IDENTIFICACION DE SISTEMAS LINEALES SOBRE REDES DE COMUNICACIONE

ARANCIBIA CARRASCO, FELIPE

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE ELECTRÓNICA

Identificación de Sistemas lineales sobre redes de comunicaciones

Tesis de Grado presentada por

Felipe Arancibia Carrasco

como requisito parcial para optar al título de

Ingeniero Civil Electrónico

y al grado de

Magíster en Ciencias de la Ingeniería Electrónica

Profesor Guía
Dr. Rodrigo Carvajal

Profesor Co-guía
Dr. Juan C. Agüero

Valparaíso, 2021.

TÍTULO DE LA TESIS:

Identificación de Sistemas lineales sobre redes de comunicaciones

AUTOR:

Felipe Esteban Arancibia Carrasco

TRABAJO DE TESIS, presentado en cumplimiento parcial de los requisitos para el título de Ingeniero Civil Electrónico y el grado de Magíster en Ciencias de la Ingeniería Electrónica de la Universidad Técnica Federico Santa María.

Dr. Rodrigo Carvajal _____

Dr. Juan C. Agüero _____

Dr. Gonzalo Carvajal _____

Dr. Ramón Delgado _____

Valparaíso, Enero de 2021.

dedicado a mi familia

y amigos

AGRADECIMIENTOS

En primer lugar, quisiera agradecer el apoyo de mi familia en estos años de Universidad, especialmente a mi madre, ya que sin su entrega culminar esta etapa hubiese sido más complejo.

Me gustaría agradecer a Nany, mi compañera los últimos tres años, quien me acompañó en la parte más difícil y estresante de la carrera. Agradezco su apoyo constante, ayudándome a ser mejor tanto en lo académico como en lo personal, motivándome en todo momento. Gracias por toda la ayuda entregada los últimos meses y por todas salidas para distraernos de la Universidad.

Por otro lado, es importante mencionar las amistades y personas que conocí en este proceso universitario, cuyos lazos hicieron más agradable los años, dejando historias que recordaré por el resto de mi vida. Destaco la amistad de Papito, Fredes y Facundo, las cuales espero que perduren por mucho tiempo.

Finalmente, agradecer a los profesores que me guiaron durante mi carrera, dejando una cantidad inmensa de aprendizaje. Especialmente mencionar a los profesores Rodrigo Carvajal y Juan Carlos Agüero, quienes estuvieron siempre disponibles para ayudarme en el proceso de la tesis. Además de agradecer el apoyo del Centro Basal "Advanced Center for Electrical and Electronic Engineering" (AC3E) - FB0008 y al proyecto ANID-FONDECYT 1181158.

CONTENIDO

RESUMEN

El objetivo principal de la identificación de sistemas es modelar matemáticamente un sistema a partir de la utilización de datos. Esta área de la ingeniería de control ha generado un gran interés en variados campos, destacando la integración de redes de comunicación y computación.

En esta tesis abordamos el problema de estimación conjunta de parámetros de un sistema dinámico a través de redes de comunicación, integrando las restricciones que conllevan, considerando particularmente un escenario donde pueden haber retardos desconocidos en la señal y posible pérdida de paquetes.

Para realizar la estimación de los parámetros del sistema dinámico, se desarrolla un algoritmo en base a la estimación por Máxima Verosimilitud a través del algoritmo Esperanza-Maximización, debido a la incorporación de pérdidas de datos en el sistema. Además, para estimar el retardo de tiempo causado por la red, se consideran técnicas de selección de modelos, en particular el criterio de información de Akaike.

Finalmente, la efectividad del algoritmo propuesto se analiza mediante diversos ejemplos, simulados en el software MATLAB. Con el objetivo de observar el comportamiento de la estimación de parámetros en función de la probabilidad de perdida de datos, se realizan ejemplos números en conjunto con un análisis de sensibilidad.

ABSTRACT

The main goal of System Identification is to mathematically model a system from empirical data. This area of Control systems has attracted much interest from different scientific fields, particularly when other areas such as communication networks and computing are involved.

In this thesis we address the joint estimation the parameters of a networked system that is subject to packet loss and delays.

In order to estimate the parameters of the dynamic system, we develop a Maximum Likelihood estimator that includes lost data due to packet collision or cyberattacks. Since we deal with unavailable (hidden) data, we develop our technique via the Expectation-Maximization algorithm. In addition, the estimation of the network delay we utilize Akaike's information criterion.

The benefits of our proposal are shown via numerical simulations in MATLAB. In order to assess in detail the performance of our technique, we also carry out several examples within a sensitivity analysis.

Capítulo 1

INTRODUCCIÓN

1.1. Descripción del problema

La Identificación de Sistemas se encarga de modelar a través de expresiones matemáticas un sistema mediante la utilización de los datos de entrada y salida, convirtiéndose en un campo concreto y bien estudiado en su enfoque clásico [1]. En la actualidad, una de las tendencias dominantes en el desarrollo de sistemas físicos es la creciente interconexión de sistemas a través de redes de comunicaciones, que incluyen sistemas de comunicación inalámbrica o cableada.

Este crecimiento tecnológico ha generado interés en variados campos, destacando el área del transporte, modelado ambiental, redes inteligentes, entre otras [2], por lo que en el área de Identificación de Sistemas también se han integrado estos sistemas de comunicación, generando nuevos paradigmas, reconociendo nuevos problemas y explorando nuevas soluciones [3]. Esto se debe a que la información transmitida por estos canales generan nuevas incertidumbres, no incorporadas en el enfoque clásico en el área de Identificación de Sistemas. Los canales de comunicaciones tienen entre sus restricciones más importantes: i) limitaciones del ancho de banda, ii) muestreo de las señales, iii) retardo de procesamiento y transmisión en las señales debido a la red y iv) pérdida de información [4]. En teoría de control y estimación de estados ya se han incorporado la presencia de redes de comunicaciones, área llamada Sistemas de Control en red o NCS (por sus siglas en inglés, *Networked Control System*) [5], [6], [7], por lo tanto, para el modelado y comportamiento de nuestra red de comunicación podemos basarnos en los trabajos realizados en esta área.

De esta manera, es fundamental estudiar de forma conjunta la Identificación de Sistemas con los Sistemas de Comunicaciones, enfocándose en la estimación de sistemas dinámicos, integrando las restricciones que puede generar la integración de redes de comunicaciones. En la presente Tesis se considera un sistema dinámico con parámetros desconocidos el cual transmite y recibe sus señales a través de un canal de comunicación. El objetivo es realizar la estimación de parámetros del sistema considerando dos restricciones que puede generar el canal de comunicación, las cuales son retardo y pérdida de información. Debido a la pérdida de información que existe, se realiza la estimación de Máxima Verosimilitud a través del uso de algoritmo

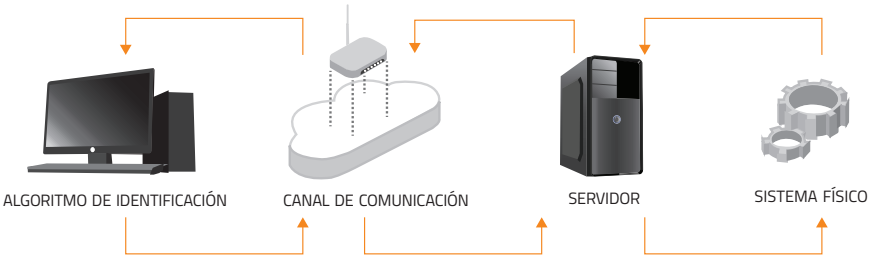

Figura 1.1. Esquema de la identificación de sistemas sobre redes de comunicación

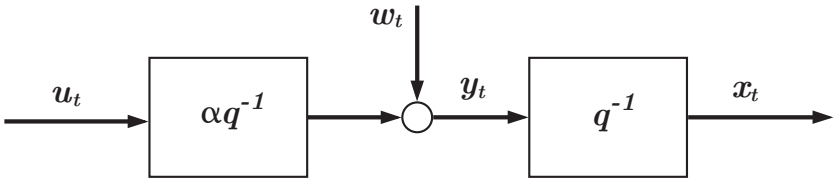

Figura 1.2. Retardo unitario generado por un canal de comunicación

Esperanza-Maximización o EM (por sus siglas en inglés, *Expectation-Maximization*). Para modelar las restricciones consideradas nos basamos en el modelado realizado en la literatura actual. El esquema general del problema de tesis puede ser observado en la Figura 1.1.

Para ilustrar de forma clara, el siguiente ejemplo muestra cómo una red de comunicación puede afectar la estimación de parámetros en la Identificación de Sistemas. Supongamos un sistema lineal en tiempo discreto modelado como un retardo unitario con una ganancia $\alpha = 2$. Además, la señal de salida del sistema y_t es transmitida a través de una red de comunicación modelada con un retardo unitario como se muestra en la Figura 1.2 (q^{-1} es el operador retardo unitario, $q^{-1}y_t = y_{t-1}$), donde además w_t es ruido gaussiano con media cero y varianza σ^2 y x_t es la señal obtenida a través del canal.

El sistema dinámico puede ser escrito como una regresión lineal dada por

$$y_t = \alpha u_{t-1} + w_t. \tag{1.1.1}$$

Para la estimación del parámetro α se propone usar un método tradicional en la Identificación de Sistemas, el cual se basa en mínimos cuadrados [8]. El método de mínimos cuadrados o LS (por sus siglas en inglés, *Least Squares*) se implementa bajo el supuesto de que no hay canal de comunicación, resultando en:

$$\hat{\alpha}_{LS} = \left(\sum_{t=1}^{N} \varphi_t \varphi_t^T \right)^{-1} \left(\sum_{t=1}^{N} \varphi_t x_t \right), \tag{1.1.2}$$

con $\varphi_t = u_{t-1}$. Entonces, la estimación es distinta dependiendo de si:

- El canal no introduce un retardo, por lo tanto, la estimación se realiza considerando $x_t = y_t$.
- El canal introduce un retardo, el cual no se sabe que existe, por lo tanto los datos disponibles de la señal de salida son $x_t = y_{t-1}$. En este caso, se ocupa el mismo algoritmo LS escrito en (1.1.2), el cual se obtiene sin saber que hay una red de comunicación.

Se realizan 1000 simulaciones de Montecarlo [9] para $t = 1 \dots N$, con $N = 1000$. Esto se muestra en la Figura 1.3

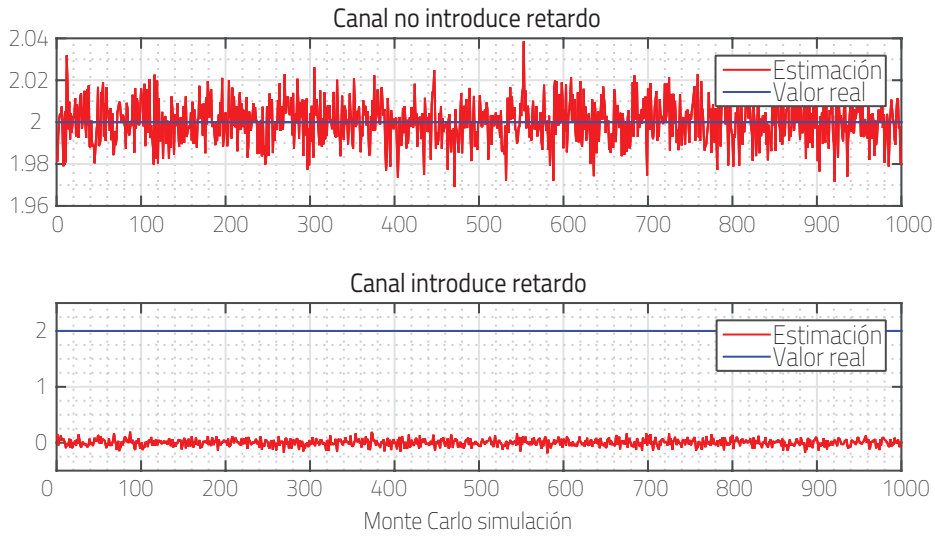

Figura 1.3. Estimación usando LS con el par $\{u_t, x_t\}$ y con el par $\{u_t, y_t\}$

Como se observa en la Figura 1.3 la estimación es errónea cuando se considera el retardo unitario, pues el algoritmo desarrollado se realiza sin considerarlo. Así, se muestra como una incertidumbre tan simple, como el retardo, afecta en la estimación, por lo que se requiere un estudio más detallado para incorporar la incertidumbre que introduce el canal.

1.2. Contribución de la Tesis

Las principales contribuciones de la tesis son las siguientes:

1. Se propone una técnica de identificación de sistemas en red que considera pérdidas de paquetes y retardos. La estimación de los parámetros de interés

se realiza por máxima verosimilitud a través del algoritmo EM y el criterio de información de Akaike.

2. Se modela la pérdida de paquetes acorde al estándar Ethernet, lo que típicamente implica la pérdida de varias mediciones consecutivas. Se asume un modelo *Output Error* para el sistema dinámico, y para la pérdida de paquetes se asume un proceso Bernoulli.

3. Se muestra a través de extensivas simulaciones que al modelar la pérdida de paquetes adecuadamente, los datos no observados no afectan a la estimación de los parámetros del sistema ni del retardo, excepto cuando la tasa de pérdida de paquetes es superior al 70 %.

El método propuesto permite estimar el retardo de tiempo mientras que otros métodos no lo hacen. Además, otros métodos disponibles en la literatura involucran descenso en el gradiente para resolver el problema de estimación [10], [11]. A diferencia de lo anterior, en esta tesis la estimación de los parámetros del sistema se realiza a través del algoritmo EM. Finalmente, en este trabajo, la pérdida de datos se modela como un proceso Bernoulli, a diferencia de otros trabajos, donde los datos perdidos se pueden asumir que no son observados de manera estructural en un modelo de espacio de estados [12] o pueden ser modelados como sistemas lineales de salto Markoviano [6].

1.3. Publicación asociada a la Tesis

Artículo en Conferencia:

a) F. Arancibia, R. Carvajal and J.C. Agüero, "Parameter Estimation over Wired Networks with Time-Delay and Packet Collision", in *2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, Valparaíso, Chile, 2019, pp. 1-6.

Abstract: In this paper, we address a parameters estimation problem of a dynamic system using an Output Error model, in which the output signal is transmitted through wired networks. We consider that the wired network may cause unknown delays and collision packet in data. We use the Akaike information criterion to estimate the time delay caused by wired network. Furthermore, due to the packet loss caused by wired network, we use Expectation Maximization algorithm to estimate the dynamic system parameters. Finally, we test the proposed algorithm effectiveness with numerical examples and a sensitivity analysis.

1.4. Estructura de la Tesis

Capítulo 1. Se introducen los conceptos básicos y generales que determinarán la temática de la investigación. Se visualizará la estructura general de la tesis, comenzando desde la descripción de la problemática hasta la obtención de resultados experimentales.

Capítulo 2. Se explican los conceptos fundamentales en relación a la estimación por máxima verosimilitud. Se presenta el algoritmo EM, el cual permite obtener el estimador de máxima verosimilitud para problemas con datos latentes y además se introduce la selección de modelos, en particular el criterio de información de Akaike.

Capítulo 3. Se exponen conceptos básicos sobre redes de computadores y comunicaciones. Se detallan las causas de la pérdida de datos y los tipos de retardos generados por las redes de comunicaciones.

Capítulo 4. Se introduce el área de sistemas de control sobre redes, presentando los componentes que este presenta. Se muestra el modelado con pérdida de datos y retardos, en el cual nos basamos en esta tesis.

Capítulo 5. Se muestra cómo se modelará el canal de comunicación. Se expone la solución al problema, realizando una estimación conjunta de los parámetros del sistema dinámico presentado. Se desarrolla un algoritmo iterativo basado en EM para solucionar la problemática presentada en la tesis.

Capítulo 6. Se muestran distintas simulaciones, en las cuales varían las restricciones generadas por el canal de comunicación. Se presentan los resultados obtenidos por el algoritmo desarrollado.

Capítulo 7. Se presentan las conclusiones del trabajo realizado en la tesis y se discuten los alcances sobre trabajos a futuro.

Capítulo 2

IDENTIFICACIÓN DE SISTEMAS POR MÁXIMA VEROSIMILITUD

La Identificación de Sistemas se utiliza para modelar matemáticamente un sistema dinámico mediante la utilización de los datos disponibles, los cuales suelen ser las señales de entradas y las mediciones (salidas) del sistema. Típicamente los datos medidos incorporan ruido, perturbaciones o incertidumbres, por lo que a través del modelado de procesos estocásticos se busca obtener la formulación matemática de la información no disponible. Los métodos más comunes para encontrar los estimadores de un sistema son el método de los momentos [13], mínimos cuadrados [8], estimadores bayesianos [14], entre otros.

En el año 1922 el método de estimación por Máxima Verosimilitud o ML (por sus siglas en inglés *Maximum Likelihood*) se popularizó a los desarrollos de R. A. Fisher [15]. Sin embargo, este método fue utilizado por varios matemáticos y estadísticos de diferentes épocas. Posteriormente, este método fue implementado en el área de identificación de sistemas por K. Astrom y T. Bohlin [16] siendo ampliamente utilizado en la actualidad [17], [18], [10], [19].

2.1. Estimación por Máxima Verosimilitud

Supongamos una muestra aleatoria de datos observados $\mathbf{y}_1, \cdots, \mathbf{y}_N$ de N observaciones generada por una función de densidad de probabilidad o pdf (por sus siglas en inglés *probability density function*) desconocida $p(\mathbf{y}|\theta_0)$. Definiendo el vector de parámetros como $\theta = \begin{bmatrix}\theta_1 & \cdots & \theta_k\end{bmatrix}$, existe una familia paramétrica $\{p(\mathbf{y}|\theta)|\theta \in \Theta\}$, siendo Θ el espacio paramétrico, el cual es el conjunto de posibles valores de θ. Por lo tanto $\theta = \theta_0$ es el verdadero valor del parámetro.

El objetivo de la estimación por ML es encontrar el vector de parámetros θ más probable que haya generado la muestra de datos observados [20], [21], [22]. Para esto, en primer lugar se debe encontrar la función de densidad conjunta de la muestra de datos observados definida como:

$$\mathcal{L}_N(\theta) = p(\mathbf{y}_1, \cdots, \mathbf{y}_N|\theta), \tag{2.1.1}$$

siendo esta la **función de verosimilitud**. Cuando las muestras son independientes e idénticamente distribuidas (iid) la función de verosimilitud se puede reorganizar como:

$$\mathcal{L}_N(\theta) = \prod_{i=1}^{N} p(\mathbf{y}_i|\theta). \tag{2.1.2}$$

Por lo tanto, la idea principal de método de estimación por máxima verosimilitud es encoger un estimador $\theta \in \Theta$ tal que maximice la función de verosimilitud. Este valor es el **estimador de máxima verosimilitud** o MLE (por sus siglas en inglés, *Maximum Likelihood Estimator*) y se denota por $\hat{\theta}_{ML}$. El problema se puede definir como:

$$\hat{\theta}_{ML} = \arg\max_{\theta\in\Theta} \mathcal{L}_N(\theta). \tag{2.1.3}$$

En la práctica, encontrar el estimador de máxima verosimilitud usando la ecuación (2.1.3) es complejo, por lo que conviene trabajar con el logaritmo natural de la función de verosimilitud:

$$\ell_N(\theta) = \ln\left[\mathcal{L}_N(\theta)\right], \tag{2.1.4}$$

siendo esta la **función log-verosimilitud**. Debido a que la función logaritmo natural es monótonamente creciente, el máximo de la función (2.1.4) se logra con el mismo valor θ, siendo equivalente maximizar ambas funciones [22]. Así el estimador de máxima verosimilitud se escribe como:

$$\hat{\theta}_{ML} = \arg\max_{\theta\in\Theta} \ell_N(\theta). \tag{2.1.5}$$

Si la función log-verosimilitud es diferenciable con respecto a θ, se define la función *Score* [22]:

$$\nabla\ell_N(\theta) = \left[\frac{\partial\ell_N(\theta)}{\partial\theta_1} \quad \cdots \quad \frac{\partial\ell_N(\theta)}{\partial\theta_k}\right]^T \text{ cuando } \Theta \subset \mathbf{R}^k. \tag{2.1.6}$$

De esta forma, las condiciones necesarias para que exista un máximo (o mínimo) son:

$$\nabla\ell_N(\theta) = \begin{bmatrix}0 & \cdots & 0\end{bmatrix}^T. \tag{2.1.7}$$

Estas ecuaciones pueden resolver de manera analítica el problema de maximización para algunos casos. En caso contrario la estimación de máxima verosimilitud debe realizarse usando optimizaciones numéricas.

2.1.1. Propiedades del MLE

El MLE posee propiedades deseables en un estimador, siendo las razones por la cuales se utiliza este método para estimar parámetros de un modelo paramétrico [22]. En primer lugar, se tiene que el estimador de máxima verosimilitud es invariante bajo transformaciones de θ.

Teorema 2.1. ***(Propiedad de invarianza)*** *Si $\hat{\theta}_{ML}$ es el estimador de máxima verosimilitud, para cualquier transformación $g(\theta)$ de θ, se tiene que el estimador de máxima verosimilitud de $g(\theta)$ es $g(\hat{\theta}_{ML})$.*

Demostración. Ver [22]. □

Además, a pesar de que no hay garantías de que un MLE sea un estimador insesgado, bajo ciertas condiciones de regularidad la estimación ML posee las siguientes propiedades a medida de que el tamaño de la muestra aumenta:

Teorema 2.2. *$\hat{\theta}_{ML}$ es un estimador asintóticamente insesgado de θ.*

Demostración. Ver [22]. □

Teorema 2.3. ***(Consistencia)*** *Sea $\mathbf{y}_1, \cdots, \mathbf{y}_N$ una muestra aleatoria de N observaciones independientes e idénticamente distribuidas. Si $\hat{\theta}_{ML}$ el estimador de máxima verosimilitud obtenido con la muestra de datos observados, entonces $\hat{\theta}_{ML}$ es consistente. Es decir:*

$$\hat{\theta}_{ML} \xrightarrow{p} \theta_0 \quad \textit{cuando } N \to \infty. \tag{2.1.8}$$

Demostración. Ver [22]. □

Teorema 2.4. ***(Normalidad Asintótica)*** *Sea $\mathbf{y}_1, \cdots, \mathbf{y}_N$ una muestra aleatoria de N observaciones independientes e idénticamente distribuidas. Si $\hat{\theta}_{ML}$ es el estimador de máxima verosimilitud obtenido con la muestra de datos observados, y bajo ciertas condiciones, el estimador converge en distribución a una variable Gaussiana, específicamente:*

$$\sqrt{N}(\hat{\theta}_{ML} - \theta_0) \xrightarrow{d} \mathcal{N}(0, \mathcal{I}(\theta)^{-1}), \tag{2.1.9}$$

donde $\mathcal{I}(\theta)$ es la matriz de información de Fisher [23].

Demostración. Ver [22]. □

Teorema 2.5. ***(Eficiencia)*** *El estimador de máxima verosimilitud $\hat{\theta}_{ML}$ es eficiente, es decir, alcanza de asintóticamente la cota inferior de Cramer-Rao.*

Demostración. Ver [22]. □

2.2. Algoritmo EM

El algoritmo EM es un método iterativo que se utiliza para obtener el MLE cuando hay variables latentes o escondidas (como variables internas o de estado de un sistema, o mediciones no adquiridas) [24]. Incluso para problemas donde no hay variables latentes, el cálculo del estimador de máxima verosimilitud se puede facilitar usando este método [25].

En el algoritmo EM se define el conjunto de datos observados $\mathcal{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_N\}$, provenientes de la función de densidad de probabilidad $p(\mathbf{y}|\theta)$, con $\theta \in \Theta$ el vector de parámetros del sistema. Se define además el conjunto de datos completos $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$, donde $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ es el conjunto de variables latentes (o variables ocultas).

Utilizando el Teorema de Bayes [25] se tiene que:

$$p(\mathcal{X}, \mathcal{Y}|\theta) = p(\mathcal{X}|\mathcal{Y}, \theta)p(\mathcal{Y}|\theta). \tag{2.2.1}$$

Aplicando logaritmo a (2.2.1), y reordenando se tiene:

$$\log p(\mathcal{Y}|\theta) = \log p(\mathcal{X}, \mathcal{Y}|\theta) - \log p(\mathcal{X}|\mathcal{Y}, \theta), \tag{2.2.2}$$

$$\ell_N(\theta) = \log p(\mathcal{X}, \mathcal{Y}|\theta) - \log p(\mathcal{X}|\mathcal{Y}, \theta). \tag{2.2.3}$$

Supongamos que se dispone de una estimación de θ, $\hat{\theta}^{(i)}$, en la i-ésima iteración. Ya que sólo los datos observables están disponibles, se aplica la esperanza condicional respecto a la distribución del conjunto de datos completos $\mathcal{Z}$ dado los datos observados $\mathcal{Y}$ en (2.2.3) y la estimación actual de los parámetros $\hat{\theta}^{(i)}$:

$$\ell_N(\theta) = \mathcal{Q}(\theta, \hat{\theta}^{(i)}) - \mathcal{H}(\theta, \hat{\theta}^{(i)}), \tag{2.2.4}$$

donde:

$$\mathcal{Q}(\theta, \hat{\theta}^{(i)}) = \mathbb{E}_{\mathcal{Z}}\left\{\log p(\mathcal{Z}|\theta)|\mathcal{Y}, \hat{\theta}^{(i)}\right\}, \tag{2.2.5}$$

$$\mathcal{H}(\theta, \hat{\theta}^{(i)}) = \mathbb{E}_{\mathcal{Z}}\left\{\log p(\mathcal{X}|\mathcal{Y}, \theta)|\mathcal{Y}, \hat{\theta}^{(i)}\right\}. \tag{2.2.6}$$

De (2.2.4) se deduce que:

$$\ell_N(\theta) - \ell_N(\hat{\theta}^{(i)}) = \left(\mathcal{Q}(\theta, \hat{\theta}^{(i)}) - \mathcal{Q}(\hat{\theta}^{(i)}, \hat{\theta}^{(i)})\right) - \left(\mathcal{H}(\theta, \hat{\theta}^{(i)}) - \mathcal{H}(\hat{\theta}^{(i)}, \hat{\theta}^{(i)})\right), \tag{2.2.7}$$

lo que se puede reescribir como:

$$p(\mathcal{X}|\mathcal{Y}, \theta) = \frac{p(\mathcal{X}, \mathcal{Y}|\theta)}{p(\mathcal{Y}|\theta)} = p(\mathcal{Z}|\mathcal{Y}, \theta). \tag{2.2.8}$$

Entonces tenemos:

$$\begin{aligned}\mathcal{H}(\theta,\hat{\theta}^{(i)})-\mathcal{H}(\hat{\theta}^{(i)},\hat{\theta}^{(i)}) &= \mathbb{E}_{\mathcal{Z}}\left\{\log p(\mathcal{Z}|\mathcal{Y},\theta)|\mathcal{Y},\hat{\theta}^{(i)}\right\}-\mathbb{E}_{\mathcal{Z}}\left\{\log p(\mathcal{Z}|\mathcal{Y},\hat{\theta}^{(i)})|\mathcal{Y},\hat{\theta}^{(i)}\right\}\\ &= \mathbb{E}_{\mathcal{Z}}\left\{\log p(\mathcal{Z}|\mathcal{Y},\theta)-\log p(\mathcal{Z}|\mathcal{Y},\hat{\theta}^{(i)})|\mathcal{Y},\hat{\theta}^{(i)}\right\}\\ &= \mathbb{E}_{\mathcal{Z}}\left\{\log\frac{p(\mathcal{Z}|\mathcal{Y},\theta)}{p(\mathcal{Z}|\mathcal{Y},\hat{\theta}^{(i)})}|\mathcal{Y},\hat{\theta}^{(i)}\right\},\end{aligned} \tag{2.2.9}$$

y usando la desigualdad de Jensen [26], se tiene que:

$$\begin{aligned}\mathbb{E}_{\mathcal{Z}}\left\{\log\frac{p(\mathcal{Z}|\mathcal{Y},\theta)}{p(\mathcal{Z}|\mathcal{Y},\hat{\theta}^{(i)})}|\mathcal{Y},\hat{\theta}^{(i)}\right\} &\leqslant \log\left[\mathbb{E}_{\mathcal{Z}}\left\{\frac{p(\mathcal{Z}|\mathcal{Y},\theta)}{p(\mathcal{Z}|\mathcal{Y},\hat{\theta}^{(i)})}|\mathcal{Y},\hat{\theta}^{(i)}\right\}\right]\\ &= \log\left[\int\frac{p(\mathcal{Z}|\mathcal{Y},\theta)}{p(\mathcal{Z}|\mathcal{Y},\hat{\theta}^{(i)})}p(\mathcal{Z}|\mathcal{Y},\hat{\theta}^{(i)})d\mathcal{Z}\right]\\ &= \log\left[\int p(\mathcal{Z}|\mathcal{Y},\theta)d\mathcal{Z}\right]\\ &= 0.\end{aligned} \tag{2.2.10}$$

Por lo tanto, para cualquier valor de θ, la función $\mathcal{H}(\theta,\hat{\theta}^{(i)})$ es decreciente. Además, si se escoge una secuencia de parámetros $\{\hat{\theta}^{(i)}\}$ de tal manera que se cumpla:

$$\mathcal{Q}(\hat{\theta}^{(i+1)},\hat{\theta}^{(i)}) \geqslant \mathcal{Q}(\theta,\hat{\theta}^{(i)}), \tag{2.2.11}$$

aseguramos que la función log-verosimilitud cumpla:

$$\ell_N(\hat{\theta}^{(i+1)}) \geqslant \ell_N(\hat{\theta}^{(i)}). \tag{2.2.12}$$

Esto muestra que la función log-verosimilitud $\ell_N(\theta)$ no disminuye después de una iteración del algoritmo EM. Si la desigualdad (2.2.11) es estricta, la función de verosimilitud aumentará. Entonces, para una secuencia acotada de valores de la función de log-verosimilitud $\{\ell_N(\hat{\theta}^{(i)})\}$, $\ell_N(\hat{\theta}^{(i)})$ converge monótonamente a algún valor. Entonces, la optimización se realiza maximizando la función auxiliar $\mathcal{Q}(\theta,\hat{\theta}^{(i)})$, ya que maximizando $\mathcal{Q}(\theta,\hat{\theta}^{(i)})$, la estimación obtenida $\hat{\theta}^{(i+1)}$ aumentará el valor de la función de verosimilitud hasta converger a un punto de silla de $\ell(\theta)$ [27].

2.2.1. Convergencia del algoritmo EM

A pesar de que $\ell_N(\hat{\theta}^{(i)})$ converge monótonamente a algún valor, este valor puede ser un máximo local. Además, si $\ell_N(\hat{\theta}^{(i)})$ tiene varios puntos estacionarios, tales como puntos de silla o máximo global o local, la convergencia usando el algoritmo EM a un punto estacionario depende de la condición inicial $\hat{\theta}^{(0)}$ [27].

Si realizamos la derivada respecto a θ de (2.2.4) se tiene:

$$\frac{\partial \ell_N(\theta)}{\partial \theta} = \frac{\partial \mathcal{Q}(\theta, \hat{\theta}^{(i)})}{\partial \theta} - \frac{\partial \mathcal{H}(\theta, \hat{\theta}^{(i)})}{\partial \theta}. \tag{2.2.13}$$

Además, la desigualdad (2.2.9) implica que:

$$\mathcal{H}(\theta, \hat{\theta}^{(i)}) \leqslant \mathcal{H}(\hat{\theta}^{(i)}, \hat{\theta}^{(i)}), \tag{2.2.14}$$

entonces:

$$\left.\frac{\partial \mathcal{H}(\theta, \hat{\theta}^{(i)})}{\partial \theta}\right|_{\theta=\hat{\theta}^{(i)}} = 0. \tag{2.2.15}$$

Sea $\theta^\star$ un vector de parámetros arbitrario. Si se elige $\hat{\theta}^{(i)} = \theta^\star$, se tiene de (2.2.15) que:

$$\left.\frac{\partial \ell_N(\theta)}{\partial \theta}\right|_{\theta=\theta^\star} = \left.\frac{\partial \mathcal{Q}(\theta, \theta^\star)}{\partial \theta}\right|_{\theta=\theta^\star}. \tag{2.2.16}$$

Supongamos que $\theta = \theta^*$ es un punto estacionario de $\ell_N(\theta)$, se tiene de (2.2.16) que:

$$\left.\frac{\partial \ell_N(\theta)}{\partial \theta}\right|_{\theta=\theta^*} = \left.\frac{\partial \mathcal{Q}(\theta, \theta^*)}{\partial \theta}\right|_{\theta=\theta^*} = 0. \tag{2.2.17}$$

Se puede ver de (2.2.17) que el algoritmo EM puede converger a puntos estacionarios, tales como puntos de silla o máximos locales, si $\mathcal{Q}(\theta, \hat{\theta}^{(i)})$ es maximizado en el punto θ^*.

2.2.2. Procedimiento

Entonces el algoritmo EM busca encontrar el estimador de máxima verosimilitud resolviendo la ecuación de datos incompletos (2.1.7) indirectamente mediante un proceso iterativo en términos de la función log-verosimilitud de los datos completos. Este proceso iterativo consta de dos pasos, los cuales justifican el nombre del algoritmo. Específicamente, se comienza con una estimación inicial de los parámetros $\hat{\theta}^{(0)}$ y luego comienza el proceso iterativo:

Paso E: Obtener la función auxiliar:

$$\mathcal{Q}(\theta, \hat{\theta}^{(i)}) = \mathbb{E}_{\mathcal{Z}} \left\{ \log \left[p(\mathcal{Z}|\theta) \right] | \mathcal{Y}, \hat{\theta}^{(i)} \right\} \tag{2.2.18}$$

Paso M: Calcular estimación $\hat{\theta}^{(i+1)}$, tal que maximice función auxiliar:

$$\hat{\theta}^{(i+1)} = \arg\max_{\theta} \mathcal{Q}(\theta, \hat{\theta}^{(i)}). \tag{2.2.19}$$

Detener en caso de cumplir algún criterio de convergencia definido previamente; en caso contrario, volver al **Paso E**.

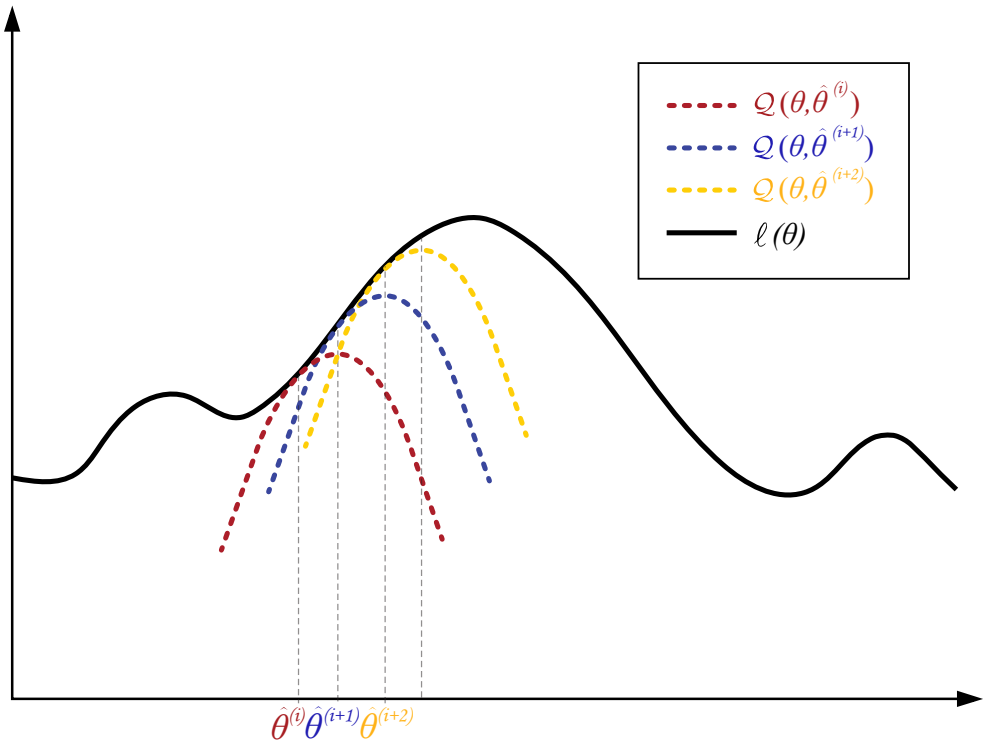

Figura 2.1. Representación gráfica del algoritmo EM

En la Figura 2.1 se observa gráficamente el procedimiento realizado por el algoritmo EM en términos de la función auxiliar $\mathcal{Q}(\theta, \hat{\theta}^{(i)})$. En la i-ésima iteración, se obtiene una función auxiliar $\mathcal{Q}(\theta, \hat{\theta}^{(i)})$ usando la estimación $\hat{\theta}^{(i)}$ (representada en rojo). Luego se maximiza esta función auxiliar con respecto a θ, obteniendo una nueva estimación $\hat{\theta}^{(i+1)}$. Se calcula nuevamente la función auxiliar a partir de esta nueva estimación, obteniendo una nueva función $\mathcal{Q}(\theta, \hat{\theta}^{(i+1)})$ (representada en azul). Posteriormente, esta función se maximiza con respecto θ, produciendo una nueva estimación $\hat{\theta}^{(i+2)}$, y a partir de esta estimación, se obtiene una nueva función auxiliar $\mathcal{Q}(\theta, \hat{\theta}^{(i+2)})$, la cual se maximiza. Este procedimiento continúa hasta que se logra el criterio de convergencia definido. De esta manera, el valor estimado de θ al finalizar el algoritmo EM puede ser igual al MLE (esto se observa en el caso de la Figura 2.1).

2.3. Selección de modelos

Tradicionalmente los métodos de estimación de parámetros proporcionan un procedimiento para estimar parámetros desconocidos de un modelo paramétrico con dimensiones y estructura especificados. En algunas ocasiones la estimación de parámetros se realiza desconociendo la identidad en general de este. Es por esto que la selección del modelo es una tarea primordial en la investigación científica, donde su objetivo es seleccionar un modelo estadístico a partir de un conjunto de modelos propuestos [28], [29]. En este contexto, se asumen datos empíricos obtenidos de un modelo único y verdadero, y se realiza un análisis estadístico para encontrar entre los propuestos el modelo que se ajusta de mejor forma a estos. El término mejor es bastante discutido, pues éste depende del enfoque realizado, ya que muchos métodos de selección utilizados ni siquiera se basan en un criterio explícito de cuál es el mejor modelo [28].

Por ejemplo, podemos considerar dos modelos paramétricos y encontrar el MLE de cada uno en base a los datos observados. Así, el modelo paramétrico más complejo puede ajustarse de forma más precisa a los datos. Si nos basamos en esto, lo primordial sería proponer modelos más complejos para obtener un mejor ajuste a los datos reales. Sin embargo, no siempre esta complejidad es necesaria en el ámbito científico, pues en general se desea obtener un modelo simple para generar observaciones a través de una simulación. Debido a esto, una buena forma de selección de modelos es equilibrar el ajuste con la complejidad, considerando la complejidad como el número de parámetros del modelo propuesto. Luego, la selección de modelo debe definir un criterio bien justificado de cuál es el "mejor" modelo, donde este criterio debe considerar el hecho de que los datos son finitos y "ruidosos". Este criterio debe ajustarse a un marco de inferencia estadística, tal como inferencia bayesiana o a través de estimación por máxima verosimilitud. Además, este criterio tiene que poder representarse en un valor cuantificado para cada modelo propuesto, por que la inferencia se basa en el conjunto completo de modelos. Para la selección de modelos generalmente se utilizan dos enfoques: la selección de modelos basada en teoría de la información, específicamente en la divergencia de Kullback-Leibler [30] y la selección de modelos bayesiana basada en factores de Bayes [31].

2.3.1. Información de Kullback-Leibler

En 1951 la divergencia de Kullback-Leibler fue introducida por S. Kullback y R. A. Leibler [30], donde consideraban una medida de distancia o divergencia entre dos distribuciones.

Se denota f como la verdadera distribución y g el modelo aproximado, donde $f = p(\mathbf{x}|\theta_0)$ con θ_0 es el vector de parámetros verdadero y $g = p(\mathbf{x}|\theta)$ con $\theta \in \Theta \subset \mathbf{R}^k$. La divergencia de Kullback-Leibler se refiere a la información perdida al usar el modelo g para aproximar una distribución realizada por f [32].

Para las distribuciones f y g de una variable aleatoria continua, la divergencia

Kullback-Leibler se define como la integral:

$$D_{KL}(f,q) = \int_{\Omega} p(\mathbf{x}|\theta_0) \log\left(\frac{p(\mathbf{x}|\theta_0)}{p(\mathbf{x}|\theta)}\right) d\mathbf{x}, \tag{2.3.1}$$

mientras que para variables aleatorias discretas, la divergencia de Kullback-Leibler se define como la sumatoria:

$$D_{KL}(f,q) = \sum_{\mathbf{x}\in\Omega} p(\mathbf{x}|\theta_0) \log\left(\frac{p(\mathbf{x}|\theta_0)}{p(\mathbf{x}|\theta)}\right), \tag{2.3.2}$$

donde Ω es el conjunto donde se miden las distribuciones. La notación $D_{KL}(f,q)$ denota la información perdida cuando g es usado para aproximar f.

Entonces se busca el mejor modelo de un conjunto de modelos, en el sentido de que éste pierda la mínima información posible. Es decir, se busca el modelo g que minimice $D_{KL}(f,q)$. La divergencia de Kullback-Leibler es siempre no negativa, es decir:

$$D_{KL}(f,q) \geqslant 0, \tag{2.3.3}$$

con $D_{KL}(f,q) = 0$ si y solo si $f = g$ en casi todo punto. Además, en términos formales la distribución de Kullback-Leibler no es una distancia métrica pues no es simétrica, es decir:

$$D_{KL}(f,q) \neq D_{KL}(g,f), \tag{2.3.4}$$

ni cumple con la desigualdad triangular [26].

A pesar de sus virtudes, no se puede usar la divergencia de Kullback-Leibler como un criterio de selección de modelos debido a que se requiere el conocimiento del modelo real. Cuando no se cuenta con el modelo generador, se debe usar la divergencia de K-L como una base para un criterio de selección de modelos.

La información de Kullback-Leibler para distribuciones continuas puede expresarse como:

$$D_{KL}(f,q) = \int_{\Omega} p(\mathbf{x}|\theta_0) \log p(\mathbf{x}|\theta_0) d\mathbf{x} - \int_{\Omega} p(\mathbf{x}|\theta_0) \log p(\mathbf{x}|\theta) d\mathbf{x}, \tag{2.3.5}$$

lo que se puede reescribir como:

$$D_{KL}(f,g) = \mathbb{E}_f\{\log p(\mathbf{x}|\theta_0)\} - \mathbb{E}_f\{\log p(\mathbf{x}|\theta)\}, \tag{2.3.6}$$

donde el primer término es una constante que depende solo de la distribución verdadera y denotaremos como C. Luego:

$$D_{KL}(f,g) = C - \mathbb{E}_f\{\log p(\mathbf{x}|\theta)\}. \tag{2.3.7}$$

En el análisis de los datos observados se desconoce el valor de f. Por lo tanto, solo el término derecho de la igualdad (2.3.7), llamada la divergencia relativa de Kullback-Leibler, debe estimarse para cada modelo del conjunto.

2.3.2. Criterio de información de Akaike

Hirotugu Akaike propuso en [33], [34] el uso de la divergencia de Kullback-Leibler como base para la selección de modelos. Debido a que ésta necesita el conocimiento de la distribución verdadera, f no se puede calcular directamente. Por esto, Akaike propone una forma metódica de selección de modelos usando los datos observados y la función log-verosimilitud evaluada en el MLE.

Akaike muestra que debemos usar la esperanza de la divergencia de K-L para medir la "distancia" entre f y g. Así se tiene:

$$\mathbb{E}_{\mathcal{Y}}\left\{D_{KL}(f,g)\right\} = \int_{\Omega} p(\mathbf{x}|\theta_0)\log p(\mathbf{x}|\theta_0)d\mathbf{x} - \int_{\Omega} p(\mathbf{y}|\theta_0)\left[\int_{\Omega} p(\mathbf{x}|\theta_0)\log p(\mathbf{x}|\hat{\theta}(\mathbf{y}))d\mathbf{x}\right]d\mathbf{y}. \quad (2.3.8)$$

Entonces para obtener un criterio de selección de modelos basado en la divergencia de K-L, se debe estimar:

$$\mathbb{E}_{\mathcal{Y}}\mathbb{E}_{\mathcal{X}}\left\{\log g(\mathcal{X}|\hat{\theta}(\mathcal{Y}))\right\} = \int_{\Omega} p(\mathbf{y}|\theta_0)\left[\int_{\Omega} p(\mathbf{x}|\theta_0)\log p(\mathbf{x}|\hat{\theta}(\mathbf{y}))d\mathbf{x}\right]d\mathbf{y}. \quad (2.3.9)$$

De esta forma, Akaike encontró una relación formal entre la divergencia de K-L y la Teoría de la Verosimilitud. Descubrió que el valor de log-verosimilitud evaluado en el MLE era una estimación sesgada de (2.3.9), donde este sesgo es aproximadamente igual al número de parámetros estimables del modelo propuesto. Por lo tanto, un estimador aproximadamente insesgado de (2.3.9) para muestras grandes y modelos "buenos" es:

$$\log \mathcal{L}_N(\hat{\theta}_{MV}|\mathcal{Y}) - \dim_\theta, \quad (2.3.10)$$

donde $\ell_N(\hat{\theta}_{MV}) = \log \mathcal{L}_N(\hat{\theta}_{MV}|\mathcal{Y})$ es la función log-verosimilitud evaluado en el MLE y $\dim_\theta$ es el número de parámetros del modelo.

Con esto Akaike definió un criterio de información multiplicando el término (2.3.10) por -2, obteniendo:

$$\text{AIC} = -2\ell_N(\hat{\theta}) + 2\dim_\theta . \quad (2.3.11)$$

Esto se conoce como criterio de información de Akaike o AIC (por sus siglas en inglés, *Akaike's Information Criterion*). Con este se tiene una estimación de la distancia relativa esperada entre el modelo ajustado y el mecanismo verdadero desconocido, el cual generó los datos observados. Así los modelos simples tendrán un valor mayor en términos de bondad de ajustes, pero pequeños en términos de penalización debido a una menor cantidad de parámetros. Al contrario, si el modelo es complejo el valor que tomará la bondad de ajuste será menor, pero se tendrá una mayor penalización.

El procedimiento que se debe realizar para seleccionar el modelo correcto es calcular AIC como está escrito en (2.3.11) para cada uno de los modelos propuestos

y seleccionar el modelo que tiene menor valor de AIC. A pesar de que este criterio elige el mejor modelo del conjunto, si todos los modelos son deficientes en un sentido absoluto, AIC no nos advierte sobre esto.

En este contexto supongamos que tenemos un conjunto de modelos $\mathcal{M}_1, \ldots, \mathcal{M}_k$ donde cada modelo tiene una función de probabilidad:

$$\mathcal{M}_j = \{p_j(\mathrm{y}|\theta_j) : \theta_j \in \Theta_j\}. \tag{2.3.12}$$

Sea $\hat{\theta}_j$ el MLE del modelo $\mathcal{M}_j$. De esta forma el AIC de cada modelo es:

$$\mathrm{AIC}_j = -2\ell_N(\hat{\theta}_j) + 2\dim_{\theta_j}, \tag{2.3.13}$$

y se debe seleccionar el modelo cuyo valor AIC_j sea mínimo.

2.3.3. AIC para muestreos pequeños

Cuando el número de parámetros estimables es grande en relación al número de los datos observadas N existe una gran probabilidad de que AIC elija modelos sobreajustados, es decir, con muchos parámetros. Para estos casos existe una versión de AIC que corrige el sesgo de segundo orden, la cual se llama AIC_c, y se denota como:

$$\mathrm{AIC}_c = \mathrm{AIC} + \frac{2\dim_\theta(\dim_\theta + 1)}{N - \dim_\theta - 1}. \tag{2.3.14}$$

En [28] recomiendan usar (2.3.14) cuando $N/\dim_\theta < 40$. Debido a que AIC_c converge a AIC cuando $N \to \infty$ en la práctica siempre podríamos utilizar esta forma corregida.

2.4. Algoritmo EM y AIC

Como se observa en la ecuación 2.3.11, la función que se ocupa para calcular el valor AIC es la función de verosimilitud para los datos completos. Para un conjunto completo con datos latentes es necesario ocupar el algoritmo EM, y como se expuso en la sección 2.2 y se observa en la Figura 2.1, el algoritmo EM converge al estimador de máxima verosimilitud mediante un proceso iterativo. Entonces, para obtener el valor correspondiente AIC se puede realizar el algoritmo EM hasta la convergencia, y reemplazar la función de verosimilitud en la ecuación 2.3.11 por el valor de la función auxiliar del algoritmo EM evaluada en el estimador de máxima verosimilitud obtenido [35]. Es decir, se tiene

$$AIC = -2\mathcal{Q}(\hat{\theta}^{(i+1)}, \hat{\theta}^{(i)}) + 2\dim_\theta, \tag{2.4.1}$$

donde $\dim_\theta$ es el número de parámetros estimables del modelo propuesto.

Capítulo 3

REDES DE COMPUTADORES

Este Capítulo sirve como una descripción general de las redes de computadoras. En este se presentan conceptos y terminologías claves de esta área para comprender mejor los futuros capítulos.

Se explica el tipo de amenazas con las que debe lidiar día a día los sistemas de redes, y además presentamos los retardos y pérdida de paquetes, para apreciar porque se deben considerar como restricciones en nuestro modelado del problema.

3.1. Contexto

Internet es una red informática que interconecta miles de dispositivos de redes de comunicación en todo el mundo que utilizan la familia de protocolos TCP/IP. Estos dispositivos hoy en día pueden ser computadoras de escritorio, servidores, computadoras portátiles, televisores, entre otros. A estos dispositivos se les denomina *hosts*, sistemas finales o anfitriones.

Los sistemas finales están conectados entre sí por una red de enlaces de comunicación y conmutadores de paquetes. Cuando un *host* quiere enviar datos a otro *host*, el *host* origen segmenta los datos, resultando paquetes de información, también conocido como paquetes, y estos en el *host* destino se vuelven a unir en los datos originales.

Existen una gran variedad de conmutadores de paquetes, pero los dos tipos más usados son los enrutadores y los conmutadores de capa de enlace. Tanto uno como el otro envían paquetes hacia sus destinos. La secuencia de enlaces de comunicación entre los *hosts* involucrados se conoce como ruta a través de la red, o simplemente ruta.

Finalmente, los *hosts*, conmutadores y otras redes de comunicación se comunican a través de protocolos, los que controlan tanto el envío como la recepción de información. El Protocolo de control de transmisión y el Protocolo de Internet o TCP/IP (por sus siglas en inglés, *Transmission Control Protocol and the Internet Protocol*) es un conjunto de los protocolos más importantes de Internet.

3.2. Protocolos de comunicación

Un protocolo de comunicaciones es un sistema de reglas que definen el formato y el orden de la información intercambiada entre dos o más entidades comunicantes (computadores, celulares, entre otros). Este define las acciones que se deben tomar tanto en la transmisión como en la recepción, la sincronización de la comunicación y los posibles métodos de recuperación de errores.

Como se mencionó anteriormente la familia de protocolos de internet es uno de los conjuntos más importantes que permiten la transmisión de datos entre computadoras, el cual también se conoce como conjunto de protocolos TCP/IP. Entre estos protocolos hay dos que destacan en el envío de datos por la red, que son el protocolo de control de transmisión y el protocolo de datagramas de usuario. Ambos protocolos actúan en el nivel de transporte según el modelo OSI (por sus siglas en inglés, *Open Systems Interconnection*) [36].

3.2.1. Protocolo TCP

El protocolo de control de transmisión o TCP (por su siglas en inglés, *Transmission Control Protocol*) es uno de los principales protocolos del conjunto de protocolos de Internet y fue formalizado por Vint Cerf y Robert Kahn en 1974 [37], [38].

Este protocolo está orientado a la conexión, es decir, se establece una conexión entre el cliente y el servidor antes de que se envíen los datos. También ofrece un servicio de transferencia de datos confiable entre los agentes involucrados.

Estructura del segmento TCP

SEGMENTO TCP

<table>
<tr><td>Bits</td><td colspan="3">0-15</td><td colspan="2">16-31</td></tr>
<tr><td>0</td><td colspan="3">Puerto de origen</td><td colspan="2">Puerto de destino</td></tr>
<tr><td>32</td><td colspan="5">Número de secuencia</td></tr>
<tr><td>64</td><td colspan="5">número de acuse de recibo (ACK)
o número de acuse de recibo negativo (NACK)
dependiendo de la bandera activada</td></tr>
<tr><td>96</td><td>Longitud de cabecera</td><td>Reservado</td><td>Bits de bandera</td><td colspan="2">Tamaño de ventana</td></tr>
<tr><td>128</td><td colspan="3">Suma de verificación</td><td colspan="2">Puntero urgente</td></tr>
<tr><td>160</td><td colspan="5">Opciones TCP</td></tr>
<tr><td>224</td><td colspan="5">Datos</td></tr>
</table>

Figura 3.1. Estructura del segmento TCP

El cliente pasa un flujo de los datos a través del *socket* (la puerta del proceso),

y TCP se encarga de convertir este flujo de datos en fragmentos, en donde luego TCP conduce estos datos al búfer de envío de la conexión. TCP toma uno de los fragmentos de datos y le añade una cabecera, creando así un segmento TCP. Finalmente este segmento se desplaza como un datagrama del Protocolo de Internet (IP) para intercambiarse con el servidor [39].

El segmento TCP se puede observar en la Figura 3.1, el cual consta de la cabecera y el campo de datos. La cabecera del segmento TCP contiene los siguientes campos:

- Puertos: Tanto el puerto origen como el puerto destino emplean 16 bits, y este campo identifica la aplicación emisora y receptora junto a sus respectivas direcciones IP. La dirección IP en combinación con el puerto correspondiente se refiere al *socket*.
- Número de secuencia y número de acuse de recibo: Estos dos campos emplean 32 bits. Estos campos sirven para implementar un servicio de transferencia de datos confiable.
- Longitud de cabecera: este campo especifica el tamaño de la cabecera TCP en palabras de 32 bits. Es necesario ya que el campo de opciones TCP tiene un tamaño variable.
- Reservado: Contiene 3 bits. Es para usos futuros y debe estar en cero.
- Bits de bandera: Este campo contiene 6 bits, cada uno con diferentes banderas. El bit ACK se usa para indicar que el valor transportado en el campo de acuse de recibo es válido. Los bits RST, SYN y FIN se utilizan para configurar y desconectar la conexión. Específicamente, RST se utiliza para resetear la conexión, SYN es para sincronizar números de secuencia y solo el primer paquete enviado debe tener esta bandera activada y FIN indica el final de la transmisión de datos para finalizar una conexión TCP siendo el último paquete del remitente. El bit PSH indica que el receptor debe pasar los datos almacenados en búfer a la capa superior inmediatamente. Finalmente, el bit URG informa al servidor que hay datos urgentes y deben priorizarse.
- Tamaño de ventana: Este campo contiene 16 bits y sirve para indicar la cantidad de bytes que el receptor está dispuesto a recibir. Se utiliza para el control de flujo.
- Suma de verificación: Este campo contiene 16 bits. Este corresponde a una función que tiene como finalidad detectar los errores tanto en la cabecera como en los datos. Verifica que entre los valores obtenidos, al hacer la comprobación inicial, no difieran de la comprobación final tras realizarse la transmisión.
- Puntero urgente: Este campo contiene 16 bits. Este se establece cuando la bandera URG es 1. El valor de este campo indica la cantidad de datos del paquete a partir del primer byte, que el receptor considera datos urgentes.

- Opciones TCP: Este campo es para añadir características no integradas en la cabecera. Además, si la cabecera no termina con un tamaño múltiplo de 32 bits se agrega un relleno con ceros.

Operación de TCP

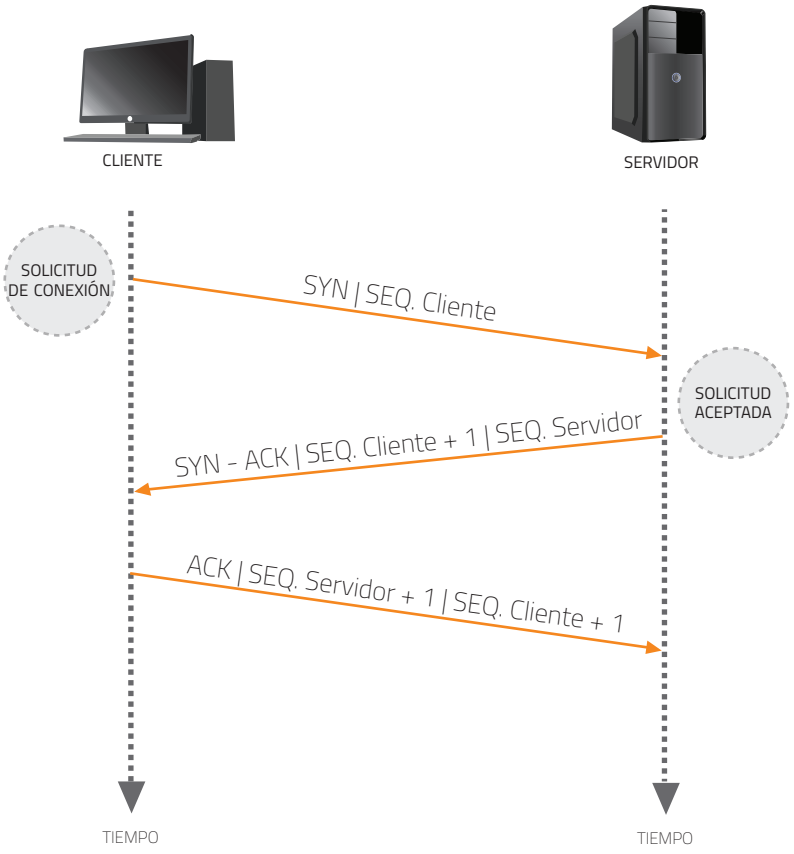

Figura 3.2. Establecimiento de la conexión en TCP

Las conexiones en TCP se componen de tres fases [40], [39]. En primer lugar, sucede el proceso de establecer la conexión fiable en la transferencia de datos. En la Figura 3.2 se observa que este proceso comienza con el cliente realizando una apertura de un puerto enviando un paquete SYN inicial al servidor como parte de establecer la conexión. Luego, el servidor comprueba si este puerto está abierto, en donde si este no lo está, el servidor envía al cliente un paquete de respuesta con el bit RST activado, lo que significa que se rechaza la conexión. En caso contrario, el servidor confirma la petición con un paquete SYN/ACK. Finalmente, el cliente responde al servidor con un ACK, estableciendo la conexión realizando un apretón

de manos por 3 vías o el proceso de *3-way handshake*.

Posteriormente, se procede a la transferencia de datos. En este proceso, gracias al número de secuencia, se evita el desorden de los paquetes. De la misma forma, la suma de verificación cumple la función de detectar errores, mientras que el acuse de recibo permite detectar pérdidas o retardos

Finalmente, ocurre la fase de cierre de conexión. Este proceso se presenta en la Figura 3.3, donde se observa que el cliente envía un paquete con el bit FIN activado y el servidor responde con un ACK. Luego el servidor envía un paquete con el bit FIN activado al cliente y este responde con un ACK, finalizando la conexión en un proceso de apretón de manos por 4 vías o *four-way handshake*.

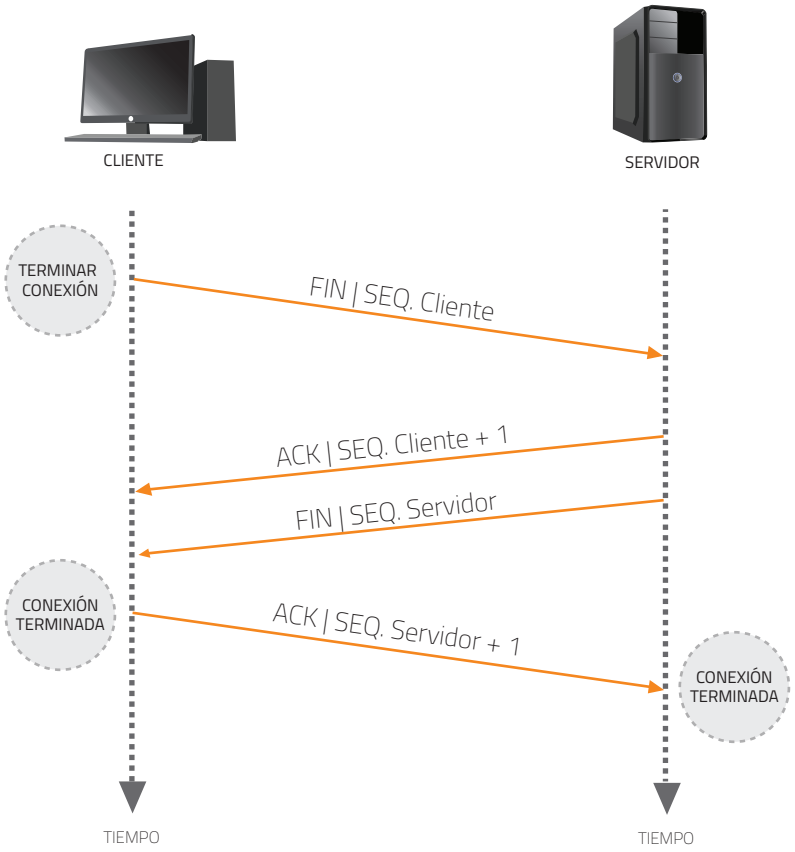

Figura 3.3. Fin de la conexión en TCP

3.2.2. Protocolo UDP

El protocolo de datagramas de usuario o UDP (por su siglas en inglés, *User Datagram Protocol*) es un protocolo del nivel de transporte que se basa en el intercambio

de datagramas. Este fue diseñado por David P. Reed en 1980, siendo documentado formalmente en [41]. A diferencia de TCP, la transmisión a través de la red se realiza sin que se haya establecido previamente una conexión entre el cliente y servidor. Este protocolo no es siempre fiable, ya que no emplea control de flujo, lo que puede hacer que los paquetes lleguen a su destino de forma desordenada. Además, debido a que no hay confirmación de entrega ni recepción no se sabe si hay pérdida en los datos.

Estructura del segmento UDP

El segmento UDP se puede observar en la Figura 3.4, y consta de la cabecera y el campo de datos [40]. La cabecera del segmento UDP contiene los siguientes campos:

- Puertos: Tanto el puerto origen como el puerto destino emplean 16 bits. Este campo identifica la aplicación emisora y receptora junto a sus respectivas direcciones IP. Debido a que en UDP el origen no solicita respuestas, el puerto de origen es opcional. Si este no se utiliza, se rellena con ceros.
- Longitud UDP: Este campo especifica la longitud en bytes del encabezado y los datos UDP.
- Suma de verificación: Este campo se puede utilizar en UDP para comprobar los errores del encabezado y los datos.

SEGMENTO UDP

Bits	0-15	16-31
0	Puerto de origen	Puerto de destino
32	Longitud UDP	Suma de verificación
64	Datos	

Figura 3.4. Estructura del segmento UDP

3.3. Dominio de Colisión

Se denomina dominio de colisión al segmento de red conectado por un medio compartido o mediante repetidores [42]. En este las transmisiones de datos simultáneas chocan entre sí, afectando tanto a las redes inalámbricas como a las primeras versiones de Ethernet. Como se observa en la Figura 3.5, una colisión de paquetes ocurre

cuando más de un nodo intenta enviar un paquete en un segmento de red al mismo tiempo.

En el dominio de colisión solo un dispositivo puede transmitir a la vez, mientras que los otros deben escuchar a la red y no transmitir mientras otros ya transmiten con el fin de evitar colisiones. Las colisiones reducen la eficiencia de la red, debido a que estas requieren que los nodos cancelen el envío y retransmitan los paquetes.

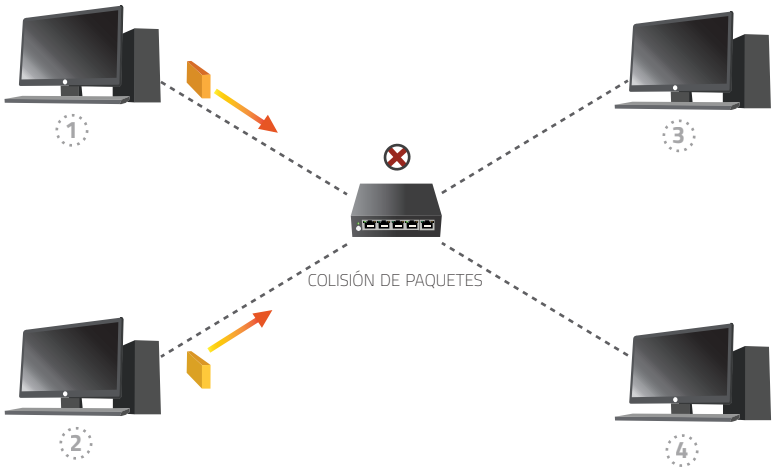

Figura 3.5. Esquema de colisión de paquetes

3.4. Capa de enlace

3.4.1. Sistema ALOHA

El sistema ALOHA (por sus siglas en inglés, *Additive Links On-line Hawaii Area*) [43], [44] es un sistema de redes informáticas desarrollado en la Universidad de Hawaii, el cual entró en funcionamiento en 1971. Este demostró públicamente una red inalámbrica por paquete de datos, siendo pionera para futuros protocolos de red como Ethernet.

En este sistema cada nodo puede transmitir un paquete de datos en cualquier momento, pero estos deben escuchar para saber si el canal está siendo utilizado, permitiendo que todos los medios puedan compartir el mismo medio al mismo tiempo.

Existe un problema en el sistema ALOHA, que es cuando dos nodos (o más) empiezan a transmitir aproximadamente al mismo tiempo. Los nodos deben escuchar un acuse de recibo, y en caso de no recibirlo significa que algo lo había impedido, siendo posiblemente una colisión debido a que se enviaron los paquetes simultáneamente, como se observa en la Figura 3.6. En este caso, se reenvía el paquete de datos, pero si este no recibe un acuse de recibo en reiteradas veces la transmisión finalmente no se realiza.

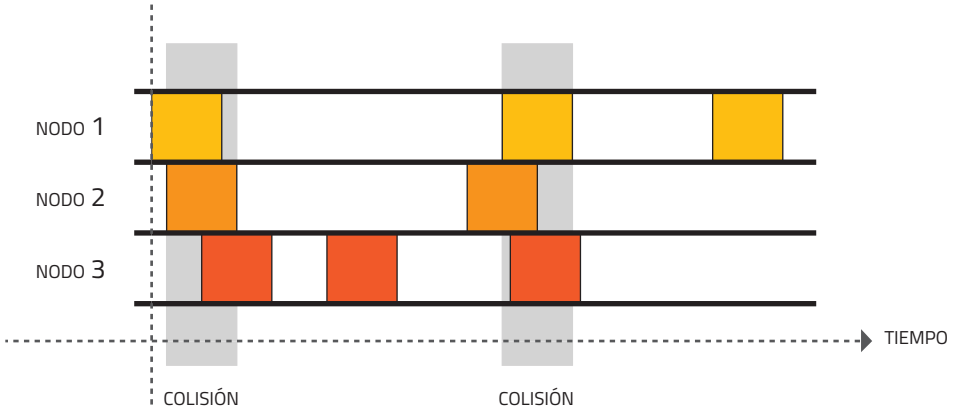

Figura 3.6. Ejemplo de transmisión en ALOHA

Si la red se satura, el número de colisiones puede crecer de forma drástica, llegando a un punto donde todos los paquetes colisionan. De esta manera la utilización máxima de este canal se encuentra alrededor del 18 %.

3.4.2. Sistema ALOHA ranurado

ALOHA ranurado, S-ALOHA o Slotted ALOHA [45], es una versión mejorada del protocolo original ALOHA, el cual introduciría ranuras de tiempo en el protocolo de envío, aumentando el rendimiento máximo hasta aproximadamente un 37 %.

En este sistema un nodo puede transmitir al comienzo de la ranura en un intervalo de tiempo definido, reduciendo así las colisiones. Un ejemplo se puede observar en la Figura 3.7, donde todos los nodos transmiten simultaneamente en un principio. Debido a esto, los nodos retransmiten en un momento posterior, hasta que logren transmitir los datos sin colisión.

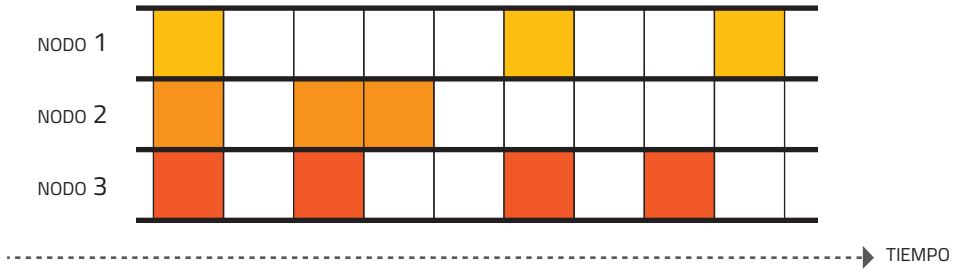

Figura 3.7. Ejemplo de transmisión en ALOHA ranurado

3.4.3. Acceso Múltiple con Escucha de Señal Portadora

Ya sea en ALOHA ranurado como puro, la decisión de un nodo de transmitir es independiente de la actividad de los otros nodos que se encuentran conectados al canal de transmisión. Un nodo transmitirá independientemente de la actividad de otro nodo. Para entender el comportamiento de la actividad de los nodos, podemos entenderlo como una discusión en la cual no existen modales al momento de la conversación. Por lo tanto, hay dos reglas importantes a tener en cuenta:

1. Escuchar antes de hablar. Si alguien está hablando, primero hay que esperar hasta que termine. En el contexto de las redes, a esto se le llama detección de portadora, donde un nodo escucha el canal antes de transmitir. Si una trama de otro nodo se está transmitiendo actualmente al canal, un nodo espera hasta no detectar transmisiones, esperara un corto periodo de tiempo y luego comenzará a transmitir.

2. Si alguien más comienza a hablar al mismo tiempo, deje de hablar. En el contexto de las redes, a esto se le denomina detección de colisiones, donde un nodo transmisor escucha el canal mientras está transmitiendo. Si el nodo detecta que otro nodo está transmitiendo una trama de interferencia, dejará de transmitir y esperará una cantidad aleatoria de tiempo antes de repetir el ciclo de detección y transmisión.

Estas son las dos reglas que se encuentran incorporadas en la familia de protocolos de acceso múltiple con escucha de portadora o CSMA (por sus siglas en inglés, del inglés *Carrier Sense Multiple Access*) y acceso múltiple con escucha de portadora con detección de colisiones o CSMA/CD (por sus siglas en inglés, del inglés *Carrier Sense Multiple Access with Collision Detection*) [40], [46], [47]. Se han propuesto muchas variaciones de CSMA y CSMA/CD, por lo que consideraremos algunas de las características más importantes y fundamentales de estos protocolos.

A pesar de que se realiza detección de portadoras, aún así se ocurren colisiones independiente de si un nodo se abstenga de transmitir siempre que detecte a otro nodo transmitiendo. Para comprender este escenario se puede ilustrar utilizando diagramas de espacio y tiempo. [48]. La Figura 3.8 grafica la ubicación de cuatro nodos (A, B, C y D) donde el eje horizontal representa la posición de cada nodo en el espacio y el eje vertical representa el tiempo.

En el momento $t0$, el nodo B detecta que el canal se encuentra inactivo, pues no hay ningún otro nodo transmitiendo. Bajo esta premisa, el nodo B comienza a transmitir, con sus bits propagándose en ambas direcciones a lo largo del medio de transmisión. La propagación de los bits de B, con el aumento del tiempo indica que se necesita una cantidad de tiempo distinta de cero para que los bits de B se propaguen (aunque la velocidad de propagación esté cerca de la velocidad de la luz) a lo largo del medio de transmisión. En el momento $t1$ ($t1 > t0$), el nodo D tiene una trama para enviar. Sin embargo, el nodo B está transmitiendo actualmente en

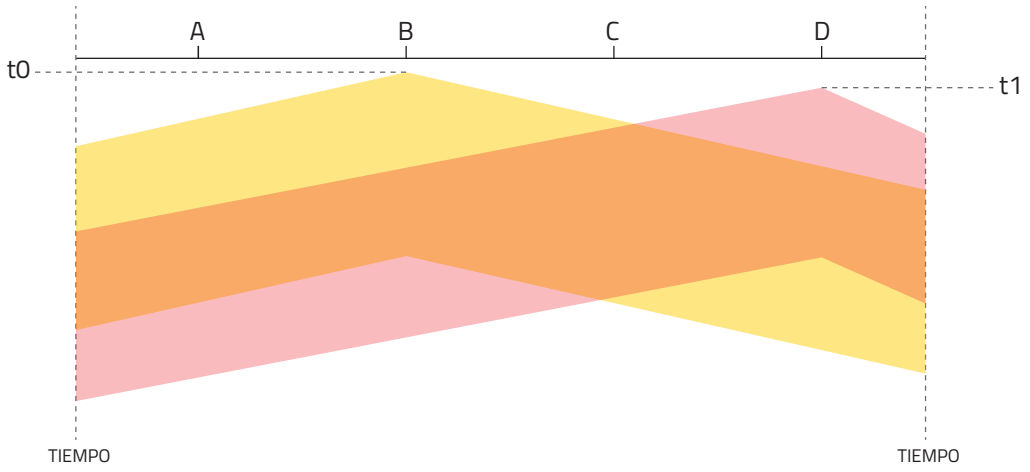

Figura 3.8. Diagrama de espacio y tiempo en CSMA

el tiempo t1, y los bits que transmite B todavía tienen que llegar a D y, por lo tanto, D detectará que el canal está inactivo en t1. De acuerdo con el protocolo CSMA, D comienza a transmitir su trama. Poco tiempo después, la transmisión de B comienza a interferir con la transmisión de D en D. Es evidente que el retardo de propagación de un canal de extremo a extremo de un canal de transmisión, el tiempo que tarde la señal en propagarse desde uno de los nodos a otro, juegue un papel crucial en la determinación de su desempeño, ya que cuanto más largo sea el retardo de propagación, mayor será la probabilidad de que un nodo de detección de portadora aún no pueda detectar una transmisión que ya ha comenzado en otro nodo de la red.

3.4.4. Acceso Múltiple con Escucha de Señal Portadora con detección de colisiones

En la Figura 3.8 los nodos no realizan detección de colisión, por lo que tanto B como D continúan transmitiendo sus tramas a pesar de la colisión. Cuando un nodo realiza una detección de colisión, cesa la transmisión inmediatamente. La Figura 3.9 muestra el mismo escenario que en la Figura 3.8, exceptuando que los dos nodos abortan cada uno su transmisión poco tiempo después de detectar una colisión. De esta manera, al agregar detección de colisión a un protocolo de acceso múltiple, ayudará notoriamente al rendimiento del protocolo en sí, ya que no trasmitirá una trama inútil o dañada (por interferencia con otra trama) en su totalidad.

Antes de analizar el protocolo CSMA/CD, resumamos los pasos de su funcionamiento desde la perspectiva de un adaptador de red (en un nodo) adjunto a un canal de difusión:

1. El adaptador obtiene un datagrama de la capa de red, prepara una trama de

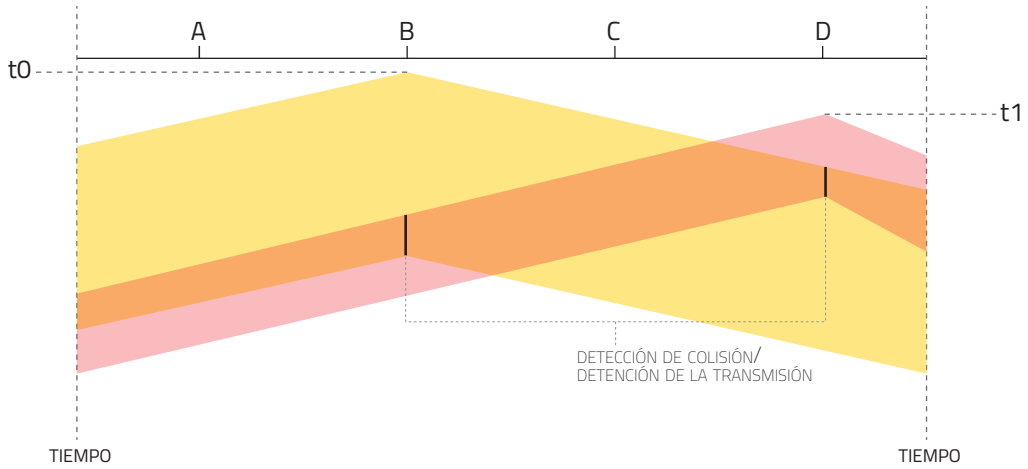

Figura 3.9. Diagrama de espacio y tiempo en CSMA/CD

la capa de enlace y coloca el búfer del adaptador de tramas.

2. Si el adaptador detecta inactividad en el canal, es decir, no hay energía de señal entrando al adaptador desde el canal, comenzará a transmitir la trama. En cambio, si el adaptador detecta actividad, esperará hasta que no detecte energía de señal y luego comenzará a transmitir su trama.

3. Mientras transmite, el adaptador monitorea la presencia de energía de señal proveniente de otros adaptadores que utilizan el canal de transmisión.

4. Si el adaptador transmite toda la trama, sin detectar la energía de la señal de otros adaptadores, el adaptador termina con la trama. En cambio, si el adaptador detecta energía de señal de otros adaptadores mientras transmite, dejará de transmitir su trama.

5. Después de dejar de transmitir su trama, el adaptador esperará un periodo de tiempo aleatorio y luego continuará con el paso número 2.

Es de esperar que se necesite una cantidad de tiempo aleatoria, en lugar de fija, para continuar con la transmisión, ya que si dos nodos transmiten tramas al mismo tiempo y luego ambos esperan la misma cantidad de tiempo fijo, seguirán colisionando para siempre. Pero ¿Cuánto debe ser el valor del intervalo para poder elegir el tiempo de espera aleatorio? Si el intervalo es grande y el número de nodos es pequeño, es probable que el tiempo de espera sea grande (con el canal inactivo) antes de repetir el ciclo de detección y transmisión. En cambio, si el intervalo es pequeño y el número de nodos en colisión es grande, es probable que los valores aleatorios elegidos sean casi los mismos y los nodos vuelvan a colisionar. Sería beneficioso que

un intervalo sea corto cuando el número de nodos en colisión sea pequeño y largo cuando el número de nodos en colisión sea grande.

El algoritmo de retroceso exponencial binario utilizado en Ethernet, así como en los protocolos de acceso múltiple de redes de cable DOCSIS resuelven dicha problemática. Al transmitir una trama que ya haya experimentado n colisiones, un nodo elige el valor de K al azar (0,1,2, ..., 2n – 1). Por ende, mientras más colisiones experimente una trama, mayor será el intervalo determinado por K. Para Ethernet, la cantidad real de tiempo que espera un nodo es de K*512 bits, es decir, K veces la cantidad de tiempo necesario para enviar 512 bits a Ethernet, y el valor máximo que puede tomar n está limitado a 10 [40].

Analicemos el siguiente ejemplo. Supongamos que un nodo transmitirá una trama por primera vez y que mientras lo hace, detecta una colisión. Luego, el nodo elige K=0 con una probabilidad de 0.5, o escoge K=1 con una probabilidad de 0.5. Si el nodo elige K=0, detecta el canal inmediatamente. Si el nodo elige K=1, esperará 512 bits (0.01 microsegundos para una red Ethernet de 100 Mbps) antes de comenzar el ciclo de detección y transmisión. Después de 10 o más colisiones, K se elige con la misma probabilidad de $\{0, 1, 2, ..., 1023\}$. De esta manera, el tamaño de los conjuntos por los cuales se elige K, crece exponencialmente con el número de colisiones, y por esta razón, este algoritmo se conoce como retroceso exponencial binario. También podemos observar que cada vez que un nodo prepara una nueva trama para la transmisión, ejecuta el algoritmo CSMA/CD, sin tener en cuenta las colisiones que puedan haber ocurrido en el pasado. Por lo tanto, es posible que un nodo con una nueva trama pueda colarse inmediatamente en una transmisión exitosa, mientras varios otros nodos están en retroceso.

3.4.5. Ethernet

Una red de área local o LAN (por sus siglas en inglés, *Local Area Network*)[40], es una red de computadores que conecta múltiples dispositivos dentro de un área limitada, el cual puede ser una residencia, escuela, laboratorio, universidad, entre otros.

La tecnología disponible en la actualidad para redes de área local es extensa, siendo posible escoger el tipo de cable, la topología de este, entre otras características. Ethernet y Wi-Fi son dos de las tecnologías más comunes.

Siendo inventada por Robert Metcalfe y David Boggs a mediados de la década de 1970 [49], en la actualidad Ethernet tiene casi todo el mercado de redes de área local cableadas, siendo entre la década de 1980 y 1990 el tiempo donde más enfrentó a otras tecnologías LAN como Token Ring, ATM y FDDI [40].

Las causas de su éxito se deben a su implementación temprana en comparación con otras tecnologías, por lo que los expertos de redes se familiarizaron en gran medida con esta red. También debido a que, tanto Token Ring como FDDI y ATM eran más costosas y complejas, generando una desconfianza al cambiar una red ya conocida y más barata. Y la última gran causa fue que pese a la mayor velocidad que

ofrecían las nuevas tecnologías, Ethernet siempre se fue actualizando y produciendo versiones que funcionan a velocidades iguales o mejores.

La primera red Ethernet inventada tenía las características esenciales de la red Ethernet actual, las cuales constan de un protocolo CSMA/CD para minimizar la probabilidad de colisión con retroceso exponencial binario. Era una LAN de difusión con topología de bus, es decir toda la información transmitida viaja y es procesada por todos los adaptadores conectados al bus. Luego la mayoría de las empresas y universidades cambiaron la topología por una en estrella basada en concentradores. Este tipo de conexión de Ethernet con topología en estrella es una LAN de transmisión. Es decir, cada vez que un concentrador recibía un bit de información en una de sus interfaces, este enviaba una copia a todas sus otras interfaces. El problema era que si un concentrador recibía tramas de dos interfaces diferentes al mismo tiempo ocurría una colisión, teniendo los nodos que retransmitir. Finalmente a principio de la década de los 2000, se mantuvo la topología, pero se basaron en una conexión mediante un conmutador, teniendo la ventaja de que un conmutador es sin colisiones y confiable.

3.5. Redes bajo ataques

La seguridad de la red es un tema importante en las redes de comunicación debido a la frecuencia y la cantidad de amenazas que se presentan en el día a día, así como también a los posibles ataques nuevos y más poderosos que se crean por ciberdelincuentes.

Una clase común que amenaza la seguridad de las redes se conoce como ataque de denegación de servicio o *DoS attack* (por sus siglas en inglés, *denial-of-service attack*) [50]. Este ataque consiste en que el perpetrador busca inutilizar la red o un recurso de red para los usuarios legítimos temporalmente o indefinidamente.

3.5.1. Tipos

Un ataque DoS puede realizarse de varias formas, las cuales básicamente consisten en las siguientes.

Ataque de vulnerabilidad

Consiste en enviar algunos mensajes a una aplicación o sistema operativo vulnerable de un host específico. Si la secuencia de paquetes es enviada correctamente, el servicio de la red puede detenerse o fallar.

Inundación de ancho de banda

El ciberdelincuente envía una cantidad extensa de paquetes al host objetivo con el propósito de congestionar el enlace de acceso, impidiendo la correcta llegada de paquetes legítimos a su destino.

Inundación de conexión

El ciberdelincuente envía una cantidad extensa de conexiones TCP semiabiertas o completamente abiertas al host objetivo. Este host puede inundarse de la gran cantidad de conexiones falsas a tal punto que deja de aceptar conexiones legítimas.

3.5.2. Ataque de denegación de servicio distribuido

Supongamos un ataque a un host objetivo donde todo el tráfico proviene de una única fuente. El enrutador puede fácilmente detectar el ataque y bloquear todo el tráfico de esa fuente antes de que el tráfico llegue al host objetivo. En este caso para vulnerar la seguridad del host objetivo el atacante envía el tráfico por medio de múltiples fuentes o esclavos, al que se le llama ataque de denegación de servicio distribuido o *DDoS attack* (por sus siglas en inglés, *Distributed DoS attack*) [51], ilustrado en la Figura 3.10. Este tipo de ataque es más difícil de detectar y es posible iniciar un ataque a un host objetivo, debido a que el tráfico que inunda a la víctima proviene de múltiples fuentes. También es complicado distinguir usuarios legítimos debido a que el ataque se distribuye de varios puntos de origen.

La cantidad de ataques DDoS aumenta año a año, siendo un problema real en la actualidad. Estos se aprovechan de un conjunto de computadores infectados y controlados por un atacante de forma remota, también llamados botnets. De esta forma, estos pueden conseguir más esclavos [50]. Algunos ejemplos comunes de ataques DDoS son la inundación de UDP, la inundación de SYN, entre otros [51]

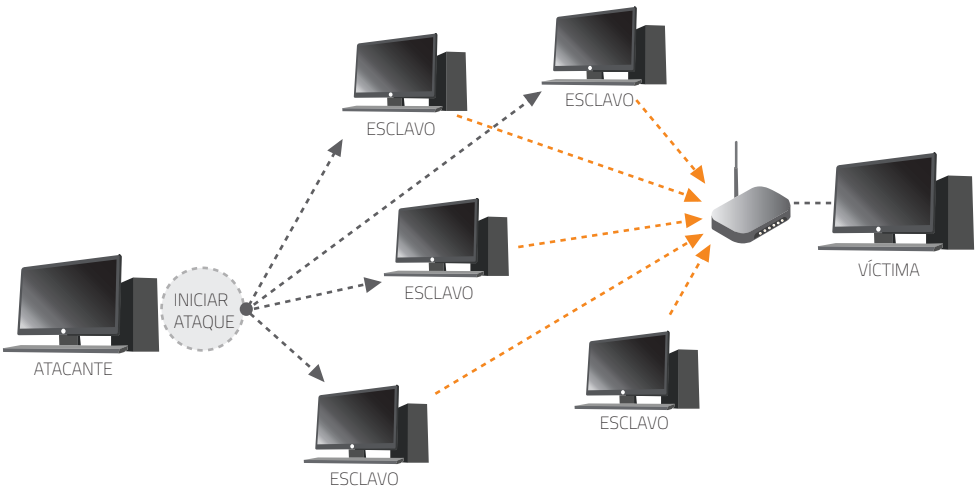

Figura 3.10. Esquema de ataque de denegación de servicio distribuido

3.6. Pérdida de paquetes

Los paquetes son la unidad de datos enviada y recibida en los sistemas de redes. La pérdida de paquetes ocurre cuando uno o más paquetes no llegan a su destino. Esta pérdida en aplicaciones en tiempo real como la transmisión de audio o vídeo y puede afectar en la calidad de la experiencia del usuario.

Uno de los factores para medir el rendimiento en un nodo es en términos de la probabilidad de pérdida de paquetes, que se obtiene como una relación entre los paquetes perdidos con respecto a los paquetes enviados.

Normalmente se espera que las redes experimenten pérdidas de paquetes, pero se debe tener en consideración que estos no afecten negativamente el rendimiento general del sistema. La pérdida de paquetes se debe a errores en la transmisión de datos, generalmente a través de redes inalámbricas, o la congestión de la red [40].

A pesar de ser un problema esperable en los sistemas de redes, se nombran las causas más comunes.

Red congestionada

Debido a la capacidad finita que presenta la cola de un nodo es posible presenciar una pérdida de paquetes, ya que si se le ofrece durante un tiempo prolongado una mayor cantidad de paquetes de lo que el segmento de red puede enviar, este debe descartar el paquete.

Existe la condición de colapso por congestión, que es cuando la congestión limita la comunicación útil de la red, y esta ocurre usualmente donde el tráfico entrante excede el ancho de banda saliente, como se ilustra en la Figura 3.11.

La tecnología de las redes en el presente tiene muchas técnicas de control y prevención de la congestión para tratar de evitar el colapso de congestión, los cuales incluyen CSMA/CA, CSMA/CD, reducir la velocidad de transferencia, o mediante el reenvío automático de los paquetes de datos perdidos.

Ataques en la red

Otra causa de la pérdida de paquetes pueden ser los ataques a la seguridad de la red. En la actualidad son muy comunes los ataques DoS y DDoS, existiendo el ataque de caída de paquetes o ataque de agujero negro [52], en el cual un ciberdelincuente accede al enrutador haciendo que los paquetes que supuestamente se retransmiten se descarten. Debido a que la pérdida de paquetes es común en las redes, es complicado identificar y evitar este tipo de ataques, debiendo tener una monitorización intensiva para ver comportamiento extraños en la red o pérdidas totales las cuales podrían ser una señal de este ataque.

El ataque de caída de paquetes es con más frecuencia implementado en redes inalámbricas debido a que estas tienen una arquitectura diferente a la de una red cableada.

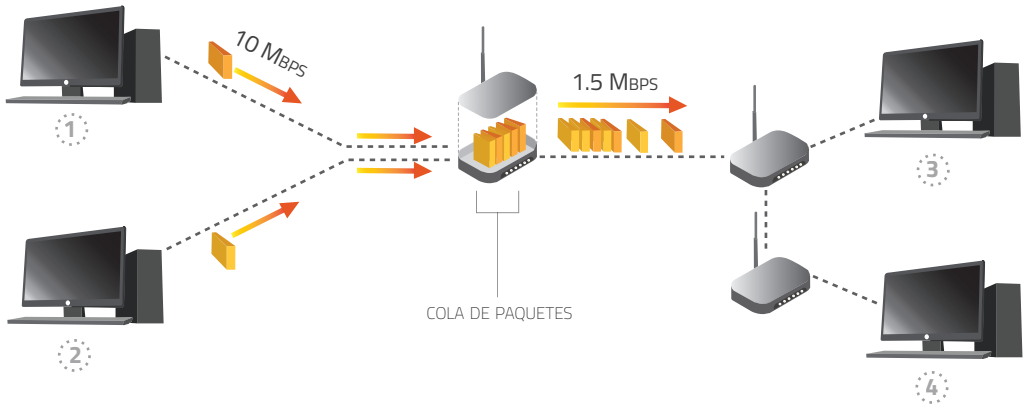

Figura 3.11. Esquema de congestión en una red de comunicación

Problemas de *hardware* y *software*

Los posibles problemas de *hardware* o *software* también podría afectar el rendimiento de la red provocando posibles pérdidas de paquetes. Por el lado del *software*, es posible que los programas que hagan que la red funcione de manera correcta presente errores a través del tiempo debido al desarrollo de estos, por lo que es recomendable mantener actualizado el *software* de los dispositivos para evitar errores en el rendimiento. Por otro lado, el uso de *hardware* defectuoso, obsoleto o desactualizado también puede afectar el rendimiento de la red, por lo tanto es aconsejable actualizar, mejorar y realizar un mantenimiento constante del *hardware* usado en la conexión del sistema para evitar la pérdida de paquetes.

3.7. Retardo de tiempo en paquete de datos

Un paquete comienza en un *host* (un nodo origen), este pasa por una serie de enrutadores y finaliza en otro *host* (el nodo destino). En este camino recorrido, el paquete padece varios tipos de retardos a través de cada nodo. Estos retardos son: retardo de proceso, retardo de cola, retardo de transmisión y retardo de propagación; donde el retardo total de la red es la suma de todos [40], [53].

Se puede observar en la Figura 3.12 que el camino recorrido por el paquete enviado por el host origen es: nodo origen luego el enrutador 1 y finalmente al enrutador 2. Se observa un enlace de salida que conecta el enrutador 1 con el enrutador 2, el cual está antecedido por una cola (o búfer, memoria interna). Cuando el paquete de datos llega al enrutador 1 desde el host, el primero examina el encabezado del paquete para establecer el enlace de salida correcto y posteriormente enviarlo por este. Los paquetes de datos sólo pueden ser transmitidos por un enlace si es que

no hay otro paquete transmitiéndose y si no hay otro paquete en la cola. En caso contrario, el paquete que llega se colocará en la cola.

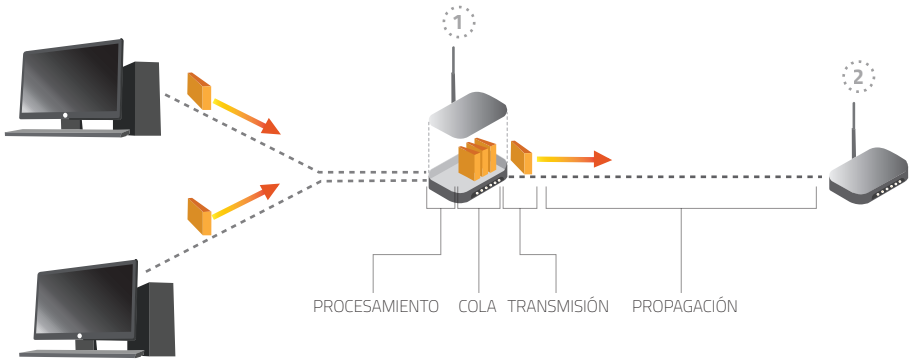

Figura 3.12. Esquema de retardos de tiempo

Retardo de procesamiento

El retardo de procesamiento es el tiempo necesario para analizar el encabezado de los paquetes de datos y establecer el campo de dirección destino. También puede incluir otros factores como el tiempo necesario para analizar los errores a nivel de bit que ocurrieron entre el nodo origen y el enrutador (en la Figura 3.12, corresponde al enrutador 1). Este retardo en los enrutadores de alta velocidad está en el orden de los microsegundos o menos.

Retardo de cola

Tiempo que el paquete espera en la cola para ser transmitido al enlace. Este retardo depende de cuántos paquetes están en cola, antes del paquete actual, esperando a ser transmitidos. Por lo tanto, si la cola está vacía y no se está transmitiendo ningún paquete, el retardo de cola será cero.

Retardo de transmisión

El retardo de transmisión es el tiempo transcurrido para transmitir todos los bits del paquete en el enlace. Este tiempo puede ser calculado denotando L la longitud del paquete en bits y denotando como R la velocidad de transmisión del enlace correspondiente en $bits/s$. Por ejemplo, para un enlace Ethernet de 10 Mbps, la velocidad es $R = 10$ Mbps; para un enlace Ethernet de 100 Mbps, la velocidad es $R = 100$ Mbps. Este retardo suele ser del orden de microsegundos a milisegundos.

Retardo de propagación

El retardo de propagación es el tiempo necesario para que las señales (los paquetes) se propaguen desde el comienzo del enlace al destino. El bit se propaga a la velocidad de propagación del enlace, el cual depende del medio físico de este (fibra óptica, cable de cobre de par trenzado, etc). Si se denota d como la distancia entre los dos enrutadores, s la velocidad de propagación del enlace, el retardo de propagación puede ser calculado como d/s.

Capítulo 4

SISTEMAS DE CONTROL SOBRE REDES DE COMUNICACIONES

En el Capítulo presente se consideran las restricciones generadas por la red mencionadas en el Capítulo 3, en específico la pérdida de paquetes y retardo en el tiempo de las señales. En esta sección, se describe como la red de comunicación será modelada en base a la literatura actual.

4.1. Particularidades de los sistemas de control sobre redes de comunicaciones

El avance tecnológico en las últimas décadas, especialmente en el manejo de la información, ha cambiado de gran manera el foco en la ingeniería de los sistemas de control. En la actualidad, se tiene una mayor presencia de la computación y redes, lo cual ha resultado en su integración en distintos niveles de operaciones de máquinas y procesos de información. Esto ha motivado el estudio del área de sistemas de control sobre redes o NCS, (por sus siglas en inglés, *Networked Control Systems*) [54]. Debido a que los dispositivos de hardware para redes hoy en día son más accesibles en términos generales, es que los sistemas de control se comunican con sensores, actuadores y controladores a través de una red de comunicación como se observa en la Figura 4.1. Esto brinda ventajas en comparación con los sistemas de control tradicionales, tales como diseños a mayor escala, bajos costos de instalación, mayor flexibilidad de reestructuración y un mejor mantenimiento.

A pesar de las ventajas, la integración de la red de comunicación a sistemas de control cambia el análisis y diseño tradicional de estos, ya que en general se asume que las señales se intercambian sin incertidumbres. Como se mencionó en el Capítulo 1, los canales de comunicaciones tienen entre sus restricciones más importantes: i) limitaciones del ancho de banda, ii) muestreo de las señales, iii) retardo de procesamiento y transmisión en las señales debido a la red y iv) pérdida de información. Estas incertidumbres pueden afectar de manera significativa el rendimiento

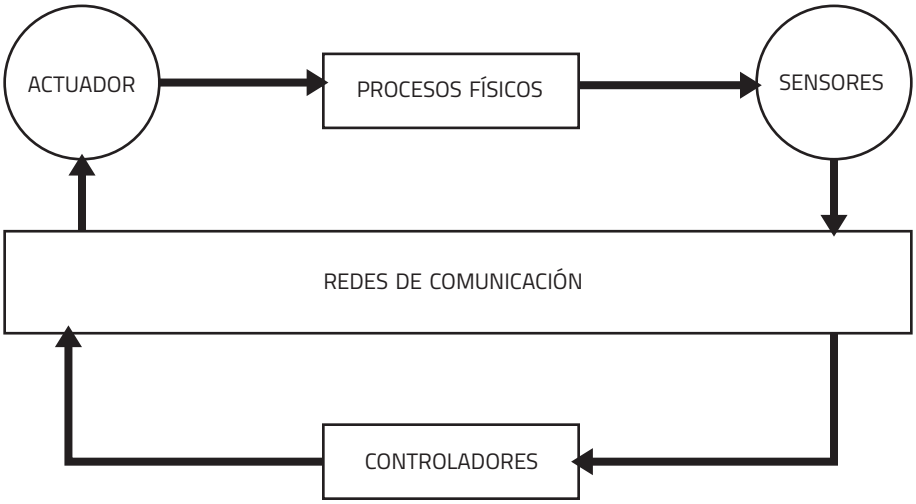

Figura 4.1. Diagrama de bloques de un sistema de control sobre redes de comunicaciones

del control y filtrado. Por ejemplo, existen tasas de pérdidas de paquetes críticas por encima de la cual las matrices de covarianza del error de estimación de estado del filtro de Kalman divergirá [6], [55]. Además existe una tasa de datos positiva crítica por debajo de la cual no existe ningún esquema de cuantificación y control capaz de estabilizar una planta inestable [56], [57].

Si pensamos en los sockets TCP, excluyendo el protocolo UDP en un socket TCP/IP general, estos realizan una transmisión de paquetes de datos altamente confiable, independientemente de las posibles colisiones que puedan ocurrir en el medio de transmisión física, debido al aviso y retransmisión que realizan. Por lo tanto, en una red con poca carga, TCP dará como resultado un retardo de paquetes, pero no la pérdida de paquetes. Aún así, debido a la posibilidad de que los datos sean transmitidos por protocolo UDP, o que estén sujetos a ataques de denegación de servicio o *DoS*, específicamente un ataque de caída de paquetes, en la presente Tesis se considera y estudia la pérdida de paquetes en el envío de datos. Entonces, debido a la diversidad de restricciones que pueden presentarse en las redes de comunicaciones, el modelado de la red puede ser muy variado. En particular, en este Capítulo se realiza un estudio de la literatura disponible en el área de sistemas de control sobre redes de comunicaciones para modelar el problema propuesto en la Tesis, que consta en un sistema el cual presenta retardo y pérdida de paquetes.

4.2. Componentes de un sistema de control sobre redes de comunicaciones

Los sistemas de control sobre redes de comunicación tiene componentes los cuales hacen que se deba analizar y replantear temáticas de la teoría clásica de control automático. Estos son los siguientes componentes:

Red de capacidad limitada

Existen diversas redes de comunicación actuales que pueden ser integradas en sistemas de control, entre las que destacan Ethernet, FireWire, Fieldbuses, DeviceNet, redes inalámbricas como Bluetooth o IEEE 802.11, entre otras. Cada una de estas redes tiene su propio protocolo, el cual está planeado para aplicaciones específicas según el requerimiento del usuario. El rendimiento del sistema de control sobre red depende del rendimiento de esta red, ya que cada una tiene su velocidad de transmisión, retardos de tiempo, pérdida de paquetes, entre otras características. Como se mencionó anteriormente, estas características no integradas en el estudio del control clásico requiere ser examinado y estudiado.

Sensores

Estos miden las diversas señales físicas de un proceso, y pueden ser sensores electrónicos, químicos, ópticos, entre otros. Estos componentes son los artefactos más sencillos de la red, y dado que el número suele ser mayor al número de actuadores, es deseable que sean de bajo costo. Estos presentan limitaciones en la detección, computación y comunicación, por lo que también se deben tener en consideración en el análisis y diseño.

Controladores

En los sistemas de control sobre redes, los sensores transmiten los datos al controlador en forma de paquetes. El controlador de la misma forma, envía la señal de salida a la planta en forma de trama o paquete a través de una red de comunicación. Por lo mismo, la realimentación puede contener información limitada debido a la interacción que se sufre entre lo físico y lo cibernético.

4.3. Modelado de sistemas de control sobre redes

En la actualidad existe una gran variedad de artículos del área de Control sobre redes, base desde la cual nos basaremos para construir nuestro problema de interés.

Se podrá ver que la mayoría del trabajo en estimación en el área de sistemas de control sobre redes se ha desarrollado enfocado a estimación de estados, principalmente a estrategias de control.

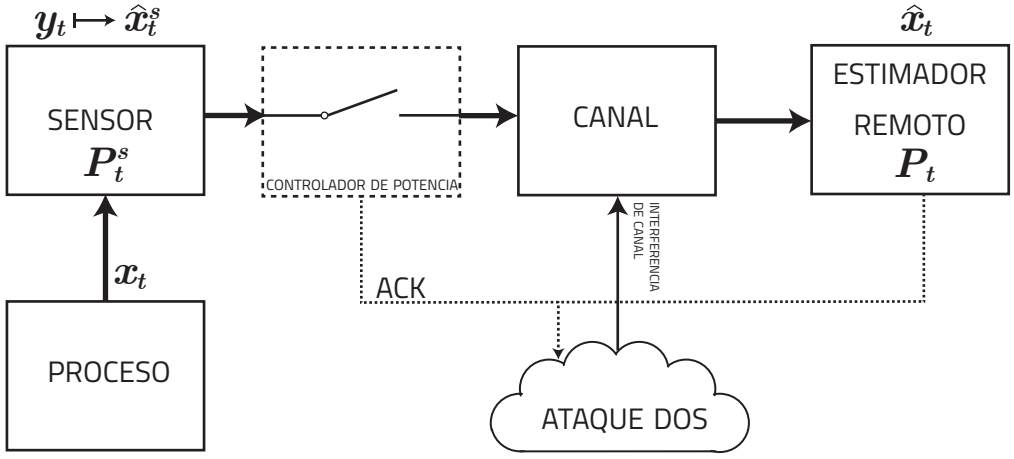

Figura 4.2. Modelo propuesto en [58], [59]

4.3.1. Ataques DoS en la estimación de estado remoto con información asimétrica

Como se mencionó anteriormente, las redes de comunicación traen variados beneficios a los sistemas físicos, pero a su vez introducen diferentes restricciones. Debido a esto, D. Quevedo *et. al* [58], [59] considera la estimación de estados en un entorno adversario. Considera un sensor el cual envía estimaciones del estado local a un estimador remoto a través de un canal de comunicación, el cual esta vulnerable a congestiones debido a ataques inteligentes de denegación de servicio, como se observa en la Figura 4.2.

En este se menciona que en la práctica, el sensor debe tomar estrategia de transmisión para evitar ataques de interferencia, mientras que el atacante intentará reconocer estas acciones y modificar su patrón de ataque en consecuencia. En el artículo se considera que el estimador remoto informará al sensor de la información de pérdida de paquetes mediante el envío de una trama de reconocimiento corto (ACK) inmediatamente, y por otro lado, el atacante no puede acceder a los ACK. Esto se explica debido a que la comunicación de igual a igual utiliza cifrados conmutativos para respaldar una validación segura de los ACK. Por lo tanto, el sensor conoce bien el estado del paquete de datos, mientras que el atacante, sin el conocimiento del ACK, solo tiene información parcial. Así, debido a que la información de reconocimiento del estimador remoto al sensor está oculta al atacante, se tiene información asimétrica entre el sensor y el atacante, modelando esto mediante un juego estocástico bayesiano.

Para el modelado de la Figura 4.2, se tiene el siguiente sistema lineal invariante en el tiempo:

$$x_{t+1} = Ax_t + w_k \tag{4.3.1}$$
$$y_k = Cx_k + v_k \tag{4.3.2}$$

donde x_t es el vector de estado del sistema en el tiempo t, y_t es la medición de salida obtenida por el sensor, w_t y v_t representan ruido blanco aleatorio Gaussiano de media cero.

Así, debido al avance de los sensores inteligentes, y ya que estos están equipados con memoria y sistemas integrados en chips, se les permiten consultar información histórica y ejecutar algunos algoritmos recursivos simples en los datos recopilados. Con capacidades de almacenamiento y computación, el sensor de la Figura 4.2 puede procesar las mediciones recopiladas y_0^t ejecutando un filtro de Kalman, en lugar de transmitirlas directamente, y luego estimar el estado del proceso x_t localmente, denotado por $\hat{x}_t^s$.

Para el canal de comunicación, se considera que la energía para la detección, el cálculo y la transmisión está restringida para los nodos de sensores. Por tanto, como se muestra en la Figura 4.2, se requiere que el sensor decida el nivel de energía de transmisión al cual enviar las estimaciones obtenidas $\hat{x}_t^s$ al estimador remoto. Al mismo tiempo, al emitir una señal para interferir con el canal, el atacante es capaz de sabotear la entrega de $\hat{x}_t^s$ y así degradar la calidad de la estimación. Al igual que el sensor, el atacante tiene un presupuesto de energía limitado y debe determinar la energía de interferencia en cada momento. Además, se asume que el canal entre el sensor y el estimador es sin memoria, y tiene ruidos blancos aditivos gaussianos. Y el paquete de datos transmitidos puede llegar al estimador remoto con errores desconocidos debido al ruido del canal, desvanecimiento de la señal, efectos multitrayecto, etc. Se introduce la tasa de error de paquete o PER (por sus siglas en inglés, Packet-Error-Rate), para medir las pérdidas de paquete. Bajo este escenario, la pérdida de paquetes bajo ataques DoS, la llegada del paquete se puede caracterizar por un proceso aleatorio binario, denotado por η_t. Donde $\eta_t = 0$ significa la pérdida de paquetes y $\eta_t = 1$ en caso contrario. Entonces, la probabilidad de llegada de paquetes se define de la siguiente manera:

$$p(\eta_t = 1) = 1 - PER_t. \tag{4.3.3}$$

En el artículo se considera un mecanismo de comunicación-retroalimentación entre el estimador y el sensor, como se muestra en la Figura 4.2. El estimador remoto informará al sensor de la información de pérdida de paquetes η_t mediante el envío de una trama ACK corta inmediatamente, es decir, antes del instante $t + 1$. Este mecanismo es una parte esencial de los protocolos de Internet (por ejemplo, el protocolo TCP/IP). Se supone que el ACK es recibido de forma fiable por el sensor, a diferencia del atacante, el cual no tiene acceso a los ACK. Y se tiene que, tanto el sensor como el atacante pueden monitorear la potencia de transmisión y la potencia de interferencia en cada momento después de la transmisión del paquete.

Basado en lo escrito anteriormente, se tiene que el conjunto de información para la toma de decisiones del atacante es diferente al del sensor. La distinción entre ellos surge de la disponibilidad de paquetes ACK, es decir, la estructura de información para el sensor es similar a TCP, mientras que para el atacante, es similar a UDP.

Para la solución, se tiene que debido a la limitación de potencia de transmisión y congestión, los diseños de estrategia del sensor y el atacante están acoplados. El objetivo del sensor es asegurarse de que el estimador remoto esté suficientemente informado del proceso, sin desperdiciar energía; y por otro lado, el atacante tiene la intención de interrumpir la comunicación confiable entre el sensor y el usuario, también sin gastar más energía de la requerida. Con objetivos opuestos, los procedimientos de toma de decisiones del sensor y el atacante están vinculados de forma interactiva. Así, se propone un algoritmo basado en el aprendizaje por refuerzo de agentes múltiples para encontrar tales estrategias.

4.3.2. Estimación óptima considerando estrategias de compensación de pérdida de datos

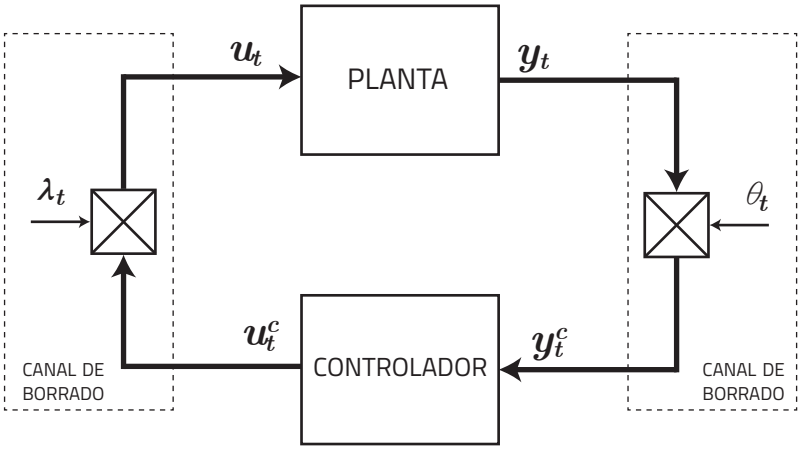

Figura 4.3. Sistema de control sobre redes sujeto a pérdida de datos

En los artículo de F. Vargas *et. al* [60], [61] se estudia sobre estimación óptima en un sistema de control sobre redes, específicamente donde los canales de comunicación son canales con pérdidas como se observa en la Figura 4.3. Para solucionar el problema de pérdidas, el controlador incorpora un compensador de pérdida de datos que reemplaza las mediciones faltantes con estimaciones óptimas. Específicamente, toda la secuencia de señales de control se transmite y almacena en una memoria intermedia en el lado del actuador como se observa en la Figura 4.4 . Entonces, si ocurre una interrupción, la señal de control perdida puede ser reemplazada por la predicción

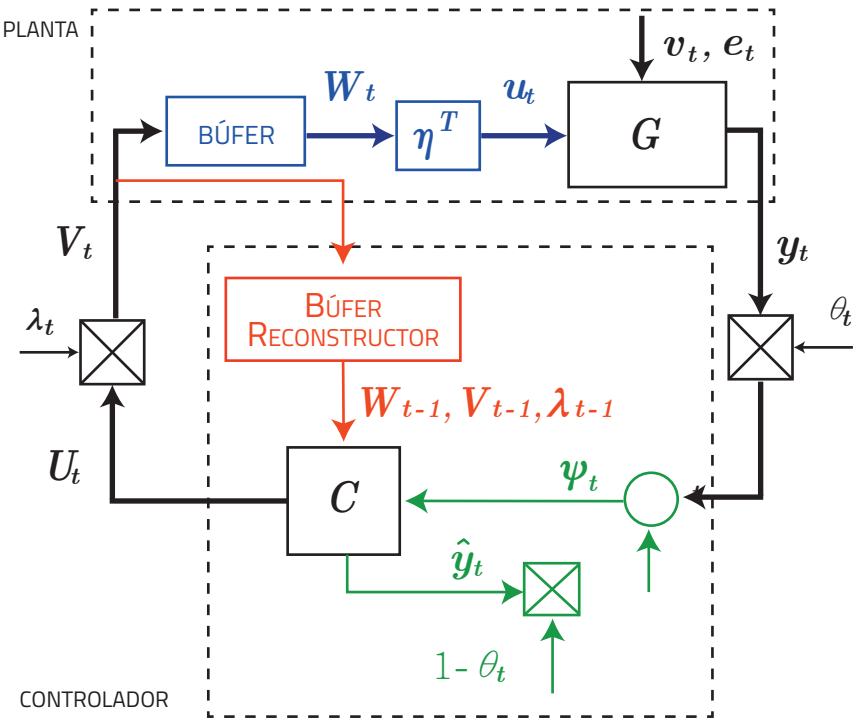

Figura 4.4. Estrategia propuesta en [60], [61] de compensación de pérdida de datos

almacenada en la memoria intermedia del actuador. El búfer se sobrescribe cada vez que llega una nueva secuencia de control. Esta estrategia de compensación se conoce como control predictivo en paquetes (también conocido como control predictivo en red).

Para esto se considera una planta de múltiples entradas y múltiples salidas o MIMO (por sus siglas en inglés, *Multiple-input Multiple-output*), de tiempo discreto, lineal, variante en el tiempo, descrita por:

$$x_{t+1} = A_t x_t + B_t u_t + v_t, \tag{4.3.4}$$

$$x_t = C_t x_t + e_t \tag{4.3.5}$$

donde x_t es el estado, u_t es la entrada del actuador, y_t es la salida del sensor, v_t y e_t son ruidos gausianos independientes de media cero. A_t, B_t y C_t son matrices de dimensiones apropiadas.

Como se mencionó, la planta se comunica con el controlador a través de dos canales con pérdidas independientes. Para modelar la pérdida de datos, se adoptó la descripción del canal de borrado, en la que la entrada del canal se multiplica por un

proceso de Bernoulli que toma valores 0 o 1. Entonces, si en el momento t el proceso de Bernoulli es cero, todo el vector de datos se pierde; de lo contrario, los datos se reciben correctamente. En la Figura 4.3 se muestra el sistema de forma general, donde λ y θ son los procesos de Bernoulli relacionados con la pérdida de paquetes de control y la pérdida de paquetes de medición, respectivamente.

Para afrontar la pérdida de datos, se usó la arquitectura de control de la Figura 4.4, donde el bloque denotado por C concentra las capacidades de procesamiento del controlador.

La arquitectura incluye un par de elementos, los cuales son:

Compensación por pérdidas de medición

Se considera que el controlador incorpora un compensador de pérdida de datos descrito por:

$$\psi_t = \theta_t y_t + (1 - \theta_t)\hat{y}_t, \tag{4.3.6}$$

donde ψ_t es la señal vectorial que realmente llega al controlador, y $\hat{y}_t$ es una estimación de los datos faltantes. Por lo tanto, si y_t se pierde, entonces $\theta_t = 0$, y por lo tanto $\psi_t = \hat{y}_t$, reemplazando los datos faltante. Por otro lado, si $\theta_t = 1$ entonces $\psi_t = y_t$. Esta compensación requiere conocer, en el lado del controlador, el estado instantáneo de θ_t, que de hecho se mantiene desde que el controlador reconoce si se recibió una medición o no.

Compensación por pérdidas de medición

La estrategia adoptada para hacer frente a la pérdida de la señal de control es tal que el controlador envía a la planta no solo la señal de control actual, sino también un conjunto de señales de control que se utilizarán para reemplazar los datos faltantes. Todo el conjunto de señales de control se condensa en un paquete de datos U_t, dado por

$$U_t = \begin{bmatrix} u_{t|t}^T & u_{t+1|t}^T & \cdots & u_{t+N-1|t}^T \end{bmatrix}^T \tag{4.3.7}$$

donde $u_{m|t}^t$ con $m > t$, representa el vector de control que se aplicará en el momento m, pero calculado utilizando la información disponible en el momento k (una predicción). El número de vectores de control $N \leqslant 1$ determina el tamaño del paquete y, por tanto, el horizonte de predicción. Esta estrategia predictiva empaquetada requiere almacenar en un búfer, el conjunto de entradas de control que se aplicarán en caso de una interrupción, como se observa en la Figura 4.4. La dinámica del búfer se describe mediante un vector W_t de modo que su primer elemento es siempre el vector de control u_t aplicado en el actuador. Así, definiendo

$$\eta = \begin{bmatrix} I_{n_u} & 0 & \cdots & 0 \end{bmatrix}^T, \tag{4.3.8}$$

se tiene que el vector de control que llega al actuador está dado por

$$u_t = \eta^T W_k, \tag{4.3.9}$$

donde W_t corresponde a los datos almacenados en el búfer en el instante de tiempo t. Dichos datos almacenados se describen mediante

$$W_t = \lambda_t U_t + (1 - \lambda_t) L W_{t-1}, \tag{4.3.10}$$

donde L es una matriz triangular superior. Por lo tanto, si $\lambda_t = 0$, el búfer se desplaza y el segundo vector de control en W_{t-1} se convierte en el primer vector de W_t y, por lo tanto, se aplica al actuador. Si $\lambda_t = 1$, la nueva secuencia de control U_t actualiza el búfer.

Reconstrucción del búfer

En los trabajos realizados por F. Vargas [60], [61], se supone el uso de un protocolo similar a TCP, es decir, se envía un acuse de recibo retardado de un paso desde el lado del actuador al controlador, acusando si se recibió o no la señal de control anterior, lo que equivale a conocer en el lado del controlador, y en el tiempo t, la realización de λ_{t-1}. Esto permite reconstruir V_{t-1} y el estado del búfer W_{t-1} en el tiempo $t-1$, asumiendo que W_{k-2} se conoce. Esto se muestra en la Figura 4.4 con un bloque en el lado del controlador dedicado a proporcionar esta información (una muestra retrasada). Por lo tanto, se considera que ambas secuencias V^{t-1} y W^{k-1} son conocidas en el controlador en el tiempo t (incluido el estado inicial del búfer).

Planteamiento del problema

Para buscar la solución al problema, se combina la dinámica relacionada con la planta y el búfer en un estado auxiliar

$$\xi_{t+1} = \begin{bmatrix} x_t^T & W_{t-1}^T \end{bmatrix}^T. \tag{4.3.11}$$

Entonces, la descripción del espacio de estado respectivo del sistema auxiliar correspondiente viene dada por

$$\xi_{t+1} = \check{A}_t(\lambda_t)\xi_t + \check{B}_t(\lambda_t)U_t + \check{v}_t, \tag{4.3.12}$$

$$y_t = \check{C}_t \xi_t + e_t, \tag{4.3.13}$$

donde

$$\check{A}_t(\lambda_t) = \begin{bmatrix} A_t & B_t\eta^T \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & I(1-\lambda_t) \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & L \end{bmatrix}, \tag{4.3.14}$$

$$\check{B}_t(\lambda_t) = \begin{bmatrix} B_t\eta^T \\ I \end{bmatrix} \lambda_t, \tag{4.3.15}$$

$$\check{C}_t = \begin{bmatrix} C_t & 0 \end{bmatrix}, \tag{4.3.16}$$

$$\check{v}_t = \begin{bmatrix} v_t \\ 0 \end{bmatrix}. \tag{4.3.17}$$

Así, el objetivo fue estimar el estado auxiliar de la planta ξ_t y la salida de la planta y_t, dados los datos disponibles, que se definen mediante dos conjuntos de información, $\mathcal{I}_1^t = (\psi^t, V^t, W^t, \lambda^t)$ e $\mathcal{I}_2^t = (\psi^t, V^{t-1}, W^{t-1}, \lambda^{t-1})$, siendo $\mathcal{I}_2^t = (\psi^t, \mathcal{I}_1^{t-1})$. En particular, el interés es determinar las estimaciones óptimas:

$$\xi_{t|t} = \mathcal{C}_t^{\xi,\text{opt}}(\mathcal{I}_2^t), \tag{4.3.18}$$

$$\xi_{t+1|t} = \mathcal{C}_{t+1}^{\xi,\text{opt}}(\mathcal{I}_2^t), \tag{4.3.19}$$

$$\hat{y}_{t|t-1} = \mathcal{C}_t^{y,\text{opt}}(\mathcal{I}_1^{t-1}), \tag{4.3.20}$$

donde $\mathcal{C}_t^{\xi,\text{opt}}$, $\mathcal{C}_{t+1}^{\xi,\text{opt}}$ y $\mathcal{C}_t^{y,\text{opt}}$ son las tranformaciones afines óptimas que minimizan el error de estimación cuadrática media de ξ_t, ξ_{t+1} y y_t, respectivamente, dados los conjuntos de información correspondientes.

4.4. Modelado del problema de interés

Se puede observar que lo mostrado previamente se realiza pensando en estrategias de control. Dado lo anterior, el trabajo realizado en la tesis es un estudio del efecto de la pérdida de paquetes y del retardo de tiemmpo, específicamente en la identificación de sistemas, por lo tanto es diferente a este tipo de trabajos. Pero basado en los trabajos de sistemas de control sobre redes podremos modelas nuestro problema de interés.

4.4.1. Sistemas con pérdida de paquetes

Como se mencionó anteriormente, una de las incertidumbres a tener en consideración en los canales de comunicación es la pérdida de paquetes. Esto puede ocurrir debido a la naturaleza poco confiable de algunos enlaces, a la congestión o el desvanecimiento inducido por trayectos múltiples en las redes inalámbricas. Estos fenómenos, tanto en sistemas de control sobre redes como en identificación de sistemas, pueden tener un efecto de deterioro en el desempeño del controlador o estimador.

Los sistemas bajo pérdidas de paquetes tienen un amplio estudio en la literatura de NCS [10], [55], [62], [63], en donde este proceso se modela comúnmente como un proceso independiente e idénticamente distribuido o una cadena de Markov. En el modelado de pérdida de paquetes como cadena de Markov se tienen dos casos: paquetes descartados o los enviados correctamente [54].

El proceso de Markov es más general y realista cuando se trata de modelar la pérdida de paquetes, ya que considera la correlación temporal en las condiciones de red [54], [62]. Sin embargo, la problemática expuesta en la presente tesis se considerará la pérdida de paquetes como un proceso Bernoulli, con el fin de realizar un estudio estadístico desde el caso más simple.

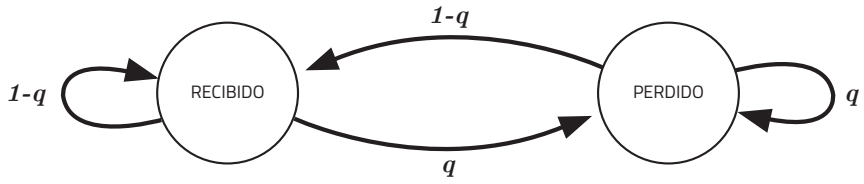

Figura 4.5. Modelo Bernoulli para pérdida de paquetes

De esta forma, la recepción o pérdida está representada por una variable aleatoria binaria λ_t, donde $\lambda_t = 1$ indica que el paquete que contiene la información se ha entregado con éxito, mientras que $\lambda_t = 0$ corresponde a la pérdida del paquete. Se supone que el proceso de pérdida de paquetes $\{\lambda_t\}_{t\geqslant 0}$ es un proceso Bernoulli i.i.d. con distribución de probabilidad:

$$\mathrm{p}\left(\lambda_t = 1\right) = 1 - q = 1 - \mathrm{p}\left(\lambda_t = 0\right), \tag{4.4.1}$$

donde $q \in (0, 1)$ es la tasa de pérdida de paquetes. El modelo de Bernoulli tiene el comportamiento representado en la Figura 4.5, donde observamos que no existe correlación entre un estado u otro.

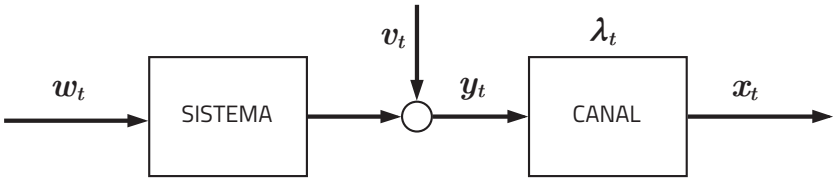

Figura 4.6. Configuración de la red

Entonces el modelado en general se observa en la figura 4.6, el cual es un sistema que tiene como señal de salida $\mathbf{y}_t$, la cual es transmitida por un canal de comunicación que induce posibles pérdidas de paquetes resultando en la señal $\mathbf{x}_t$.

Esto se puede escribir como:

$$\mathbf{x}_t = \begin{cases} \mathbf{y}_t & \text{si } \lambda_t = 1 \\ \text{Dato perdido} & \text{si } \lambda_t = 0. \end{cases} \tag{4.4.2}$$

4.4.2. Sistemas con retardo de tiempo

Otro problema común, inducido por las redes de comunicaciones, son los retardos de tiempos, debido a las limitaciones tanto de *hardware* como *software*. Esto puede ocasionar un mal desempeño del rendimiento en el sistema de control.

Este problema tiene un amplio estudio en la literatura de NCS [64], [65], [66] y sigue siendo importante hasta en redes actuales como las redes inalámbricas 5G [67].

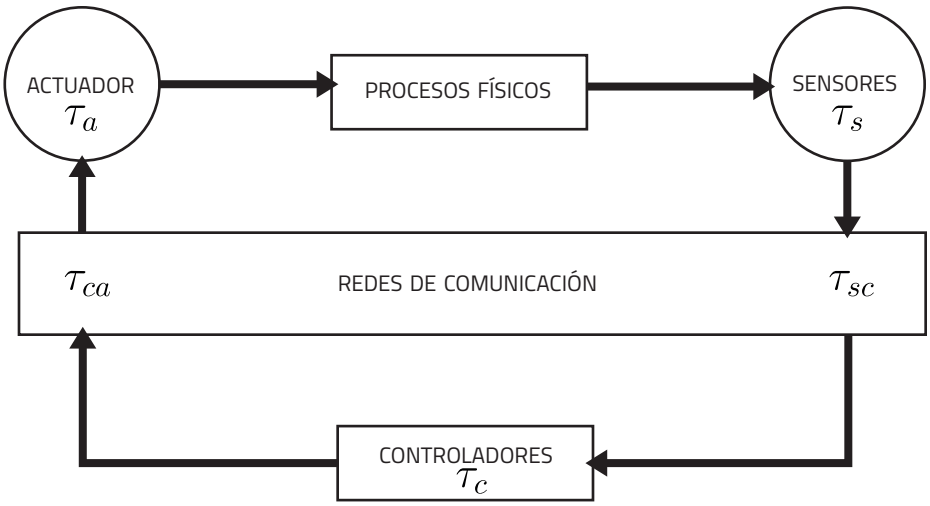

Figura 4.7. Diagrama en bloques de sistemas en red bajo retardos de tiempo

Este puede ser modelado como un retardo de tiempo limitado y constante, aleatorio e ilimitado, entre otras formas [4].

Para efectos de nuestra problemática en la presente tesis, consideraremos un modelo similar al expuesto en [64], [67], el cual es presentado en la Figura 4.7. Así se integra el retardo provocado por la red tanto para el problema de sistema de control como para el algoritmo de identificación. Así, la suma de los retardos generados, tanto por la red de comunicación como por los componentes físicos del sistema, se consideran una constante τ. Es decir $\tau = \tau_a + \tau_s + \tau_{ca} + \tau_{sc} + \tau_c$ (ver Figura 4.7).

Capítulo 5

ESTIMACIÓN DE PARÁMETROS DE UN SISTEMA SUJETO A PÉRDIDA DE PAQUETES Y RETARDO DE TIEMPO

En base a lo estudiado en los capítulos anteriores, nace la interrogante de cómo estimar un sistema dinámico sujeto a redes de comunicación y la manera en cómo se deben considerar las restricciones causadas por dicho escenario.

En este capítulo expondremos la problemática principal, con la finalidad de proponer una solución basada en estimación por ML a través del algoritmo EM junto al criterio de información de Akaike para estimar el retardo del sistema.

5.1. Planteamiento del problema

Consideramos el problema de estimación conjunta de parámetros de un sistema dinámico sujeto a un canal de comunicación como se observa en la Figura 5.1. El sistema dinámico incluida la red es modelado como *Output-Error* [1] y el canal de comunicación genera un retardo desconocido τ y una posible pérdida de paquetes. Además se asumen interferencias en este, modeladas como ruido blanco Gaussiano

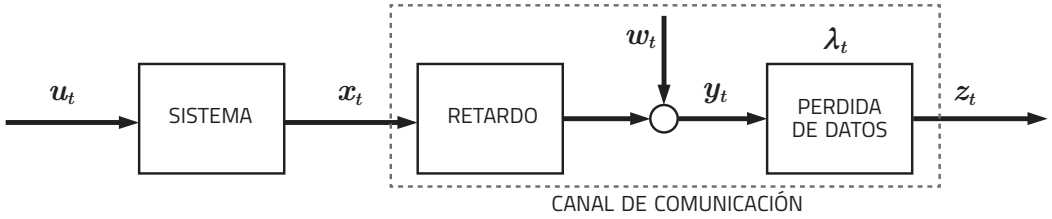

Figura 5.1. Configuración de la red propuesta en la Tesis

aditivo. El problema está dado por:

$$\mathbf{x}_t = \frac{B(q^{-1},\theta)}{F(q^{-1},\theta)}\mathbf{u}_t, \tag{5.1.1}$$

$$\mathbf{y}_t = \mathbf{x}_{t-\tau} + \mathbf{w}_t, \tag{5.1.2}$$

y los polinomios son:

$$B(q^{-1},\theta) = b_1 q^{-1} + b_2 q^{-2} + \cdots + b_{n_b} q^{-n_b}, \tag{5.1.3}$$

$$F(q^{-1},\theta) = 1 + f_1 q^{-1} + f_2 q^{-2} + \cdots + f_{n_f} q^{-n_f}, \tag{5.1.4}$$

donde q^{-1} es el operador retardo unitario (i.e. $q^{-1}\mathbf{x}_t = \mathbf{x}_{t-1}$), $\mathbf{u}_t$ es la entrada del sistema en , $\mathbf{x}_t$ es la salida del sistema sin ruido y $\mathbf{w}_t$ es el ruido blanco Gaussiano aditivo. Para modelar la pérdida de paquetes en el sistema, se considera que la red de comunicación transmite un paquete de datos $\tilde{\mathbf{y}}_\mathbf{p}$ de largo L a un receptor, el cual se define como:

$$\tilde{\mathbf{y}}_\mathbf{p} = \begin{bmatrix} \mathbf{y}_{L(\mathbf{p}-1)+1} \\ \vdots \\ \mathbf{y}_{L\cdot\mathbf{p}} \end{bmatrix}, \tag{5.1.5}$$

donde $\mathbf{p} > 0$ es el número del paquete de datos correspondiente. Que este sea recibido correctamente, esta dado por una variable aleatoria de distribución Bernoulli $\lambda_\mathbf{p}$, tal que $p(\lambda_\mathbf{p} = 1) = 1 - q$ es la probabilidad de recibir el paquete, y $p(\lambda_\mathbf{p} = 0) = q$ es el caso contrario, por lo tanto:

$$\mathbf{z}_t = \begin{cases} \mathbf{y}_t & \text{si } \lambda_\mathbf{p} = 1 \\ \text{Dato perdido} & \text{si } \lambda_\mathbf{p} = 0. \end{cases} \tag{5.1.6}$$

Respecto al ruido aditivo en (5.1.1), asumimos que w_t es ruido blanco Gaussiano con varianza σ^2. Además, asumimos que $\mathbf{u}_t = \mathbf{x}_t = 0$ y $\mathbf{w}_t = 0$ para $t \leqslant 0$. Por otro lado, el retardo es considerado como un corrimiento en el polinomio $B(q^{-1},\theta)$:

$$\hat{B}(q^{-1},\theta) = q^{-\tau} B(q^{-1},\theta) \tag{5.1.7}$$

$$= b_1 q^{-1-\tau} + b_2 q^{-2-\tau} + \cdots + b_{n_b} q^{-n_b-\tau}, \tag{5.1.8}$$

Así, el vector de parámetros que define el sistema dinámico en (5.1.1) está dado por:

$$\beta^T = \begin{bmatrix} \theta^T & \sigma^2 \end{bmatrix}^T, \tag{5.1.9}$$

donde:

$$\theta^T = \begin{bmatrix} b_1 & \cdots & b_{n_b} & f_1 & \cdots & f_{n_f} \end{bmatrix}. \tag{5.1.10}$$

Además, asumimos que existe un vector de parámetros $\beta = \beta_0$ que define el sistema "verdadero".

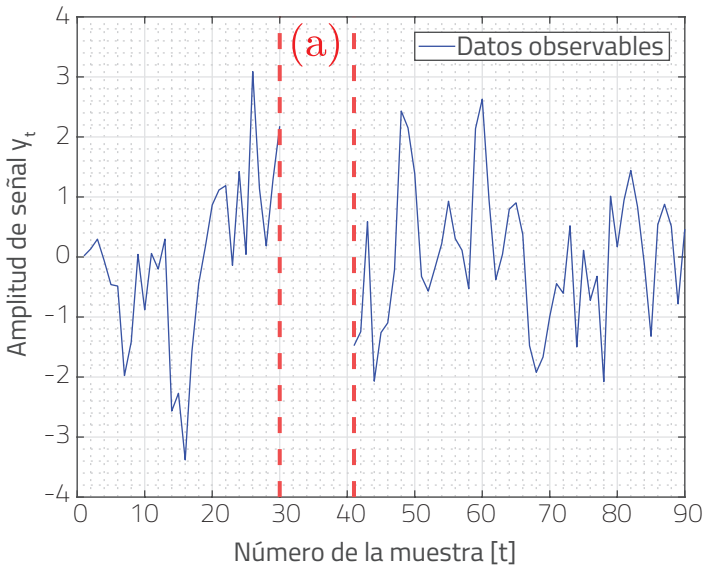

Figura 5.2. Señal con probabilidad de pérdida de paquetes del 10 %

Observación 1. Para una red tradicional, como Ethernet, la probabilidad de colisión esta entre $5-10\,\%$ y se tiene que la carga útil de un paquete de datos está entre 46-1500 octetos [40]. Esto quiere decir que en una cuantización de 8 bits cada paquete de datos puede contener entre 46 y 1500 datos.

En la Figura 5.2 podemos ver un ejemplo de colisión de paquetes realizado en MATLAB. Cabe destacar que los datos en MATLAB son tipo double, por lo que estos tienen un tamaño de 8 octetos. La red de comunicación simulada tiene probabilidad de colisión de paquetes del 10 % y los datos se empaquetan con $L = 10$, teniendo 80 octetos cada paquete. En **(a)** de la Figura 5.2 se observa una colisión de paquetes sin retransmisión, por lo que, entre 31 y 40 no se tiene conocimiento de los datos enviados por el transmisor. Debido a esto, el conjunto de datos en la salida tiene valores no observables, no siendo posible ocupar métodos clásicos de identificación.

5.2. Estimación conjunta de τ y β

En esta sección, se desarrolla un algoritmo iterativo para evaluar el vector de parámetros y el retardo del sistema.

5.2.1. Algoritmo EM para estimar el vector de parámetros

El objetivo de la identificación es estimar los parámetros del sistema dinámico a través de estimación de máxima verosimilitud usando los datos:

$$\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_N\}, \tag{5.2.1}$$

$$\mathcal{U} = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_N\}. \tag{5.2.2}$$

En este caso, el problema de estimación puede ser escrito como el siguiente problema de optimización:

$$\hat{\beta} = \arg\max_{\beta} \ell(\beta), \tag{5.2.3}$$

donde $\ell(\beta) = \log\left[p(\mathcal{Z}|\beta)\right]$ es la función log-verosimilitud. Suponemos que el vector de parámetros β_0, la entrada $\mathbf{u}_t$ y el ruido $\mathbf{w}_t$ satisfacen las condiciones de regularidad, garantizando que la solución β del problema de optimización en (5.2.3) converge (en probabilidad o c.s.) a la solución verdadera β_0.

Debido a que el problema de interés puede sufrir colisiones de paquetes, no tenemos absoluta disponibilidad de la salida, por lo que el problema de optimización (5.2.3) no puede ser calculado directamente. Esto nos lleva al uso del algoritmo EM [24], en el cual tenemos que los datos perdidos y los datos observables, se denotarán como $\mathcal{Y}_{\mathbf{miss}}$ e $\mathcal{Y}_{\mathbf{obs}}$, respectivamente. Elegimos el conjunto de datos completo como:

$$\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N\}. \tag{5.2.4}$$

Por lo tanto, la función auxiliar en el algoritmo EM está dada por:

$$\mathcal{Q}(\beta, \hat{\beta}^{(i)}) = \mathbb{E}\left\{\log\left[p(\mathcal{Y}|\beta)\right] | \mathcal{Y}_{\mathbf{obs}}, \hat{\beta}^{(i)}\right\}, \tag{5.2.5}$$

donde $\hat{\beta}^{(i)}$ es la estimación actual y la siguiente estimación está dada por:

$$\hat{\beta}^{(i+1)} = \arg\max_{\beta} \mathcal{Q}(\beta, \hat{\beta}^{(i)}). \tag{5.2.6}$$

Cálculo de la función log-verosimilitud con el conjunto de datos completo

Para desarrollar el algoritmo EM en (5.2.5) y (5.2.6), primero necesitamos obtener una expresión para la función log-verosimilitud completa.

Lema 1. *La función de máxima verosimilitud con el conjunto de datos completo* $\log\left[p(\mathcal{Y}|\beta)\right]$ *esta dado por:*

$$\log\left[p(\mathcal{Y}|\beta)\right] = -\frac{N}{2}\log\left[2\pi\sigma^2\right] - \frac{1}{2\sigma^2}\sum_{t=1}^{N}\varepsilon_t^2, \tag{5.2.7}$$

donde:

$$\varepsilon_t = \mathbf{y}_t - \frac{\hat{B}(q^{-1}, \theta)}{F(q^{-1}, \theta)}\mathbf{u}_t \tag{5.2.8}$$

$$= \mathbf{y}_t - \varphi_t^T\theta. \tag{5.2.9}$$

y el vector de regresores φ_t esta dado por:

$$\varphi_t^T = \left[\mathbf{u}_{t-1-\tau} \quad \cdots \quad \mathbf{u}_{t-n_b-\tau} \quad -\mathbf{x}_{t-1} \quad \cdots \quad \mathbf{x}_{t-n_f}\right]. \tag{5.2.10}$$

Demostración. A partir de (5.1.1) junto a (5.1.7), y utilizando la Regla de Bayes, tenemos que la función log-verosimilitud completa $\log\left[p(\mathcal{Y}|\beta)\right]$ está dada por:

$$\log\left[p(\mathcal{Y}|\beta)\right] = \log\prod_{t=1}^{N} p(\mathbf{y}_t|\mathbf{y}_{1:t-1},\beta) \tag{5.2.11}$$

$$= \sum_{t=1}^{N}\log\left[p(\mathbf{y}_t|\mathbf{y}_{1:t-1},\beta)\right], \tag{5.2.12}$$

donde $\mathbf{y}_{1:t-1} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{t-1}\}$.

Considerando que $\mathbf{w}_t$ es ruido blanco Gaussiano, tenemos que la función log-verosimilitud esta dada por:

$$\log\left[p(\mathcal{Y}|\beta)\right] = -\frac{N}{2}\log\left[2\pi\sigma^2\right] - \frac{1}{2\sigma^2}\sum_{t=1}^{N}\varepsilon_t^2 \tag{5.2.13}$$

con:

$$\varepsilon_t = \mathbf{y}_t - \frac{\hat{B}(q^{-1},\theta)}{F(q^{-1},\theta)}\mathbf{u}_t. \tag{5.2.14}$$

Usando el vector de regresores de (5.2.10) se obtiene:

$$\varepsilon_t = \mathbf{y}_t - \varphi_t^T\theta. \tag{5.2.15}$$

□

Cálculo de la función auxiliar $\mathcal{Q}(\beta, \hat{\beta}^{(i)})$

Tal como se menciona en el Capítulo 2, el algoritmo EM es un procedimiento iterativo que produce una secuencia de estimaciones $\hat{\beta}^{(i)}$ para $i = 1, 2, \ldots$ de los parámetros β, que garantiza la convergencia a un máximo local de la función log-verosimilitud. Por lo tanto, para la implementación del algoritmo EM, se tiene lo siguiente.

Lema 2. *La función auxiliar $\mathcal{Q}(\beta, \hat{\beta}^{(i)})$ del algoritmo EM dado en* (5.2.5) *esta dada por:*

$$\mathcal{Q}(\beta,\hat{\beta}^{(i)}) = -\frac{N}{2}\log\left[2\pi\sigma^2\right] - \frac{1}{2\sigma^2}\sum_{t=1}^{N}\left\{(\hat{\mathbf{y}}_t - \varphi_t^T\theta)^2 + \Sigma_t^{\mathbf{y}}\right\}, \tag{5.2.16}$$

donde:

$$\hat{\mathbf{y}}_t = \mathbb{E}\left\{\mathbf{y}_t|\mathcal{Y}_{\mathbf{obs}},\hat{\beta}^{(i)}\right\}, \tag{5.2.17}$$

$$\Sigma_t^{\mathbf{y}} = \mathbb{E}\left\{\mathbf{y}_t^2|\mathcal{Y}_{\mathbf{obs}},\hat{\beta}^{(i)}\right\} - \hat{\mathbf{y}}_t^2. \tag{5.2.18}$$

Demostración. La función auxiliar en el algoritmo EM es obtenida tomando la esperanza condicional de la función log-verosimilitud con el conjunto de datos completa dada en el Lema 2. Así, se obtiene:

$$\mathcal{Q}(\beta, \hat{\beta}^{(i)}) = -\frac{N}{2}\log\left[2\pi\sigma^2\right] - \frac{1}{2\sigma^2}\sum_{t=1}^{N}\left\{\varepsilon_t^2|\mathcal{Y}_{\mathbf{obs}}, \hat{\beta}^{(i)}\right\}. \tag{5.2.19}$$

Usando (5.2.15) en (5.2.19) se tiene:

$$\begin{aligned}\mathcal{Q}(\beta, \hat{\beta}^{(i)}) = &-\frac{N}{2}\log\left[2\pi\sigma^2\right] \\ &- \frac{1}{2\sigma^2}\sum_{t=1}^{N}\mathbb{E}\left\{\mathbf{y}_t^2 - 2\mathbf{y}_t\varphi_t^T\theta + \theta^T\varphi_t\varphi_t^T\theta|\mathcal{Y}_{\mathbf{obs}}, \hat{\beta}^{(i)}\right\} \quad (5.2.20)\\ = &-\frac{N}{2}\log\left[2\pi\sigma^2\right] - \frac{1}{2\sigma^2}\sum_{t=1}^{N}\left\{(\hat{\mathbf{y}}_t - \varphi_t^T\theta)^2 + \Sigma_t^{\mathbf{y}}\right\}, \quad (5.2.21)\end{aligned}$$

donde $\hat{\mathbf{y}}_t$ y $\Sigma_t^{\mathbf{y}}$ están dados en (5.2.17) y (5.2.18), respectivamente. □

Optimizando la función auxiliar $\mathcal{Q}(\beta, \hat{\beta}^{(i)})$

Para calcular las iteraciones en (5.2.6) tenemos que optimizar la función auxiliar $\mathcal{Q}(\beta, \hat{\beta}^{(i)})$ con respecto al vector de los parámetros $\hat{\beta}$.

Lema 3. *El vector de parámetros $\hat{\beta} = [\hat{\theta} \quad \hat{\sigma}^2]$ que optimiza la función auxiliar $\mathcal{Q}(\beta, \hat{\beta}^{(i)})$ en* (5.2.16) *con respecto β esta dado por:*

$$\hat{\theta} = \left[\sum_{t=1}^{N}\varphi_t\varphi_t^T\right]^{-1}\left[\sum_{t=1}^{N}\varphi_t\hat{\mathbf{y}}_t\right], \tag{5.2.22}$$

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{t=1}^{N}\left\{(\hat{\mathbf{y}}_t - \varphi_t^T\hat{\theta})^2 + \Sigma_t^{\mathbf{y}}\right\} \tag{5.2.23}$$

donde $\hat{\mathbf{y}}_t$ y $\Sigma_t^{\mathbf{y}}$ están dados en (5.2.17) *y* (5.2.18)*, respectivamente.*

Demostración. Tomando la derivada de $\mathcal{Q}(\beta, \hat{\beta}^{(i)})$ con respecto a θ e igualando a cero, se tiene:

$$\frac{\partial\mathcal{Q}(\beta, \hat{\beta}^{(i)})}{\partial\theta} = 0 \Rightarrow \hat{\theta} = \left[\sum_{t=1}^{N}\varphi_t\varphi_t^T\right]^{-1}\left[\sum_{t=1}^{N}\varphi_t\hat{\mathbf{y}}_t\right]. \tag{5.2.24}$$

Por otro lado, usando $\hat{\theta}$ en la función auxiliar en (5.2.16) se puede concentrar en términos de σ^2, es decir,

$$\mathcal{Q}^c(\sigma^2, \hat{\beta}^{(i)}) = \frac{N}{2}\log\left[2\pi\sigma^2\right] - \frac{1}{\sigma^2}\sum_{t=1}^{N}\left\{(\hat{\mathbf{y}}_t - \varphi_t^T\hat{\theta})^2 + \Sigma_t^{\mathbf{y}}\right\}. \tag{5.2.25}$$

Tomando la derivada de la función concentrada en (5.2.25) con respecto a σ^2 e igualando a cero, se tiene:

$$\frac{\partial \mathcal{Q}^c(\sigma^2, \hat{\beta}^{(i)})}{\partial \sigma^2} = 0 \;\Rightarrow \hat{\sigma}^2 = \frac{1}{N}\sum_{t=1}^{N}\left\{(\hat{\mathbf{y}}_t - \varphi_t^T\hat{\theta})^2 + \Sigma_t^{\mathbf{y}}\right\}. \tag{5.2.26}$$

□

Calculando $\hat{\mathbf{y}}_t$ y $\Sigma_t^{\mathbf{y}}$

Para obtener la solución de (5.2.22) y (5.2.23) se necesita calcular el valor explícito de $\hat{\mathbf{y}}_t$ y $\Sigma_t^{\mathbf{y}}$. Primero, notamos que si el dato es observable se tiene $\hat{\mathbf{y}}_t = \mathbf{y}_t$ y $\Sigma_t^{\mathbf{y}} = 0$. Por otro lado, consideramos $\mathbf{r}_t(\theta) = \varphi_t^T\theta$, entonces se tiene:

$$\mathcal{Y} = \mathcal{R}(\theta) + \mathcal{W}, \tag{5.2.27}$$

con $\mathcal{R}(\theta) = \{\mathbf{r}_1(\theta), \mathbf{r}_2(\theta), \cdots, \mathbf{r}_N(\theta)\}$ y $\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N\}$. Entonces $\mathcal{Y}$ es una variable Gaussiana con media $\mathcal{R}(\theta)$ y varianza $\sigma^2 I_N$, En nuestro problema, definimos $\mathcal{Y}_{\mathbf{obs}}$ e $\mathcal{Y}_{\mathbf{miss}}$ como:

$$\begin{bmatrix}\mathcal{Y}_{\mathbf{obs}}\\ \mathcal{Y}_{\mathbf{miss}}\end{bmatrix} = \begin{bmatrix}\mathbf{S}\\ \mathbf{S}^{\perp}\end{bmatrix}\mathcal{Y}, \tag{5.2.28}$$

donde $\mathbf{S}$ es una matriz de selección que recoge los datos observados de los datos completos y $\mathbf{S}^{\perp}$ es el complemento de $\mathbf{S}$. Además, la matriz que multiplica $\mathcal{Y}$ en (5.2.28) es una matriz de permutación invertible. Así, es posible encontrar la distribución conjunta de $\mathcal{Y}_{\mathbf{obs}}$ e $\mathcal{Y}_{\mathbf{miss}}$ en términos de σ^2 y θ. Considerando que el traspuesto de una matriz de permutación es igual a su inversa, tenemos que:

$$\begin{bmatrix}\mathcal{Y}_{\mathbf{obs}}\\ \mathcal{Y}_{\mathbf{miss}}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{S}\mathcal{R}(\theta)\\ \mathbf{S}^{\perp}\mathcal{R}(\theta)\end{bmatrix}, \sigma^2 I_N\right). \tag{5.2.29}$$

Finalmente, calculamos $\hat{\mathbf{y}}_t$ y $\Sigma_t^{\mathbf{y}}$ utilizando la fórmula para la distribución Gaussiana condicional. Se obtiene:

$$\hat{\mathbf{y}}_t = \begin{cases}\mathbf{y}_t & \text{si } \mathbf{y}_t \text{ es observable}\\ \varphi_t^T\hat{\theta}^{(i)} & \text{si } \mathbf{y}_t \text{ no es observable}\end{cases} \tag{5.2.30}$$

$$\Sigma_t^{\mathbf{y}} = \begin{cases}0 & \text{si } \mathbf{y}_t \text{ es observable}\\ (\hat{\sigma}^2)^{(i)} & \text{si } \mathbf{y}_t \text{ no es observable.}\end{cases} \tag{5.2.31}$$

5.2.2. Estimación de τ usando el criterio de información de Akaike

El criterio de información de Akaike establece una relación entre la estimación de máxima verosimilitud y la información de Kullback-Leibler [29]. Esta relación puede expresarse como:

$$\text{AIC} = -\log\left[p\left(\mathcal{Z}|\beta\right)\right] + \mathcal{L}, \tag{5.2.32}$$

donde $p(\mathcal{Z}|\beta)$ es la función de verosimilitud en el conjunto de datos completos, $\mathcal{Z}$ son las mediciones, β los parámetros a estimar, y $\mathcal{L}$ es el número de parámetros estimados. En AIC, el objetivo es obtener un modelo que minimice la expresión en (5.2.32), no solo proporcionando un buen ajuste sino también con el mínimo de parámetros posibles, penalizando la dimensión de los parámetros.

Para estimar el parámetro τ, en la tesis nos enfocaremos en el uso de este método. De esta forma, podemos construir un conjunto de modelos $\mathcal{M}$, en el cual se escogen modelos con distintos n_f y n_b, y para cada valor escogido, se escoge un conjunto de valores τ el cual lo reemplazaremos en (5.2.10).

Para obtener el resultado en (5.2.32), la función log-verosimilitud completa en (5.2.32) es reemplazada por el valor obtenido de la función auxiliar $\mathcal{Q}(\beta, \hat{\beta}^{(i)})$ al converger el algoritmo EM. La estimación de $\hat{\tau}$ y $\hat{\theta}$ corresponden a los valores del modelo que minimiza la función (5.2.32).

Capítulo 6

SIMULACIONES

En este capítulo se analiza el desempeño del algoritmo propuesto en la presente Tesis. El objetivo principal es realizar experimentos de simulaciones en el software MATLAB, realizando ejemplos numéricos en dos distintos escenarios. Este capítulo finaliza con un análisis de sensibilidad que destaca los resultados de la estimación de parámetros del sistema.

6.1. Ejemplo numérico

Consideremos el siguiente sistema:

$$\mathbf{x}_t = \frac{b_1 q^{-1}}{1 + a_1 q^{-1}} \mathbf{u}_t \tag{6.1.1}$$

$$\mathbf{y}_t = \mathbf{x}_{t-\tau} + \mathbf{w}_t \tag{6.1.2}$$

donde $a_1 = -0{,}5$, $b_1 = 1$, $\mathbf{w}_t \sim \mathcal{N}(0, \sigma^2)$ con $\sigma^2 = 0{,}1$.

Se quiere estimar los parámetros a_1, b_1, σ^2 y τ. Las simulaciones se realizan considerando las siguientes condiciones:

- El largo de los datos es $N = 100$.
- Se realizan 100 simulaciones de Montecarlo.
- La señal de entrada es un ruido blanco Gaussiano de media cero y varianza $\sigma_u^2 = 1$.
- En el conjunto de modelos consideramos solo un modelo con $n_b = 1$ y $n_f = 1$, y $\tau = 0 \ldots 7$ (esto quiere decir que el conjunto de modelos contiene 8 modelos).

Además, el desempeño de los estimadores han sido evaluados con error cuadrático medio normalizado de los parámetros del sistema, dado por:

$$\text{NMSE } \hat{\theta} = \frac{1}{M} \sum_{i=1}^{M} \frac{||\hat{\theta}_i - \theta_0||_2^2}{||\theta_0||_2^2}, \tag{6.1.3}$$

y el error cuadrático medio normalizado de la varianza del ruido:

$$\text{NMSE } \hat{\sigma}^2 = \frac{1}{M}\sum_{i=1}^{M}\frac{||\hat{\sigma}_i^2 - \sigma_0^2||_2^2}{||\sigma_0^2||_2^2}, \tag{6.1.4}$$

donde θ_0 y σ_0^2 son los valores verdaderos del sistema y M es la simulación de Montecarlo correspondiente.

6.1.1. Caso 1: Sistema dinámico sujeto a un retardo desconocido.

Se considera un sistema con un retardo de $\tau = 2$ y una probabilidad de recibir el paquete de datos de $p(\lambda_{\mathbf{p}} = 1) = 1 - q$ con $q = 0$ (no hay pérdida de paquetes).

Los resultados se muestran en las Figura 6.1, Figura 6.2 y Figura 6.3. En la Figura 6.1 vemos que se estima correctamente el retardo en las 100 simulaciones de Montecarlo. En la Figura 6.2 logramos ver que se estima correctamente los parámetros que definen el sistema. El error cuadrático medio normalizado de la varianza es 0,0019 y el de los parámetros es $8{,}56 \times 10^{-4}$.

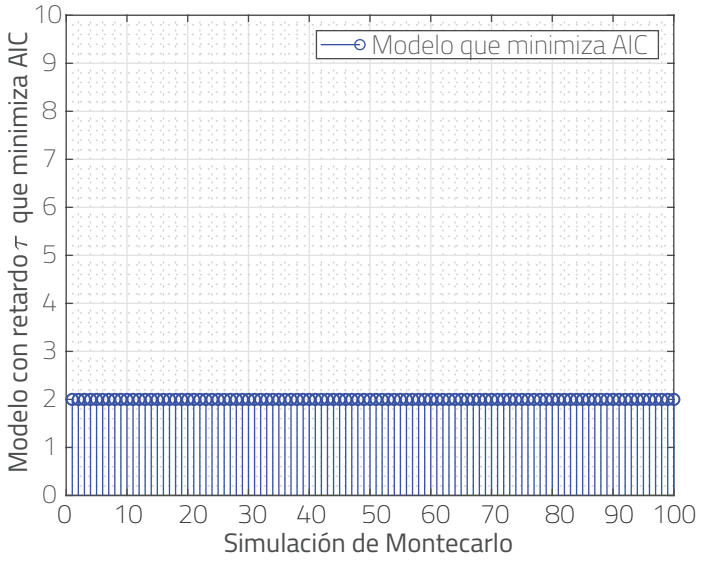

Figura 6.1. Elección de mejor modelo para *Caso 1.*

6.1.2. Caso 2: Sistema dinámico sujeto a colisión de paquetes.

Se considera un sistema un retardo de $\tau = 3$ y una probabilidad de recibir el paquete de datos de $p(\lambda_{\mathbf{p}} = 1) = 1 - q$ con $q = 0{,}1$ (90 % de probabilidad de recibir el paquete, la cual es una probabilidad de entrega considerada normal en una red Ethernet),y con paquetes de largo $L = 5$. Debido a la cantidad de modelos

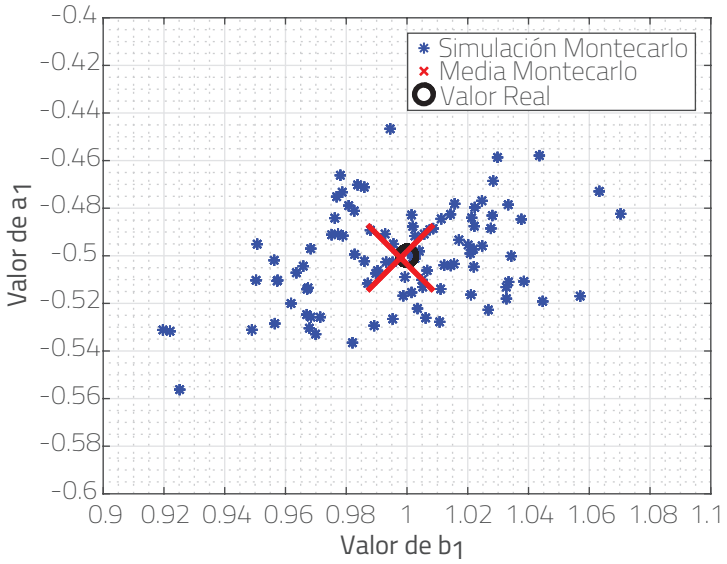

Figura 6.2. Estimación de parámetros a_1 y b_1 para *Caso 1.*

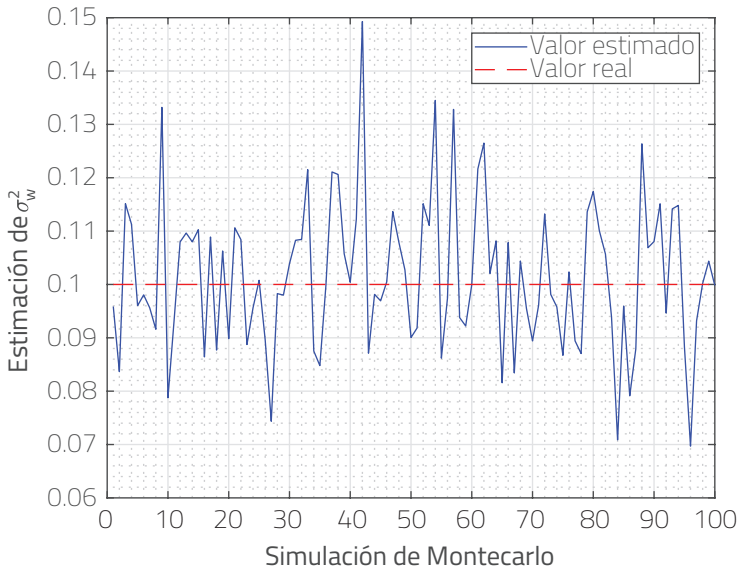

Figura 6.3. Estimación de σ^2 para *Caso 1.*

e iteraciones que se debe realizar por el uso del algoritmo EM y AIC, realizamos una aproximación de AIC para optimizar el cálculo computacional [35].

Los resultados se muestran en las Figura 6.4, Figura 6.5 y Figura 6.6. En la

Figura 6.4 logramos ver que también se estima correctamente el retardo en las 100 simulaciones de Montecarlo. En la Figura 6.5 logramos ver que se estima correctamente los parámetros que definen el sistema. El error cuadrático medio normalizado de la varianza es 0,0025 y el de los parámetros es $9{,}77 \times 10^{-4}$.

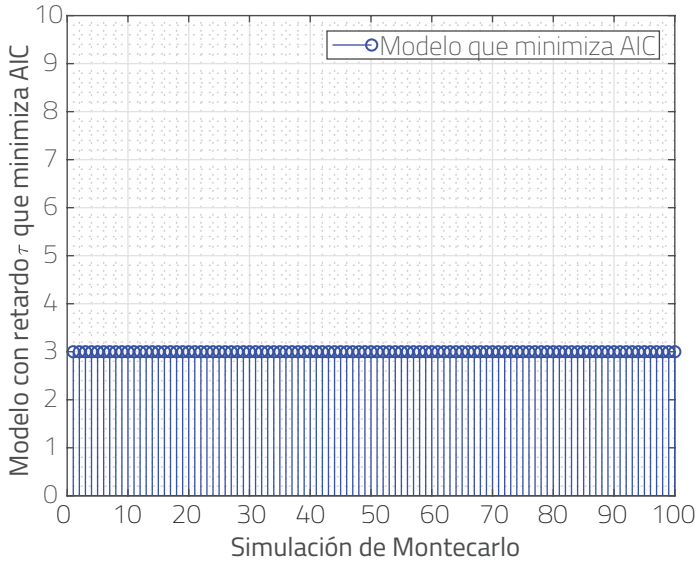

Figura 6.4. Elección de mejor modelo para *Caso 2*.

6.1.3. Análisis de sensibilidad variando el valor de la probabilidad de colisión de paquetes.

Debido a que en los casos anteriores no se logra apreciar cuando la estimación comienza a ser poco confiable se realiza un análisis de sensibilidad, variando el valor de la probabilidad de colisión de paquetes $p(s_{\mathbf{p}} = 0) = q$ y con un retardo de $\tau = 3$. Se realiza un análisis detallado de 50 casos, variando el parámetro q de 0.02 en 0.02 en el intervalo de 0 a 1. Para este estudio revisamos el valor NMSE de $\hat{\sigma}^2$ y $\hat{\theta}$, graficamos en un diagrama de cajas el valor de los parámetros a_1 y b_1, y se grafica la cantidad de veces que es estimado de forma correcta el valor de τ para cada caso.

Los resultados se muestran en las Figuras 6.7, 6.8, 6.9 y 6.10. En la Figura 6.7 se observa que los parámetros a_1 y b_1 tienen una media cercana al valor verdadero. Sin embargo, a medida que el valor q crece las estimaciones presentan una mayor varianza. Esto también se observa en las Figuras 6.9 y 6.10, donde podemos apreciar que el NMSE se mantiene sin variabilidad hasta aproximadamente $q = 0{,}7$, donde el valor empieza a crecer de manera considerable. Por otro lado, en la Figura 6.8 se observa que el retardo τ es estimado de manera correcta en todas las simulaciones de Montecarlo hasta aproximadamente $q = 0{,}7$, donde comienzan a fallar de manera

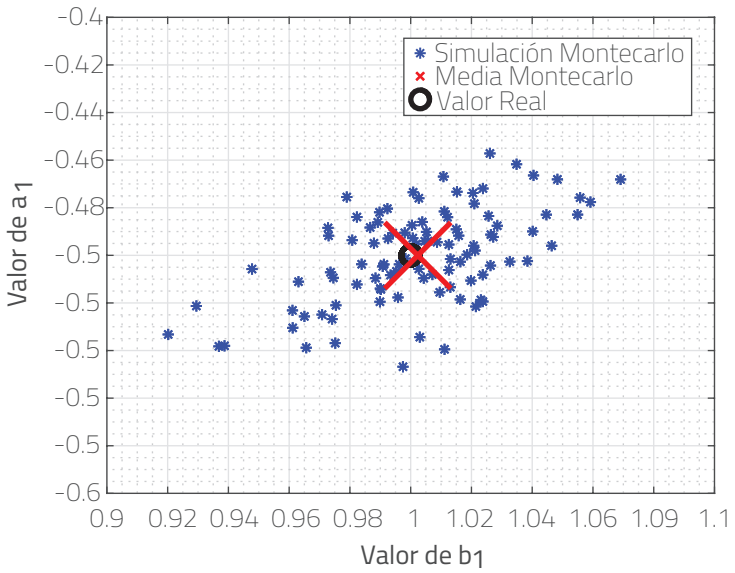

Figura 6.5. Estimación de parámetros a_1 y b_1 para *Caso 2*.

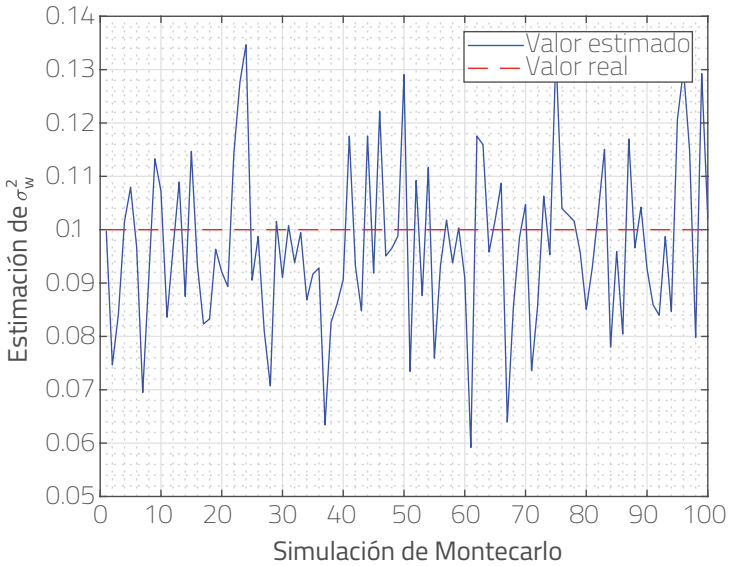

Figura 6.6. Estimación de σ^2 para *Caso 2*.

considerable hasta $q = 1$ (falla en su totalidad). Con esto podemos concluir que la estimación comienza a perder confiabilidad, tanto en τ como β, a medida que q aumenta, teniendo un mayor error en la estimación desde aproximadamente $q = 0{,}7$.

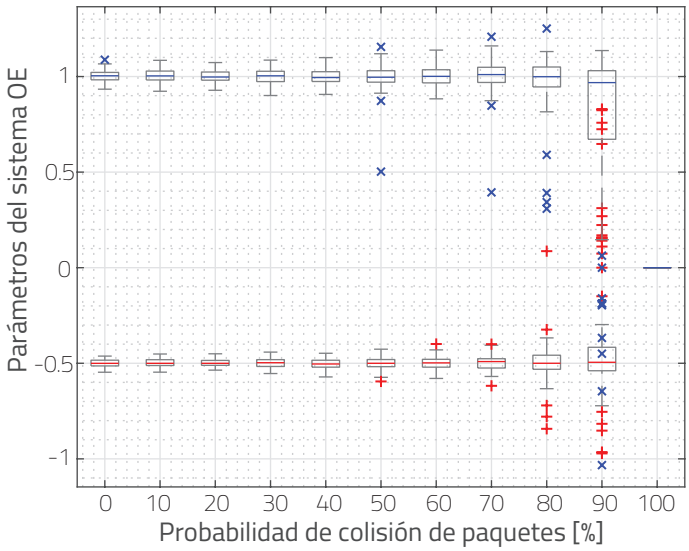

Figura 6.7. Estimación los parámetros a_1 (en rojo) y b_1 (en azul) variando el valor de la probabilidad de colisión de paquetes $p(\lambda_{\mathbf{p}} = 0) = q$.

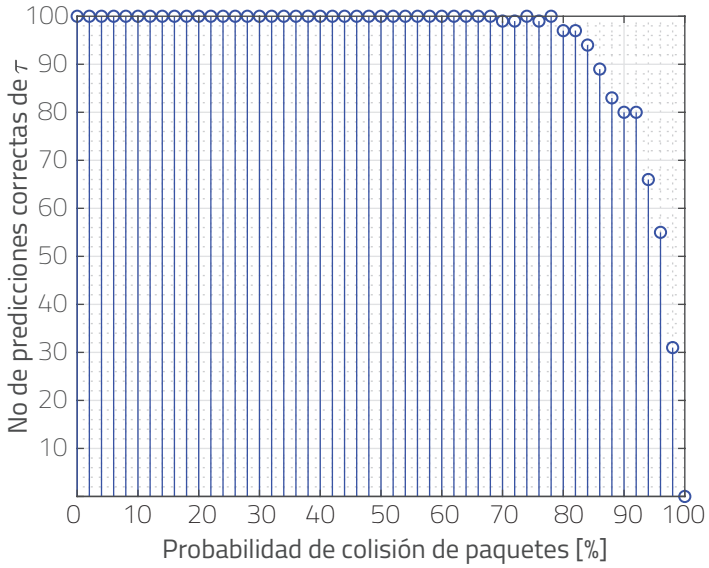

Figura 6.8. Número de estimaciones correctas de $\hat{\tau}$ variando el valor de la probabilidad de colisión de paquetes $p(\lambda_{\mathbf{p}} = 0) = q$.

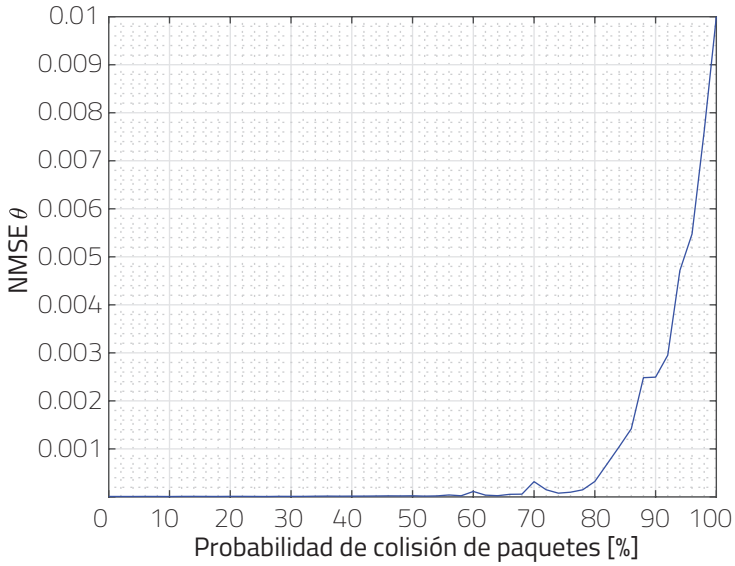

Figura 6.9. NMSE de $\hat{\theta}$ variando el valor de la probabilidad de colisión de paquetes $p(\lambda_{\mathbf{p}} = 0) = q$.

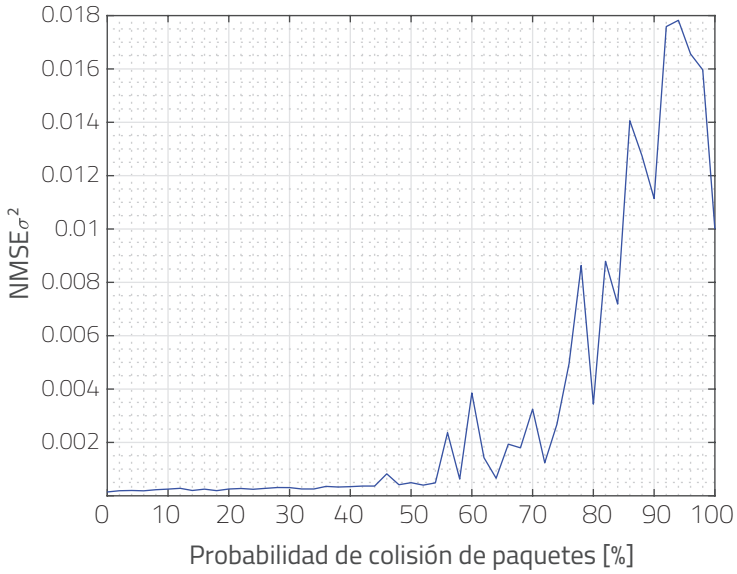

Figura 6.10. NMSE de $\hat{\sigma}^2$ variando el valor de la probabilidad de colisión de paquetes $p(\lambda_{\mathbf{p}} = 0) = q$.

Capítulo 7

CONCLUSIONES Y TRABAJOS A FUTURO

7.1. Conclusiones

La presentación de esta tesis expone la estimación conjunta de parámetros, tanto en el retardo del tiempo como de los parámetros del sistema dinámico, a través de estimación por máxima verosimilitud, basado en el algoritmo EM y el criterio de información de Akaike.

En una primera parte, se estudiaron los conceptos básicos de la estimación por máxima verosimilitud, algoritmo EM y el criterio de información de Akaike, con el objetivo de desarrollar un algoritmo de estimación de parámetros en un sistema sobre redes de comunicaciones. Para un mayor entendimiento del funcionamiento de las redes de comunicación, se estudiaron los conceptos básicos de redes de computadores, detallando las causas de la pérdida de datos y los retardos. La finalidad del estudio inicial permitió comprender el funcionamiento de las principales restricciones en redes de comunicación, con el motivo de incluirlas en el modelado de una manera sencilla.

Adicionalmente, el estudio de los sistemas de control sobre redes de comunicación trajo a presencia el avance investigativo respecto al modelado de las restricciones consideradas en la presente tesis. De esta manera, se expone el modelado en el contexto de sistemas de control sobre redes.

Posteriormente, se desarrolla un estimador por máxima verosimilitud a través del algoritmo EM, considerando las incertidumbres causadas por el canal de comunicación. Se realiza una estimación conjunta de los parámetros del sistema dinámico, y el retardo causado por la red, mediante el criterio de información de Akaike.

En base a lo anterior, se llevan a cabo dos casos de experimentación, donde en el primero se ejecuta una simulación de un sistema dinámico, sujeto a un retardo desconocido, y el segundo caso, se genera una simulación de un sistema dinámico sujeto a retardo desconocido y a la pérdida de paquetes. Para determinar el desempeño de la estimación en los casos simulados, se usa una métrica de error cuadrático medio normalizado. En ambos casos el error cuadrático medio normalizado indicó que la estimación se aproxima a los valores reales. Por consiguiente, se realiza un análisis

de sensibilidad variando la probabilidad de pérdida de paquetes, con el propósito de determinar el momento en el cual la estimación de parámetros comienza a fallar.

Finalmente, hemos mostrado un método de estimación general para un sistema dinámico, en el cual nos basamos en ML y AIC para la estimación conjunta de β y τ. Los ejemplos numéricos muestran la efectividad de nuestro enfoque, observando que la estimación de los parámetros del sistema logra obtenerse de forma fiable. Además, a partir del análisis de sensibilidad realizado concluye que la estimación de parámetros comienza a fallar con valores cercanos al 70 %, debido a que la pérdida de información disminuye el desempeño del estimador.

7.2. Trabajo a futuro

Complementando el trabajo realizado, un objetivo para trabajos a futuro es la posibilidad de ampliar este método para otros criterios de información, como por ejemplo BIC. Además, se puede añadir el estudio de estimación conjunta entre sistemas dinámicos y redes de comunicaciones, basándonos en comportamientos más complejos que estas tienen o considerando redes más reales. Por ejemplo, se puede realizar un enfoque similar al realizado en [68], [58], donde usan un enfoque de teoría de juegos entre el estimador y el ataque *DoS* realizado.

Además se puede ampliar el estudio en el envío y recibo de paquetes en una red de comunicación de forma experimental, diseñando un sistema a escala de un sistema de comunicación real como se observa en la Figura 7.1. De esta forma, se puede generar una situación donde la red de comunicación es atacada maliciosamente por un ataque *DoS* mediante un inyector de tráfico diseñado en un FPGA ZedBoard, y los datos serían leídos en computadores personales por un analizador de protocolos, como por ejemplo Wireshark. Así, se busca aplicar el algoritmo desarrollado con datos experimentales reales. De esta manera, se podría analizar cómo afectan los distintos protocolos de capa de transporte y el tráfico inyectado en el ataque.

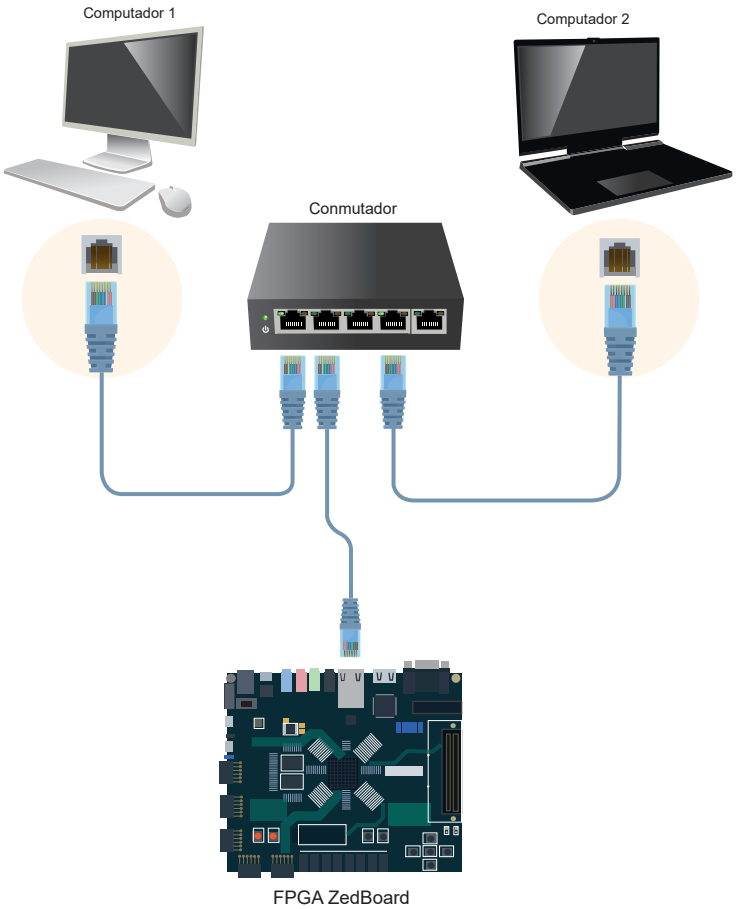

Figura 7.1. Esquema experimental para la identificación de sistemas en red

REFERENCIAS

[1] L. Ljung, *System Identification: Theory for the User*. USA: Prentice-Hall, Inc., 1986.

[2] K.-D. Kim and P. R. Kumar, "Cyber–physical systems: A perspective at the centennial," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1287–1308, 2012.

[3] W. Le-Yi and Z. Wen-Xiao, "System identification: new paradigms, challenges, and opportunities," *Acta automatica sinica*, vol. 39, no. 7, pp. 933–942, 2013.

[4] M. S. Mahmoud and M. M. Hamdan, "Fundamental issues in networked control systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 5, pp. 902–922, 2018.

[5] M. Trivellato and N. Benvenuto, "State control in networked control systems under packet drops and limited transmission bandwidth," *IEEE Transactions on Communications*, vol. 58, no. 2, pp. 611–622, 2010.

[6] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.

[7] G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans, "Feedback control under data rate constraints: An overview," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 108–137, 2007.

[8] T. Söderström and P. Stoica, "Instrumental variable methods for system identification," *Circuits, Systems and Signal Processing*, vol. 21, no. 1, pp. 1–9, 2002.

[9] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[10] D. Marelli, K. You, and M. Fu, "Identification of ARMA models using intermittent and quantized output observations," *Automatica*, vol. 49, no. 2, pp. 360–369, 2013.

[11] F. Ding, G. Liu, and X. P. Liu, "Parameter estimation with scarce measurements," *Automatica*, vol. 47, no. 8, pp. 1646–1655, 2011.

[12] A. J. Isaksson, "Identification of ARX-models subject to missing data," *IEEE Transactions on Automatic Control*, vol. 38, no. 5, pp. 813–819, 1993.

[13] J. Munkhammar, L. Mattsson, and J. Rydén, "Polynomial probability distribution estimation using the method of moments," *PloS one*, vol. 12, no. 4, p. e0174573, 2017.

[14] T. S. Jaakkola and M. I. Jordan, "Bayesian parameter estimation via variational methods," *Statistics and Computing*, vol. 10, no. 1, pp. 25–37, 2000.

[15] R. A. Fisher, "On the mathematical foundations of theoretical statistics," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 222, no. 594-604, pp. 309–368, 1922.

[16] K. J. Åström and T. Bohlin, "Numerical identification of linear dynamic systems from normal operating records," *IFAC Proceedings Volumes*, vol. 2, no. 2, pp. 96–111, 1965.

[17] J. C. Agüero, J. I. Yuz, G. C. Goodwin, and R. A. Delgado, "On the equivalence of time and frequency domain maximum likelihood estimation," *Automatica*, vol. 46, no. 2, pp. 260–270, 2010.

[18] B. I. Godoy, G. C. Goodwin, J. C. Agüero, D. Marelli, and T. Wigren, "On identification of FIR systems having quantized output data," *Automatica*, vol. 47, no. 9, pp. 1905–1915, 2011.

[19] T. Söderström and U. Soverini, "Errors-in-variables identification using maximum likelihood estimation in the frequency domain," *Automatica*, vol. 79, pp. 131–143, 2017.

[20] G. C. Goodwin and R. L. Payne, *Dynamic system identification: experiment design and data analysis.* USA: Academic Press, 1977.

[21] K. Astrom, "Maximum likelihood and prediction error methods," *IFAC Proceedings Volumes*, vol. 12, no. 8, pp. 551–574, 1979.

[22] R. J. Rossi, *Mathematical statistics: An introduction to likelihood based inference.* John Wiley & Sons, 2018.

[23] E. L. Lehmann, *Elements of large-sample theory.* Springer Science & Business Media, 2004.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[25] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions*, vol. 382. John Wiley & Sons, 2007.

[26] R. Durrett, *Probability: theory and examples*, vol. 49. Cambridge university press, 2019.

[27] C. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of statistics*, pp. 95–103, 1983.

[28] K. P. Burnham and D. R. Anderson, *Model selection and multimodel inference: A practical information-theoretic approach*. Springer, 2 ed., 2002.

[29] K. P. Burnham and D. R. Anderson, "Multimodel inference: understanding AIC and BIC in model selection," *Sociological methods & research*, vol. 33, no. 2, pp. 261–304, 2004.

[30] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[31] R. E. Kass and A. E. Raftery, "Bayes factors," *Journal of the american statistical association*, vol. 90, no. 430, pp. 773–795, 1995.

[32] S. Kullback, *Information theory and statistics*. Courier Corporation, 1997.

[33] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *2nd International Symposium on Information Theory, 1973*, pp. 267–281, Akademiai Kaido, 1973.

[34] H. Akaike, "A new look at the statistical model identification," *IEEE transactions on automatic control*, vol. 19, no. 6, pp. 716–723, 1974.

[35] R. Carvajal, G. Urrutia, and J. C. Agüero, "An optimization-based algorithm for model selection using an approximation of Akaike's Information Criterion," in *2016 Australian Control Conference (AuCC)*, pp. 217–220, IEEE, 2016.

[36] A. L. Russell, "The internet that wasn't," *IEEE Spectrum*, vol. 50, no. 8, pp. 39–43, 2013.

[37] R. Kahn and V. Cerf, "A protocol for packet network intercommunication," *IEEE Transactions on Communications*, vol. 22, no. 5, pp. 637–648, 1974.

[38] V. Cerf, Y. Dalal, and C. Sunshine, "Specification of Internet Transmission Control Program," RFC 675, RFC Editor, December 1974. `http://www.rfc-editor.org/rfc/rfc675.txt`.

[39] J. Postel, "Transmission Control Protocol," RFC 793, RFC Editor, September 1981. `http://www.rfc-editor.org/rfc/rfc793.txt`.

[40] J. Kurose and K. Ross, *Computer networks: A top down approach featuring the internet*. Pearson, 6 ed., 2013.

[41] J. Postel, "User Datagram Protocol," STD 6, RFC Editor, August 1980. `http://www.rfc-editor.org/rfc/rfc768.txt`.

[42] T. Lammle, *CCNA: Cisco Certified Network Associate: Fast Pass*. John Wiley & Sons, 2008.

[43] N. Abramson, "The ALOHA system: another alternative for computer communications," in *Proceedings of the November 17-19, 1970, fall joint computer conference*, pp. 281–285, 1970.

[44] N. Abramson, "Development of the ALOHANET," *IEEE transactions on Information Theory*, vol. 31, no. 2, pp. 119–123, 1985.

[45] L. G. Roberts, "ALOHA packet system with and without slots and capture," *ACM SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.

[46] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part I-carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE transactions on Communications*, vol. 23, no. 12, pp. 1400–1416, 1975.

[47] S. S. Lam, "A carrier sense multiple access protocol for local networks," *Computer Networks (1976)*, vol. 4, no. 1, pp. 21–32, 1980.

[48] M. Molle, K. Sohraby, and A. Venetsanopoulos, "Space-time models of asynchronous CSMA protocols for local area networks," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 6, pp. 956–968, 1987.

[49] R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed packet switching for local computer networks," *Communications of the ACM*, vol. 19, no. 7, pp. 395–404, 1976.

[50] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher, *Internet denial of service: attack and defense mechanisms (Radia Perlman Computer Networking and Security)*. Prentice Hall PTR, 2004.

[51] F. Lau, S. H. Rubin, M. H. Smith, and L. Trajkovic, "Distributed denial of service attacks," in *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0*, vol. 3, pp. 2275–2280, IEEE, 2000.

[52] M. Al-Shurman, S.-M. Yoo, and S. Park, "Black hole attack in mobile ad hoc networks," in *Proceedings of the 42nd annual Southeast regional conference*, pp. 96–97, 2004.

[53] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data networks*. Prentice-Hall International New Jersey, 2 ed., 1992.

[54] K. You, N. Xiao, and L. Xie, *Analysis and design of networked control systems*. Springer, 2015.

[55] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[56] G. N. Nair and R. J. Evans, "Exponential stabilisability of finite-dimensional linear systems with limited data rates," *Automatica*, vol. 39, no. 4, pp. 585–593, 2003.

[57] G. N. Nair and R. J. Evans, "Stabilizability of stochastic linear systems with finite feedback data rates," *SIAM Journal on Control and Optimization*, vol. 43, no. 2, pp. 413–436, 2004.

[58] K. Ding, S. Dey, D. E. Quevedo, and L. Shi, "Stochastic game in remote estimation under DoS attacks," *IEEE control systems letters*, vol. 1, no. 1, pp. 146–151, 2017.

[59] K. Ding, X. Ren, D. E. Quevedo, S. Dey, and L. Shi, "Dos attacks on remote state estimation with asymmetric information," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 2, pp. 653–666, 2018.

[60] F. Cid, F. J. Vargas, and A. Maass, "Feedback control over lossy channels: Optimal estimation considering data-loss compensation strategies," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 5366–5371, IEEE, 2017.

[61] F. Arriagada and F. Vargas, "Optimal state prediction in feedback systems with data loss compensation strategies," in *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, pp. 1–7, IEEE, 2017.

[62] K. You and L. Xie, "Minimum data rate for mean square stabilizability of linear systems with markovian packet losses," *IEEE Transactions on Automatic Control*, vol. 56, no. 4, pp. 772–785, 2010.

[63] J. Wu and T. Chen, "Design of networked control systems with packet dropouts," *IEEE Transactions on Automatic control*, vol. 52, no. 7, pp. 1314–1319, 2007.

[64] T. Wigren, "Networked and delayed recursive identification of nonlinear systems," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 5851–5858, IEEE, 2017.

[65] Y.-L. Wang and Q.-L. Han, "Modelling and controller design for discrete-time networked control systems with limited channels and data drift," *Information Sciences*, vol. 269, pp. 332–348, 2014.

[66] N. Vatanski, J.-P. Georges, C. Aubrun, E. Rondeau, and S.-L. Jämsä-Jounela, "Networked control with delay measurement and estimation," *Control Engineering Practice*, vol. 17, no. 2, pp. 231–244, 2009.

[67] R. A. Delgado, K. Lau, R. H. Middleton, and T. Wigren, "Networked delay control for 5G wireless machine-type communications using multiconnectivity," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 4, pp. 1510–1525, 2018.

[68] K. Ding, Y. Li, D. E. Quevedo, S. Dey, and L. Shi, "A multi-channel transmission schedule for remote state estimation under DoS attacks," *Automatica*, vol. 78, pp. 194–201, 2017.