional y con presencia de retardo asimétrico $e(t) = q_l(t) - q_r(t - 0.7s)$	61
4.33. Errores de seguimiento sin cancelación del par gravitacional y con presencia de retardo asimétrico $e(t) = q_r(t) - q_l(t - 0.3s)$	62
4.34. Errores de observación para retardo asimétrico $z_l = q_l(t) - \hat{q}_l(t)$	63
4.35. Errores de observación para retardo asimétrico $z_r = q_r(t) - \hat{q}_r(t)$	64
4.36. Velocidad observada $\dot{\hat{q}}_l(t)$ (—) vs $\dot{\hat{q}}_r(t - 0.7s)$ (- - - -) [°].	65
4.37. Velocidad observada $\dot{\hat{q}}_r(t)$ (- - - -) vs $\dot{\hat{q}}_l(t - 0.3s)$ (—) [°].	66
4.38. Superficie virtual $p_l(t)$ (—) vs superficie real $p_r(t)$ (—) sin cancelación del par gravitacional y con retardo asimétrico	67
4.39. Posiciones medidas $q_l(t)$ (—) vs posiciones estimadas $\hat{q}_r(t)$ (- - - -) [°].	68
4.40. Posiciones medidas $q_r(t)$ (—) vs posiciones estimadas $\hat{q}_l(t)$ (- - - -) [°].	69
A.1. Robot <i>Geomagic Touch de 3D Systems</i>	73
A.2. Vista isométrica	75

Índice de tablas

A.1. Especificaciones técnicas del robot <i>Geomagic Touch de 3D Systems</i> . . .	76
A.2. Relación giro-cuentas de encoders <i>Geomagic Touch de 3D Systems</i> . . .	76

Introducción

Desde los inicios del ser humano pensante, éste ha desarrollado herramientas y utensilios para cubrir sus necesidades y buscar mejorar su calidad de vida. Tal vez una de estas necesidades fuese quizá, disponer de los frutos de los árboles, empleando pértigas para este fin, este pudiese ser el antecedente más antiguo de la manipulación a distancia. Desde aquel día hasta hoy, la capacidad de disponer de objetos lejanos se ha extendido incluso sin estar presente, vía teleoperación. El término “teleoperación” proviene del prefijo *tele* que proviene del griego y significa *a distancia*, naturalmente *teleoperación* indica operación a distancia.

Hoy en día los sistemas teleoperados robóticos son ideales para realizar tareas con gran repetitividad, ambientes que pueden ser peligrosos, tareas en escalas nano o macro métricas, algunos ejemplos son la industria aeroespacial, química, nuclear, submarina, médica etc.

Es evidente que la dependencia a los sistemas teleoperados robóticos cada día se va reafirmando y más aún cuando se deben realizar tareas cada vez más demandantes y complejas.

Un sistema de teleoperación consta de manera general de 5 elementos básicos:

- Operador humano.
- Manipulador local y su controlador.
- Canal de comunicación.
- Manipulador remoto y su controlador
- Ambiente.

Estos sistemas de teleoperación pueden clasificarse en unilateral y bilateral. El primero corresponde al caso en el que solo el robot maestro/local envía información de posición/velocidad y fuerza al esclavo/remoto. En el segundo caso existe comunicación entre ambos robots, lo que permite que intercambien la información necesaria, como por ejemplo posición-posición, fuerza-posición, posición-fuerza y fuerza-fuerza.

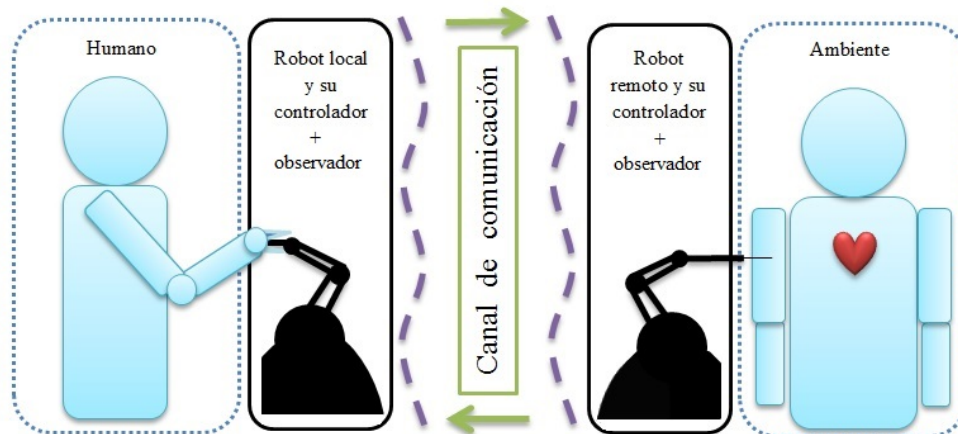


Figura 1.1: Elementos de un sistema de teleoperación bilateral

El sistema de teleoperación bilateral mostrado en la Figura 1.1 está constituido por un manipulador llamado robot local/maestro controlado por el humano, un canal de comunicación, un segundo manipulador que sigue el movimiento del robot local/maestro conocido como robot esclavo/remoto y algún ambiente. De manera general, el esquema propuesto opera de la siguiente forma: el humano aplica una fuerza al efector final del robot maestro llevándolo por la trayectoria deseada, de esta manera se genera un perfil de la posición y velocidad de su efector final que a su vez se transmite al robot remoto vía el canal de comunicación, como consecuencia del esquema de control propuesto, éste trata de seguir la trayectoria generada por el humano, con o sin la presencia de algún ambiente que interfiriera.

Este esquema se dice bilateral debido al hecho de que ambos robots se envían información. Para fines prácticos este esquema es bilateral si el humano se intercambia de lugar y sigue teniendo control sobre el robot sobrante.

Desde el punto de vista de control los objetivos principales en los sistemas de teleoperación bilaterales local-remoto son:

Estabilidad. Garantizar la estabilidad del sistema completo.

Telepresencia. Proporcionar al operador la sensación de estar en contacto con el objeto o ambiente remoto.

Transparencia. Si el operador solo siente la dinámica del objeto o entorno que está manipulando se dice que el sistema es transparente.

1.1. Motivación

El problema de mayor interés para este trabajo es la aparición de un retardo en el canal de comunicación. Su presencia puede inestabilizar al sistema si el controlador no fue diseñado considerando este efecto. Sin embargo, la caracterización del retardo mismo puede ser una tarea difícil. En muchos de los trabajos reportados se ataca el problema de dos formas, la primera es considerar al retardo constante, para el análisis de estabilidad se toma este retardo constante como su cota, la segunda y la cual se apega a un comportamiento real es considerar al retardo variante en el tiempo, sin embargo también es necesario conocer una cota.

Es evidente que este tipo de sistemas son cada vez más utilizados para realizar distintas tareas en diferentes áreas, y que aún tienen mucho por desarrollar. Estos sistemas han crecido enormemente especialmente en tareas específicas como lo pueden ser: manejo de material peligroso y radioactivo, realización de cirugías sin la presencia del médico cirujano, la exploración espacial, mantenimiento y reparación de estaciones espaciales. Muchas de estas tareas usan esquemas de control complejos y difíciles de implementar. Debido a estos y otros factores, surge la idea de diseñar un esquema que sea fácilmente implementable y que garantice la estabilidad del sistema en lazo cerrado.

1.2. Estado del arte

1.2.1. Control de posición/fuerza en sistemas teleoperados

Cuando se opera un robot a distancia, es deseable tener alguna información acerca de la interacción que existe entre el ambiente y el robot remoto y posibles efectos con presencia del retardo. La retroalimentación del ambiente remoto puede ser provista al operador humano en diferentes formas: audio, visuales o táctiles. Cuando existe fuerza de retroalimentación desde el esclavo hacia el robot local representando información de contacto con el ambiente, se provee un mayor sentido de la telepresencia. Cuando este proceso funciona también de manera inversa se dice ser un sistema controlado bilateralmente.

Los inicios de los sistemas teleoperados como tal, datan de 1898. El primer sistema considerado como teleoperado unilateral fue un pequeño barco controlado con señales de radio frecuencia creado por el inventor e ingeniero Serbio Nikola Tesla. Posteriormente en la década de los 40's se inventa el primer sistema teleoperado bilateral desarrollado por Raymond Goertz, este no era más que un sistema mecánico que permitía manipular los materiales peligrosos usados en la construcción de la bomba atómica.

Según [1] este evento fue el comienzo de la teleoperación. En la década de los 60's es cuando se observa que existía el problema para garantizar la estabilidad en la reflexión de fuerza en sistemas teleoperados, algunos análisis y soluciones han sido presentadas desde entonces.

1. INTRODUCCIÓN

En 1965 surge el primer trabajo donde trata el problema del retardo en los sistemas teleoperados [2]. Sin embargo, no se usó reflexión de fuerza por lo que el efecto del retardo no fue evidente. No fue hasta 1966, que [3] usa reflexión de fuerza y el efecto desestabilizante del retardo es evidente.

Uno de los primeros trabajos teóricos para sistemas teleoperados con retardo es [4]. En donde se demuestra que el envío de la velocidad hacia el robot esclavo y la fuerza hacia el robot maestro en tareas de teleoperación de fuerza reflejada causa inestabilidad incluso con retardos relativamente pequeños, debido a que tal transmisión de datos representa un bloque de comunicación no pasivo. Para hacer pasivo el canal de comunicación, en [5] se propone una novedosa manera de analizar al sistema de teleoperación. El análisis divide al sistema en cinco bloques: el operador humano, el robot local, el canal de comunicación, el robot remoto y el ambiente. Con uso de teoría de pasividad, de *scattering* y una analogía de elementos mecánicos/eléctricos se presenta una ley de control bilateral que garantiza la estabilidad para ciertos ambientes con ciertos retardos. En este trabajo se concluye que la inestabilidad en el sistema de teleoperación bilateral es debida al bloque de comunicación, por lo tanto la ley de control propuesta minimiza las pérdidas en la línea y de este modo se garantiza que el bloque de comunicación sea pasivo.

Uno de los trabajos que se enfocó en menor conocimiento del modelo, por lo menos para el sistema remoto es el visto en [6], donde se propone incorporar los controladores adaptables [7] y [8].

En muchos de los trabajos mencionados no se había logrado tener un buen desempeño en estado estable para la posición y el seguimiento de fuerza. Se sabía que un análisis basado en una arquitectura utilizando variables de dispersión de onda garantizaban que la red de comunicación fuese pasiva, pero no se garantizaba la sincronización en estado estable de ambos robots cuando la posición inicial de ambos robots es diferente. Una nueva arquitectura que elimina la diferencia de posición en estado estable entre los robots y que además garantiza un seguimiento en estado estable de fuerza en contacto con algún ambiente remoto es presentado en [9]. Esta arquitectura propuesta es similar a la utilizada con variables de dispersión de onda, agregando un controlador proporcional en cada robot que usa la posición retardada como señal de referencia.

En [10] se prueba matemáticamente, usando el Lema de Barbalat que un simple esquema proporcional derivativo (PD) garantiza la operación estable del sistema teleoperado. Sin embargo, este esquema supone mapeos estables \mathcal{L}_∞ de velocidad-fuerza pertenecientes al humano y al ambiente, suposiciones que resultan difíciles de verificar. El primer controlador propuesto para movimiento libre consta para ambos robots de un término proporcional al error de posición retardada, más la inyección de suficiente amortiguamiento. Esta inclusión de amortiguamiento sin retardo debe dominar a las ganancias proporcionales como se menciona en [11], de esta manera se puede garantizar que las velocidades pertenecen al espacio \mathcal{L}_2 . Para el segundo controlador PD, para el cual se considera inyección de amortiguamiento en función de las velocidades retardadas, no demuestra tener mejor desempeño que el controlador con inyección de amortiguamiento sin retardo. Para el caso de movimiento restringido se propone un controlador que con-

tiene un término proporcional al error de posición más inyección de amortiguamiento para el lado del remoto. Para el lado del local, se plantea utilizar la lectura de la fuerza del remoto con retardo más inyección de amortiguamiento, resultando tener buen desempeño.

Muchos esquemas para sistemas bilaterales han demostrado poseer estabilidad robusta ante retardos de tiempo constantes en la red de comunicación usando conceptos de pasividad, teoría de *scattering* y teoría de redes de dos puertos. Sin embargo muchos esquemas no garantizan seguimiento en posición, en [12] se propone una arquitectura adaptable que usa la retroalimentación de estados para definir una nueva salida pasiva, compuesta por la posición y velocidad. Para movimiento libre se propone un esquema de control usando la nueva salida pasiva para garantizar la coordinación de ambos robots, mientras que para movimiento restringido se garantiza acotamiento final de las trayectorias sobre la superficie en contacto pasiva y considerando al humano no pasivo. Para sistemas de teleoperación se han mencionado algunos de los trabajos más relevantes. En muchos de estos se ha supuesto que se conoce el modelo dinámico de los robots o que las velocidades están disponibles. En ocasiones la construcción y costo de los robots puede interceder con el conocimiento de los parámetros y velocidades del robot. Sin embargo, algunos trabajos que no fueron diseñados inicialmente para tareas de teleoperación pueden ser modificados para este fin.

En [13] se desarrolla la teoría para un novedoso esquema control-observador que no necesita el conocimiento del modelo y además usa únicamente las mediciones de posición, evitando el uso de tacómetros al momento de su implementación. Con este esquema se garantiza que tanto los errores de seguimiento como los de observación tenderán a cero. En [14] se prueba un esquema de control PD bien conocido y la inclusión de un observador lineal demuestra base experimentos ser robusto ante incertidumbres paramétricas del robot y garantiza acotamiento final de los errores de observación y de seguimiento.

1.3. Planteamiento del problema

La mayoría de los investigadores se han enfocado en desarrollar leyes de control para sistemas teleoperados con el objetivo de que el efector final del manipulador remoto siga trayectorias deseadas de posición y fuerza comandadas por un operador humano. Sin embargo muchos de estos trabajos reportados en la literatura consideran que la dinámica de ambos manipuladores local y remoto son conocidas, además de suponer que las velocidades están disponibles. En este trabajo se propone un esquema que evita el conocimiento del modelo dinámico del sistema con excepción del término gravitacional y además, con base en un estimador no se requieren medir directamente las velocidades de las articulaciones. El esquema control-observador diseñado será aplicado a un sistema de teleoperación local-remoto, bajo dos circunstancias:

- Dado un sistema de teleoperación bilateral local-remoto donde el movimiento del remoto está restringido por una superficie conocida, se desea que éste siga una trayectoria suave en posición comandada por el robot local.

Para el caso de movimiento restringido se considera a la superficie rígida y que el contacto es puntual. Así mismo, el movimiento del lado del local debe estar restringido por el ambiente remoto.

1.4. Contribución

Con este trabajo se unifica la idea de usar un controlador que permita llevar a cabo la sincronización de un sistema local-remoto y agregar un observador lineal para evitar el uso de tacómetros. Este enfoque es novedoso y sencillo, además resulta ser muy útil para aplicaciones prácticas.

La contribución principal de este trabajo es la presentación de un esquema control-observador que no requiere ningún conocimiento del modelo de los manipuladores, con excepción del término gravitacional; además garantiza la sincronización de posición para sistemas de teleoperación bilateral local-remoto. El esquema propuesto posee robustez ante la presencia de retardos de tiempo constantes.

1.5. Organización de la tesis

El documento está organizado de la siguiente manera: en el Capítulo 2 se presentan los conceptos básicos para el desarrollo de la teoría planteada, es decir el modelo dinámico del robot y sus propiedades, que resultan fundamentales para el diseño y análisis del esquema de control-observador. En el Capítulo 3 se presenta el esquema control-observador propuesto para resolver el problema de seguimiento de posición sin el conocimiento de velocidades y que considera la presencia de retardos en el canal de comunicación. En el Capítulo 4 se muestran los resultados experimentales del esquema control-observador diseñados bajo diferentes circunstancias propuestas en el Capítulos 3. Finalmente, el Capítulo 5 presenta las conclusiones, observaciones y el trabajo futuro.

Marco teórico

Este capítulo tiene como objetivo sentar las bases matemáticas usadas a lo largo del trabajo. Se inicia con la notación adoptada y culmina con el modelo dinámico del robot *Geomagic Touch* haciendo hincapié en algunas de sus propiedades que son útiles para el diseño y análisis del esquema control-observador presentado en el siguiente capítulo.

2.1. Notación adoptada

A continuación se presenta la notación adoptada:

- $\Re := (-\infty, \infty)$, $\Re_{>0} = (0, \infty)$, $\Re_{\geq 0} := [0, \infty)$
- El mínimo y máximo valor propio de alguna matriz \mathbf{A} son denotados como $\lambda_{\min}(\mathbf{A})$ y $\lambda_{\max}(\mathbf{A})$
- $\|\mathbf{x}\|$ se mantiene para la norma Euclidiana estándar del vector \mathbf{x} .
- Los subíndices $i = l, r$ están asignados para los robos local (*local*) y remoto (*remoto*) respectivamente.
- En algunos casos el argumento de las señales dependientes del tiempo es omitido $\mathbf{q}_i \equiv \mathbf{q}_i(t)$, $\dot{\mathbf{q}}_i \equiv \dot{\mathbf{q}}_i(t)$ etc.
- Las señales con argumento de tiempo más retardo; está dada por $(\bar{\cdot})$, por ejemplo $\bar{\mathbf{q}}_i(t) \equiv \mathbf{q}_i(t - T_i)$.
- La estimación de alguna variable está dada por $(\hat{\cdot})$, por ejemplo $\hat{\mathbf{q}}_i(t) \equiv \hat{\mathbf{q}}_i$.
- La estimación de alguna variable con retardo está dada por $(\tilde{\cdot})$, por ejemplo $\tilde{\mathbf{q}}_i \equiv \hat{\mathbf{q}}_i(t - T_i)$.

2.2. Modelo dinámico

Esta sección aborda el modelo dinámico del sistema de teleoperación usado y también algunas propiedades útiles.

2.2.1. Modelo dinámico de un sistema de teleoperación

Los robots local y remoto son modelados como manipuladores seriales, de eslabones rígidos y articulaciones rotacionales, ambos con n grados de libertad, pero no necesariamente la misma configuración cinemática. La dinámica no lineal del robot local está dada por [15]:

$$\mathbf{H}_l(\mathbf{q}_l)\ddot{\mathbf{q}}_l + \mathbf{C}_l(\mathbf{q}_l, \dot{\mathbf{q}}_l)\dot{\mathbf{q}}_l + \mathbf{D}_l\dot{\mathbf{q}}_l + \mathbf{g}_l(\mathbf{q}_l) = \boldsymbol{\tau}_l - \boldsymbol{\tau}_h, \quad (2.1)$$

mientras que la dinámica del robot remoto está dada por :

$$\mathbf{H}_r(\mathbf{q}_r)\ddot{\mathbf{q}}_r + \mathbf{C}_r(\mathbf{q}_r, \dot{\mathbf{q}}_r)\dot{\mathbf{q}}_r + \mathbf{D}_r\dot{\mathbf{q}}_r + \mathbf{g}_r(\mathbf{q}_r) = \boldsymbol{\tau}_e - \boldsymbol{\tau}_r, \quad (2.2)$$

donde $\mathbf{q}_i \in \mathfrak{R}^n$ es el vector de coordenadas generalizadas, $\mathbf{H}_i(\mathbf{q}_i) \in \mathfrak{R}^{n \times n}$ es la matriz de inercia, la cual es simétrica y positiva definida, $\mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i \in \mathfrak{R}^n$ es el vector de Coriolis y pares debido a fuerzas centrífugas, $\mathbf{D}_i \in \mathfrak{R}^{n \times n}$ es una matriz semidefinida positiva, en cuya diagonal principal están contenidos los coeficientes de fricción viscosa, $\mathbf{g}_i(\mathbf{q}_i) \in \mathfrak{R}^n$ es el vector de pares gravitacionales, $\boldsymbol{\tau}_i \in \mathfrak{R}^n$ es el vector de pares actuando sobre cada articulación, $\boldsymbol{\tau}_h \in \mathfrak{R}^n$ representa el par aplicado por el humano al robot local y $\boldsymbol{\tau}_e \in \mathfrak{R}^n$ representa los pares debido a la interacción del robot remoto con algún ambiente. El modelo posee las siguientes propiedades importantes:

Propiedad 2.2.1 *Se cumple que $\lambda_{hi} \|\mathbf{x}\|^2 \leq \mathbf{x}^T \mathbf{H}_i(\mathbf{q}_i) \mathbf{x} \leq \lambda_{Hi} \|\mathbf{x}\|^2 \forall \mathbf{q}_i, \in \mathfrak{R}^n, \mathbf{x} \in \mathfrak{R}$, y $0 < \lambda_{hi} < \lambda_{Hi} < \infty$ con $\lambda_{hi} \triangleq \min_{\forall \mathbf{q}_i \in \mathfrak{R}^n} \lambda_{\min}(\mathbf{H}_i(\mathbf{q}_i))$ y $\lambda_{Hi} \triangleq \max_{\forall \mathbf{q}_i \in \mathfrak{R}^n} \lambda_{\max}(\mathbf{H}_i(\mathbf{q}_i))$.*

Propiedad 2.2.2 *Con una apropiada definición de $\mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)$, $\dot{\mathbf{H}}_i(\mathbf{q}_i) - 2\mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)$ es antisimétrica.*

2.3. Control de posición/fuerza híbrido

En los sistemas teleoperados, en el mejor de los casos se desea tener alguna descripción sobre la interacción del robot manipulador remoto y el ambiente. Esta descripción

debe a su vez estar en términos de las restricciones naturales y artificiales. Evidentemente ambas restricciones deben estar expresadas con referencia al marco restringido, ya que cuando un manipulador está en contacto con una superficie rígida, existirá una restricción natural que físicamente no le permitiría atravesar la superficie, sino únicamente desplazarse sobre ella. En esta circunstancia se dice que el manipulador está en movimiento restringido y por lo tanto existen algunas direcciones que están sujetas a la restricción de la posición del efector final y también habrá otras a la restricción de la interacción de fuerza, por lo que no es posible imponer simultáneamente valores arbitrarios de posición y fuerza a lo largo de cada dirección.

Cuando se le agrega el retardo debido al canal de comunicación y el robot local es llevado por una persona generando de este modo una trayectoria puede o no ser compatible con las restricciones impuestas por el ambiente. Por este motivo es necesaria tener una descripción analítica de la fuerza de interacción existente entre el manipulador remoto y la superficie.

2.3.1. Descomposición ortogonal

Para resolver el problema de posición y fuerza, Raibert y Craig propusieron en 1981 el llamado control de posición-fuerza híbrida, con el cual se propone un modo de selección que permite distinguir las componentes del control de posición y de fuerza. Sin embargo, este enfoque funcionaba bien sobre superficies planas.

El principio de ortogonalización fue propuesto [16], el cual es una noción extendida del control híbrido (control de fuerza y posición) de robots manipuladores bajo restricciones geométricas. Cuando el efector final del manipulador está en contacto puntual y se mueve a lo largo de una trayectoria aplicando una fuerza, la velocidad será tangente en el punto de contacto, mientras que la fuerza aplicada será normal a la superficie. En esencia el principio de ortogonalización permite descomponer la tarea de control en dos subespacios ortogonales facilitando al mismo tiempo tanto el diseño de la ley de control y el análisis de estabilidad. Estos dos subespacios son la velocidad-posición en el espacio articular y el vector de fuerzas de contacto cuyas componentes son ejercidas en las correspondientes articulaciones.

En cuanto a la superficie con la que el manipulador remoto está en contacto, en este trabajo se asume que puede ser modelada como una restricción holonómica o integrable, es decir que pueda ser escrita por una ecuación algebraica de la siguiente forma:

$$\varphi(\mathbf{q}) = 0, \quad (2.3)$$

donde $(\varphi)(\mathbf{q}) : \mathfrak{R}^n \rightarrow \mathfrak{R}^l$, l es el número de restricciones. Para poder incluir las fuerzas de contacto (remoto-superficie) al modelo dinámico es necesario usar los multiplicadores de Lagrange [17]. El modelo dinámico bajo restricción en su movimiento está dado por;

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} + \frac{\partial \mathcal{D}}{\partial \dot{\mathbf{q}}} = \boldsymbol{\tau} + \frac{\partial \varphi}{\partial \mathbf{q}} \boldsymbol{\lambda}, \quad (2.4)$$

donde $\boldsymbol{\lambda} \in \mathfrak{R}^l$ es el vector de multiplicadores de Lagrange y \mathcal{L} es el lagrangiano del manipulador. Sustituyendo (2.3) en (2.4) se obtiene

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \mathbf{J}_\varphi^T(\mathbf{q})\boldsymbol{\lambda}, \quad (2.5)$$

donde $\mathbf{J}_\varphi^T(\mathbf{q}) = \nabla\varphi(\mathbf{q}) \in \mathfrak{R}^{n \times l}$ es el Jacobiano de la restricción y permite mapear cualquier vector sobre un plano perpendicular a la superficie en el punto de contacto. Al derivar (2.3) respecto al tiempo se obtiene

$$\dot{\varphi}(\mathbf{q}) = \mathbf{J}_\varphi(\mathbf{q})\dot{\mathbf{q}} = 0. \quad (2.6)$$

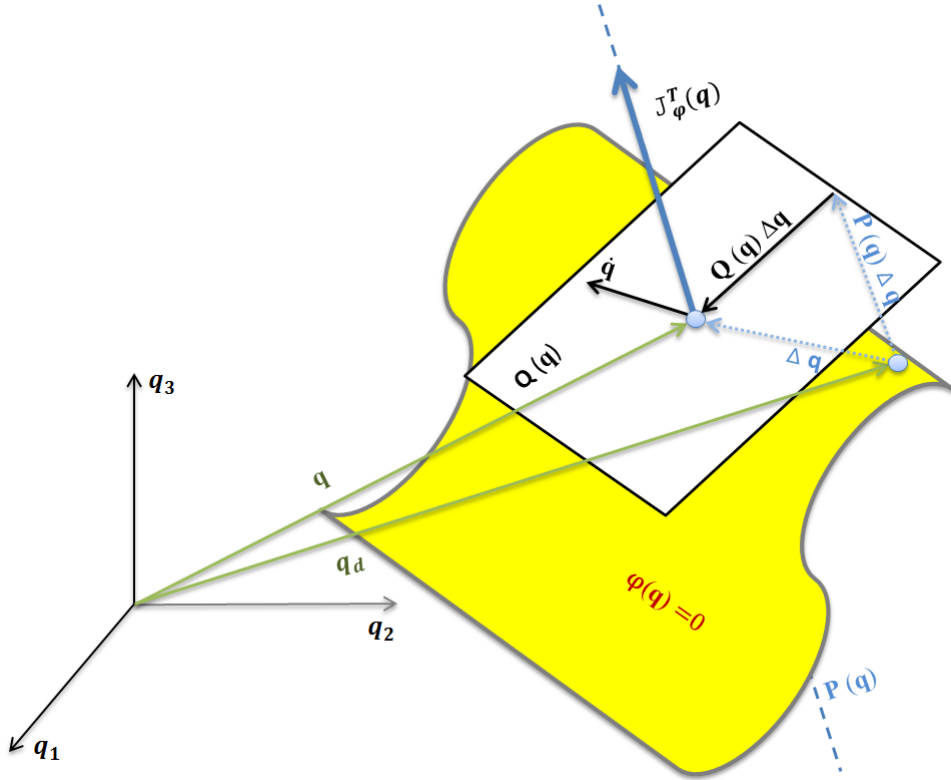


Figura 2.1: Principio de Ortogonalización

De la Figura 2.1 se observa que la velocidad $\dot{\mathbf{q}}$ está contenida en un plano tangente $\mathbf{Q}(\mathbf{q})$ sobre el punto de contacto, mientras que la fuerza es normal a este plano. La matriz de proyección $\mathbf{P}(\mathbf{q}) \in \mathfrak{R}^{n \times n}$, tal que $\mathbf{P}(\mathbf{q})\mathbf{v} = \mathbf{v}$, $\forall \mathbf{v} \in \mathbb{R}\{\mathbf{J}_\varphi^T\}$, (donde $\mathbb{R}\{\mathbf{J}_\varphi^T\}$ es el espacio rango de la matriz $\mathbf{J}_\varphi^T(\mathbf{q})$) y está dada por

$$\mathbf{P}(\mathbf{q}) = \mathbf{J}_\varphi^+(\mathbf{q})\mathbf{J}_\varphi(\mathbf{q}), \quad (2.7)$$

donde $\mathbf{J}_\varphi^+(\mathbf{q})$ es la matriz Pseudoinversa de Moore Penrose dada por

$$\mathbf{J}_\varphi^+(\mathbf{q}) \triangleq \mathbf{J}_\varphi^T(\mathbf{q})[\mathbf{J}_\varphi(\mathbf{q})\mathbf{J}_\varphi^T(\mathbf{q})]^{-1}, \quad (2.8)$$

De manera similar se tiene a la matriz de proyección $\mathbf{Q}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ definida tal que $\mathbf{Q}(\mathbf{q})\mathbf{y} = \mathbf{y} \forall \mathbf{y} \in \mathcal{N}\{\mathbf{J}_\varphi\}$, (donde $\mathcal{N}\{\mathbf{J}_\varphi\}$ es el espacio nulo de la matriz $\mathbf{J}_\varphi(\mathbf{q})$). La matriz de proyección $\mathbf{Q}(\mathbf{q})$ mapea un vector del espacio articular al plano tangente a la superficie en el punto de contacto y está dada por:

$$\mathbf{Q}(\mathbf{q}) = \mathbf{I}_{n \times n} - \mathbf{P}(\mathbf{q}). \quad (2.9)$$

El Jacobiano de la restricción $\mathbf{J}_\varphi(\mathbf{q})$ está contenido en el plano generado por $\mathbf{P}(\mathbf{q})$, por lo que se debe satisfacer $\mathbf{J}_\varphi(\mathbf{q})\mathbf{P}(\mathbf{q}) = \mathbf{J}_\varphi(\mathbf{q})$ y $\mathbf{P}(\mathbf{q})\mathbf{J}_\varphi^T(\mathbf{q}) = \mathbf{J}_\varphi^T(\mathbf{q})$; además $\mathbf{Q}(\mathbf{q})\mathbf{J}_\varphi^T(\mathbf{q}) = 0$ y $\mathbf{Q}(\mathbf{q})\mathbf{J}_\varphi(\mathbf{q}) = 0$.

Usando las definiciones anteriores, el vector de velocidad $\dot{\mathbf{q}}$ puede escribirse como:

$$\dot{\mathbf{q}} = \mathbf{Q}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{P}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{Q}(\mathbf{q})\dot{\mathbf{q}}. \quad (2.10)$$

Debido a que la velocidad está contenida en el plano tangente, el vector de velocidad $\dot{\mathbf{q}}$ no tiene componente en $\mathbf{P}(\mathbf{q})$.

Esquema control observador propuesto

En este capítulo se aborda el diseño del esquema controlador observador para el problema de seguimiento de posición de manipuladores rígidos bajo restricciones de movimiento. El esquema diseñado evita tanto el conocimiento del modelo dinámico del manipulador, como la medición explícita de velocidad. Estas propiedades hacen que este esquema sea sencillo a la hora de implementarlo en un sistema de teleoperación. En los sistemas de control bilaterales local-remoto, los principales objetivos son: robustez, sensación de presencia o telepresencia, transparencia y buen desempeño. Sin embargo, cuando existe un retardo en el canal de comunicación o existen pérdidas de paquetes, algunos de los objetivos antes mencionados son sacrificados y optimizados, buscando la estabilidad del sistema en lazo cerrado. El esquema que se presenta a continuación toma en cuenta retardos en el canal de comunicación, teniendo como principal propiedad que en movimiento restringido la persona tendrá la sensación de telepresencia, pero se sacrificará en gran medida la transparencia.

3.1. Diseño del observador

Los sistemas bilaterales local-remoto permiten a las personas desempeñar tareas en un ambiente remoto o inaccesible, mientras se provee una retroalimentación kinestésica hacia el operador humano. En este tipo de aplicaciones es más útil diseñar un estimador de estado que se sobreponga al problema de señales contaminadas con ruido tras usar un tacómetro, sin mencionar que agrega costos, peso, etc.

Por lo tanto, supóngase que las mediciones de velocidades no están disponibles, por lo que el objetivo es diseñar una ley de control para seguimiento de posición que no use las mediciones de velocidad explícitamente. Adicionalmente además para su diseño se debe tomar en cuenta que existen retardos de tiempo constante impuestos por el canal de comunicación dados por $T_l > 0$ y $T_r > 0$. Por simplicidad se define

$$\bar{\mathbf{q}}_i = \bar{\mathbf{q}}_i(t) \triangleq \mathbf{q}_i(t - T_i). \quad (3.1)$$

Sí $(\hat{\cdot})$ es el valor estimado de (\cdot) , entonces el error de observación está dado por

$$\mathbf{z}_i \triangleq \mathbf{q}_i - \hat{\mathbf{q}}_i. \quad (3.2)$$

Basado en [13], se proponen los siguientes observadores

$$\dot{\hat{\mathbf{q}}}_i = \dot{\hat{\mathbf{q}}}_{oi} + \Lambda_{zi} \mathbf{z}_i + \mathbf{K}_{di} \mathbf{z}_i \quad (3.3)$$

$$\dot{\hat{\mathbf{q}}}_{oi} = \mathbf{K}_{di} \Lambda_{zi} \int_0^t \mathbf{z}_i(\vartheta) d\vartheta, \quad (3.4)$$

donde $\Lambda_{zi}, \mathbf{K}_{di} \in \mathbb{R}^{n \times n}$ son matrices diagonales positivas.

3.2. Diseño control local-remoto

El esquema de control a diseñar debe usar las variables estimadas obtenidas con el observador para garantizar el seguimiento de posición. Basado en [18], se define

$$\dot{\hat{\mathbf{q}}}_{oi} = \dot{\hat{\mathbf{q}}}_i - \Lambda_{zi} \mathbf{z}_i \quad (3.5)$$

$$\boldsymbol{\sigma}_i = \int_0^t (\mathbf{K}_{\beta i} \mathbf{s}_i + \text{sign}(\mathbf{s}_i)) d\vartheta \quad \boldsymbol{\sigma}_i(0) = \mathbf{0}, \quad (3.6)$$

donde $\mathbf{K}_{\beta i} \in \mathbb{R}^{n \times n}$ es una matriz diagonal definida positiva y $\text{sign}(\mathbf{s}_i) = [\text{sign}(s_{i1}), \dots, \text{sign}(s_{in})]^T$ con elementos s_{ij} de \mathbf{s}_i para $j = 1, \dots, n$, donde

$$\mathbf{s}_r = \dot{\hat{\mathbf{q}}}_r - \bar{\dot{\mathbf{q}}}_1 + \Lambda_{xr}(\hat{\mathbf{q}}_r - \bar{\mathbf{q}}_1) \triangleq \Delta \dot{\mathbf{q}}_r + \Lambda_{xr} \Delta \mathbf{q}_r \quad (3.7)$$

$$\mathbf{s}_l = \dot{\hat{\mathbf{q}}}_l - \bar{\dot{\mathbf{q}}}_r + \Lambda_{xl}(\hat{\mathbf{q}}_l - \bar{\mathbf{q}}_r) \triangleq \Delta \dot{\mathbf{q}}_l + \Lambda_{xl} \Delta \mathbf{q}_l, \quad (3.8)$$

con

$$\bar{\mathbf{q}}_i = \hat{\mathbf{q}}_i(t - T_i) \quad \text{y} \quad \bar{\dot{\mathbf{q}}}_i = \dot{\hat{\mathbf{q}}}_i(t - T_i). \quad (3.9)$$

Considere ahora las siguientes variables

$$\dot{\hat{\mathbf{q}}}_{rr} \triangleq \bar{\dot{\mathbf{q}}}_1 - \Lambda_{xr}(\hat{\mathbf{q}}_r - \bar{\mathbf{q}}_1) - \mathbf{K}_{\gamma r} \boldsymbol{\sigma}_r \quad (3.10)$$

$$\dot{\hat{\mathbf{q}}}_{rl} \triangleq \bar{\dot{\mathbf{q}}}_r - \Lambda_{xl}(\hat{\mathbf{q}}_l - \bar{\mathbf{q}}_r) - \mathbf{K}_{\gamma l} \boldsymbol{\sigma}_l \quad (3.11)$$

y

$$\mathbf{s}_{oi} \triangleq \dot{\hat{\mathbf{q}}}_{oi} - \dot{\hat{\mathbf{q}}}_{ri}, \quad (3.12)$$

donde $\mathbf{K}_{\gamma_i} \in \mathbb{R}^{n \times n}$ son matrices diagonales definidas positivas. Basado en todas las definiciones previas, la ley de control para el robot remoto está dada por

$$\boldsymbol{\tau}_r = \mathbf{K}_{pr} \mathbf{s}_{or} - \mathbf{g}_r(\mathbf{q}_r). \quad (3.13)$$

Para el manipulador local es propuesto

$$\boldsymbol{\tau}_l = -\mathbf{K}_{al} \dot{\mathbf{q}}_l + \mathbf{g}_l(\mathbf{q}_l) - \mathbf{K}_{pl} \mathbf{s}_{ol}, \quad (3.14)$$

donde $\mathbf{K}_{al}, \mathbf{K}_{pr}, \mathbf{K}_{pl} \in \mathbb{R}^{n \times n}$ son matrices diagonales positivas.

3.2.1. Dinámica propuesta del humano

Suposición 3.2.1 *El par inducido por el humano está descrito por un modelo tipo PD de la forma*

$$\boldsymbol{\tau}_h = \mathbf{K}_{ph}(\mathbf{q}_l - \mathbf{q}_{hd}) + \mathbf{K}_{vh}(\dot{\mathbf{q}}_l - \dot{\mathbf{q}}_{hd}), \quad (3.15)$$

donde $\mathbf{K}_{ph}, \mathbf{K}_{vh} \in \mathbb{R}^{n \times n}$ son matrices positivas definidas y $\mathbf{q}_{hd} \in \mathbb{R}^n$ representa la trayectoria deseada que la persona quiere seguir. \triangle

Observación 3.2.1 *La Suposición 3.2.1 es una combinación del comportamiento dinámico del humano propuesto en [15] y [19], donde en el primer trabajo se asume que la persona es un sistema pasivo, mientras que en el segundo se asume que la persona se comporta como un PID. Nótese, sin embargo, que debido a la inclusión de $\dot{\mathbf{q}}_{hd}$ en (3.15), $\boldsymbol{\tau}_h$ no representa un comportamiento pasivo del humano.* \triangle

3.2.2. Resultado principal

Se establece el resultado principal de este trabajo.

Proposición 3.2.1 *Considérese el sistema bilateral con retardo (2.1)–(2.2) en lazo cerrado con los observadores (3.3)–(3.4) y las leyes de control (3.13)–(3.14), y supóngase que*

- a) *La fuerza de entrada del humano está dada por la Suposición 3.2.1 y está acotada.*
- b) *El operador humano mueve las articulaciones del robot local lentamente.*
- c) *La fuerza externa τ_e es debida al contacto con una superficie rígida y está acotada.*
- d) *τ_h no es lo suficientemente grande para sobrepasar el par de entrada τ_1 en (3.14) cuando el robot local tiende a moverse en la dirección de la restricción impuesta por el ambiente al robot remoto.*

Entonces, un conjunto de ganancias pueden ser siempre encontrado tal que

- i. *Los errores de observación sean arbitrariamente pequeños, i. e. $z_1 \approx \mathbf{0}$, $\dot{z}_1 \approx \mathbf{0}$, $z_r \approx \mathbf{0}$, y $\dot{z}_r \approx \mathbf{0}$.*
- ii. *Todos los errores de seguimiento permanezcan acotados.*
- iii. *Las trayectorias del robot remoto satisfagan $\mathbf{q}_r(t) \approx \mathbf{q}_1(t - T_1)$ y $\dot{\mathbf{q}}_r(t) \approx \dot{\mathbf{q}}_1(t - T_1)$.*
- iv. *El movimiento del robot local tienda a ser posible únicamente en la dirección permitida por la restricción real sobre el lado del robot remoto, i. e. el operador humano tendrá la sensación de telepresencia, pero no de transparencia.*

△

Observación 3.2.2 *Nótese que los observadores dados en este trabajo son más simples que el introducido en [13]. Como consecuencia, ya no puede ser garantizado que los errores de observación tenderán a cero, sino únicamente pueden ser hechos arbitrariamente pequeños. Desde un punto de vista práctico, la simplificación vale la pena.*

△

Observación 3.2.3 *La Condición a) en la Proposición 3.2.1 es necesaria únicamente para llevar a cabo el análisis de estabilidad, pero desde un punto de vista práctico no se requiere para la implementación del esquema control-observador. Ciertamente, el acotamiento de la fuerza del humano puede darse por sentado.*

△

Observación 3.2.4 *Nótese que el esquema de control no emplea mediciones de fuerza por lo que no puede ser garantizado que la fuerza que el operador humano está sintiendo cuando trata de mover el manipulador local en la dirección restringida es proporcional o igual a la aplicada por el robot remoto sobre el entorno real. Por otra parte, la Condición d) en la Proposición 3.2.1 significa que la persona no violará la restricción impuesta cuando se empuja demasiado fuerte.*

△

Para demostrar la Proposición 3.2.1, considérese el siguiente teorema auxiliar.

Teorema 3.2.1 [20] *Sea $\mathcal{D} \subset \mathbb{R}^n$ un dominio que contenga el origen y $V : [0, \infty) \times \mathcal{D} \rightarrow \mathbb{R}$ una función continuamente diferenciable tal que*

$$\alpha_1(\|\mathbf{x}\|) \leq V(t, \mathbf{x}) \leq \alpha_2(\|\mathbf{x}\|) \quad (3.16)$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(t, \mathbf{x}) \leq -W_3(\mathbf{x}), \quad \forall \|\mathbf{x}\| \geq \mu > 0, \quad (3.17)$$

$\forall t \geq 0$ y $\forall \mathbf{x} \in \mathcal{D}$, donde α_1 y α_2 son funciones clase \mathcal{K} , $W_3(\mathbf{x})$ es una función continua positiva definida y $\mathbf{f} : [0, \infty) \times \mathcal{D} \rightarrow \mathbb{R}^n$ es continua por intervalos en t y localmente

3. ESQUEMA CONTROL OBSERVADOR PROPUESTO

Lipschitz en \mathbf{x} sobre $[0, \infty) \times \mathcal{D}$. Sea $r > 0$ tal que $\mathcal{B}_r = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| \leq r\} \subset \mathcal{D}$ y supóngase que

$$\mu < \alpha_2^{-1}(\alpha_1(r)). \quad (3.18)$$

Entonces, existe una función β de la clase \mathcal{KL} y para cada estado inicial $\mathbf{x}(t_0)$ que satisface

$$\|\mathbf{x}(t_0)\| \leq \alpha_2^{-1}(\alpha_1(r)), \quad (3.19)$$

existe $T \geq 0$ (dependiente de $\mathbf{x}(t_0)$ y μ) tal que la solución de $\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x})$ satisface

$$\|\mathbf{x}\| \leq \beta(\|\mathbf{x}(t_0)\|, t - t_0), \quad \forall t_0 \leq t \leq t_0 + T \quad (3.20)$$

$$\|\mathbf{x}\| \leq \alpha_1^{-1}(\alpha_2(\mu)), \quad \forall t \geq t_0 + T. \quad (3.21)$$

Además, si $\mathcal{D} = \mathbb{R}^n$ y α_1 pertenece a la clase \mathcal{K}_∞ , entonces (3.20)–(3.21) son válidas para algún estado inicial $\mathbf{x}(t_0)$, sin ninguna restricción sobre cuan grande pueda ser μ .

△

Antes que todo, defina las siguientes variables.

$$\mathbf{s}_{qi} \triangleq \dot{\mathbf{q}}_i - \dot{\mathbf{q}}_{ri} \quad (3.22)$$

$$\mathbf{r}_i \triangleq \dot{\mathbf{q}}_i - \dot{\mathbf{q}}_{oi} = \dot{\mathbf{z}}_i + \mathbf{\Lambda}_{zi} \mathbf{z}_i. \quad (3.23)$$

Ahora, considérese el sistema (2.1) en lazo cerrado con la ley de control (3.14)

$$\mathbf{H}_1(\mathbf{q}_1) \ddot{\mathbf{q}}_1 + \mathbf{C}_1(\mathbf{q}_1, \dot{\mathbf{q}}_1) \dot{\mathbf{q}}_1 + \mathbf{D}_1 \dot{\mathbf{q}}_1 = -\mathbf{K}_{al} \dot{\mathbf{q}}_1 - \mathbf{K}_{pl} \mathbf{s}_{ol} - \boldsymbol{\tau}_h. \quad (3.24)$$

Después sustituyendo (3.22) y (3.23) en (3.12) se obtiene $\mathbf{s}_{ol} = \dot{\mathbf{q}}_{ol} - \dot{\mathbf{q}}_{rl} = \mathbf{s}_{ql} - \mathbf{r}_1$ por lo que (3.24) se convierte en

$$\mathbf{H}_1(\mathbf{q}_1) \dot{\mathbf{s}}_{ql} + \mathbf{C}_1(\mathbf{q}_1, \dot{\mathbf{q}}_1) \mathbf{s}_{ql} + \mathbf{K}_{vl} \mathbf{s}_{ql} = \mathbf{K}_{al} \dot{\mathbf{z}}_1 + \mathbf{K}_{pl} \mathbf{r}_1 + \mathbf{y}_{al} - \boldsymbol{\tau}_h, \quad (3.25)$$

donde

$$\mathbf{K}_{vl} \triangleq \mathbf{D}_1 + \mathbf{K}_{al} + \mathbf{K}_{pl} \quad (3.26)$$

$$\mathbf{y}_{al} \triangleq -(\mathbf{H}_1(\mathbf{q}_1) \ddot{\mathbf{q}}_{rl} + \mathbf{C}_1(\mathbf{q}_1, \dot{\mathbf{q}}_1) \dot{\mathbf{q}}_{rl} + \mathbf{D}_1 \dot{\mathbf{q}}_{rl} + \mathbf{K}_{al} \dot{\mathbf{q}}_{rl}). \quad (3.27)$$

De manera similar para el robot remoto, tomando en cuenta (2.2), (3.13), (3.22) y (3.23) se obtiene

$$\mathbf{H}_r(\mathbf{q}_r) \dot{\mathbf{s}}_{qr} + \mathbf{C}_r(\mathbf{q}_r, \dot{\mathbf{q}}_r) \mathbf{s}_{qr} + \mathbf{K}_{vr} \mathbf{s}_{qr} = \mathbf{K}_{pr} \mathbf{r}_r + \mathbf{y}_{ar} + \boldsymbol{\tau}_e, \quad (3.28)$$

donde

$$\mathbf{K}_{vr} = \mathbf{D}_r + \mathbf{K}_{pr} \quad (3.29)$$

$$\mathbf{y}_{ar} = -(\mathbf{H}_r(\mathbf{q}_r)\ddot{\mathbf{q}}_{rr} + \mathbf{C}_r(\mathbf{q}_r, \dot{\mathbf{q}}_r)\dot{\mathbf{q}}_{rr} + \mathbf{D}_r\dot{\mathbf{q}}_{rr}). \quad (3.30)$$

En cuanto a los observadores, de (3.3)–(3.4) se obtiene para el robot manipulador local

$$\ddot{\mathbf{q}}_1 - \ddot{\hat{\mathbf{q}}}_1 + \mathbf{\Lambda}_{z1}\dot{\mathbf{z}}_1 + \mathbf{K}_{d1}\dot{\mathbf{z}}_1 + \mathbf{K}_{d1}\mathbf{\Lambda}_{z1}\mathbf{z}_1 = \ddot{\mathbf{q}}_1, \quad (3.31)$$

que de acuerdo a (3.23) se llega a la siguiente dinámica del observador en lazo cerrado

$$\dot{\mathbf{r}}_1 + \mathbf{K}_{d1}\mathbf{r}_1 = \ddot{\mathbf{q}}_1. \quad (3.32)$$

El mismo análisis puede ser llevado a cabo para el robot remoto para conseguir

$$\dot{\mathbf{r}}_r + \mathbf{K}_{dr}\mathbf{r}_r = \ddot{\mathbf{q}}_r. \quad (3.33)$$

La Proposición 3.2.1 se demuestra en 3 pasos.

1. Supóngase que la dinámica del humano puede ser descrita por (3.15), y defínase el error de seguimiento del humano como $\Delta\mathbf{q}_h \triangleq \mathbf{q}_1 - \mathbf{q}_{hd}$. Además, defina antes una variable de deslizamiento dada por $\mathbf{s}_h \triangleq \Delta\dot{\mathbf{q}}_h + \mathbf{\Lambda}_h\Delta\mathbf{q}_h$, donde $\mathbf{\Lambda}_h \triangleq \mathbf{K}_{vh}^{-1}\mathbf{K}_{ph}$. Tomando en cuenta estas definiciones, (3.15) se puede reescribir simplemente como

$$\boldsymbol{\tau}_h = \mathbf{K}_{vh}\mathbf{s}_h. \quad (3.34)$$

Uno puede considerar a \mathbf{s}_h como un estado más en el sistema en lazo cerrado, cuya dinámica puede ser considerada y dada por

$$\mathbf{H}_1(\mathbf{q}_1)\dot{\mathbf{s}}_h + \mathbf{C}_1(\mathbf{q}_1, \dot{\mathbf{q}}_1)\mathbf{s}_h + \mathbf{K}_{Dvh}\mathbf{s}_h = \boldsymbol{\tau}_1 + \mathbf{y}_{ah}, \quad (3.35)$$

con $\mathbf{K}_{Dvh} = \mathbf{K}_{vh} + \mathbf{D}_1$

$$\mathbf{y}_{ah} = -\mathbf{H}_1(\mathbf{q}_1)\ddot{\mathbf{q}}_{rh} - \mathbf{C}_1(\mathbf{q}_1, \dot{\mathbf{q}}_1)\dot{\mathbf{q}}_{rh} - \mathbf{D}_1\dot{\mathbf{q}}_{rh} - \mathbf{g}_1(\mathbf{q}_1), \quad (3.36)$$

y $\dot{\mathbf{q}}_{rh} = \dot{\mathbf{q}}_{hd} - \mathbf{\Lambda}_h\Delta\mathbf{q}_h$. Además nótese que $\mathbf{s}_h = \dot{\mathbf{q}}_1 - \dot{\mathbf{q}}_{rh}$. El estado en lazo cerrado está dado por

$$\mathbf{x} \triangleq \begin{bmatrix} \mathbf{s}_{ql} \\ \mathbf{s}_{qr} \\ \mathbf{s}_h \\ \mathbf{r}_1 \\ \mathbf{r}_r \end{bmatrix}. \quad (3.37)$$

3. ESQUEMA CONTROL OBSERVADOR PROPUESTO

Considérese la siguiente región

$$\mathcal{D}_h \triangleq \{ \mathbf{x} \in \mathbb{R}^{5n} \mid \|\mathbf{x}\| \leq x_{\max} \}, \quad (3.38)$$

donde $x_{\max} > 0$. Es posible mostrar que cada variable de interés está acotada si $\mathbf{x} \in \mathcal{D}_h$, siempre y cuando se suponga que las posiciones \mathbf{q}_l y \mathbf{q}_r , así como sus derivadas, están acotadas. Este resultado se basa en el hecho de que \mathbf{q}_{hd} se asume estar acotada.

2. Para mostrar el acotamiento final de las trayectorias $\mathbf{x} \in \mathcal{D} \forall t$, considérese

$$V_h(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{N} \mathbf{x}, \quad (3.39)$$

con $\mathbf{N} \triangleq$ diagonal a bloques $\{ \mathbf{H}_1(\mathbf{q}_l), \mathbf{H}_r(\mathbf{q}_r), \mathbf{H}_1(\mathbf{q}_l), \mathbf{I}, \mathbf{I} \}$. Tomando en cuenta la Propiedad 2.2.1, se satisface

$$\lambda_{h1} \|\mathbf{x}\|^2 \leq V_h(\mathbf{x}) \leq \lambda_{h2} \|\mathbf{x}\|^2, \quad (3.40)$$

con $\lambda_{h1} \triangleq \frac{1}{2} \min_{\mathbf{x} \in \mathbb{R}^{5n}} \lambda_{\min}(\mathbf{N})$ y $\lambda_{h2} \triangleq \frac{1}{2} \max_{\mathbf{x} \in \mathbb{R}^{5n}} \lambda_{\max}(\mathbf{N})$. Hay que recalcar que $\|\mathbf{s}_h\| \leq x_{\max}$ y $\|\boldsymbol{\tau}_e\| \leq e_{\max}$ para algún $0 < e_{\max} < \infty$. Entonces, la derivada de $V_h(\mathbf{x})$ a lo largo de las trayectorias (3.25), (3.28), (3.32), (3.33) y (3.35) es

$$\begin{aligned} \dot{V}_h(\mathbf{x}) &= \mathbf{s}_{ql}^T \mathbf{H}_1 \dot{\mathbf{s}}_{ql} + \mathbf{s}_{qr}^T \mathbf{H}_r \dot{\mathbf{s}}_{qr} + \mathbf{s}_h^T \mathbf{H}_1 \dot{\mathbf{s}}_h + \mathbf{r}_l^T \dot{\mathbf{r}}_l + \mathbf{r}_r^T \dot{\mathbf{r}}_r \\ &\quad + \frac{1}{2} [\mathbf{s}_{ql}^T \dot{\mathbf{H}}_1 \mathbf{s}_{ql} + \mathbf{s}_{qr}^T \dot{\mathbf{H}}_r \mathbf{s}_{qr} + \mathbf{s}_h^T \dot{\mathbf{H}}_1 \mathbf{s}_h]. \end{aligned}$$

Desarrollando, se obtiene:

$$\begin{aligned} \dot{V}_h(\mathbf{x}) &= \mathbf{s}_{ql}^T [\mathbf{K}_{al} \dot{\mathbf{z}}_l + \mathbf{K}_{pl} \mathbf{r}_l + \mathbf{y}_{al} - [\mathbf{K}_{vh} \mathbf{s}_h + \mathbf{C}_l \mathbf{s}_{ql} + \mathbf{K}_{vl} \mathbf{s}_{ql}]] + \frac{1}{2} \mathbf{s}_{ql}^T \dot{\mathbf{H}}_1 \mathbf{s}_{ql} \\ &\quad + \mathbf{s}_{qr}^T [\mathbf{K}_{pr} \mathbf{r}_r + \mathbf{y}_{ar} + \boldsymbol{\tau}_e - [\mathbf{C}_r \mathbf{s}_{qr} + \mathbf{K}_{vr} \mathbf{s}_{qr}]] + \frac{1}{2} \mathbf{s}_{qr}^T \dot{\mathbf{H}}_r \mathbf{s}_{qr} \\ &\quad + \mathbf{s}_h^T [\boldsymbol{\tau}_l + \mathbf{y}_{ah} - [\mathbf{C}_l \mathbf{s}_h + \mathbf{K}_{Dvh} \mathbf{s}_h]] + \mathbf{r}_l^T \dot{\mathbf{r}}_l + \mathbf{r}_r^T \dot{\mathbf{r}}_r + \frac{1}{2} \mathbf{s}_h^T \dot{\mathbf{H}}_1 \mathbf{s}_h, \end{aligned}$$

donde $\mathbf{K}_{Dvh} = \mathbf{K}_{vh} + \mathbf{D}_l$. Empleando la Propiedad 2.2.2, se obtiene

$$\begin{aligned} \dot{V}_h(\mathbf{x}) &= -\mathbf{s}_{ql}^T \mathbf{K}_{vl} \mathbf{s}_{ql} + \mathbf{s}_{ql}^T [\mathbf{K}_{al} \dot{\mathbf{z}}_l + \mathbf{K}_{pl} \mathbf{r}_l] - \mathbf{r}_l^T \mathbf{K}_{dl} \mathbf{r}_l - \mathbf{s}_{qr}^T \mathbf{K}_{vr} \mathbf{s}_{qr} + \mathbf{s}_{qr} \mathbf{K}_{pr} \mathbf{r}_r \\ &\quad - \mathbf{r}_r \mathbf{K}_{dr} \mathbf{r}_r - \mathbf{s}_h \mathbf{K}_{Dvh} \mathbf{s}_h + \mathbf{s}_{ql}^T [\mathbf{y}_{al} - \mathbf{K}_{vh} \mathbf{s}_h] + \mathbf{s}_{qr} [\mathbf{y}_{ar} + \boldsymbol{\tau}_e] + \mathbf{s}_h \boldsymbol{\tau}_l \\ &\quad + \mathbf{s}_h \mathbf{y}_{ah} + \mathbf{r}_l^T \ddot{\mathbf{q}}_l + \mathbf{r}_r \ddot{\mathbf{q}}_r, \end{aligned} \quad (3.41)$$

Tomando en cuenta que se analiza el caso exclusivamente cuando $\mathbf{x} \in \mathcal{D}$, de acuerdo con el Paso 1. de la demostración deben existir constantes positivas tales que

$$\beta_l \triangleq \lambda_{\max}(\mathbf{K}_{al}) \left(\frac{\lambda_{\max}(\mathbf{\Lambda}_{zi})}{\lambda_{\min}(\mathbf{\Lambda}_{zi})} + 1 \right) \|\mathbf{r}_i\| + \lambda_{\max}(\mathbf{K}_{pl}) \quad (3.42)$$

$$\beta_h \triangleq \alpha_l + \alpha_r + \alpha_h + a_l + a_r \quad (3.43)$$

$$\alpha_l \triangleq \max_{\forall \mathbf{x} \in \mathcal{D}_h} \|\mathbf{y}_{al}\| + \lambda_{\max}(\mathbf{K}_{vh}) x_{\max} \quad (3.44)$$

$$\alpha_r \triangleq \max_{\forall \mathbf{x} \in \mathcal{D}_h} \|\mathbf{y}_{ar}\| + e_{\max} \quad (3.45)$$

$$\alpha_h \triangleq \max_{\forall \mathbf{x} \in \mathcal{D}_h} \|\boldsymbol{\tau}_l + \mathbf{y}_{ah}\| \quad (3.46)$$

$$a_l \triangleq \max_{\forall \mathbf{x} \in \mathcal{D}_h} \|\ddot{\mathbf{q}}_l\| \quad (3.47)$$

$$a_r \triangleq \max_{\forall \mathbf{x} \in \mathcal{D}_h} \|\ddot{\mathbf{q}}_r\|. \quad (3.48)$$

El valor β_l se obtiene a partir de la definición (3.23) y empleando las siguientes desigualdades

$$\|\mathbf{z}_i\| \leq \frac{1}{\lambda_{\min}(\mathbf{\Lambda}_{zi})} \|\mathbf{r}_i\| \quad (3.49)$$

y

$$\|\dot{\mathbf{z}}_i\| \leq \left(\frac{\lambda_{\max}(\mathbf{\Lambda}_{zi})}{\lambda_{\min}(\mathbf{\Lambda}_{zi})} + 1 \right) \|\mathbf{r}_i\|. \quad (3.50)$$

Tomando en cuenta (3.42)–(3.50), es posible reescribir $\dot{V}_h(\mathbf{x})$ en (3.41) como

$$\begin{aligned} \dot{V}_h(\mathbf{x}) \leq & -\lambda_{\min}(\mathbf{K}_{vl}) \|\mathbf{s}_{ql}\|^2 + \beta_l \|\mathbf{s}_{ql}\| \|\mathbf{r}_l\| - \lambda_{\min}(\mathbf{K}_{dl}) \|\mathbf{r}_l\|^2 \\ & -\lambda_{\min}(\mathbf{K}_{vr}) \|\mathbf{s}_{qr}\|^2 + \lambda_{\max}(\mathbf{K}_{pr}) \|\mathbf{s}_{qr}\| \|\mathbf{r}_r\| - \lambda_{\min}(\mathbf{K}_{dr}) \|\mathbf{r}_r\|^2 \\ & -\lambda_{\min}(\mathbf{K}_{vh}) \|\mathbf{s}_h\|^2 \\ & + \beta_h \|\mathbf{x}\|. \end{aligned} \quad (3.51)$$

Supóngase que las ganancias son escogidas de tal manera que se satisface

$$\lambda_{\min}(\mathbf{K}_{vl}) \geq 1 + 2\delta \quad (3.52)$$

$$\lambda_{\min}(\mathbf{K}_{vr}) \geq 1 + 2\delta \quad (3.53)$$

$$\lambda_{\min}(\mathbf{K}_{dl}) \geq \frac{\beta_l^2}{4} + 2\delta \quad (3.54)$$

$$\lambda_{\min}(\mathbf{K}_{dr}) \geq \frac{\lambda_{\max}(\mathbf{K}_{pr})^2}{4} + 2\delta \quad (3.55)$$

$$\lambda_{\min}(\mathbf{K}_{vh}) \geq 2\delta, \quad (3.56)$$

donde δ es una constante positiva. Entonces, es fácil obtener

$$\dot{V}_h(\mathbf{x}) \leq -2\delta\|\mathbf{x}\|^2 + \beta_h\|\mathbf{x}\|. \quad (3.57)$$

Si se define

$$\mu_h \triangleq \frac{\beta_h}{\delta}, \quad (3.58)$$

entonces siempre que $\|\mathbf{x}\| \geq \mu_h$ uno tiene

$$\dot{V}_h(\mathbf{x}) \leq -\delta\|\mathbf{x}\|^2 \triangleq -W_3(\mathbf{x}). \quad (3.59)$$

Eligiendo $\alpha_1(\|\mathbf{x}\|) = \lambda_{h1}\|\mathbf{x}\|^2$ y $\alpha_2(\|\mathbf{x}\|) = \lambda_{h2}\|\mathbf{x}\|^2$ en (3.40) puede ser visto que las condiciones (3.16) y (3.17) pueden ser cumplidas en \mathcal{D}_h mediante la elección apropiada de ganancias, i.e (3.52)-(3.56). De acuerdo con (3.19) la condición inicial debe cumplir con

$$\|\mathbf{x}(t_0)\| \leq \sqrt{\frac{\lambda_{h1}}{\lambda_{h2}}} x_{\text{máx}}, \quad (3.60)$$

y de acuerdo con (3.21) el estado final estará finalmente acotado por

$$\|\mathbf{x}(t)\| \leq \sqrt{\frac{\lambda_{h2}}{\lambda_{h1}}} \mu_h \triangleq b_{\text{fh}}. \quad (3.61)$$

Nótese que δ debe ser suficientemente grande tal que $\mu_h < \sqrt{\frac{\lambda_{h1}}{\lambda_{h2}}} x_{\text{máx}}$. Sin embargo la cota final b_{fh} *no se puede hacer arbitrariamente pequeña*. La razón es que el movimiento de cada manipulador está restringido ya sea por el ambiente o por el humano, con lo que es imposible alcanzar seguimiento o regulación sin hacer más suposiciones, como será visto abajo en el siguiente paso.

3. Hasta el momento se ha demostrado que siempre que $\boldsymbol{\tau}_h \neq \mathbf{0}$ y/o $\boldsymbol{\tau}_e \neq \mathbf{0}$, el acotamiento final de las trayectorias \mathbf{x} puede ser garantizado, pero la cota final no puede ser hecha arbitrariamente pequeña. Sin embargo, la dinámica de los diferentes subsistemas puede ser además analizada de manera separada bajo algunas condiciones. En cuanto a los observadores, uno tiene de (3.32)-(3.33) que

$$\|\mathbf{r}_i\| \leq \frac{1}{\lambda_{\text{mín}}(\mathbf{K}_{di})} \|\ddot{\mathbf{q}}_i\|, \quad (3.62)$$

para $i = 1, r$. Esto significa que con el incremento de $\lambda_{\text{mín}}(\mathbf{K}_{di})$ el valor de $\|\mathbf{r}_i\|$ puede ser hecho arbitrariamente pequeño sin violar ninguna restricción porque $\|\ddot{\mathbf{q}}_i\|$ está acotada en \mathcal{D}_h . En cuanto a los errores de observación, de (3.49)-(3.50), ambos \mathbf{z}_i y $\dot{\mathbf{z}}_i$ pueden ser hechos arbitrariamente pequeños a su vez, como se expresa en el inciso i . de la Proposición 3.2.1.

4. Supóngase que $\boldsymbol{\tau}_h \neq \mathbf{0}$ y $\boldsymbol{\tau}_e = \mathbf{0}$ y recuérdese que se ha demostrado que para $\boldsymbol{x} \in \mathcal{D}_h$ cada variable de interés está acotada siempre que las condiciones dadas en el inciso 2. se cumplan. Como se explicó antes, haciendo las ganancias arbitrariamente grandes llevaría a un conflicto entre las restricciones impuestas, ya sea por el humano o por ambiente y el movimiento del manipulador. Ya que la cota final ha sido garantizada para el sistema completo en lazo cerrado, un análisis más fino puede ser llevado a cabo separadamente para el manipulador remoto y el local. Por lo anterior, considérese la dinámica en lazo cerrado dada por (3.28) y defina una función positiva definida

$$V_r(\mathbf{s}_{qr}) = \frac{1}{2} \mathbf{s}_{qr}^T \mathbf{H}_r(\mathbf{q}_r) \mathbf{s}_{qr}, \quad (3.63)$$

cuya derivada puede ser fácilmente calculada como

$$\dot{V}_r = -\mathbf{s}_{qr}^T \mathbf{K}_{vr} \mathbf{s}_{qr} + \mathbf{s}_{qr}^T \mathbf{K}_{pr} \mathbf{r}_r + \mathbf{s}_{qr}^T \mathbf{y}_{ar} \quad (3.64)$$

$$\leq -\lambda_{\min}(\mathbf{K}_{vr}) \|\mathbf{s}_{qr}\|^2 + (\lambda_{\max}(\mathbf{K}_{pr}) \|\mathbf{r}_r\| + \alpha_r) \|\mathbf{s}_{qr}\|, \quad (3.65)$$

con α_r dada en (3.45) para $e_{\max} = 0$. Nótese que para garantizar el acotamiento final, las condiciones (3.52)–(3.56) deben satisfacerse y que $\|\mathbf{r}_r\|$ pueden ser hechas arbitrariamente pequeñas. De este modo, con el uso una vez más del Teorema 3.2.1, se puede concluir que $\|\mathbf{s}_{qr}\|$ pueden ser hecha arbitrariamente pequeña sin violar ninguna restricción.

Una vez que ha sido mostrado que el estado \boldsymbol{x} está acotado y que puede ser hecho arbitrariamente pequeño, se puede mostrar que los errores $\Delta \mathbf{q}_r$, $\Delta \dot{\mathbf{q}}_r$, $\Delta \mathbf{q}_l$ and $\Delta \dot{\mathbf{q}}_l$ tienden a cero. Considerando (3.7), (3.10) y (3.22) se obtiene

$$\mathbf{s}_{qr} = \dot{\mathbf{q}}_r - \ddot{\hat{\mathbf{q}}}_l + \boldsymbol{\Lambda}_{xr}(\hat{\mathbf{q}}_r - \ddot{\hat{\mathbf{q}}}_l) + \mathbf{K}_{\gamma r} \boldsymbol{\sigma}_r = \dot{\mathbf{z}}_r + \mathbf{s}_r + \mathbf{K}_{\gamma r} \boldsymbol{\sigma}_r, \quad (3.66)$$

o

$$\mathbf{s}_r + \mathbf{K}_{\gamma r} \boldsymbol{\sigma}_r = \mathbf{s}_{qr} - \dot{\mathbf{z}}_r. \quad (3.67)$$

De (3.6) y (3.67) se obtiene

$$\dot{\mathbf{s}}_r = -\mathbf{K}_{\gamma r} (\mathbf{K}_{\beta r} \mathbf{s}_r + \text{sign}(\mathbf{s}_r)) + \dot{\mathbf{s}}_{qr} - \ddot{\mathbf{z}}_r. \quad (3.68)$$

Mediante el uso de esta ecuación y el hecho de que $\dot{\mathbf{s}}_{qr} - \ddot{\mathbf{z}}_r$ está acotada en \mathcal{D}_h , se puede mostrar que $\mathbf{s}_r \equiv \mathbf{0}$ en tiempo finito [13]. Entonces, de (3.7) es claro que $\Delta \mathbf{q}_r, \Delta \dot{\mathbf{q}}_r \rightarrow \mathbf{0}$, lo cual implica que

$$\hat{\mathbf{q}}_r \rightarrow \ddot{\hat{\mathbf{q}}}_l \quad \text{y} \quad \dot{\hat{\mathbf{q}}}_r \rightarrow \ddot{\dot{\hat{\mathbf{q}}}}_l. \quad (3.69)$$

De manera similar puede ser mostrado que

$$\hat{\mathbf{q}}_l \rightarrow \ddot{\hat{\mathbf{q}}}_r \quad \text{y} \quad \dot{\hat{\mathbf{q}}}_l \rightarrow \ddot{\dot{\hat{\mathbf{q}}}}_r. \quad (3.70)$$

Esto conduce a la conclusión de que

$$\mathbf{q}_r(t) \approx \mathbf{q}_l(t - T_l) \quad \text{y} \quad \dot{\mathbf{q}}_r(t) \approx \dot{\mathbf{q}}_l(t - T_l), \quad (3.71)$$

Ésto prueba el inciso *iii.* de la Proposición 3.2.1. Nótese que no se puede garantizar

$$\mathbf{q}_l(t) \approx \mathbf{q}_r(t - T_r) \quad \text{y} \quad \dot{\mathbf{q}}_l(t) \approx \dot{\mathbf{q}}_r(t - T_r), \quad (3.72)$$

debido a la fuerza aplicada por el operador humano. En efecto, lo único que se puede decir es que los errores de seguimiento están acotados para el manipulador local. En realidad, no se puede afirmar otra cosa sin hacer suposiciones adicionales, como se hace en el siguiente paso de la demostración.

5. El paso final de la demostración depende en gran medida de la suposición sobre el comportamiento que tiene el operador humano, que permite tener cierto grado de sensación de telepresencia. Por esta razón, debería ser entendido que la ley de control τ_1 en (3.14) está pensada para hacer que el manipulador local siga al manipulador remoto. Esto puede ser apreciado claramente en movimiento libre. Cuando una persona aplica una fuerza, el correspondiente par de entrada τ_h evita que τ_1 alcance este objetivo, dependiendo de la magnitud de la fuerza que el operador aplique. Al mismo tiempo, es más difícil prever las trayectorias que el robot local seguirá. Estableciendo \mathbf{K}_{al} , \mathbf{K}_{pl} muy grandes podría significar a su vez hacer \mathbf{K}_{vl} en (3.26) muy grande. De esta forma, si el par generado por el operador es demasiado pequeño, este más bien podría ser considerado como una perturbación que puede ser compensada. En lugar de eso, si las ganancias en τ_1 son relativamente pequeñas y la persona aplica una fuerza grande, entonces el manipulador local no tendería a seguir a la posición del robot remoto retardada porque τ_h se sobrepondría al efecto de τ_1 . Finalmente, si el humano mueve el efector final del manipulador local lentamente como se asume en el inciso *b)* de la Proposición 3.2.1, entonces el robot remoto seguirá la posición retardada local y a su vez el manipulador local seguirá suavemente el movimiento en aquella dirección porque la ley de control tratará de alcanzar este objetivo, manteniendo los errores de seguimiento pequeños. Cuando el robot remoto toca la superficie rígida y su movimiento está restringido, los errores de seguimiento sobre el lado del robot local se incrementarán en aquella misma dirección y la acción de τ_1 se incrementará para tratar de disminuir el error. Esto se convierte en algo evidente para el operador humano. Si el operador no empuja lo suficientemente fuerte cuando la sensación de tocar algo aparece, entonces el operador será capaz de reconocer la superficie remota, por lo que surge la sensación de telepresencia, como se estableció en el inciso *iv.* de la Proposición 3.2.1. Sin embargo, el humano no sería capaz de sentir que tan fuerte el robot remoto está empujando, así que la transparencia no puede ser alcanzada. Esto concluye la demostración.

Resultados experimentales

En esta sección son presentados los resultados experimentales del esquema controlador observador diseñado en el Capítulo 3, bajo las restricciones que pueden aparecer debidas al operador humano y el ambiente con $\tau_h \neq 0$ y $\tau_e \neq 0$. Los experimentos se realizaron con dos robots *Geomagic Touch*. Cada uno de ellos cuenta con actuadores en sus tres primeras articulaciones.

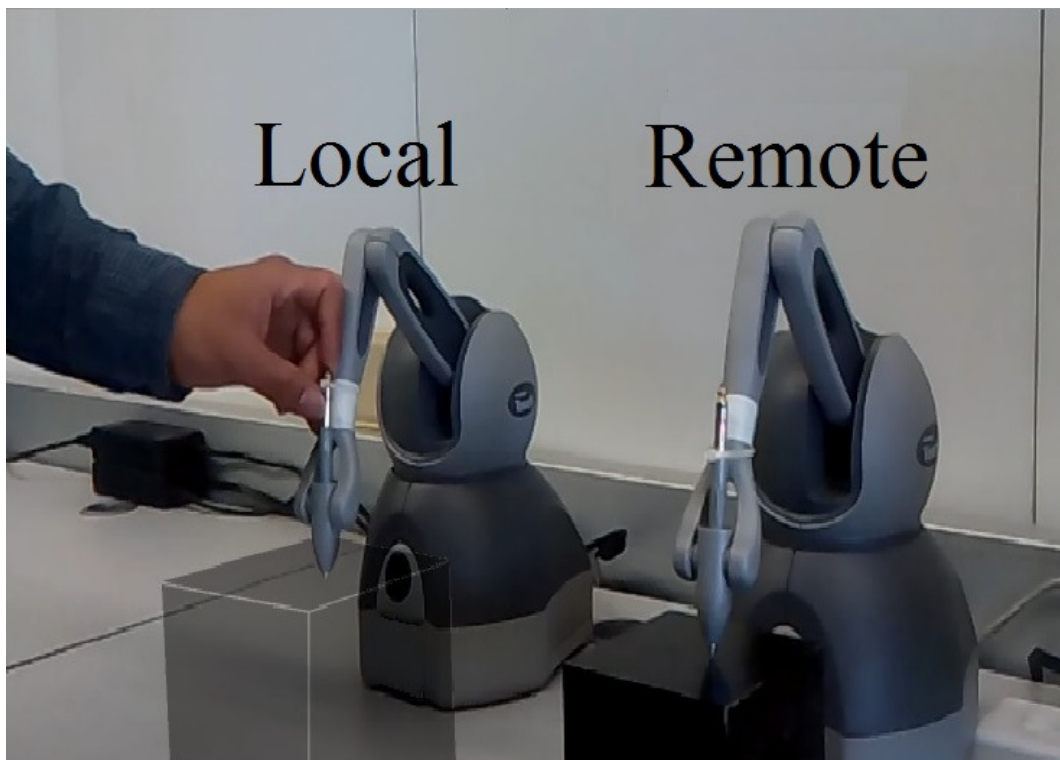


Figura 4.1: Sistema de teleoperación con dos *Geomagic Touch*

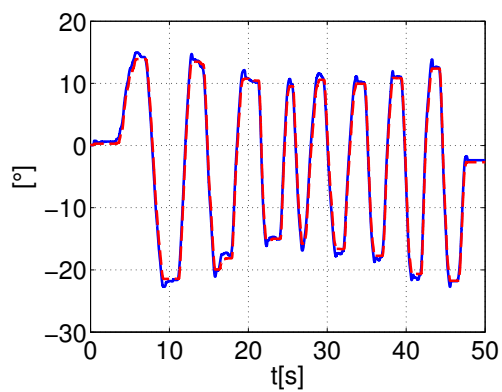
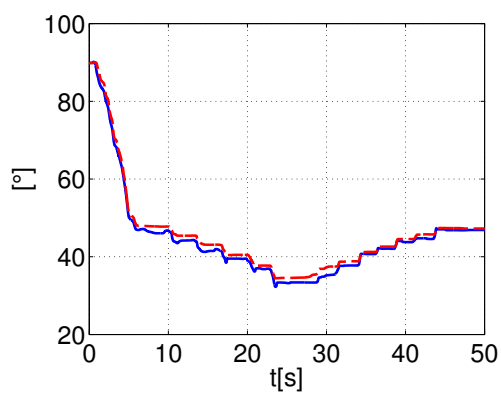
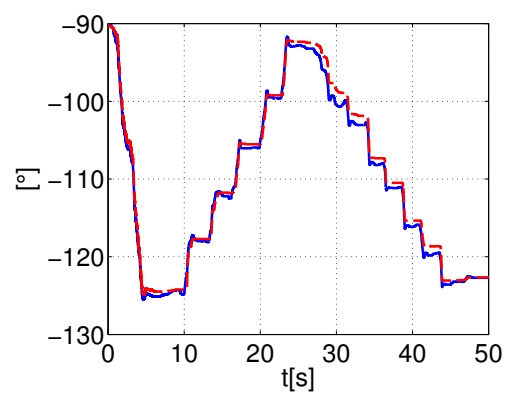
En la Figura 4.1 se muestra en esencia la plataforma experimental. El operador humano mueve el último eslabón del robot local mientras el efector final del robot remoto está siempre en contacto con un objeto. En esta situación la persona, sin estar especialmente adiestrada, tendrá la sensación de estar tocando la superficie.

El algoritmo fue implementado en Lenguaje C++ en el entorno de desarrollo de *Visual Studio*. Los retardos fueron inducidos artificialmente vía software, así que pueden ser establecidos arbitrariamente y constantes. Los códigos pueden ser revisados en el Apéndice A.3.

Los siguientes parámetros usados fueron determinados empíricamente: $\mathbf{K}_{al} = \text{diag} \{0.005, 0.005, 0.005\}$, $\mathbf{K}_{pl} = \text{diag} \{0.065, 0.065, 0.065\}$, $\mathbf{\Lambda}_{xl} = \text{diag} \{15, 18, 12\}$, $\mathbf{\Lambda}_{xr} = \text{diag} \{15, 15, 10\}$, $\mathbf{K}_{\beta r} = 0.000001\mathbf{I}$, $\mathbf{K}_{\beta l} = 0.000001\mathbf{I}$, $\mathbf{K}_{\gamma r} = 0.15\mathbf{I}$, $\mathbf{K}_{\gamma l} = 0.15\mathbf{I}$, $\mathbf{K}_{pr} = \text{diag} \{0.06, 0.06, 0.06\}$, $\mathbf{K}_{dl} = \text{diag} \{120, 115, 120\}$, $\mathbf{\Lambda}_{zr} = \text{diag} \{0.5, 0.5, 0.5\}$, $\mathbf{\Lambda}_{zl} = \text{diag} \{0.5, 0.5, 0.5\}$, y $\mathbf{K}_{dr} = \text{diag} \{120, 115, 120\}$.

4.1. Movimiento restringido sin retardo caso ideal

El primer experimento se llevó a cabo sin la presencia de retardos. Tiene como finalidad establecer un punto de comparación.

(a) $q_{11}(t)$ (—) vs $q_{r1}(t)$ (- - -)(b) $q_{12}(t)$ (—) vs $q_{r2}(t)$ (- - -)(c) $q_{13}(t)$ (—) vs $q_{r3}(t)$ (- - -)**Figura 4.2:** Posiciones sin retardo $q_l(t)$ (—) vs $q_r(t)$ (- - -) [°].

4. RESULTADOS EXPERIMENTALES

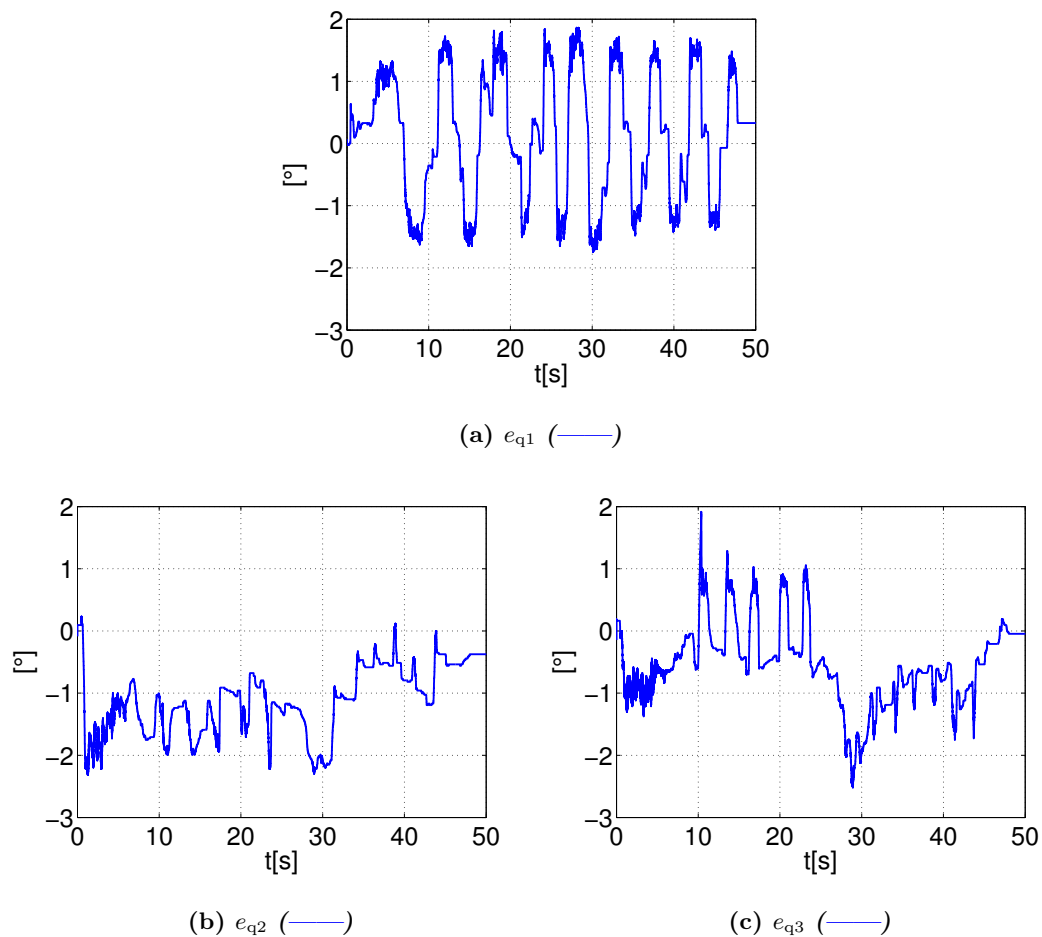
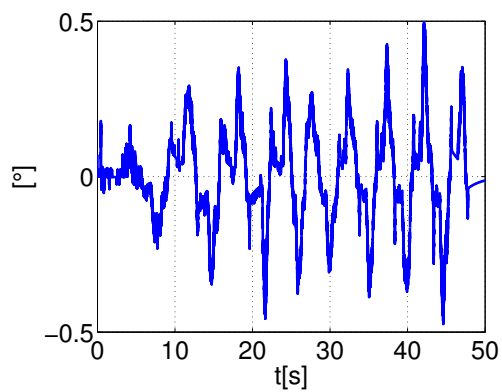
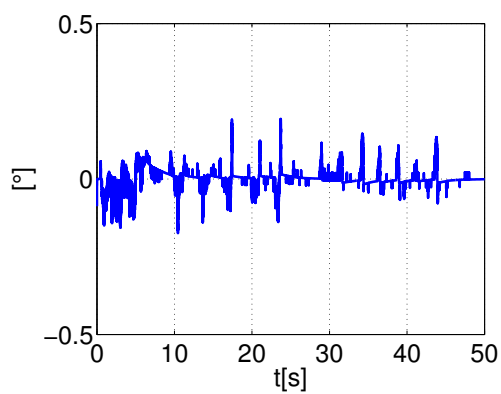


Figura 4.3: Errores de seguimiento sin presencia de retardo $e(t) = q_1(t) - q_r(t)$

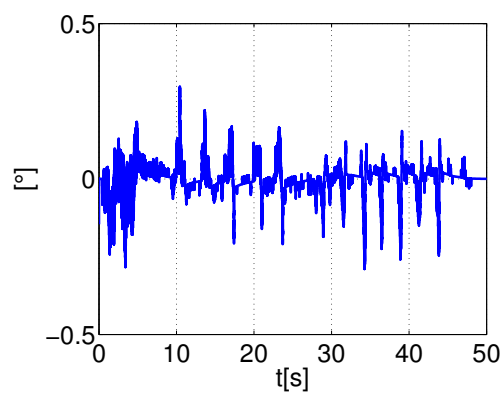
En la Figura 4.2 se muestran las posiciones sin retardo $q_1(t)$ (—) vs $q_r(t)$ (- - -) [°], se puede observar que sin presencia de retardos ambos robots se siguen sin ningún problema. El error de seguimiento mostrado en la Figura 4.3 demuestra esta afirmación, ya que el error permanece dentro de un margen de $\pm 2^\circ$.



(a) z_{11} (—)



(b) z_{12} (—)



(c) z_{13} (—)

Figura 4.4: Errores de observación sin presencia de retardo $z(t) = q_1(t) - \hat{q}_1(t)$

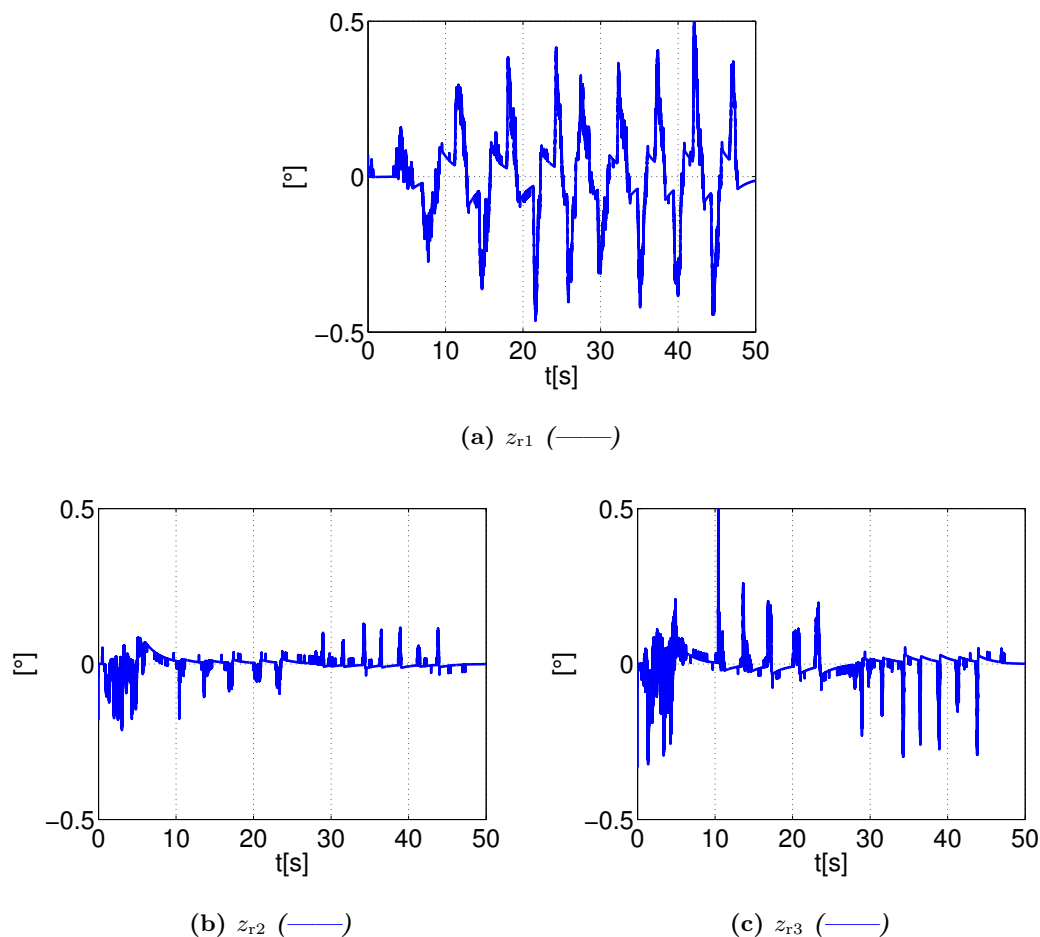
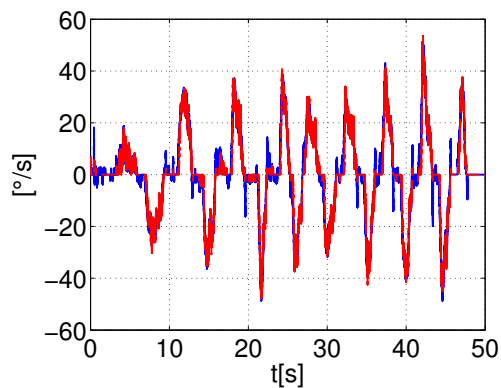
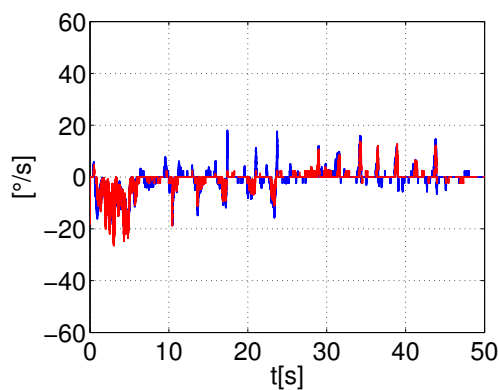
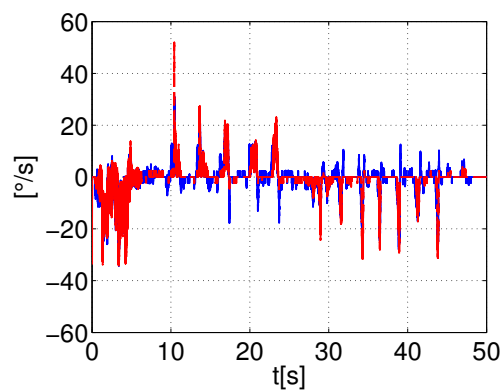


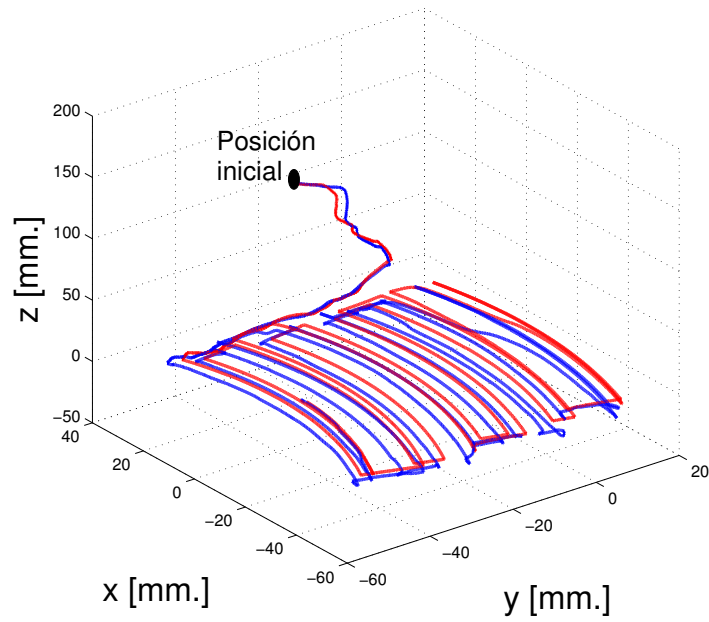
Figura 4.5: Errores de observación sin presencia de retardo $z(t) = q_r(t) - \hat{q}_r(t)$

En las Figuras 4.4 y 4.5 se pueden observar los errores de observación, el desempeño de cada observador montado en cada robot es bueno. El error de observación para ambos casos no rebasa los $\pm 0.5[^\circ]$, este error grande en parte se debe al perfil que describe el humano, sin presencia del retardo, el humano puede describir trayectorias no tan suaves, como era de esperarse, con perfiles no tan suaves este observador arroja un error más grande.

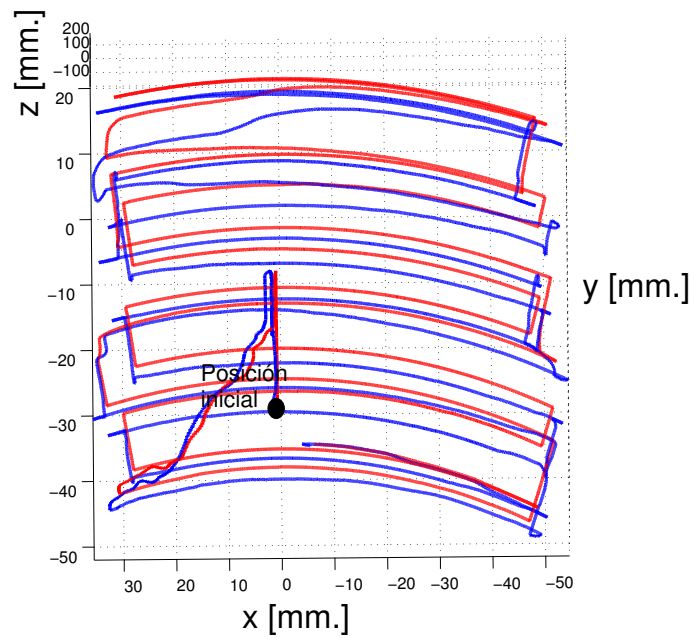
(a) $\dot{q}_1(t)$ (—) vs $\dot{q}_{r1}(t)$ (- - -)(b) $\dot{q}_{12}(t)$ (—) vs $\dot{q}_{r2}(t)$ (- - -)(c) $\dot{q}_{13}(t)$ (—) vs $\dot{q}_{r3}(t)$ (- - -)**Figura 4.6:** Velocidades estimadas sin retardo $\hat{q}_i(t)$ (—) vs $\hat{q}_{r_i}(t)$ (- - -) [$^{\circ}/s$].

En la Figura 4.6 se observan las velocidades obtenidas mediante el observador propuesto. Como se puede observar las velocidades estimadas $\hat{q}_i(t)$ (—) son muy similares a las velocidades estimadas $\hat{q}_{r_i}(t)$ (- - -) [$^{\circ}/s$]. En este caso, sin presencia del retardo se observa que existen picos grandes en las velocidades, sin la presencia del retardo los perfiles que puede seguir el humano pueden ser no tan suaves.

4. RESULTADOS EXPERIMENTALES



(a) *Vista isométrica*



(b) *Vista superior*

Figura 4.7: Superficie virtual $p_1(t)$ (—) vs superficie física $p_r(t)$ (- - -) sin presencia de retardo

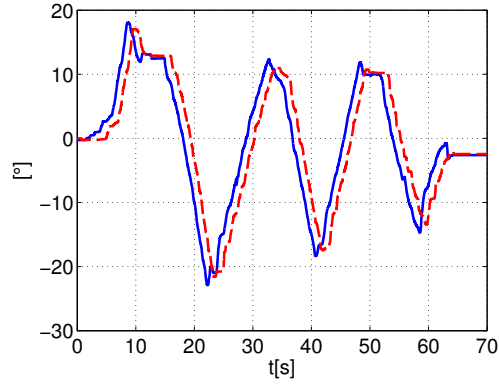
En la Figura 4.7 se observa que la descripción de la superficie por parte del robot local es similar a la restricción física a la cual está sujeta el robot remoto. Como se mostrará en los siguientes resultados, este desempeño se verá afectado, pero la estabilidad del sistema de teleoperación se mantendrá.

4.2. Movimiento restringido con retardo

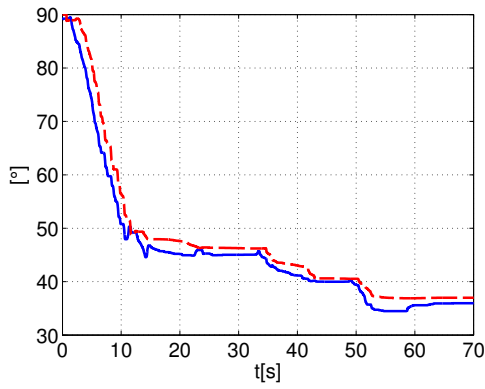
Se llevaron a cabo tres experimentos considerando el retardo. El primero se realizó con un retardo asimétrico, es decir, el tiempo de ida y de regreso de la información útil (posición y velocidad) son diferentes, mientras que el segundo es realizado con un retardo simétrico. El tercer experimento se realizó con la finalidad de demostrar el funcionamiento y desempeño del esquema propuesto aún en el escenario donde no se conoce nada sobre el modelo dinámico.

4.2.1. Retardo asimétrico

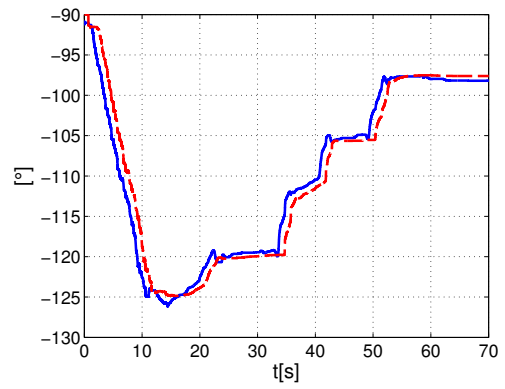
Para este experimento los retardos de tiempo fueron establecidos en $T_1 = 0.3s$ y $T_r = 0.7s$. En este caso se toma en cuenta el escenario donde el operador humano mueve el efector final, mientras que el robot remoto se encuentra restringido por una caja de aluminio como puede verse en la Figura 4.1.



(a) $q_{11}(t)$ (—) vs $q_{r1}(t - T_r)$ (- - -)



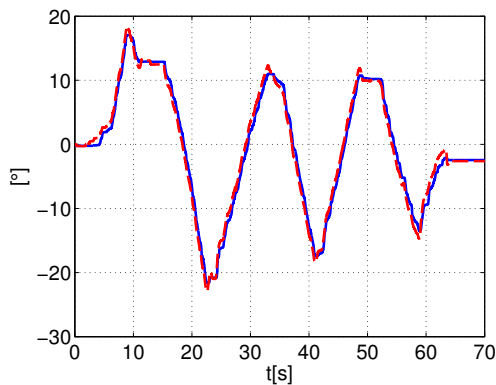
(b) $q_{12}(t)$ (—) vs $q_{r2}(t - T_r)$ (- - -)



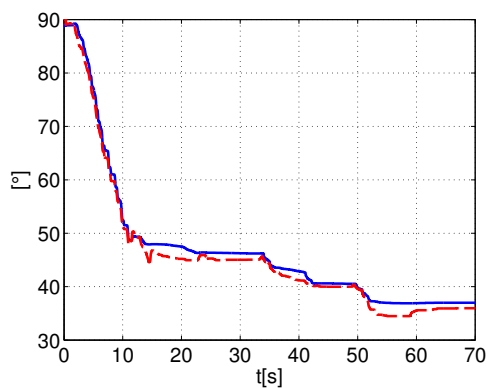
(c) $q_{13}(t)$ (—) vs $q_{r3}(t - T_r)$ (- - -)

Figura 4.8: Posiciones con retardo asimétrico $q_1(t)$ (—) vs $q_r(t - 0.7s)$ (- - -) [°].

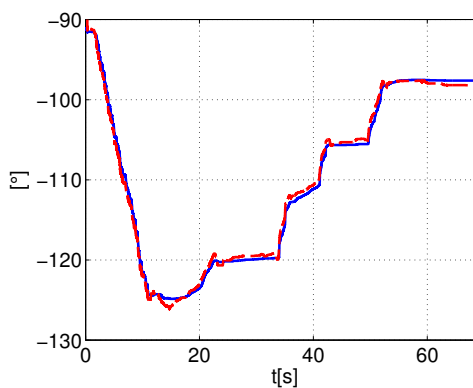
4. RESULTADOS EXPERIMENTALES



(a) $q_{r1}(t)$ (---) vs $q_1(t - T_1)$ (—)



(b) $q_{r2}(t)$ (---) vs $q_2(t - T_1)$ (—)



(c) $q_{r3}(t)$ (---) vs $q_3(t - T_1)$ (—)

Figura 4.9: Posiciones con retardo asimétrico $q_r(t)$ (---) vs $q_r(t - 0.3s)$ (—) [$^\circ$].

En las Figuras 4.8 y 4.9 se muestran los perfiles de posición de ambos robots. El algoritmo propuesto permite que ambos robots se sigan entre sí. Esta afirmación cobra validez al observar las Figuras 4.10 y 4.11. En la primera se pueden observar el error de seguimiento que presenta el robot local hacia el remoto, notesé que la amplitud de los errores es mayor que la segunda Figura, este efecto de diferencia entre los errores es debido a que el robot local está tratando de seguir un perfil de referencia atrasado $0.7s$, el humano no tendrá la sensación de tocar la superficie sino hasta $0.7s$ después, este retardo implica a su vez, que el error sea más grande. Visto desde el lado del robot remoto éste seguirá un perfil de referencia atrasado $0.3s$, por lo que puede actualizar su posición con mayor frecuencia, este efecto se ve reflejado en un error de seguimiento más pequeño.

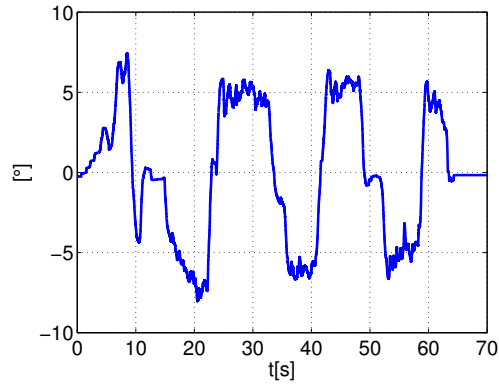
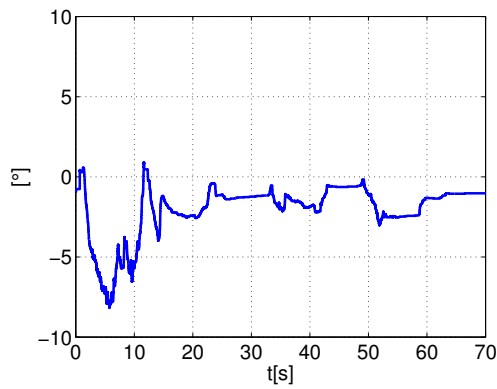
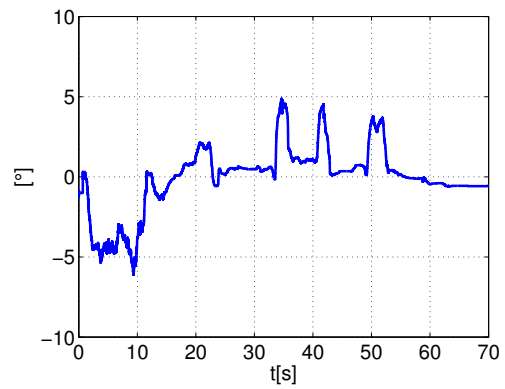
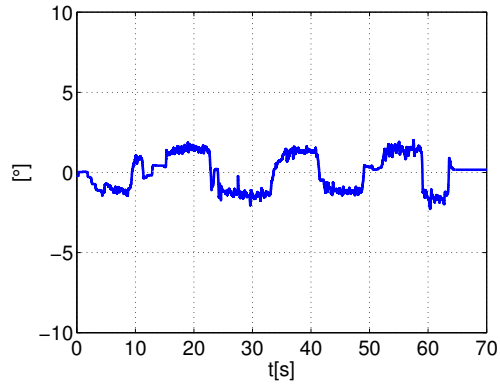
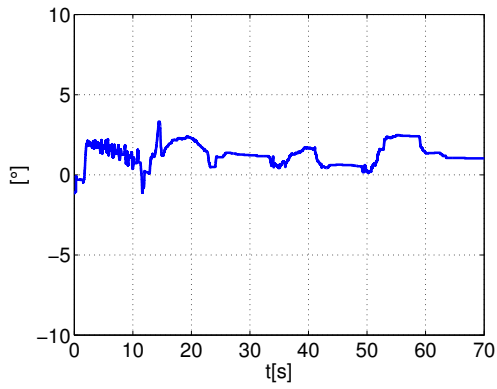
(a) e_{q1} (—)(b) e_{q2} (—)(c) e_{q3} (—)

Figura 4.10: Errores de seguimiento con presencia de retardo asimétrico $e(t) = q_1(t) - q_1(t - 0.7s)$

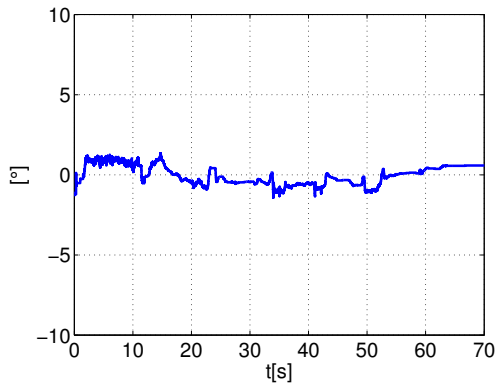
4. RESULTADOS EXPERIMENTALES



(a) e_{q1} (—)



(b) e_{q2} (—)



(c) e_{q3} (—)

Figura 4.11: Errores de seguimiento con presencia de retardo asimétrico $e(t) = q_r(t) - q_l(t - 0.3s)$

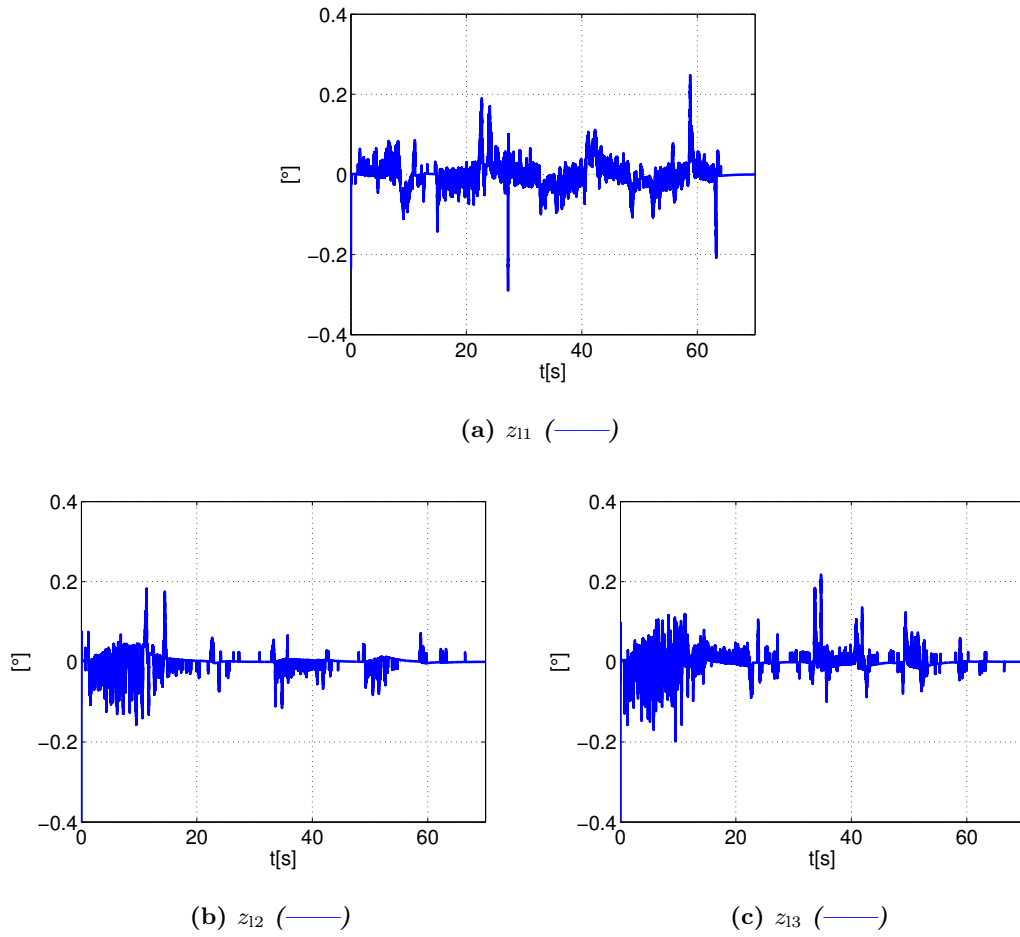


Figura 4.12: Errores de observación con presencia de retardo asimétrico $z_i = q_i(t) - \hat{q}_i(t)$

4. RESULTADOS EXPERIMENTALES

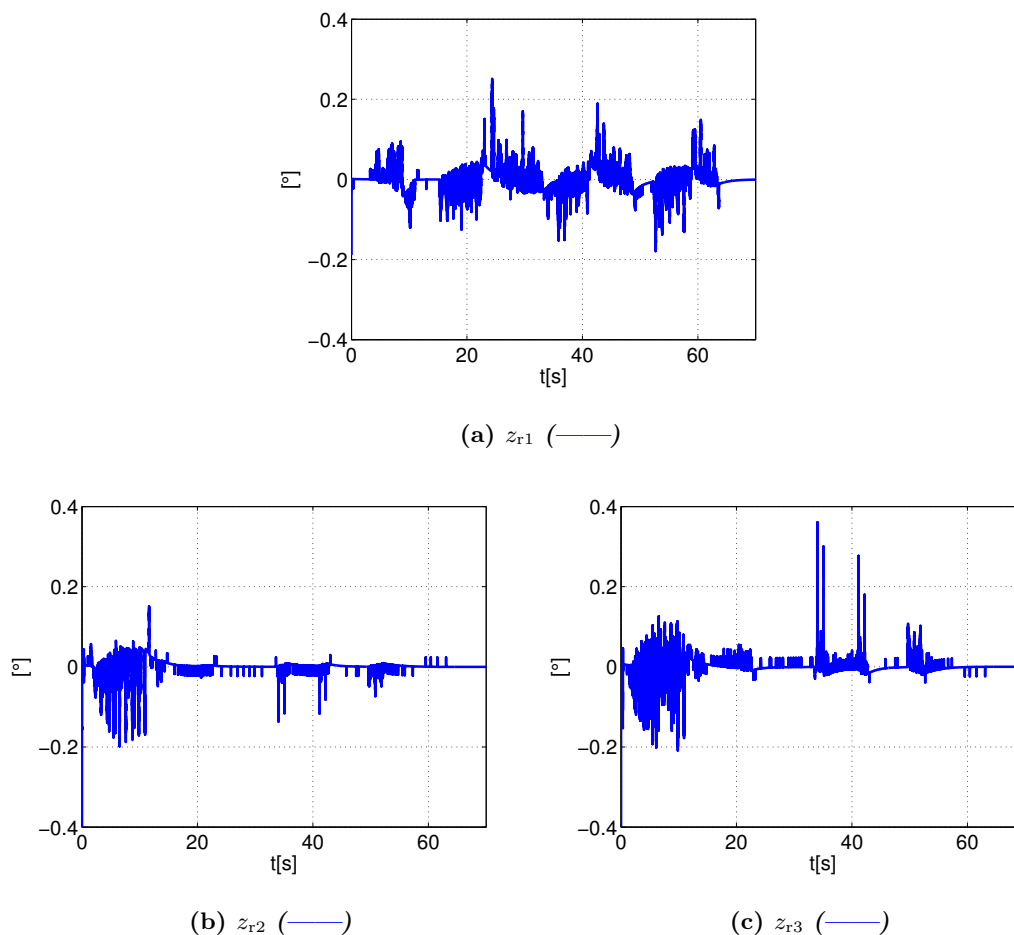


Figura 4.13: Errores de observación con presencia de retardo asimétrico $z_r = q_r(t) - \hat{q}_r(t)$

A diferencia de las Figuras 4.4 y 4.5, las amplitudes de los errores mostrados en las Figuras 4.12 y 4.13 son menores. Los errores muestran tener una amplitud de $\pm 0.4[^\circ]$. Una consecuencia de la aparición del retardo, es que ahora el humano está obligado a describir perfiles de posición y velocidad más suaves.

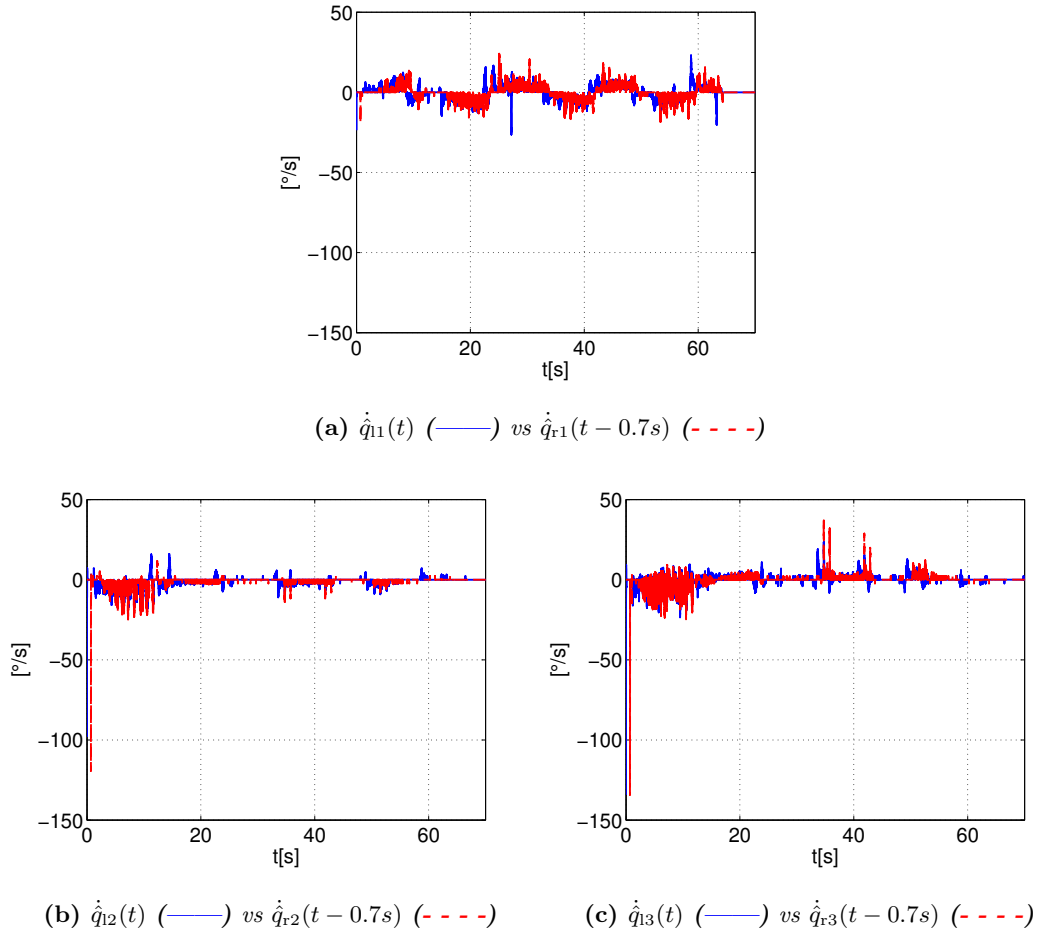


Figura 4.14: Velocidad observada $\dot{q}_i(t)$ (—) vs $\dot{q}_r(t - 0.7s)$ (- - - -) [$^{\circ}$].

4. RESULTADOS EXPERIMENTALES

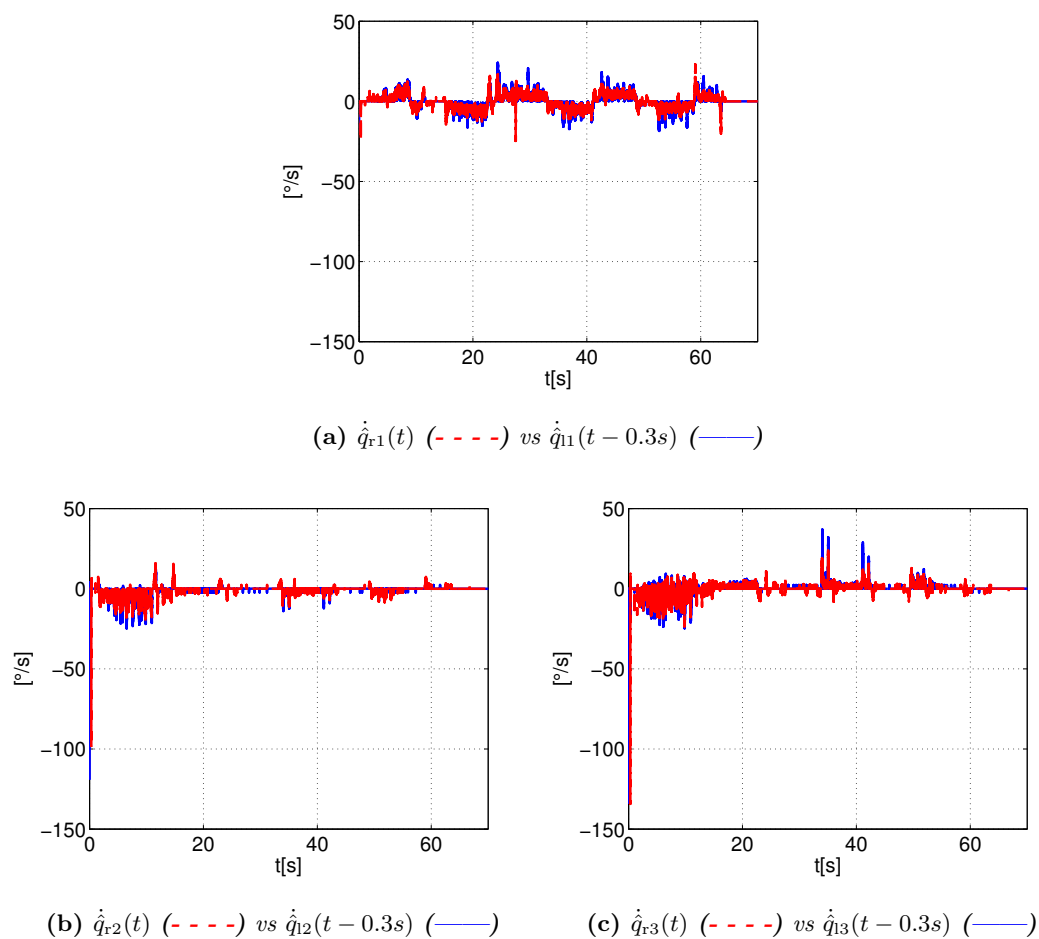


Figura 4.15: Velocidad observada $\dot{q}_r(t)$ (---) vs $\dot{q}_l(t - 0.3s)$ (—) [$^\circ$].

En las Figuras 4.14 y 4.15 se muestran las velocidades obtenidas con el observador propuesto, a diferencia del experimento anterior se obtienen velocidades de menor amplitud.

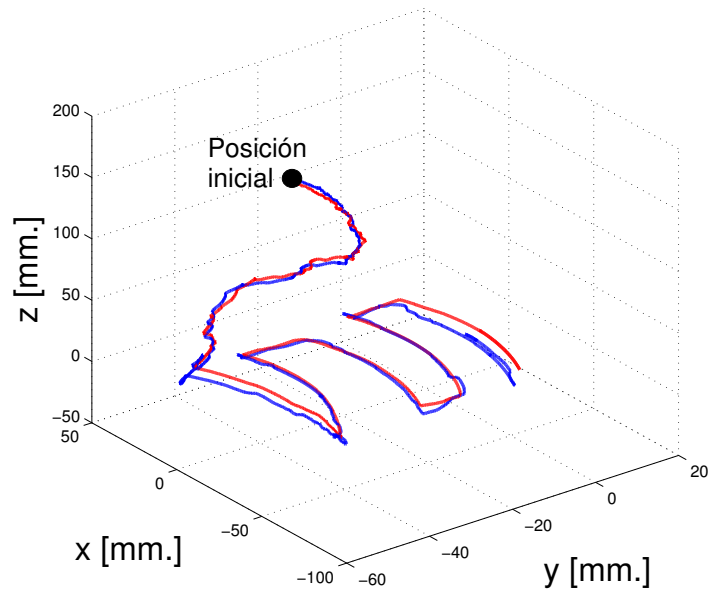
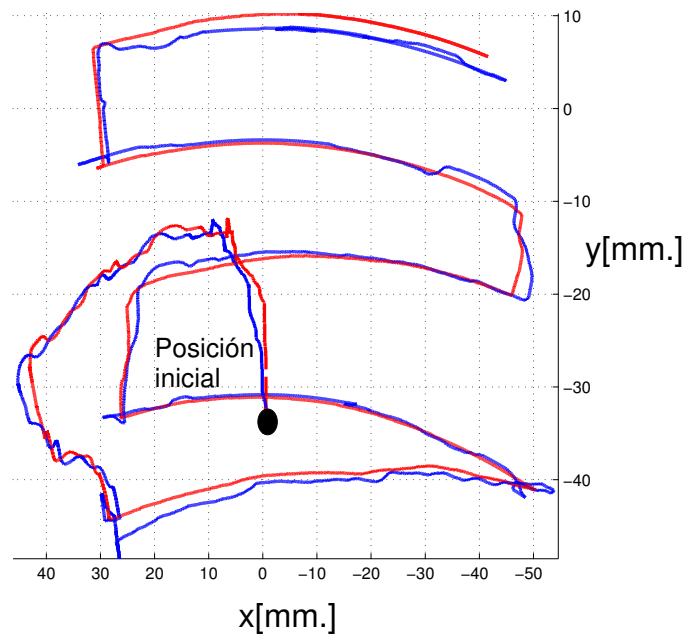
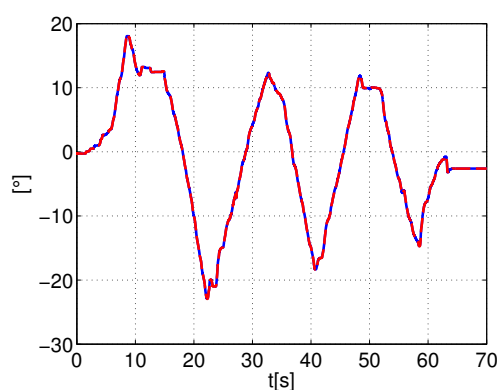
(a) *Vista isométrica*(b) *Vista superior*

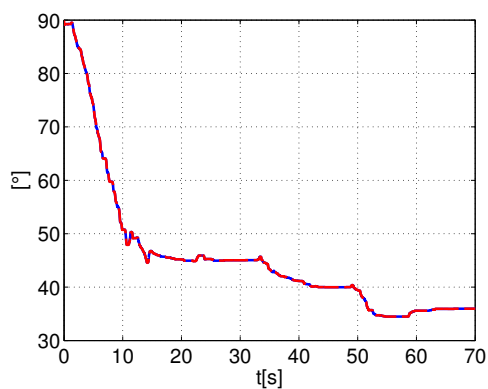
Figura 4.16: Superficie real $p_1(t)$ (—) vs superficie virtual $p_r(t)$ (---) con presencia de retardo asimétrico

4. RESULTADOS EXPERIMENTALES

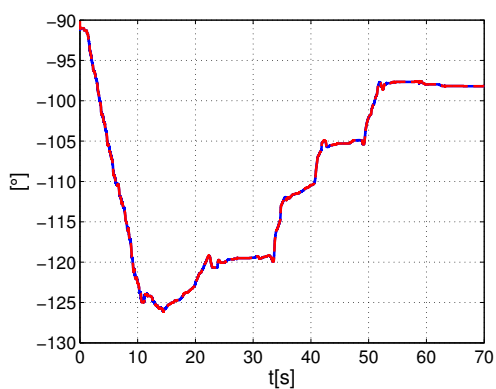
En la Figura 4.16 se puede observar la superficie percibida por el operador humano, nótese que esta superficie percibida es muy similar a la restricción física. Como era de esperarse se aprecia una superficie algo deformada, además ya está presente una pérdida de transparencia, este efecto es debido al retardo y a la inyección de amortiguamiento del lado del robot local. Como se esperaba la pasividad se contrapone a la transparencia. En este trabajo se tiene como prioridad garantizar que el sistema sea estable. A pesar de este sacrificio, aún con retardos grandes se mantiene la sensación de telepresencia y las trayectorias del sistema continúan siendo acotadas.



(a) $q_{11}(t)$ (—) vs $\hat{q}_{r1}(t)$ (- - -)

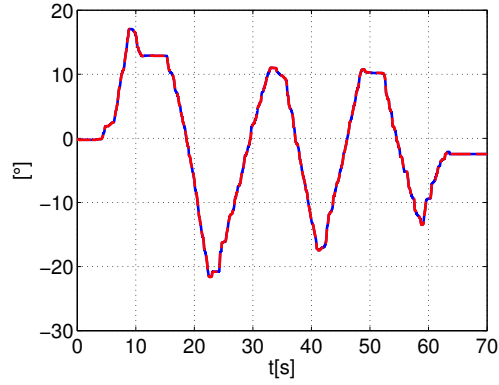
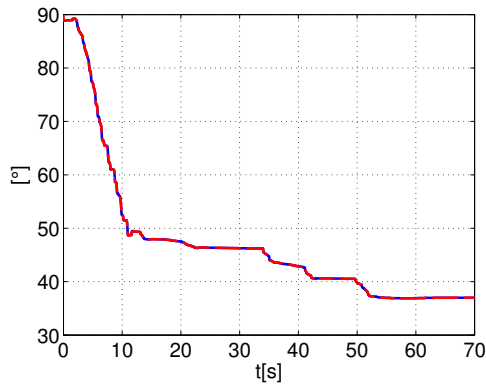
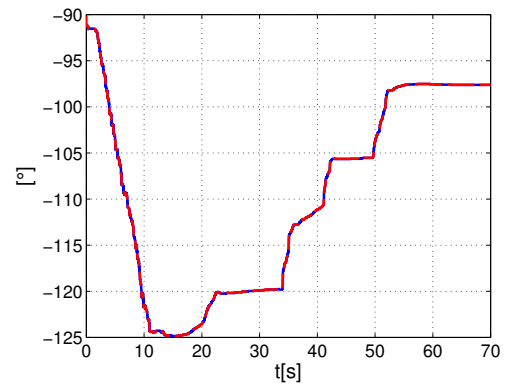


(b) $q_{12}(t)$ (—) vs $\hat{q}_{r2}(t)$ (- - -)



(c) $q_{13}(t)$ (—) vs $\hat{q}_{r3}(t)$ (- - -)

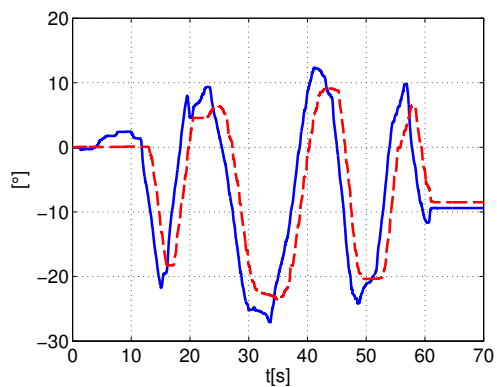
Figura 4.17: Posiciones medidas $q_i(t)$ (—) vs posiciones estimadas $\hat{q}_r(t)$ (- - -) [°].

(a) $q_{r1}(t)$ (—) vs $\hat{q}_1(t)$ (- - -)(b) $q_{r2}(t)$ (—) vs $\hat{q}_{12}(t)$ (- - -)(c) $q_{r3}(t)$ (—) vs $\hat{q}_{13}(t)$ (- - -)**Figura 4.18:** Posiciones medidas $q_r(t)$ (—) vs posiciones estimadas $\hat{q}_i(t)$ (- - -) [$^\circ$].

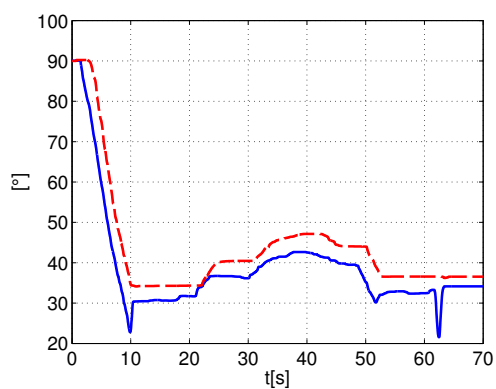
Las Figuras 4.15 y 4.16 muestran las posiciones estimadas obtenidas con los observadores, es claro que se obtiene una posición estimada muy aproximada a la posición real de cada robot.

4.2.2. Retardo simétrico

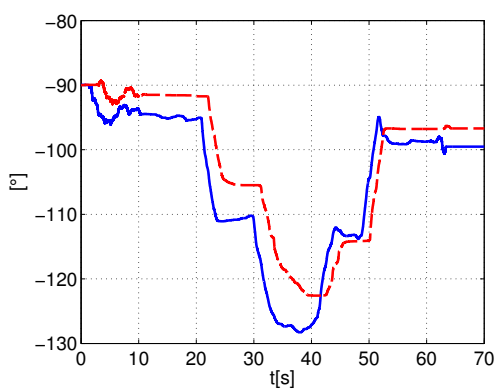
El siguiente experimento tiene como objetivo emular un retardo de tiempo similar en el envío y recepción de información. Para este experimento, los retardos de tiempo fueron establecidos en $T_l = 0.5s$ y $T_r = 0.5s$.



(a) $q_{11}(t)$ (—) vs $q_{r1}(t - T_r)$ (- - -)

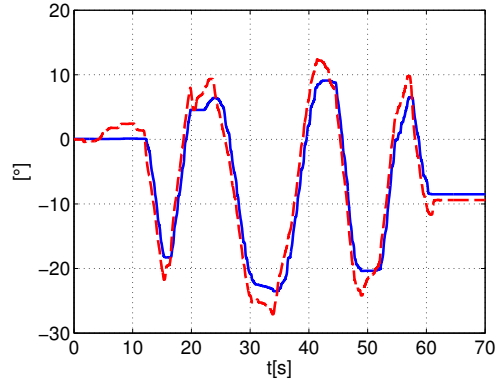
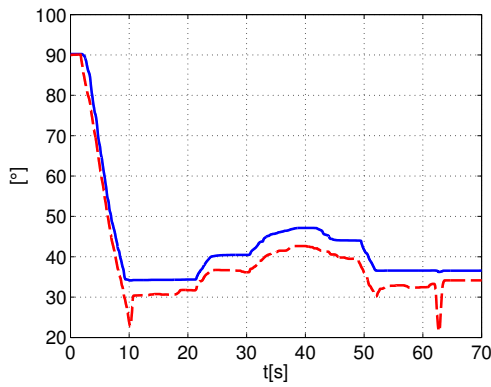
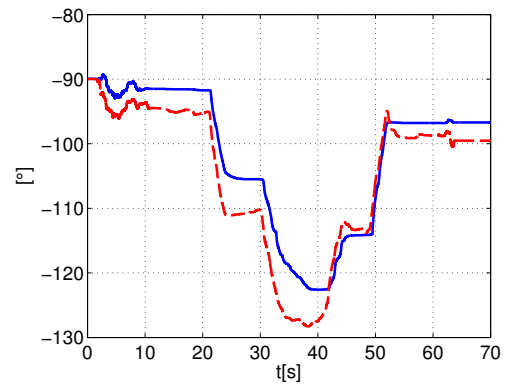


(b) $q_{12}(t)$ (—) vs $q_{r2}(t - T_r)$ (- - -)



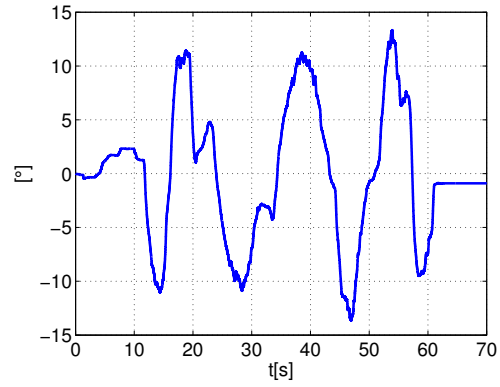
(c) $q_{13}(t)$ (—) vs $q_{r3}(t - T_r)$ (- - -)

Figura 4.19: Posiciones con retardo simétrico $q_1(t)$ (—) vs $q_r(t - 0.5s)$ (- - -) [°].

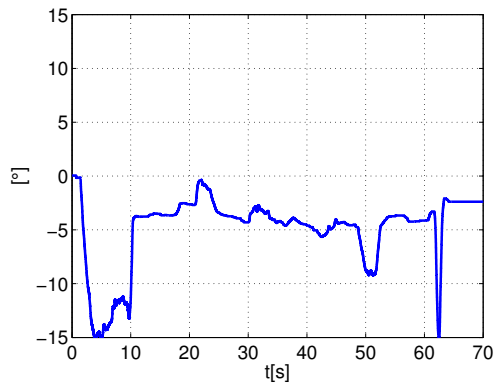
(a) $q_{r1}(t)$ (---) vs $q_{r1}(t - T_i)$ (—)(b) $q_{r2}(t)$ (---) vs $q_{r2}(t - T_i)$ (—)(c) $q_{r3}(t)$ (---) vs $q_{r3}(t - T_i)$ (—)**Figura 4.20:** Posiciones con retardo simétrico $q_r(t)$ (---) vs $q_r(t - 0.5s)$ (—)[°].

En las Figuras 4.19 y 4.20 se muestran los perfiles de posición de ambos robots. Nuevamente el algoritmo propuesto permite que ambos robots se sigan entre sí. Sin embargo, existe un error más grande como se puede apreciar en las Figuras 4.21 y 4.22.

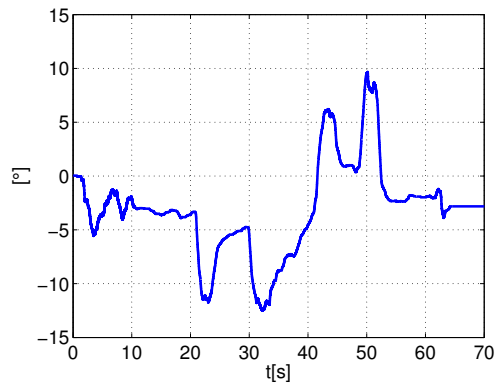
4. RESULTADOS EXPERIMENTALES



(a) e_{q1} (—)



(b) e_{q2} (—)



(c) e_{q3} (—)

Figura 4.21: Errores de seguimiento con presencia de retardo simétrico $e(t) = q_1(t) - q_r(t - 0.5s)$

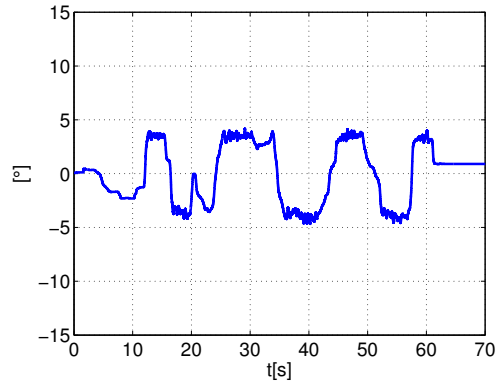
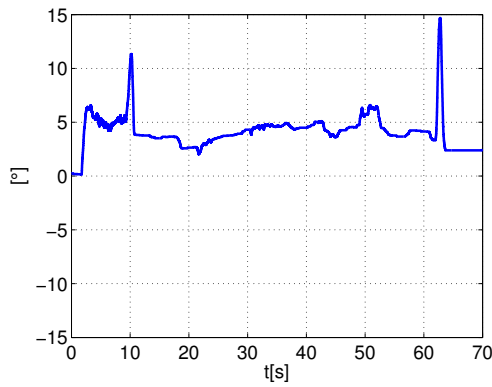
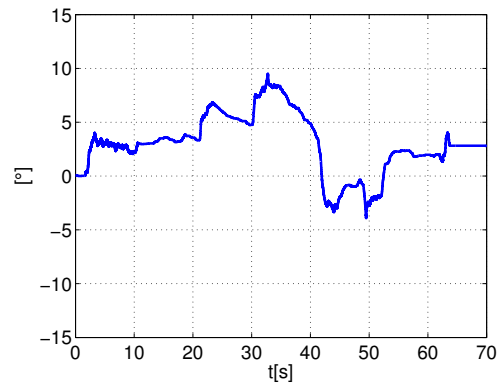
(a) e_{q1} (—)(b) e_{q2} (—)(c) e_{q3} (—)

Figura 4.22: Errores de seguimiento con presencia de retardo simétrico $e(t) = q_r(t) - q_l(t - 0.5s)$

4. RESULTADOS EXPERIMENTALES

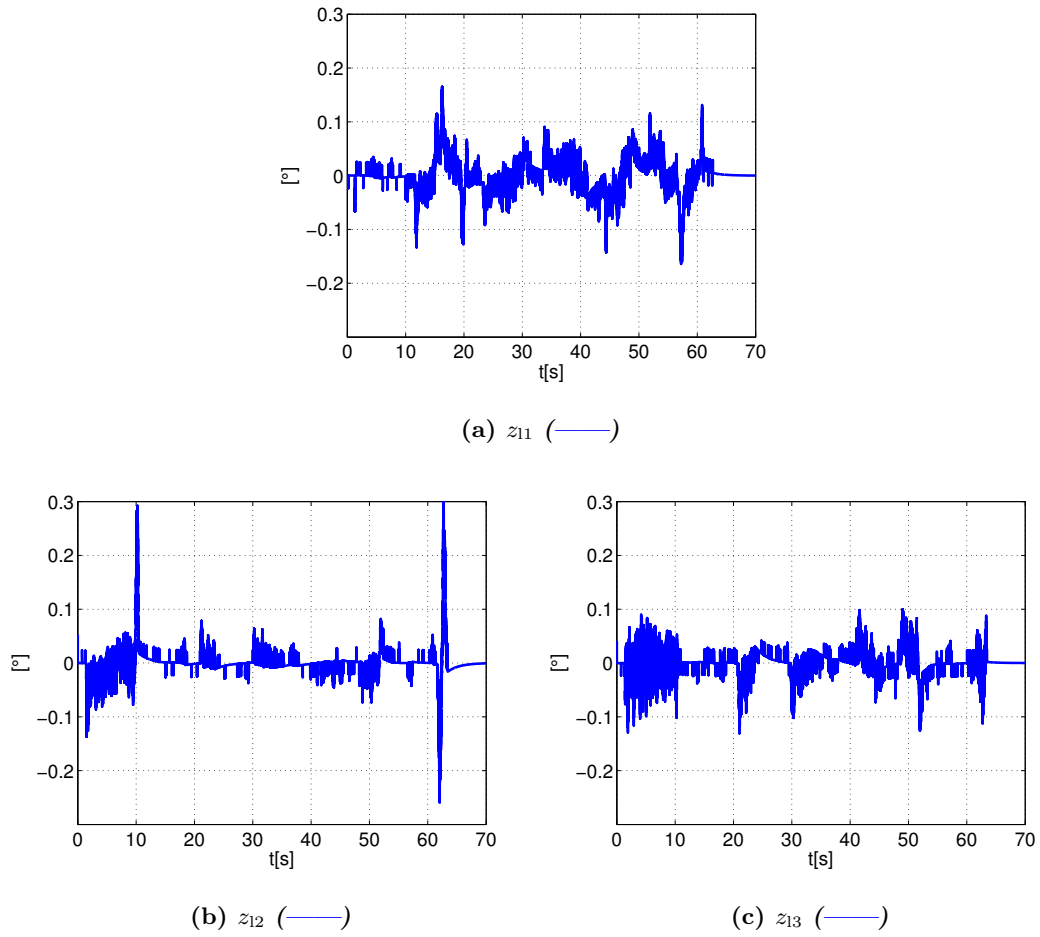


Figura 4.23: Errores de observación con presencia de retardo asimétrico $z_1 = q_1(t) - \hat{q}_1(t)$

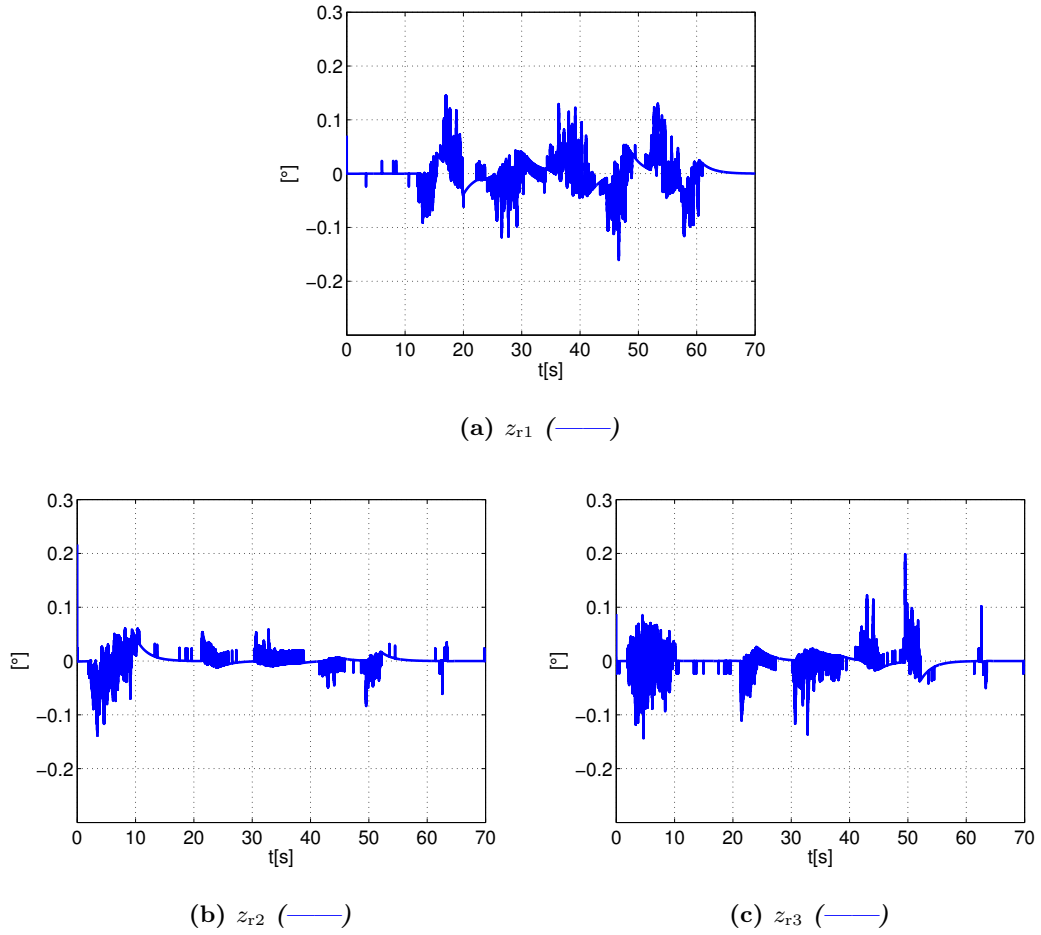


Figura 4.24: Errores de observación con presencia de retardo asimétrico $z_r = q_r(t) - \hat{q}_r(t)$

En las Figuras 4.23 y 4.24 se muestran los errores de observación, con retardos iguales de ida y regreso, el observador propuesto demuestra tener buen desempeño. Se observa que existe un error con una amplitud de $\pm 0.3^\circ$.

4. RESULTADOS EXPERIMENTALES

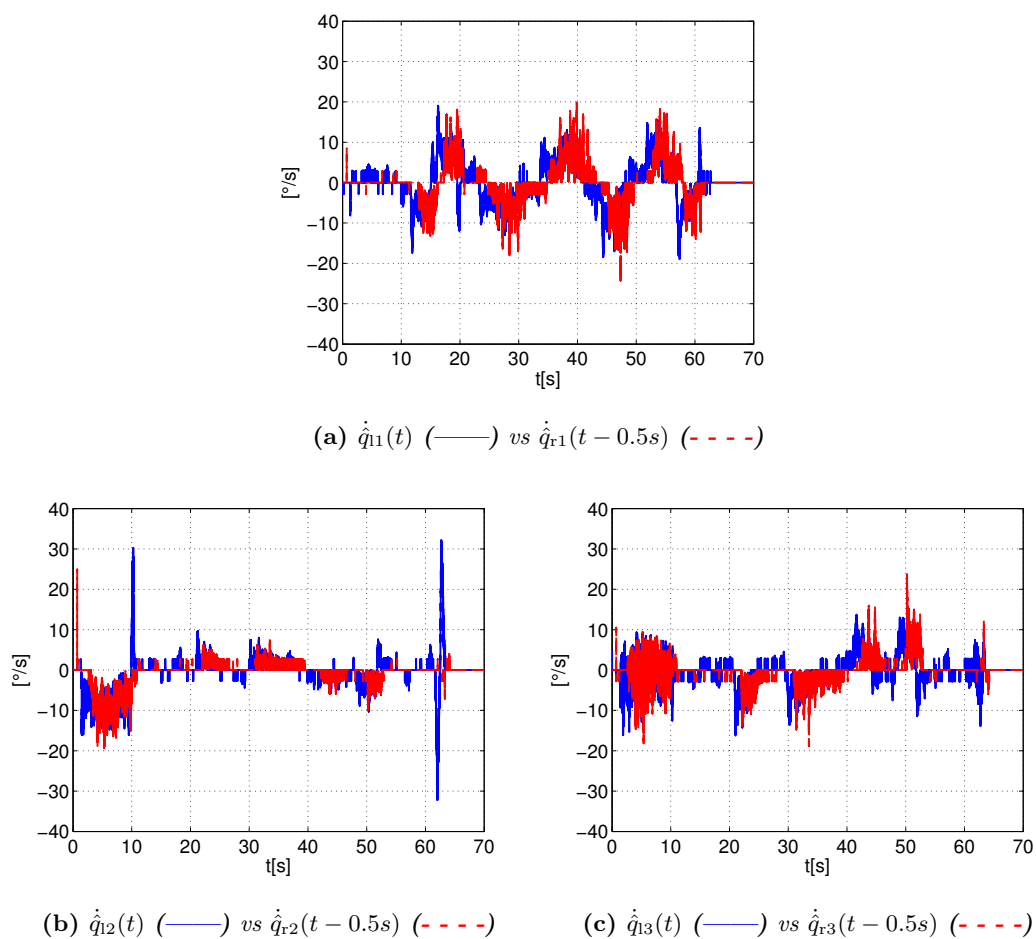
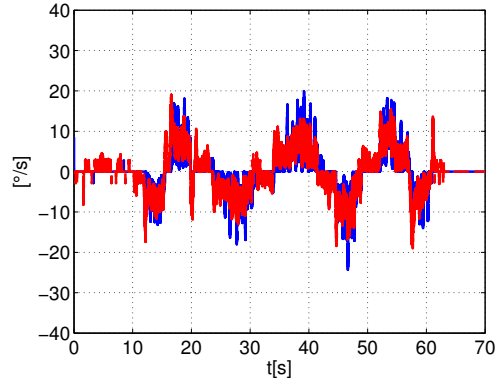
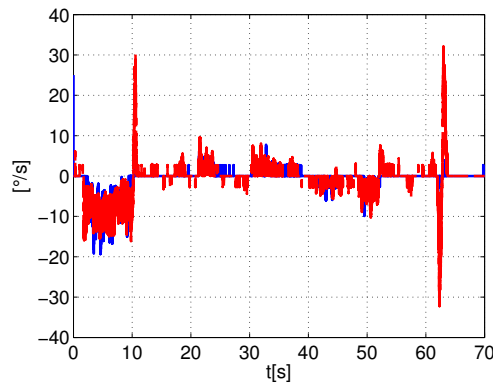
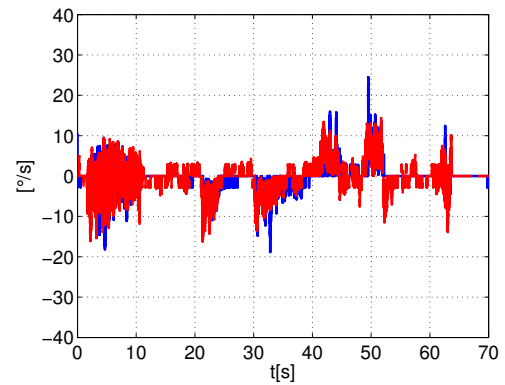


Figura 4.25: Velocidad observada $\dot{q}_i(t)$ (—) vs $\dot{q}_r(t - 0.5s)$ (- - - -) [$^\circ$].

(a) $\dot{q}_{r1}(t)$ (---) vs $\dot{q}_{11}(t - 0.5s)$ (—)(b) $\dot{q}_{r2}(t)$ (---) vs $\dot{q}_{12}(t - 0.5s)$ (—)(c) $\dot{q}_{r3}(t)$ (---) vs $\dot{q}_{13}(t - 0.5s)$ (—)**Figura 4.26:** Velocidad observada $\dot{q}_r(t)$ (---) vs $\dot{q}_1(t - 0.5s)$ (—) [°].

En las Figuras 4.25 y 4.26 se observan las velocidades obtenidas mediante el observador diseñado. Sin embargo se observa que la velocidad para la segunda articulación de ambos robots tiene picos, aproximadamente a los diez y sesenta segundos, estos tiempos son precisamente cuando se toca por primera vez la superficie y cuando se trata de atravesar la superficie. En esos instantes el controlador intenta corregir rápidamente el crecimiento del error.

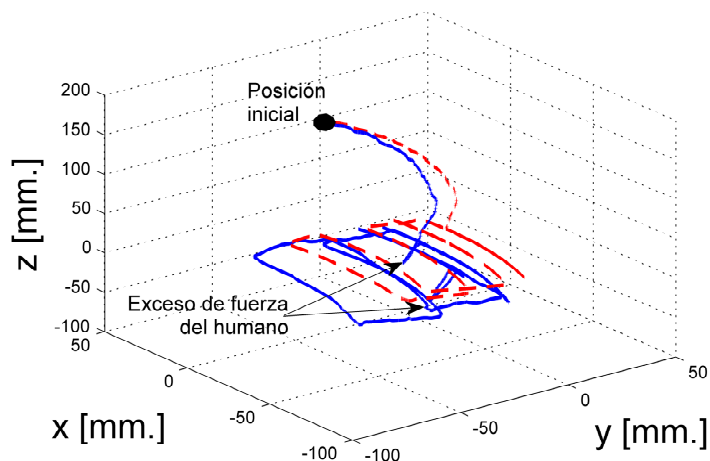
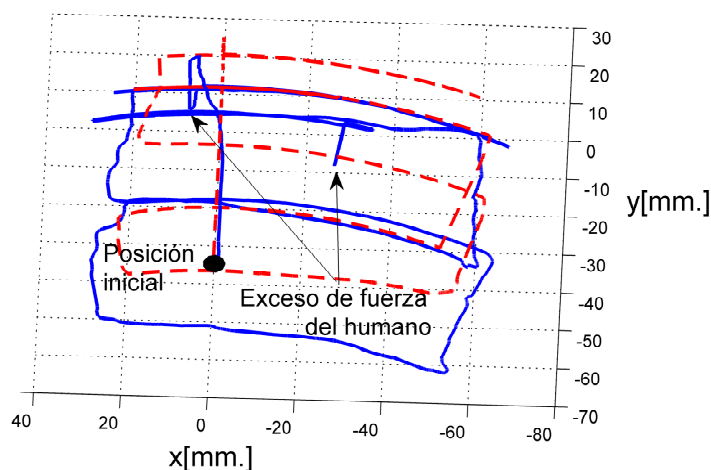
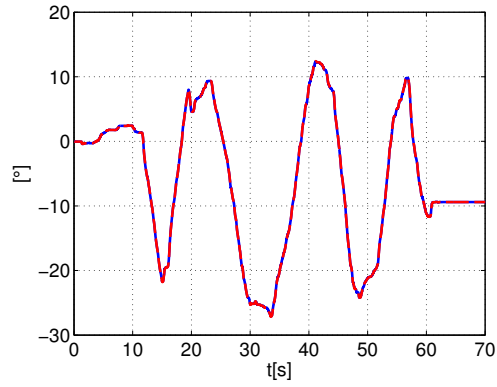
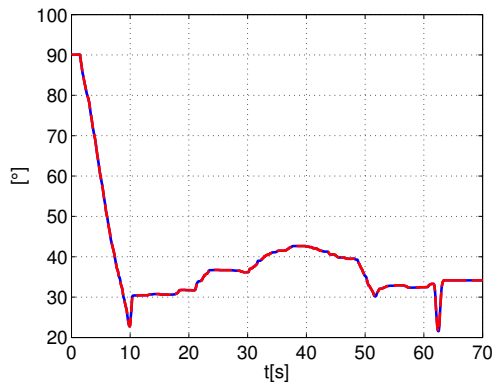
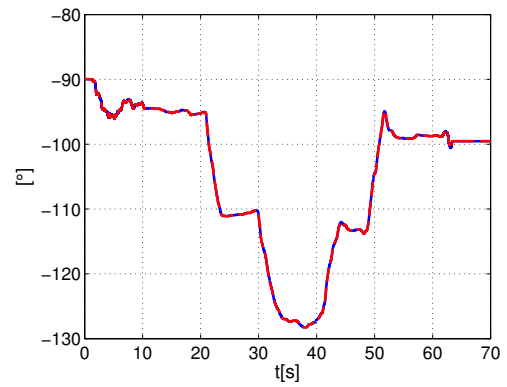
(a) *Vista isométrica*(b) *Vista superior*

Figura 4.27: Superficie virtual $p_1(t)$ (—) vs superficie real $p_r(t)$ (- - -) con presencia de retardo simétrico

En las Figura 4.27 se puede observar la superficie percibida por el humano $p_1(t)$ (—), a pesar de que el humano intenta atravesar la superficie como se aprecia mejor en a), el inciso a) de la Proposición 3.2.1 aún se mantiene. Se puede observar además que el controlador corrige este exceso de fuerza por parte del humano, teniendo la sensación de telepresencia como lo afirma el inciso *iv.*, bajo este escenario se sigue manteniendo la estabilidad del sistema.

(a) $q_{11}(t)$ (—) vs $\hat{q}_{r1}(t)$ (- - -)(b) $q_{12}(t)$ (—) vs $\hat{q}_{r2}(t)$ (- - -)(c) $q_{13}(t)$ (—) vs $\hat{q}_{r3}(t)$ (- - -)**Figura 4.28:** Posiciones medidas $q_i(t)$ (—) vs posiciones estimadas $\hat{q}_r(t)$ (- - -) [$^{\circ}$].

4. RESULTADOS EXPERIMENTALES

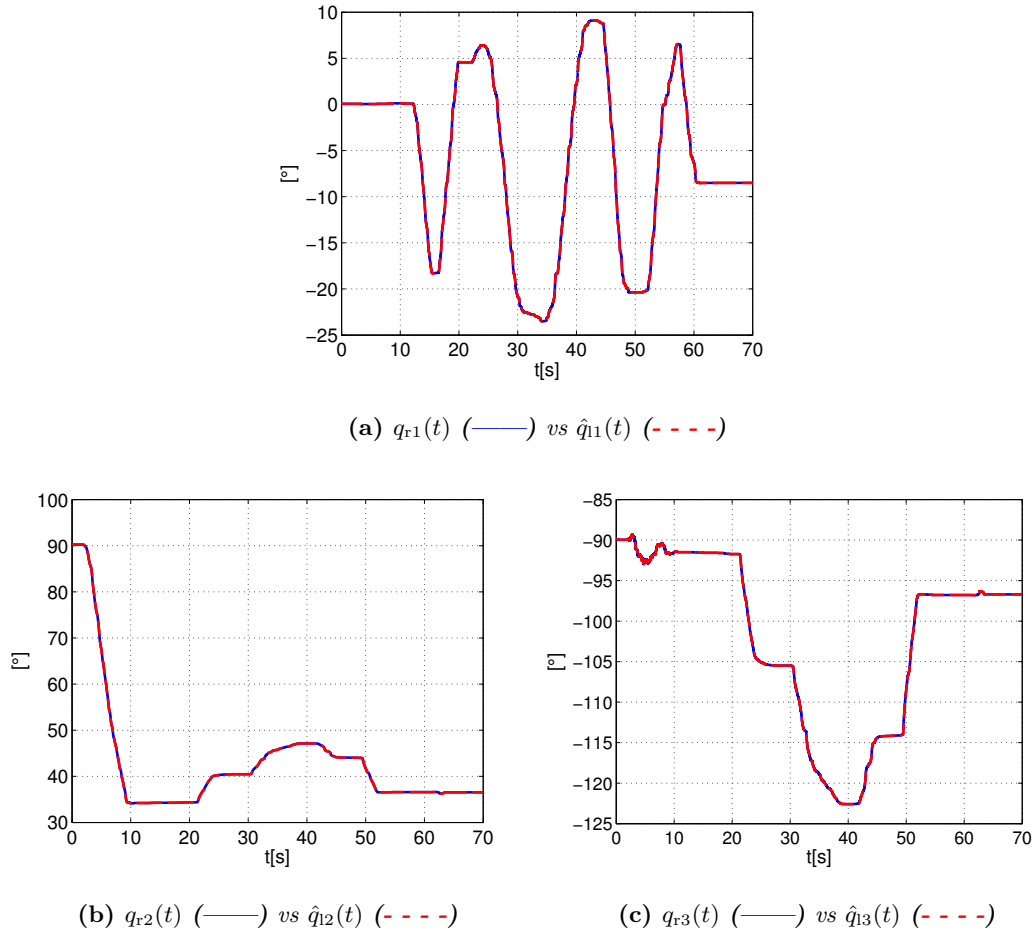


Figura 4.29: Posiciones medidas $q_r(t)$ (—) vs posiciones estimadas $\hat{q}_i(t)$ (- - -) [°].

Con el fin de mostrar que el observador funciona de buena manera, en las Figuras 4.28 y 4.29 se muestran las posiciones estimadas.

4.2.3. Sin cancelación del par gravitacional

El siguiente experimento se realizó para un retardo $T_1 = 0.3s$ y $T_r = 0.7s$, con la intención de mostrar que el esquema que se propone funciona sin el conocimiento de ningún parámetro del modelo de ambos robots. Este experimento se trató de reproducir de manera similar al segundo mostrado en este trabajo. El esquema propuesto, en la parte del controlador involucra la cancelación de la gravedad, sin embargo hay que mencionar que se propuso hacerlo de esta manera, únicamente para comodidad del operador humano. A continuación se presentan un experimento, en el cual se ha omitido la cancelación del par gravitacional en ambos manipuladores.

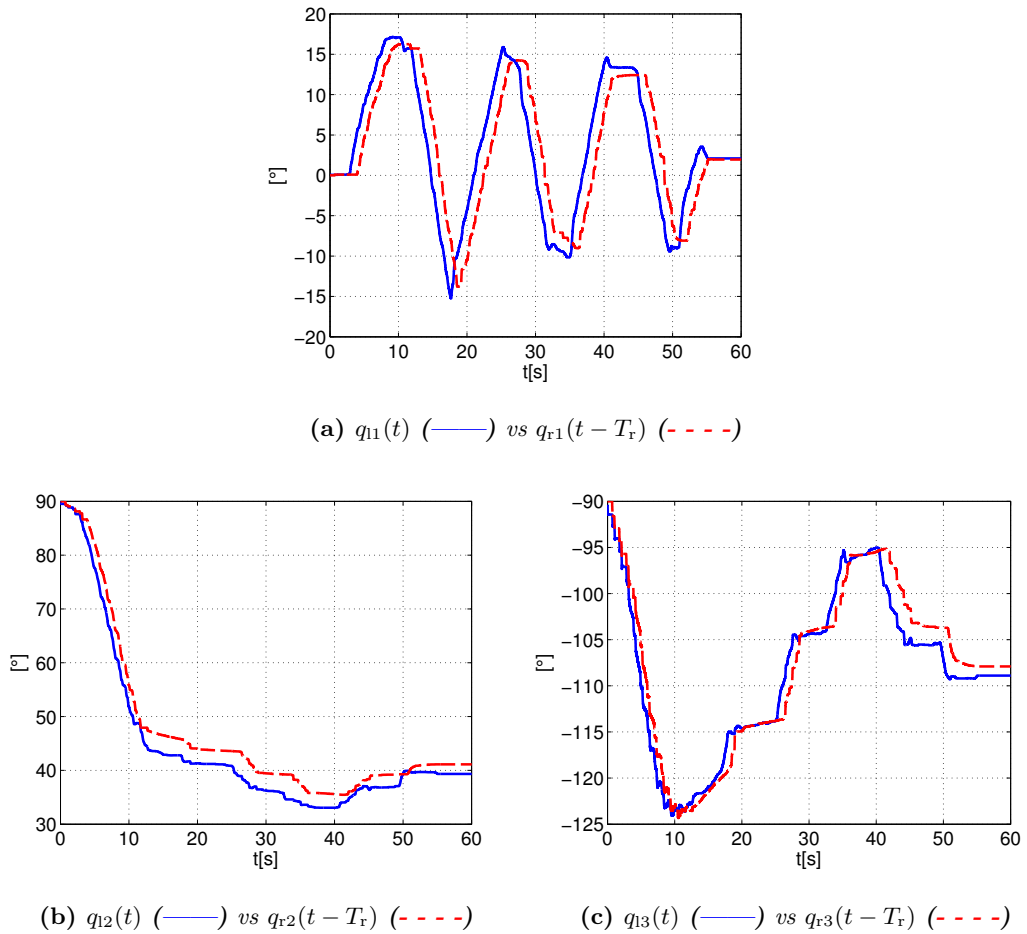
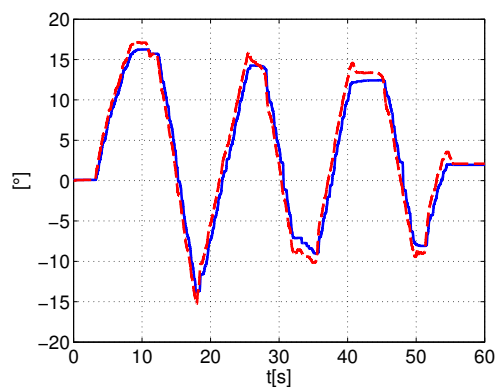
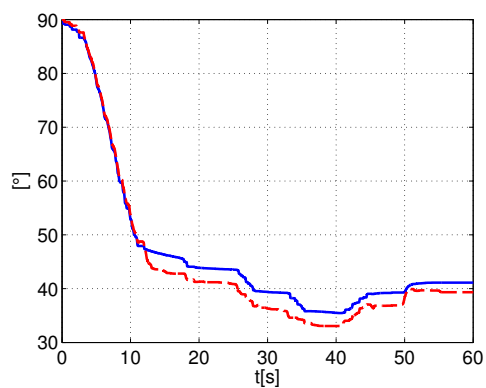


Figura 4.30: Posiciones sin cancelación del par gravitacional y con presencia de retardo asimétrico $q_1(t)$ (—) vs $q_r(t - 0.7s)$ (- - -) [°].

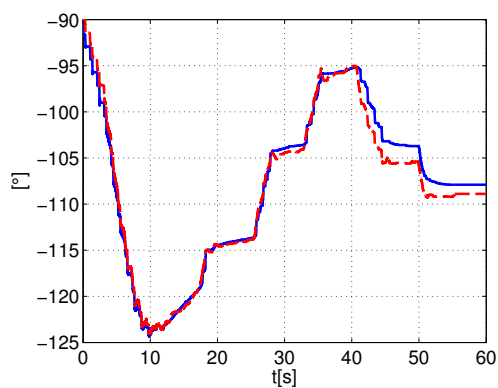
4. RESULTADOS EXPERIMENTALES



(a) $q_{r1}(t)$ (---) vs $q_{11}(t - T_1)$ (—)



(b) $q_{r2}(t)$ (---) vs $q_{12}(t - T_1)$ (—)



(c) $q_{r3}(t)$ (---) vs $q_{13}(t - T_1)$ (—)

Figura 4.31: Posiciones sin cancelación del par gravitacional y con presencia de retardo asimétrico $q_r(t)$ (---) vs $q_1(t - 0.3s)$ (—) [°].

En las Figuras 4.30 y 4.31 se observan los perfiles de posición de ambos robots, las trayectorias fueron similares a las Figuras 4.8 y 4.9.

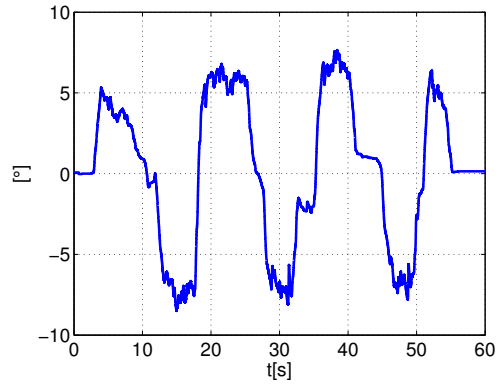
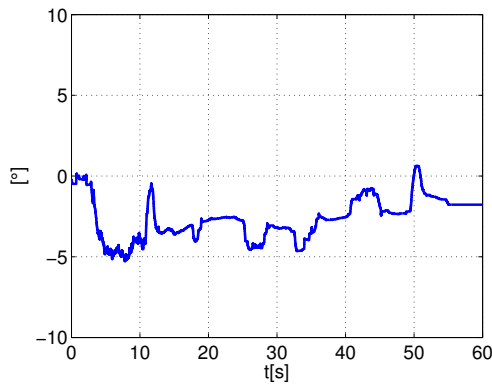
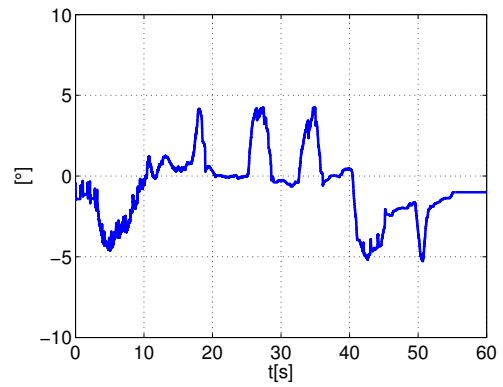
(a) e_{q1} (—)(b) e_{q2} (—)(c) e_{q3} (—)

Figura 4.32: Errores de seguimiento sin cancelación del par gravitacional y con presencia de retardo asimétrico $e(t) = q_l(t) - q_r(t - 0.7s)$

4. RESULTADOS EXPERIMENTALES

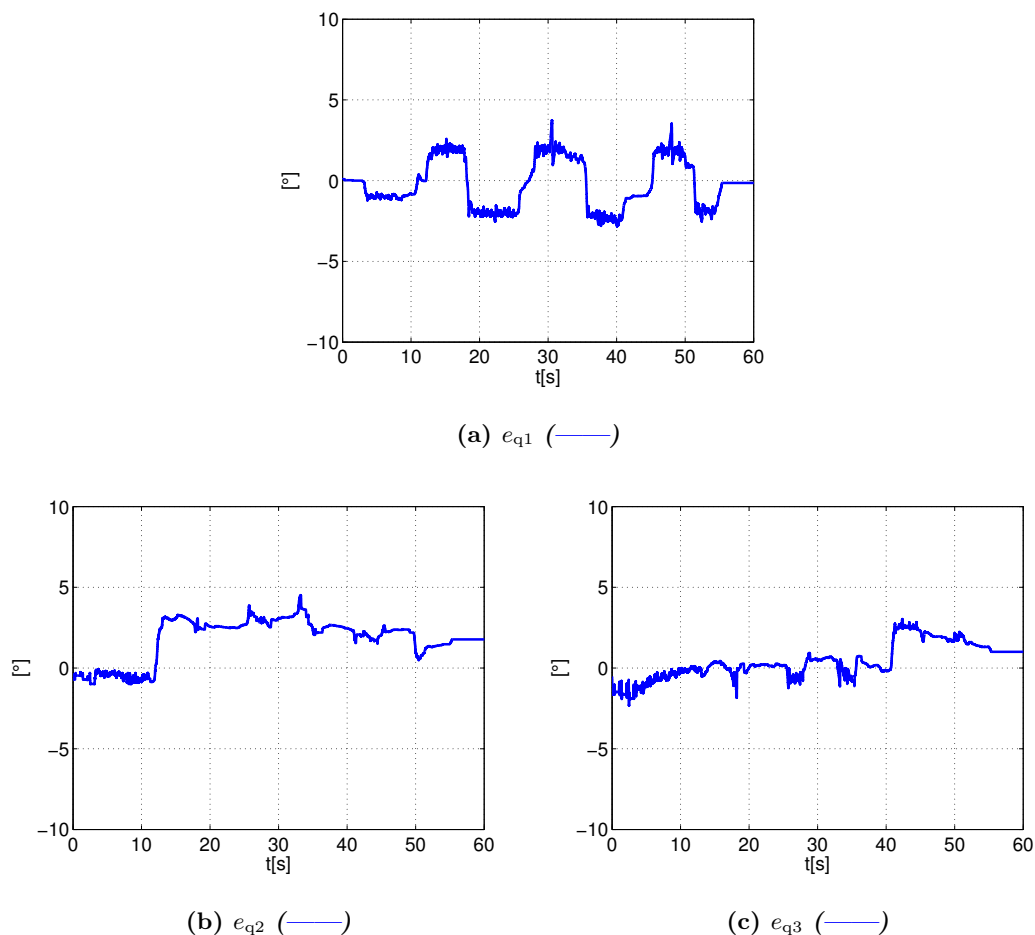
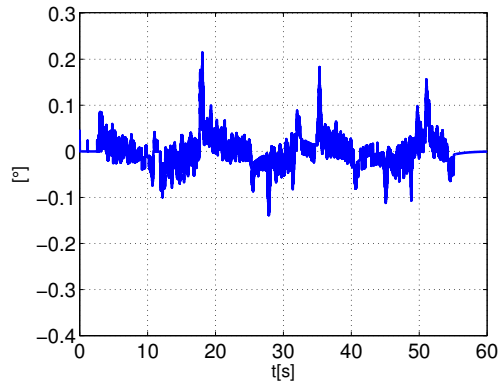


Figura 4.33: Errores de seguimiento sin cancelación del par gravitacional y con presencia de retardo asimétrico $e(t) = q_r(t) - q_l(t - 0.3s)$

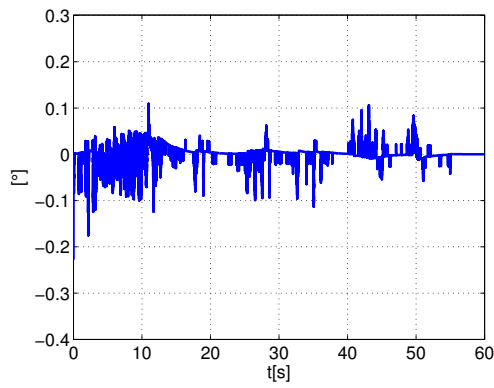
En las Figuras 4.32 y 4.33 se muestra que el esquema funciona como se esperaba y no necesita conocer el modelo de los robots, este es un gran aporte. En experimentos anteriores se canceló el vector de gravedad, simplemente se hizo por comodidad del operador humano. Esta capacidad de no cancelar la gravedad tiene sus limitaciones, este experimento se llevó a cabo gracias a que los robots son pequeños y no tienen dimensiones que dificulten su manipulación.

Comparando las Figuras 4.32 y 4.33 con las Figuras 4.8 y 4.9, se puede observar que las amplitudes de los errores de seguimientos son menores cuando no se compensa la gravedad. Aún falta demostrar hasta qué punto la selección de ganancias del controlador podría considerar esta compensación de gravedad como una perturbación tratando de

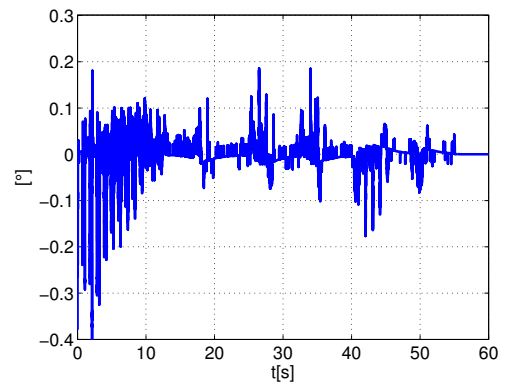
atenuarla, tratar de demostrarlo puede ser una tarea difícil.



(a) z_{11} (—)



(b) z_{12} (—)



(c) z_{13} (—)

Figura 4.34: Errores de observación para retardo asimétrico $z_1 = q_1(t) - \hat{q}_1(t)$

4. RESULTADOS EXPERIMENTALES

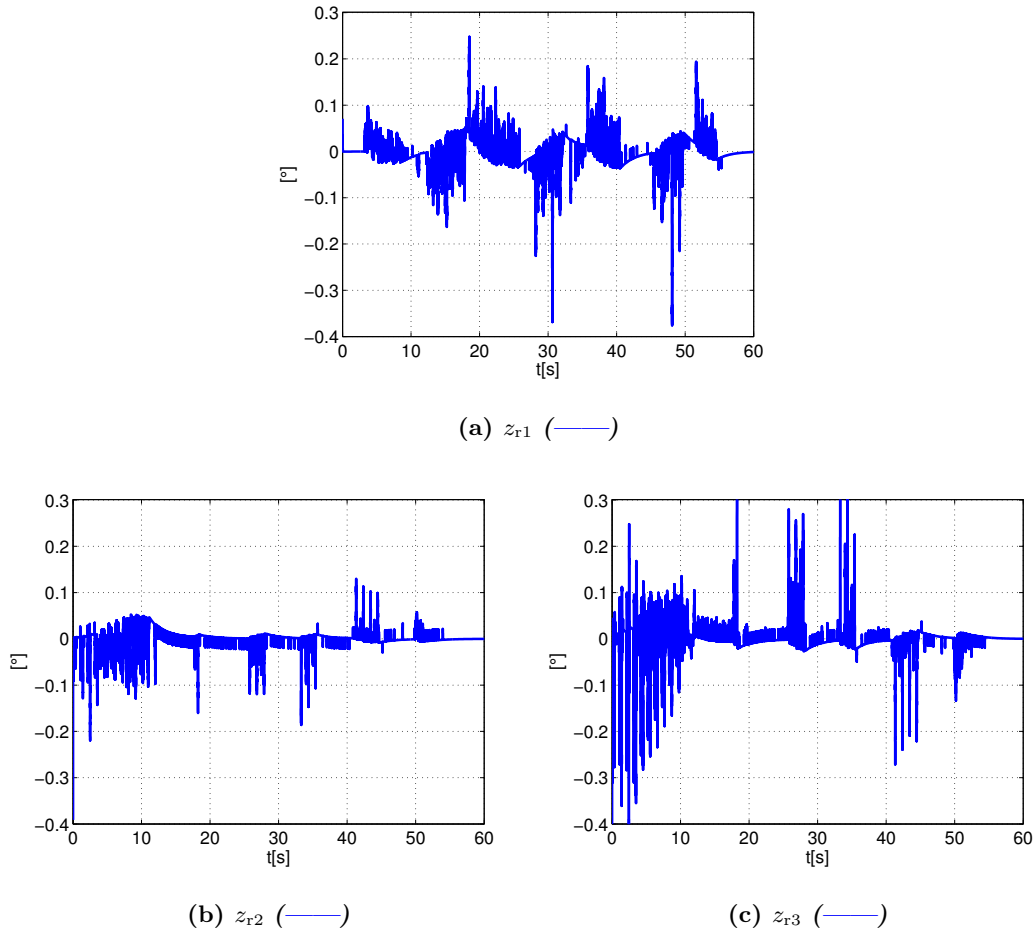
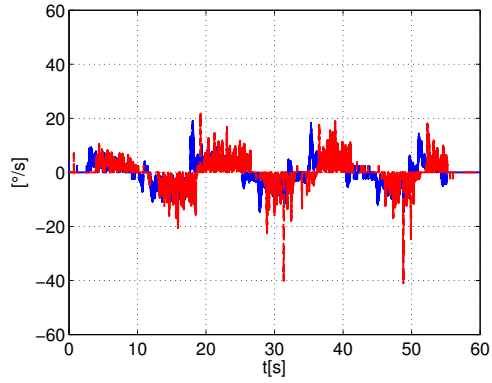
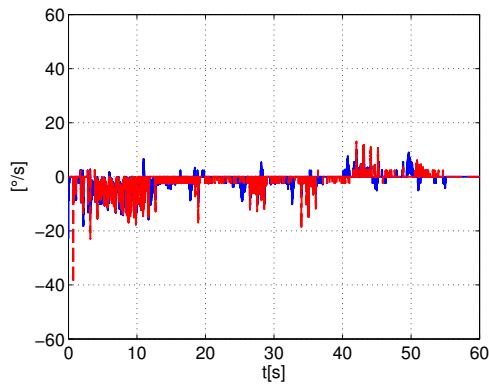


Figura 4.35: Errores de observación para retardo asimétrico $z_r = q_r(t) - \hat{q}_r(t)$

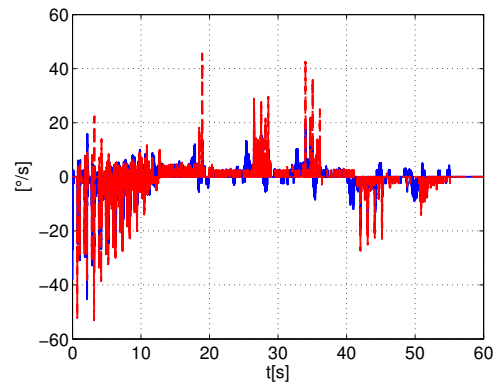
En las Figuras 4.34 y 4.35 se puede observar que las amplitudes de los errores de observación se mantienen pequeños. Como era de esperarse el observador funciona como se esperaba.



(a) $\dot{q}_{11}(t)$ (—) vs $\dot{q}_{r1}(t - 0.7s)$ (- - -)



(b) $\dot{q}_{12}(t)$ (—) vs $\dot{q}_{r2}(t - 0.7s)$ (- - -)



(c) $\dot{q}_{13}(t)$ (—) vs $\dot{q}_{r3}(t - 0.7s)$ (- - -)

Figura 4.36: Velocidad observada $\dot{q}_i(t)$ (—) vs $\dot{q}_r(t - 0.7s)$ (- - -) [$^\circ$].

4. RESULTADOS EXPERIMENTALES

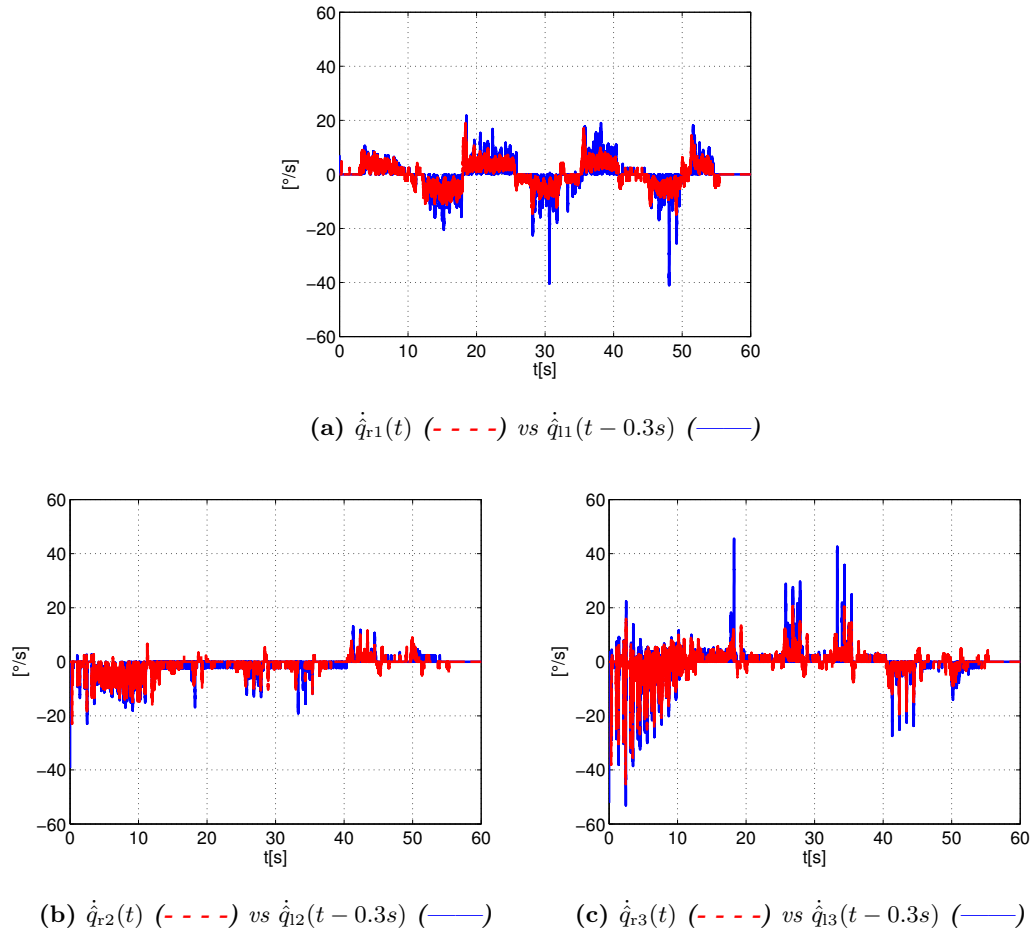


Figura 4.37: Velocidad observada $\dot{q}_r(t)$ (---) vs $\dot{q}_i(t - 0.3s)$ (—) [°].

En las Figuras (4.36) y (4.37) se observan las velocidades estimadas con el observador.

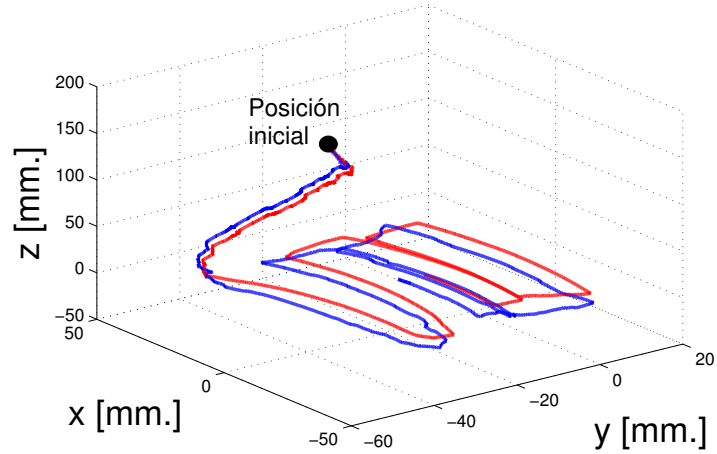
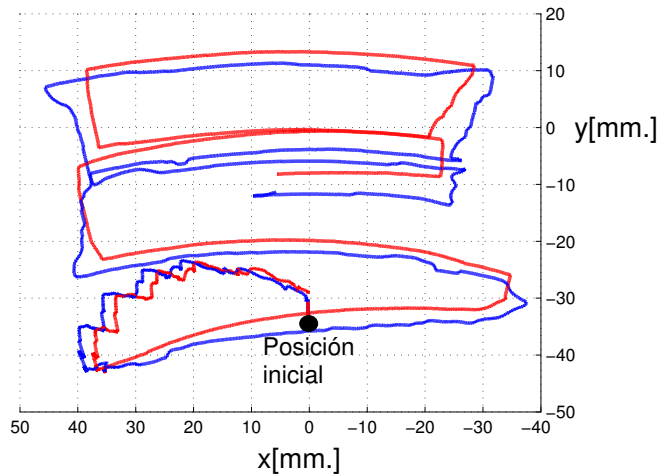
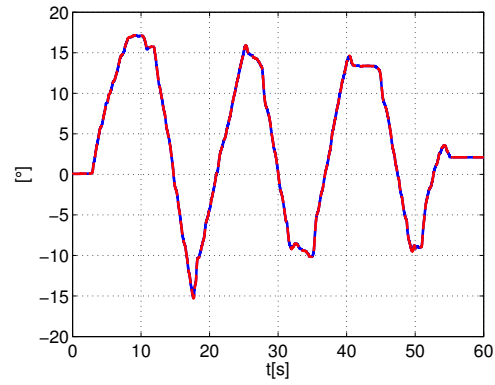
(a) *Vista isométrica*(b) *Vista superior*

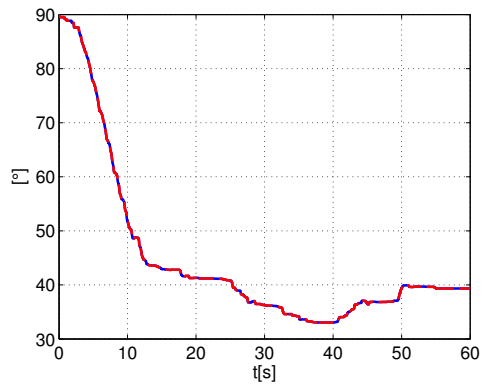
Figura 4.38: Superficie virtual $p_1(t)$ (—) vs superficie real $p_r(t)$ (---) sin cancelación del par gravitacional y con retardo asimétrico

En la Figura 4.32 se observa la superficie percibida por el humano, nuevamente se obtiene una superficie muy parecida a la restricción física. Sin embargo se observa un plano que presenta ciertos pasos algo abruptos, esta trayectoria puede ser debido al retardo y al comportamiento del humano.

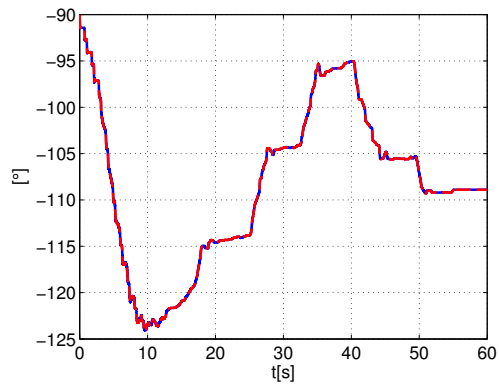
4. RESULTADOS EXPERIMENTALES



(a) $q_{11}(t)$ (—) vs $\hat{q}_{r1}(t)$ (- - -)



(b) $q_{12}(t)$ (—) vs $\hat{q}_{r2}(t)$ (- - -)



(c) $q_{13}(t)$ (—) vs $\hat{q}_{r3}(t)$ (- - -)

Figura 4.39: Posiciones medidas $q_i(t)$ (—) vs posiciones estimadas $\hat{q}_r(t)$ (- - -) [$^{\circ}$].

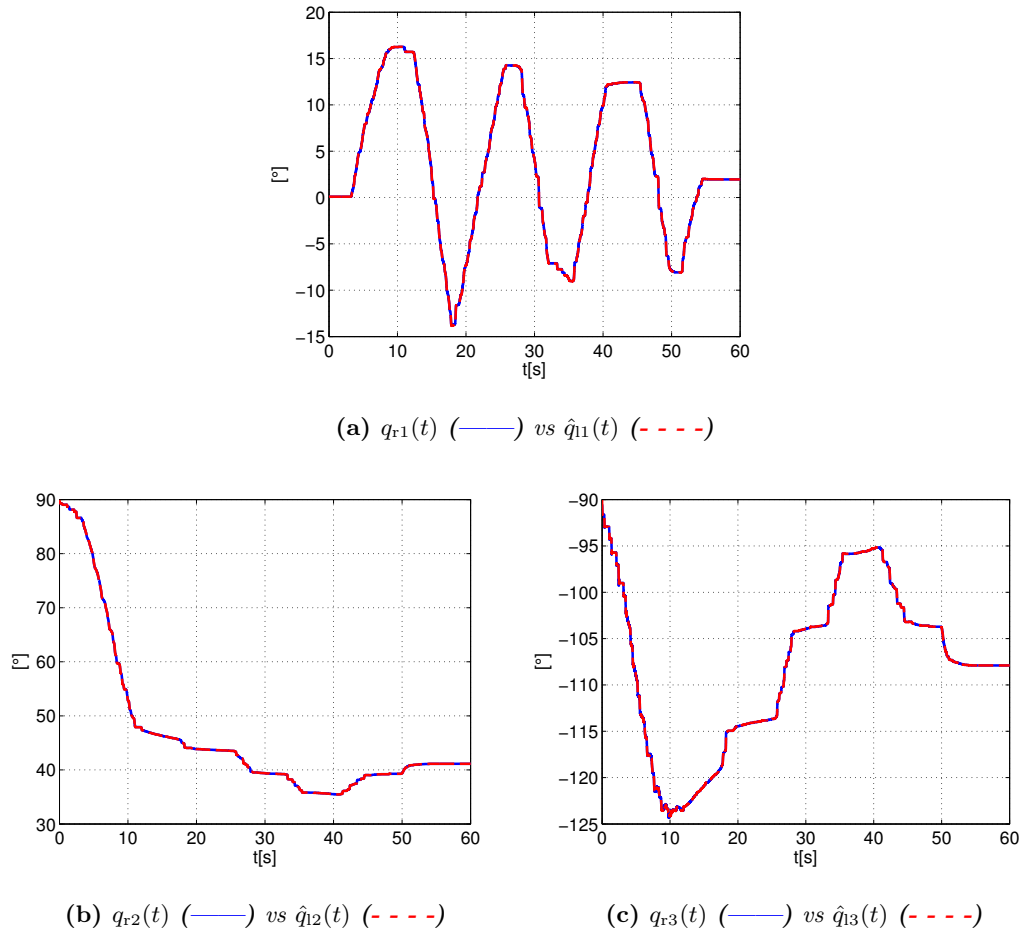


Figura 4.40: Posiciones medidas $q_r(t)$ (—) vs posiciones estimadas $\hat{q}_i(t)$ (- - - -) [$^\circ$].

Las Figuras 4.39 y 4.40 muestran las posiciones estimadas obtenidas por los observadores, se observa que el observador funciona como se esperaba.

Este experimento demuestra que en ambos casos, con o sin el conocimiento del vector de pares gravitacionales el esquema propuesto funciona bien para robots con dinámica como la de los *Geomagic Touch*.

Aún queda pendiente demostrar sí, efectivamente el esquema funciona mejor, como aparentemente lo fue cuando no se compensa la gravedad.

Conclusiones

Este trabajo presenta un innovador y práctico esquema controlador-observador para sistemas teleoperados bilateralmente diseñado para lidiar con la presencia de retardos constantes en el canal de comunicación. El esquema propuesto resuelve el problema de incertidumbres en el modelo; no es necesario conocerlo, por lo que es robusto. Además, no necesita del conocimiento de las velocidades articulares, ya que también incluye un observador que obtiene una muy buena estimación de las velocidades reales. Con base en experimentación se demostró que el esquema propuesto es muy práctico además de tener muy buen desempeño. Al usar un observador lineal para obtener una estimación de las velocidades articulares, se evita el uso de tacómetros, además del ruido presente en estas señales.

En el desarrollo teórico, se usó el análisis de Lyapunov para demostrar el acotamiento de la solución de la ecuación de estado de la dinámica de los errores de seguimiento y observación. Esta solución tiene propiedades de acotamiento uniforme y acotamiento final que fueron demostradas en el Capítulo 3 sin conocer la solución explícita de la ecuación de estado. Cabe mencionar que gran parte del desarrollo teórico parte de la suposición de que puede expresar de manera simplificada el comportamiento dinámico del operador humano. También se demostró, mediante experimentación que tanto los errores de observación, como los errores de seguimiento entre ambos robots son arbitrariamente pequeños, cabe resaltar que los retardos fueron simétricos y asimétricos. Es importante resaltar que desde un punto de vista teórico no es necesario la cancelación del vector de gravedad, los experimentos muestran que el esquema funciona bien y no necesita conocer el modelo de los robots, este es un gran aporte. En el último experimento no se canceló el vector de gravedad por comodidad del operador humano. Esto se puede hacer dado que los robots son pequeños y no tienen dimensiones que dificulten su manipulación. Una desventaja del esquema propuesto es lograr una sintonización adecuada para las ganancias. Resulta difícil determinar hasta qué punto pueden ser atenuadas por la inyección de amortiguamiento o por la compensación de gravedad en el robot maestro, ya que esta inyección o compensaciones pueden verse como perturbaciones a compensar; además de lidiar con un comportamiento del humano que pudiese

5. CONCLUSIONES

ser distinto al contemplado. Quizás este comportamiento pueda verse como una fuerza externa variante en el tiempo o quizás con algún retardo, que resultaría difícil de compensar por parte del contralor del robot remoto.

Todos los resultados obtenidos sugieren que las ganancias están en función del tiempo de retardo. Esto se comprobó comparando el caso con y sin retardo y verificar que el sistema será inestable con un retardo sustancial.

5.1. Trabajo futuro

Como posible continuación de este trabajo, podría realizarse un análisis considerando retardos variantes en el tiempo y pérdida de información por el canal de comunicación.

El algoritmo funciona utilizando las posiciones, a partir de esta información el operador humano tiene la sensación de tocar alguna superficie, sin embargo no es posible afirmar que la fuerza que éste aplique es la misma (por lo menos escalada) sobre la superficie, por ello se plantea la posibilidad de agregar al esquema un término que permita realizar control de fuerza. También se contempla la posibilidad de usar un sensor de fuerza o buscar la inclusión de un observador de fuerza.

Código/Manuales/Publicaciones

A.1. Robot *Geomagic Touch de 3D Systems*

Para la validación experimental del esquema propuesto se usarán dos robots *Geomagic Touch de 3D Systems* como maestro y esclavo. El robot *Geomagic Touch* está diseñado especialmente para realizar tareas *hápticas*; sin embargo, gracias a su código abierto puede ser usado para distintos fines.



Figura A.1: Robot *Geomagic Touch de 3D Systems*

El robot *Geomagic Touch de 3D Systems* de la Figura A.1 cuenta con seis articulaciones, de las cuales, sólo las primeras tres están actuadas mediante motores de corriente directa. Para las articulaciones q_1 , q_2 y q_3 se puede obtener la posición de manera muy precisa por medio de *encoders* digitales, mientras que para q_4 , q_5 y q_6 la posición es obtenida con potenciómetros.

A.1.1. Modelo Cinemático

En esta sección, se aborda el modelo cinemático del robot *Geomagic Touch*. El modelo cinemático de un robot tiene como objetivo describir el movimiento de un robot sin considerar las fuerzas y torques causantes del movimiento.

A.1.1.1. Cinemática Directa

La cinemática directa es empleada para determinar la posición y orientación del efector final dados los valores de las variables articulares del robot. En este trabajo solo se considera la posición del centro de la muñeca $P_c(x_c, y_c, z_c)$ del robot *Geomagic Touch*. Debido a la construcción mecánica del robot, la convención de Denavit Hartenberg no puede ser aplicada directamente, esto es debido a que la tercera articulación se mueve de manera independiente, el ángulo de giro θ_3 usualmente se mide desde x_2 hacia x_3 siendo z_2 el eje de giro, sin embargo basta con una pequeña modificación para poder medir el ángulo de giro θ_3 de acuerdo a esta convención. Así $\theta_3 = q_3 - \theta_2 - 90^\circ$, donde q_3 es la posición angular medida desde un eje vertical en sentido anti horario, con esta transformación se pueden obtener los modelos cinemáticos siguiendo la convención de Denavit Hartenberg.

La Cinemática Directa respecto a la Figura A.2 del centro de la muñeca $P_c(x_c, y_c, z_c)$ respecto al sistema base, esta dada por:

$$x_c = (a_2 \cos(\theta_2) + a_3 \cos(\theta_2 + \theta_3)) \cos \theta_1, \quad (\text{A.1})$$

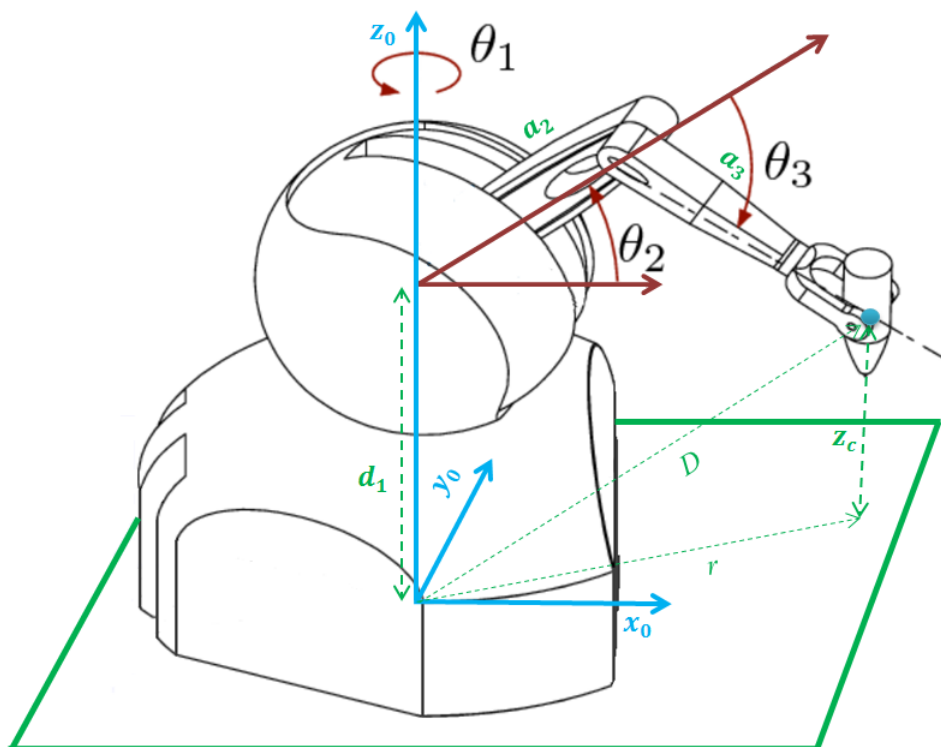
$$y_c = (a_2 \cos(\theta_2) + a_3 \cos(\theta_2 + \theta_3)) \sin \theta_1, \quad (\text{A.2})$$

$$z_c = (a_2 \sin(\theta_2) + a_3 \sin(\theta_2 + \theta_3)) + d_1, \quad (\text{A.3})$$

A.1.1.2. Cinemática Inversa

La cinemática inversa es empleada para determinar los valores de las variables articulares dada la posición y la orientación del efector final. En general, es más complicado que en la cinemática directa, ya que no siempre puede obtenerse una solución analítica cerrada a este problema. Sin embargo, se puede obtener una solución para el problema cinemático inverso mediante un enfoque geométrico.

Dado el centro de la muñeca del robot *Geomagic touch* $p_c(x_c, y_c, z_c)$, las soluciones para q_1 , q_2 y q_3 de acuerdo a la Figura A.2 son respectivamente


Figura A.2: Vista isométrica

$$\theta_1 = \tan^{-1} \left(\frac{y_c}{x_c} \right), \quad (\text{A.4})$$

$$\theta_2 = \tan^{-1} \left(\frac{s}{r} \right) - \tan^{-1} \left(\frac{a_3 \cos(\theta_3)}{a_2 + a_3 \sin(\theta_3)} \right), \quad (\text{A.5})$$

$$\theta_3 = \tan^{-1} \left(\frac{\sqrt{1 - D^2}}{D} \right), \quad (\text{A.6})$$

donde $\cos(q_3) = \frac{s^2 + r^2 - a_2^2 - a_3^2}{2a_2a_3} =: D$, $s = z_c - d_1$ y $r^2 = x_c^2 + y_c^2$.

Para determinar las ecuaciones cinemáticas y dinámicas del robot *Geomagic Touch*, es necesario conocer algunas características importantes, como son: resolución que ofrecen los *encoders*, el espacio de trabajo, etc. Estos datos (ofrecidos por el fabricante) se muestran en la Tabla A.1.

Tabla A.1: Especificaciones técnicas del robot *Geomagic Touch de 3D Systems*

Especificaciones	Descripción
Peso	3 Libras (1360.78 g)
Grados de Libertad	6
Rigidez	Eje X 1.26 N / mm Eje Y 2.31 N / mm Eje Z 1.02 N / mm
Espacio de trabajo	160 w x 120 h x 70 d [mm]
Comunicación	Puerto Usb o puerto Ethernet conforme a RJ45
Fuerza máxima nominal	0.75 lbf (3.3 N)
Resolución nominal de la posición	0.055 [mm]

Con ayuda del paquete de librerías *Open Haptics* proporcionadas por *3D Systems*, es posible obtener el número de pulsos de los *encoders* en las seis articulaciones del robot y disponer libremente de los torques a los motores.

Tabla A.2: Relación giro-cuentas de encoders *Geomagic Touch de 3D Systems*

	θ_1	θ_2	$\theta_3 = f_{min}(\theta_2)$	$\theta_3 = f_{max}(\theta_2)$
Cuentas	-2435 a 2380	-60 a -4420	-841 a 3180	2290 a 4901
Grados	-57 a 57	0 a 105	$-20.8 < \theta_3 < 73.5$	$-49.9 < \theta_3 < 11.25$

A partir de la Tabla A.2, se puede observar que el rango para la tercera articulación está limitado por la posición de la segunda, esto es debido al diseño mecánico del robot.

A.2. Pares de gravedad

El esquema propuesto está diseñado para evitar en lo posible, el conocimiento del modelo dinámico del robot, solo basta con conocer el vector de pares gravitacionales. El cálculo en línea de este vector $\mathbf{g}_m(\mathbf{q}_m)$ depende de la posición articular y del conocimiento de parámetros de la estructura del robot, es decir;

$$\mathbf{g}_m(\mathbf{q}_m) = \begin{bmatrix} 0 \\ m_2 g_0 \cos(\theta_2) + m_3 g_0 \cos(\theta_2 + \theta_3) \\ m_3 g_0 \cos(\theta_2 + \theta_3) \end{bmatrix}, \quad (\text{A.7})$$

Los parámetros m_2 y m_3 fueron determinados experimentalmente usando el esquema de control adaptable basado en pasividad de [Slotine & Li 1986].

$$\mathbf{H}_m(\mathbf{q}_m)\dot{\mathbf{r}} + \mathbf{C}_m(\mathbf{q}_m, \dot{\mathbf{q}}_m)\mathbf{r} + \mathbf{K}_m\mathbf{r} + \mathbf{g}_m(\mathbf{q}_m) = \mathbf{Y}_m\tilde{\boldsymbol{\theta}}_m, \quad (\text{A.8})$$

donde

$$\begin{aligned} \mathbf{v} &= \dot{\mathbf{q}}_m^d - \Lambda\tilde{\mathbf{q}}_m \\ \mathbf{a} &= \dot{\mathbf{v}} = \ddot{\mathbf{q}}_m^d - \Lambda\dot{\tilde{\mathbf{q}}}_m \\ \mathbf{r} &= \dot{\mathbf{q}}_m - \mathbf{v} = \dot{\tilde{\mathbf{q}}}_m + \Lambda\tilde{\mathbf{q}}_m \end{aligned}$$

Para determinar la estimación de $\boldsymbol{\theta}_m$ se usó la ley de adaptación gradiente,

$$\dot{\tilde{\boldsymbol{\theta}}}_m = -\Gamma^{-1}\mathbf{Y}^T(\mathbf{q}_m, \dot{\mathbf{q}}_m, \mathbf{a}, \mathbf{v})\mathbf{r}, \quad (\text{A.9})$$

Los valores obtenidos para las masas de los eslabones 2 y 3 respectivamente son $m_2 = 0.0085$ kg., $m_3 = 0.01$ kg., y donde $g_0 = 9.79$ es la constante de gravedad en Ciudad Universitaria, México.

A.3. Código Visual Studio C++

```

1 // PhantomDlg.cpp : implementation file
2 #include "stdafx.h"
3 #include "Phantom.h"
4 #include "PhantomDlg.h"
5 #include <HD/hd.h> // Library of haptic device
6 #include <HDU/hduError.h>
7 #include <HDU/hduVector.h>
8 // User includes
9 #include <math.h> // Math operations
10 #include "mmsystem.h" // Multimedia timer
11 #include "analysis.h" // Matrix operations, etc.
12 #include <NIDAQmx.h> // Sensor de fuerza
13 // CPhantomDlg dialog
14 // User definitions
15 #define pi 3.1415926535
16 #define MAX_GRAF_ROWS 60000
17 HHD hHDm;
18 HHD hHDs;
```

```

19 bool initialized = false , schedulerStarted = false ; // user flags
20 double taum[3] = {0.0,0.0,0.0};
21 double taus[3] = {0.0,0.0,0.0}; // Input arrays
22 double qm[3] = {0.0};
23 double qs[3] = {0.0}; // Position arrays
24 const double T = 0.001; // Sample time
25 const int n = 3; // Number of joints
26 HDSchedulerHandle servoLoopHandle;
27 bool iCHome = true , homeCompletedFlag = true , iCControl = true ,
    controlCompletedFlag = true; // User flags
28 \*Variables para teleoperacion*\
29 bool iCTele=true , teleoperacionCompletedFlag = true;
30 MMRESULT homeTimerID , controlTimerID , teleoperacionTimerID;
31 double grafi[MAX_GRAF_ROWS][31] = {0.0};
32 const int grafSkip = 0; // Number of time intervals to discard for the
    output file
33 int indx = 0;
34 const double angle_final_effector = 15.0*pi/180.0;
35 const double a2 = 0.145 , a3 = sqrt(.135*.135 + .04*.04 - 2.0*0.135*.04*cos
    (pi-angle_final_effector));
36 const double gamma_final_effector = asin(0.04*sin(pi-angle_final_effector)
    /a3);
37 const double T1=0.3;
38 const double T2=0.7;
39 int CONTT1=0;
40 int CONTT2=0;
41 double BARqm[n] = {0.0};
42 double BARESTqm[n] = {0.0};
43 double BARESTDqm[n] = {0.0};
44 double BARqs[n] = {0.0};
45 double BARESTqs[n] = {0.0};
46 double BARESTDqs[n] = {0.0};
47 double ALMBARqm[MAX_GRAF_ROWS][n]={0.0};
48 double ALMBARqs[MAX_GRAF_ROWS][n]={0.0};
49 double ALMBARESTDqm[MAX_GRAF_ROWS][n]={0.0};
50 double ALMBARESTDqs[MAX_GRAF_ROWS][n]={0.0};

```



```

51 double ALMBARESTqm[MAX_GRAF_ROWS][n]={0.0};
52 double ALMBARESTqs[MAX_GRAF_ROWS][n]={0.0};
53 // Force sensor init
54 TaskHandle taskHandle=0;
55 int error=0;
56 float64 data;
57 #define DAQmxErrChk(functionCall) if( DAQmxFailed(error=(functionCall)) )
    goto Error; else
58 double lambda;
59 CPhantomDlg::CPhantomDlg(CWnd* pParent /*=NULL*/)
60 : CDialog(CPhantomDlg::IDD, pParent)
61 {
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME); }
62 void CPhantomDlg::DoDataExchange(CDataExchange* pDX)
63 {
    CDialog::DoDataExchange(pDX);
64 DDX_Control(pDX, IDC_EDIT1, m_statusTextBox);
65 DDX_Control(pDX, IDC_READENCODERS, m_readEncoders);
66 DDX_Control(pDX, IDC_ENCODER1, m_encoderBox1);
67 DDX_Control(pDX, IDC_ENCODER2, m_encoderBox2);
68 DDX_Control(pDX, IDC_ENCODER3, m_encoderBox3);
69 DDX_Control(pDX, IDC_FORCESENSOR, m_ForceValueBox);
70 DDX_Control(pDX, IDC_PROGRESS1, m_ForceBarCtrl);
71 DDX_Control(pDX, IDC_ENCODER4, m_encoderBox4);
72 DDX_Control(pDX, IDC_ENCODER5, m_encoderBox5);
73 DDX_Control(pDX, IDC_ENCODER6, m_encoderBox6);
74 DDX_Control(pDX, IDC_TIEMPO, m_TIEMPO);
75 DDX_Control(pDX, IDC_TIEMPO, m_TIEMPO);
76 }
77 BEGIN_MESSAGE_MAP(CPhantomDlg, CDialog)
78 ON_WM_PAINT()
79 ON_WM_QUERYDRAGICON()
80 ON_WM_CLOSE()
81 ON_BN_CLICKED(IDC_INITIALIZE, &CPhantomDlg::OnBnClickedInitialize)
82 ON_BN_CLICKED(IDC_CALIBRATION, &CPhantomDlg::OnBnClickedCalibration)
83 ON_BN_CLICKED(IDC_READENCODERS, &CPhantomDlg::OnBnClickedReadencoders)
84 ON_BN_CLICKED(IDC_HOME, &CPhantomDlg::OnBnClickedHome)
85 ON_BN_CLICKED(IDC_CONTROL, &CPhantomDlg::OnBnClickedControl)

```

```
86 ON_BN_CLICKED(IDC_Teleoperacion , &CPhantomDlg:: OnBnClickedTeleoperacion)
87 END_MESSAGE_MAP()
88 // CPhantomDlg message handlers
89 BOOL CPhantomDlg:: OnInitDialog ()
90 {
91     CDialog:: OnInitDialog ();
92     // Set the icon for this dialog. The framework does this
93     // automatically
94     // when the application's main window is not a dialog
95     SetIcon(m_hIcon, TRUE); // Set big icon
96     SetIcon(m_hIcon, FALSE); // Set small icon
97     // TODO: Add extra initialization here
98     return TRUE; // return TRUE unless you set the focus to a
99     // control
100 }
101 void CPhantomDlg:: OnPaint ()
102 {
103     if (IsIconic ())
104     {
105         CPaintDC dc(this); // device context for painting
106         SendMessage(WM_ICONERASEBKGND, reinterpret_cast <WPARAM>(dc.
107             GetSafeHdc()), 0);
108         // Center icon in client rectangle
109         int cxIcon = GetSystemMetrics(SM_CXICON);
110         int cyIcon = GetSystemMetrics(SM_CYICON);
111         CRect rect;
112         GetClientRect(&rect);
113         int x = (rect.Width() - cxIcon + 1) / 2;
114         int y = (rect.Height() - cyIcon + 1) / 2;
115         // Draw the icon
116         dc.DrawIcon(x, y, m_hIcon);
117     }
118     else
119     {
120         CDialog:: OnPaint ();
121     }
122 }
```

```

119 void CPhantomDlg:: OnClose() // When close button...
120 {
121     timeEndPeriod(1);
122     if(!iCHome)
123     {
124         timeKillEvent(homeTimerID);
125         homeCompletedFlag = iCHome = true;     }
126     if(!iCControl)
127     {
128         timeKillEvent(controlTimerID);
129         controlCompletedFlag = iCControl = true;     }
130     if(!iCTele)
131     {
132         timeKillEvent(teleoperacionTimerID);
133         teleoperacionCompletedFlag = iCTele = true;     }
134     if (initialized&&hdIsEnabled(HD_FORCE_OUTPUT))
135     hdDisable(HD_FORCE_OUTPUT);
136     hdUnschedule(servoLoopHandle);
137     if(schedulerStarted)
138     hdStopScheduler();
139     if(initialized)
140     {
141         hdDisableDevice(hHDm);
142         hdDisableDevice(hHDs);     }
143     FILE *outFile;
144     if(fopen_s(&outFile, "Data.m", "w")!=0){
145         MessageBox(_T("No se pudo crear el archivo para graficar")
146             );
147     }
148     else{
149         for(int i=0; i<indx; i++){
150             fprintf(outFile, "%f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f\n",
151                 grafi[i][0], grafi[i][1], grafi[i][2], grafi[i][3], grafi[i][4],
152                 grafi[i][5], grafi[i][6], grafi[i][7], grafi[i][8], grafi[i][9],
153                 grafi[i][10], grafi[i][11], grafi[i][12], grafi[i][13], grafi[i][14],
154                 grafi[i][15], grafi[i][16], grafi[i][17], grafi[i][18], grafi[i][19],
155                 grafi[i][20], grafi[i][21], grafi[i][22], grafi[i][23], grafi[i][24],
156                 grafi[i][25], grafi[i][26], grafi[i][27], grafi[i][28], grafi[i][29],
157                 grafi[i][30]);}

```

```

154         fclose ( outFile );
155     }
156     exit ( 0 );
157 }
158 // The system calls this function to obtain the cursor to display while
        the user drags
159 // the minimized window.
160 HCURSOR CPhantomDlg::OnQueryDragIcon ()
161 {     return static_cast <HCURSOR>(m_hIcon); }
162 HDCallbackCode HDCALLBACK CalibrationStatusCallback ( void * pUserData )
163 {     HDenum *pStatus = ( HDenum * ) pUserData;
164         hdBeginFrame ( hHDm );
165         hdUpdateCalibration ( HD_CALIBRATION_INKWELL );
166         *pStatus = hdCheckCalibration ();
167         hdEndFrame ( hHDm );
168         hdBeginFrame ( hHDs );
169         hdUpdateCalibration ( HD_CALIBRATION_INKWELL );
170         *pStatus = hdCheckCalibration ();
171         hdEndFrame ( hHDs );
172         return HD_CALLBACK_DONE;
173 }
174 typedef struct
175 {     hduVector3Dd position;
176 } DeviceStateStruct;
177 DeviceStateStruct state;
178 HDCallbackCode HDCALLBACK ServoLoopCallback ( void * pUserData )
179 {
180     DeviceStateStruct *pState = static_cast <DeviceStateStruct * >(
        pUserData );
181     HDdouble torque [ 3 ];
182     hdBeginFrame ( hHDm );
183     hdGetDoublev ( HD_CURRENT_JOINT_ANGLES, pState->position );
184     torque [ 0 ] = -1000.0*taum [ 0 ];
185     torque [ 1 ] = 1000.0*taum [ 1 ];
186     torque [ 2 ] = 1000.0*taum [ 2 ];
187     qm [ 0 ] = -state.position [ 0 ];

```

```

188     qm[1] = state.position[1];
189     qm[2] = state.position[2] - 0.5*pi - qm[1] - gamma_final_effector;
190     hdSetDoublev(HD_CURRENT_JOINT_TORQUE, torque);
191     hdBeginInitFrame(hHDs);
192     hdGetDoublev(HD_CURRENT_JOINT_ANGLES, pState->position);
193     torque[0] = -1000.0*taus[0];
194     torque[1] = 1000.0*taus[1];
195     torque[2] = 1000.0*taus[2];
196     qs[0] = -state.position[0];
197     qs[1] = state.position[1];
198     qs[2] = state.position[2] - 0.5*pi - qs[1] - gamma_final_effector;
199     hdSetDoublev(HD_CURRENT_JOINT_TORQUE, torque);
200     hdEndFrame(hHDm);
201     hdEndFrame(hHDs);
202     return HD_CALLBACK_CONTINUE;
203 }
204 void CPhantomDlg::OnBnClickedInitialize()
205 {
206     HDErrorInfo error;
207     HDstring MasterRobot = "Default Device";
208     HDstring SlaveRobot = "p2";
209     hHDm = hdInitDevice(MasterRobot);
210     if (HD_DEVICE_ERROR(error = hdGetError()))
211     {
212         MessageBox(_T("Master Device not Found!"));
213         return;
214     }
215     hHDs = hdInitDevice(SlaveRobot);
216     if (HD_DEVICE_ERROR(error = hdGetError()))
217     {
218         MessageBox(_T("Slave Device not Found!"));
219         return;
220     }
221     servoLoopHandle = hdScheduleAsynchronous(ServoLoopCallback, &state
222         , HD_MAX_SCHEDULER_PRIORITY);
223     hdMakeCurrentDevice(hHDm);
224     if (!hdIsEnabled(HD_FORCE_OUTPUT))
225     hdEnable(HD_FORCE_OUTPUT);
226     if (HD_DEVICE_ERROR(error = hdGetError()))
227     MessageBox(_T("Force output enable error!"));

```

```
223     hdMakeCurrentDevice(hHDs);
224     if (!hdIsEnabled(HD_FORCEOUTPUT))
225     hdEnable(HD_FORCEOUTPUT);
226     if (HD_DEVICE_ERROR(error = hdGetError()))
227     MessageBox(_T("Force output enable error!"));
228     if(!schedulerStarted)
229     {
230         hdStartScheduler();
231         schedulerStarted = true;
232         Sleep(1500);
233     }
234     if (HD_DEVICE_ERROR(error = hdGetError()))
235     {
236         MessageBox(_T("Servo loop initialization error"));
237         hdDisableDevice(hHDm);
238         hdDisableDevice(hHDs);
239         exit(-1);
240     }
241     else
242     {
243         initialized = true;
244         m_statusTextBox.SetWindowTextW(_T("*** Phantom Robot
245             initialized ***"));
246     }
247     timeBeginPeriod(1);
248 }
249 void CPhantomDlg::OnBnClickedCalibration()
250 {
251     if(initialized)
252     {
253         int supportedCalibrationStyles;
254         int calibrationStyle;
255         HD_ErrorInfo error;
256         hdGetIntegerv(HD_CALIBRATION_STYLE, &supportedCalibrationStyles);
257         if (supportedCalibrationStyles & HD_CALIBRATION_INKWELL)
258             calibrationStyle = HD_CALIBRATION_INKWELL;
259         else
260         {
261             MessageBox(_T(" Sorry, no ink-well calibration available "));
262             return;
263         }
264         if (HD_DEVICE_ERROR(error = hdGetError()))
265         m_statusTextBox.SetWindowTextW(_T("*** Failed to start the
266             scheduler***"));
```

```
257     HDenum status;
258     hdScheduleSynchronous(CalibrationStatusCallback, &status,
259                           HD_DEFAULT_SCHEDULER_PRIORITY);
260     if (status == HD_CALIBRATION_NEEDS_MANUAL_INPUT)
261         MessageBox(_T(" Please put the device into the ink-well "));
262     else
263     {
264         m_statusTextBox.SetWindowTextW(_T("*** Calibration done
265         ***"));
266     }
267     return;
268 }
269 else
270 {
271     MessageBox(_T(" Please initialize first the Phantom device
272     "));
273 }
274 }
275 void CPhantomDlg::OnBnClickedReadencoders()
276 {
277     CString text[6];
278     text[0].Format(L"%.3f", qm[0]*180.0/pi);
279     text[1].Format(L"%.3f", qm[1]*180.0/pi);
280     text[2].Format(L"%.3f", qm[2]*180.0/pi);
281     text[3].Format(L"%.3f", qs[0]*180.0/pi);
282     text[4].Format(L"%.3f", qs[1]*180.0/pi);
283     text[5].Format(L"%.3f", qs[2]*180.0/pi);
284     m_encoderBox1.SetWindowTextW(text[0]);
285     m_encoderBox2.SetWindowTextW(text[1]);
286     m_encoderBox3.SetWindowTextW(text[2]);
287     m_encoderBox4.SetWindowTextW(text[3]);
288     m_encoderBox5.SetWindowTextW(text[4]);
289     m_encoderBox6.SetWindowTextW(text[5]);
290     return;
291 }
292 }
293 void CPhantomDlg::OnBnClickedHome()
294 {
295     if (!iCHome)
```

```
290     {           timeKillEvent(homeTimerID);
291               homeCompletedFlag = iCHome = true;           }
292     if(!iCControl)
293     {           timeKillEvent(controlTimerID);   }
294     homeTimerID = timeSetEvent( T*1000, 0, HomeTimerProc, 0,
                TIME_PERIODIC);           // Home timer initialization
295 }
296 void CALLBACK CPhantomDlg::HomeTimerProc(UINT uID, UINT uMsg, DWORD dwUser
    , DWORD dw1, DWORD dw2)
297 {static double tini = 0.0;
298 static double tf = 1.0;
299 static double emi[n] = {0.0}, em_1[n] = {0.0}, esi[n] = {0.0}, es_1[n] =
    {0.0};
300 static double bm0[n] = {0.0}, bm3[n] = {0.0}, bm4[n] = {0.0}, bm5[n] =
    {0.0};
301 static double bs0[n] = {0.0}, bs3[n] = {0.0}, bs4[n] = {0.0}, bs5[n] =
    {0.0};
302 const double qmdf[n] = {0.0*pi/180.0, 90.0*pi/180.0, -90.0*pi/180.0};
303 const double qsdf[n] = {0.0*pi/180.0, 90.0*pi/180.0, -90.0*pi/180.0};
304 double t = 0.0, qmd[n] = {0.0}, em[n] = {0.0}, emp[n] = {0.0}, qsd[n] =
    {0.0}, es[n] = {0.0}, esp[n] = {0.0};
305 double c1 = 0.0, c2 = 0.0, c3 = 0.0, c23 = 0.0, s1 = 0.0, s2 = 0.0, s3 =
    0.0, s23 = 0.0;
306 const double kpm[n] = {3.0,3.0,3.0}, kim[n] = {1.0,1.0,1.0}, kdm[n] =
    {0.02,0.02,0.02};
307 const double kps[n] = {3.0,3.0,3.0}, kis[n] = {1.0,1.0,1.0}, kds[n] =
    {0.02,0.02,0.02};
308 double xd[3] = {0.0}, x[3] = {0.0};
309 const double xdf[3] = {0.15,0.0,-0.16};
310 double D = 0.0;
311 CPhantomDlg *pMainWnd = (CPhantomDlg *) AfxGetApp()->m_pMainWnd;
312 if(iCHome)
313     {
314         tini = timeGetTime();
315         for(int i=0; i<n; i++)
316     {
```



```
317     bm0[i] = qm[i];
318     bm3[i] = 10*(qmdf[i]-bm0[i])/(tf*tf*tf);
319     bm4[i] = -15*(qmdf[i]-bm0[i])/(tf*tf*tf*tf);
320     bm5[i] = 6*(qmdf[i]-bm0[i])/(tf*tf*tf*tf*tf);
321     bs0[i] = qs[i];
322     bs3[i] = 10*(qsdf[i]-bs0[i])/(tf*tf*tf);
323     bs4[i] = -15*(qsdf[i]-bs0[i])/(tf*tf*tf*tf);
324     bs5[i] = 6*(qsdf[i]-bs0[i])/(tf*tf*tf*tf*tf);
325 }
326 pMainWnd->m_statusTextBox.SetWindowTextW(_T("*** Reaching home
        position ***"));
327 }
328 t = (timeGetTime() - tini)/1000.0;
329 for(int i=0; i<n; i++)
330 {
331     if(t<=tf)
332     {
333         qmd[i] = bm0[i] + bm3[i]*t*t*t + bm4[i]*t*t*t*t + bm5[i]*t*t*t*t*t
        ;
334         qsd[i] = bs0[i] + bs3[i]*t*t*t + bs4[i]*t*t*t*t + bs5[i]*t*t*t*t*t
        ;
335     }
336     else
337     {
338         qmd[i] = qmdf[i];
339         qsd[i] = qsdf[i];
340     }
341 }
342 for(int i=0; i<n; i++)
343 {
344     em[i] = qm[i] - qmd[i];
345     es[i] = qs[i] - qsd[i];
346     if(iCHome)
347     {
348         es_1[i] = es[i];
349         em_1[i] = em[i];
```

```
350     }
351     esp[i] = (es[i] - es_1[i])/T;
352     emp[i] = (em[i] - em_1[i])/T;
353     taum[i] = -kpm[i]*em[i] - kim[i]*emi[i] - kdm[i]*emp[i];
354     taus[i] = -kps[i]*es[i] - kis[i]*esi[i] - kds[i]*esp[i];
355     }
356     /* if (iCHome)
357     indx = 0;
358     grafi[indx][0] = t;
359     grafi[indx][1] = q[0]*180.0/pi;
360     grafi[indx][2] = q[1]*180.0/pi;
361     grafi[indx][3] = q[2]*180.0/pi;
362     grafi[indx][4] = qd[0]*180.0/pi;
363     grafi[indx][5] = qd[1]*180.0/pi;
364     grafi[indx][6] = qd[2]*180.0/pi;
365     grafi[indx][7] = taus[0];
366     grafi[indx][8] = taus[1];
367     grafi[indx][9] = taus[2];
368     indx++;
369     */
370     for (int i=0; i<n; i++)
371     {
372     emi[i] += em[i]*T;
373     em_1[i] = em[i];
374     esi[i] += es[i]*T;
375     es_1[i] = es[i];
376     }
377     if (t>t f&&homeCompletedFlag)
378     {
379     pMainWnd->m_statusTextBox.SetWindowText(_T("*** Home position ***
380         "));
381     homeCompletedFlag = false;
382     }
383     if (iCHome)
384     iCHome = false;
385     return;
```

```

385 }
386 void CPhantomDlg:: OnBnClickedControl ()
387 {
388     if (!iCHome)
389     {
390         timeKillEvent (homeTimerID);
391     }
392     if (!iCTele)
393     {
394         timeKillEvent (teleoperacionTimerID);
395     }
396     if (!iCControl)
397     {
398         timeKillEvent (controlTimerID);
399         controlCompletedFlag = iCControl = true;
400     }
401     homeTimerID = timeSetEvent ( T*1000, 0, ControlTimerProc, 0,
402         TIME_PERIODIC);
403 }
404 void CPhantomDlg:: OnBnClickedTeleoperacion ()
405 {
406     if (!iCHome)
407     {
408         timeKillEvent (homeTimerID);
409     }
410     if (!iCControl)
411     {
412         timeKillEvent (controlTimerID);
413     }
414     if (!iCTele)
415     {
416         timeKillEvent (teleoperacionTimerID);
417         teleoperacionCompletedFlag= iCTele = true;
418     }
419     homeTimerID = timeSetEvent ( T*1000, 0, TeleoperacionTimer, 0,
420         TIME_PERIODIC);
421 }
422 void CALLBACK CPhantomDlg:: TeleoperacionTimer (UINT uID, UINT uMsg, DWORD
423     dwUser, DWORD dw1, DWORD dw2)
424 {
425     //OBSERVADOR Y CONTROL EN MOVIMIENTO LIBRE
426     static double tini = 0.0;
427     static int grafFlag = 0;
428     const double g0 = 9.78;
429     const int p = 5;
430     CString TIEMPO;
431     const double tf = 1.0;
432     double t = 0.0;
433     const double Kpm[n] = {7,8,6};

```

```

418     const double Kam[n] = {.01,.03,.02};
419     const double Kps[n] = {.4,.9,.8};
420     const double Kdm[n] = {30,30,30};
421     const double Kds[n] = {20,50,20};
422     const double Kbetas[n] = {0.0000005,0.0000009,0.0000007};
423     const double Kbetam[n] = {0.0000001,0.0000001,0.0000001};
424     const double DELTAzm[n] = {1,1,.5};
425     const double DELTAzs[n] = {1,1,1};
426     const double DELTAs[n] = {10,12,10};
427     const double DELTAm[n] = {10,12,10};
428     const double Kgamas[n] = {.4,.5,.25};
429     const double Kgamam[n] = {.2,.2,.15};
430     double SIGMAS[n] = {0,0,0};
431     double SIGMAM[n] = {0,0,0};
432     static double INTm[n] = {0,0,0};
433     static double INTs[n] = {0,0,0};
434     static double dSIGMAS[n] = {0,0,0};
435     static double dSIGMAM[n] = {0,0,0};
436     double signos =0;
437     double signom =0;
438     double gm[n] ={0,0,0};
439     double c2 = 0.0, c23 = 0.0;
440     double zm[n] = {0,0,0};
441     double zs[n] = {0,0,0};
442     double HATqm[n] = {0,0,0};
443     double BARdqm[n] = {0,0,0};
444     double ESTdqom[n] = {0,0,0};
445     double ESTdqos[n] = {0,0,0};
446     double dqos[n]= {0,0,0};
447     double dqom[n]= {0,0,0};
448     double Ss[n] = {0,0,0};
449     double Sm[n] = {0,0,0};
450     double dqrs[n] = {0,0,0};
451     double dqrm[n] = {0,0,0};
452     double Sos[n] = {0,0,0};
453     double Som[n] = {0,0,0};

```

```

454     double BARdqs[n] = {0,0,0};
455     double ESTdqm[n] = {0,0,0};
456     double ESTdqs[n] = {0,0,0};
457     static double ESTqm[n]; // = {0.0*pi/180.0, 90.0*pi/180.0, -90.0*pi
        /180.0};
458     static double ESTqs[n]; // = {0.0*pi/180.0, 90.0*pi/180.0, -90.0*pi
        /180.0};
459     CPhantomDlg *pMainWnd = (CPhantomDlg *) AfxGetApp()->m_pMainWnd;
460     c2 = cos(qm[1]);
461     c23 = cos(qm[1]+qm[2]);
462     if (iCTele)
463     {
464         ESTqm[0] = 0.0*pi/180.0;
465         ESTqm[1] = 90.0*pi/180.0;
466         ESTqm[2] = -90.0*pi/180.0;
467         ESTqs[0] = 0.0*pi/180.0;
468         ESTqs[1] = 90.0*pi/180.0;
469         ESTqs[2] = -90.0*pi/180.0;
470         tini = timeGetTime();
471         pMainWnd->m_statusTextBox.SetWindowTextW(_T("*** Control
        in progress ***"));
472     }
473     t = (timeGetTime() - tini)/1000.0;
474     TIEMPO.Format(_T("%f"), t);
475     pMainWnd->m_TIEMPO.SetWindowTextW(TIEMPO);
476     gm[0]=0;
477     gm[1]=0.0085*g0*c2+0.01*g0*c23;
478     gm[2]=0.01*g0*c23;
479     //Guardamos todos los datos
480     for (int i=0; i<n; i++)
481     {
482         ALMBARqm[indx][i]=qm[i];
483         ALMBARqs[indx][i]=qs[i];
484         ALMBARESTdqm[indx][i]=ESTdqm[i];
485         ALMBARESTdqs[indx][i]=ESTdqs[i];
486         ALMBARESTqm[indx][i]=ESTqm[i];
487         ALMBARESTqs[indx][i]=ESTqs[i];
488     }

```

```

487     if (t<T1)
488     {
489         BARqm[0]=0.0* pi / 180.0;
490         BARqm[1]=90.0* pi / 180.0;
491         BARqm[2]= -90.0* pi / 180.0;
492         BARESTqm[0]=0;
493         BARESTqm[1]=90.0* pi / 180.0;
494         BARESTqm[2]= -90.0* pi / 180.0;
495         BARESTdqm[0]=0;
496         BARESTdqm[1]=0;
497         BARESTdqm[2]=0;
498     }
499     if (t<T2)
500     {
501         BARqs[0]=0;
502         BARqs[1]=90.0* pi / 180.0;
503         BARqs[2]= -90.0* pi / 180.0;
504         BARESTqs[0]=0;
505         BARESTqs[1]=90.0* pi / 180.0;
506         BARESTqs[2]= -90.0* pi / 180.0;
507         BARESTdqs[0]=0;
508         BARESTdqs[1]=0;
509         BARESTdqs[2]=0;
510     }
511     if (t>=T1)
512     {
513         CONTT1++;
514         for(int i=0; i<n; i++)
515         {
516             BARqm [ i]=ALMBARqm[CONTT1] [ i ];
517             BARESTdqm [ i]=ALMBARESTdqm[CONTT1] [ i ];
518             BARESTqm [ i]=ALMBARESTqm[CONTT1] [ i ];
519         }
520     }
521     if (t>=T2)
522     {
523         CONTT2++;
524         for(int i=0; i<n; i++)
525         {
526             BARqs [ i]=ALMBARqs[CONTT2] [ i ];
527             BARESTdqs [ i]=ALMBARESTdqs[CONTT2] [ i ];
528             BARESTqs [ i]=ALMBARESTqs[CONTT2] [ i ];

```

```

522         }
523     }
524     /* SIN RETARDO
525     for (int i=0; i<n; i++)
526     {
527         BARqm[i]=qm[i];
528         BARESTdqm[i]=ESTdqm[i];
529         BARqs[i]=qs[i];
530         BARESTdqs[i]=ESTdqs[i]; } */
531     for (int i=0; i<n; i++)
532     {
533         //Master Observer
534         zm[i] = qm[i]-ESTqm[i];
535         INTm[i]=INTm[i]+zm[i]*T;
536         ESTdqom[i]=Kdm[i]*DELTAzm[i]*INTm[i];
537         ESTdqm[i]=ESTdqom[i]+DELTAzm[i]*zm[i]+Kdm[i]*zm[i];
538         //Slave Observer
539         zs[i] = qs[i]-ESTqs[i];
540         INTs[i]=INTs[i]+zs[i]*T;
541         ESTdqos[i]=Kds[i]*DELTAzs[i]*INTs[i];
542         ESTdqs[i]=ESTdqos[i]+DELTAzs[i]*zs[i]+Kds[i]*zs[i];
543         //INTEGRO LAS VELOCIDADES ESTIMADAS PARA COMPARAR
544         ESTqm[i]=ESTqm[i]+ESTdqm[i]*T;
545         ESTqs[i]=ESTqs[i]+ESTdqs[i]*T;
546     }
547     for (int i=0; i<n; i++)
548     {
549         //Controller design
550         Ss[i]=ESTdqs[i]-BARESTdqm[i]+DELTAxs[i]*(ESTqs[i]-BARESTqm[i]);
551         Sm[i]=ESTdqm[i]-BARESTdqs[i]+DELTAxm[i]*(ESTqm[i]-BARESTqs[i]);
552         dqos[i]=ESTdqs[i]-DELTAzs[i]*zs[i];
553         dqom[i]=ESTdqm[i]-DELTAzm[i]*zm[i];
554         //Funcion signo se define como sign(Ss) = Ss/|Ss|=sign(Ss[i]);
555         if (Ss[i]>0)
556             signos =1;
557         if (Ss[i]<0)
558             signos =-1;
559         else

```

```

558     signos=0;
559     if (Ss[i]>0)
560     signom =1;
561     if(Ss[i]<0)
562     signom =1;
563     else
564     signom=0;
565     dSIGMAS[i]=Kbetas[i]*Ss[i]+signos;
566     SIGMAS[i]=SIGMAS[i]+dSIGMAS[i]*T;
567     dSIGMAM[i]=Kbetam[i]*Sm[i]+signom;
568     SIGMAM[i]=SIGMAM[i]+dSIGMAM[i]*T;
569     dqrs[i]=BARESTdqm[i]-DELTAxs[i]*(ESTqs[i]-BARESTqm[i])-Kgamas[i]*
        SIGMAS[i];
570     dqrm[i]=BARESTdqs[i]-DELTAxm[i]*(ESTqm[i]-BARESTqs[i])-Kgamam[i]*
        SIGMAM[i];
571     Sos[i]=dqos[i]-dqrs[i];
572     Som[i]=dqom[i]-dqrm[i];
573     }
574     for(int i=0; i<n; i++)
575     {
576     taus[i]=-0.1*Kps[i]*Sos[i];
577     taum[i]=-Kam[i]*ESTdqm[i]+gm[i]-0.005*Kpm[i]*Som[i];
578     }
579     if(t>tf&&teleoperacionCompletedFlag)
580     {
581     pMainWnd->m_statusTextBox.SetWindowTextW(_T("*** Control and
        Observer in progress ***"));
582     teleoperacionCompletedFlag = false;
583     }
584     if(iCTele)
585     indx = 0;
586     if(grafFlag==0)
587     {
588         grafi[indx][0] = t;
589         grafi[indx][1] = qm[0];
590         grafi[indx][2] = qm[1];
591         grafi[indx][3] = qm[2];

```



```
591         grafi[indx][4] = qs[0];
592         grafi[indx][5] = qs[1];
593         grafi[indx][6] = qs[2];
594         grafi[indx][7] = ESTdqm[0];
595         grafi[indx][8] = ESTdqm[1];
596         grafi[indx][9] = ESTdqm[2];
597         grafi[indx][10] = ESTdqs[0];
598         grafi[indx][11] = ESTdqs[1];
599         grafi[indx][12] = ESTdqs[2];
600         grafi[indx][13] = zm[0];
601         grafi[indx][14] = zm[1];
602         grafi[indx][15] = zm[2];
603         grafi[indx][16] = zs[0];
604         grafi[indx][17] = zs[1];
605         grafi[indx][18] = zs[2];
606         grafi[indx][19] = ESTqm[0];
607         grafi[indx][20] = ESTqm[1];
608         grafi[indx][21] = ESTqm[2];
609         grafi[indx][22] = ESTqs[0];
610         grafi[indx][23] = ESTqs[1];
611         grafi[indx][24] = ESTqs[2];
612         grafi[indx][25] = BARqm[0];
613         grafi[indx][26] = BARqm[1];
614         grafi[indx][27] = BARqm[2];
615         grafi[indx][28] = BARqs[0];
616         grafi[indx][29] = BARqs[1];
617         grafi[indx][30] = BARqs[2];
618         indx++;
619         grafFlag = grafSkip+1;
620     }
621     grafFlag--;
622     if(iCTele)
623     iCTele = false;
624     return;
625 }
```


Bibliografía

- [1] P. Hokayem and M. Spong, “Bilateral teleoperation: an historical survey,” *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006. 3
- [2] W. R. Ferrell, “Remote manipulation with transmission delay,” *IEEE Trans. Human Factors Electron.*, vol. HFE-6, pp. 24–32, Sept 1965. 4
- [3] W. R. Ferrell, “Delayed force feedback,” *IEEE Trans. Human Factors Electron.*, vol. HFE-8, pp. 449–455, Oct. 1966. 4
- [4] M. W. S. Robert J. Anderson, “Asymptotic stability for force reflecting teleoperation with delay,” *IEEE Int. Conf. Robotics and Automation, Scottsdale, AZ*, 1989. 4
- [5] M. W. S. Robert J. Anderson, “Bilateral control of teleoperators with time delay,” *ieeetr VOL 34 NO 5*, 1989. 4
- [6] N. G. Slotine Jean Jacques, “Asymptotic stability for force reflecting teleoperation with delay,” *IEEE Int. Conf. Robotics and Automation, Scottsdale, AZ.*, •. 4
- [7] J. J. E. Slotine and W. Li, “On the adaptive control of robot manipulators,” *International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987. 4
- [8] J. J. E. Slotine and W. Li, “Composite adaptive control of robot manipulators,” *Automatica*, vol. 25, no. 4, pp. 509–519, 1989. 4
- [9] R. O. N. E. B. Nikhil Chopra, Mark W. Spong, “On tracking performance in bilateral teleoperation,” *IEEE Transactions on Robotics, Vol. 22, No. 4*, Agosto 2006. 4
- [10] N. B. L. B. Emmanuel Nuño, Romeo Ortega, “A globally stable pd controller for bilateral teleoperators,” *IEEE Transactions on Robotics, Vol. 24, No. 3*, JUNE 2008. 4
- [11] C. P. G. H. J. Artigas, J. Vilanova, “Time domain passivity control-based telepresence with time delay,” *Proc. IEEE Int. Conf. Intell. Robots Syst.*, pp. 4205–4210, 2006. 4

- [12] R. L. Nikhil Chopra, Mark W. Spong, “Time domain passivity control-based telepresence with time delay,” *ELSEVIER, Automatica 44 (2008) 2142–2148*, 23 Dic 2007. 5
- [13] M. A. Arteaga-Pérez, A. M. Castillo-Sánchez, and V. Parra-Vega, “Cartesian control of robots without dynamic model and observer design,” *Automatica*, vol. 42, pp. 473–480, 2006. 5, 16, 19, 25
- [14] M. A. Arteaga-Pérez and R. Kelly, “Robot control without velocity measurements: New theory and experimental results,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 297–308, 2004. 5
- [15] E. Nuño, R. Ortega, N. Barabanov, and L. Basañez, “A globally stable PD controller for bilateral teleoperators,” *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 753–758, 2008. 10, 17
- [16] S. Arimoto, Y. H. Liu, and T. Naniwa, “Model-based adaptive hybrid control for geometrically constrained robots,” in *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, pp. 618–623, 1993. 11
- [17] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, Florida, USA: CRC Press, 1994. 11
- [18] V. P.-V. Marco A. Arteagaa, Adrián Castillo-Sánchez, “Cartesian control of robots without dynamic model and observer design,” *ELSEVIER, Automatica 42 (2006) 473 – 480*, 8 Noviembre 2005. 16
- [19] A. Rodríguez-Ángeles, M. A. Arteaga-Pérez, R. Portillo-Vélez, and C. Cruz-Villar, “Transparent bilateral master-slave control based on virtual surfaces: Stability analysis and experimental results,” *International Journal of Robotics and Automation*, vol. 30, no. 2, pp. 128–139, 2015. 17
- [20] H. K. Khalil, *Nonlinear Systems, 3rd ed.* Upper Saddle River, New Jersey. U .S. A.: Prentice-Hall, 2002. 19