



UNIVERSIDAD
DE MÁLAGA

Tesis Doctoral

Modelado y regulación del flujo de datos sensoriales de robots teleoperados por Internet

Ana Gago Benítez
2017


Tesis Doctoral
Doctorado en Ingeniería Mecatrónica
Dpt. de Ingeniería de Sistemas y Automática
Universidad de Málaga





UNIVERSIDAD
DE MÁLAGA

AUTOR: Ana Gago Benítez

 <http://orcid.org/0000-0003-0171-0496>

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional:

<http://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>

Cualquier parte de esta obra se puede reproducir sin autorización pero con el reconocimiento y atribución de los autores.

No se puede hacer uso comercial de la obra y no se puede alterar, transformar o hacer obras derivadas.

Esta Tesis Doctoral está depositada en el Repositorio Institucional de la Universidad de Málaga (RIUMA): riuma.uma.es



UNIVERSIDAD DE MÁLAGA
DEPARTAMENTO DE
INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

El Dr. D. Juan Antonio Fernández Madrigal, director de la tesis titulada “Modelado y regulación del flujo de datos sensoriales de robots teleoperados por Internet” realizada por Dña. Ana Gago Benítez, certifica su idoneidad para la obtención del título de Doctor en Ingeniería Mecatrónica.

Málaga, 16 de Enero de 2017



Dr. D. Juan Antonio Fernández Madrigal



UNIVERSIDAD
DE MÁLAGA

Dept. de Ingeniería de Sistemas y Automática
Universidad de Málaga
Doctorado en Ingeniería Mecatrónica



Modelado y regulación del flujo de datos sensoriales de robots teleoperados por Internet

AUTORA: Ana Gago Benítez

DIRECTOR: Juan Antonio Fernández Madrigal

Tesis defendida el 14 de Julio de 2017

TRIBUNAL:

Javier González Jiménez (Universidad de Málaga)

José Luis Blanco Claraco (Universidad de Almería)

Eduardo Fernández Moral (INRIA Sophia-Antipolis)





UNIVERSIDAD
DE MÁLAGA

*A los que no están
y a los que están a punto de llegar*



UNIVERSIDAD
DE MÁLAGA

TABLA DE CONTENIDO

Resumen.....	iii
Agradecimientos.....	v
PARTE I.....	1
1 Introducción.....	3
1.1 Motivación	5
1.2 Marco de la tesis.....	6
1.3 Contribuciones	6
1.4 Estructura de la tesis.....	7
2 Modelado probabilístico de retardos temporales.....	9
2.1 Introducción y estado del arte	9
Técnicas existentes de modelado de secuencias de tiempo y su coste computacional	12
2.2 Modelado de los tiempos de transmisión mediante distribuciones de probabilidad marginal	13
Experimentos en escenarios reales	14
Distribuciones de probabilidad marginal más apropiadas para este tipo de retardos	20
Detección de regímenes con el método Confidence Interval.....	24
2.3 Modelado de retardos temporales con las distribuciones Log-normal y Log- logística 28	
Hipótesis Log-normal.....	28
Algoritmo basado en AD para modelar escenarios completos con distribución Log-normal.....	31
Hipótesis Log-logística.....	33
Algoritmos de modelado on-line de secuencias completas de retardos basados en la hipótesis Log-logística: stateless y state-based	37
3 Regulación del flujo de datos.....	41
3.1 Introducción y estado del arte	42
3.2 Regulación fina: variación de la cantidad de datos sensoriales	44
Experimentos con algoritmo de regulación fina	45
3.3 Regulación gruesa: activación / desactivación de sensores	47
4 Aplicación para el control remoto de robots móviles por Web.....	49
4.1 Descripción de la aplicación: control remoto de robot móviles por la Web.....	49
4.2 Experimentos.....	55



5 Resumen de publicaciones compendiadas.....59

5.1 Publicación A. "Marginal Probabilistic Modeling of the Delays in the Sensory Data Transmission of Networked Telerobots" 59

5.2 Publicación B. "A computationally efficient algorithm for modeling multi-regime delays in the sensory flow of networked telerobots" 59

5.3 Publicación C. "Log-logistic modelling of sensory flow delays in networked telerobots" 60

5.4 Publicación D. "Hierarchical Regulation of Sensor Data Transmission for Networked Telerobots" 60

6 Conclusiones y futuros trabajos.....61

7 Bibliografía.....63

PARTE II.....71

A. Marginal Probabilistic Modeling of the Delays in the Sensory Data Transmission of Networked Telerobots.....A73

B. A computationally efficient algorithm for modeling multi-regime delays in the sensory flow of networked telerobots.....B74

C. Log-logistic modelling of sensory flow delays in networked telerobots..C75

D. Hierarchical Regulation of Sensor Data Transmission for Networked Telerobots.....D7

6



RESUMEN

Esta tesis se centra en la problemática de teleoperar robots móviles a través de Internet de la manera más óptima posible, realizando una regulación que adapte la cantidad de datos sensoriales transmitidos desde el robot móvil hacia el teleoperador de forma automática.

Los robots teleoperados por Internet se controlan remotamente a través de redes y componentes de propósito general que son altamente heterogéneos y exhiben tiempos de respuesta aleatorios. A pesar de esto, su teleoperación correcta requiere de un flujo adecuado de información desde los sensores del robot hacia estaciones remotas. Para garantizar los requisitos de tiempo de tal flujo, se necesita: (i) una buena estimación probabilística *on-line* de los retardos en la transmisión sensorial, lo que para muchas aplicaciones modernas debe ser altamente eficiente desde un punto de vista computacional y por tanto basarse en modelos relativamente simples, como distribuciones de probabilidad marginales —por ejemplo cuando el sistema incluye una interfaz de cliente basada en web—; (ii) un detector *on-line* de no linealidades en las secuencias de retardos de transmisión, como cambios de regímenes, ráfagas o valores atípicos, necesario para satisfacer la hipótesis requerida de datos *iid* (independientes e idénticamente distribuidos) con el fin de aplicar esas distribuciones marginales; y (iii) regular la transmisión de los datos sensoriales para, por un lado, maximizar la cantidad de datos transmitidos, y, por otro, minimizar el retardo de esas transmisiones para la teleoperación activa.

Esta tesis presenta varias contribuciones en cada uno de estos problemas con el fin de satisfacer los requisitos enumerados. Entre las contribuciones se encuentra el estudio de las distribuciones de probabilidad marginal que, bajo suposiciones moderadas, constituyen una buena aproximación de la distribución real de los retardos sin necesidad del conocimiento de la dinámica del sistema que produce tales retardos. Las distribuciones propuestas han demostrado ser eficientes en su cómputo y consiguen representar aceptablemente los retardos con tan solo modificaciones menores en el software del robot. Por otra parte, se han introducido varias técnicas de medición de ajustes de bondad y algoritmos de detección de no linealidades en secuencias de retardos. Finalmente se presentan varios tipos de

regulación del flujo de datos sensoriales basados en lo anterior, que han sido utilizados tanto en entornos reales como simulados; se ha podido aplicar el trabajo realizado mediante el uso de una aplicación para el control remoto de robots móviles por web desarrollada por personal del grupo.

Palabras clave: Telerobots en red, teleoperación, retardos de tiempo estocásticos, caracterización estadística, detección de no linealidades, regulación automática, interfaces hombre-máquina.

AGRADECIMIENTOS

En primer lugar, debo expresar mi gratitud a mi tutor, al Prof. Dr. Juan Antonio Fernández Madrigal por darme la oportunidad de formar parte del grupo MAPIR (*Machine Perception and Intelligent Robotics Group*) en el departamento ISA (Ingeniería de Sistemas y Automática) en la Universidad de Málaga. También estoy en deuda con él por las incontables horas que hemos pasado juntos discutiendo sobre los diferentes temas de investigación y guiándome a través del difícil arte de escribir artículos científicos. También doy las gracias al comité de miembros por haber aceptado revisar esta tesis.

Naturalmente, un proyecto de investigación como esta tesis no habría sido posible sin el equipo de personas involucradas en nuestro grupo MAPIR. Gracias al Prof. Dr. Javier González Jiménez, Dra. Ana Cruz-Martín, Dr. Vicente Arévalo, al Dr. Cipriano Galindo y al Dr. Francisco Ángel Moreno. Y por supuesto, también tengo que mencionar al resto de compañeros estudiantes de doctorado (algunos ya doctores) en nuestro laboratorio, que han sido un gran apoyo en todo momento: Javier G. Monroy, J. Raúl Ruiz-Sarmiento, Eduardo Fernández, Francisco Meléndez, Carlos Sánchez, Mariano Tarifa, Ángel Martínez Tenor, Manuel López, Emil Khatib, Miguel Algaba y Gregorio Navidad. Gracias a la aportación de cada miembro durante determinadas etapas, el entorno de trabajo se constituyó como un lugar perfecto donde poder desarrollar, en muchas ocasiones de manera colaborativa, las distintas fases de este trabajo.

También debo agradecer a mi familia, que siempre ha sido un apoyo para mí durante estos años. Siempre han creído en mí y han sido una inspiración a lo largo de esta carrera de investigación. Por último, pero no por ello menos importante, estoy totalmente en deuda con Juan, que no sólo ha sufrido mis innumerables horas de trabajo por las noches o incluso en vacaciones, sino que siempre me ha animado en los momentos más difíciles. En gran parte esta tesis es también suya.

Ana Gago Benítez
Málaga
Enero 2017

El trabajo descrito en esta tesis fue financiado por la Junta de Andalucía y la Unión Europea (FEDER) a través del proyecto de investigación de excelencia P08-TIC-04282 del Plan Andaluz de investigación, Desarrollo e Innovación.

Parte I. Descripción de la tesis



UNIVERSIDAD
DE MÁLAGA

1 INTRODUCCIÓN

Internet es una red global a la que se conectan una amplísima diversidad de dispositivos. Está formada por múltiples tramos de comunicaciones, de cable o inalámbricas, de distintas características. Cualquier pareja de dispositivos conectados a Internet pueden transmitir datos entre sí, pero los tiempos empleados en tales transmisiones no son deterministas (predecibles [1]) debido a diversos factores, como protocolos de comunicación con procedimientos estocásticos intrínsecos o sistemas contruidos con componentes de propósito general, no de tiempo real, *off-the-shelf*. En este tipo de sistemas, donde el software y el hardware no proporcionan garantías estrictas de tiempos en las transmisiones, no se proporciona tampoco ningún tipo de medio para facilitar la instalación o modificación de los protocolos de comunicación ni ningún software de aplicación con el fin de mejorar estos tiempos. Todo esto lleva a que las transmisiones de datos entre dos dispositivos cualesquiera que se encuentren conectados a Internet consuman un periodo de tiempo probabilístico, el cual suele exhibir además cambios bruscos de régimen y ráfagas rápidas.

Si un nodo conectado a Internet quiere recibir repetidamente datos de múltiples fuentes, como es el caso de aplicaciones de monitorización remota, *broadcasting* multimedia, teleoperación, etc., donde se desea recibir información periódicamente, esta estocasticidad en los tiempos de envío y recepción de datos hace difícil cumplir requisitos temporales, es decir, dificulta la consecución de los tiempos de transmisión requeridos por la tarea. Una solución consiste en proveer a estas aplicaciones de mecanismos que permitan reaccionar ante esas variaciones de tiempo, ajustando las transmisiones de los datos dinámicamente, para que los requisitos puedan satisfacerse lo mejor posible en un sentido estadístico.

En esta tesis el objetivo es satisfacer de ese modo los requisitos de tiempo en la teleoperación de robots por Internet.

Nuestra aproximación es regular la cantidad de información sensorial que se transmite a la estación remota, de modo que cuando se prevean mayores retardos se solicite menor información y viceversa. Los retardos, al tener una fuerte

dependencia con la cantidad de datos transmitidos, admiten este tipo de regulación, cuyo objetivo es mantener en un nivel adecuado los tiempos de transmisión.

El problema de tratar con retardos de tiempo estocásticos ha sido ya tratado en sistemas de control de red estocásticos [2] (NCS *Networked Control Systems*), pero sólo considerando la impredecibilidad de la red (no de otros componentes, como los sistemas operativos) y bajo la suposición de un conocimiento exacto de la dinámica de la planta a controlar; en la comunidad de redes de comunicaciones (de nuevo, sólo el comportamiento de la red, y considerando la posibilidad de modificar los protocolos de transmisión); en la comunidad de multimedia distribuida [3] (donde no se consideran los estrictos requisitos de tiempo que tienen que cumplir los robots); etc. La teleoperación de robots tiene sus propias características y requiere, por tanto, soluciones propias.

En esta tesis, y con el compendio de artículos presentados, se han investigado metodologías que permitan, fundamentalmente a partir de la regulación de flujo de datos sensoriales, que un robot móvil opere con el máximo determinismo en las circunstancias anteriormente descritas, así como dotarlo de la capacidad de adaptarse de una forma óptima a las condiciones de retardos de transmisión presentes en cada momento. En particular, se ha pretendido desarrollar soluciones prácticas para la teleoperación que no supongan grandes modificaciones de los componentes software y hardware instalados en el sistema, ya que eso las haría inviables para la mayoría de las aplicaciones reales existentes.

Para mantener la cantidad de información transmitida lo más cerca posible al punto óptimo, lo primero que se necesita es un buen estimador estadístico en línea de los retardos, en base a modelos matemáticos precisos. Este modelado de retardos temporales estocásticos sin información explícita del comportamiento de los componentes del sistema puede basarse en varias técnicas [4]: el *state-space reconstruction* reconoce la estructura geométrica subyacente en el comportamiento observado en la secuencia de datos; en *machine learning* se usan redes Bayesianas o neuronales que pueden explorar de forma adaptativa un gran espacio de modelos posibles; los *hidden Markov models* (HMM) pueden asimismo predecir la evolución de la totalidad de la densidad de probabilidad para un conjunto de estados (o regímenes) en la secuencia de retardos de transmisión... Sin embargo, todas estas herramientas presentan la desventaja de una alta complejidad teórica y, sobre todo, computacional.

Aunque la tesis está enfocada hacia la Robótica Móvil, también tiene aplicaciones evidentes en otras áreas que requieran de la optimización de tiempos de transmisión de datos, como robótica de teleasistencia [1], telecirugía [5], exploración submarina [6], operaciones espaciales [7], etc.

1.1 Motivación

Las aplicaciones reales de los robots móviles han experimentado un importante impulso en las últimas décadas. Han surgido desarrollos completamente funcionales de avances previos que no habían salido anteriormente de los laboratorios de investigación, como robots limpiadores [8], [9], vigilantes [10], [11], guías [12], [13], de ayuda a los cuerpos de seguridad [14], [15], de asistencia [16], [17], etc. Sin embargo, la autonomía de los robots actuales no es tan alta como sería deseable, fundamentalmente debido a la incertidumbre presente en los entornos en los que deben operar y a la complejidad de las interacciones con seres humanos individuales o grupos de personas. Teniendo esto en cuenta, sigue siendo de gran importancia el control externo de estos robots con el objetivo de suplir la falta de capacidad de adaptación de los mismos a circunstancias imprevistas [18]. En particular, la teleoperación o control remoto es una de las modalidades de control externo más relevantes (en muchos casos no se trata de una operación continua a lo largo del tiempo, pero eso no siempre reduce la dificultad del problema).

La teleoperación de robots bajo fuertes requisitos de tiempo ha sido tradicionalmente abordada usando componentes predecibles: redes con acceso al medio determinista, sistemas operativos de tiempo real duro, etc. Sin embargo, los robots móviles actuales, en su estadio de integración con entornos humanos, deben adaptarse al máximo a dichos entornos y no al revés, y, por tanto, la disponibilidad de redes deterministas o software para el teleoperador con esas características no están garantizados en la mayoría de los casos. Por este motivo, el estudio de las posibilidades de teleoperación en esas circunstancias, y el aprovechamiento óptimo de las mismas, es un campo de investigación de gran interés hoy día, y una de las motivaciones principales de esta tesis.

También es importante destacar que hace varias décadas que la Robótica Móvil dio un importante paso al incluir el modelado de la incertidumbre como un aspecto clave de algunos de sus problemas más importantes, especialmente la construcción de modelos del entorno (mapas) y la localización precisa de robots en el espacio (lo que se denomina conjuntamente SLAM [19], [20], [21], [22]). Estos procedimientos de estimación de estados de sistemas reales a partir de datos sensoriales con incertidumbre tienen su extensión natural en el control de sistemas estocásticos, tanto continuos [23] como discretos [24], [25]. Sin embargo, el control estocástico no ha obtenido aún resultados en Robótica tan relevantes como los relativos a la creación de mapas y localización, entre otras cosas porque se necesita disponer de modelos previos de los componentes del sistema para poder tomar decisiones óptimas, y éstos son difíciles de obtener. Avanzar por tanto en el control estocástico de sistemas no modelables es también una línea motivadora para la comunidad científica.

1.2 Marco de la tesis

Esta tesis es el resultado de cuatro años de trabajo de la autora como miembro del grupo *Machine Perception and Intelligent Robotics (MAPIR)*¹, el cual se encuentra dentro del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Málaga. La investigación realizada ha sido principalmente financiada por el proyecto de investigación de excelencia de la Junta de Andalucía y la Unión Europea (FEDER): “Control Remoto Óptimo y Adaptable de Robots Móviles con Componentes No Deterministas”², a través de un contrato como ingeniera superior asociada al mismo.

Durante este período, la autora completó con éxito el programa doctoral en Ingeniería Mecatrónica, coordinado por el mismo departamento del que es miembro, donde obtuvo un conocimiento sólido sobre los pilares fundamentales de la Robótica: sistemas de control, sistemas electrónicos, sistemas mecánicos y programación.

1.3 Contribuciones

Esta tesis se presenta como compendio de las siguientes publicaciones (incluidas en la segunda parte, **Parte II: Compendio de publicaciones**). Se presenta un resumen de cada una de ellas en el capítulo 5 de la primera parte.

A. Gago-Benitez, J.A. Fernandez-Madriral, A. Cruz-Martin, "**Marginal Probabilistic Modeling of the Delays in the Sensory Data Transmission of Networked Telerobots**", *Sensors*, vol. 14, no. 2, pp. 2305-2349, 2014. DOI: 10.3390/s140202305

A. Gago-Benitez, J.A. Fernandez-Madriral, A. Cruz-Martin, "**A computationally efficient algorithm for modeling multi-regime delays in the sensory flow of networked telerobots**", *International Conference on Control, Robotics and Cybernetics (ICCRC 2012)*, Cape Town (Southafrica), 2012. ISBN 978-0-7918-6004-5.

A. Gago-Benitez, J.A. Fernandez-Madriral, A. Cruz-Martin, "**Log-logistic modelling of sensory flow delays in networked telerobots**", *IEEE Sensors*, vol. 13, no. 8, 2013. DOI: 10.1109/JSEN.2013.2263381

A. Martinez-Tenor, A. Gago-Benitez, J.A. Fernandez-Madriral, A. Cruz-Martin, R. Asenjo-Plaza, A. Gonzalez-Navarro, "**Hierarchical Regulation of Sensor Data Transmission for Networked Telerobots**", *IEEE Sensors*, Valencia (Spain), 2014. DOI: 10.1109/ICSENS.2014.6985036

Asimismo, el trabajo realizado ha dado lugar a otras publicaciones científicas:

¹ <http://mapir.isa.uma.es/>

² P08-TIC-04282

A. Gago-Benitez, J.A. Fernandez-Madrigal, C. Galindo, A. Cruz-Martin, "**Statistical Characterization of the Time-Delay for Web-Based Networked Telerobots**", 5th International Workshop in Applied Probability (IWAP'10), Madrid (Spain), 2010.

A. Gago-Benitez, J.A. Fernandez-Madrigal, C. Galindo, A. Cruz-Martin, "**Caracterización de los retardos de bucles sensoriales para robots controlados por Web** (in Spanish)", XXXI Jornadas de Automática, Jaén, 2010.

A. Gago-Benitez, J.A. Fernandez-Madrigal, "**Log-normal modeling of non-linear sequences of delays with classical change detection methods**", International Workshop on Applied Probability (IWAP'2012), Jerusalem (Israel), 2012.

A. Gago-Benitez, J.A. Fernandez-Madrigal, A. Cruz-Martin, "**Log-logistic modeling of sensory flow delays in networked telerobots**", IEEE Sensors, Taipei (Taiwan), 2012. DOI: 10.1109/ICSENS.2012.6411420

También se han concedido las siguientes patentes a este trabajo:

Juan Antonio Fernandez Madrigal, Ana Gago Benítez, Ana Maria Cruz Martin, Ángel Martínez Tenor, Rafael Asenjo Plaza, Maria Angeles Gonzalez Navarro (31/10/2014). "**Métodos implementados en computador y sistemas informáticos de regulación automática de la cantidad de datos transmitidos entre dispositivos conectados a una red**". P201400876 | ES2565878. Universidad de Málaga. STATUS: EXAMEN FORMAL Y EXAMEN PREVIO

Ángel Martínez Tenor, Juan Antonio Fernandez Madrigal, Ana Maria Cruz Martin, Ana Gago Benítez, Rafael Asenjo Plaza, Maria Angeles Gonzalez Navarro (31/10/2014). "**Métodos implementados en computador y sistemas informáticos de activación y desactivación automáticas de transmisiones concurrentes de datos entre dispositivos conectados a una red**". P201400877 | ES2550728. Universidad de Málaga. STATUS: EXAMEN FORMAL Y EXAMEN PREVIO.

1.4 Estructura de la tesis

El resto de capítulos de la primera parte de esta tesis se organiza como sigue:

Capítulo 2: Modelado probabilístico de retardos temporales. Aquí se presenta un análisis estadístico de los tiempos de retardo del flujo sensorial para robots teleoperados por Internet, desarrollando una caracterización probabilística de estos retardos para todo el sistema, no sólo la red. Se presentan también las tres distribuciones de probabilidad marginales que han sido más relevantes y que dan mejores resultados a la hora de predecir los

retardos. Finalmente se presentan algoritmos de detección de cambios de regímenes en los retardos del flujo sensorial en dicho sistema, que buscan satisfacer la necesaria hipótesis *iid*.

Capítulo 3: Regulación del flujo de datos. Aquí se propone un sistema de procesamiento de datos para la regulación automática de la cantidad de información que se transmite entre el teleoperador y el robot móvil conectado a Internet, para que esa transmisión sea lo más óptima posible. Se presentan dos métodos de regulación y sus experimentos correspondientes.

Capítulo 4: Aplicación para el control remoto de robots móviles por web. Este capítulo describe la implementación de un controlador jerárquico en una interfaz de teleoperador basada en web, portable y no invasiva, es decir, que requiere modificaciones mínimas en la red u otros componentes de software de la aplicación. Se presenta su estructura, cómo se ha diseñado y los experimentos realizados.

Capítulo 5: Resumen de publicaciones. Este capítulo lista los artículos que conforman el compendio de la tesis, describiendo brevemente su contenido y el papel de la autora en los mismos.

Capítulo 6: Conclusiones y trabajos futuros. Finalmente se discuten las conclusiones que se pueden extraer del trabajo realizado, así como las líneas de investigación que quedan abiertas, e interesantes extensiones a dicho trabajo.

2 MODELADO PROBABILÍSTICO DE RETARDOS TEMPORALES

En este capítulo presentamos el análisis estadístico exhaustivo realizado durante la tesis de los tiempos de retardo del flujo sensorial para robots teleoperados por Internet, desarrollando asimismo una caracterización probabilística de estos retardos para todo el sistema, no sólo la red. Nuestro enfoque, basado en distribuciones marginales e probabilidad, reduce la complejidad computacional del problema, permitiendo implementaciones eficientes en máquinas que tienen limitaciones computacionales como teléfonos móviles, tablets, etc, cada vez más utilizados como interfaces. Este análisis es, por tanto, innovador ya que afronta simultáneamente el problema desde un punto de vista de la eficiencia, en un entorno heterogéneo (es decir, que involucra componentes distintos a la red, como sistemas operativos, interfaces de teleoperación y software de aplicación), y sin conocimiento previo de la dinámica de los retardos.

Primeramente se mostrarán los trabajos que existen en la literatura relacionados con este problema y cómo se afronta, dependiendo del campo de aplicación. Posteriormente se presentarán las distribuciones de probabilidad más prometedoras para estimar estos retardos, se mostrarán experimentos realizados con robots reales, obteniendo las señales de retardos de cada experimento y justificando el uso de distribuciones marginales para afrontar el problema. Nos enfocaremos en las distribuciones de probabilidad que han sido más relevantes y que dan mejor resultado a la hora de predecir los retardos. Finalmente se presentan los algoritmos de detección de cambios de regímenes en los retardos del flujo sensorial.

2.1 Introducción y estado del arte

Los telerrobots, y, en general, las plataformas con sensores controlados de forma remota, están presentes en muchas aplicaciones avanzadas, como ya se ha mencionado en el capítulo anterior. En particular, los telerrobots móviles en red [26] son una clase de telerrobots móviles controlables a través de redes como Internet, por lo que son accesibles al público en general a través de, por ejemplo, la

World Wide Web. La función principal de una plataforma móvil de este tipo consiste en recibir y ejecutar comandos de movimiento que se emiten desde una estación de usuario remota, que a su vez muestra información adquirida por los sensores del robot, típicamente cámaras, datos de distancia, etc. (véase la figura 1). Esta tarea de control remoto tiene requisitos de tiempo que deben ser satisfechos: el usuario no podría realizar correctamente la navegación del robot si no le llegaran datos sensoriales del mismo con suficiente frecuencia; sin embargo, especialmente cuando los sensores son capaces de proveer una cantidad de datos (p. ej. cámaras), sí podría controlar satisfactoriamente al robot aunque recibiera menos datos de éste.

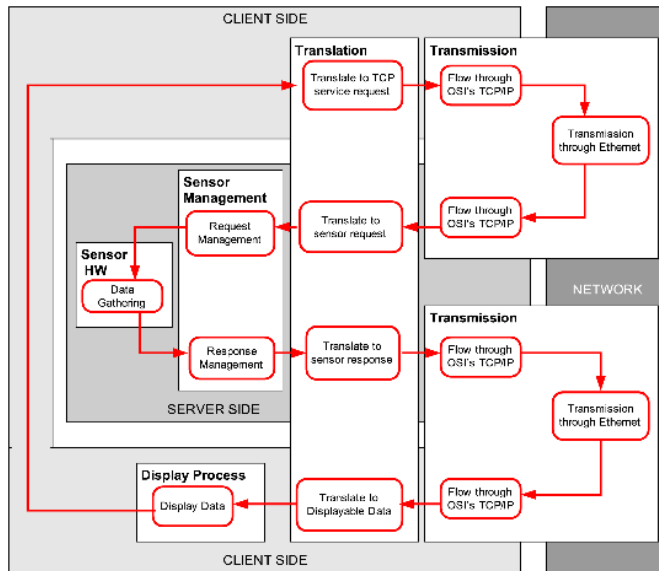


Figura 1. Esquema general de un bucle sensorial de un robot móvil controlado por Internet (*hardware y software*).

Desafortunadamente, la tecnología actual de Internet presenta problemas importantes para realizar control remoto en tiempo real debido a su estocasticidad, pero es importante destacar que no sólo la red es un problema en este aspecto: tanto en los sistemas operativos de propósito general como en aplicaciones software es imposible garantizar en todas las situaciones que el tiempo de flujo de información esté acotado. Todos estos componentes constituyen sistemas en tiempo real suaves [27], donde sólo se pueden satisfacer requisitos estadísticos, por ejemplo, "los datos del sensor deben llegar cada 500 milisegundos con un 90% de probabilidad". Una buena estimación probabilística de los retardos en la transmisión de los datos sensoriales es consecuentemente necesaria con el fin de predecir la satisfacción de tales requisitos.

Se pueden encontrar en la literatura enfoques diferentes, no necesariamente relacionados con la Robótica, para tratar con retardos estocásticos de este tipo: en la comunidad de redes de comunicaciones (*networking*), por lo general, se intenta mejorar la calidad de servicio de la red (*Quality of Services*: QoS) basándose en la modelización y el control del flujo de información [28] [3], lo que se resuelve mediante modificaciones en los protocolos de red o el hardware [29], o por medio de enfoques más exóticos, tales como la inserción de retardos artificiales en la vía de transmisión con el fin de compensar los retardos ya existentes [30]; en el área del Control Automático se han propuesto diferentes modelos teóricos para controlar la dinámica completa de sistemas remotos o distribuidos dedicados al control de procesos —los cuales incluyen retardos temporales— [31] [32] [33]; una cantidad importante de investigación se está enfocando en los sistemas de control en red (*Networked Control Systems*: NCS), aunque es común en estas comunidades considerar los componentes que no son la red como deterministas y, por lo general, conocer la dinámica de la planta a controlar. Por último, también existen enfoques para controlar la sincronización del flujo de datos en aplicaciones multimedia [3] [34] [35], pero en este campo no tienen que hacer frente a los estrictos requisitos de tiempo de control de la teleoperación de robots.

Hay que insistir en que la mayoría de soluciones presentes en la literatura para el modelado y la regulación de los retardos que se producen en los sistemas en red sólo se ocupan de la parte de red: típicamente se considera el retardo de extremo a extremo, el cual incluye solamente la conversión A/D, paquetización, propagación por la red, encolado y almacenamiento en búfer [36], o bien el RTT (*Round Trip Time*), el retardo temporal existente entre el envío de un paquete a través de la red y la recepción del acuse de recibo [37].

En este capítulo nos centraremos en el análisis de modelos matemáticos minimalistas para los retardos de tiempo completos (no sólo los retardos en la red) que se encuentran en el tipo de aplicaciones descritas anteriormente. Hay que tener en cuenta que las aplicaciones de robots controlados por red pueden ser generalizadas, bajo el punto de vista de la modelización y la regulación de sus retardos, a cualquier escenario en el que sensores remotos envíen datos a una estación a través de una red estocástica con otros componentes de software no deterministas. Desde este punto de vista general, el enfoque que presenta este trabajo busca satisfacer los siguientes objetivos:

- La aplicación requiere que los datos lleguen antes de un tiempo concreto —al menos bajo restricciones probabilísticas— como se explicó anteriormente, con el fin de ser útil.
- Sólo se pueden hacer mínimas modificaciones al sistema existente.
- No tenemos conocimiento preciso sobre la dinámica de los retardos.
- Disponemos de una potencia de cálculo modesta.

Teniendo en cuenta todas estas limitaciones, presentamos un análisis estadístico de los retardos de tiempo del flujo sensorial para robots controlados en

red. Asimismo, demostramos que las secuencias temporales de los retardos recogidos de un sistema de este tipo se pueden modelar con éxito con herramientas estadísticas simples basadas en distribuciones de probabilidad marginales, especialmente cuando los cambios bruscos en la señal son adecuadamente detectados, es decir, de forma rápida y con alta sensibilidad, ya que esos cambios bruscos pueden utilizarse para separar la secuencia en segmentos que no contengan no linealidades (en términos estadísticos, las distribuciones de probabilidad marginales son modelos correctos siempre y cuando haya independencia temporal entre los valores en la secuencia).

En este trabajo hemos analizado los escenarios en los que se espera que este enfoque funcione, y en general, las características que muestran. La principal conclusión respecto al modelado de los retardos es que nuestra propuesta de distribución marginal reduce la complejidad computacional con respecto a otros métodos y mantiene un adecuado nivel de exactitud. En particular, proporciona significancia estadística, es decir, los modelos obtenidos pueden explicar los datos en un sentido estadístico.

Técnicas existentes de modelado de secuencias de tiempo y su coste computacional

Los métodos estándares que se aplican comúnmente para caracterizar secuencias generales de valores aleatorios funcionan mediante la representación de toda la secuencia con un modelo único estructurado, en lugar de separar la secuencia en segmentos independientes como proponemos aquí. Los dos métodos más comunes que se encuentran en la literatura son las series temporales y los modelos ocultos de Markov (*Hidden Markov Models*: HMM), ambos con costes computacionales muy conocidos [38]. Se suelen utilizar de forma *off-line* debido a esos costes.

Las series temporales pueden definirse formalmente de varias maneras:

- Modelos ARMA, que son $O(m^3T)$, donde T es la longitud de la serie y $m = p + q$ la suma de órdenes del modelo –estos órdenes deben ser definidos previamente con algún proceso adicional–. No son capaces de representar señales con cambios bruscos o con tendencias.
- Cuando la serie tiene tendencia, podemos utilizar un modelo ARIMA, más complicado, que es $O(T^2)$ [39], pero tampoco podemos hacer frente a los cambios bruscos en la señal.
- Cuando la señal ha de ser segmentada debido a la presencia de estos cambios bruscos, existen algoritmos en tiempo real basados en ARMA, pero son $O(m^3T^2)$ [40].

- Por último, se pueden encontrar algoritmos más complejos y específicos de series temporales, pero presentan aún peores costes computacionales [41].

Por otro lado, los HMM tratan naturalmente con señales que cambian abruptamente; las representan como la salida de un proceso estocástico estructurado que varía su estado (oculto) probabilísticamente. Desafortunadamente, el aprendizaje de los parámetros de un HMM, por lo general, requiere reunir secuencias sensiblemente más largas de valores que en el caso ARMA, y, además, su complejidad es $O(N^2T^2)$, donde N es el número de estados considerados del sistema. Ese número debe ser estimado previamente con algún otro procedimiento.

En contraste con estos enfoques estándares, nuestro método de distribución marginal es $O(T)$, ya que la estimación de los parámetros de distribución y el algoritmo de detección de cambios son, a lo sumo, $O(T)$. En el peor de los casos, cuando se utiliza un algoritmo de optimización no lineal para encontrar los parámetros de la distribución –lo cual se puede evitar si empleamos estimación de máxima verosimilitud (*Maximum Likelihood Estimation*: MLE), como es el caso en muchos de nuestros modelos, por ejemplo la log-normal [42]–, el número de pasos de convergencia para la optimización puede acotarse con una constante en todos los casos prácticos. Por otro lado, el algoritmo de detección de cambios consiste en la aplicación de una expresión matemática, que es simplemente $O(1)$.

2.2 Modelado de los tiempos de transmisión mediante distribuciones de probabilidad marginal

Como hemos visto en el apartado anterior, usar distribuciones de probabilidad marginal para representar la secuencia de retardos en el bucle sensorial tiene un coste computacional $O(T)$ a la hora de estimar los parámetros. Este factor hace a esta opción altamente ventajosa si queremos implementarlo en sistemas que incluyen interfaz de cliente basada en web, por ejemplo; sin embargo, hay que tener en cuenta la independencia estadística en la secuencia de retardos para poder aplicar este tipo de distribuciones marginales.

Para comprobar si las muestras de tiempo a las que nos enfrentamos son independientes e idénticamente distribuidas (*iid*), se han analizado los coeficientes de correlación en los datos adquiridos de diversos escenarios reales: los retardos serían dependientes si el autocorrelograma (ACF) de los mismos subiera por encima de los límites de confianza correspondientes; de lo contrario, la hipótesis de independencia en datos de retardo secuenciales no puede ser rechazada [43]. Además, otros investigadores han comprobado que estas muestras también demuestran independencia ante no-linealidades de orden mayor o de otro tipo¹.

¹ Comunicación de uno de los revisores de la tesis doctoral.

En nuestro estudio [44], los ACFs muestran una fuerte dependencia en muchos de los escenarios reales originales cuando la no estacionalidad no se detecta. En el caso de que se detecten cambios de regímenes o ráfagas, los ACF se quedan más cerca de cero, quedando por debajo de los límites de confianza en la mayoría de los casos. Esto apoya nuestra hipótesis de considerar secuencias *iid* de retardos, siempre y cuando se separen los regímenes y podamos detectar ráfagas y valores atípicos aislados.

Durante el desarrollo de la tesis hemos explorado métodos estadísticos de control (SPC: *Statistical Process Control*) como el *Cusum* [45], pero no se obtuvieron resultados relevantes en cuanto a la detección de cambios bruscos en la señal, debido a la distribución no clásica de la misma.

Hay que señalar que también podrían aparecer efectos sistemáticos en los retardos, dando lugar a distribuciones multimodales. En este trabajo asumimos que son despreciables en la mayoría de las situaciones reales estocásticas, como se ha observado en nuestros experimentos.

En resumen: las suposiciones *iid* (basada en una correcta detección de no linealidades) y de estocasticidad (no considerando influencias sistemáticas) nos permiten proporcionar una solución simple y de poco peso computacional al problema, en comparación con los métodos estándares que se aplican comúnmente.

Experimentos en escenarios reales

Para estudiar la diversidad de combinaciones de componentes no deterministas que se pueden encontrar en sistemas robots teleoperados por red, hemos seleccionado una serie de escenarios (enumerados en la Tabla 1), obteniendo de este modo una idea bastante completa de los posibles comportamientos de los retardos en la transmisión de datos sensoriales.

Escenario	Robot	Sensor	Resolución	Lugar	PC Cliente	PC Servidor	Red
#1	SANCHO	Láser	181 puntos	Mismo edificio	A	a	1
#2	SANCHO	Cámara web	20% B&N	Misma ciudad	B	a	2
#3	SANCHO	Cámara web	40% B&N	Misma ciudad	B	a	2
#4	SANCHO	Cámara web	30% B&N	Misma ciudad	B	b	3
#5	SANCHO	Cámara web	100% color	Misma ciudad	B	b	3
#6	SANCHO	Cámara web	10% B&N	Mismo edificio	A	b	4
#7	SANCHO	Cámara web	50% B&N	Mismo edificio	A	b	4

#8	SANCHO	Cámara web	100% B&N	Mismo edificio	A	b	4
#9	SURVEYOR	Cámara web	5.86% a todo color	Mismo edificio	C	c	5
#10	GIRAFF	Odometría	9323 bytes	Mismo edificio	D	d	1+4
#11	sólo odometría simulada PHP	Null (echo)	20 bytes	Ituzaingó (Argentina) Málaga (España)	E	e	4+6+7
#12	simulador robot móvil C++	Odometría	80 bytes	Misma ciudad	F	e*	2+4
#13	Arduino	Entrada analógica	27 bytes	Mismo escritorio (p2p)	G	f	8
#14	robot completo simulado PHP	Null (echo)	4 bytes	Mismo edificio	H	h	4
#15	robot completo simulado PHP	Cámara simulada	100% RGB	Mismo ordenador	H	g	No net
#16	robot completo simulado PHP	Láser simulado	2048 puntos	Mismo edificio	H	h	4
#17	robot completo simulado PHP	Cámara simulada	100% RGB	Mismo PC	H	h	4
#18	robot completo simulado PHP	Cámara simulada	100% RGB	Plymouth (GB) Málaga (España)	I	e	4+6+7

Tabla 1. Resumen de los escenarios experimentales que han sido creados para modelar el retardo en la transmisión de datos sensoriales de nuestros robots teleoperados en red.

Legendas: **A:** Pentium M @ 1.8Ghz, 1Gb RAM, LINUX, FireFox. **B:** Intel Core Duo T7200 @ 2Ghz, 2Gb RAM, LINUX, FireFox. **C:** Android smartphone Galaxy SII. **D:** Escritorio PC Intel R CoreTM i5 3330 @ 3GHz, 8GB RAM, Windows XP, interfáz de teleoperación gráfica de [46]. **E:** Portátil intel core 2 duo T6400 @ 2GHz, 2GB RAM, Windows XP. **F:** Portátil Intel core 2 duo T7200 @ 2GHz, 2GB RAM, Linux. **G:** Escritorio PC Intel core i7 960 @ 3.20 GHz, 12MB RAM, Linux. **H:** Escritorio PC Intel core i7 940 @ 2.93 GHz, 12 GB RAM, Linux. **I:** Portátil Intel Core 2 Duo P9600 @ 2.53 GHz, 4 GB RAM, Linux. **a:** Intel Pentium M @ 2Ghz, 1Gb RAM, LINUX. **b:** Intel Pentium IV @ 3.2Ghz, 1Gb RAM, WINDOWS. **c:** Surveyor, Android. **d:** Mini-PC, Intel core i3 M @ 2.10 GHz, 3GB RAM, Windows Embedded Standard. **e:** Escritorio PC Intel core i7 950 @ 3.07GHz, 16GB RAM, Linux. **f:** Placa Arduino con Atmel Atmega 328P @ 16MHz, 2KB RAM, no OS. **g:** Escritorio PC Intel core i7 940 @ 2.93 GHz, 12 GB RAM, Linux. **h:** Mini PC Intel core i5 650 @ 3.20 GHz, 4 GB RAM, Windows 7 Enterprise N. **1:** WiFi 802.11b/g, proveedor Universidad local en el mismo edificio que el robot. **2:** WiFi 802.11b/g, ISP proveedor situado en Madrid (a unos 600 km de nuestro laboratorio). **3:** RTC 56Kbps, ISP proveedor situado en Madrid (a unos 600 km de nuestro laboratorio). **4:** Par trenzado 1GBs Cable. **5:** WiFi ad-hoc. **6:** Cable de datos submarino [47]. **7:** WiFi 802.11b/g ISP proveedor en Ituzaingó, Argentina. **8:** USB 2.0 cable punto a punto.

*: simulación de robot ejecutándose en una máquina virtual con Windows XP y el sistema de desarrollo BABEL [26].

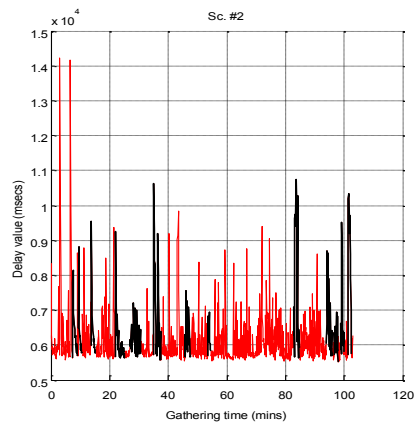
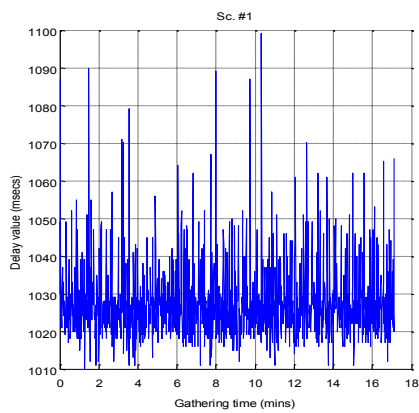
Ya que nuestro enfoque está dirigido a no usar ningún conocimiento acerca de la planta que está produciendo los retardos, su aplicabilidad no se puede demostrar analíticamente para cada combinación existente de red, hardware y software, pero se puede observar que los componentes de la tabla 1 son representativos de una amplia gama de aplicaciones prácticas: en lo referente a la red, algunas de nuestras configuraciones incluyen múltiples segmentos, ya que los experimentos se han llevado a cabo desde lugares geográficamente distantes, mientras que otros han sido realizados simplemente con un segmento de red en el sistema o incluso con una conexión simple punto a punto; el hardware de los robots es, obviamente, fijo, pero los robots utilizados son muy diferentes (véase figura 2) y, en particular, tienen diferentes potencias de cálculo, basadas en microcontroladores, ordenadores portátiles estándar y mini-PCs.

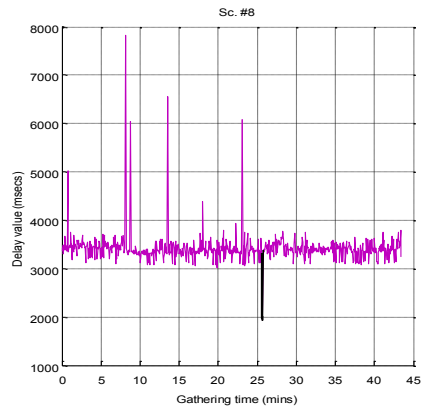
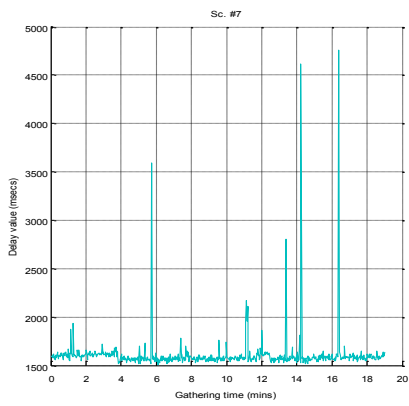
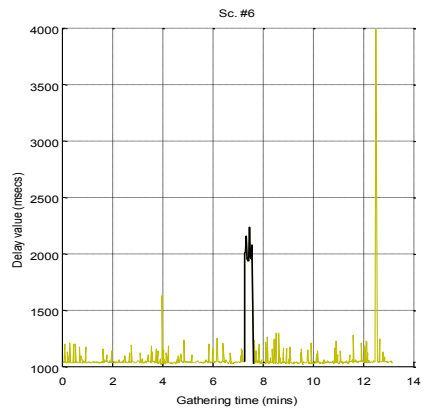
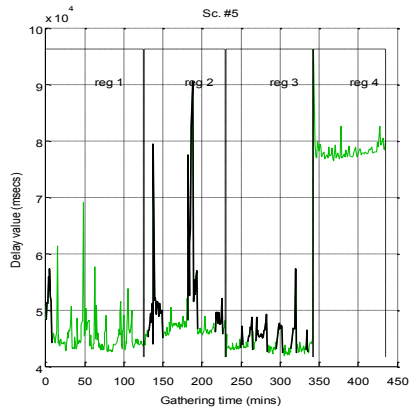
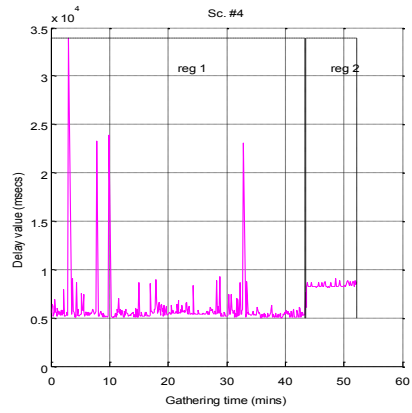
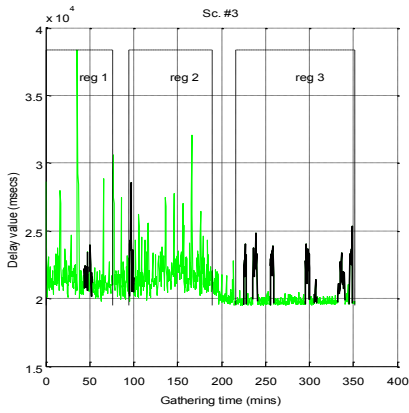
La figura 3 muestra las secuencias de retardos que hemos medido en el bucle sensorial para cada uno de los escenarios definidos en la Tabla 1, según el momento en el que se midió cada valor de retardo. Cada valor de retardo correspondiente al tiempo transcurrido desde que se solicita una captura de datos a un sensor hasta que se solicita la siguiente, es decir, el retardo comprende tramos de transmisión por red y software de sistema y aplicación. A la hora de medir estos tiempos se ha considerado que los ordenadores encargados de esas medidas tienen sincronizados sus relojes, típicamente mediante el algoritmo NTP [50].

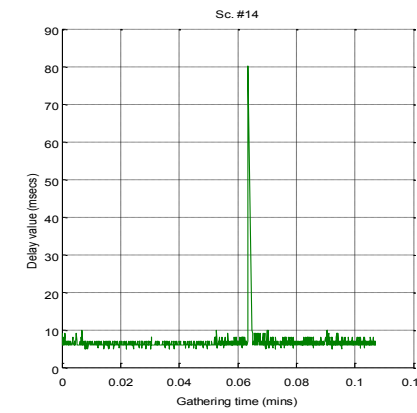
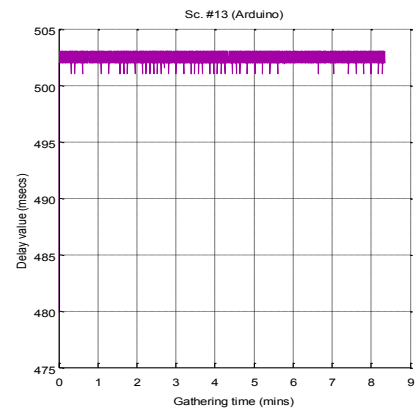
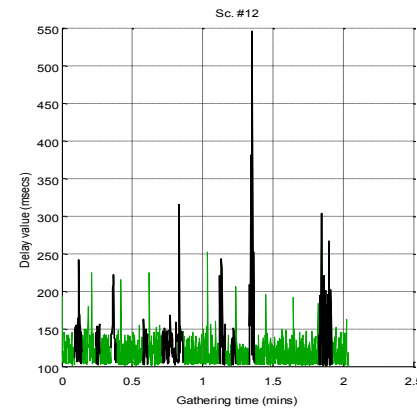
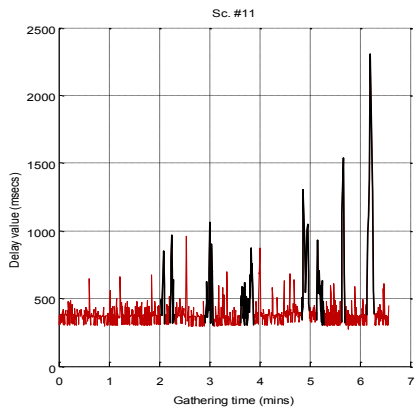
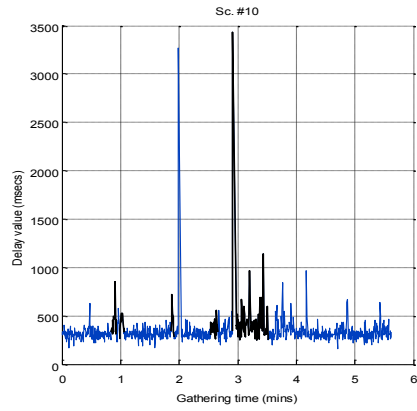
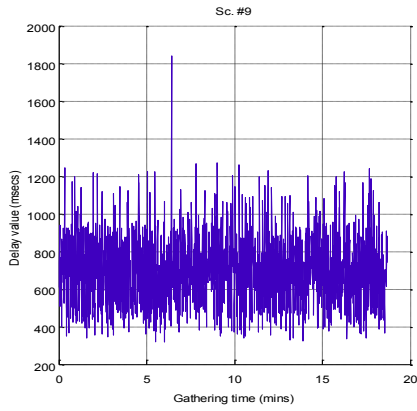
Es fácil observar los cambios bruscos de regímenes, los valores atípicos y las ráfagas en esas señales. También, algunas de ellas exhiben cambios suaves sostenidos, es decir, sus tendencias varían. En estas figuras se han dibujado líneas gruesas para separar las partes de los escenarios en los que podemos detectar visualmente cambios bruscos u otras no linealidades (es decir, estas partes son casi estacionarias, lo que hemos llamado "regímenes"). También se han marcado algunas ráfagas que son fáciles de distinguir a simple vista.



Figura 2. Robots utilizados para los escenarios reales: prototipo SANCHO [48] (izquierda-arriba), Surveyor SRV-1 [49] (izquierda-abajo) y GIRAFF [46] (derecha).







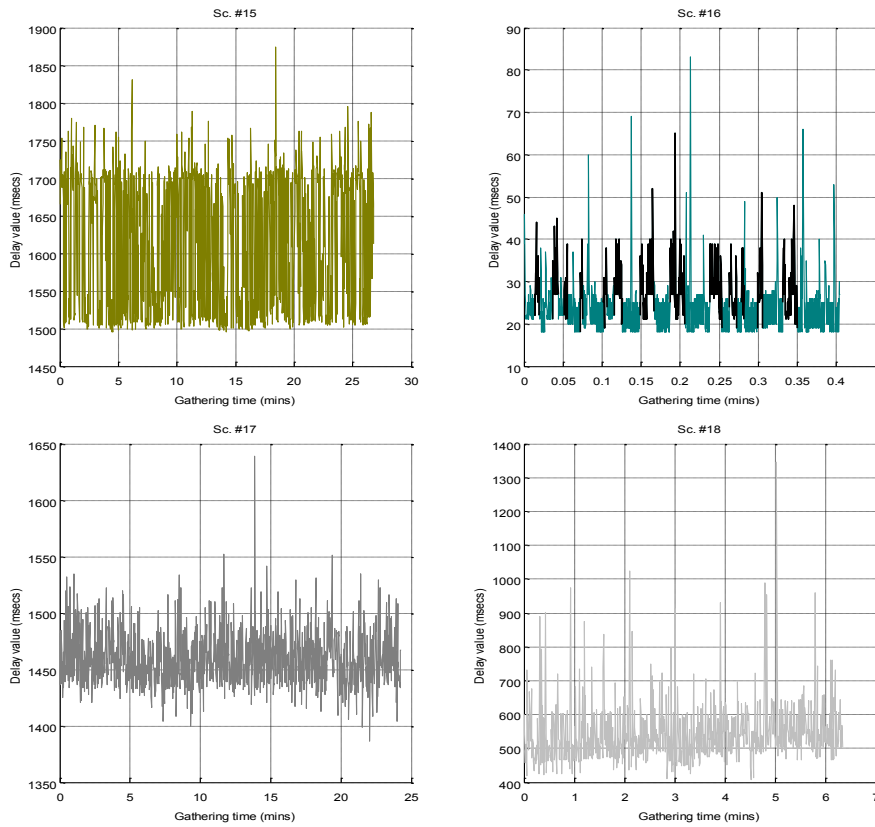


Figura 3. Retardos reales, en milisegundos, que se recogieron del bucle sensorial de los escenarios de la Tabla 1. Se observan las ráfagas y cambios de régimen (detectados visualmente con líneas negras), y, en algunos casos, las variaciones suaves de tendencia. La parte inferior del eje x muestra el tiempo absoluto cuando fue recogido cada dato de la muestra.

Distribuciones de probabilidad marginal más apropiadas para este tipo de retardos

Para modelar el comportamiento estocástico marginal de las secuencias de retardos, hemos elegido una serie de funciones de densidad de probabilidad paramétricas (que se enumeran en la tabla 2). En su momento se estudiaron distribuciones no paramétricas como la *Spline* [51] o la basada en *kernels* gaussianos [52], pero resultaron ser más propensas a modelar las variaciones de una muestra particular en lugar del proceso subyacente estocástico de toda la señal (es decir, *sobreajuste*), siendo modelos pobres cuando se recogían nuevas muestras [44].

Los modelos que hemos seleccionado reflejan algunas ideas de sentido común acerca de las características observadas en los datos reales, a saber:

- Cuando el sistema tiene retardos muy cortos, por ejemplo, cuando los sensores envían pocos datos, como es el caso de los sensores típicos de distancia u odometría de un robot móvil, o cuando las comunicaciones y/o software son muy rápidas, la forma de la distribución estadística de los retardos suele ser muy parecida a una exponencial [53]. Con el fin de hacer este modelo más flexible, también hemos incluido en nuestro análisis la distribución Erlang, que puede explicar los datos procedentes de una suma de k componentes distribuidos exponencialmente e independientes.
- En la literatura de redes, los retardos de transmisión en la red, cuando no son tan cortos, se explican a menudo por distribuciones con cola larga y pronunciada. Por ejemplo, se ha propuesto la distribución log-normal para el tráfico en la red [54]. No existe ningún informe en la literatura que muestre que el retardo total (es decir, la red + cliente + servidor) apoye este modelo, pero nuestros resultados finales corroboran que son aptos para más escenarios que otros modelos; por otra parte, cuando los tres componentes del retardo total se dividen (tiempo de transmisión desde el cliente, tiempo de procesamiento en el robot, tiempo de transmisión de la respuesta sensorial), o cuando usamos un método de detección de cambios, la log-normal y la log-logística se convierten en modelos paramétricos de referencia [42] [55].
- Con el fin de aumentar la diversidad de los modelos mencionados anteriormente, también hemos probado la distribución de Weibull, que es otra generalización de la exponencial. Por las mismas razones, hemos incluido la distribución Gamma, que generaliza la Erlang y ha sido propuesta para modelar el tiempo de ida y vuelta de las transmisiones por red inalámbrica [37].

pdf	Constraints
$\text{Erlang}(x; \alpha, \beta, \lambda) = \frac{(x-\gamma)^{(\alpha-1)}}{\beta^\alpha (\alpha-1)!} e^{-\frac{(x-\gamma)}{\beta}}$	$x, \alpha, \beta, \gamma \geq 0 \quad x, \alpha, \beta, \gamma > 0$
$\text{Gamma}(x; \alpha, \beta, \lambda) = \frac{(x-\gamma)^{(\alpha-1)}}{\beta^\alpha \Gamma(\alpha)} e^{-\frac{(x-\gamma)}{\beta}}$	$x, \alpha, \beta, \gamma \geq 0 \quad x, \alpha, \beta, \gamma > 0$
$\text{Lognormal}(x; \mu, \sigma, \gamma) = \frac{1}{(x-\gamma)\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x-\gamma)-\mu)^2}{2\sigma^2}}$	$\mu, \gamma \geq 0 \quad x, \sigma^2 > 0$
$\text{Weibull}(x; k, \lambda, \gamma) = \frac{k}{\lambda} \left(\frac{x-\gamma}{\lambda} \right)^{k-1} e^{-((x-\gamma)/\lambda)^k}$	$\lambda, k, \gamma, x \geq 0$
$\text{Exponential}(x; \lambda, \gamma) = \lambda e^{-\lambda(x-\gamma)}$	$\lambda, \gamma, x \geq 0$

Tabla 2. Formas matemáticas de las pdfs paramétricas usadas en nuestra caracterización. Todas ellas son consideradas tri-paramétricas, siendo γ el desplazamiento, es decir, el límite inferior para los retardos.

Para encontrar el mejor ajuste para cualquiera de estos modelos, primero hemos construido un histograma de los datos con $2n^{2/5}$ celdas², donde n es el tamaño de la muestra de ajuste. A continuación, utilizamos la estimación de probabilidad máxima (MLE) para cada distribución. Aclarar que se han utilizado métodos que permiten realizar la bondad de ajuste con los parámetros estimados. Después, para estudiar la bondad de ajuste, hemos empleado tanto el test Chi-cuadrado (χ^2) como el de Kolmogorov-Smirnov (K-S) con 0.05 de significancia. Como se mostrará más adelante, el primero da peores resultados en general; esto se debe a su mayor sensibilidad a muestras más pequeñas.

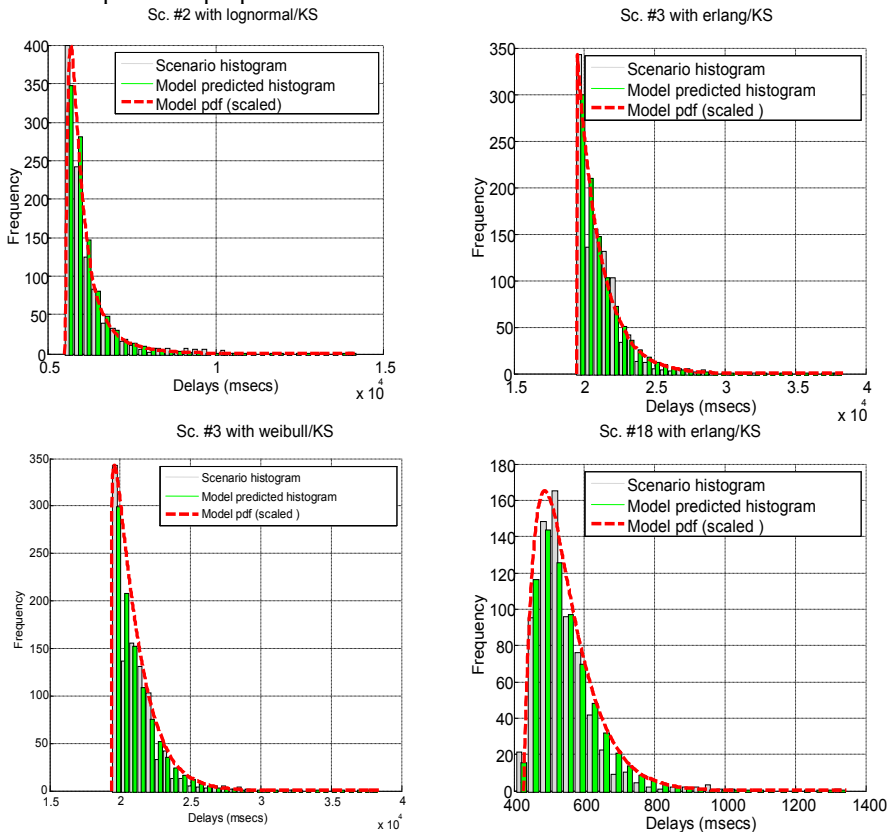
Después de ejecutar los tests de bondad, vemos en la Tabla 3 que, como se esperaba, debido a la naturaleza no lineal de la mayoría de los casos, sólo unos pocos escenarios pueden ser explicados completamente por los modelos paramétricos: #3, #2 y #18. El resto no son compatibles con la hipótesis nula. Los escenarios no rechazados se muestran en la figura 4, donde los ajustes no se han realizado sobre los histogramas, sino sobre los propios datos. Se han pintado de esta manera solo para fines ilustrativos. Estos resultados muestran que, para una elección de distribución paramétrica, podrían ser preferibles la log-normal o Erlang cuando no somos capaces de detectar valores atípicos, ráfagas y regímenes; pero estos modelos, aun así, no son lo suficientemente buenos y podrían ser aplicables a muy pocos casos reales.

² De acuerdo con [63], esta fórmula es útil para dividir el histograma en compartimientos con la misma frecuencia, para que los test de bondad de ajuste de Chi-cuadrado funcionen adecuadamente.

	Exponencial (p-valor)	Erlang (p-valor)	Gamma (p-valor)	Log-normal (p-valor)	Weibull (p-valor)
χ^2	-	-	-	-	-
KS	-	#3(0.03) #18(0.06)	#18(0.08)	#2(0.07) #18(0.03)	#3(0.03)

Tabla 3. Hipótesis no rechazadas de cada escenario sin procesar.

En resumen, la Erlang y la Log-normal son las distribuciones de probabilidad marginal que mejor explican el comportamiento de las muestras, pero hemos de mejorar este modelado haciendo una detección exhaustiva de los distintos regímenes que aparecen en las muestras y representando cada régimen con una distribución, como veremos en los próximos apartados, con algoritmos específicos diseñados para este propósito.



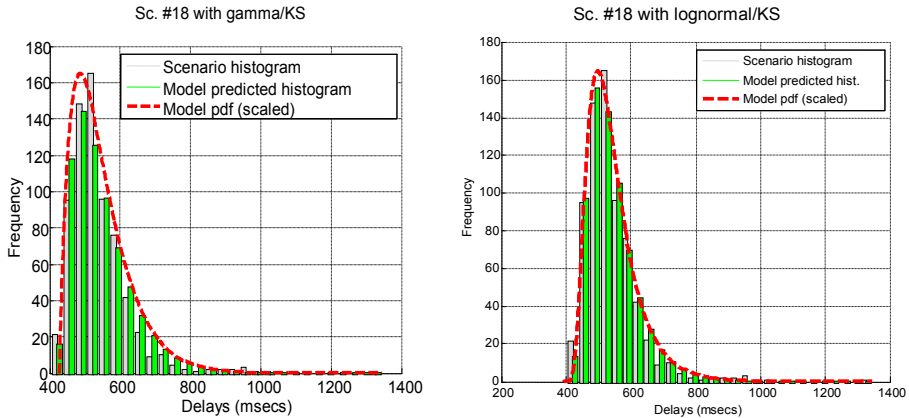


Figura 4. Histogramas ilustrativos y ajustes de los modelos que no rechazan uno o los dos tests de hipótesis, según la Tabla 3, con modelos paramétricos. Nótese que los ajustes en esta tesis se han realizado siempre sobre los datos, no sobre los histogramas de los mismos.

Detección de regímenes con el método Confidence Interval

La primera aproximación a la detección de cambios abruptos que se realizó en esta tesis fue [44]. Este método es capaz de separar la secuencia temporal en partes que no contienen no linealidades. No puede garantizar, sin embargo, que los segmentos resultantes pasen el test de hipótesis, sólo que deberían ser modelados por separado.

El método se basa en inferencia estadística básica; trata de calcular la probabilidad de que un valor pertenezca o no a un modelo conocido o, en nuestro caso, que se encuentre en una "outlier-region" definida en la cola derecha del modelo actual (dado un nivel de significancia α). Esta definición ha proporcionado un procedimiento eficiente de detección de cambios, con buenos resultados incluso con secuencias cortas de retardos.

Según la literatura, la detección de regímenes podría basarse en detectar valores atípicos (*outliers*). Existen métodos de detección de *outliers* que no imponen ningún modelo probabilístico particular para los datos [56] [57]; otras propuestas buscan valores que pueden ser consistentes con el conjunto global de datos pero no con sus vecinos locales, los llamados *spatial outliers* [58]. Pero estas técnicas independientes de la distribución suelen necesitar una gran capacidad computacional. Por lo general, los detectores se diseñan para gestionar espacios de alta dimensión de manera eficiente en la minería de datos [59] [60], por lo que requieren grandes conjuntos de éstos –largas secuencias de retardos en nuestro caso– lo que no suele ser posible en nuestro problema.

El método que presentamos en [44] funciona de la siguiente manera. Estimamos un modelo inicial mediante MLE con las primeras 40 observaciones de

retardos. Esto, además de proporcionar los parámetros para la distribución seleccionada, genera unos intervalos de confianza del 95% de esos parámetros (razón por la que nuestro método sólo funciona con distribuciones paramétricas). Cuando llega un nuevo valor de retardo, se clasificará como valor atípico si no está dentro de un rango definido a la izquierda por el *offset* estimado en la distribución, y a la derecha por x_α . Este x_α se calcula encontrando el valor del eje x de la distribución elegida que satisface el nivel de significancia predefinido:

$$1 - cdf(x = x_\alpha) = \alpha \quad (ec.1)$$

Si resolvemos la ecuación anterior para x_α , entonces x_α es función de α y de los parámetros estimados de la distribución, los cuales tienen varios valores posibles si usamos el intervalo de confianza del 95% obtenido en su estimación. Para conseguir un valor de x_α que detecte tantos valores atípicos como sea posible (es decir, minimizar x_α), hay que seleccionar los parámetros de la distribución adecuados dentro del conjunto que nos da ese intervalo de confianza.

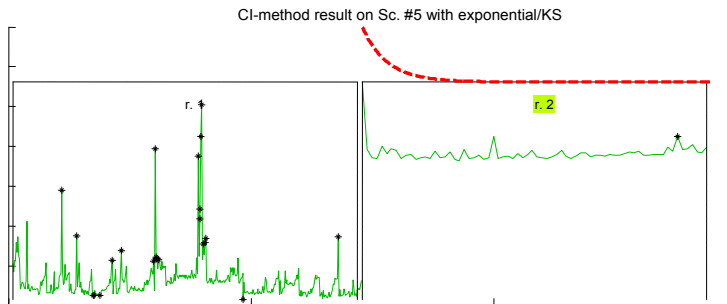
En nuestro método, si el nuevo valor de retardo se encuentra fuera del rango calculado, debe tratarse como un valor atípico. Si esta situación persiste de cinco a cuarenta valores atípicos consecutivos, todos se descartan del escenario como una ráfaga. Cuarenta resultados atípicos consecutivos indican un cambio de régimen. Si esto ocurre, estos cuarenta retardos se incluirán en el comienzo del nuevo régimen. Por último, se consideran los valores atípicos aislados (*singleton*) como parte de la distribución actual, pero si el número de valores atípicos consecutivos es mayor que uno y menor de cinco, se descartan.

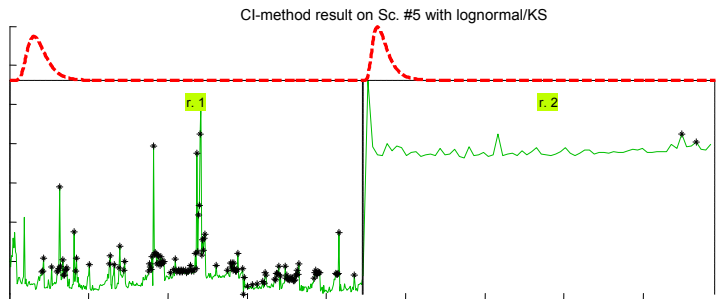
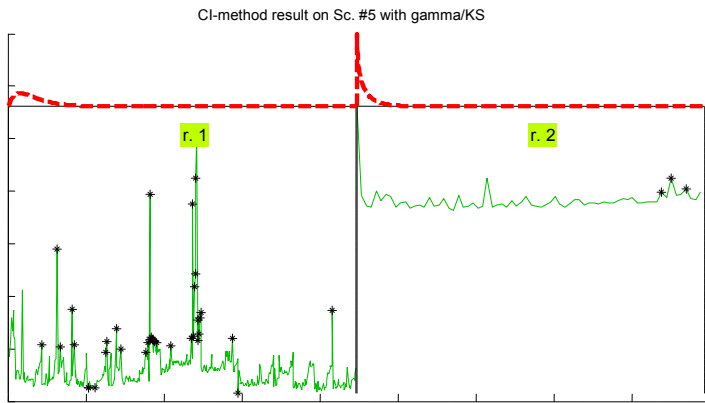
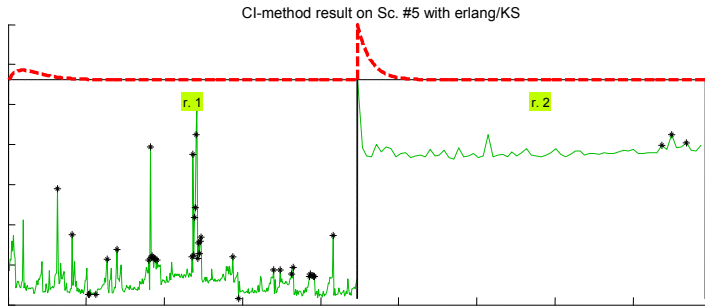
En la tabla 4 se muestra cómo este método produce mejores ajustes que los retardos no procesados (ver Tabla 3) en todos los modelos paramétricos: con las distribuciones paramétricas anteriormente comentadas, explica una parte de los escenarios presentados (no rechaza la hipótesis nula de los tests), especialmente en el caso de la distribución log-normal.

	Exponential (#regs; p-value)	Erlang (#regs; p-value)	Gamma (#regs; p-value)	Log-normal (#regs; p-value)	Weibull (#regs; p-value)
χ^2	-	-	-	#17(0.17)	-
				0.7%	
KS	#4(1;0.19) #5(1;0.47) 1.4%	#5(2;0.09) 5.1%	#4(0.45) #5(2;0.32) #18(0.15) 10.7%	#1(14;0.27) #3(2;0.35) #4(2;0.30) #5(2;0.44) #7(4;0.28) #8(0.57) #9(3;0.30) #13(4;0.08) #16(0.10) #17(14;0.44) #18(9;0.50) 23.5%	#5(0.42) #17(0.17) 6%

Tabla 4. Tabla de los escenarios no rechazados para el método del intervalo de confianza con distribuciones paramétricas. Entre paréntesis aparecen los regímenes detectados en cada escenario junto con el p-valor (que indica la bondad del ajuste).

La figura 5 ilustra los resultados del escenario #5, que ha sido modelado con éxito. En el gráfico superior de cada figura hemos dibujado los modelos encontrados para los diferentes segmentos (regímenes) que han sido separados por el método del intervalo de confianza. Las ráfagas y valores atípicos detectados automáticamente están marcados con asteriscos negros.





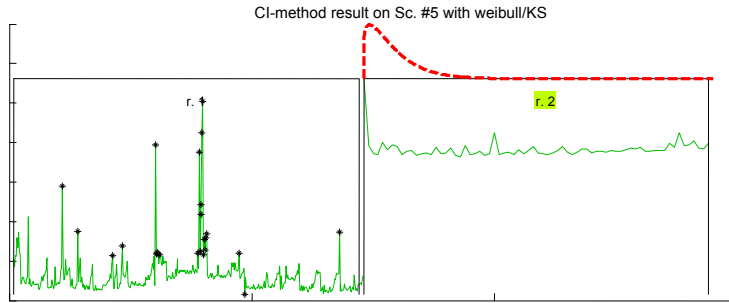


Figura 5: Gráficos de la detección de valores atípicos, ráfagas y cambios de régimen por nuestro método de intervalos de confianza (CI) con diferentes distribuciones paramétricas para el escenario #5. Los asteriscos negros corresponden a las ráfagas y los valores atípicos detectados. Todos los modelos obtenidos con éxito para los regímenes detectados por el algoritmo (líneas rojas sobre los segmentos marcados en verde) se representan con la misma relación de aspecto que la línea verde de retardos, que han sido estirados para proporcionar una comparación equitativa de las pdfs.

2.3 Modelado de retardos temporales con las distribuciones Log-normal y Log-logística

Una vez analizados los retardos temporales existentes en el bucle sensorial, vemos que las distribuciones de densidad de probabilidad paramétricas que tienen mejores opciones de encajar son las que tienen una cola larga y pronunciada (las llamadas *heavy-tailed*). Para poder justificar el uso de estas distribuciones marginales de probabilidad y seguir mejorando el modelado, nos enfocamos a continuación en diseñar una mejor detección de cambios de regímenes y ráfagas, y en elegir la distribución de probabilidad que encajara satisfactoriamente en una mayoría de escenarios reales.

Hipótesis Log-normal

Según hemos visto en el apartado anterior, una buena elección podría ser la distribución Log-normal de tres parámetros está definida por (γ, μ, σ) [61]. Para una estimación de calidad de los parámetros de esta distribución en base a un conjunto de datos, empleamos una versión modificada del MLE (*Maximum Likelihood Estimator*) para tratar con el parámetro del *offset*, basado en la resolución de esta ecuación [62] sobre los datos $(\{x_i\})$:

$$\theta(\gamma) = \ln(x_r - \gamma) - \frac{1}{n} \sum_1^n (x_i - \gamma) - k_r \left\{ \frac{1}{n} \sum_1^n \ln^2(x_i - \gamma) - \left(\frac{1}{n} \sum_1^n \ln(x_i - \gamma) \right)^2 \right\}^{1/2}$$

(ec.2)

donde n es el tamaño de la muestra, γ el desplazamiento y $k_r = \Phi^{-1} [1/(n+1)]$, siendo Φ la función distribución normal estándar.

Si disponemos de una estimación del desplazamiento $\hat{\gamma}$, los otros parámetros $\hat{\mu}$ y $\hat{\sigma}$ se pueden obtener mediante MLE transformando la muestra a una distribución normal. Por tanto, al igual que ocurre con otras distribuciones, la estimación del desplazamiento resulta crítica para el modelado.

Hay que tener en cuenta un tamaño mínimo de la muestra para realizar una estimación aceptable de ese desplazamiento, para evitar una insuficiente representación de la parte izquierda de la distribución. Para calcular este número de muestras mínimo aceptable, se ha calculado el error medio entre un desplazamiento conocido y el estimado con diferentes tamaños de muestra, usando un conjunto de distribuciones teóricas de cola larga: Log-normal, Log-logística, Exponencial, Gaussiana truncada a la izquierda, Pareto generalizada y Weibull. En la figura 6 se muestra el error (curvas en cian) para 1000 estimaciones del desplazamiento, por distribución y tamaño de muestra. La curva roja de la figura se corresponde con la moda del error (se usa la moda en lugar de la media debido a la asimetría de las curvas). Como se observa, un tamaño mínimo de muestra adecuado puede estar alrededor de 20.

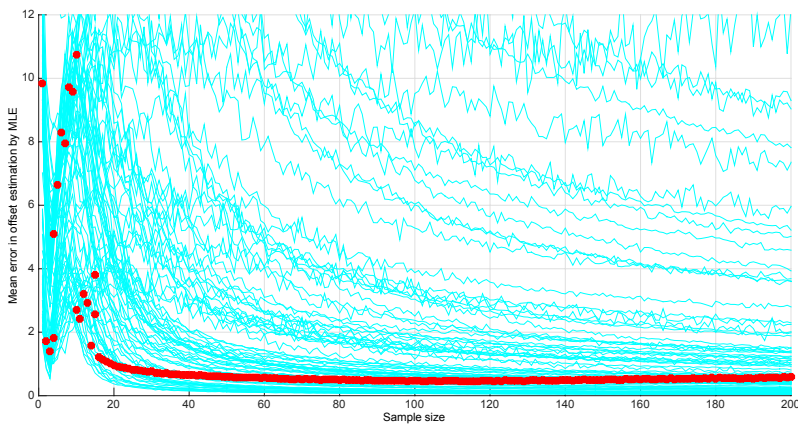


Figura 6: Error medio en la estimación del desplazamiento (cian) con un conjunto de distribuciones de cola larga, y su moda (rojo). Aproximadamente a partir de un

tamaño de muestra igual a 20 no encontramos mejoras significativas en el error de estimación.

Tras realizar la estimación de los parámetros de la distribución, necesitamos evaluar si la muestra proporciona suficiente evidencia contra nuestra distribución Log-normal de cola larga, para lo que se requiere un test de hipótesis. Hemos elegido el test de Anderson-Darling (AD), una de las herramientas estadísticas más potentes para detectar las desviaciones con respecto a una distribución normal [63], que hemos necesitado adaptar a nuestra hipótesis Log-normal.

Además de utilizar dicho test de bondad de ajuste, en esta tesis hemos ideado una medida de la compatibilidad de los escenarios con una distribución dada, que hemos llamado "ventana deslizante" o *sliding-window*. Esta medida sirve para determinar el grado de compatibilidad de escenarios reales con el modelado en base a la distribución escogida. Se trata de ver qué porcentaje de observaciones de la muestra no rechazan el test AD cuando se toma un subconjunto de la muestra dentro de una ventana de observaciones de anchura w variable.

La figura 7 muestra esta medida (*sliding-window*) para los primeros 9 escenarios reales de la figura 3 y para dos escenarios sintéticos formados por muestras tornadas aleatoriamente de distribuciones de referencia (Log-normal y Uniforme). Se puede observar que el 66,67% de nuestros escenarios tienen una alta compatibilidad ($> 65\%$) con nuestro test AD Log-normal cuando se consideran regímenes (ventanas) de tamaño razonable (menos de 80), el 22,22% tienen una considerable compatibilidad (40-50%), y el resto, sólo 11,11 %, tienen una compatibilidad baja. Por lo tanto, se puede concluir que modelar los retardos de tiempo con una distribución de probabilidad Log-normal es apropiado para una estimación *on-line* de bajo coste computacional durante ventanas (regímenes) de longitudes razonables.

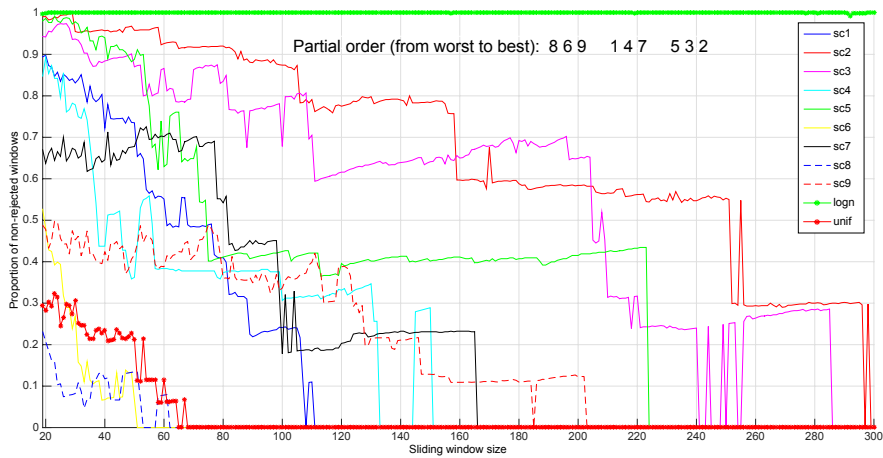


Figura 7: Comparación de la medida *sliding-window* para nueve escenarios reales diferentes tomados de la figura 3, además de para un escenario log-normal sintético (marcado en verde), y otro sintético con distribución Uniforme (marcada en rojo), que sirven para mostrar una referencia mejor y peor.

En particular, se observa que los escenarios reales #2 y #3 son especialmente compatibles con la hipótesis Log-normal y con diversos tamaños de ventana; sin embargo, en el #6, #8 y #9, los porcentajes no rechazados disminuyen por debajo del 50% para todos los tamaños prácticos de la muestra, por lo que no son tan compatibles con la hipótesis Log-normal.

Algoritmo basado en AD para modelar escenarios completos con distribución Log-normal

En esta tesis hemos diseñado un algoritmo intuitivo alternativo al *CI* que es capaz de detectar los cambios de regímenes usando el propio test AD, proporcionando una distribución Log-normal para cada régimen, y modelando cualquier escenario. El algoritmo funciona de la siguiente manera. En primer lugar, se crea un régimen de tamaño mínimo (este tamaño está dado por un parámetro llamado w) con retardos medidos en el sistema; una vez obtenido dicho régimen mínimo, se aplica el test AD: un rechazo mueve ese régimen un valor hacia adelante, olvidando el valor más antiguo, mientras que un no rechazo hace que el algoritmo intente agrandar el régimen con los siguientes valores de retardo, hasta que se añada uno que haga rechazar el test; cuando finalmente se rechaza un régimen ampliado, su porción más larga no rechazada se mantiene como un régimen definitivo del escenario.

La figura 8 muestra el comportamiento de este algoritmo de modelado (no confundir con el resultado de la medida “*sliding window*”; aquí el eje horizontal indica el índice del valor de retardo, no el valor del parámetro w) en los 9 primeros

escenarios reales de la figura 3, además de los escenarios teóricos Log-normal y Uniforme. Los mismos tres escenarios que no eran compatibles con nuestra hipótesis Log-normal en el apartado anterior (#6, #8 y #9) y el escenario uniforme teórico muestran aquí una baja proporción de retardos no rechazados (25% - 45%). El resto tiene una proporción muy alta de retardos no rechazados (> 60%) y, por lo tanto, tienen una representación adecuada con nuestro modelo de cola larga. En definitiva, el conjunto formado por el algoritmo de detección de regímenes basado en AD y la distribución Log-normal constituyen un modelo de bajo coste computacional y con resultados muy satisfactorios para una mayoría de escenarios reales.

En la figura 9 se muestran dos escenarios reales representados en el tiempo y como histogramas y pdfs cuando usamos $w = 20$. El escenario #5 muestra un ajuste mejor y más largo con una Log-normal, mientras que el escenario #9 muestra regímenes más cortos.

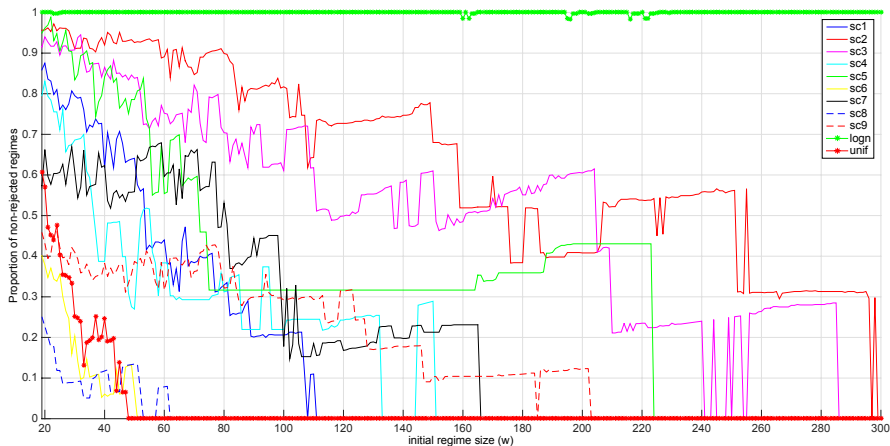


Figura 8: Resultados de nuestro algoritmo basado en el test AD al modelar escenarios de la figura 3 con diferentes valores de w .

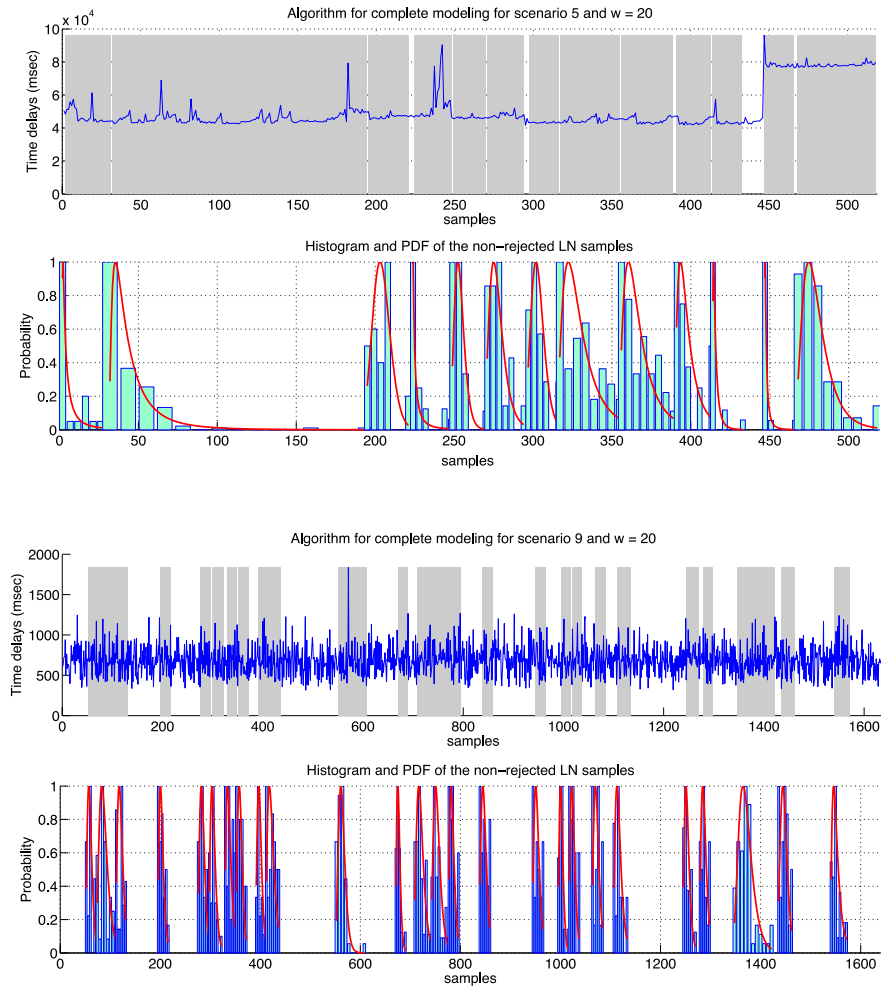


Figura 9: Regímenes detectados y modelos Log-normal encontrados por nuestro algoritmo basado en el test AD en escenarios reales #5 (arriba) y #9 (abajo). Los gráficos superiores muestran las muestras de retardos (regímenes sombreados), mientras que los gráficos inferiores muestran sus histogramas y pdfs.

Hipótesis Log-logística

Otra de las contribuciones de esta tesis ha sido la propuesta de la distribución Log-logística como una mejor aproximación al modelado de los retardos que la más clásica Log-normal (a costa de un mayor coste computacional) [55] [64].

La distribución Log-logística unidimensional triparamétrica puede ser definida por la tripla (a, b, c) . Su densidad de probabilidad y funciones de distribución acumulativa son, respectivamente [65]:

$$f_x(x; a, b, c) = \frac{[(x-a)/b]^{-1/c}}{c(x-a)\left\{1+[(x-a)/b]^{-1/c}\right\}^2} \quad (ec.3)$$

$$F_x(x; a, b, c) = 1 - \left[1 + ((x-a)/b)^{-1/c}\right]^{-c}$$

donde $a, b, c \in \mathbb{R}$, $x \geq 0$, $a \geq 0$, y $b, c > 0$ (la distribución Log-logística biparamétrica, más clásica, tiene $a = 0$). El parámetro a se denomina desplazamiento o ubicación, ya que establece un límite mínimo para x , mientras que los parámetros b (escala) y c (forma) definen el aspecto general de la función. Esta distribución tiene algunos rasgos especiales que hacen que sea más difícil de tratar que otras (por ejemplo, que la Log-normal), a saber, que sus primeros momentos no siempre existen. Nosotros forzamos que $c < 1$ para que exista su media, lo que concuerda con las formas experimentales observadas en nuestros escenarios. Si, además, $c < 1/2$, también existiría su varianza.

Para estimar los tres parámetros de la distribución se emplea el método MLE ajustado en sus valores críticos para el caso en que se están estimando los parámetros de la distribución desde los mismos datos. Además, es necesario estudiar sus limitaciones con muestras pequeñas, presentar una adaptación del test logístico de Anderson-Darling [66] a nuestro modelo triparamétrico, y definir una medida intuitiva de la compatibilidad de un escenario real con esta hipótesis Log-logística (adaptando la *sliding-window*).

Al igual que en el caso de la hipótesis Log-normal, es necesario estudiar las limitaciones del MLE para muestras pequeñas. Se ha procedido de la misma manera, y, en base a la figura 10, podemos deducir que a partir de un tamaño de muestra de 20 se logra el error mínimo en la estimación del desplazamiento; por lo tanto, podemos establecer 20 como el número mínimo de valores de retardo requeridos para estimar el modelo Log-logístico correctamente.

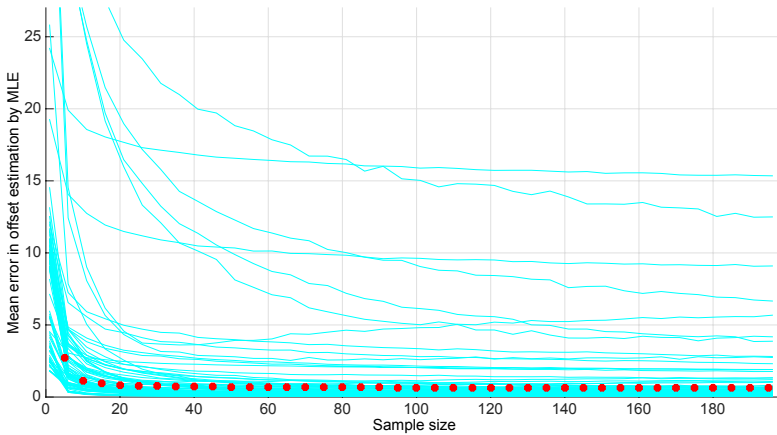


Figura 10: Estudio del error en la estimación del parámetro a (desplazamiento) de la Log-logística triparamétrica para un número de muestras generadas a partir de diversas distribuciones de cola larga. En rojo, la moda de todas las curvas. A partir de un tamaño de muestra de tamaño 20, no hay disminución significativa de errores.

También se necesita un test de hipótesis para evaluar si una muestra proporciona suficiente evidencia contra la distribución Log-logística una vez que se ha ajustado. Hemos elegido la prueba de Anderson-Darling (AD), inicialmente definida para distribuciones logísticas biparamétricas, que tienen la siguiente densidad y funciones acumulativas:

$$f_x(x; \mu, \sigma) = \frac{\exp(-(x - \mu) / s)}{\sigma(1 + \exp(-(x - \mu) / s))^2} \tag{ec. 4}$$

$$F_x(x; \mu, \sigma) = 1 / [1 + \exp(-(x - \mu) / s)]$$

Se puede apreciar la similitud entre esta distribución acumulativa y la Log-logística de la ec. 3. Para aplicar el test AD a la muestra Log-logística paramétrica $\{x_i\}$ necesitamos transformar la muestra y también los parámetros (a, b, c) obtenidos mediante el ajuste MLE.

En primer lugar, es fácil ver que la muestra desplazada $\{x_i - a\}$ debe corresponder con una Log-logística biparamétrica. También, es bien sabido en teoría básica de probabilidad que el logaritmo de los valores extraídos de una Log-logística biparamétrica puede considerarse extraído de una logística biparamétrica. Por lo tanto, el equivalente logístico biparamétrico de nuestra muestra log-logística de tres parámetros es simplemente $\{y_i\} = \log(\{x_i\} - a)$.

Ahora necesitamos calcular los dos parámetros logísticos biparamétricos μ y σ del ajuste que correspondería al ajuste MLE (a, b, c) de la muestra original. Tanto μ



como σ pueden deducirse del hecho de que si la función de distribución de la variable X es una Loglogística triparamétrica (a, b, c), la función de distribución de la variable $Z = \log(X - a)$ debe ser una logística biparamétrica que es igual a la distribución de X para cada $x \in X$. Consecuentemente, sus cdfs deben coincidir:

$$\frac{1}{1 + \left(\frac{x-a}{b}\right)^{-1/c}} = \frac{1}{1 + e^{-\left(\frac{z-\mu}{\sigma}\right)}}$$

$$\left(\frac{x-a}{b}\right)^{-1/c} = e^{-\left(\frac{\log(x-a)-\mu}{\sigma}\right)} \quad (ec. 5)$$

$$c^{-1}(\log(x-a) - \log(b)) = \sigma^{-1}(\log(x-a) - \mu)$$

$$\mu = \log(b), \sigma = c$$

Con esto hemos definido la estimación de la distribución y el test de hipótesis. La figura 11 muestra la medida de la *sliding-window* basada en la Loglogística para los escenarios reales de la figura 3, y también para un escenario log-logístico simulado que sirve de referencia. De media, el 67% de nuestros escenarios tienen una compatibilidad no despreciable (medida $> 50\%$) con la suposición log-logística usando un valor de ventana de $w < 80$. Si comparamos la misma medida de compatibilidad con *sliding-window* cuando se usa una suposición Log-normal de tres parámetros (figura 7) se puede ver la mayor idoneidad de la Loglogística triparamétrica con respecto a la Log-normal, aunque los escenarios #6, #8 y #9 no son muy compatibles con ninguna de ellas. De media, la Log-normal logra una buena compatibilidad sólo para $w < 60$, en lugar de $w < 80$ conseguido con la Loglogística. Es decir, usando la distribución de probabilidad Log-logística no sólo tenemos mayor porcentaje de escenarios que se ajustan mejor, sino que, además, las ventanas se expanden más (80 vs. 60), por lo que con el mismo modelo Log-logístico se pueden explicar más observaciones, es decir, regímenes más largos.

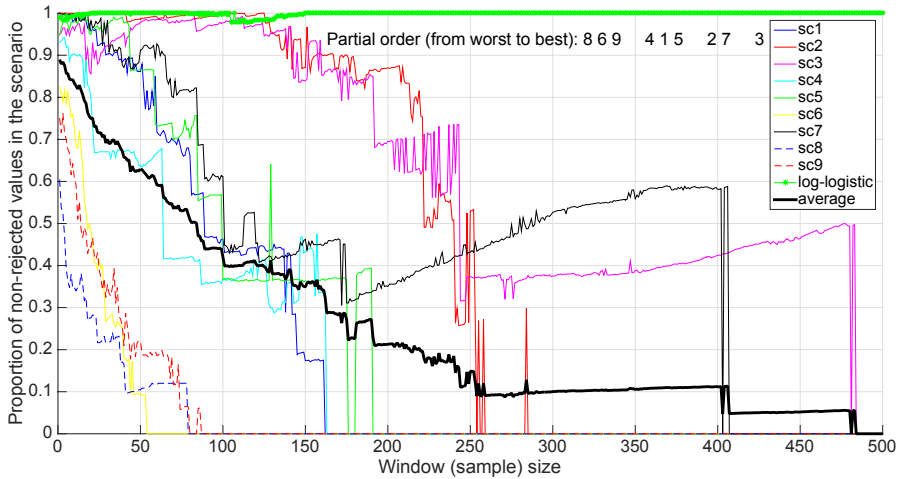


Figura 11: Comparación de la medida de la *sliding-window* para los escenarios de la figura 3 y una referencia log-logística teórica (marcada en verde). La línea gruesa negra es el promedio de todas las líneas coloreadas.

Algoritmos de modelado on-line de secuencias completas de retardos basados en la hipótesis Log-logística: stateless y state-based

El trabajo de modelado de retardos en esta tesis se ha completado con un algoritmo simple e intuitivo que detecta cambios abruptos en la secuencia de retardos y produce diferentes modelos log-logísticos para cada uno de esos regímenes, así como con una versión mejorada que utiliza el historial de retardos para construir mejores modelos. Ambos usan el ajuste y los procedimientos del test AD descritos anteriormente, y apuntan a ser soluciones *on-line* al problema del modelado para situaciones prácticas [64].

1) Algoritmo *stateless*

Este algoritmo es el mismo que se ha explicado anteriormente para la Log-normal, y usa el test AD como asesor: detecta si el modelo Log-logístico actual deja de ser el mejor para explicar nuevos valores de retardo entrantes (es decir, si se debe configurar un nuevo modelo). En otras palabras, el test AD se utiliza para detectar cambios abruptos, separando regímenes que tengan diferentes parámetros de la distribución.

En primer lugar el algoritmo recoge una secuencia de retardos de un tamaño mínimo. Ese tamaño viene dado por un parámetro llamado s , comenzando desde 20. A continuación se calcula el modelo Log-logístico de tres parámetros mediante el MLE y se aplica el test de AD por primera vez: un rechazo desliza la secuencia de tamaño s un valor de retardo hacia adelante, olvidando su valor más antiguo y añadiendo uno nuevo, mientras que un no rechazo fija la muestra testeada como un

nuevo régimen de la señal que se ampliará con los siguientes valores de retardo. Esta ampliación continuará hasta que se agregue un valor de retardo que haga que el test rechace el régimen formado hasta el momento; cuando esto ocurra, el régimen se registra como un régimen definitivo de la señal.

La figura 12 resume el comportamiento de este algoritmo de modelado en los 9 primeros escenarios reales de la figura 3 para diferentes valores de s . No debe confundirse el eje horizontal de esta figura con el de la figura 7 ó 11. Podemos observar en la figura 12 que los mismos escenarios que no eran compatibles con nuestra suposición log-logística (#6, #8 y #9) exhiben aquí una alta proporción de rechazos (más del 60%). El resto tiene una proporción muy alta de retardos no rechazados ($> 60\%$) cuando se usa $s < 60$, y por lo tanto tienen una representación adecuada con una secuencia de modelos Log-logísticos, es decir, como una secuencia de regímenes estacionarios.

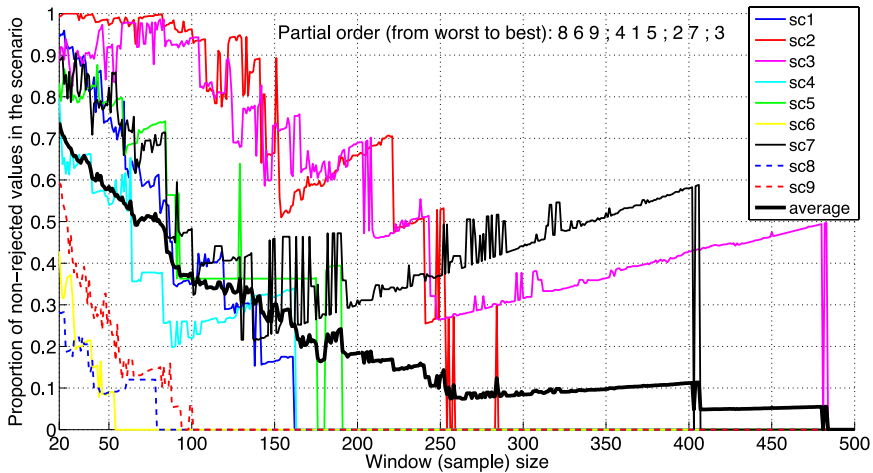


Figura 12: Resultados del algoritmo *Stateless* al modelar escenarios de la figura 3 para diferentes valores del parámetro s .

2) Algoritmo *state-based*

Este algoritmo es una extensión del algoritmo descrito anteriormente (*stateless*) para mejorar el modelado a través de la reutilización de regímenes detectados previamente (que llamamos aquí *estados* del sistema). Básicamente, este segundo algoritmo comprueba si los valores de retardo que no han sido incluidos en un nuevo régimen podrían anexarse al final de regímenes previamente definidos, o, en otras palabras, intentamos detectar cuándo el sistema visita estados (regímenes) previamente terminados.

El algoritmo funciona de la misma manera que la versión *stateless*, salvo que mientras los últimos retardos de la secuencia no pasen el test AD, verificamos si

añadiéndolos al final de un régimen previo sí hacen pasar el test (más concretamente, sólo usamos los últimos α valores de retardo obtenidos en la secuencia para esto, siendo α un parámetro del algoritmo). Si se demuestra que estos retardos amplían una serie de regímenes anteriores, se selecciona como estado revisado el que tiene mejor p-valor en el test AD, ya que el p-valor del test indica cuantitativamente la bondad de ajuste; de esta manera se amplía ese régimen anterior con más valores de retardo nuevos, y se continúa así hasta que no pase el test.

3) Resultados experimentales reales de ambos algoritmos

Para ilustrar el comportamiento efectivo de ambos algoritmos, seleccionamos dos escenarios reales (#5 y #7). Los resultados detallados usando $s = 20$ y $\alpha = 10$ para estos escenarios se muestran en las figuras 13 y 14. Las líneas verticales delgadas grises determinan el final de los regímenes fig. 13-14 izquierda). Hay que tener en cuenta que en el algoritmo *state-based* también hemos identificado los diferentes estados del sistema. Los resultados muestran claramente las ventajas de este enfoque con respecto al algoritmo *stateless*: el número de regímenes se reduce significativamente (7 vs 16 en el escenario #5 y 6 vs. 12 en el escenario #7). Como se explicó en apartados anteriores, la complejidad de ambos algoritmos es $O(n)$, por lo que el algoritmo *state-based* nos da más información, además de detectar regímenes más largos (a cambio, tiene más coste computacional medio y mayor consumo de memoria).

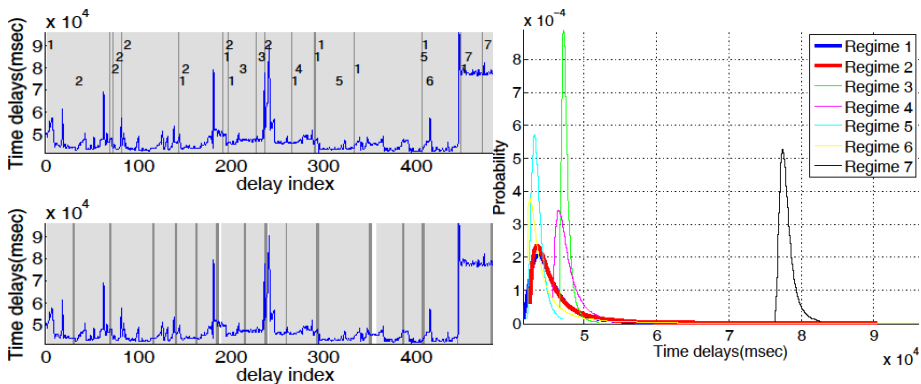


Figura 13: Regímenes detectados por el algoritmo *State-based* (izquierda-arriba) y *Stateless* (izquierda-abajo) y modelos Log-logísticos encontrados por el primero en el escenario real #5 (derecha).

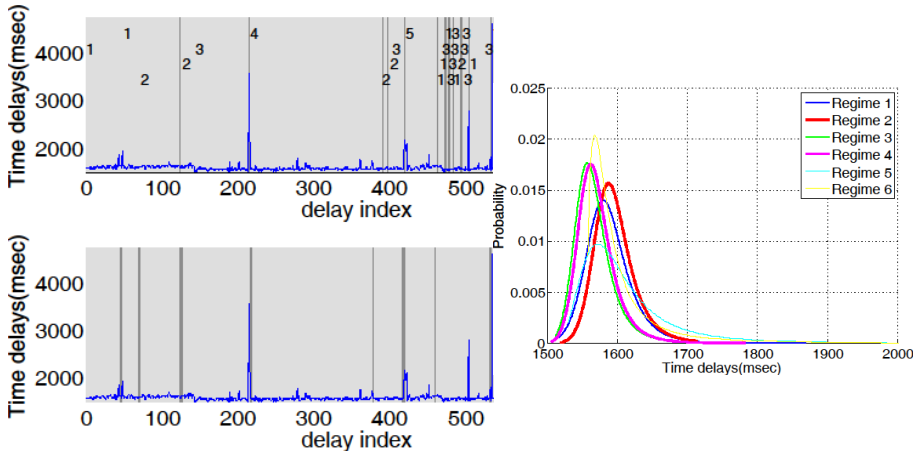


Figura 14: Regímenes detectados por los algoritmos *State-based* (izquierda-arriba) y *Stateless* (izquierda-abajo) y modelos Log-logísticos encontrados por el primero en el escenario real #7 (derecha).

Viendo estos resultados, podemos concluir que estos algoritmos más el modelado con una distribución Log-logística son una solución viable con un coste computacional razonable para la estimación *on-line* de retardos temporales en telerobots controlados por Internet.

3 REGULACIÓN DEL FLUJO DE DATOS

En este capítulo describimos un sistema de procesamiento de datos para la regulación automática de la cantidad de información que se transmite entre el teleoperador y el robot móvil conectado a Internet [67, 68, 69]. Este sistema se caracteriza porque, a pesar del comportamiento estocástico de las transmisiones, garantiza estadísticamente que la probabilidad de que dicha transmisión se realice al menos a una frecuencia predeterminada sea mayor o igual que una dada y, además, lo consigue transmitiendo la mayor cantidad de datos posible. Este sistema de regulación satisface pues los requisitos de tiempo real suave que explicamos en la introducción. Consiste en un controlador jerárquico de dos niveles: a un nivel de regulación fina, trabaja mediante la variación de la cantidad de datos sensoriales que se transmiten, y en un nivel de regulación gruesa, mediante la reconfiguración dinámica de los sensores que están activos. Nuestra propuesta supone modificaciones mínimas en el software y hardware en el camino de transmisión.

Anteriormente hemos analizado los retardos existentes en la transmisión de datos sensoriales de robots teleoperados por Internet, proponiendo modelos probabilísticos (ver capítulo 2). Ahora utilizamos estos modelos como predictores de tiempos de transmisión y construimos sobre ellos un sistema de regulación jerárquica de dos niveles capaz de variar automáticamente tanto la cantidad de datos sensoriales transmitidos como el conjunto de sensores que están transmitiendo en cada momento, para así satisfacer de forma óptima los requisitos de control en tiempo real y calidad de la aplicación. En los siguientes apartados se presenta cómo se ha implementado este controlador jerárquico. Los resultados muestran que se obtiene un mejor rendimiento combinando ambos que cuando se usan por separado, y mucho mejor rendimiento si no se usa controlador.

Parte de este trabajo (particularmente la regulación gruesa y gran parte del interfaz web) ha sido realizado por D. Ángel Martínez Tenor, ingeniero contratado en el mismo proyecto de investigación que ha amparado la presente tesis.

Este capítulo comenzará con una introducción a estos métodos de regulación y haciendo un repaso al estado del arte. Posteriormente se analizará el por qué de la

necesidad de este tipo de herramientas. Finalmente, presentamos los dos métodos ideados y sus experimentos correspondientes.

3.1 Introducción y estado del arte

Como ya se ha explicado en capítulos anteriores, si un nodo conectado a Internet quiere recibir repetidamente datos de múltiples fuentes, como es el caso en aplicaciones de monitorización remota, *broadcasting* multimedia, teleoperación, etc., donde hay una serie de sensores de los que se desea recibir información periódicamente, la estocasticidad en los tiempos de envío y recepción de datos hacen difícil cumplir tiempos de transmisión restringidos. Uno de los posibles mecanismos para poder cumplir esos requisitos, el que hemos denominado regulación gruesa al principio de este capítulo, es decidir automáticamente sobre qué fuentes están activas en cada momento para transmitir datos, dependiendo del comportamiento de los retardos. En la literatura científica no se encuentran desarrollos en este sentido, pero sí hay algunos en el campo del desarrollo en ingeniería, publicados como patentes. Estas patentes, sin embargo, abordan este mecanismo con un enfoque distinto al propuesto aquí, ya que no se centran en recibir el mayor número de datos ni regulan la cantidad de información, sólo la apariencia del interfaz que muestra los datos recibidos. A continuación mencionamos las más relevantes.

En el documento [70] se presenta un interfaz de usuario adaptativo que muestra más o menos cantidad de información textual para facilitar la lectura de la misma. La regulación de esa información puede ser automática o en función de las entradas del usuario.

En el documento [71] se plantea una interfaz de usuario adaptativo y no impuesto por la industria, como suele ser habitual. Esta interfaz se adapta según las estadísticas de uso de los comandos que el usuario suele aplicar: cuanta más soltura, más características del interfaz aparecen, y viceversa. Por tanto, es un método adaptativo y automático para desactivar/activar características en una interfaz.

El documento [72] se refiere a un interfaz hombre-máquina (HMI) adaptativo basado en el perfil psicológico del usuario y de su estado actual de bienestar (medido con el análisis de su comportamiento —nº de pulsaciones, movimientos de ratón, intervalos entre llamadas...—, y otras medidas biométricas). La idea es proporcionar un HMI adaptado al usuario y a los diferentes estados del mismo, de manera que se varíe la forma de representar la información y se pueda mejorar el rendimiento del usuario dentro del sistema.

En el documento [73] se optimizan automáticamente las presentaciones de un GUI (*Graphical User Interface*) de un dispositivo de punto de venta respecto a ciertos parámetros de rendimiento, y usando técnicas de *IA*; la justificación es que los diseños de un GUI suelen orientarse a operarios poco entrenados, resultando poco eficientes para los más entrenados.

En el documento [74] se propone la adaptación y agregación dinámica de contenidos en interfaces de usuario, en el ámbito de sistemas ERP (*Enterprise Resource Planning*), de forma que se añaden componentes a petición siempre que se tengan una serie de privilegios.

En el documento [75] se modifica la disposición de los elementos de un interfaz para adaptarse a la lateralidad (uso natural predominante de una mano u otra) en dispositivos portátiles de mano (*handheld*) con pantallas táctiles.

En el documento [76] se plantea la adaptación de los recursos del usuario en a las habilidades del mismo en una interfaz (puede ver, leer, oír, ...) en función de la recopilación de cierta información de contexto (velocidad, luz, temperatura, ...).

En el documento [77] se realiza la adaptación de la interfaz de usuario de un ordenador (colores, fondo, etc.) a las características (por ejemplo, de aspecto) de un dispositivo que se le conecte. Por ejemplo, según el color del ratón, el ordenador adapta los colores de su interfaz.

Finalmente, [78] muestra una interfaz de usuario adaptable al manejo físico del dispositivo que la implanta, incluyendo la posibilidad de interactuar con un objeto externo.

En cuanto a la regulación fina, es decir, al cambio de la cantidad de datos a transmitir en un cierto canal, existen diversas aproximaciones en la literatura, aunque, como se explicaba en el capítulo de introducción, vienen de otros campos distintos a la teleoperación de robots, y por tanto, tienen distintas suposiciones y requisitos que cumplir (por ejemplo, [79], [80]). Por otra parte es un problema tan común [3] que también existen diversos desarrollos publicados en forma de patente, como los que siguen.

En [81] se presenta un método para regular la operación del *buffer* de decodificación de un sistema de transmisión en el que la tasa de transmisión se ajusta según los cambios en los retardos existentes y la ocupación del *buffer* de codificación, para así evitar *overflows* y *underflows* en el citado *buffer*.

En las patentes [82] se regula el ancho de banda (aumentándolo o disminuyéndolo) en una plataforma de transporte multi-servicio; para ello, se compara el ancho de banda ocupado por el flujo actual con el ancho de banda distribuido.

Las patentes [83], referidas a aplicaciones de transferencia de voz, proponen una regulación basada en el control de la tasa de paquetes perdidos. Así, es posible tener diferentes tasas de paquetes perdidos dependiendo de que el modo de

codificación/decodificación a usar sea más o menos riguroso: si el modo es poco riguroso, se aumenta la tasa de paquetes perdidos, disminuyendo así el retardo.

La patente [84] presenta un servidor de medios con dispositivos QoS (*Quality of Service*, es decir, que tienen capacidades para adaptarse a distintos requisitos de calidad del servicio) y no-QoS conectados, el cual limita el ancho de banda a estos últimos para garantizar las transmisiones de los primeros.

La regulación de la cantidad de información a transmitir abordada por todos estos trabajos se trata también desde métodos estadísticos, por ejemplo en [85], orientada a tecnologías *wireless*; [86], para transmisión de vídeo en redes *narrow-band*, y [87], donde se utiliza SPC (*Statistical Process Control*, técnicas para el control estadístico de la calidad de los procesos) con el objetivo de saber si una serie de eventos están cumpliendo los SLAs (*Service Level Agreement*) que se tienen como objetivo; si no es así, se toman una serie de acciones para lograr dicho cumplimiento.

Ni en la literatura científica ni en el ámbito de las patentes se han encontrado desarrollos que unan ambos tipos de regulación, fina y gruesa.

3.2 Regulación fina: variación de la cantidad de datos sensoriales

El primer nivel de nuestro sistema de regulación jerárquico consiste en un algoritmo de regulación para cada sensor i . Dicho regulador está encargado de decidir qué cantidad de datos sensoriales se solicita en la siguiente petición a ese sensor si queremos completar esa transmisión en τ_i o menos tiempo con una probabilidad I_i o mayor. Cada sensor tiene un número finito de densidades (cantidades posibles de datos solicitados¹), y mantenemos, para cada una, un modelo de los retardos de transmisión, basado en los explicados en el capítulo 2.

Nuestro algoritmo de regulación de densidad es heurístico y está basado en un controlador proporcional clásico que intenta mantener la probabilidad de alcanzar los tiempos deseados τ_i cerca de I_i y, simultáneamente, aumentar la densidad tanto como sea posible. Tiene como entrada la densidad δ_0 que se solicitó en la última petición, y emite la densidad δ que se pedirá en la siguiente petición. Si no hay suficiente información en el modelo de los retardos para predecir $\pi_i(\delta_0)$, su salida será $\delta = \delta_0$, lo que lleva a acumular más datos para dicha predicción; de lo contrario, puede suceder que: i) $\pi_i(\delta_0) < I_i$, ii) $\pi_i(\delta_0) > I_i + \alpha$ o iii) $\pi_i(\delta_0) \in [I_i$,

¹ Consideramos una densidad como un número natural, siendo 0 la menor cantidad de datos sensoriales (por ejemplo, la resolución mínima de un sensor de cámara) y D_i-1 la máxima, teniendo así D_i densidades para el sensor i .

$\Pi_i + \alpha]$, donde $\alpha \in (0,1)$ es un número pequeño (usaremos $\alpha = 0.05$). En el caso i) no se satisface el objetivo de tiempo real suave, por lo que el algoritmo debe disminuir la densidad, siempre que $\delta_0 > 0$, para reducir los tiempos de transmisión; en el caso ii) se satisface el requisito de tiempo real suave para ese sensor, y por lo tanto se puede intentar aumentar la densidad, siempre que $\delta_0 < D_i - 1$, tratando así de mejorar el rendimiento del operador; finalmente, en el caso iii) estamos funcionando bien, pero cerca de no hacerlo, por lo tanto, mantenemos la densidad actual ($\delta = \delta_0$).

En los casos i) y ii) se calcula un cambio de densidad $\Delta\delta$ que es proporcional al "error" en probabilidad, $e = \Pi_i - \pi_i(\delta_0)$. Más concretamente, en el caso i) cambiamos a una nueva densidad $\delta = \min(\delta_0 - 1, \delta_0(1 - e))$. Si $\delta = 0$ o si todavía no se han modelado los retardos para predecir $\pi_i(\delta)$, o si $\pi_i(\delta) \geq \Pi_i$, devolvemos esa densidad δ ; de lo contrario, repetimos el salto a densidades más bajas pero ahora comenzando en $\delta_0 = \delta$. En el caso ii), en primer lugar, establecemos un cambio de tamaño $\Delta\delta = -(D_i - 1 - \delta_0)e$ y luego calculamos la nueva densidad como $\delta = \delta_0 + \Delta\delta$. Si $\Delta\delta = 0$ o si no tenemos una predicción $\pi_i(\delta)$, o si $\pi_i(\delta) \geq \Pi_i$, devolvemos esa densidad δ ; de lo contrario, $\Delta\delta = \Delta\delta - 1$ y repetimos.

Experimentos con algoritmo de regulación fina

En estos experimentos hemos usado datos reales recogidos de una *webcam* conectada a Internet y solicitados repetidamente con diferentes resoluciones. Las cámaras web son el sensor más adecuado para estas aplicaciones debido a que tienen un alto número de densidades ($D = 41$ en nuestros experimentos; cada una corresponde a una resolución de imagen RGB, desde 160×120 hasta 640×480). También es un sensor muy común en los robots móviles de hoy en día.

La figura 15 muestra los resultados obtenidos. Cada punto coloreado en ese plano es el resultado de un experimento que consiste en hacer una petición al sensor varias veces utilizando o no un método de regulación fina. La abscisa indica la densidad media solicitada desde el sensor a lo largo de todo el experimento, medida como el número de bytes transmitidos y ponderado por el tiempo durante el cual cada densidad estaba activa, mientras que la ordenada es el porcentaje de tiempo durante el cual los tiempos de transmisión fueron más cortos o igual al requisito τ_1 (aquí $\tau_1 = 0.3s$). La línea negra punteada gruesa es el requisito de probabilidad Π_1 (90% en estos experimentos): los puntos por encima o en esa línea satisfacen el par de requisitos de tiempo real suave (τ_1, Π_1), siendo los experimentos correspondientes mejores cuanto mayor sea la abscisa mientras siga cumpliéndose el requisito de la ordenada.

Los puntos rojos son experimentos, uno por densidad, cada uno con 1951 peticiones con la misma densidad cuando no se usa ningún regulador. Sólo las

densidades pequeñas (correspondientes a RGB 280x210 o menores) satisfacen (τ_1 , Π_1) en ese escenario. Hay que tener en cuenta que no podemos saber qué densidades tenían antes de hacer esta exploración exhaustiva —e incluso su situación particular de red, carga de software, etc.—, por lo que elegir por ejemplo una densidad constante de RGB 280x210, no sería una solución válida en un escenario real con este sensor. También hay que tener en cuenta que cualquier algoritmo de regulación logrará densidades y probabilidades cercanas a los puntos rojos, ya que todas las densidades posibles están ahí.

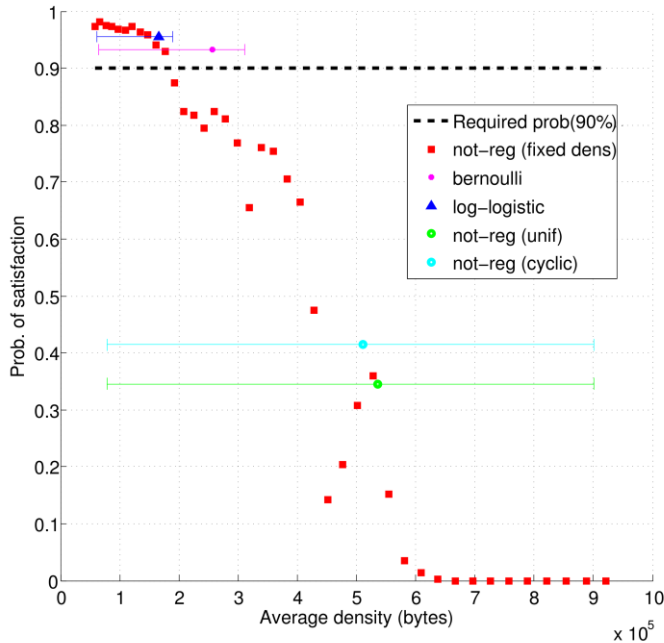


Figura 15. Estudio del algoritmo de regulación fina con dos métodos de modelado y con respecto al ajuste de una densidad constante en cada solicitud desde una cámara remota. El punto cuadrado rojo más a la derecha por encima de la línea de requerimientos de probabilidad del 90% corresponde a una resolución de la webcam de RGB 280x210 píxeles. Los rangos horizontales indican densidades ya exploradas.

La figura también muestra puntos correspondientes a otros cuatro experimentos: uno que seleccionó una densidad uniformemente aleatoria en cada petición (para 972 peticiones, etiquetado como "unif"), uno que seleccionó todas las densidades en orden, cíclicamente (1017, "cyclic"), uno con el regulador fino que hemos explicado en el apartado anterior pero utilizando como modelo de retardos

un simple predictor de probabilidades Bernoulli² (1379), y uno con el mismo regulador y el predictor Log-logístico usado en el algoritmo *stateless*, ambos descritos en el capítulo 2 (1705). Todos estos experimentos con reguladores se han realizado con tiempos de transmisión que se comportan estadísticamente como los datos reales utilizados para la curva roja. Como se muestra en la figura, nuestro algoritmo de regulación, tanto con un modelo trivial de Bernoulli como con el modelo más sofisticado Log-logístico + *stateless*, satisface el requisito de tiempo máximo permitido en las transmisiones en tiempo real, pero lo que es más importante, cuando se usa el predictor Log-logístico, las densidades solicitadas nunca violan el requisito en tiempo real, lo que sí sucede si usamos el predictor de Bernoulli (como se ve en el *boxplot* de Bernoulli, este modelo lleva a seleccionar densidades de cámara que, en ocasiones, son demasiado elevadas).

3.3 Regulación gruesa: activación / desactivación de sensores

El segundo nivel en nuestro controlador jerárquico funciona con varios sensores a la vez. Cada uno puede ser regulado como se describe en la sección anterior o no, pero debe tener la posibilidad de ser deshabilitado temporalmente — sin emitir más peticiones—; se pondrá en pausa siempre que no cumpla su requisito τ_i durante un tiempo determinado (durante un número de peticiones consecutivas).

En este nivel debemos definir $n > 1$ combinaciones de los sensores disponibles (cada combinación se llamará *ejecutivo*) para así poder ejecutar la misma tarea (por ejemplo, teleoperación del robot) de diferentes maneras y con diferentes requisitos de tiempo. Por ejemplo, un robot móvil teleoperado puede trabajar enviando datos de una cámara o de un escáner láser de distancia, o de ambos sensores pero con menos frecuencia, definiendo así $n = 3$ ejecutivos. El j -ésimo ejecutivo es formalmente una tupla $(S_j, P_j, F_j, A_j, E_j)$, donde S_j es su conjunto de sensores y el resto son conjuntos que contienen un valor por sensor de S_j ; P_j contiene pares que son requisitos de tiempo real suave para este sensor, (τ_i, Π_i) ; F_j contiene tiempos máximos permitidos para no satisfacer τ_i antes de desactivar al sensor; A_j contiene tiempos que se añaden al valor correspondiente de F_j cada vez que el sensor cambia a activo, para evitar transitorios; y E_j contiene indicadores para indicar si el sensor es esencial (es decir, no se debe desactivar siempre que podamos evitarlo). También necesitamos un orden $O = \langle x_1, x_2, \dots, x_n \rangle$ de los ejecutivos, desde el más conveniente para resolver la tarea (x_1) hasta el menos (x_n). En cualquier momento, sólo se habilitará uno, denotado por $x_{current}$; inicialmente $x_{current} = x_1$.

Varios eventos en el sistema pueden desencadenar la ejecución de nuestro algoritmo de regulación gruesa: i) cuando un sensor de $x_{current}$ está inhabilitado; ii)

² Un modelo de Bernoulli estima $p_i(d)$ como la proporción de peticiones que se han completado en t_i o menos tiempo dividido por el número total de peticiones. No representa bien la cola derecha de la distribución real.

periódicamente, cada constante de tiempo dada T_1 , para comprobar si todos los sensores siguen activos; iii) periódicamente, cada constante de tiempo dada $T_2 (> T_1)$, para intentar activar sensores previamente desactivados. Estos tiempos T_1 y T_2 no son críticos. En el caso i), si el sensor deshabilitado es esencial y hay algún otro sensor no esencial activo, este último se deshabilita para satisfacer los requisitos del primero en el futuro, mientras que si no hay otro sensor no esencial activo, el siguiente ejecutivo peor, si existe, se activará, es decir, $x_{current} = x_{current+1}$. En el caso ii), si todos los sensores están activos, cambiamos al siguiente ejecutivo mejor, si lo hay, es decir, $x_{current} = x_{current-1}$. Finalmente, en el caso iii), si hay algún sensor desactivado, se habilita para intentar recuperarlo.

La integración de este regulador grueso con el controlador fino en un esquema jerárquico y sus experimentos se muestran en el capítulo 4, donde también se explica el interfaz web desarrollado para la teleoperación de robots.

4 APLICACIÓN PARA EL CONTROL REMOTO DE ROBOTS MÓVILES POR WEB

Para la realización de experimentos, y como prueba de implementación de los métodos desarrollados en esta tesis, se ha incluido el controlador jerárquico descrito en el capítulo 3 en una interfaz de teleoperador basada en web que es altamente portable (funciona en PCs de escritorio, tabletas, teléfonos inteligentes, etc.) y no invasiva, es decir, requiere modificaciones mínimas en la red u otros componentes de software de la aplicación. El grueso de este desarrollo de la interfaz web ha sido realizado por Ángel Martínez Tenor en el mismo proyecto de investigación de esta tesis, a partir de una versión preliminar de la autora de la misma.

A continuación se presenta la funcionalidad de este interfaz de teleoperación, se describe cómo se ha diseñado y se presentan los experimentos llevados a cabo con el mismo.

4.1 Descripción de la aplicación: control remoto de robot móviles por la Web

La aplicación de teleoperación es un interfaz web para un robot móvil. Dentro del interfaz, se distinguen dos bloques: el propio interfaz de usuario y el servidor web que sirve las peticiones del mismo y lo comunica con el robot. En la figura 16 se muestra el sistema completo. El modelo Log-logístico con el algoritmo *stateless*, que fueron descritos en capítulos anteriores, están implementados dentro del interfaz de usuario, y la regulación de flujo de datos sensoriales puede estar tanto en la interfaz como en el lado del robot.

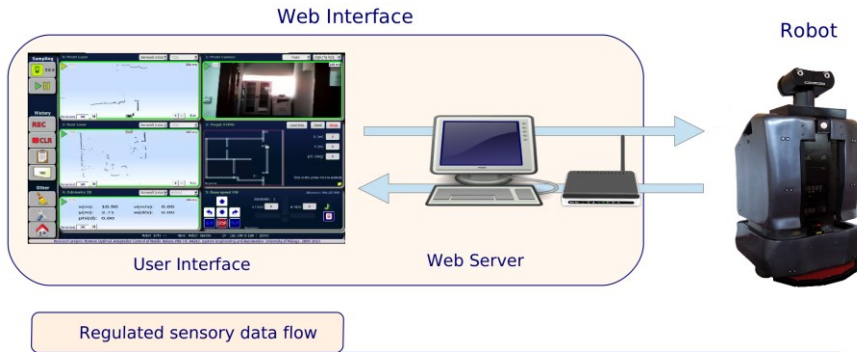


Figura 16. Esquema general del sistema para un control adaptativo y remoto óptimo de robot teleoperados por Internet.

A continuación, veremos más detenidamente la composición de cada elemento del sistema.

Robot: hardware

Para los experimentos reales hemos trabajado con el Robot Sancho [48], que tiene a bordo un ordenador Intel Core i5-650 @ 3.2 GHz x2, sobre una base Pioneer 3DX, y dispone de *webcams* estándares, y dos láseres Hokuyo UTM-30LX, así como una termo cámara (ver figura 17).



Figura 17. Robot móvil SANCHO.

Robot: software

Para la implementación software del robot, hemos trabajado sobre el sistema operativo Windows 7 Enterprise N 32 bits, con el sistema de desarrollo de software robótico BABEL [88] (ver figura 18) y la librería MRPT [89] (*Mobile Robot Programming Toolkit*).

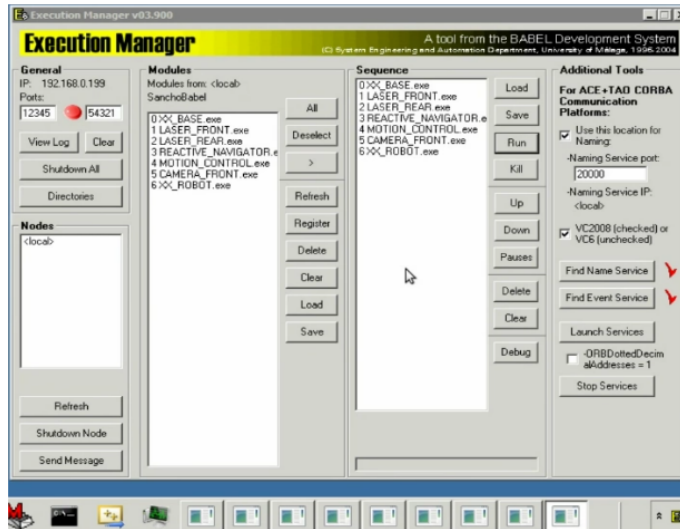


Figura 18. Plataforma BABEL para el desarrollo de módulos software robóticos.

Robot: módulos software de la arquitectura de control

En la figura 19 se muestra un diagrama con todos los sub-sistemas software empleados en nuestros experimentos. El módulo "robot" es el encargado de recibir las peticiones de la interfaz web y de distribuirlas hacia los sensores y actuadores robóticos, que se muestran en la parte inferior. Cada módulo hace uso de funcionalidades de la librería MRPT, y tiene ciertos parámetros de configuración.

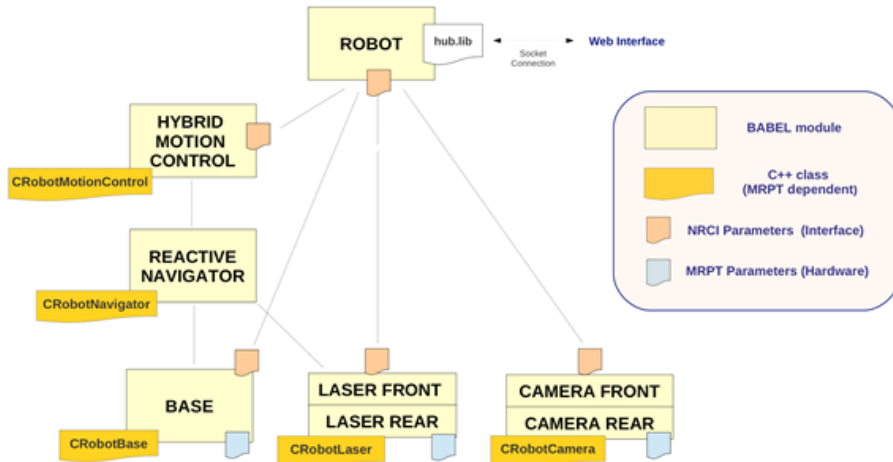


Figura 19. Diagrama de los módulos software que se ejecutan en el lado del robot.

Interfaz Web: Networked Remote Control Interface (NRCI)

En la figura 20 mostramos la interfaz web, compuesta por el interfaz de usuario, que se ejecuta en el ordenador del teleoperador, y el servidor web, que puede ejecutarse en la misma máquina, en el PC del robot o en un tercer ordenador independiente.

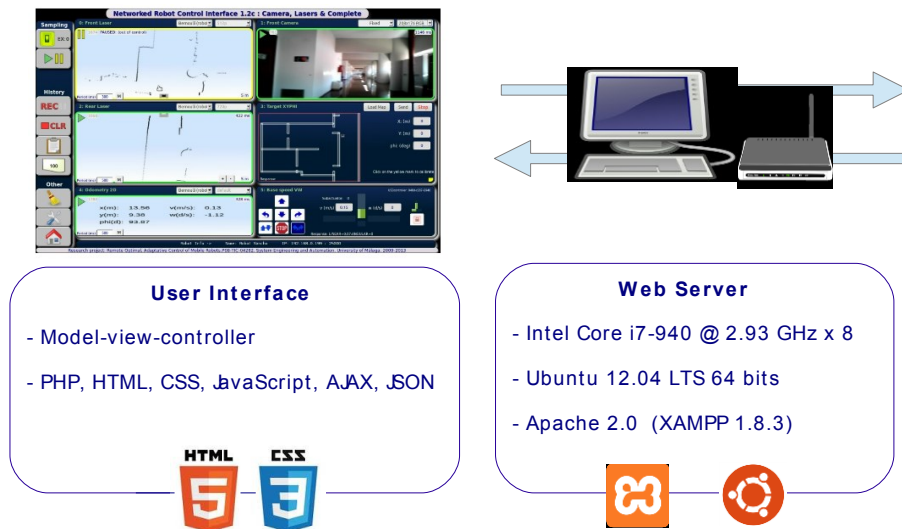


Figura 20. Interfaz web de nuestra aplicación de teleoperación, compuesta por la interfaz de control de usuario y el servidor web.

En la figura 21 se muestra la interfaz de control de usuario en la web. Esta interfaz está codificada en lenguaje javascript, y su aspecto se define en HTML y CSS. Éste es configurable para varios robots, siempre que sus capacidades sensoriales y de actuación estén recogidas en una pequeña base de datos que reside en el servidor web y que es gestionada por el código PHP ejecutado en el mismo.

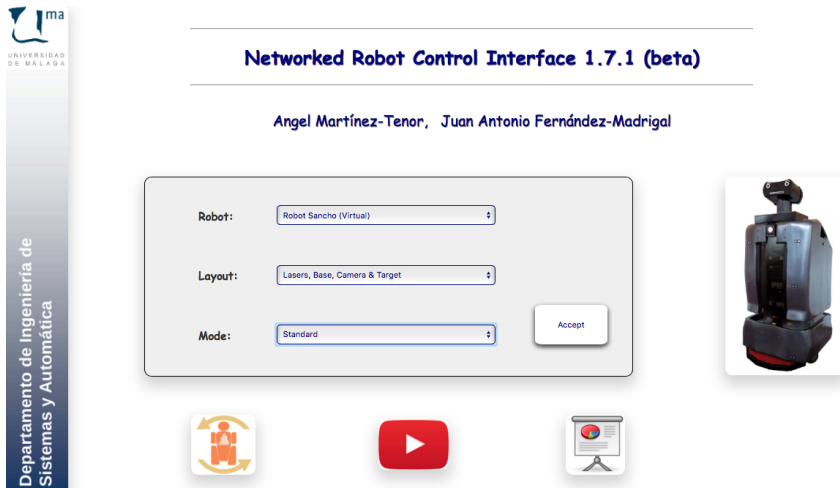


Figura 21. Interfaz de control de usuario, donde se aprecia la página de entrada que permite escoger el robot a teleoperar.

Cada sensor mostrado por la interfaz web tiene asociada una máquina de estados diferente implementada en Javascript, cuyo diagrama se muestra en la figura 22. En general, cada sensor que esté activo pedirá al robot datos sensoriales repetidamente, entrelazándose las peticiones de unos y otros por el canal de comunicación entre el cliente y el robot. La secuencia de comunicación para una petición de datos al robot sería de la siguiente manera:

1. Se compone un mensaje en el cliente (*JavaScript*), especificando la densidad de datos requerida, la cual se estima según el algoritmo de regulación y los modelos de retardos explicados en capítulos anteriores (si la regulación fina está habilitada en ese sensor y si la regulación gruesa ha activado ese sensor).
2. Se hace una petición S1 al servidor (mediante la tecnología AJAX implementada en Javascript, que está basada en comunicaciones TCP).
3. Se procesa y reenvía la petición desde el servidor web al robot mediante socket TCP (petición S2 en la figura).
4. El servidor recibe el mensaje de respuesta del robot, que se procesa y se envía de vuelta al cliente *JavaScript*.
5. En el cliente se obtienen los datos sensoriales y son mostrados en la interfaz.

En la figura se muestran los tiempos que se consideran para una petición de un sólo sensor. Estos tiempos serán la base para el modelo Log-logístico (en el caso de elegir éste) y para poder ejecutar el regulador jerárquico. El tiempo más importante es la diferencia entre la marca $t1$ y la siguiente marca $t1$, es decir, los retardos considerados corresponden a una petición completa de datos sensoriales, y por tanto incluyen tanto la estocasticidad de los componentes software como de los componentes hardware del sistema, particularmente la red de comunicaciones.

Para tener un protocolo robusto, después de cada recepción de mensaje desde el robot, éste se decodifica y analiza (se comprueba su identificador, tamaño, campos y sub campos). En caso de error, se mostrará en la interfaz un mensaje con el error detallado.

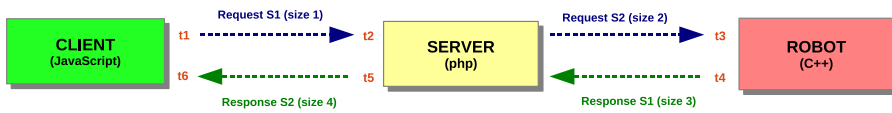


Figura 22. Secuencia del protocolo de comunicación.

Como se ha comentado, cada sensor dispone de la posibilidad de regular de forma fina la cantidad de datos sensoriales solicitados, de acuerdo con los métodos y modelos explicados en capítulos anteriores. Respecto a la regulación gruesa, el esquema utilizado de la misma se muestra en la figura 23. Ésta se compone de tareas (las que el teleoperador pretende realizar con el robot, por ejemplo navegación hacia un *target*); cada tarea se le asigna un *ejecutivo* (explicado en el capítulo anterior), que a su vez configurará los diferentes sensores y actuadores.

Si todos los sensores, o algún sensor esencial, no llegan a cumplir los requisitos de tiempo real, el ejecutivo actual se deshabilitará y se procederá a utilizar otro con requisitos más relajados, como se explicó en el capítulo anterior.

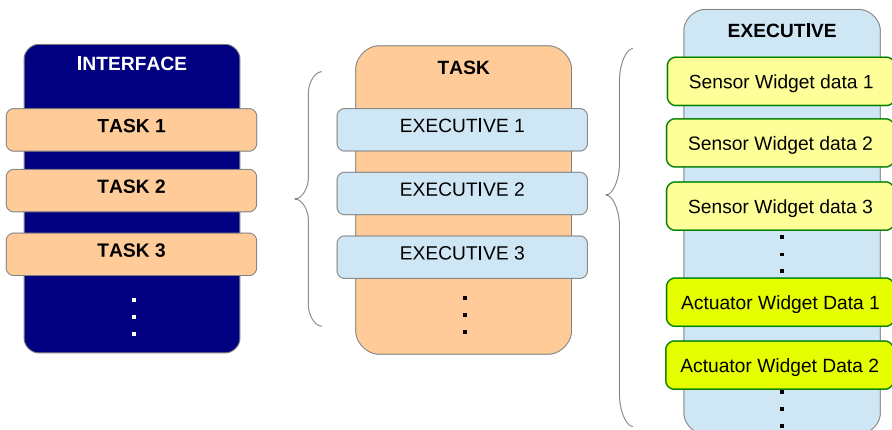


Figura 23. Estructura básica del regulador grueso en la interfaz.

Para detallar mejor la configuración de los ejecutivos, se presenta en la figura 24 cómo se han definido el ejecutivo 1 (*EX1*) y el ejecutivo 3 (*EX3*) para algunos de nuestros experimentos de la tarea de navegación, que a su vez están ordenados según sus requisitos de tiempo real asociados sean más o menos estrictos. En estos ejecutivos, se definen para cada sensor: la densidad mínima, el tiempo máximo permitido antes de desactivar al sensor y si se corresponde o no con un sensor esencial. Además, se configura qué tipo de actuador va ligado a ese ejecutivo.

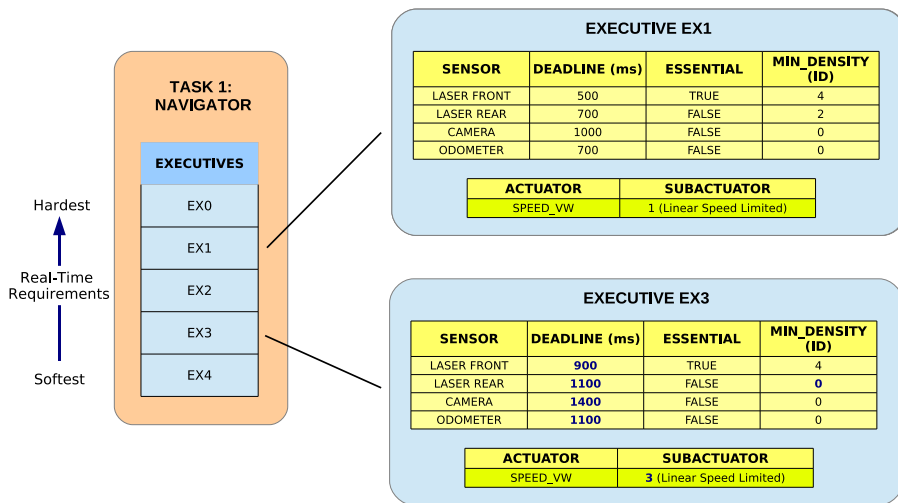


Figura 24. Estructura más detallada de la configuración de *ejecutivos*.

4.2 Experimentos

Como ya se ha dicho, la estructura de la aplicación web para el control remoto de robots móviles es altamente re-configurable, implementa nuestros algoritmos de una manera portable (funciona en PCs de escritorio, *tablets*, teléfonos inteligentes, etc. ver figura 25-izquierda y en medio) y es no invasiva. Para estos experimentos, el regulador de nivel grueso se ha dotado de 5 ejecutivos, todos ellos con dos sensores esenciales (láser frontal y odómetro) y dos no esenciales (láser trasero y cámara frontal), pero con diferentes parámetros que diferencian al ejecutivo más restrictivo (x_1). Hemos definido $T_1 = 5s$ y $T_2 = 10s$ como los parámetros del algoritmo del regulador grueso (ver capítulo 4).



Figura 25. Izquierda y en medio) Nuestro interfaz en un experimento de navegación remota real, que se puede ejecutar en tres plataformas diferentes: un PC de escritorio, una *tablet* y un teléfono inteligente sin variar la implementación. Derecha-arriba) Camino correspondiente a uno de nuestros laboratorios, donde se han realizado los experimentos de este trabajo. Derecha-abajo) Instante del experimento donde el robot sale de los límites del camino, lo que se indica al usuario mediante un cambio de color de negro a rojo en los datos del sensor del láser.

Además de relajar experimentos reales sobre el robot Sancho, para evaluar más exhaustivamente la calidad del control conseguida con nuestra interfaz, hemos realizado experimentos consistentes en conducir, con un joystick estándar, un robot simulado a lo largo de un camino que se corresponde con uno de nuestros laboratorios (figura 25-derecha). La interfaz ha sido modificada en estos experimentos para que guarde automáticamente el tiempo total dedicado a completar la vuelta, λ , y la fracción de λ en la que el robot ha estado dentro de sus límites, $o \in [0,1]$. Después de la vuelta, calcula una medida de la calidad del control, que llamamos tiempo efectivo, λ_{eff} , sumándole a λ una penalización por el tiempo transcurrido fuera de los límites de la pista: $\lambda_{\text{eff}} = \lambda + 10(1 - o)\lambda$. Cuanto mayor sea λ_{eff} , peor será la calidad del control.

Hemos llevado a cabo 6 tipos de experimentos, todos con el mismo usuario, condiciones externas y despliegue de los componentes del sistema, con el fin de aislar lo máximo posible los efectos de nuestros algoritmos de otros factores. Estos seis tipos son: A) utilizando el sistema de controlador jerárquico completo, B) lo mismo que A pero con un regulador de densidad peor: un algoritmo proporcional clásico¹, C) utilizando sólo el algoritmo de regulación fina, D) como C pero con el regulador proporcional clásico, E) sin ninguna regulación, fijando la densidad a la más pequeña, F) como E pero utilizando la mayor densidad. Hemos repetido 50 veces cada uno de estos experimentos y recogido el promedio de λ_{eff} en la tabla 5. Es evidente la ventaja, de media, de usar el sistema controlador completo (A), es

¹ La densidad siguiente se calcula como una constante (1.0 en nuestro caso) veces el error actual en probabilidad, $P_i - p(d_0)$, redondeando al entero más cercano.

decir, grueso y fino, y también el mejor comportamiento de nuestro nivel de regulación fina con respecto a otros controladores (A vs. B, C vs. D), incluso cuando no se utiliza una regulación gruesa (C vs. D). Además, obtuvimos mejores resultados con el sistema completo (A) que fijando la densidad a la más baja (E, que produce los mejores tiempos de transmisión pero minimizar la cantidad de información para al usuario). Todo esto ha sido publicado en [67].

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
55	66.9	58.4	69.1	57.5	114.7

Tabla 5. Calidad de control (media de λ_{eff} ms)



UNIVERSIDAD
DE MÁLAGA

5 RESUMEN DE PUBLICACIONES COMPENDIADAS

En este capítulo se resumen brevemente el contenido de los trabajos incluidos para el compendio de esta tesis, disponibles en la segunda parte de la tesis *Parte II: Compendio de publicaciones*.

5.1 Publicación A. "Marginal Probabilistic Modeling of the Delays in the Sensory Data Transmission of Networked Telerobots"

En esta publicación estudiamos las distribuciones de probabilidad marginales que pueden ser una buena aproximación de la distribución real de los retardos temporales generados en la transmisión de flujo sensorial para un robot teleoperado por Internet (ver sección 2.2). No conocemos la dinámica de esta secuencia de retardos, pero son más simples de calcular que otras herramientas y necesitan modificaciones menores en el robot y en el teleoperador. Dado que las secuencias de retardos muestran fuertes no linealidades en estas aplicaciones en red, para satisfacer la hipótesis *iid* requerida por el enfoque marginal se aplica un método de detección de cambios. Los resultados aquí presentados indican que algunos modelos paramétricos explican muchos escenarios reales cuando se utiliza este método de detección de cambios, mientras que algunas distribuciones no paramétricas tienen una tasa muy buena de modelado en el caso de que la detección de no linealidad no sea posible y que dividir el retardo total en sus tres términos básicos: servidor, red y tiempos del cliente.

5.2 Publicación B. "A computationally efficient algorithm for modeling multi-regime delays in the sensory flow of networked telerobots"

Teniendo en cuenta la publicación A, en este trabajo se realizó un estudio más exhaustivo de la hipótesis Log-normal triparamétrica, que es compatible con los comportamientos de cola larga observados en las secuencias reales de retardos

temporales (ver sección 2.3). Se propuso un algoritmo intuitivo basado en esa hipótesis para modelar escenarios completos. Nuestros resultados indicaron que la mayoría de las situaciones tienen comportamientos compatibles con nuestras suposiciones, lo que hace que nuestro enfoque sea adecuado para la estimación *on-line* de bajo coste.

5.3 Publicación C. "Log-logistic modelling of sensory flow delays in networked telerobots"

Anteriormente, se propusieron métodos estadísticos más sencillos que no requieren ningún conocimiento de los componentes y son adecuados para implementaciones que requieren menor coste computacional (por ejemplo, en interfaces de teléfonos móviles). En este sentido, en la publicación anterior propusimos un modelo Log-normal de tres parámetros que se ajusta razonablemente bien a estos retardos, siempre y cuando la detección de no linealidades se resuelva completamente. En la presente publicación se ha propuesto una solución más flexible: la distribución Log-logística, que se ha descubierto que se ajusta mejor a los retardos que la Log-normal (ver sección 2.3). Además, se han presentado dos algoritmos para modelar una señal de retardo completa, incluyendo las no linealidades abruptas, basados en la suposición Log-logística. Los resultados obtenidos mostraron un ajuste muy bueno en datos reales generados en una variedad de escenarios usando una serie de sensores, redes y software de aplicación.

5.4 Publicación D. "Hierarchical Regulation of Sensor Data Transmission for Networked Telerobots"

Esta publicación propone regular la información sensorial que se está transmitiendo en el sistema de teleoperación para garantizar requisitos de tiempo real suaves y también optimizar la calidad de control (ver capítulos 3 y 4). Se presenta un nuevo controlador jerárquico de dos niveles que varía la cantidad de datos de los sensores y dinámicamente reconfigura sensores activos. Este controlador se ha implementado en una interfaz de teleoperador basada en web que es altamente portable (funciona en PCs de escritorio, tabletas, teléfonos inteligentes, etc.) y no invasiva, es decir, requiere mínimas modificaciones en los componentes del sistema. Los resultados de nuestros experimentos demuestran la maximización de los datos transmitidos y garantizan los requisitos en tiempo real.

6 CONCLUSIONES Y FUTUROS TRABAJOS

En esta tesis se han realizado diversas contribuciones a la teleoperación de robots controlados por Internet, pero las herramientas diseñadas no sólo se podrían aplicar a este campo concreto de la Robótica Móvil. Al centrarnos en el estudio de optimizar al máximo, en tiempo y cantidad, los datos transmitidos por Internet sin necesidad de conocer la dinámica de estos retardos, este sistema puede ser fácilmente adaptado a otras disciplinas, como entornos multimedia, y, en general, a aplicaciones donde el coste computacional esté limitado.

En esta tesis hemos estudiado los retardos presentes en un sistema de telecontrol robótico por Internet que transmite datos remotos y cuyos tiempos empleados en tales transmisiones no son deterministas. Se trata de sistemas donde ni el hardware ni el software proporcionan garantías estrictas de cumplir tiempos en las transmisiones. Nuestro enfoque, además, permite implementaciones eficientes en máquinas que tienen limitaciones computacionales, como teléfonos móviles, *tablets*, etc, cada vez más utilizados como interfaces remotas o para mostrar datos sensoriales. Nuestras contribuciones son, por tanto, innovadoras ya que afrontan el problema simultáneamente desde un punto de vista eficiente, como un entorno heterogéneo (es decir, que involucra componentes distintos a la red, como sistemas operativos, interfaces de teleoperación y software de aplicación) y sin conocimiento previo de la dinámica de los retardos.

Para garantizar los requisitos de tiempo del flujo de datos se han estudiado una serie de distribuciones de probabilidad que son estimadas *on-line*, dotando así de modelos probabilísticos adecuados a los retardos en la transmisión sensorial. Viendo los resultados obtenidos, se puede concluir que los algoritmos de detección *on-line* de no linealidades en estos retardos (cambios de regímenes, ráfagas y valores atípicos) *stateless* y el *state-based*, más el modelado con una distribución Log-logística, constituyen una solución viable con respecto a un coste computacional limitado para un control de retardos temporales en telerobots controlados por Internet. Finalmente, se ha propuesto un regulador jerárquico a dos niveles que utiliza estos modelos y unos algoritmos específicos para regular la cantidad de información sensorial que se solicita del robot teleoperado, cumpliendo

así requisitos de tiempo real suave en los retardos de transmisión. En el primer nivel de nuestro regulador se propone el método de regulación fina, que junto con el predictor Log-logístico consigue que el sistema no viole el tiempo máximo permitido en las transmisiones, algo que si ocurre con predictores más sencillos (Bernoulli). El segundo nivel es el regulador grueso, que funciona con varios sensores a la vez, pudiendo deshabilitarlos —sin emitir más peticiones temporalmente— o no. Viendo los resultados obtenidos en los experimentos, se puede concluir que, usando el sistema controlador completo (regulación fina con el predictor Log-logístico, más el regulador grueso), existe una ventaja evidente con respecto a otros controladores y opciones probadas en las mismas pruebas.

El trabajo presentado en esta tesis deja algunas líneas de investigación abiertas. Algunas de las más interesantes se describen a continuación.

Modelado de retardos con la distribución de probabilidad *shifted Log-logistic*. Viendo la mejora que supuso pasar de una hipótesis Log-normal a una Log-logística, la idea es seguir investigando por el camino de esta última. Durante su estudio y estimación, vimos que uno de los parámetros más complicados de estimar era la localización o desplazamiento. Con la nueva propuesta de la *shifted Log-logística*, este parámetro coincide directamente con el primer momento de esta distribución

($E(X) = \mu$), lo que supondría una mejora considerable en la precisión de la estimación de los retardos temporales de nuestro sistema.

Comparativa de la bondad de predicción. Teniendo una serie de distribuciones paramétricas ya implementadas y adaptadas en nuestro sistema, la idea sería comparar todas ellas mediante una nueva herramienta que mida los errores estadísticos cometidos con cada una, y contrastar cual de ellas sería definitivamente la mejor opción para una serie de escenarios que engloben todas las posibilidades de configuración de nuestro sistema robótico telecontrolado por Internet.

Seguir desarrollando los algoritmos de detección de no linealidades. Los algoritmos propuestos en esta tesis se han desarrollado de forma heurística. Aunque hay que mencionar su gran potencial a la hora de realizar su función, se podría estudiar más a fondo y mejorar algunos aspectos para que esta detección de regímenes, ráfagas y valores atípicos sea aún más preciso.

Aplicación de nuevos métodos de regulación adaptativa. Para realizar una regulación óptima y adaptativa de nuestro sistema de control, deberíamos seguir trabajando en esta línea de investigación y así refinar aún más los resultados que ya tenemos. En este sentido, en el grupo ya se ha realizado trabajos previos en cuanto a algoritmos de regulación basados en aprendizaje por refuerzo [90].

7 BIBLIOGRAFÍA

- [1] S. Jia, Y. Hada, Y. Gang and K. Takase, "Distributed Telecare Robotic Systems Using CORBA as a Communication Architecture," in *In Proceedings of the IEEE International Conference on Robotics & Automation*, Washington, DC, USA, 2002.
- [2] Y. Tipsuwan and M. Chow, "Control methodologies in networked control systems," *J. Control Pract.*, vol. 11, pp. 1099-1111, 2003.
- [3] X. Wang y H. Schulzrinne, «Comparison of Adaptive Internet Multimedia Applications,» *IEICE Transactions on Communications*, Vol. 2E82-B, nº 6, 1999.
- [4] N. A. Gershenfeld y A. S. Weigend, The future of Time Series. Time Series Prediction: Forecasting the Future and Understanding the Past, Addison-Wesley: A. S. Weigend and N. A. Gershenfeld, eds., 1993.
- [5] M. Cavusoglu, W. Williams, F. Tendick y S. Sastry, «Robotics for telesurgery: Second generation berkeley/UCSF laparoscopic telesurgical workstation and looking towards the future applications,» *Ind. Robot*, vol. 30, pp. 22-29, 2003.
- [6] L. Whitcomb, «Underwater Robotics: Out of the Research Laboratory and into the Field,» de *In Proceedings of the IEEE International Conference on Robotics & Automation*, Washington, DC, USA, 2000.
- [7] Y. Tsumaki, T. Goshozono, K. Abe, M. Uchiyama, R. Koeppe y G. Hirzinger, «Verification of an Advanced Space Teleoperation System Using Internet,» de *In Proceedings of the IEEE International Conference on Intelligent Robot Systems*, Takamatsu, Japan, 2000.
- [8] Irobot, «Roomba robot's manufacturers» Disponible: <http://www.irobot.com/sp.cfm?pageid=122>.
- [9] H. Zhang, J. Zhang, G. Zong, W. Wang y R. Liu, «Sky Cleaner 3. A Real Pneumatic Climbing Robot for Glass-Wall Cleaning,» *Robotic & Automation Magazine IEEE*, 01 03 2006.

- [10] M. Saptharishi, C. S. Oliver, C. P. Diehl, K. S. Bhat, J. M. Dolan, A. Trebi-Ollennu y P. Khosla, «Distributed Surveillance and Reconnaissance Using Multiple Autonomous ATVs: CyberScout,» *IEEE Transactions on Robotics and Automation*, vol. 18, nº 5, 2002.
- [11] P. Y. Oh, M. Joyce y J. Gallagher, «Designing an Aerial Robot for Hover-and-Stare Surveillance,» de *12th International Conference on Advanced Robotics, ICAR'05*, Seattle, WA, USA, 2005.
- [12] M. Bennewitz, F. Faber, D. Joho, M. Schreiber y S. Behnke, «Towards a Humanoid Museum Guide Robot that Interacts with Multiple Persons,» de *Proceedings of 2005 5th IEEE-RAS International Conference on Humanoid Robots*, 2005.
- [13] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. .. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, R. Roy, J. Schulte y D. Schulz, «Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva,» *The International Journal of Robotics Research*, vol. 19, nº 11, pp. 972-999, 2000.
- [14] G. Gilbreath, D. A. Ciccimaro y H. R. Everett, «An Advanced Telereflexive Tactical Response Robot,» de *IEEE International Conference on Robotics and Automation ICRA2000*, San Francisco, CA, USA, 2000.
- [15] Y. Shimosasa, J. Kanemoto, K. Hakamada, H. Horii, T. Arika, Y. Sugawara, F. Kojio, A. Kirnura y S. Yuta, «Some results of the test operation of a security service system with autonomous guard robot,» de *26th Annual Conference of the IEEE on Industrial Electronics Society*, Nagoya, Japan, 2000.
- [16] A. Tapus, M. J. Mataric y B. Scasselati, «Socially assistive robotics,» *Robotics & Automation Magazine IEEE*, 01 03 2007.
- [17] K. Park, H. Lee, Y. Kim y Z. Bien, «A steward Robot for Human-Friendly Human-Machine Interaction in a Smart House Environment,» *IEEE Transaction on Automation Science and Engineering*, vol. 5, nº 1, 2008.
- [18] C. Galindo, J. González y J.A. Fernández-Madrigal, «Control Architecture for Human-Robot Integration. Application to a Robotic Wheelchair,» *IEEE Transactions on Systems, Man, and Cybernetics part B*, vol. 36, nº 5, pp. 1053-1067, 2006.
- [19] H. Durrant-Whyte, «Localization, Mapping and the Simultaneous Localization and Mapping Problem,» de *Australian Center for Field Robotics*, Sydney, 2002.
- [20] S. Thrun, W. Burgard y D. Fox, *Probabilistic Robotics*, The MIT Press, ISBN 978-0-262-20162-9, 2005.
- [21] J.A. Fernández-Madrigal, «Probabilistic Robotics: A Tutorial,» Dpto de Ingeniería de Sistemas y Automática, Universidad de Málaga, Málaga, 2004.

- [22] J.-L. Blanco, J.A. Fernández-Madrigal y J. González, «Toward a Unified Bayesian Approach to Hybrid Metric-Topological SLAM,» *IEEE Transactions on Robotics*, vol. 24, nº 2, 2008.
- [23] K. J. Astrom y B. Wittenmark, *Adaptive Control*, Prentice-Hall, ISBN 978-0201558661, 1994.
- [24] D. Bertsekas y S. E. Shreve, *Stochastic Optimal Control: the Discrete Case*, Athena Scientific, ISBN 1-886529-03-5, 1996.
- [25] R. S. Sutton y A. G. Barto, *Reinforcement Learning: An Introduction*, Cambridge, MA: the MIT Press, 1998.
- [26] B. Siciliano y O. Khatib, *Springer Handbook of Robotics*, Berlin, Germany: Springer, 2008.
- [27] A. Burns y A. Wellings, *Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX*, Wokingham, UK,: Addison-Wesley, 2009.
- [28] Y. Wang, M. Vuran y S. Goddard, «Cross-layer analysis of the End-to-End delay distribution in wireless sensor networks,» *IEEE/ACM Trans. Netw.*, vol. 20, pp. 305-318, 2012.
- [29] P. Liu, M. Meng, J. Gu, S. Yang y C. Hu, «Control and Data Transmission for Internet Robotics.,» de *In Proceedings of the IEEE International Conference on Robotics & Automation*, Taipei, Taiwan, 2003.
- [30] G. Alt y W. Lages, «Networked Robot Control with Delay Compensation,» de *In Proceedings of the Fifth Real-Time Linux Workshop*, Valencia, Spain, 2003.
- [31] O. Imer, S. Yksel y T. Basar, «Optimal control of LTI systems over unreliable communication links,» *Automatica*, vol. 42, pp. 1429-1439, 2006.
- [32] W. Kim, K. Ji y A. Ambike, «Networked real-time control strategy dealing with stochastic time delays and packet losses,» *J. Dyn. Syst. Meas. Control*, vol. 128, pp. 681-685, 2006.
- [33] Y. Tipsuwan y M. Chow, «Control methodologies in networked control systems,» *J. Control Pract.*, vol. 11, pp. 1099-1111, 2003.
- [34] J. Gibbon y T. Little, «The use of network delay estimation for multimedia data retrieval,» *IEEE J. Sel. Areas Commun*, vol. 14, pp. 1376-1387, 1996.
- [35] YouTube, «How YouTube Works—Computerphile.» Disponible: <https://youtu.be/OqQk7kLuaK4>.
- [36] T. Yensen, J. Lariviere, I. Lambadaris y R. Goubran, «HMM delay prediction technique for VoIP,» *IEEE Transa. Multimed*, vol. 5, pp. 444-457, 2003.
- [37] R. Lozoya Gámez, P. Martí, M. Velasco y J. Fuertes, «Wireless Network Delay Estimation for Time-Sensitive Applications,» Disponible: http://esaii.upc.edu/people/pmarti/nde_06.pdf (accessed on 25 October 2015), Research Report ESAII-RR-06-12, 2006.

- [38] K. Deng, A. Moore y M. Nechyba, «Learning to Recognize Time Series: Combining ARMA Models with Memory-Based Learning.» de *In Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, California, USA, 1997.
- [39] H. Feng y Y. Shu, «Study on Network Traffic Prediction Techniques.» de *In Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing*, Wuham, China, 2005.
- [40] A. Li, S. He y Z. Qin, «Real-Time Segmenting Time Series Data.» Lecture Notes in Computer Science. In *Web Technologies and Applications*. APWeb , Berlin Heidelberg, Germany, 2003.
- [41] V. De Buen Remiro, «Modelación Masiva X-ARIMA» Disponible: <https://www.tol-project.org/svn/tolp/OfficialTolArchiveNetwork/ArimaTools/doc/Modelacion%20X-ARIMA%20masiva/Modelacion%20X-ARIMA%20masiva.pdf>.
- [42] A. Gago-Benítez y J.A. Fernández-Madrigal, «Log-Normal Modeling of Non-Linear Sequences of Delays with Classical Change Detection Methods.» de *In Proceedings of the International Workshop on Applied Probability (IWAP'2012)*, Jerusalem, Israel, 2012.
- [43] G. Box y G. Jenkins, *Time Series Analysis: Forecasting and Control*, San Francisco, CA, USA: Holden-Day, 1976.
- [44] A. Gago-Benítez, J.A. Fernández-Madrigal y A. Cruz-Martín, «Marginal Probabilistic Modeling of the Delays in the Sensory Data Transmission of Networked Telerobots.» *Sensors*, vol. 14, n° 2, pp. 2305-2349, 2014.
- [45] J. S. Oakland, *Statistical Process Control*, Fifth Edition.
- [46] J. González-Jiménez, C. Galindo y J. Ruiz-Sarmiento, «Technical Improvements of the Giraffe Telepresence Robot based on User's Evaluation.» de *In Proceedings of the 21st IEEE International Symposium on Robot and Human Interactive Communication*, Paris, France, 2012.
- [47] «Submarine Cable Map,» 2013. Disponible: <http://submarine-cable-map-2013.telegeography.com>.
- [48] J. González, C. Galindo, J. Blanco, J.A. Fernández-Madrigal, V. Arévalo y F. Moreno, «SANCHO, a Fair Host Robot. A Description.» de *In Proceedings of the IEEE International Conference on Mechatronics*, Málaga, Spain, 2009.
- [49] Surveyor. Disponible: <http://www.surveyor.com>.
- [50] D. Mills, «Internet Time Synchronization: The Network Time Protocol.» de *IEEE Transaction on Communications*, 1991.
- [51] D. House, «Spline Curves» Disponible: <http://people.cs.clemson.edu/~dhouse/courses/405/notes/splines.pdf>.
- [52] L. Wasserman, *All of Statistics: A Concise Course in Statistical Inference*, New York, NY, USA: Springer., 2003.

- [53] J. Cao, W. Cleveland, D. Lin y D. Sun, Internet Traffic Tends Toward Poisson and Independent as the Load Increases. In *NonLinear Estimation and Classification*, New York, NY, USA: Springer, 2002.
- [54] I. Antoniou, V. Ivanov y P. Zrelov, «On the log-normal distribution of network traffic,» *J. Phys. D: Nonlinear Phenom*, vol. 167, pp. 72-85, 2002.
- [55] A. Gago-Benítez, J. Fernández-Madrigal y A. Cruz-Martín, «Log-logistic modeling of sensory flow delays in networked telerobots,» de *IEEE Sensors*, Taipei, Taiwan, 2012.
- [56] R. Ng y J. Han, «Efficient and Effective Clustering Methods for Spatial Data Mining,» de *Proceedings of the Very Large Data Bases Conference*, Zurich, 1994.
- [57] E. Acuna y C. Rodriguez, «Meta Analysis Study of Outlier Detection Methods in Classification,» Technical paper, Department of Mathematics, University of Puerto Rico, Disponible: <http://academic.uprm.edu/eacuna/paperout.pdf>.
- [58] C. Lu, D. Chen y Y. Kou, «Algorithms for Spatial Outlier Detection,» de *IEEE International Conference on Data-Mining (ICDM'03)*, Melbourne, FL, USA, 2003.
- [59] E. Knorr y R. Nr, «Algorithms for Mining Distance-Based Outliers in Large Datasets,» de *24th International Conference Very Large Data Base (VLDB)*, New York, NY, USA, 1998.
- [60] S. Bay y M. Schwabacher, «Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule,» de *9th ACM-SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, 2003.
- [61] A. Gago-Benítez, J.A. Fernández-Madrigal y A. Cruz-Martín, «A computationally efficient algorithm for modeling multi-regime delays in the sensory flow of networked telerobots,» de *ICCRC*, Cape Town, South Africa, 2012.
- [62] A. C. Cohen y B. J. Whitten, «Estimation in the Three-Parameter Lognormal Distribution,» *Journal of the American Statistical Association*, vol. 75, n° 370, p. 399, 1980.
- [63] R. D'Agostino y M. Stephens, *Goodness-of-fit Techniques*, New York, NY, USA: Marcel Dekker, Inc, 1986.
- [64] A. Gago-Benítez, J.A. Fernández-Madrigal y A. Cruz-Martín, «Log-logistic modelling of sensory flow delays in networked telerobots,» *IEEE Sensors*, vol. 13, n° 8, 2013.
- [65] M. I. Ahmad, C. D. Sinclair y A. Werritty, «Log-logistic flood frequency analysis,» *Journal of Hydrology*, vol. 98, pp. 205-224, 1988.
- [66] M. A. Stephens, «EDF Test of fit for the logistic distribution,» Department of Statistics, Stanford, California, 1979.

- [67] A. Martínez-Tenor, A. Gago-Benitez, J.A. Fernández-Madriral, A. Cruz-Martin, A. Asenjo-Plaza y A. Gonzalez-Navarro, «Hierarchical Regulation of Sensor Data Transmission for Networked Telerobots,» de *IEEE Sensors*, Valencia (Spain), 2014.
- [68] J.A. Fernández-Madriral, A. Gago-Benitez, A. Cruz-Martin, A. Martínez-Tenor, A. Asenjo-Plaza y A. Gonzalez-Navarro, «Métodos implementados en computador y sistemas informáticos de regulación automática de la cantidad de datos transmitidos entre dispositivos conectados a una red». España Patente 2 565 878, 9 9 2016.
- [69] A. Martínez-Tenor, J.A. Fernández-Madriral, A. Cruz-Martin, A. Gago-Benitez, A. Asenjo-Plaza y A. Gonzalez-Navarro, «Métodos implementados en computador y sistemas informáticos de activación y desactivación automáticas de transmisiones concurrentes de datos entre dispositivos conectados a una red». España Patente 2 550 728, 19 5 2016.
- [70] L. Dameshek, A. Martlage y C. Knapheide, «User Interface Image Element Display and Adaptation System». Patente US2007245306, 18 10 2007.
- [71] M. Chang, M. Eyer, A. Krishnan, A. Proehl, D. Yang, F. Zustak, P. Shintani, N. Colsey, B. Candelore y D. Golden, «Simplified user interface by adaptation based on usage history». Patente US2003090515, 15 05 2003.
- [72] C. Dinges y M. Schlereth, «Adaptation of a human-machine interface as a function of a psychological profile and a current state of being of a user». Patente US2003174122, 18 09 2003.
- [73] J. Otto, A. Luchene, M. Mueller y R. Mueller, «Graphical user interface adaptation system for a point of sale device». Patente US2008306886, 11 12 2008.
- [74] A. Rauh, L. Erbe, T. Gieselmann, F. Mueller, S. Vidhyaan, N. Paul, S. Haffner, V. Tomar y H. Schmidt, «Dynamic User Interface Content Adaptation And Aggregation». Patente US2012166976, 28 06 2012.
- [75] H. Young, M. Bohan, S. Herbst, C. O'sullivan, C. Phipps y E. Vargas, «Method And Apparatus For User Interface Adaptation». Patente WO2007118019, 18 10 2007.
- [76] S. Ronkainen y J. Saarinen, «Method Device And System For User Interface Adaptation». Patente WO2012123621, 20 09 2012.
- [77] P. Panay, S. Drasnin, M. McCormack, J. Inman, S. Seixeiro, A. Reed, R. Groene y J. Johnson, «Adaptation Of The User Interface To Mimic Physical Characteristics Of A Peripheral». Patente WO2013184393, 12 12 2013.
- [78] B. Travis y C. Jaeho, «Interface Adaptation System». Patente CN102246116, 16 11 2011.
- [79] R. Oboe, «Web-Interfaced, Force-Reflecting Teleoperation System,» de *IEEE Transactions on Industrial Electronics*, 2001.

- [80] J. Wu, Q.-S. Jia, K. H. Johansson y L. Shi, «Event-Based Sensor Data Scheduling: Trade-Off Between Communication Rate and Estimation Quality,» de *IEEE Transaction on Automatic Control*, 2013.
- [81] N. D. Wells, «Buffer Resynchronisation in a variable transmission rate coder». Patente GB2274041, 06 07 1994.
- [82] L. Cheng, J. Zhang y J. Zhao, «Method for implementing automatic grading bandwidth regulation on multi-service transmission platform». Patente CN101184041, 21 05 2008.
- [83] F. Rong y F. Qi, «Method for realizing dynamic jitter buffering regulation in speed sound transmission course». Patente CN100417129, 05 10 2005.
- [84] C. J. Sparrell, A. D. Vasilevsky, C. Ruzszyk y J. yhang, «Local area-networked system having intelligen traffic control and efficient bandwidth management». 05 05 2009.
- [85] S. Huang, «Data transmission rate regulation method, device and wireless access point». 22 01 2014.
- [86] J. Ge, «Regulation and control method for video transmission on narrow-band network». 28 08 2013.
- [87] W. Falk y D. W. Thomas, «Managing service leve agreements using statistical process control in a networked computing environment». Patente US2012131172, 14 05 2012.
- [88] J.A. Fernández-Madrigal, «The BABEL Development System for Integrating Heterogeneous Robotic Software,» 2003.
- [89] J. Blanco Claraco, «Mobile Robot Programming Toolkit (MRPT),» <http://www.mrpt.org>, 2014.
- [90] J.A. Fernandez-Madrigal, C. Galindo, E. Cruz-Martin, A. Cruz-Martin y J. Gonzalez-Jimenez, «Automatic Regulation of the Information Flow in the Control Loops of a Web Teleoperated Robot,» de *IEEE International Conference on Robotics and Automation (ICRA)*, Rome (Italy), 2007.



UNIVERSIDAD
DE MÁLAGA

Parte II. Compendio de publicaciones



UNIVERSIDAD
DE MÁLAGA

A. MARGINAL PROBABILISTIC MODELING OF THE DELAYS IN THE SENSORY DATA TRANSMISSION OF NETWORKED TELEROBOTS

Authors: *Ana Gago-Benítez, Juan-Antonio Fernández-Madrigal and Ana Cruz-Martín*

Published in: *Sensors*

Volume: *14*

Pages: *2305-2349*

Year: *2014*

DOI: *10.3390/s140202305*

Abstract: Networked telerobots are remotely controlled through general purpose networks and components, which are highly heterogeneous and exhibit stochastic response times; however their correct teleoperation requires a timely flow of information from sensors to remote stations. In order to guarantee these time requirements, a good on-line probabilistic estimation of the sensory transmission delays is needed. In many modern applications this estimation must be computationally highly efficient, e.g., when the system includes a web-based client interface. This paper studies marginal probability distributions that, under mild assumptions, can be a good approximation of the real distribution of the delays without using knowledge of their dynamics, are efficient to compute, and need minor modifications on the networked robot. Since sequences of delays exhibit strong non-linearities in these networked applications, to satisfy the iid hypothesis required by the marginal approach we apply a change detection method. The results reported here indicate that some parametrical models explain well many more real scenarios when using this change detection method, while some non-parametrical distributions have a very good rate of successful modeling in the case that non-linearity detection is not possible and that we split the total delay into its three basic terms: server, network and client times.

B. A COMPUTATIONALLY EFFICIENT ALGORITHM FOR MODELING MULTI-REGIME DELAYS IN THE SENSORY FLOW OF NETWORKED TELEROBOTS

Authors: *Ana Gago-Benítez, Juan-Antonio Fernández-Madrigal and Ana Cruz-Martin*

Published in: *2012 International Conference on Control, Robotics and Cybernetics (ICRC 2012)*

Pages: *853-859*

Year: *2012*

DOI: *10.1115/1.860045_ch109*

Abstract: Networked telerobots are a class of robots remotely controlled over general-purpose networks, like the Internet. They involve heterogeneous components: network, operating systems, application software, interfaces, etc., most of them with a stochastic response-time behaviour. In spite of that, it is often necessary to model and predict these delays, which exhibit multiple regimes, bursts and outliers.

This modeling may be done with diverse approaches: state-space methods, hidden markov models, neural networks, etc. However, they have high computational demands, not generally available in networked telerobots, where the client interface is implemented in a web-browser. In a previous work [7] we studied several simple statistical methods based on parametrical distributions, suitable for efficient on-line modeling. In this paper we carry out a more in-depth study under the three-parametrical lognormal hypothesis, which is compatible with long-tailed behaviours observed in the delays. We propose an intuitive algorithm based on that hypothesis for modeling complete scenarios. Our results indicate that most situations have behaviours compatible with our assumptions, which makes our approach suitable for low-cost on-line estimation.

C. LOG-LOGISTIC MODELLING OF SENSORY FLOW DELAYS IN NETWORKED TELEROBOTS

Authors: *Ana Gago-Benítez, Juan-Antonio Fernández-Madrigal and Ana Cruz-Martín*

Published in: *IEEE Sensors Journal*

Volume: *13*

Pages: *2944-2953*

Year: *2013*

DOI: *10.1109/JSEN.2013.2263381*

Abstract: This paper deals with the modelling of the delays in the transmission of sensory data coming from a networked telerobot, which would allow us to predict future times of arrival and provide guarantees on the time requirements of these systems. Considering these delay sequences as an uni-dimensional temporal signal, they easily exhibit rich stochastic behaviour —abrupt changes of regime and bursts—, due to the heterogeneity of the hardware and software components in the data path. There exist approaches for modelling this kind of signals without explicit knowledge of the system components: state-space reconstruction, hidden Markov models, neural networks, etc., but they are mostly focused on the stochasticity of the network only, without taking into account other elements in the sensory flow that also have an important influence in the delays. Previously, we have proposed simpler statistical methods that do not require any component knowledge either and are suitable for more lightweight implementations (e.g., in mobile phone interfaces). In this sense we reported elsewhere a log-normal three-parametrical model that fits reasonably well these delays as long as change detection is completely solved. Now we propose a more flexible solution: the log-logistic distribution, which has been found to fit delays better than the log-normal. In addition we present two algorithms to model an entire delay signal including abrupt non-linearities, based on the log-logistic assumption. Our results show quite good fittings of real datasets gathered from a number of combinations of sensors, networks and application software, provided that some mild assumptions hold.

D. HIERARCHICAL REGULATION OF SENSOR DATA TRANSMISSION FOR NETWORKED TELEROBOTS

Authors: *Ángel Martínez-Tenor, Ana Gago-Benítez, Juan-Antonio Fernández-Madrigal, Ana Cruz-Martín, Rafael Asenjo and Ángeles Navarro.*

Published in: *IEEE Sensors, Valencia (Spain)*

Pages: 466-469

Year: 2014

DOI: *10.1109/ICSENS.2014.6985036*

Abstract: Networked telerobots carry sensors that send data, with stochastic transmission times, to a remote human operator, who must execute some real-time control task (e.g., navigation). In this paper we propose to regulate the sensory information being transmitted in order to guarantee soft real-time requirements and also optimize the quality of control, through a novel two-level hierarchical controller that both varies the amount of transmitted sensor data and dynamically reconfigures active sensors. Our controller has been implemented in a web-based teleoperator interface that is highly portable (it runs on desktop PCs, tablets, smartphones, etc.) and non-invasive, i.e., requires minimal modifications in the existing components of the system, thus being suitable for many applications. Here we present our regulation methods and the results of some experiments. They demonstrate the maximization of the transmitted data while guaranteeing the real-time requirements.