



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

**SISTEMAS DE CONTROL SOBRE REDES UTILIZANDO REDES NEURONALES
Y MODELOS OCULTOS DE MARKOV**

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

PRESENTA:
ALONSO RAMOS FERNÁNDEZ

TUTOR PRINCIPAL
HÉCTOR BENÍTEZ PÉREZ
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN
SISTEMAS

CIUDAD UNIVERSITARIA, CD. MX., ENERO 2021



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A mi familia. Muchas gracias por apoyarme en todo momento

Abstract

This document presents the design of a neural predictive controller for the stabilization of networked control systems with stochastic communication delay. This control scheme uses control of discrete systems with constant time delay, reset-based control systems, neural networks, and hidden Markov models.

The proposed control is given by the sum of a nominal control and a corrective control. The nominal control is designed for a linearized model with constant time delay. Corrective control is generated by a reset-based control and a neural network-based control. The latter is proposed as a one-layer neural network whose input is the control system error and whose output is a control signal. Neural network parameters are learned online to minimize controller error using the gradient descent method.

Subsequently, a hidden Markov model is proposed to predict the exit of the system taking into account its stochastic nature. The parameters of the hidden Markov model are learned through an online learning algorithm and based on the observations of the system. The output of the system is predicted based on the latest measurements, that is, using a sliding time window approach, which is achieved through the forward algorithm. The prediction of the output allows a predictive control law, which consists of using the prediction as the new input to the neural network.

Finally, the proposed method was implemented in a non-linear multiple-input multiple-output system and the performance was evaluated through simulations. The results show that the prediction system is capable of modeling the stochastic nature of the system and that the neural predictive controller adapts and compensates for the disturbances due to the stochastic communication delay showing a better performance compared to other controllers.

Resumen

Este documento presenta el diseño de un controlador neuronal predictivo para la estabilización de sistemas de control sobre redes con retardo de comunicación estocástico. Este esquema de control utiliza control de sistemas discretos con retardo de tiempo constante, sistemas de control basados en *reset*, redes neuronales y modelos ocultos de Markov.

El control propuesto viene dado por la suma de un control nominal y un control correctivo. El control nominal se diseña para un modelo linealizado con retardo de tiempo constante. El control correctivo se genera mediante un control basado en *reset* y un control basado en redes neuronales. Este último se propone como una red neuronal de una capa cuya entrada es el error del sistema de control y cuya salida es una señal de control. Los parámetros de la red neuronal se aprenden en línea para minimizar el error del controlador mediante el método del descenso por gradiente.

Posteriormente, se propone un modelo oculto de Markov para predecir la salida del sistema tomando en cuenta la naturaleza estocástica de este. Los parámetros del modelo oculto de Markov se aprenden por medio de un algoritmo de aprendizaje en línea y con base en las observaciones del sistema. La salida del sistema se predice con base en las últimas mediciones, es decir, se utiliza un enfoque de ventana de tiempo deslizante, lo cual se logra utilizando el algoritmo de avance. La predicción de la salida permite plantear una ley de control predictivo, la cual consiste en utilizar a la predicción como la nueva entrada a la red neuronal.

Finalmente, el método propuesto se implementó a un sistema no lineal múltiple-entrada múltiple-salida y se evaluó el desempeño mediante simulaciones. Los resultados demuestran que el sistema de predicción es capaz de modelar la naturaleza estocástica del sistema y que el control neural predictivo se adapta y compensa a las perturbaciones debidas al retardo de comunicación estocástico mostrando un mejor desempeño en comparación a otros controladores.

Índice general

1. Introducción	1
1.1. Justificación	1
1.2. Objetivo	2
1.3. Estructura de la tesis	3
1.4. Estado del arte	3
1.5. Sistemas de control sobre redes	6
1.5.1. Modelado de NCSs como un TDS en tiempo continuo	7
1.5.2. Modelado de NCSs como un TDS en tiempo discreto	11
2. Diseño de un controlador neuronal predictivo	15
2.1. Control nominal	15
2.2. Sistema de control basado en <i>reset</i>	20
2.3. Control basado en redes neuronales	24
2.4. Sistema de predicción basado en un modelo oculto de Markov	27
3. Implementación del controlador	41
4. Resultados	59
4.1. Pitch	62
4.2. Yaw	76
4.3. Comparación del desempeño	78
4.4. Análisis del retardo mínimo y máximo	82
5. Conclusiones	87
Bibliografía	93

Introducción

1.1. Justificación

Con el surgimiento de tecnologías de redes de comunicación de alta velocidad, el número de aplicaciones de sistemas de control distribuidos sobre estas ha incrementado rápidamente. Las redes de comunicación permiten a un grupo de dispositivos intercambiar información de forma remota, fácil y económica. El uso de las redes de comunicación digital en los sistemas de control se debe a muchas razones, por ejemplo, en aplicaciones donde se requiere cubrir una gran distancia física, además, gracias a la modularidad y la facilidad de implementación que proporcionan, el mantenimiento de los sistemas de control se facilita con lo cual también se reducen costos. Gracias a estos beneficios cada vez más sectores industriales están aplicando redes de comunicación a sistemas de control. Entre estas industrias se encuentra la industria de la manufactura, que busca crear sistemas automatizados en los que sensores, actuadores y controladores intercambien información y al mismo tiempo permitan a usuarios en distintos sitios acceder a esta y realizar co-manipulación de los procesos coordinadamente. Las aplicaciones en la industria de las redes de comunicación inalámbricas también están en aumento, por ejemplo, en la industria espacial, en la aviación y en la teleoperación de robots ([Liu and Rees \(2005\)](#)). El control basado en internet también ha demostrado tener muchas aplicaciones prácticas en la industria de la manufactura, esto gracias a que el internet puede ser usado como el medio de comunicación sin tener que gastar dinero en construir una red de comunicación dedicada al sistema de control, además, es posible usar al internet para controlar y supervisar procesos industriales ([Yang \(2011\)](#)).

Los sistemas de control en los que la transmisión de la información funciona a través de redes de comunicación digital son conocidos como sistemas de control sobre redes (Networked Control Systems (NCSs)). Por otra parte, utilizar una red de comunicación digital en un sistema de control también conlleva la presencia de fenómenos complejos entre los cuales se encuentran la pérdida de paquetes de datos, la presencia de señales muestreadas y el retardo de tiempo inducido por la red de comunicación; este retardo

también es conocido simplemente como retardo de comunicación. En la presente tesis, como se explicará detalladamente en la sección 1.5, se observa a estos fenómenos desde el punto de vista estocástico pues se deben a diversos factores presentes en un sistema de cómputo, por ejemplo, un ancho de banda limitado, congestión de la red y una administración de recursos (memoria y procesador) poco eficiente.

Estos fenómenos complejos son adversos en el sentido de que traen como consecuencia un degradamiento en el desempeño del controlador e incluso pueden provocar la inestabilidad del sistema. Por lo tanto, para que un NCS funcione de acuerdo al desempeño deseado se debe diseñar un controlador que tome en cuenta estos fenómenos y que logre compensar sus efectos adversos, lo cual es un reto enorme sobre todo por la naturaleza estocástica de estos; por lo tanto los NCS representan un área de estudio muy amplio, con muchas estrategias propuestas hasta el momento y con muchos retos aún por resolver.

Por otra parte, dos conceptos muy importantes del campo de la inteligencia artificial son las redes neuronales y los modelos probabilísticos gráficos. En trabajos anteriores se han demostrado los beneficios de utilizar redes neuronales para mejorar el desempeño de NCSs gracias a la adaptabilidad que estas presentan. Por otra parte, utilizando modelos probabilísticos gráficos es posible modelar y hacer predicciones de sistemas estocásticos. En un NCS, el retardo de comunicación hace que el controlador no cuente con la información actual del sistema lo cual degrada su desempeño. Contar con la predicción de la salida del sistema y plantear una ley de control predictivo es un método para generar una señal de control más acorde con la actualidad del sistema y así compensar al retardo de comunicación. Por estas razones, en la presente tesis se profundiza más en el uso de estos conceptos a NCSs.

1.2. Objetivo

La presente tesis tiene por objetivo diseñar un controlador neuronal predictivo que sea capaz de estabilizar un NCS con un mejor desempeño en comparación a otros controladores. El cumplimiento del objetivo se demuestra con un enfoque basado en ejemplos y por medio de simulaciones. Para el diseño del controlador y las simulaciones de ejemplos de efectividad se tomó en cuenta un retardo de comunicación y un periodo de muestreo (presencia de señales muestreadas) aleatorio, sin embargo, no se consideró a la pérdida de paquetes de datos pues es un tema de estudio bastante amplio por si mismo.

Con el propósito de estudiar y verificar su aplicación a los NCSs el controlador propuesto se compone, en parte, de una red neuronal de una sola capa y de un modelo oculto de Markov (Hidden Markov Model (HMM)), un modelo probabilístico gráfico muy popular. Se propone usar a la red neuronal para generar un control adaptable a las perturbaciones de naturaleza estocástica que existen en un NCS, incluyendo al

retardo de tiempo, es decir, se busca diseñar un controlador que cambie con base en los datos medidos en tiempo real y una vez que pase la etapa de aprendizaje genere una señal adecuada para mejorar el desempeño del controlador. Por otra parte, con el HMM se propone modelar la naturaleza estocástica de los NCSs y predecir la salida del sistema para plantear una ley de control predictivo que ayude a compensar los efectos del retardo de comunicación.

1.3. Estructura de la tesis

Esta tesis se estructura como sigue. En la sección 1.4 se expone la revisión de la literatura de controladores propuestos para estabilizar NCSs estocásticos. En la sección 1.5 se describe el modelado de NCSs como sistemas con retardo de tiempo (Time-Delay Systems (TDSs)) desde dos perspectivas: en tiempo continuo y en tiempo discreto.

En el capítulo 2 se propone justificadamente un controlador neuronal predictivo con base en los controladores neuronal predictivos planteados en [Cho and Fadali \(2011\)](#) y [Cho and Lee \(2008\)](#). El controlador propuesto en la presente tesis presenta varias diferencias respecto a [Cho and Fadali \(2011\)](#) y [Cho and Lee \(2008\)](#), siendo la mayor de estas el uso de un HMM para la predicción de la salida del sistema, mientras que se tiene en común el uso de redes neuronales.

Posteriormente, en el capítulo 3, el controlador propuesto se implementa para lograr el objetivo de estabilización en un NCS en específico: un sistema electromecánico no lineal múltiple-entrada múltiple-salida (Multiple-input Multiple-output (MIMO)) que cuenta con 4 hélices como actuadores. Finalmente, en el capítulo 5, se presentan los resultados de las simulaciones con los cuales se demuestra el cumplimiento de los objetivos de esta tesis.

1.4. Estado del arte

En [Fridman \(2014\)](#) se propone un funcional de Lyapunov (o funcional de Lyapunov-Krasovskii) con el cual se garantiza la estabilidad asintótica de NCSs lineales. Los funcionales de Lyapunov son la generalización de las funciones clásicas de Lyapunov a sistemas con retardo de tiempo. Para obtener el funcional de Lyapunov en [Fridman \(2014\)](#) se hace un análisis en tiempo continuo y se modela a los NCSs como sistemas con un retardo variable en el tiempo, el cual, como se explicará en la sección 1.5.1, se define como una función del tiempo que es continua a trazos y que está acotada por arriba y por abajo.

Existen otros enfoques en los que los NCS son modelados por medio de ecuaciones diferenciales estocásticas. Este tipo de ecuaciones son similares a las ecuaciones diferen-

1. INTRODUCCIÓN

ciales de la teoría de control clásica en el sentido de que evolucionan en tiempo continuo, sin embargo, se diferencian por la presencia de procesos estocásticos (generalmente el proceso de Wiener también conocido como movimiento Browniano). Al utilizar este tipo de ecuaciones para representar a los NCSs el retardo es caracterizado como un proceso estocástico. Un ejemplo en el que se considera este enfoque es Verriest (2002), donde representan a NCS lineales con retardos aleatorios por medio de ecuaciones diferenciales tipo Ito y se utilizan extensiones estocásticas de la teoría de Lyapunov-Krasovskii para llegar a condiciones suficientes de estabilidad estocástica global y asintótica.

Otro marco de referencia son recientes extensiones de la teoría del control digital en donde se utilizan ecuaciones en diferencias tipo Ito para representar a los NCSs y se utilizan procesos estocásticos para considerar al retardo de comunicación. De acuerdo con Ma and Wang (2019), en años recientes, extensiones de la teoría de sistemas de control en tiempo discreto han empezado a recibir creciente atención simplemente porque se puede considerar que actualmente la mayoría de las estrategias de control son implementadas en tiempo discreto. En Ma and Wang (2019) se plantean varias técnicas discretas de control robusto para contrarrestar a las perturbaciones estocásticas provocadas por el retardo de comunicación y garantizar la estabilidad de NCS estocásticos en tiempo discreto. La mayoría de las técnicas propuestas son variaciones del control por modos deslizantes para sistemas en tiempo discreto, por ejemplo, la combinación del control por modos deslizantes con el control H_∞ .

Para estudiar a los NCSs, sin recurrir al cálculo estocástico ni las ecuaciones diferenciales funcionales y desde un marco de referencia adaptable al retardo de tiempo, se puede utilizar teoría del campo de la inteligencia artificial. Por ejemplo, en Wanigasekara and Almkhles (2019) se introduce el controlador basado en redes neuronales NARMA-L2, el cual se compone de dos pasos. Primero se entrena una red neuronal multicapa por medio del algoritmo de retropropagación para aprender un modelo autorregresivo de media móvil no lineal (Nonlinear Autoregressive-Moving Average (NARMA)) para modelar la dinámica del sistema. Posteriormente, se plantea como la señal de control a la función que minimiza el error de seguimiento, esta función también se propone como un modelo NARMA y se aprende utilizando otra red neuronal y el algoritmo de retropropagación. Sus resultados demuestran que el desempeño del control basado en redes neuronales es superior a otros controladores, como el controlador PID.

Otro ejemplo es Cho and Lee (2008), aquí se plantea un controlador que tiene como objetivo resolver el problema de seguimiento de un sistema no lineal sobre redes. Los NCSs se modelan como un sistema en tiempo discreto con retardo de comunicación representado por una función del tiempo y se logra el objetivo por medio de la suma de dos señales de control diseñadas por separado: la compensante y la nominal. La primera se diseña para un retardo constante (el cual se puede considerar como su valor esperado) usando teoría de TDS lineales discretos, mientras que la segunda se trata de un control neuronal predictivo y es para compensar las perturbaciones generadas por los demás

valores que puede tomar el retardo. El control neuronal predictivo consiste de una red neuronal recurrente cuya entrada es el error de seguimiento y la salida es la señal de control compensante. Esta red neuronal se entrena con un aprendizaje fuera de línea por medio del método del descenso por gradiente y teniendo como función objetivo a minimizar al error cuadrático de seguimiento. Adicionalmente se modela la naturaleza estocástica de un NCS por medio de una cadena de Markov. Con base en este modelo es posible predecir la salida del sistema a partir de la salida actual y establecer una ley de control predictivo.

Un ejemplo similar a [Cho and Lee \(2008\)](#) es [Cho and Fadali \(2011\)](#). Las principales diferencias con [Cho and Lee \(2008\)](#) son que en [Cho and Fadali \(2011\)](#) se plantea a los NCSs estocásticos en tiempo continuo, el control nominal es un control por linealización exacta y se diseña sin considerar mínimamente al retardo de tiempo, además, el control compensante se compone de una red neuronal (la cual ahora es una capa de perceptrones) con aprendizaje en línea, una red Bayesiana dinámica para la predicción y un control basado en *reset*.

Sobre el control basado en *reset*, de acuerdo con [Barreiro and Baños \(2012\)](#), se ha demostrado con un enfoque basado en ejemplos que el control basado en *reset* es capaz de alcanzar prestaciones inalcanzables con controladores lineales, especialmente para el caso de plantas sometidas a limitaciones fundamentales lineales, sin embargo, esta afirmación está lejos de ser demostrada de una manera rigurosa y cuantitativa, y es una línea que merece investigaciones adicionales. Las limitaciones fundamentales lineales que hace referencia son retardos de tiempo, ceros de fase no mínima y polos inestables $1/(s - p)$ con $p \gg 0$. Estas limitaciones llegan a imposibilitar el funcionamiento práctico de un sistema de control lineal. En [Barreiro and Baños \(2012\)](#) también se plantea un ejemplo de un sistema de control lineal que cumple con regulación y al cual se le añadió un control basado en *reset*, más específicamente, el elemento reseteado de primer orden (First Order Reset Element (FORE)), el cual es el control basado en *reset* que se propone utilizar en la presente tesis. El resultado fue que se mantuvo la rapidez del sistema (tiempo de levantamiento), sin embargo, las oscilaciones se redujeron significativamente, mejorando la robustez del sistema y reduciendo el tiempo de asentamiento.

Un trabajo en donde se utiliza control predictivo en NCSs es [Liu and Rees \(2005\)](#). En este trabajo se plantea a los sistemas de control predictivo sobre redes (Networked Predictive Control Systems (NPCSs)) y se demostró, por medio de simulaciones y estudios experimentales, que el control predictivo es un método eficaz para compensar al retardo de comunicación. Los NPCSs se componen de tres partes: un controlador diseñado sin tomar en cuenta al retardo de comunicación usando métodos de control tradicionales (PID, LQG, etc.), un generador de predicciones de control (Control Prediction Generator (CPG)) y un compensador del retardo de comunicación (Network Delay Compensator (NDC)). El CPG es diseñado para generar, con base en las salidas anteriores del sistema, un conjunto de predicciones de control futuras. El NDC se en-

cuentra en el nodo de la planta y, por medio de un sistema de estimación del retardo de comunicación, conoce el valor de este y selecciona de entre las predicciones hechas la entrada de control para la planta. En el presente trabajo se adopta un enfoque similar pues se predice la salida del sistema y con base en esta predicción, se establece una ley de control predictivo, sin embargo, como no se cuenta con el NDC y no es posible conocer el retardo de comunicación, la predicción se hace para un valor esperado del retardo de comunicación.

1.5. Sistemas de control sobre redes

En los NCSs los sensores, los actuadores y el controlador intercambian información a través de una red de comunicación digital. Las ventajas de utilizar este tipo de sistemas son, por ejemplo, un costo más reducido, facilidad de instalación y mantenimiento, y un incremento en la flexibilidad del sistema. Sin embargo, desde el punto de vista de control, los NCS conllevan nuevos retos pues cerrar el lazo sobre una red de comunicación introduce perturbaciones como el retardo de comunicación, intervalos de muestreo variables, y limitaciones por la programación de envío de paquetes de comunicación (scheduling communication constraints), las cuales afectan el desempeño del sistema e incluso la estabilidad del mismo (Fridman (2014)).

En los NCSs reales también pueden suceder pérdidas de paquetes de información. Estos paquetes de información son mediciones de la salida o valores calculados de la señal de control. Las pérdidas suceden cuando hay una falla en el protocolo de comunicación o colisión entre mensajes (Fridman (2014)). A pesar de que los protocolos de comunicación están equipados con mecanismos de retransmisión, estos solo pueden reenviar la información por un tiempo limitado, después del cual se deja de intentar el envío y por lo tanto el paquete de información se pierde, es decir, no llega a su destino. Otra situación que puede provocar que se pierdan datos de información es que un nuevo paquete de información esté disponible cuando un viejo paquete aún no sea haya transmitido. En este caso, la mayoría de los mecanismos de retransmisión están programados o se encuentran en la configuración para descartar los datos viejos que no se han transmitido y transmitir el paquete más nuevo disponible. De este manera, el controlador y la planta siempre reciben los datos más recientes lo cual es ventajoso con fines de control aunque formalmente se pierde información.

Hay varios enfoques para estudiar a los NCSs, entre los que se encuentra la teoría de TDS. En esta teoría, a su vez, se consideran dos marcos de referencia: el tiempo continuo y el tiempo discreto. En la sección 1.5.1 se va a desarrollar el modelado de NCS desde el punto de vista de la teoría de TDS en tiempo continuo y posteriormente, en la sección 1.5.2, desde el enfoque de sistemas de control en tiempo discreto.

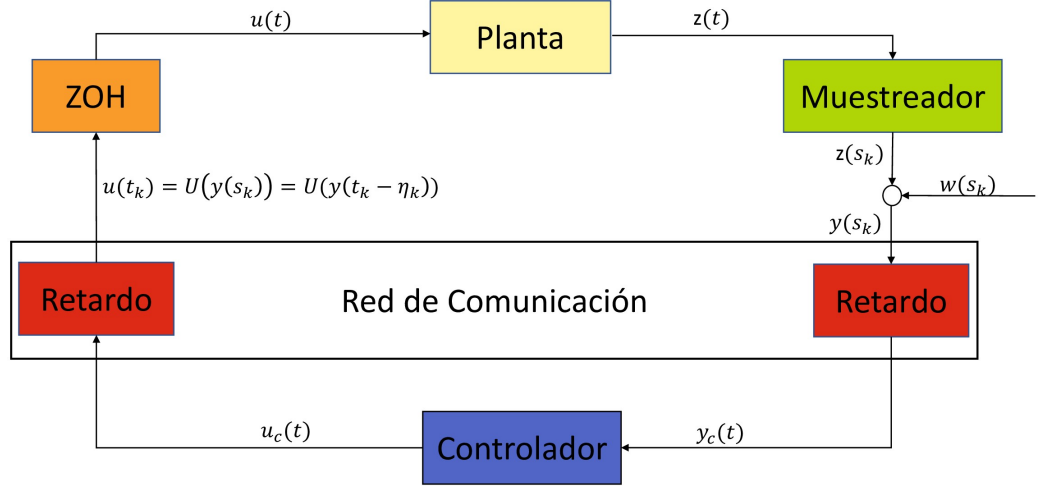


Figura 1.1: Sistema de control de retroalimentación de la salida sobre redes.

1.5.1. Modelado de NCSs como un TDS en tiempo continuo

Para estudiar los NCSs utilizando la teoría de TDSs se to man en consideración las siguientes imperfecciones: intervalos de muestreo variables, el retardo de comunicación, y la pérdida de paquetes de información (Fridman (2014)). Sin embargo, en el presente trabajo solo se van a considerar a las primeras dos; considere la siguiente planta de una entrada y una salida (Single-Input-Single-Output (SISO)) a controlar:

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t)) \\ z(t) &= h(x(t))\end{aligned}\tag{1.1}$$

Donde $x(t) \in \mathbb{R}^n$ es el vector de estado, $f : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$, $h : \mathbb{R}^n \rightarrow \mathbb{R}$, $u(t) \in \mathbb{R}$ es la entrada de control y $z(t) \in \mathbb{R}$ es la salida del sistema. También considere que si $x(t) = \bar{0}$ y $u(t) = 0$ entonces $\dot{x}(t) = \bar{0}$, en otras palabras, el origen ($x(t) = \bar{0}$) de 1.1 es un punto de equilibrio. Ahora considere el sistema de control de retroalimentación de la salida sobre redes mostrado en la figura 1.1. El muestreador representa a un sensor el cual mide la salida del sistema $z(t)$ cada cierto intervalo de tiempo. Al tiempo transcurrido entre dos mediciones de la salida se conoce como el periodo de muestreo, el cual es variable en el tiempo.

El muestreador induce una discontinuidad al sistema de control pues debido a este la señal que llega al controlador es discontinua. El instante en el que se mide la salida del sistema $z(s_k)$ es denotado por s_k y se conoce como un instante de muestreo. Este instante se define formalmente de acuerdo a la definición 1 (Castillo (2020)).

Definición 1. Sea $\{s_k\}$ una secuencia estrictamente creciente en $(0, \infty) \forall k \in \mathbb{Z}^+$. Entonces $\{s_k\}$ es una secuencia de muestreo si en cada instante s_k se mide la salida

1. INTRODUCCIÓN

del sistema $z(s_k)$. Cada s_k en una secuencia de muestreo se conoce como el k -ésimo instante de muestreo.

Una vez definido el instante de muestreo también es posible definir formalmente al periodo de muestreo.

Definición 2. El k -ésimo periodo de muestreo se define como: $T_k = s_{k+1} - s_k, \forall k \in \mathbb{Z}^+$.

Por otra parte, la salida del sistema puede ser corrompida por una señal de ruido de medición $w(s_k)$, por lo tanto se define:

$$y(s_k) = z(s_k) + w(s_k), \quad \forall k \in \mathbb{Z}^+ \quad (1.2)$$

Donde $y(s_k)$ es la información enviada por el sensor al controlador a través de la red de comunicación en los instantes de muestreo, sin embargo, esta información no llega inmediatamente al nodo del controlador, sino que pasa un tiempo $\tau_{y(s_k)}$ de duración aleatoria, el cual es conocido como el retardo de observación. A continuación, se define formalmente al retardo de observación ([Castillo \(2020\)](#)):

Definición 3. Sea s_k es un instante de muestreo, el k -ésimo retardo de observación $\tau_{y(s_k)} \in \mathbb{R}^+ \forall k \in \mathbb{Z}^+$ se define como el tiempo transcurrido desde s_k hasta la llegada de la información $y(s_k)$ al controlador.

Se considera que el controlador calcula una ley de control únicamente cuando este detecta que un nuevo dato proveniente del sensor llega a él, lo cual es conveniente, pues de lo contrario se calcularían leyes de control con datos desactualizados y se perdería tiempo de procesamiento e incluso se podrían perder datos de medición más actuales. Una vez llega la información al controlador, este calcula una ley de control y la envía al actuador, sin embargo, nuevamente esta información no llega inmediatamente al controlador, sino que pasa un tiempo $\tau_{u(t_k)}$ de duración aleatoria, el cual es conocido como el retardo de control. A continuación se define formalmente al retardo de control ([Castillo \(2020\)](#)):

Definición 4. Sea s_k un instante de muestreo, el k -ésimo retardo de control $\tau_{u(t_k)} \in \mathbb{R}^+ \forall k \in \mathbb{Z}^+$ se define como el tiempo transcurrido desde la llegada de la información $y(s_k)$ al controlador hasta la llegada de la ley de control calculada con base en $y(s_k)$ al actuador.

También se toma en cuenta que el actuador está programado para que cuando una señal de control llega a él, esta se mantenga así hasta que llegue una nueva señal de control, en otras palabras, teóricamente la acción o salida del actuador permanece igual hasta que recibe una señal de control nueva. Esta consideración se le conoce como control de retenedor de orden cero (Zero Order Hold (ZOH)) y está representada en la

figura 1.1 por el bloque ZOH. Debido a que se considera que el controlador y el ZOH actualizan su salida tan pronto como reciben un nuevo dato se dice que son elementos conducidos por eventos, a diferencia del muestreador, el cual se dice que es conducido por tiempo (Fridman (2014)).

Es importante mencionar que la consideración de que el controlador actualiza su salida tan pronto como lee un nuevo dato no pasa en un NCS real pues en realidad al controlador le toma un tiempo procesar la información y calcular la ley de control. Este tiempo puede ser considerablemente grande dependiendo de la velocidad de procesamiento del controlador y de si la ley de control es muy compleja e implica muchos cálculos, por lo tanto se debe tomar en consideración para el diseño del controlador. El modelo considerado en el presente trabajo sí toma en cuenta que existe el tiempo de procesamiento del controlador pues su efecto se puede considerar sin ningún problema como un aumento del retardo de control $\tau_{u(t_k)}$. Por otra parte, cada instante de tiempo que se actualiza la señal de control en el nodo del actuador, es decir, el instante de actualización del ZOH, se le conoce como instante de actualización, formalmente:

Definición 5. Sea $\{t_k\}$ una secuencia estrictamente creciente en $(0, \infty) \forall k \in \mathbb{Z}^+$. Entonces $\{t_k\}$ es una secuencia de actualización si la señal de control en el nodo del actuador es actualizada en cada instante t_k a consecuencia de la información medida $y(s_k)$. Cada t_k en una secuencia de muestreo se conoce como el k -ésimo instante de actualización.

Ahora que se definieron a los instantes de muestreo y de actualización es posible definir la señal de control de 1.1, la cual, se trata del control de ZOH antes mencionado:

$$u(t) = U(y(s_k)), \quad t_k \leq t < t_{k+1}, \quad \forall k \in \mathbb{Z}^+ \quad (1.3)$$

Donde $U : \mathbb{R} \rightarrow \mathbb{R}$; note que de la definición 5 se tiene que $t_k > s_k$, por lo tanto, $t_k - s_k$ representa el tiempo que existe entre el muestreo y la actualización de la señal de control. Además, es claro que: $t_k - s_k = \tau_{u(t_k)} + \tau_{y(s_k)} \forall k \in \mathbb{Z}^+$, es decir, el tiempo que existe entre el muestreo y la actualización de la señal de control es la suma de los retardos de tiempo producidos por la red de comunicación, por lo tanto se puede definir al retardo de comunicación de acuerdo a la siguiente definición.

Definición 6. Sea $\{s_k\}$ una secuencia de muestreo y $\{t_k\}$ una secuencia de actualización como se presentó en las definiciones 1 y 5. Entonces se define al k -ésimo retardo de comunicación como $\eta_k = t_k - s_k, \forall k \in \mathbb{Z}^+$, donde $\eta_k > 0$.

Note que el retardo de comunicación η_k puede ser más grande que el periodo de muestreo T_k (definición 2) siempre y cuando la secuencia de actualización $\{t_k\}$ se man-

1. INTRODUCCIÓN

tenga estrictamente creciente. Con base en la definición 6 la ecuación 1.3 se puede escribir:

$$u(t) = U(y(t_k - \eta_k)), \quad t_k \leq t < t_{k+1}, \quad \forall k \in \mathbb{Z}^+ \quad (1.4)$$

Desde el punto de vista de TDSs en tiempo continuo para el estudio de NCSs conviene definir la siguiente función del tiempo:

$$\tau(t) = t - t_k + \eta_k, \quad t_k \leq t < t_{k+1}, \quad \forall k \in \mathbb{Z}^+ \quad (1.5)$$

Con base en 1.5, 1.4 puede escribirse:

$$u(t) = U(y(t - \tau(t))) \quad (1.6)$$

Sustituyendo 1.6 en 1.1 se obtiene finalmente el modelo en lazo cerrado de un NCS:

$$\dot{x}(t) = f(x(t), U(y(t - \tau(t)))) \quad (1.7)$$

En la ecuación 1.7 se puede apreciar claramente porque se dice que un NCS es un sistema con retardo variable en el tiempo y por lo tanto la teoría de TDSs en tiempo continuo es aplicable para su estudio. Además, cuando se implementa un sistema de control real es muy difícil saber con precisión cada cuanto tiempo se actualizará la señal de control en el nodo del actuador (los instantes de actualización) y cada cuanto se medirá la salida del sistema en el nodo del sensor (los instantes de muestreo). Esto es debido a que el tiempo transcurrido entre estos instantes fluctúa aleatoriamente pues depende, entre otros factores, de la congestión de la red de comunicación y de la administración de recursos en el procesador. Como se puede apreciar en 1.7, el retardo en un NCS depende de estos instantes, por lo tanto, estos instantes hacen que el retardo tenga una naturaleza aleatoria.

Note que de la ecuación 1.5, $\dot{\tau}(t) = 1 \forall t \neq t_k$ y $\tau(t) > 0 \forall t \in \mathbb{R}^+$ dado que $\eta_k > 0 \forall k \in \mathbb{Z}^+$. Además, asumiendo que existe un lapso de tiempo máximo τ_m entre s_k y t_{k+1} , entonces:

$$t_{k+1} - s_k = t_{k+1} - t_k + \eta_k \leq \tau_m \quad (1.8)$$

De 1.5 y 1.8 se tiene que $\tau(t) < \tau_m$. Por lo tanto se puede decir que bajo la perspectiva de TDSs el retardo variante en el tiempo a la entrada $\tau(t)$ está acotado por arriba y por abajo. En la figura 1.2 se puede apreciar un ejemplo de $\tau(t)$.

Es importante mencionar que en este modelo se hace la consideración de que η_{k+1} depende de cierta forma de η_k . Esto es debido a que $t_k = \eta_k + s_k$ y que $\{t_k\}$ es estrictamente creciente. Para ejemplificar esto considere $s_k = 3$ y $\eta_k = 0.5$, por lo tanto, $t_k = 3.5$, ahora considere $s_{k+1} = 3.1$, por lo tanto para mantener a $\{t_k\}$ estrictamente creciente se tiene que cumplir que $\eta_{k+1} > 0.4$, de lo contrario se tendría $t_{k+1} \leq t_k$ y por lo tanto $\{t_k\}$ no sería estrictamente creciente. Este es un factor importante que sucede

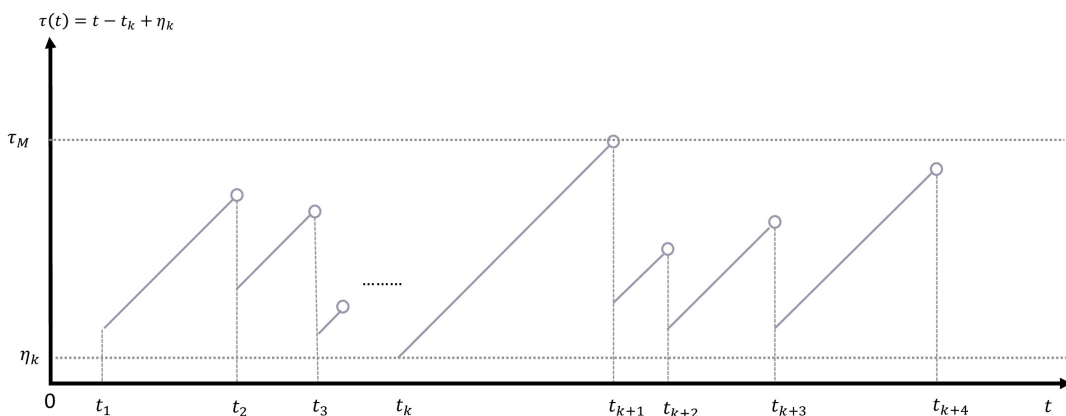


Figura 1.2: Retardo de tiempo $\tau(t)$ en un NCS (Fridman (2014)).

en los NCS reales y que se debe tomar en cuenta al momento de hacer las simulaciones.

Como se mencionó anteriormente, las pérdidas de información se deben a diversas razones y también afectan considerablemente el desempeño del controlador. Sin embargo, en el modelo presentado hasta el momento y, en general, en el presente trabajo no se toman en cuenta. En cambio, en Fridman (2014) para tomar en cuenta a las pérdidas de paquetes de información se considera que η_k y η_{k+1} son independientes, entonces se puede dar el caso de que $\{t_k\}$ no sea estrictamente creciente, en cuyo caso se descartan los paquete de información viejos, es decir, se descartan la señales de control que lleguen al actuador después que el paquete de información más reciente. Empero, en el presente trabajo no se descarta ningún dato.

1.5.2. Modelado de NCSs como un TDS en tiempo discreto

La teoría del control digital, también conocida como la teoría de sistemas de control en tiempo discreto, se caracteriza por la consideración de que el periodo de muestreo es constante y en principio fue planteada sin retardo de comunicación, sin embargo, como se explicará a continuación, existen extensiones de esta teoría que sí consideran al retardo.

Considere nuevamente al sistema a controlar 1.1, el cual es un sistema SISO en tiempo continuo. Al cerrar el lazo de control, como se mencionó en la sección 1.5.1, el periodo de muestreo (definición 2) no es constante pues depende de los instantes de muestreo los cuales suceden cada cierto intervalo de tiempo de duración aleatoria. Sin embargo, en los NCSs reales se puede considerar que el periodo de muestreo tiene poca varianza y un valor medio, $h_m \in \mathbb{R}^+$, el cual puede ser estimado de datos experimentales. Por lo tanto, con fines de diseño del controlador, se puede considerar que el periodo de muestreo es constante e igual al valor medio, es decir, $T_k = h_m, \forall k \in \mathbb{Z}^+$.

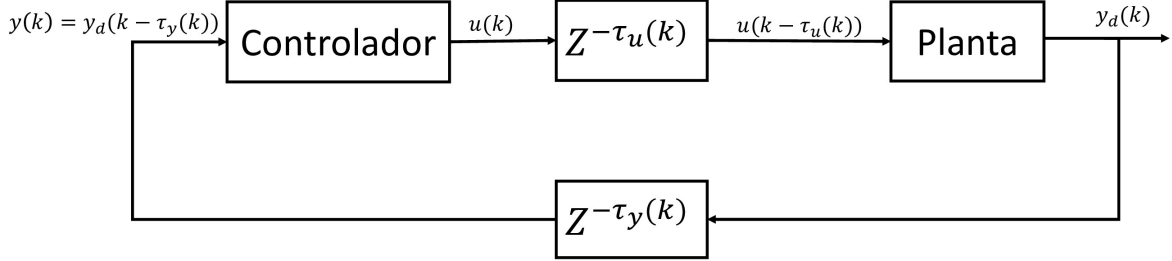


Figura 1.3: Diagrama de bloques de un NCS en tiempo discreto

Además, si se considera que la señal de control ZOH, $u(t)$, se actualiza (a consecuencia de una información medida anteriormente debido al retardo de comunicación) justo cuando se lee un nuevo dato en el nodo de sensor, es decir, cada periodo de tiempo h_m , entonces se puede considerar que la dinámica del sistema 1.1 evoluciona en tiempo discreto, es decir, se puede modelar la evolución del sistema 1.1 con ecuaciones en diferencias. Es importante mencionar que, aún bajo estas consideraciones, las definiciones de los instantes de muestreo y de actualización, definiciones 1 y 5, respectivamente, y las definiciones del periodo de muestreo y del retardo de comunicación, definiciones 2 y 6, respectivamente, se sostienen si se redefinen a los retardos de observación y de control, definiciones 3 y 4, respectivamente, pues las desviaciones del periodo de muestreo real respecto a h_m se pueden añadir o sustraer, según sea el caso, a los retardos de observación y de control.

Por otra parte, con un método numérico para resolver ecuaciones diferenciales ordinarias, por ejemplo, el método de Euler, dado un h_m , se pueden encontrar ecuaciones en diferencias que modelen la dinámica de 1.1 en tiempo discreto, es decir:

$$\begin{aligned} x(k+1) &= f_d(x(k), u_d(k)) \\ z(k) &= h(x(k)) \end{aligned} \quad (1.9)$$

Donde $k \in \mathbb{Z}^+ \cup \{0\}$ es el tiempo discreto, el cual cumple con: $t_k = kh_m$, donde $t_k \forall k$ son instantes del tiempo continuo utilizado en el modelo original 1.1 para lo cuales, bajo las consideraciones del muestreador y del ZOH mencionadas anteriormente, se cumple: $x(k) \approx x(t_k)$, donde $x(k)$ se calcula con 1.9, mientras que $x(t_k)$ con 1.1. En otras palabras, 1.9 es una aproximación de 1.1 en la que se toman en cuenta ciertas consideraciones convenientes para el diseño del controlador, por lo tanto, se va a utilizar 1.9 para el planteamiento del esquema de control. Por otra parte, $x(k) \in \mathbb{R}^n$ es el vector de estado, $f_d: \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$, $h: \mathbb{R}^n \rightarrow \mathbb{R}$, $u_d(k) \in \mathbb{R}$ es la entrada de control y $z(k) \in \mathbb{R}$ es la salida del sistema (señal a medir). Además, se cumple que si $x(k) = \bar{0}$ y $u_d(k) = 0$ entonces $x(k+1) = \bar{0}$, en otras palabras, el origen ($x(k) = \bar{0}$) de 1.9 es un punto de equilibrio.

Al igual que en el tiempo continuo consideramos un retardo de comunicación variable y ruido de observación. Por lo tanto, una expresión matemática general para un sistema NCS en tiempo discreto es (Cho and Lee (2008)):

$$\begin{aligned} x(k+1) &= f_d(x(k), u(k - \tau_u(k))) \\ y_d(k) &= z(k) + w(k) = h(x(k)) + w(k) \\ y(k) &= y_d(k - \tau_y(k)) \end{aligned} \quad (1.10)$$

Donde $u(k - \tau_u(k)) = u_d(k)$, $u(k)$ es la señal de control calculada en el controlador en el tiempo k , $y(k) \in \mathbb{R}$ es la señal que llega al controlador en el tiempo k , $w(k)$ representa ruido de medición y $\tau_u(k), \tau_y(k) \in \mathbb{Z}^+$, son los retardos de control y observación, respectivamente. Además, se consideran las siguientes condiciones iniciales para $u(k)$: $u(-\tau_u(0)) = u(1 - \tau_u(1)) = u(2 - \tau_u(2)) = \dots = u(0) = 0$, es decir, $u(k) = 0 \forall k \leq 0$. También se consideran las siguientes condiciones iniciales para $y_d(k)$: $y_d(-\tau_y(0)) = z(-\tau_y(0)) + w(-\tau_y(0)) = y_d(1 - \tau_y(1)) = z(1 - \tau_y(1)) + w(1 - \tau_y(1)) = y_d(2 - \tau_y(2)) = z(2 - \tau_y(2)) + w(2 - \tau_y(2)) = \dots = y_d(-1) = z(-1) + w(-1) = 0$, es decir, $y_d(k) = 0 \forall k < 0$. Finalmente, se considera que los retardos de control y observación son variantes en el tiempo y cumplen con:

$$\begin{aligned} \tau_u(k) &= T_u + \Delta\tau_u(k), \quad \forall k \in \mathbb{Z}^+ \cup \{0\} \\ |\Delta\tau_u(k)| &\leq \mu_1 < T_u, \quad \mu_1 \in \mathbb{Z}^+, \Delta\tau_u(k) \in \mathbb{Z}, \forall k \in \mathbb{Z}^+ \cup \{0\} \\ \tau_y(k) &= T_y + \Delta\tau_y(k), \quad \forall k \in \mathbb{Z}^+ \cup \{0\} \\ |\Delta\tau_y(k)| &\leq \mu_2 < T_y, \quad \mu_2 \in \mathbb{Z}^+, \Delta\tau_y(k) \in \mathbb{Z}, \forall k \in \mathbb{Z}^+ \cup \{0\} \end{aligned} \quad (1.11)$$

Donde $T_u, T_y \in \mathbb{Z}^+$ se pueden considerar como los valores esperados de $\tau_u(k)$ y $\tau_y(k)$, respectivamente. En la figura 1.3 se muestra el diagrama de bloques de 1.10.

Por otra parte, considerando que los retardos de control y observación, $\tau_u(t_k)$ y $\tau_y(s_k)$, de las definiciones 3 y 4 tienen valores esperados, $E(\tau_u(t_k)) \in \mathbb{R}^+$ y $E(\tau_y(s_k)) \in \mathbb{R}^+$, los cuales pueden ser estimados experimentalmente, entonces T_u y T_y se puede calcular como:

$$\begin{aligned} T_y &= \left\lceil \frac{E(\tau_y(s_k))}{h_m} \right\rceil \\ T_u &= \left\lceil \frac{E(\tau_u(t_k))}{h_m} \right\rceil \end{aligned} \quad (1.12)$$

Donde $\lceil \cdot \rceil : \mathbb{R} \rightarrow \mathbb{Z}$ es la función techo. En 1.10 se toman en consideración al ruido de medición, al muestreador y al retardo de comunicación estocástico, por lo tanto, en el siguiente capítulo se diseña el controlador con base en 1.10 pues este se trata de un modelo de un sistema dinámico al cual se le aplicó un sistema de control sobre redes.

Diseño de un controlador neuronal predictivo

En este capítulo se presenta un controlador neuronal predictivo novedoso para el NCS planteado en la sección 1.5.2. El esquema de control propuesto se diseña considerando tiempo discreto y se divide en un control nominal y en un control compensante o correctivo. El control nominal es obtenido utilizando teoría de sistemas discretos lineales con retardo de tiempo constante, con base en lo planteado en Fridman (2014). El control correctivo, por su parte, se compone de un control *reset*, un control basado en redes neuronales y un sistema de predicción por medio de un modelo oculto de Markov. El control nominal se plantea en la sección 2.1, mientras que el correctivo en las secciones 2.2, 2.3 y 2.4.

El esquema de control propuesto en este capítulo toma como principal referencia a Cho and Fadali (2011), pues ahí el controlador también se divide en un control nominal y en un correctivo, además, este último también se compone de un control *reset*, un control basado en redes neuronales y un sistema de predicción. Sin embargo, en Cho and Fadali (2011) se considera tiempo continuo, su control nominal no considera al retardo y no se utiliza un modelo oculto de Markov para la predicción de la salida del sistema, sino otro modelo probabilístico gráfico.

2.1. Control nominal

El control nominal se diseña para un valor constante del retardo de comunicación y se plantea con base en Fridman (2014). Considere el sistema no lineal a controlar 1.1; en el contexto de sistemas determinísticos, el objetivo de control sería que el origen de 1.1 sea asintóticamente estable en el sentido de Lyapunov, sin embargo, debido a la presencia de procesos estocásticos en el lazo de control, incluyendo retardos de tiempo, la demostración rigurosa sería difícil de abordar. Por lo tanto, el objetivo del controlador

neuronal predictivo pasa a ser que si el sistema 1.1 tiene una condición inicial $x(0) \neq \bar{0}$ se cumpla que $x(t)$ se aproxime y permanezca cerca de cero. Asimismo, se busca mejorar el desempeño en comparación de los siguientes controladores: linealización exacta, retroalimentación lineal de estados y control de sistemas lineales discretos con retardo de tiempo constante.

Ahora considere el NCS SISO en tiempo discreto 1.10, es decir, la aproximación del sistema a controlar 1.1 tomando en cuenta el periodo de muestreo, el retardo de comunicación estocástico y el ruido de medición. El primer paso del controlador neuronal predictivo es expresar la dinámica y la salida de 1.10 como la suma de una dinámica lineal y una no lineal de acuerdo a 2.1.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k - \tau_u(k)) + \Delta f(x(k)) \\ y_d(k) &= Cx(k) + \Delta h(x(k)) + w(k) \\ y(k) &= y_d(k - \tau_y(k)) \end{aligned} \quad (2.1)$$

Donde $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times 1}$ y $C \in \mathbb{R}^{1 \times n}$, mientras que $\Delta f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ y $\Delta h : \mathbb{R}^n \rightarrow \mathbb{R}$ son funciones no lineales. En el caso de que 1.10 sea un sistema lineal entonces $\Delta f(x(k)) = 0$ y $\Delta h(x(k)) = 0$ y el controlador neuronal predictivo se puede implementar sin ningún problema. Solo en el caso de que para un sistema en específico no se pueda hacer esta separación, entonces para ese sistemas no es posible implementar el control nominal ni el control *reset* y se debe pasar directamente al control neuronal (sección 2.3) y al sistema de predicción (sección 2.4). Con base en 1.11, 2.1 puede ser reescrito como:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k - T_u) + \Delta f_2(x(k), \Delta \tau_u(k)) \\ y_d(k) &= Cx(k) + \Delta h(x(k)) + w(k) \\ y(k) &= y_d(k - T_y) + \Delta y_d(\Delta \tau_y(k)) \end{aligned} \quad (2.2)$$

Donde Δy_d y Δf_2 son funciones no lineales. Con base en 2.2 la ecuación 2.3 define al sistema nominal.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu_{nom_1}(k - T_u) \\ y(k) &= Cx(k - T_y), \end{aligned} \quad (2.3)$$

Donde $u_{nom_1}(k)$ es la señal de control nominal en el tiempo k la cual es calculada considerando únicamente al sistema nominal 2.3. El objetivo del control $u_{nom_1}(k)$ es hacer que el origen de 2.3 sea asintóticamente estable. Observando 2.2 se aprecia como Δf_2 , Δh , Δy_d y $w(k)$ corresponden a perturbaciones no lineales del sistema nominal y que resultan de los retardos de tiempo variables e inciertos $\Delta \tau_u$ y $\Delta \tau_y$, de las no linealidades del sistema y del ruido de medición. En otras palabras, el NCS de 1.10 es separado en un sistema nominal lineal con retardo de comunicación constante y una perturbación no lineal que incluye un retardo de tiempo variable e incierto.

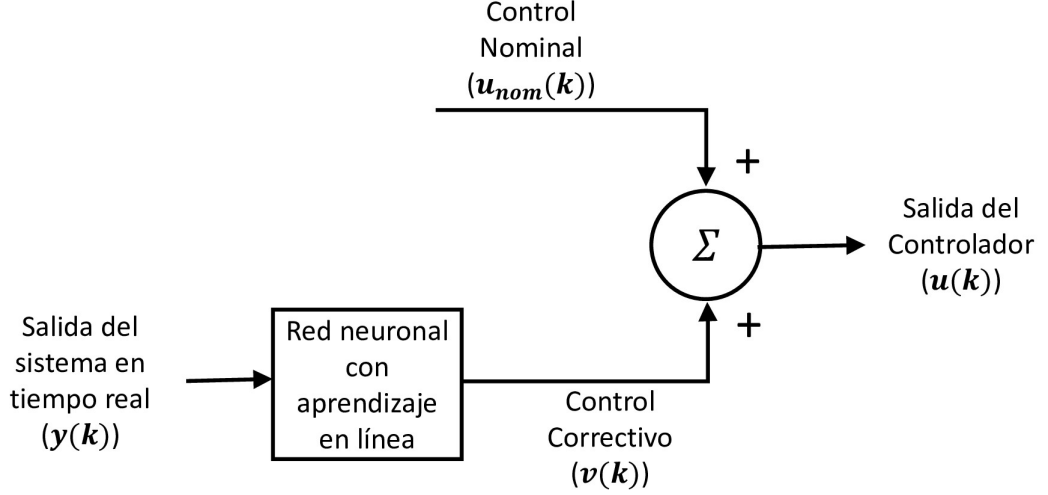


Figura 2.1: Configuración de la señal de control para un NCS.

De acuerdo a esta separación, basada en [Cho and Lee \(2008\)](#), se adopta un procedimiento de diseño del controlador en dos pasos. Primero se va a diseñar un control de retroalimentación de estados para el sistema nominal y posteriormente se va a diseñar un control correctivo con aprendizaje en línea el cual va a corregir de manera adaptable la señal de control nominal por medio de una red neuronal. Por lo tanto, la señal de control $u(k)$ del sistema original 2.1 es la suma de un control nominal ($u_{nom}(k)$) y del control correctivo con aprendizaje en línea ($v(k)$) de acuerdo a 2.4 y como ilustra la figura 2.1.

$$u(k) = u_{nom}(k) + v(k) \quad (2.4)$$

Sustituyendo el control 2.4 en el sistema original 2.1 el sistema en lazo cerrado queda:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu_{nom}(k - \tau_u(k)) + Bv(k - \tau_u(k)) + \Delta f(x(k)) \\ y(k) &= Cx(k - \tau_y(k)) + \Delta h(x(k - \tau_y(k))) + w(k - \tau_y(k)) \end{aligned} \quad (2.5)$$

Por otra parte, sustituyendo el control 2.4 en 2.2 el sistema en lazo cerrado queda:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu_{nom}(k - T_u) + Bv(k - T_u) + \Delta f_2(x(k), \Delta \tau_u(k)) \\ y(k) &= Cx(k - T_y) + \Delta h(x(k - T_y)) + w(k - T_y) + \Delta z(\Delta \tau_y(k)) \end{aligned} \quad (2.6)$$

En 2.6 se aprecia como $u_{nom}(k)$ puede ser diseñado para estabilizar asintóticamente al sistema nominal, mientras que $v(k)$ es una señal de control que se diseña para contrarrestar a las perturbaciones del modelo nominal.

Empezando con el control que estabiliza asintóticamente al sistema nominal 2.3, el cual está basado en Fridman (2014), considere que el par (A, C) es observable y el par (A, B) es estabilizable para $T_u = T_y = 0$ y defina el siguiente estado aumentado:

$$\tilde{x}(k) = [x(k - T_y) \quad u_{nom_1}(k - 1) \quad u_{nom_1}(k - 2) \quad \dots \quad u_{nom_1}(k - T_\eta)]^T \in \mathbb{R}^{n+T_\eta} \quad (2.7)$$

Donde:

$$T_\eta = T_y + T_u \quad (2.8)$$

Con el estado aumentado 2.7 se obtiene el siguiente sistema aumentado equivalente al sistema nominal 2.3 pero sin retardo:

$$\begin{aligned} \tilde{x}(k + 1) &= \tilde{A}\tilde{x}(k) + \tilde{B}u_{nom_1}(k) \\ \tilde{y}(k) &= \tilde{C}\tilde{x}(k) \end{aligned} \quad (2.9)$$

Donde:

$$\begin{aligned} \tilde{A} &= \begin{bmatrix} A & \bar{0} & \bar{0} & \dots & \bar{0} & B \\ \bar{0}^T & 0 & 0 & \dots & 0 & 0 \\ \bar{0}^T & 1 & 0 & \dots & 0 & 0 \\ \bar{0}^T & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \bar{0}^T & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{(n+T_\eta) \times (n+T_\eta)} \\ \bar{0} &= [0 \dots 0]^T \in \mathbb{R}^n \\ \tilde{B} &= [\bar{0} \quad 1 \quad 0 \quad 0 \quad \dots \quad 0]^T \in \mathbb{R}^{n+T_\eta} \\ \tilde{C} &= [C \quad 0 \quad 0 \quad \dots \quad 0] \in \mathbb{R}^{1 \times (n+T_\eta)} \end{aligned} \quad (2.10)$$

Note que se cumple:

$$\tilde{y}(k) = Cx(k - T_y) = y(k) \quad (2.11)$$

Es decir, la salida del estado aumentado 2.9 es la misma que la salida del sistema nominal 2.3. Defina el siguiente estado aumentado estimado:

$$\hat{\tilde{x}}(k) = [\hat{x}(k - T_y) \quad u_{nom_1}(k - 1) \quad u_{nom_1}(k - 2) \quad \dots \quad u_{nom_1}(k - T_\eta)]^T \in \mathbb{R}^{n+T_\eta} \quad (2.12)$$

Utilizando al observador de Luenberger para estimar $\hat{\tilde{x}}(k)$:

$$\begin{aligned} \hat{\tilde{x}}(k + 1) &= \tilde{A}\hat{\tilde{x}}(k) + L [\tilde{y}(k) - \hat{\tilde{y}}(k)] + \tilde{B}u_{nom_1}(k) \\ &= \tilde{A}\hat{\tilde{x}}(k) + L [\tilde{C}\hat{\tilde{x}}(k) - \tilde{C}\hat{\tilde{x}}(k)] + \tilde{B}u_{nom_1}(k) \end{aligned} \quad (2.13)$$

Donde $L \in \mathbb{R}^{n+T_\eta}$ es una constante de diseño del observador. Definiendo al error de observación:

$$e_o(k) = \tilde{x}(k) - \hat{\tilde{x}}(k) \quad (2.14)$$

Entonces se tiene:

$$\begin{aligned} e_o(k+1) &= \tilde{x}(k+1) - \hat{\tilde{x}}(k+1) \\ &= \tilde{A}\tilde{x}(k) + \tilde{B}u_{nom_1}(k) - \tilde{A}\hat{\tilde{x}}(k) - L\tilde{C}(\tilde{x}(k) - \hat{\tilde{x}}(k)) - \tilde{B}u_{nom_1}(k) \\ &= (\tilde{A} - L\tilde{C})e_o(k) \end{aligned} \quad (2.15)$$

Considere la siguiente ley de control:

$$\begin{aligned} u_{nom_1}(k) &= -K\hat{\tilde{x}}(k) \\ &= -K(\tilde{x}(k) - e_o(k)) \end{aligned} \quad (2.16)$$

Donde $K \in \mathbb{R}^{1 \times (n+T_\eta)}$ es una constante de diseño del controlador. Sustituyendo 2.16 en 2.9:

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{A}\tilde{x}(k) - \tilde{B}K(\tilde{x}(k) - e_o(k)) \\ &= (\tilde{A} - \tilde{B}K)\tilde{x}(k) + \tilde{B}Ke_o(k) \end{aligned} \quad (2.17)$$

De 2.15 y 2.17 se cumple:

$$\begin{aligned} \begin{bmatrix} \tilde{x}(k+1) \\ e_o(k+1) \end{bmatrix} &= \begin{bmatrix} \tilde{A} - \tilde{B}K & \tilde{B}K \\ \bar{0} & \tilde{A} - L\tilde{C} \end{bmatrix} \begin{bmatrix} \tilde{x}(k) \\ e_o(k) \end{bmatrix} \\ &= A_{aum} \begin{bmatrix} \tilde{x}(k) \\ e_o(k) \end{bmatrix} \end{aligned} \quad (2.18)$$

Para que $\tilde{x}(k) \rightarrow \bar{0}$ y $e_o(k) \rightarrow \bar{0}$, A_{aum} debe ser Schur, esto es, que el módulo de sus valores propios sea menor a uno. Debido a que A_{aum} es una matriz triangular sus valores propios son los de $\tilde{A} - \tilde{B}K$ junto con los de $\tilde{A} - L\tilde{C}$, por lo tanto, se puede escoger de manera independiente K y L tal que $\tilde{A} - \tilde{B}K$ y $\tilde{A} - L\tilde{C}$ sean Schur, en otras palabras se cumple con el principio de separación. También sea puede comprobar que si el par (A, C) es observable y el par (A, B) es estabilizable para $T_\eta = 0$, entonces el par (\tilde{A}, \tilde{C}) es observable y el par (\tilde{A}, \tilde{B}) es estabilizable. Por otra parte, si $\tilde{x}(k) \rightarrow \bar{0}$ también $x(k) \rightarrow \bar{0}$ del sistema nominal 2.3. Por lo tanto, el control que estabiliza asintóticamente a 2.3 es 2.16. Expresando 2.16 de forma diferente:

$$\begin{aligned} u_{nom_1}(k) &= -K\hat{\tilde{x}}(k) \\ &= -K_2\hat{\tilde{x}}(k - T_y) - K_3 \begin{bmatrix} u_{nom_1}(k-1) \\ u_{nom_1}(k-2) \\ \vdots \\ u_{nom_1}(k - T_\eta) \end{bmatrix} \end{aligned} \quad (2.19)$$

Donde $K = [K_2 K_3]$, $K_2 \in \mathbb{R}^{1 \times n}$, $K_3 \in \mathbb{R}^{1 \times T_\eta}$ y $\hat{x}(k - T_y)$ es la salida del observador de Luenberger 2.13. Note que las entradas a este observador son la salida $\tilde{y}(k) = y(k)$ y la entrada del $u_{nom_1}(k)$ del sistema nominal 2.3, por lo tanto, se puede considerar que el observador de Luenberger 2.13 puede ser representado por una función $F_o : \mathbb{R}^2 \rightarrow \mathbb{R}^n$ que cumple con:

$$\hat{x}(k - T_y) = F_o(y(k), u_{nom_1}(k)) = F_o(Cx(k - T_y), u_{nom_1}(k)) \quad (2.20)$$

Con base en 2.19 el control $u_{nom}(k)$ de 2.4 es:

$$u_{nom}(k) = -K_2 \hat{x}(k - \tau_y(k)) - K_3 \begin{bmatrix} u_{nom}(k - 1) \\ u_{nom}(k - 2) \\ \vdots \\ u_{nom}(k - T_\eta) \end{bmatrix} \quad (2.21)$$

Donde $\hat{x}(k - \tau_y(k))$ es calculado a partir de la salida del sistema original 2.5 ($y(k) = Cx(k - \tau_y(k)) + \Delta h(x(k - \tau_y(k))) + w(k - \tau_y(k))$) y $u_{nom}(k)$ (2.21) de acuerdo a la función F_o de la ecuación 2.20, es decir:

$$\hat{x}(k - \tau_y(k)) = F_o(y(k), u_{nom}(k)) = F_o\left(Cx(k - \tau_y(k)) + \Delta h(x(k - \tau_y(k))) + w(k - \tau_y(k)), u_{nom}(k)\right) \quad (2.22)$$

Con el control 2.21 el sistema original 2.5 en lazo cerrado queda:

$$\begin{aligned} x(k + 1) &= Ax(k) - BK_2 \hat{x}(k - \tau_\eta(k)) - BK_3 \begin{bmatrix} u_{nom}(k - 1 - \tau_u(k)) \\ u_{nom}(k - 2 - \tau_u(k)) \\ \vdots \\ u_{nom}(k - T_\eta - \tau_u(k)) \end{bmatrix} + Bv(k - \tau_u(k)) + \Delta f(x(k)) \\ y(k) &= Cx(k - \tau_y(k)) + \Delta h(x(k - \tau_y(k))) + w(k - \tau_y(k)) \end{aligned} \quad (2.23)$$

Donde $\tau_\eta(k) = \tau_y(k) + \tau_u(k)$. El siguiente paso en el diseño del control es diseñar el control correctivo con aprendizaje en línea $v(k)$.

2.2. Sistema de control basado en *reset*

El control $v(k)$ de 2.4 se diseña para generar una señal que corrija de manera adaptable la señal del control nominal 2.21 y así mejorar el desempeño del sistema de control, por lo tanto llamaremos a $v(k)$ como el control correctivo. El control correctivo que se presenta en esta tesis está basado en el correctivo planteado en [Cho and Fadali \(2011\)](#), sin embargo, ahí se plantea en tiempo continuo, por lo tanto, en el presente trabajo se presenta un equivalente en tiempo discreto.

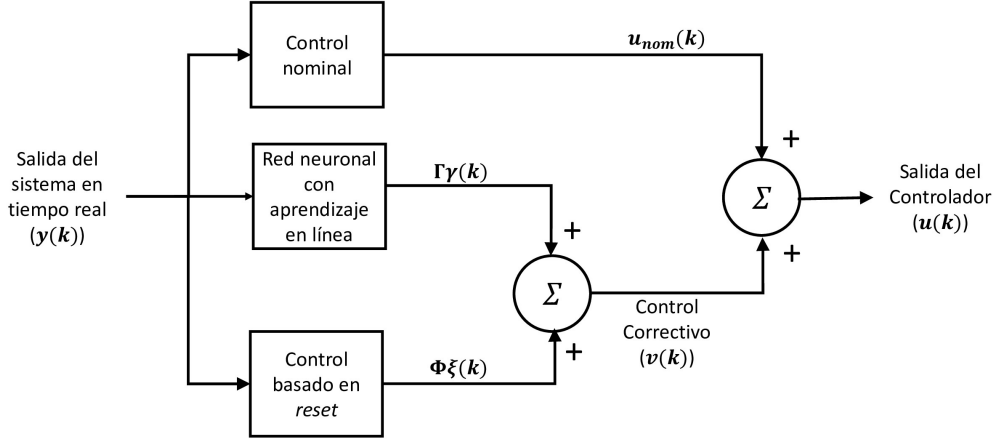


Figura 2.2: Configuración de la señal de control para un NCS.

Debido a las perturbaciones del sistema nominal 2.3, el desempeño del controlador puede verse significativamente afectado si solo se tiene al control nominal, especialmente debido a las perturbaciones provocadas por el retardo de comunicación variable e incierto. Debido a que el objetivo del controlador neuronal predictivo es lograr estabilizar al sistema con un buen desempeño, el objetivo del control $v(k)$ es mejorar el desempeño (ya sea en estado estacionario o transitorio) en comparación a simplemente tener al control nominal 2.21.

El control $v(k)$ se compone a su vez de dos controles: un control estático $\Phi\xi(k) \in \mathbb{R}$ y un control con aprendizaje en línea (adaptable) $\Gamma\gamma(k) \in \mathbb{R}$; este último se logra por medio de una red neuronal. Más específicamente $v(k)$ de 2.4 está dado por:

$$v(k) = \Phi\xi(k) + \Gamma\gamma(k) \quad (2.24)$$

Donde $\xi(k) \in \mathbb{R}$ es un control conocido como control basado en *reset* o simplemente como control *reset*, $\gamma(k) \in \mathbb{R}^L$ es la salida de una capa de L perceptrones y $\Phi \in \mathbb{R}$ y $\Gamma \in \mathbb{R}^{1 \times L}$ son constantes de diseño del controlador. La figura 2.2 ilustra a 2.24 y es una versión más completa de la figura 2.1. Sustituyendo 2.24 en 2.23:

$$x(k+1) = Ax(k) - BK_2\hat{x}(k - \tau_\eta(k)) - BK_3 \begin{bmatrix} u_{nom}(k-1 - \tau_u(k)) \\ u_{nom}(k-2 - \tau_u(k)) \\ \vdots \\ u_{nom}(k - T_\eta - \tau_u(k)) \end{bmatrix} + \quad (2.25)$$

$$B\Gamma\gamma(k - \tau_u(k)) + B\Phi\xi(k - \tau_u(k)) + \Delta f(x(k))$$

$$y(k) = Cx(k - \tau_y(k)) + \Delta h(x(k - \tau_y(k))) + w(k - \tau_y(k))$$

De acuerdo con [Cho and Fadali \(2011\)](#) el control *reset* sirve para contrarrestar a las perturbaciones del sistema nominal resultantes de la linealización ($\Delta f(x(k))$), mientras que el control con aprendizaje en línea sirve para contrarrestar de manera adaptable a las perturbaciones provocadas por el retardo de tiempo estocástico. En esta sección se plantea el control *reset* ($\Phi\xi(k)$) mientras que el planteamiento de la red neuronal se encuentra en la sección [2.3](#). En [Cho and Fadali \(2011\)](#) el control *reset* viene dado por:

$$\begin{cases} \dot{\xi}(t) = \phi\xi(t) + e(t), & \text{si } e(t) \neq 0 \\ \xi(t^+) = 0, & \text{si } e(t) = 0 \end{cases} \quad (2.26)$$

Donde, $\phi \in \mathbb{R}$ es una constante de diseño del controlador y $e(t)$ es definido como la desviación entre un valor deseado de referencia $r(t)$ y la salida con retardo de observación $y(t)$ del sistema original, es decir:

$$e(t) = r(t) - y(t) \quad (2.27)$$

A la ecuación [2.26](#) se le conoce como el elemento reseteado de primer orden (First Order Reset Element (FORE)). La primera ecuación representa el modo continuo, mientras que la segunda el modo discreto o de reseteo que se dispara en los cruces por cero de la señal de error ($e(t) = 0$) y provoca un reinicio a cero $\xi(t^+) = 0$ y la consiguiente discontinuidad de salto, $\xi(t^+) \neq \xi(t^-)$. Para plantear el equivalente de [2.27](#) en tiempo discreto se propone utilizar el método de Euler, el cual se define a continuación:

Definición 7. Sea $\dot{y}(t) = f(t, y(t))$, $y(t_0) = y_0$ una ecuación diferencial ordinaria. Sea $t_k = t_0 + kh_m$, donde $k \in \mathbb{Z}^+ \cup \{0\}$ y $h_m \in \mathbb{R}^+$ es el tamaño del paso de tiempo de t_k a $t_{k+1} = t_k + h = t_0 + (k+1)h_m \forall k \in \mathbb{Z}^+ \cup \{0\}$. Sea $y_e(k)$ la aproximación de Euler de la solución de $\dot{y}(t) = f(t, y(t))$, $y(t_0) = y_0$ en el tiempo t_k , es decir, $y_e(k) \approx y(t_k) \forall k \in \mathbb{Z}^+$. Sea $y_e(0) = y(t_0) = y_0$. Entonces: $y_e(k+1) = y_e(k) + h_m f(t, y(t))|_{t=t_k, y(t)=y_e(k)} \forall k \in \mathbb{Z}^+$.

Utilizando la definición [7](#), la aproximación de Euler de la primera ecuación de [2.26](#) es:

$$\begin{aligned} \xi(k+1) &= \xi(k) + h_m(\phi\xi(t) + e(t))|_{t=t_k, \xi(t)=\xi(k)} \\ &= \xi(k) + h_m(\phi\xi(k) + e(k)) \\ &= (1 + h_m\phi)\xi(k) + h_me(k) \end{aligned} \quad (2.28)$$

Donde $h_m \in \mathbb{R}^+$ es el periodo de muestreo utilizado para obtener el modelo [1.9](#) y para $e(t = t_k)$ se utiliza un equivalente en tiempo discreto de [2.27](#), es decir, $e(k)$. En la presente tesis, debido a que se busca el objetivo de estabilización $e(k)$ se define como:

$$e(k) = -y(k) \quad (2.29)$$

Con base en 2.26 y 2.28 el control basado en *reset* $\xi(k)$ del control correctivo $v(k)$ (2.24) se define como:

$$\begin{cases} \xi(k) = (1 + h_m\phi)\xi(k-1) + h_me(k-1), & \text{si } e(k) \neq 0 \\ \xi(k) = 0, & \text{si } e(k) = 0 \end{cases} \quad (2.30)$$

Como condición inicial para $\xi(k)$ se utiliza $\xi(0) = 0$. Note en 2.30 que $\xi(k)$ sigue teniendo el comportamiento de *reset* deseado cuando la señal de error cruza por cero. Con base en 2.24 y 2.30 para que el control $\Phi\xi(k)$ quede definido solo hace falta dar un valor a las constantes Φ y ϕ . En Cho and Fadali (2011) las constantes Φ y ϕ se escogen considerando al sistema 2.31 (el equivalente en tiempo continuo). Comparando 2.25 con 2.31 se aprecia como en Cho and Fadali (2011) proponen diseñar el control basado en *reset* considerando $x(k) = \hat{x}(k)$ (lo cual es razonable por el diseño del observador Luenberger 2.13) y sin considerar al retardo de comunicación ni a las no linealidades, lo cual se busca compensar con la adición del control basado en redes neuronales.

$$\begin{aligned} x(k+1) &= (A - BK_2)x(k) + B\Phi\xi(k) \\ y(k) &= Cx(k). \end{aligned} \quad (2.31)$$

Sustituyendo 2.29 en 2.30:

$$\begin{cases} \xi(k) = (1 + h_m\phi)\xi(k-1) - h_my(k-1), & \text{si } e(k) \neq 0 \\ \xi(k) = 0, & \text{si } e(k) = 0 \end{cases} \quad (2.32)$$

Sustituyendo la segunda ecuación de 2.31 en 2.32:

$$\begin{cases} \xi(k) = (1 + h_m\phi)\xi(k-1) - h_mCx(k-1), & \text{si } e(k) \neq 0 \\ \xi(k) = 0, & \text{si } e(k) = 0 \end{cases} \quad (2.33)$$

De 2.33 (con $e(k) \neq 0$) y la primera ecuación de 2.31 se obtiene el siguiente sistema aumentado:

$$\begin{aligned} x(k+1) &= (A - BK_2)x(k) + B\Phi\xi(k) \\ \xi(k+1) &= (1 + h_m\phi)\xi(k) - h_mCx(k) \end{aligned} \quad (2.34)$$

Expresando 2.34 de forma matricial:

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ \xi(k+1) \end{bmatrix} &= \begin{bmatrix} A - BK_2 & B\Phi \\ -h_mC & 1 + h_m\phi \end{bmatrix} \begin{bmatrix} x(k) \\ \xi(k) \end{bmatrix} \\ &= A_{aum2} \begin{bmatrix} x(k) \\ \xi(k) \end{bmatrix} \end{aligned} \quad (2.35)$$

Finalmente, para que 2.35 sea asintóticamente estable se necesitan escoger los parámetros del controlador Φ y ϕ de tal manera que $A_{aum2} \in \mathbb{R}^{(n+1) \times (n+1)}$ sea Schur. Incluso se puede modificar K_2 si es necesario, siempre y cuando $\tilde{A} - \tilde{B}K$ de 2.10 siga siendo Schur. En la práctica debido a que la señal que llega al controlador es discontinua, de

hecho es una sucesión de puntos aislados, por lo que tampoco es continua a trazos, se puede dar con facilidad el caso de que la señal de error cruce por cero y no detectarlo, además, debido al ruido de medición se vuelve aún más complicado detectar cuando la señal de error cruza por cero. Por lo tanto se propone la siguiente modificación a 2.30:

$$\begin{cases} \xi(k) = (1 + h_m\phi)\xi(k-1) + h_me(k-1), & \text{si } |e(k)| > q \\ \xi(k) = 0, & \text{si } |e(k)| \leq q \end{cases} \quad (2.36)$$

Donde $q \in \mathbb{R}^+$ es un parámetro a escoger cercano a cero. Una vez que se terminó de diseñar el control $\Phi\xi(k)$ en la siguiente sección se diseña el control basado en redes neuronales ($\gamma(k)$) del control correctivo $v(k)$ (2.24).

2.3. Control basado en redes neuronales

Se ha demostrado que un controlador basado en redes neuronales, una vez que pasa la etapa de aprendizaje, provee una señal adecuada con fines de control. En este caso se busca plantear una ley de control adaptable utilizando redes neuronales para corregir el desempeño del controlador nominal 2.21. Al igual que con el control *reset* ($\Phi\xi(k)$), en esta sección se plantea un equivalente en tiempo discreto del controlador neuronal planteado en Cho and Fadali (2011), en donde se plantea en tiempo continuo. Más específicamente, en Cho and Fadali (2011) el control basado en redes neuronales se propone como la salida de una red neuronal de una capa de L perceptrones. En Cho and Lee (2008) también usan un control basado en una red neuronal, sin embargo, esta es una red neuronal recurrente.

Cada componente del vector $\gamma(k) \in \mathbb{R}^L$ de 2.24, $\gamma_i(k) \in \mathbb{R}$, $i = 1 \dots L$, es la salida de un perceptrón que en su conjunto forman a una red neuronal de una capa, por lo tanto, cada $\gamma_i(k)$ es gobernada por:

$$\gamma_i(k) = \varphi_i(W_i(k)\sigma(k)), \quad i = 1 \dots L \quad (2.37)$$

Donde $\sigma(k)$ es el vector de entrada a la red neuronal y se define como:

$$\sigma(k) = \left[e(k) \quad \frac{e(k) - e(k-1)}{h_m} \right]^T \in \mathbb{R}^2 \quad (2.38)$$

Donde $e(k)$ es de acuerdo a 2.29, $e(0) = 0$ y h_m es el periodo de muestreo utilizado para obtener el modelo 1.9. Por otra parte, en la ecuación 2.37, $W_i(k) = [w_{i1}(k) \quad w_{i2}(k)] \in \mathbb{R}^{1 \times 2}$ es el renglón i de la matriz de parámetros a aprender de la red neuronal ($W(k) = \{w_{ij}(k)\} \in \mathbb{R}^{L \times 2}$) y $\varphi_i : \mathbb{R} \rightarrow \mathbb{R}$ es una función sigmoide no lineal definida como:

$$\varphi_i(\cdot) = a_i \tanh(b_i \cdot) \quad (2.39)$$

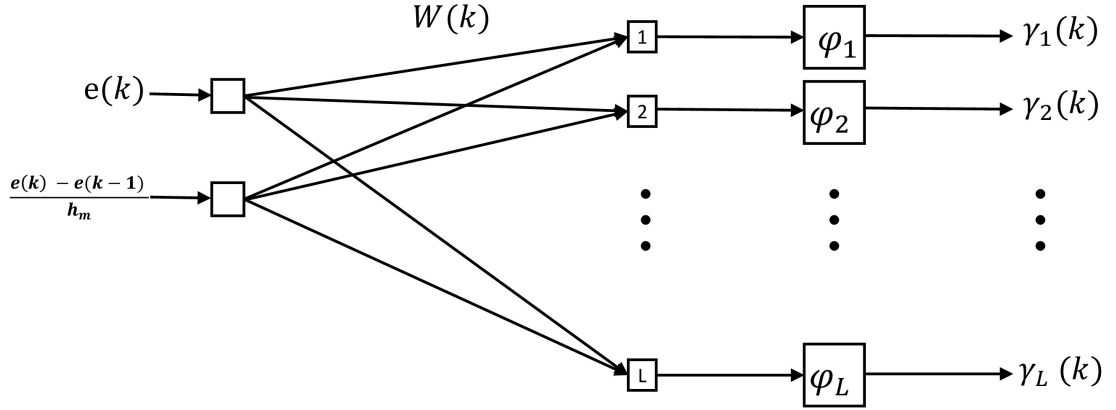


Figura 2.3: Modelo de una red neuronal de una sola capa

Con las constantes de escalamiento $a_i, b_i \in \mathbb{R}$. La red neuronal de una sola capa propuesta se muestra en la figura 2.3.

Como marco de referencia, en [Cho and Fadali \(2011\)](#) el vector de entrada a la red neuronal ($\sigma(t)$) se propone como:

$$\sigma(t) = [1 \quad e(t) \quad \dot{e}(t)]^T \in \mathbb{R}^3 \quad (2.40)$$

Donde $e(t)$ es el error de seguimiento 2.27. Sin embargo, note que con este vector de entrada la salida de la red neuronal no es cero a pesar de que el error sea cero, debido el objetivo de control es la estabilización y el origen del espacio de estados en un punto de equilibrio entonces esto es contraproducente. Por lo tanto, en [Cho and Fadali \(2011\)](#) $w_{1i} = 1$ se utiliza con fines de seguimiento. Por las razones anteriores se propuso el vector de entrada a la red 2.38, en el cual se considera tiempo discreto y es muy similar al propuesto en [Cho and Fadali \(2011\)](#).

Continuando con el diseño del controlador neuronal, la matriz $W(k)$ óptima que minimiza una función de costo $J(k)$ se determina iterativamente usando el método del descenso por gradiente. Debido a que el objetivo de control es la estabilización, la función objetivo $J(k)$ se escoge como:

$$J(k) = \frac{1}{2} e^2(k). \quad (2.41)$$

Debido a que $J(k)$ es función de $e(k)$, $e(k)$ es función de $\gamma(k)$ y $\gamma(k)$ es función de $W(k)$, se puede considerar que $J(k)$ es función de los pesos de cada perceptrón de la red, es decir, $J(k)$ es función del renglón i de $W(k)$ ($W_i(k)$). Debido a que el gradiente apunta en la dirección de máximo crecimiento de una función, el método del descenso por gradiente establece que para minimizar la función $J(k)$ se debe actualizar $W_i(k)$

2. DISEÑO DE UN CONTROLADOR NEURONAL PREDICTIVO

en la dirección opuesta al gradiente $\frac{\partial J(k)}{\partial W_i(k)} \in \mathbb{R}^{1 \times 2}$, es decir, se debe seguir la siguiente regla de aprendizaje.

$$W_i(k+1) = W_i(k) - \eta_i(k) \frac{\partial J(k)}{\partial W_i(k)}, \quad i = 1 \dots L \quad (2.42)$$

Donde $\eta_i(k) \in (0, 1)$ es un parámetro de aprendizaje conocido como la tasa de aprendizaje y es muy importante pues de este valor depende que tanto cambian los parámetros $W_i(k)$ en cada iteración. La derivada parcial en 2.42 es expandida usando la regla de la cadena como:

$$\begin{aligned} \frac{\partial J(k)}{\partial W_i(k)} &= \left[\frac{\partial J(k)}{\partial w_{i1}(k)} \quad \frac{\partial J(k)}{\partial w_{i2}(k)} \right] \\ &= \left[\frac{\partial J(k)}{\partial e(k)} \frac{\partial e(k)}{\partial y(k)} \frac{\partial y(k)}{\partial \gamma_i(k)} \frac{\partial \gamma_i(k)}{\partial \psi_i(k)} \frac{\partial \psi_i(k)}{\partial w_{i1}(k)} \quad \frac{\partial J(k)}{\partial e(k)} \frac{\partial e(k)}{\partial y(k)} \frac{\partial y(k)}{\partial \gamma_i(k)} \frac{\partial \gamma_i(k)}{\partial \psi_i(k)} \frac{\partial \psi_i(k)}{\partial w_{i2}(k)} \right] \\ &= \frac{\partial J(k)}{\partial e(k)} \frac{\partial e(k)}{\partial y(k)} \frac{\partial y(k)}{\partial \gamma_i(k)} \frac{\partial \gamma_i(k)}{\partial \psi_i(k)} \left[\frac{\partial \psi_i(k)}{\partial w_{i1}(k)} \quad \frac{\partial \psi_i(k)}{\partial w_{i2}(k)} \right] \end{aligned} \quad (2.43)$$

Donde:

$$\begin{aligned} \frac{\partial J(k)}{\partial e(k)} &= e(k) = -y(k) \\ \frac{\partial e(k)}{\partial y(k)} &= -1 \\ \frac{\partial y(k)}{\partial \gamma_i(k)} &\approx \frac{y(k) - y(k-1)}{\gamma_i(k) - \gamma_i(k-1)} \\ \psi_i(k) &= W_i(k) \sigma(k) \in \mathbb{R} \\ \frac{\partial \gamma_i(k)}{\partial \psi_i(k)} &= a_i b_i (1 - \tanh^2(b_i \psi_i(k))) \\ \frac{\partial \psi_i(k)}{\partial w_{i1}(k)} &= e(k) \\ \frac{\partial \psi_i(k)}{\partial w_{i2}(k)} &= \frac{e(k) - e(k-1)}{h_m} \end{aligned} \quad (2.44)$$

Donde la tercera ecuación de 2.44 es una aproximación usada con frecuencia. Substituyendo 2.44 en 2.42 se obtiene:

$$W_i(k+1) = W_i(k) - \eta_i(k) \left(y(k) \left(\frac{y(k) - y(k-1)}{\gamma_i(k) - \gamma_i(k-1)} \right) a_i b_i \left(1 - \tanh^2(b_i W_i(k) \sigma(k)) \right) \right) \sigma^T(k) \quad (2.45)$$

Donde $\eta_i(k)$ se obtiene de la siguiente regla de aprendizaje:

$$\eta_i(k) = \eta_i(k-1) - \epsilon(e(k)) \eta_i^2(k-1) \quad (2.46)$$

Donde $\epsilon : \mathbb{R} \rightarrow \mathbb{R}^+ \cup \{0\}$ es una función definida positiva. Con la ecuación 2.46 termina el planteamiento del controlador basado en redes neuronales $\gamma(k) \in \mathbb{R}^L$ (ecuación 2.37) y, por lo tanto, termina el planteamiento del control correctivo 2.24, el cual se adiciona al controlador nominal 2.21 como está planteado en 2.4 y en la figura 2.2. Con la adición del control basado en redes neuronales se busca que el sistema de control sea más robusto a perturbaciones, incluyendo a las variaciones del retardo de tiempo, y presente una mejor respuesta transitoria, lo cual se puede lograr gracias a la adaptación los parámetros.

2.4. Sistema de predicción basado en un modelo oculto de Markov

Debido al retardo de tiempo, el controlador no cuenta con la salida medida del sistema más reciente, por lo tanto, la señal de control que se generaría con el control propuesto hasta el momento sería una señal que no va de acuerdo a la situación actual del sistema, lo cual degradaría el desempeño. Por lo tanto, en esta sección se presenta un modelo oculto de Markov (*Hidden Markov Model* (HMM)) para predecir la salida del sistema 1.10 tomando en cuenta la naturaleza estocástica de este y con base en esta predicción se plantea un control predictivo.

Utilizando la ley de control predictivo no se generará una señal de control con información desactualizada, sino que con base en una predicción que se asemeja más a la actualidad del sistema y por lo tanto, el desempeño del controlador mejorará. En [Cho and Lee \(2008\)](#) y [Cho and Fadali \(2011\)](#) se hacen planteamientos similares, sin embargo, en el primero se usa una cadena de Markov y en el segundo un modelo oculto de Markov entrada-salida (*Input/output HMM*).

Un HMM es una gráfica acíclica dirigida en la que los nodos representan variables aleatorias y los bordes representan probabilidades condicionales entre dichas variables. Además, en un HMM se cumple la condición de Markov, la cual plantea independencias condicionales entre las variables aleatorias, más específicamente ([Theodoridis \(2015\)](#)): *Cada variable aleatoria (nodo) es condicionalmente independiente del conjunto de todos sus no descendientes dado el conjunto de todos sus padres*. Por estas razones se dice que un HMM es una forma de representar gráficamente independencias y dependencias condicionales o estadísticas entre varias variables aleatorias.

Además, en un HMM algunos bordes dirigidos se pueden interpretar como una secuencia de eventos o estados que experimenta una variable aleatoria a lo largo del tiempo. Por el concepto de evolución en el tiempo de una variable, como se explicará más adelante, en un HMM la condición de Markov se puede leer ([Theodoridis \(2015\)](#)): *El futuro es independiente del pasado dado el presente*, donde el futuro hace referencia a alguna variable aleatoria en un HMM, el pasado a sus no descendientes y el presente a

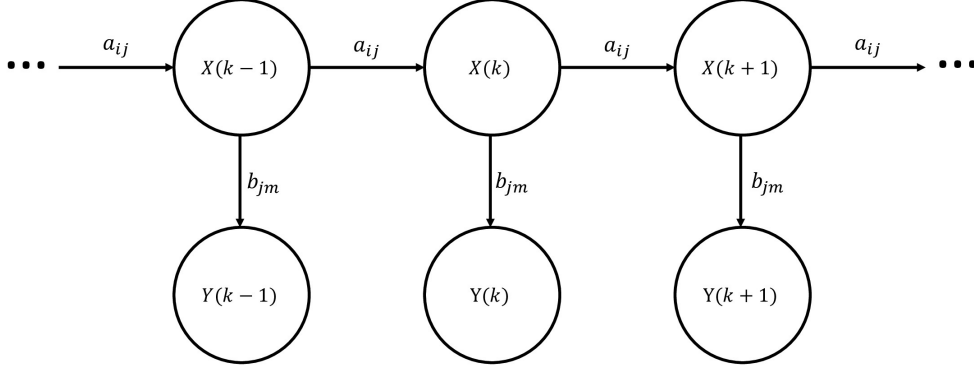


Figura 2.4: Modelo oculto de Markov para un NCS

sus padres. Por lo tanto, la condición de Markov indica un tipo de causalidad y por eso es que un HMM puede ser usado para modelar sistemas dinámicos estocásticos causales, tal como 1.10.

Además de representar un tipo de causalidad (dependencias e independencias condicionales) entre una secuencia de eventos o estados que experimenta una variable aleatoria, un HMM también es útil cuando esta secuencia de eventos permanece oculta, esto es, que no la observamos directamente (Jurafsky and Martin (2008)). Para poder inferir sobre los eventos ocultos en un HMM también hay una secuencia de eventos *observados*, esto es que conocemos su valor. Por lo tanto, se puede decir que un HMM es un modelo probabilístico en el que existen eventos observados y ocultos relacionados de forma causal.

Para empezar con la representación de 1.10 como un HMM, note que la observación $y(k)$ en 1.10 depende de un estado medido anteriormente $x(k - \tau_y(k))$, del retardo de tiempo estocástico $\tau_y(k)$ y del ruido de medición $w(k)$:

$$y(k) = h(x(k - \tau_y(k))) + w(k - \tau_y(k)) \quad (2.47)$$

Debido al retardo estocástico y al ruido de medición es claro que con base en la observación $y(k)$ no es posible conocer $h(x(k - T_y))$, es decir, el estado medido hace T_y periodos de muestreo, por lo tanto, en la presente tesis, con el objetivo de modelar la naturaleza estocástica del retardo de tiempo, se plantea un HMM que relacione $h(x(k - T_y))$ y $y(k)$, donde $y(k)$ es una variable observable y se dice que $h(x(k - T_y))$ es una variable oculta o latente. En la figura 2.4 se muestra el HMM donde las variables continuas $h(x(k - T_y - 1))$, $y(k - 1)$, $h(x(k - T_y))$, $y(k)$, $h(x(k - T_y + 1))$ y $y(k + 1)$ son representadas por las variables aleatorias discretas $X(k - 1)$, $Y(k - 1)$, $X(k)$, $Y(k)$, $X(k + 1)$ y $Y(k + 1)$, respectivamente, $\forall k \in \mathbb{Z}^+$. Se dice que $X(k) \forall k \in \mathbb{Z}^+$ son las variables ocultas y $Y(k) \forall k \in \mathbb{Z}^+$ son las variables observables.

En la figura 2.4 se representan las relaciones causales (dependencias e independencias condicionales), mencionadas anteriormente, entre una secuencia de estados que experimenta una variable aleatoria oculta ($X(k)$) y una secuencia de eventos que experimenta una variable aleatoria observable ($Y(k)$) a lo largo del tiempo. Más específicamente, debido a la condición de Markov, en el HMM de la figura 2.4 se cumplen las siguientes probabilidades condicionales (Jurafsky and Martin (2008)):

$$P(X(k+1) = x_i | X(1) = x_j, X(2) = x_k, \dots, X(k) = x_l) = P(X(k+1) = x_i | X(k) = x_l) \quad (2.48)$$

$$P(Y(k) = y_i | Y(1) = y_j, Y(2) = y_k, \dots, Y(N) = y_l, X(1) = x_j, X(2) = x_k, \dots, X(k) = x_i, \dots, X(N) = x_l) = P(Y(k) = y_i | X(k) = x_i) \quad (2.49)$$

Donde, $x_i, x_j, x_k, x_l \in \mathbb{X}$, $y_i, y_j, y_k, y_l \in \mathbb{Y}$ y N es el número total de observaciones. \mathbb{X} y \mathbb{Y} , por su parte, son los espacios de estados de $X(k)$ y $Y(k)$, respectivamente, los cuales son obtenidos por un muestreo de las variables $h(x(k - T_y))$ y $y(k)$, es decir:

$$\begin{aligned} \mathbb{X} &= \{x_1, x_2, \dots, x_{N_1}\} \\ \mathbb{Y} &= \{y_1, y_2, \dots, y_{N_2}\} \\ x_i &= \{h(x(k - T_y)) | h(x(k - T_y)) \in [a_i, b_i]\}, \quad i = 1 \dots N_1 \\ y_i &= \{y(k) | y(k) \in [c_i, d_i]\}, \quad i = 1 \dots N_2 \end{aligned} \quad (2.50)$$

Donde $a_i, b_i, c_i, d_i \in \mathbb{R}$. La estructura de los espacios de estados \mathbb{X} y \mathbb{Y} , es decir, los valores para $a_i, b_i, i = 1, \dots, N_1$ y $c_i, d_i, i = 1 \dots N_2$, así como los valores de N_1 y N_2 , dependen del rango dinámico del sistema, en otras palabras, se deben escoger con base en el conocimiento previo del rango de movimiento del sistema. Por otra parte, el axioma de la probabilidad establece que dada una variable aleatoria discreta $X(k)$ y su espacio de estados \mathbb{X} se cumple:

$$\sum_{i=1}^{N_1} P(X(k) = x_i) = 1 \quad (2.51)$$

Por lo tanto, también se cumple:

$$\sum_{i=1}^{N_2} P(Y(k) = y_i) = 1 \quad (2.52)$$

Las ecuaciones 2.51 y 2.52 se cumplirán durante todo el tiempo que el sistema de control este en operación independientemente de la selección de N_1 y N_2 si y solo si la distribución de probabilidad inicial para $X(k)$: $P(X(0)) \in \mathbb{R}^{1 \times N_1}$, donde $P(X(0))$ es un vector de probabilidad, se escoge de tal manera que cumpla con el axioma de la probabilidad.

2. DISEÑO DE UN CONTROLADOR NEURONAL PREDICTIVO

Generalizando, los vectores de probabilidad $P(X(k)) \in \mathbb{R}^{1 \times N_1}$ y $P(Y(k)) \in \mathbb{R}^{1 \times N_2}$ se definen como:

$$\begin{aligned} P(X(k)) &= [P(X(k) = x_1) \quad P(X(k) = x_2) \quad P(X(k) = x_3) \quad \dots \quad P(X(k) = x_{N_1})] \\ P(Y(k)) &= [P(Y(k) = y_1) \quad P(Y(k) = y_2) \quad P(Y(k) = y_3) \quad \dots \quad P(Y(k) = y_{N_2})] \end{aligned} \quad (2.53)$$

Del axioma de la probabilidad, un vector de probabilidad cumple con que todas sus componentes pertenecen al rango $[0, 1]$ y la suma de componentes es 1 ([Grossman \(2010\)](#)).

En la ecuación 2.48 se ve claramente porque en un HMM la condición de Markov se puede leer tal como se mencionó anteriormente: *El futuro es independiente del pasado dado el presente*, o también ([Jurafsky and Martin \(2008\)](#)): *Cuando se predice el futuro el pasado no importa, solo el presente*. Mientras que la ecuación 2.49 se puede leer ([Theodoridis \(2015\)](#)): *Las observaciones son independientes del futuro y el pasado dado el presente*, o también ([Jurafsky and Martin \(2008\)](#)): *Cualquier observación depende solo de la variable oculta que la produce y no de cualquier otra variable oculta o de cualquier otra observación*.

Formalmente, un HMM queda especificado por los parámetros de la tabla 2.1 ([Jurafsky and Martin \(2008\)](#)), es decir, a partir de la tabla 2.1 es posible conocer las distribuciones de probabilidad de las variables aleatorias discretas $X(k)$ y $Y(k) \forall k \in \mathbb{Z}^+$ y hacer inferencia sobre estas. El objetivo final de cualquier modelo probabilístico gráfico es hacer inferencia. Para el caso de un HMM existen varios tipos de inferencia que es posible hacer, por ejemplo, calcular la verosimilitud, calcular las probabilidades marginales de las variables aleatorias involucradas, decodificar y predecir ([Theodoridis \(2015\)](#)). Básicamente, calcular la verosimilitud consiste en dada una secuencia de observación O , calcular $P(O)$. Decodificar consiste en dadas una secuencia de observación O , calcular la mejor secuencia de estados ocultos. Finalmente, predecir (el tipo inferencia a desarrollar en el presente trabajo) consiste en dada una secuencia de observación O , estimar de manera óptima el valor de la siguiente observación.

De la tabla 2.1 se observa como las matrices A y B son matrices de probabilidad, esto es que cada uno de sus renglones es un vector de probabilidad. Por la condición de Markov, temporalmente $P(X(k))$ y $P(Y(k))$ dependen únicamente de $P(X(k-1))$ y $P(X(k))$, respectivamente, cumpliéndose:

$$\begin{aligned} P(X(k)) &= P(X(k-1))A \\ P(Y(k)) &= P(X(k))B \end{aligned} \quad (2.54)$$

Aplicando de manera recursiva la primera ecuación de 2.54 se puede calcular en el tiempo actual k la probabilidad $P(X(k+1))$, es decir:

$$P(X(k+1)) = P(X(k))A = (P(X(k-1))A)A = P(X(k-1))A^2 \quad (2.55)$$

$X = \{x_1, x_2, \dots, x_{N_1}\}$	Un conjunto de N_1 estados
$Y = \{y_1, y_2, \dots, y_{N_2}\}$	Un conjunto de N_2 tipos de observaciones
$A = \{a_{ij}\}, \quad i, j = 1, \dots, N_1,$ $a_{ij} = P(X(k) = x_j X(k-1) = x_i), \sum_{j=1}^{N_1} a_{ij} = 1 \forall i$	Una matriz de probabilidades de transición
$O = \{o_1, o_2, \dots, o_N\}, \quad o_i \in Y \forall i$	Una secuencia de N observaciones
$B = \{b_{jm}\}, \quad m = 1, \dots, N_2, j = 1, \dots, N_1,$ $b_{jm} = P(Y(k) = y_m X(k) = x_j), \sum_{m=1}^{N_2} b_{jm} = 1 \forall j$	Una matriz de probabilidades de observación
$\pi = \{\pi_1, \pi_2, \dots, \pi_{N_1}\}, P(X(0) = x_i) = \pi_i, i = 1, \dots, N_1$	Una distribución de probabilidad inicial sobre los estados

Tabla 2.1: Parámetros que definen a un HMM (Jurafsky and Martin (2008)).

Aplicando la ecuación 2.55 de manera recursiva, se puede calcular en el tiempo actual k la probabilidad $P(X(k+T))$:

$$P(X(k+T)) = P(X(k))A^T, \quad T = 1, 2, \dots, \infty \quad (2.56)$$

Con base en 2.54 y 2.56 también se cumple:

$$P(Y(k+T)) = P(X(k+T))B = P(X(k))A^T B, \quad T = 1, 2, \dots, \infty \quad (2.57)$$

En el presente trabajo, se proponen X , Y y π de acuerdo al rango de medición específico de cada sistema. Además, se obtiene O midiendo la salida del sistema, por lo tanto solo hace falta determinar A y B para que el HMM quede especificado y sea posible hacer inferencia sobre $X(k)$ y $Y(k)$. De manera similar como se hizo para los parámetros de la red neuronal, se utiliza un algoritmo de aprendizaje en línea con base en datos para determinar a las matrices A y B que mejor describan la relación 2.47 de naturaleza estocástica.

Para aprender los parámetros A y B de un HMM con base en datos, el algoritmo más popular es el algoritmo de avance-retroceso. Este algoritmo también es conocido como el algoritmo de Baum-Welch y es un caso particular del algoritmo esperanza-maximización (Expectation-Maximization Algorithm), el cual sirve para conocer los parámetros de una distribución de probabilidad de una variable aleatoria oculta dado un conjunto de datos de una variable aleatoria observable. Sin embargo, el algoritmo de avance-retroceso tiene grandes desventajas, por ejemplo, no es un algoritmo de aprendizaje en línea, las soluciones óptimas locales son inevitables, el análisis matemático es difícil de abordar, es difícil de implementar y es computacionalmente costoso (Cho and

Fadali (2011). Por lo tanto, en la presente tesis se utiliza el algoritmo de aprendizaje en línea para un HMM propuesto en Cho and Lee (2009). El análisis matemático de este algoritmo es simple y tiene una baja carga computacional, por lo tanto es adecuado para ser implementado en línea.

Con el algoritmo de aprendizaje de Cho and Lee (2009) se actualizan A y B cuando una nueva observación o_k se encuentra disponible en el tiempo actual k , por lo tanto, las matrices de transición y de observación pasan a ser una función del tiempo, $A(k) = \{a_{ij}(k)\}$ y $B(k) = \{b_{jm}(k)\}$. El primer paso del algoritmo de aprendizaje es definir a cada probabilidad de transición y de observación de acuerdo a:

$$\begin{aligned}
 a_{ij}(k) &= \begin{cases} \frac{m_{ij}(k)}{\sum_{j=1}^{N_1} m_{ij}(k)}, & \text{si } \sum_{j=1}^{N_1} m_{ij}(k) \neq 0 \\ \frac{1}{N_1}, & \text{en otro caso} \end{cases} \\
 b_{jm}(k) &= \begin{cases} \frac{n_{jm}(k)}{\sum_{m=1}^{N_2} n_{jm}(k)}, & \text{si } \sum_{m=1}^{N_2} n_{jm}(k) \neq 0 \\ \frac{1}{N_2}, & \text{en otro caso} \end{cases}
 \end{aligned} \tag{2.58}$$

La definición 2.58 de las matrices $A(k)$ y $B(k)$ se puede considerar como una normalización de las matrices $M(k) = \{m_{ij}(k)\}$ y $N(k) = \{n_{jm}(k)\}$ para que cada renglón de $A(k)$ y $B(k)$ cumpla con ser un vector de probabilidad, es decir, que cumpla con que todas sus componentes pertenezcan al rango $[0, 1]$ y la suma de componentes sea 1. $m_{ij}(k)$ y $n_{jm}(k)$, por otra parte, se conocen como la verosimilitud promedio de $a_{ij}(k)$ y $b_{jm}(k)$, respectivamente, y se calculan como:

$$\begin{aligned}
 m_{ij}(k) &= \left(\frac{1}{k}\right) \sum_{n=0}^k \zeta_{ij}(n), & k = 1, \dots, \infty \\
 \zeta_{ij}(0) &= \zeta_{ij0} \\
 n_{jm}(k) &= \left(\frac{1}{k}\right) \sum_{n=0}^k \beta_{jm}(n), & k = 1, 2, \dots, \infty \\
 \beta_{jm}(0) &= \beta_{jm0}
 \end{aligned} \tag{2.59}$$

Donde los parámetros $\zeta_{ij}(k)$ y $\beta_{jm}(k)$, $k = 1, 2, \dots, \infty$ son seleccionados como 1 ó 0 con base en las observaciones. Así por ejemplo, si se observa que $Y(k) = y_m$ y que $X(k) = x_j$ (suponiendo que es posible observar el valor que toma $X(k)$), entonces $\beta_{jm}(k) = 1$, de lo contrario $\beta_{jm}(k) = 0$. De manera similar, si se observa que $X(k) = x_j$ (suponiendo que es posible observar el valor que toma $X(k)$) y que $X(k-1) = x_i$, entonces $\zeta_{ij}(k) = 1$,

de lo contrario $\zeta_{ij}(k) = 0$, tal como se establece a continuación:

$$\zeta_{ij}(k) = \begin{cases} 1, & \text{si } X(k) = x_j \quad \text{y} \quad X(k-1) = x_i \\ 0, & \text{en otro caso} \end{cases} \quad (2.60)$$

$$\beta_{jm}(k) = \begin{cases} 1, & \text{si } Y(k) = y_m \quad \text{y} \quad X(k) = x_j \\ 0, & \text{en otro caso} \end{cases}$$

$\zeta_{ijo}, \beta_{ijo} \in \mathbb{Z}^+ \cup \{0\}$ de 2.59, por su parte, son seleccionados de acuerdo al conocimiento previo del sistema y son muy importantes para que el algoritmo de aprendizaje funcione correctamente debido a que estos valores van a determinar el valor inicial ($A(1)$ y $B(1)$) de las matrices $A(k)$ y $B(k)$ tal como establecen 2.58 y 2.59.

De 2.59 y 2.60 se puede observar que las verosimilitudes promedio, $m_{ij}(k)$ y $n_{jm}(k)$ (con $\zeta_{ijo} = \beta_{ijo} = 0$), son simplemente el número de veces que sucede una sucesión de eventos, de $X(k-1)$ a $X(k)$ y de $X(k)$ a $Y(k)$, respectivamente, entre el número de sucesiones totales (k). Por lo tanto, la verosimilitud promedio se define prácticamente igual que la probabilidad frecuencial, la cual define a la probabilidad como el cociente entre el número de eventos favorables y el número de eventos totales, cuando el número de eventos tiende a infinito.

Para hacer el algoritmo computacionalmente más eficiente, se calculan $m_{ij}(k)$ y $n_{jm}(k)$ de acuerdo a 2.61, que es la versión recursiva usual de la verosimilitud promedio (ecuación 2.59).

$$m_{ij}(k) = \left(\frac{k}{k+1}\right) m_{ij}(k-1) + \left(\frac{1}{k+1}\right) \zeta_{ij}(k), \quad k = 1, \dots, \infty$$

$$m_{ij}(0) = \zeta_{ijo}$$

$$n_{jm}(k) = \left(\frac{k}{k+1}\right) n_{jm}(k-1) + \left(\frac{1}{k+1}\right) \beta_{jm}(k), \quad k = 1, \dots, \infty$$

$$n_{jm}(0) = \beta_{ijo}$$
(2.61)

De 2.58, 2.60 y 2.61 también se puede observar que si $X(k) = x_j$ y $X(k-1) = x_i$ entonces $a_{ij}(k)$ incrementa (respecto a $a_{ij}(k-1)$), mientras que todas las demás probabilidades de transición en su mismo renglón de la matriz $A(k)$ decremantan. De manera similar, si $X(k) = x_j$ y $Y(k) = y_m$ entonces $b_{jm}(k)$ incrementa (respecto a $b_{jm}(k-1)$), mientras que todas las demás probabilidades de observación en su mismo renglón de la matriz $B(k)$ decremantan. Por lo tanto, el algoritmo aumenta y decrementa los parámetros del HMM de forma correcta de acuerdo a las observaciones.

Como se puede observar, el algoritmo es fácil de implementar y es computacionalmente eficiente. Sin embargo, se supuso que es posible observar $X(k)$, lo cual en realidad no es posible, por lo tanto es necesario determinar $X^*(k)$, es decir, un valor óptimo de

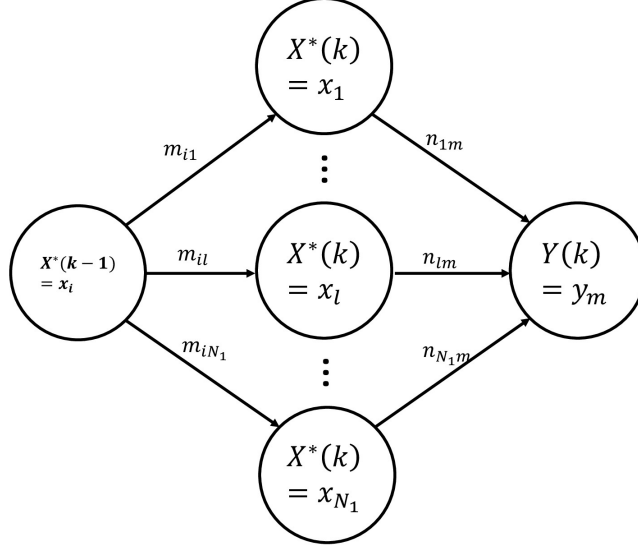


Figura 2.5: N_1 caminos entre $X^*(k-1)$ y $Y(k)$

$X(k)$. Tomando en cuenta que en un HMM es posible observar $Y(k)$ y que se conoce $X^*(k-1)$, entonces $X^*(k)$, como se ilustra en la figura 2.5, solo puede ser seleccionado de N_1 caminos diferentes.

Es importante mencionar que se debe escoger una condición inicial para $X^*(0)$, es decir, $X^*(0) = x_i \in X$. Por otra parte, en un HMM la probabilidad de cada camino se puede calcular como la probabilidad conjunta $P(Y(k) = y_m, X(k) = x_l, X(k-1) = i)$. Utilizando la condición de Markov (ecuación 2.49) y la definición de probabilidad condicional se tiene que:

$$\begin{aligned}
 P(Y(k) = y_m, X(k) = x_l, X(k-1) = i) \\
 &= P(Y(k) = y_m | X(k) = x_l, X(k-1) = i) P(X(k) = x_l, X(k-1) = i) \\
 &= P(Y(k) = y_m | X(k) = x_l) P(X(k) = x_l | X(k-1) = i) P(X(k-1) = i) \\
 &= b_{lm} a_{il} P(X(k-1) = i)
 \end{aligned} \tag{2.62}$$

Con base en 2.62 y tomando en cuenta que la probabilidad $P(X(k-1) = i)$ es común a todos los caminos, entonces la verosimilitud de cada camino que ilustra la figura 2.5 se puede calcular utilizando las últimas verosimilitudes calculadas, $m_{il}(k-1)$ y $n_{lm}(k-1)$, de acuerdo a:

$$J_l(k) = m_{il}(k-1) n_{lm}(k-1), \quad l = 1, \dots, N_1 \tag{2.63}$$

Finalmente, se escoge como $X^*(k)$ al valor con el que se tenga la verosimilitud $J_l(k)$

máxima, es decir:

$$\begin{aligned} X^*(k) &= x_q \\ q &= \underset{l}{\operatorname{argmax}} J_l(k) \end{aligned} \quad (2.64)$$

Este esquema es razonable pues el camino con la verosimilitud máxima es el que tiene la mayor probabilidad de ser el verdadero de entre los caminos de la figura 2.5. Una vez que se determina $X^*(k)$ entonces el algoritmo de aprendizaje continua con la ecuación 2.60 solo que considerando $X^*(k)$ y $X^*(k-1)$ en lugar de $X(k)$ y $X(k-1)$, respectivamente, con lo cual termina la descripción del algoritmo de aprendizaje de los parámetros del HMM.

Con el algoritmo de aprendizaje en línea de [Cho and Lee \(2009\)](#), que se acaba de describir, se determinan a las matrices $A(k)$ y $B(k)$ con base en datos, con lo cual es posible hacer inferencia sobre $X(k)$ y $Y(k)$. El tipo de inferencia a realizar en el presente trabajo, como se mencionó anteriormente, es predecir. El modo de hacer predicciones cambia totalmente dependiendo del modelo probabilístico, por lo tanto, el modo de predecir en el presente trabajo es muy distinto del modo en que se hicieron las predicciones en [Cho and Lee \(2008\)](#) y [Cho and Fadali \(2011\)](#). Otras diferencias importantes respecto a estos trabajos es que en ellos no se consideró a las observaciones para hacer las predicciones (únicamente para aprender los parámetros del modelo probabilístico) y por esta razón en estos trabajos tampoco se tomó en cuenta el enfoque de ventana de tiempo deslizante.

De acuerdo con [Theodoridis \(2015\)](#) predecir consiste en, dados los parámetros que definen a un HMM (tabla 2.1) y una secuencia de k observaciones, $O_k = \{o_1, o_2, \dots, o_k\}$, calcular al vector de probabilidad $P(Y(k+1)|O_k) \in \mathbb{R}^{1 \times N_2}$:

$$P(Y(k+1)|O_k) = [P(Y(k+1) = y_1|O_k) \quad P(Y(k+1) = y_2|O_k) \quad \dots \quad P(Y(k+1) = y_{N_2}|O_k)] \quad (2.65)$$

Con base en este vector, como se explicará más adelante, es posible estimar de manera óptima el valor de la siguiente observación. Empezando con el procedimiento para calcular $P(Y(k+1)|O_k)$ se supone (por el momento) que se conoce $P(X(k) = x_i, O_k) \in [0, 1] \subset \mathbb{R}$, $i = 1 \dots N_1$, es decir, la probabilidad conjunta de estar en un cierto estado x_i y de haber visto la secuencia de observación O_k . Por otra parte, considerando que $A(k)$ varía lentamente se puede calcular $P(X(k) = x_i) \in [0, 1] \subset \mathbb{R}$, $i = 1 \dots N_1$, es decir, la probabilidad marginal de estar en el tiempo k en un cierto estado x_i , de acuerdo a la primera ecuación de 2.54. Este cálculo se realiza debido a que si se conocen $P(X(k) = x_i, O_k)$ y $P(X(k) = x_i)$ entonces es posible calcular $P(O_k|X(k) = x_i) \in [0, 1] \subset \mathbb{R}$ de acuerdo a la definición de probabilidad condicional:

$$P(O_k|X(k) = x_i) = \frac{P(X(k) = x_i, O_k)}{P(X(k) = x_i)}, \quad i = 1 \dots N_1 \quad (2.66)$$

Además, si se conoce $P(X(k) = x_i, O_k)$, $i = 1 \dots N_1$, entonces es posible calcular $P(O_k) \in [0, 1] \subset \mathbb{R}$, de acuerdo a:

$$P(O_k) = \sum_{i=1}^{N_1} P(X(k) = x_i, O_k) \quad (2.67)$$

Una vez que se conoce $P(O_k)$, $P(X(k) = x_i)$ y $P(O_k|X(k) = x_i)$ entonces es posible conocer a $P(X(k) = x_i|O_k) \in [0, 1] \subset \mathbb{R}$ de acuerdo al teorema de Bayes:

$$P(X(k) = x_i|O_k) = \frac{P(X(k) = x_i) P(O_k|X(k) = x_i)}{P(O_k)}, \quad i = 1 \dots N_1 \quad (2.68)$$

Sustituyendo 2.66 en 2.68 da por resultado que es posible calcular $P(X(k) = x_i|O_k)$ sin tener que calcular $P(X(k) = x_i)$, pues:

$$\begin{aligned} P(X(k) = x_i|O_k) &= \frac{P(X(k) = x_i) P(X(k) = x_i, O_k)}{P(O_k) P(X(k) = x_i)} \\ &= \frac{P(X(k) = x_i, O_k)}{P(O_k)} \end{aligned} \quad (2.69)$$

Nótese que por 2.67 se debe cumplir:

$$\sum_{i=1}^{N_1} P(X(k) = x_i|O_k) = 1 \quad (2.70)$$

Por lo tanto $P(X(k)|O_k) \in \mathbb{R}^{1 \times N_1}$, al igual que $P(X(k))$, es un vector de probabilidad, donde:

$$P(X(k)|O_k) = [P(X(k) = x_1|O_k) \quad P(X(k) = x_2|O_k) \quad \dots \quad P(X(k) = x_{N_1}|O_k)] \quad (2.71)$$

Una vez que se conoce al vector $P(X(k)|O_k)$, considerando que $A(k)$ y $B(k)$ cambian lentamente, se puede calcular al vector de probabilidad $P(Y(k+1)|O_k)$ con base a las ecuaciones 2.54 y 2.55, de acuerdo a:

$$\begin{aligned} P(X(k+1)|O_k) &= P(X(k)|O_k)A(k) \\ P(Y(k+1)|O_k) &= P(X(k+1)|O_k)B(k) \end{aligned} \quad (2.72)$$

Con base $P(Y(k+1)|O_k)$ es posible en el tiempo actual k dar un valor estimado de manera óptima de la variable aleatoria observable en el tiempo $k+1$ dada una secuencia de observación O_k , sin embargo, debido a que generalmente el retardo de observación es mayor a un periodo de muestro, esta predicción no representa la actualidad del sistema de acuerdo a la ecuación 2.47 y por lo tanto no sería muy útil con fines de control. De acuerdo a la ecuación 2.47 la predicción de la variable observable en el tiempo $k+T_y$ es un estimado de la actualidad del sistema, sin embargo, considerando también al retardo

de control, es conveniente predecir el valor de la variable observable en el tiempo $k + T_\eta$, donde T_η es de acuerdo a 2.8, de esta manera, cuando llegue la señal de control a la planta, esta señal va a ser de acuerdo a la actualidad del sistema en ese momento. En consecuencia, con base en las ecuaciones 2.56 y 2.57 y considerando que $A(k)$ y $B(k)$ varían lentamente se calcula $P(Y(k + T_\eta)|O_k)$, de acuerdo a:

$$\begin{aligned} P(X(k + T_\eta)|O_k) &= P(X(k)|O_k)A^{T_\eta}(k) \\ P(Y(k + T_\eta)|O_k) &= P(X(k + T_\eta)|O_k)B(k) \end{aligned} \quad (2.73)$$

En consecuencia, el valor estimado de manera óptima de la variable aleatoria observable en el tiempo $k + T_\eta$ dada una secuencia de observación O_k , $\hat{Y}(k + T_\eta|O_k) \in \mathbb{Y}$, se puede escoger como la componente más grande del vector $P(Y(k + T_\eta)|O_k)$, es decir:

$$\begin{aligned} \hat{Y}(k + T_\eta|O_k) &= y_u \\ u &= \underset{m}{\operatorname{argmax}} P(Y(k + T_\eta) = y_m|O_k) \end{aligned} \quad (2.74)$$

Finalmente, recordando que la variable continua $y(k + T_\eta)$ es representada por la variable aleatoria discreta $Y(k + T_\eta)$, se puede obtener el estimado de la salida del sistema 1.10 en el tiempo $k + T_\eta$, $\hat{y}(k + T_\eta)$, a partir de $\hat{Y}(k + T_\eta|O_k)$ de la siguiente forma:

$$\hat{y}(k + T_\eta) = \frac{c_u + d_u}{2} \quad (2.75)$$

Donde, c_u y d_u son los límites de la ecuación 2.50 que definen a y_u de la ecuación 2.74, es decir:

$$y_u = \{y(k)|y(k) \in [c_u, d_u]\} \quad (2.76)$$

Otra forma de obtener $\hat{y}(k + T_\eta)$ a partir de $P(Y(k + T_\eta)|O_k)$ de 2.72, es la salida promedio o ponderada:

$$\hat{y}(k + T_\eta) = \sum_{i=1}^{N_2} v_i P(Y(k + T_\eta) = y_i|O_k) \quad (2.77)$$

Donde v_i es el valor medio del rango que define a cada $y_i \in \mathbb{Y}$, es decir:

$$\begin{aligned} v_i &= \frac{c_i + d_i}{2}, \\ y_i &= \{y(k)|y(k) \in [c_i, d_i]\} \quad i = 1 \dots N_2 \end{aligned} \quad (2.78)$$

En el presente trabajo se utiliza la salida ponderada (ecuación 2.77) debido a que de esta forma $\hat{y}(k + T_\eta)$ puede tomar valores continuos. Como se puede apreciar, el procedimiento presentado hasta el momento para calcular $\hat{y}(k + T_\eta)$ es fácil de implementar, sin embargo, al inicio se supuso que se conocían a los valores $P(X(k) = x_i, O_k)$, $i = 1 \dots N_1$, los cuales, en realidad, se tienen que calcular.

2. DISEÑO DE UN CONTROLADOR NEURONAL PREDICTIVO

Para hacer estos cálculos utilizaremos el algoritmo de avance, el cual sirve para calcular la probabilidad de ver una serie de k observaciones, $O_k = \{o_1, o_2, \dots, o_k\}$, en un HMM, en otras palabras, el algoritmo de avance se utiliza para calcular $P(O_k)$, sin embargo, tal como indica 2.67, para conocer $P(O_k)$ primero hay que conocer $P(X(k) = x_i, O_k), i = 1 \dots N_1$, por lo tanto es posible utilizar este algoritmo para hacer estos cálculos.

El algoritmo de avance va almacenando los valores $P(X(1) = x_p, o_1), P(X(2) = x_p, o_1, o_2), \dots, P(X(k-1) = x_p, o_1, o_2, \dots, o_{k-1}), p = 1, \dots, N_1$, en una matriz $\alpha = \{\alpha_{pt}\} \in \mathbb{R}^{N_1 \times k}$, la cual se va llenando conforme el algoritmo se aproxima a su cálculo final ($P(X(k) = x_p, O_k) \forall p$). Las columnas de la matriz α están asociadas al número de observaciones, es decir, habrá una columna por cada observación. Los renglones, por su parte, están asociados al número de estados que puede tomar la variable oculta. De esta manera, la componente α_{pt} representa la siguiente probabilidad (Jurafsky and Martin (2008)):

$$\alpha_{pt} = P(X(t) = x_p, o_1, o_2, \dots, o_t) \quad (2.79)$$

Por lo tanto, las probabilidades que se buscan ($P(X(k) = x_p, O_k), p = 1 \dots N_1$), son las componentes de la última columna de la matriz α . El algoritmo de avance es eficiente porque la matriz α se va llenando por columnas y para calcular la columna t , únicamente se utilizan los cálculos de la columna anterior ($t-1$), las matrices $A(k)$ y $B(k)$ y q , donde q es el valor de la observación en el tiempo t , es decir, $o_t = y_q \in \mathbb{Y}$. De hecho, se puede comprobar que la componente α_{pt} se puede calcular de acuerdo a la siguiente ecuación (Jurafsky and Martin (2008)).

$$\alpha_{pt} = \begin{cases} \pi_p b_{pq}(k), & \text{si } t = 1 \\ \left(\sum_{s=1}^{N_1} \alpha_{s(t-1)} a_{sp}(k) \right) b_{pq}(k), & \text{si } 2 \leq t \leq k \end{cases} \quad (2.80)$$

Con 2.80 se puede calcular $\alpha_{pk} = P(X(k) = x_p, o_1, o_2, \dots, o_k), p = 1 \dots N_1$, en línea de manera computacionalmente eficiente únicamente utilizando $\alpha_{p(k-1)}, A(k)$ y $B(k)$ y q . Sin embargo, el cálculo de $P(X(k) = x_p, o_1, o_2, \dots, o_k)$ involucra todas las observaciones hechas hasta el momento, por lo tanto, el cálculo de $P(Y(k+T_\eta) = y_i | o_1, o_2, \dots, o_k), i = 1 \dots N_2$, también involucrará todas las observaciones, lo cual puede llegar a ser contraproducente pues en los sistemas de control el número de observaciones puede llegar a ser muy grande y no todas las observaciones son necesarias para predecir la siguiente salida sino que, por el contrario, los datos más recientes deben ser más relevantes (Cho and Lee (2009)).

Para tomar en consideración este esquema se adopta una ventana de tiempo deslizante de tamaño N_w y en el presente trabajo se entiende por predecir a, dada la secuencia de las últimas N_w observaciones, $O_{N_w} = \{o_{k-N_w+1}, o_{k-N_w+2}, \dots, o_k\}$, calcular al vector de probabilidad $P(Y(k+T_\eta) | O_{N_w}) \in \mathbb{R}^{1 \times N_2}$. En consecuencia, cada vez que llegue un nuevo dato al controlador con el algoritmo de avance se calcula

$P(X(k) = x_p, o_{k-N_w+1}, o_{k-N_w+2}, \dots, o_k), p = 1 \dots N_1$, para lo cual en cada iteración, $k = 1, \dots, \infty$, se tiene que calcular una matriz $\alpha \in \mathbb{R}^{N_1 \times N_w}$, donde cada componente α_{pt} es de acuerdo a la ecuación 2.81, la cual es una modificación de la ecuación 2.80.

$$\alpha_{pt} = \begin{cases} P(X(k - N_w) = x_p) b_{pq}(k), & \text{si } t = 1 \\ \left(\sum_{s=1}^{N_1} \alpha_{s(t-1)} a_{sp}(k) \right) b_{pq}(k), & \text{si } 2 \leq t \leq N_w, \end{cases} \quad p = 1 \dots N_1 \quad (2.81)$$

Donde, para el cálculo de cada α_{pt} , q es el valor de la observación en el tiempo $k - N_w + t$, es decir, q cumple con la siguiente ecuación:

$$o_{k-N_w+t} = y_q \in \mathbb{Y} \quad (2.82)$$

Y con las siguientes condiciones iniciales:

$$o_{2-N_w} = o_{3-N_w} = o_{4-N_w} = \dots = o_{-1} = o_0 = y_i \in \mathbb{Y} \quad (2.83)$$

Además, $P(X(k - N_w) = x_p), p = 1 \dots N_1$, de 2.81 se calcula en forma de vector como $P(X(k - N_w)) = P(X(k - N_w - 1))A(k)$, de acuerdo a la ecuación 2.54, con las siguientes condiciones iniciales:

$$P(X(1 - N_w)) = P(X(2 - N_w)) = \dots = P(X(-1)) = P(X(0)) \quad (2.84)$$

Resumiendo toda la sección 2.4, con el método de [Cho and Lee \(2009\)](#) se aprenden en línea con base en datos a las matrices $A(k)$ y $B(k)$. Posteriormente, con el método de avance (ecuación 2.81) se calculan $P(X(k) = x_j, O_{N_w}), j = 1 \dots N_1$. Después, con las ecuaciones 2.67, 2.69 y 2.73 se calcula $P(Y(k + T_\eta) | O_{N_w})$. Por consiguiente, se predice la siguiente salida del sistema $\hat{y}(k + T_\eta)$ dadas las últimas N_w observaciones de acuerdo a 2.77.

Para terminar con el control neuronal predictivo se utiliza $\hat{y}(k + T_\eta)$ para generar una señal de control. Primero se define (con base en 2.29) a la predicción del error:

$$\hat{e}(k) = -\hat{y}(k + T_\eta) \quad (2.85)$$

La predicción del error se utiliza en lugar de la señal de error $e(k)$ de la ecuación 2.29 para definir un nuevo vector de entrada a la red neuronal:

$$\hat{\sigma}(k) = \left[\hat{e}(k) \quad \frac{\hat{e}(k) - \hat{e}(k-1)}{h} \right]^T \in \mathbb{R}^3 \quad (2.86)$$

Sustituyendo al vector de entrada de la ecuación 2.38. También se utiliza 2.85 para definir una nueva función de costo para aprender los parámetros de la red neuronal:

$$\hat{J}(k) = \frac{1}{2} \hat{e}^2(k) \quad (2.87)$$

Sustituyendo a la función de costo 2.41. Con estas modificaciones al control neuronal se concluye el diseño del controlador neuronal predictivo. Gracias a la predicción de la

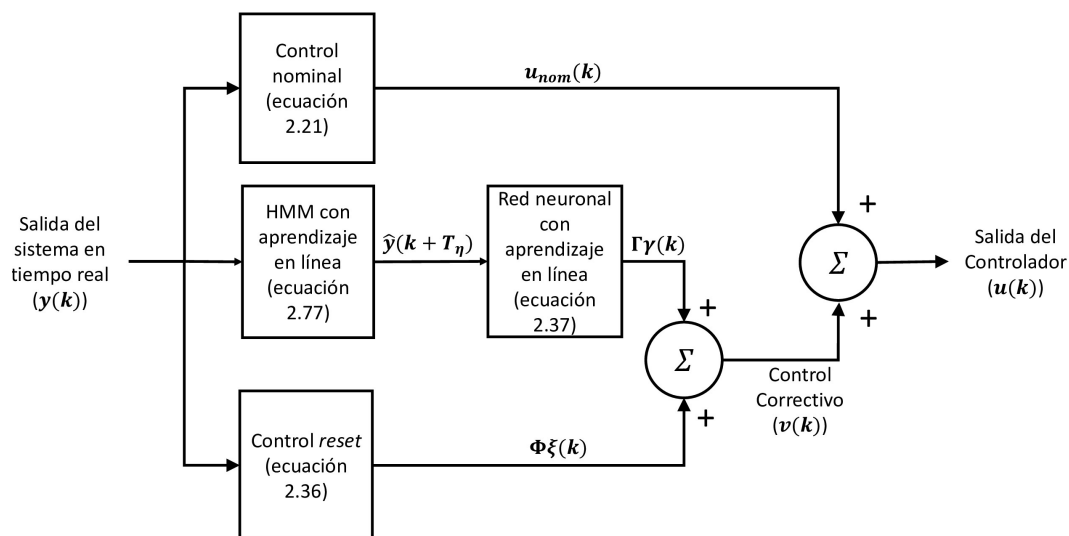


Figura 2.6: Configuración de la señal de control para un NCS.

salida del sistema el control neuronal va a generar una señal de control más acorde a la actualidad del sistema y, por lo tanto, el desempeño del controlador mejora respecto a solo tener el control nominal (diseñado para un retardo de tiempo constante).

Resumiendo el control neuronal predictivo propuesto en este capítulo, primero se diseñó un control nominal considerando un retardo de tiempo constante y un sistema linealizado, posteriormente, se agregó un control basado en *reset* el cual es una componente del control correctivo y sirve para compensar a las perturbaciones del sistema nominal mejorando el desempeño del controlador. A continuación se diseñó un control basado en redes neuronales con aprendizaje en línea, el cual es la segunda parte del control correctivo y sirve para contrarrestar a las perturbaciones resultantes del retardo de comunicación estocástico. El aprendizaje de la red neuronal se logra por medio del método del descenso por gradiente. Finalmente, se planteó un sistema de predicción basado en un HMM empleando el algoritmo de avance. El HMM también cuenta con aprendizaje de sus parámetros en línea y se logra por medio del método planteado en [Cho and Lee \(2009\)](#). La predicción de la salida del sistema un cierto número de periodos de muestreo adelante se utiliza como nueva entrada a la red neuronal y gracias a esto mejorará significativamente el desempeño del controlador. En la figura 2.6 se ilustra como interactúan todas las componentes del control neuronal predictivo y es una versión más completa de la figura 2.2.

Implementación del controlador

Con el fin de mostrar la efectividad del controlador planteado en el capítulo 2, en este se implementa detalladamente en el modelo de un sistema dinámico en particular. El modelo es de un sistema electromecánico con 4 hélices como actuadores, por lo tanto, es un sistema tipo cuadricóptero, sin embargo, el sistema se encuentra anclado a la tierra, por lo tanto, no es posible cambiar su posición, únicamente su orientación, con lo cual solo tiene 3 grados de libertad: los ángulos de *pitch*, *roll* y *yaw*. Más específicamente, el modelo es del sistema conocido como *3 DOF HOVER* desarrollado por la empresa *Quanser*. En la figura 3.1 se puede ver al sistema y las ecuaciones 3.1 son el modelo al cual se llegó con base en la información contenida en [Quanser \(2013\)](#), utilizando las ecuaciones de Euler-Lagrange.

$$\begin{aligned}
 x(t) &= [x_1(t) \ x_2(t) \ x_3(t) \ x_4(t) \ x_5(t) \ x_6(t)]^T \\
 &= [\theta(t) \ \dot{\theta}(t) \ \phi(t) \ \dot{\phi}(t) \ \psi(t) \ \dot{\psi}(t)]^T \\
 \dot{x}(t) &= \begin{bmatrix} x_2(t) \\ \frac{-mgl}{J_p} \text{sen}(x_1(t)) - \frac{b}{J_p} x_2(t) - \frac{K_f L}{J_p} (V_b - V_f) \\ x_4(t) \\ \frac{-mgl}{J_r} \text{sen}(x_3(t)) - \frac{b}{J_r} x_4(t) - \frac{K_f L}{J_r} (V_l - V_r) \\ x_6(t) \\ \frac{-b}{J_y} x_6(t) - \frac{K_t}{J_y} (V_b + V_f - V_r - V_l) \end{bmatrix} \quad (3.1) \\
 z(t) &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x(t)
 \end{aligned}$$

En la tabla 3.1 se encuentra el nombre, el valor numérico y las unidades de los parámetros y las variables utilizadas en el modelo. En la figura 3.2 se muestra un diagrama de cuerpo libre del sistema que sirve como referencia para algunas variables del modelo 3.1. Aquí F_b , F_f , F_l y F_r son las fuerzas de empuje de cada par motor-hélice, las cuales son iguales al voltaje de su respectivo motor por una constante de fuerza-empuje, tal



Figura 3.1: Quanser 3 DOF Hover (Quanser (2013)).

como se expresa en 3.2.

$$\begin{aligned} F_b &= K_f V_b \\ F_f &= K_f V_f \\ F_l &= K_f V_l \\ F_r &= K_f V_r \end{aligned} \tag{3.2}$$

Como se puede apreciar 3.1 es un sistema MIMO no lineal de orden 6 con 4 entradas (V_f, V_b, V_l, V_r) y 3 salidas ($\theta(t), \phi(t), \psi(t)$). Sin embargo, si se definen:

$$\begin{aligned} U_1 &= \frac{K_f L}{J_p} (V_b - V_f) \\ U_2 &= \frac{K_f L}{J_r} (V_l - V_r) \\ U_3 &= \frac{K_t}{J_y} (V_b + V_f - V_r - V_l) \end{aligned} \tag{3.3}$$

Entonces el sistema 3.1 puede ser visto como 3 sistemas SISO de orden 2: 3.4, 3.5 y 3.6,

Nombre Parámetro/Variable	Símbolo	Valor numérico/Unidades
Masa total del Hover	m	2.85 [kg]
Distancia del eje de giro al centro de masa	l	0.008 [m]
Gravedad	g	9.78 [$\frac{m}{s^2}$]
Momento de inercia respecto del eje de giro del <i>pitch</i>	J_p	0.0552 [kg - m ²]
Momento de inercia respecto del eje de giro del <i>roll</i>	J_r	0.0552 [kg - m ²]
Momento de inercia respecto del eje de giro del <i>yaw</i>	J_y	0.11 [kg - m ²]
<i>Pitch</i>	$\theta(t)$	[rad]
<i>Roll</i>	$\phi(t)$	[rad]
<i>Yaw</i>	$\psi(t)$	[rad]
Coefficiente de fricción viscosa	b	0.08 [N - m - seg]
Constante de fuerza-empuje	K_f	0.1188 [$\frac{N}{V}$]
Constante de par-empuje	K_t	0.0036 [$\frac{N-m}{V}$]
Distancia del eje de giro a cada motor	L	0.197 [m]
Voltaje motor trasero	V_b	[V]
Voltaje motor frontal	V_f	[V]
Voltaje motor izquierdo	V_l	[V]
Voltaje motor derecho	V_r	[V]

Tabla 3.1: Parámetros del Quanser 3 DOF Hover

donde U_1 , U_2 y U_3 son las entradas de cada sistema.

$$\begin{aligned}
 x_\theta(t) &= [x_1(t) \quad x_2(t)]^T \\
 &= [\theta(t) \quad \dot{\theta}(t)]^T \\
 \dot{x}_\theta(t) &= \begin{bmatrix} x_2(t) \\ -\frac{mgl}{J_p} \text{sen}(x_1(t)) - \frac{b}{J_p} x_2(t) - U_1 \end{bmatrix} \\
 z(t) &= [1 \quad 0] x_\theta(t)
 \end{aligned} \tag{3.4}$$

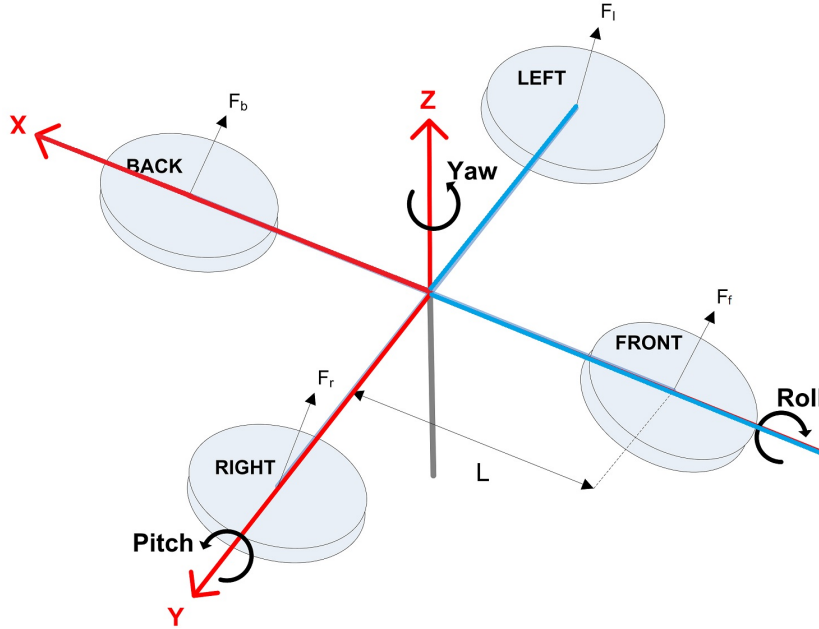


Figura 3.2: Diagrama de cuerpo libre del Quanser 3 DOF Hover (Quanser (2013)).

$$\begin{aligned}
 x_\phi(t) &= [x_1(t) \quad x_2(t)]^T \\
 &= [\phi(t) \quad \dot{\phi}(t)]^T \\
 \dot{x}_\phi(t) &= \begin{bmatrix} x_2(t) \\ \frac{-mgl}{J_r} \text{sen}(x_1(t)) - \frac{b}{J_r} x_2(t) - U_2 \end{bmatrix} \\
 z(t) &= [1 \quad 0] x_\phi(t)
 \end{aligned} \tag{3.5}$$

$$\begin{aligned}
 x_\psi(t) &= [x_1(t) \quad x_2(t)]^T \\
 &= [\psi(t) \quad \dot{\psi}(t)]^T \\
 \dot{x}_\psi(t) &= \begin{bmatrix} x_2(t) \\ \frac{-b}{J_y} x_2(t) - U_3 \end{bmatrix} \\
 z(t) &= [1 \quad 0] x_\psi(t)
 \end{aligned} \tag{3.6}$$

El control propuesto en el capítulo 2 se va a implementar para cada uno de estos sistemas SISO por separado. Posteriormente, para conocer las entradas de control del sistema original (3.1) se utilizan las ecuaciones 3.7 (obtenidas a partir de 3.3 y de la restricción algebraica $V_f = 0.6V_b$), donde los voltajes V_f , V_b , V_l , V_r son función de U_1 , U_2 y U_3 en

una relación unívoca.

$$\begin{aligned}
V_b &= \frac{J_p}{0.4K_f L} U_1 \\
V_f &= \frac{3J_p}{2K_f L} U_1 \\
V_r &= \frac{2J_p}{K_f L} U_1 - \frac{J_r}{2K_f L} U_2 - \frac{J_y}{2K_t} U_3 \\
V_l &= \frac{2J_p}{K_f L} U_1 + \frac{J_r}{2K_f L} U_2 - \frac{J_y}{2K_t} U_3
\end{aligned} \tag{3.7}$$

Empezando con 3.4, note que 3.4 es el equivalente a 1.1, es decir, es el modelo en tiempo continuo de un sistema SISO no lineal a controlar donde el origen es un punto de equilibrio. Con fines de diseño del controlador, como se mencionó en la sección 1.5.2, se puede considerar que el periodo de muestreo es constante e igual al valor medio, $h_m \in \mathbb{R}^+$. Además, considerando que la señal de control ZOH, $U_1(t)$, se actualiza (a consecuencia de una información medida anteriormente debido al retardo de comunicación) cada periodo de tiempo h_m , entonces se puede modelar la evolución del sistema 3.4 con ecuaciones en diferencias, es decir, en tiempo discreto.

Con el método de Euler de la definición 7 (dado un h_m) se encontraron las siguientes ecuaciones en diferencias que modelan la dinámica de 3.4 en tiempo discreto:

$$\begin{aligned}
x_\theta(k+1) &= \begin{bmatrix} x_1(k) + h_m x_2(k) \\ \left(1 - \frac{bh_m}{J_p}\right) x_2(k) - \frac{h_m m g l}{J_p} \text{sen}(x_1(k)) - h_m U_1(k) \end{bmatrix} \\
z(k) &= [1 \quad 0] x_\theta(k)
\end{aligned} \tag{3.8}$$

En un NCS real h_m tiene que ser estimado de datos experimentales, sin embargo, con fines de obtener resultados de simulaciones se escogió $h_m = 0.0065[\text{seg}]$. Antes de continuar, note que el origen de 3.8 es un punto de equilibrio y que 3.8 es el equivalente a 1.9. Agregando el retardo de comunicación incierto y ruido de observación se obtiene:

$$\begin{aligned}
x_\theta(k+1) &= \begin{bmatrix} x_1(k) + h_m x_2(k) \\ \left(1 - \frac{bh_m}{J_p}\right) x_2(k) - \frac{h_m m g l}{J_p} \text{sen}(x_1(k)) - h_m u(k - \tau_u(k)) \end{bmatrix} \\
y_d(k) &= z(k) + w(k) = x_1(k) + w(k) \\
y(k) &= y_d(k - \tau_y(k))
\end{aligned} \tag{3.9}$$

Donde $u(k - \tau_u(k)) = U_1(k)$, $u(k)$ es la señal de control calculada en el controlador en el tiempo k , $y(k) \in \mathbb{R}$ es la salida medida, es decir, es la señal que llega al controlador en el tiempo k , $w(k)$ representa ruido de medición y $\tau_u(k), \tau_y(k) \in \mathbb{Z}^+$ son los retardos de control y observación, respectivamente, los cuales son variantes en el tiempo y cumplen con 1.11. Además, se consideran las siguientes condiciones iniciales para $u(k)$: $u(-\tau_u(0)) = u(1 - \tau_u(1)) = u(2 - \tau_u(2)) = \dots = u(0) = 0$. También se consideran

3. IMPLEMENTACIÓN DEL CONTROLADOR

las siguientes condiciones iniciales para $y_d(k)$: $y_d(-\tau_y(0)) = z(-\tau_y(0)) + w(-\tau_y(0)) = y_d(1-\tau_y(1)) = z(1-\tau_y(1)) + w(1-\tau_y(1)) = y_d(2-\tau_y(2)) = z(2-\tau_y(2)) + w(2-\tau_y(2)) = \dots = y_d(-1) = z(-1) + w(-1) = 0$. Note que 3.9 es el equivalente a 1.10.

Hasta el momento solo se ha obtenido una expresión matemática de un sistema dinámico tipo cuadricóptero como un sistema de control sobre redes, es decir, tomando en cuenta el periodo de muestreo y el retardo de comunicación. El objetivo del controlador neuronal predictivo es que si 3.4 tiene una condición inicial $x(0) \neq \bar{0}$ se cumpla que $x(t)$ se aproxime y permanezca cerca de cero. Asimismo, se busca mejorar el desempeño en comparación de la linealización exacta y el control de sistemas discretos con retardo de tiempo constante de Fridman (2014). El primer paso de la implementación del controlador es expresar 3.9 como la suma de una dinámica lineal y una no lineal de acuerdo a 2.1.

$$\begin{aligned} x_\theta(k+1) &= \begin{bmatrix} 1 & h_m \\ 0 & 1 - \frac{bh_m}{J_p} \end{bmatrix} x_\theta(k) + \begin{bmatrix} 0 \\ -h_m \end{bmatrix} u(k - \tau_u(k)) + \begin{bmatrix} 0 \\ -\frac{h_m m g l}{J_p} \text{sen}(x_1(k)) \end{bmatrix} \\ y_d(k) &= [1 \quad 0] x_\theta(k) + w(k) \\ y(k) &= y_d(k - \tau_y(k)) \end{aligned} \quad (3.10)$$

Con base en 3.10 la ecuación 3.11 define al sistema nominal.

$$\begin{aligned} x_\theta(k+1) &= \begin{bmatrix} 1 & h_m \\ 0 & 1 - \frac{bh_m}{J_p} \end{bmatrix} x_\theta(k) + \begin{bmatrix} 0 \\ -h_m \end{bmatrix} u_{nom_1}(k - T_u) \\ y(k) &= [1 \quad 0] x_\theta(k - T_y) \end{aligned} \quad (3.11)$$

Donde T_y y T_u se calculan con base en 1.12 y en valores para $E(\tau_{u(t_k)})$ y $E(\tau_{y(s_k)})$. En un NCS real $E(\tau_{u(t_k)})$ y $E(\tau_{y(s_k)})$ tienen que ser estimados con base en datos experimentales, sin embargo, con fines de obtener resultados de simulaciones se escogieron $E(\tau_{u(t_k)}) = E(\tau_{y(s_k)}) = 0.028[\text{seg}]$. Note que para estos valores se tiene que $T_y = T_u = 5$. De 3.11 también note que se tienen los siguientes valores propios:

$$\begin{aligned} \lambda_1 \left(\begin{bmatrix} 1 & h_m \\ 0 & 1 - \frac{bh_m}{J_p} \end{bmatrix} \right) &= 1 \\ \lambda_2 \left(\begin{bmatrix} 1 & h_m \\ 0 & 1 - \frac{bh_m}{J_p} \end{bmatrix} \right) &= 0.9906 \end{aligned} \quad (3.12)$$

Por lo tanto, el origen de 3.11 es estable aunque no asintóticamente. El objetivo del control $u_{nom_1}(k)$ es hacer que el origen de 3.11 sea asintóticamente estable. También se calcula la matriz de observabilidad (M_{obs}) del sistema nominal (3.11) y el rango de

esta:

$$M_{obs} = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & h_m \end{bmatrix} \quad (3.13)$$

$$\text{rango}(M_{obs}) = 2$$

Por lo tanto el par (A, C) es observable. Además, note que de 3.11 (y también de 3.8) se cumple:

$$x_1(k+1) = x_1(k) + h_m x_2(k) \quad (3.14)$$

Por lo tanto:

$$\begin{aligned} x_2(k) &= \frac{x_1(k+1) - x_1(k)}{h_m} \\ &\approx \frac{x_1(k) - x_1(k-1)}{h_m} \end{aligned} \quad (3.15)$$

En consecuencia se puede estimar $\hat{x}(k - T_y)$ a partir de $y(k)$ de la siguiente forma:

$$\hat{x}(k - T_y) = \begin{bmatrix} x_1(k - T_y) \\ \hat{x}_2(k - T_y) \end{bmatrix} = \begin{bmatrix} x_1(k - T_y) \\ \frac{x_1(k - T_y) - x_1(k - T_y - 1)}{h_m} \end{bmatrix} = \begin{bmatrix} y(k) \\ \frac{y(k) - y(k-1)}{h_m} \end{bmatrix} \quad (3.16)$$

También note que la matriz de controlabilidad de 3.11 (M_{con}) cumple con:

$$M_{con} = [B \quad AB] = \begin{bmatrix} 0 & -h_m^2 \\ -h_m & -h_m + \frac{bh_m^2}{J_p} \end{bmatrix} \quad (3.17)$$

$$\text{rango}(M_{con}) = 2$$

Por lo tanto, el par (A, B) de 3.11 es estabilizable para $T_u = T_y = 0$. Como se demostró en la sección 2.1 el control para el sistema nominal 3.11 puede resolverse simplemente resolviendo la estabilización del siguiente sistema aumentado equivalente sin retardo:

$$\begin{aligned} \tilde{x}(k+1) &= \tilde{A}\tilde{x}(k) + \tilde{B}u_{nom_1}(k) \\ \tilde{x}(k+1) &= \begin{bmatrix} 1 & h_m & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 - \frac{bh_m}{J_p} & 0 & 0 & \dots & 0 & 0 & -h_m \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 \end{bmatrix} \tilde{x}(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} u_{nom_1}(k) \end{aligned} \quad (3.18)$$

3. IMPLEMENTACIÓN DEL CONTROLADOR

Donde se utiliza el siguiente estado aumentado:

$$\tilde{x}(k) = [x_\theta(k - T_y), u_{nom_1}(k - 1), \dots, u_{nom_1}(k - 10)]^T \in \mathbb{R}^{12} \quad (3.19)$$

Con el control 3.20 se estabiliza al sistema nominal 3.11 pues $\tilde{A} - \tilde{B}K$ es una matriz Schur.

$$\begin{aligned} u_{nom_1}(k) &= -K\hat{x}(k) \\ &= -K_2\hat{x}(k - T_y) - K_3 \begin{bmatrix} u_{nom_1}(k - 1) \\ \vdots \\ u_{nom_1}(k - 10) \end{bmatrix} \\ &= -[k_1 \quad k_2] \hat{x}(k - T_y) - [k_3 \quad k_4 \quad k_5 \quad k_6 \quad k_7 \quad k_8 \quad k_9 \quad k_{10} \quad k_{11} \quad k_{12}] \begin{bmatrix} u_{nom_1}(k - 1) \\ \vdots \\ u_{nom_1}(k - 10) \end{bmatrix} \end{aligned} \quad (3.20)$$

Donde:

$$\begin{aligned} K &= [k_1 \quad k_2 \quad \dots \quad k_{12}] \\ k_1 &= -1 \cdot 10^{-6} \\ k_2 &= -5 \cdot 10^{-6} \\ k_3 &= -8.5516461 \\ k_4 &= 32.966238755 \\ k_5 &= -75.433980284 \\ k_6 &= 113.45485275 \\ k_7 &= -117.18753174 \\ k_8 &= 84.17991468 \\ k_9 &= -41.522550866 \\ k_{10} &= 13.45890843 \\ k_{11} &= -2.588510576 \\ k_{12} &= 0.2243050517 \end{aligned} \quad (3.21)$$

Donde $\hat{x}(k - T_y)$ se calcula de acuerdo a 3.16. Con base en 2.4 la señal de control $u(k)$ del sistema original 3.10 es la suma de un control nominal ($u_{nom}(k)$) y de un control correctivo con aprendizaje en línea ($v(k)$). Con lo cual el sistema original 3.10 en lazo

cerrado queda:

$$\begin{aligned}
x_\theta(k+1) &= \begin{bmatrix} 1 & h_m \\ 0 & 1 - \frac{bh_m}{J_p} \end{bmatrix} x_\theta(k) + \begin{bmatrix} 0 \\ -h_m \end{bmatrix} u_{nom}(k - \tau_u(k)) + \begin{bmatrix} 0 \\ -h_m \end{bmatrix} v(k - \tau_u(k)) + \\
&\quad \begin{bmatrix} 0 \\ -\frac{h_m m g l}{J_p} \text{sen}(x_1(k)) \end{bmatrix} \\
y_d(k) &= [1 \quad 0] x_\theta(k) + w(k) \\
y(k) &= y_d(k - \tau_y(k))
\end{aligned} \tag{3.22}$$

Con base en 3.20 el control $u_{nom}(k)$ del sistema original 3.22 es:

$$\begin{aligned}
u_{nom}(k) &= -K_2 \hat{x}(k - \tau_y(k)) - K_3 \begin{bmatrix} u_{nom}(k-1) \\ \vdots \\ u_{nom}(k-10) \end{bmatrix} \\
&= -[k_1 \quad k_2] \hat{x}(k - \tau_y(k)) \\
&\quad - [k_3 \quad k_4 \quad k_5 \quad k_6 \quad k_7 \quad k_8 \quad k_9 \quad k_{10} \quad k_{11} \quad k_{12}] \begin{bmatrix} u_{nom}(k-1) \\ \vdots \\ u_{nom}(k-10) \end{bmatrix}
\end{aligned} \tag{3.23}$$

Donde $\hat{x}(k - \tau_y(k))$ se calcula de acuerdo 3.16 pero utilizando la salida $y(k)$ de 3.22. Note que 3.14 también se cumple en 3.22, por lo tanto, $\hat{x}(k - \tau_y(k))$ es una buena aproximación del estado $x_\theta(k - \tau_y(k))$ de 3.22. Finalmente con el control 3.23 el sistema

3. IMPLEMENTACIÓN DEL CONTROLADOR

3.22 en lazo cerrado queda:

$$\begin{aligned}
 x_\theta(k+1) &= \begin{bmatrix} 1 & h_m \\ 0 & 1 - \frac{bh_m}{J_p} \end{bmatrix} x_\theta(k) - \begin{bmatrix} 0 \\ -h_m \end{bmatrix} [k_1 \quad k_2] \hat{x}(k - \tau_\eta(k)) \\
 &\quad - \begin{bmatrix} 0 \\ -h_m \end{bmatrix} [k_3 \quad k_4 \quad k_5 \quad k_6 \quad k_7 \quad k_8 \quad k_9 \quad k_{10} \quad k_{11} \quad k_{12}] \begin{bmatrix} u_{nom}(k-1-\tau_u(k)) \\ u_{nom}(k-2-\tau_u(k)) \\ u_{nom}(k-3-\tau_u(k)) \\ u_{nom}(k-4-\tau_u(k)) \\ u_{nom}(k-5-\tau_u(k)) \\ u_{nom}(k-6-\tau_u(k)) \\ u_{nom}(k-7-\tau_u(k)) \\ u_{nom}(k-8-\tau_u(k)) \\ u_{nom}(k-9-\tau_u(k)) \\ u_{nom}(k-10-\tau_u(k)) \end{bmatrix} \\
 &\quad + \begin{bmatrix} 0 \\ -h_m \end{bmatrix} v(k - \tau_u(k)) + \begin{bmatrix} 0 \\ -\frac{h_m m g l}{J_p} \text{sen}(x_1(k)) \end{bmatrix} \\
 y(k) &= [1 \quad 0] x_\theta(k - \tau_y(k)) + w(k - \tau_y(k))
 \end{aligned} \tag{3.24}$$

Donde $\tau_\eta(k) = \tau_y(k) + \tau_u(k)$ y $\hat{x}(k - \tau_y)$ se calcula de acuerdo a 3.16. El siguiente paso es plantear al control correctivo $v(k)$ (2.24), el cual se propone como:

$$v(k) = \Phi \xi(k) + \Gamma \gamma_1(k) \tag{3.25}$$

Donde $\Gamma = 10$ y $\xi(k)$ es de acuerdo a:

$$\begin{cases} \xi(k) = (1 + h_m \phi) \xi(k-1) + h_m e(k-1), & \text{si } |e(k)| > 0.5 \left(\frac{\pi}{180} \right) [rad] \\ \xi(k) = 0, & \text{si } |e(k)| \leq 0.5 \left(\frac{\pi}{180} \right) [rad] \end{cases} \tag{3.26}$$

Donde $e(k)$ es el error del sistema de control (ecuación 2.29) y $\xi(0) = 0$. Para escoger el valor de las constantes Φ y ϕ se considera el siguiente sistema:

$$\begin{aligned}
 x_\theta(k+1) &= \left(\begin{bmatrix} 1 & h_m \\ 0 & 1 - \frac{bh_m}{J_p} \end{bmatrix} - \begin{bmatrix} 0 \\ -h_m \end{bmatrix} [k_1 \quad k_2] \right) x_\theta(k) + \begin{bmatrix} 0 \\ -h_m \end{bmatrix} \Phi \xi(k) \\
 y(k) &= [1 \quad 0] x_\theta(k)
 \end{aligned} \tag{3.27}$$

Por lo tanto, se obtiene el siguiente sistema aumentado:

$$\begin{aligned}
 x_\theta(k+1) &= \left(\begin{bmatrix} 1 & h_m \\ 0 & 1 - \frac{bh_m}{J_p} \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ -k_1 h_m & -k_2 h_m \end{bmatrix} \right) x_\theta(k) + \begin{bmatrix} 0 \\ -h_m \end{bmatrix} \Phi \xi(k) \\
 \xi(k+1) &= (1 + \phi h_m) \xi(k) + h_m e(k)
 \end{aligned} \tag{3.28}$$

Como $e(k) = -y(k) = -Cx_\theta(k)$ se tiene:

$$\begin{aligned} x_\theta(k+1) &= \begin{bmatrix} 1 & h_m \\ k_1 h_m & 1 - \frac{bh_m}{J_p} + k_2 h_m \end{bmatrix} x_\theta(k) + \begin{bmatrix} 0 \\ -h_m \end{bmatrix} \Phi \xi(k) \\ \xi(k+1) &= (1 + \phi h_m) \xi(k) - h_m [1 \quad 0] x_\theta(k) \end{aligned} \quad (3.29)$$

Expresando 3.29 de forma matricial:

$$\begin{aligned} \begin{bmatrix} x_\theta(k+1) \\ \xi(k+1) \end{bmatrix} &= \begin{bmatrix} 1 & h_m & 0 \\ k_1 h_m & 1 - \frac{bh_m}{J_p} + k_2 h_m & -h_m \Phi \\ -h_m & 0 & 1 + \phi h_m \end{bmatrix} \begin{bmatrix} x_\theta(k) \\ \xi(k) \end{bmatrix} \\ &= A_{aum2} \begin{bmatrix} x_\theta(k) \\ \xi(k) \end{bmatrix} \end{aligned} \quad (3.30)$$

Se puede comprobar que con $\Phi = -5$ y $\phi = -3$ la matriz A_{aum2} es Schur, con lo cual queda definido el control $\Phi \xi(k)$ de 3.25.

El siguiente paso del diseño del controlador neuronal predictivo es el control basado en redes neuronales. De 3.25 se observa que la red neuronal propuesta es un perceptrón, por lo tanto, la matriz de parámetros de la red neuronal tiene solo un renglón, es decir, $W(k) = [w_{11}(k) \quad w_{12}(k)] \in \mathbb{R}^{1 \times 2}$. $w_{11}(0)$ y $w_{12}(0)$ se escogieron en el rango $[-0.5, 0.5]$, más específicamente, $w_{11}(0) = 0.44$ y $w_{12}(0) = 0.09$. Por otra parte, se escogió $\varphi_1(\cdot) = 5 \tanh(0.05 \cdot)$. Finalmente, para aprender la tasa de aprendizaje $\eta_1(k)$, se escogió la siguiente regla de aprendizaje y condiciones iniciales:

$$\begin{aligned} \eta_1(k) &= \eta_1(k-1) - \frac{|y(k)|}{2} \eta_1^2(k-1) \\ \eta_1(0) &= 0.90 \end{aligned} \quad (3.31)$$

A continuación se construye el modelo oculto de Markov para la predicción de la salida del sistema. Los espacios de estados de las variables aleatorias $X(k)$ y $Y(k)$ son obtenidos por un muestreo de las variables $h(x_\theta(k - T_y))$ y $y(k)$. En este caso se tiene que $h(x_\theta(k - T_y)) = x_1(k - 5)$, pues como se mencionó anteriormente, con fines de obtener resultados de simulaciones se considera que $T_y = 5$ y de 3.9 se tiene que $h(x_\theta(k)) = x_1(k)$. Por lo

3. IMPLEMENTACIÓN DEL CONTROLADOR

tanto, los espacios de estados se definen como:

$$\begin{aligned}
 X &= \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\} \\
 Y &= \{y_1, y_2, y_3, y_4, y_5, y_6, y_7\} \\
 x_1 &= \{x_1(k-5) | x_1(k-5) \in \left(\frac{\pi}{180}\right) (-\infty, -10)\} \\
 x_2 &= \{x_1(k-5) | x_1(k-5) \in \left(\frac{\pi}{180}\right) [-10, -6]\} \\
 x_3 &= \{x_1(k-5) | x_1(k-5) \in \left(\frac{\pi}{180}\right) [-6, -3]\} \\
 x_4 &= \{x_1(k-5) | x_1(k-5) \in \left(\frac{\pi}{180}\right) [-3, 3]\} \\
 x_5 &= \{x_1(k-5) | x_1(k-5) \in \left(\frac{\pi}{180}\right) [3, 6]\} \\
 x_6 &= \{x_1(k-5) | x_1(k-5) \in \left(\frac{\pi}{180}\right) [6, 10]\} \\
 x_7 &= \{x_1(k-5) | x_1(k-5) \in \left(\frac{\pi}{180}\right) [10, \infty)\} \\
 y_1 &= \{y(k) | y(k) \in \left(\frac{\pi}{180}\right) (-\infty, -10)\} \\
 y_2 &= \{y(k) | y(k) \in \left(\frac{\pi}{180}\right) [-10, -6]\} \\
 y_3 &= \{y(k) | y(k) \in \left(\frac{\pi}{180}\right) [-6, -3]\} \\
 y_4 &= \{y(k) | y(k) \in \left(\frac{\pi}{180}\right) [-3, 3]\} \\
 y_5 &= \{y(k) | y(k) \in \left(\frac{\pi}{180}\right) [3, 6]\} \\
 y_6 &= \{y(k) | y(k) \in \left(\frac{\pi}{180}\right) [6, 10]\} \\
 y_7 &= \{y(k) | y(k) \in \left(\frac{\pi}{180}\right) [10, \infty)\}
 \end{aligned} \tag{3.32}$$

Para la salida ponderada se utilizaron los valores $v_i, i = 1..7$, de acuerdo a 2.78, excepto por los siguientes dos valores:

$$\begin{aligned}
 v_1 &= -16 \\
 v_7 &= 16
 \end{aligned} \tag{3.33}$$

Con base en el conocimiento previo del sistema se escogieron las condiciones iniciales 3.34 para las matrices $M(k)$ y $N(k)$. Más específicamente, las condiciones iniciales iguales a 6 se escogieron así debido a que se conjeturan son las transiciones entre estados (de las variables aleatorias $X(k)$ y $Y(k)$) más probables de suceder en el sistema real; las condiciones iniciales iguales a 2, por su parte, representan a transiciones que se conjeturan son poco probables de suceder y, finalmente, las condiciones iniciales iguales a 0 son transiciones que no pueden suceder en el sistema real. De esta manera, por ejemplo, si $X(0) = x_2$ se considera que lo más probable es que $Y(0) = y_2$, por otra parte, se considera que es poco probable que $Y(0) = y_1$ ó $Y(0) = y_3$ y que es imposible

que $Y(0) = y_4$, $Y(0) = y_5$, $Y(0) = y_6$ ó $Y(0) = y_7$. Estas selecciones son razonables debido a que el retardo de observación es pequeño y el rango que abarca cada estado de los espacios de estados, \mathbb{X} y \mathbb{Y} , se puede considerar amplio. A pesar de que la selección de las condiciones iniciales 3.34 se realizó de manera arbitraria, con el algoritmo de aprendizaje de las matrices $A(k)$ y $B(k)$ los valores de las probabilidades de transición y observación se actualizan en línea y con base en datos a un valor óptimo que describe la naturaleza estocástica del sistema.

$$\begin{aligned}
m_{11}(0) &= m_{22}(0) = m_{33}(0) = m_{44}(0) = m_{55}(0) = m_{66}(0) = m_{77}(0) = 6 \\
m_{12}(0) &= m_{21}(0) = m_{23}(0) = m_{32}(0) = m_{34}(0) = m_{43}(0) = m_{45}(0) \\
&= m_{54}(0) = m_{56}(0) = m_{65}(0) = m_{67}(0) = m_{76}(0) = 2 \\
m_{13}(0) &= m_{14}(0) = m_{15}(0) = m_{16}(0) = m_{17}(0) = m_{24}(0) = m_{25}(0) \\
&= m_{26}(0) = m_{27}(0) = m_{31}(0) = m_{35}(0) = m_{36}(0) = m_{37}(0) \\
&= m_{41}(0) = m_{42}(0) = m_{46}(0) = m_{47}(0) = m_{51}(0) = m_{52}(0) \\
&= m_{53}(0) = m_{57}(0) = m_{61}(0) = m_{62}(0) = m_{63}(0) = m_{64}(0) \\
&= m_{71}(0) = m_{72}(0) = m_{73}(0) = m_{74}(0) = m_{75}(0) = 0 \\
n_{11}(0) &= n_{22}(0) = n_{33}(0) = n_{44}(0) = n_{55}(0) = n_{66}(0) = n_{77}(0) = 6 \\
n_{12}(0) &= n_{21}(0) = n_{23}(0) = n_{32}(0) = n_{34}(0) = n_{43}(0) = n_{45}(0) \\
&= n_{54}(0) = n_{56}(0) = n_{65}(0) = n_{67}(0) = n_{76}(0) = 2 \\
n_{13}(0) &= n_{14}(0) = n_{15}(0) = n_{16}(0) = n_{17}(0) = n_{24}(0) = n_{25}(0) \\
&= n_{26}(0) = n_{27}(0) = n_{31}(0) = n_{35}(0) = n_{36}(0) = n_{37}(0) \\
&= n_{41}(0) = n_{42}(0) = n_{46}(0) = n_{47}(0) = n_{51}(0) = n_{52}(0) \\
&= n_{53}(0) = n_{57}(0) = n_{61}(0) = n_{62}(0) = n_{63}(0) = n_{64}(0) \\
&= n_{71}(0) = n_{72}(0) = n_{73}(0) = n_{74}(0) = n_{75}(0) = 0
\end{aligned} \tag{3.34}$$

Estos valores son muy importantes pues, como se explicó en la sección 2.4, van a determinar el valor inicial ($A(1)$ y $B(1)$) de las matrices $A(k)$ y $B(k)$, tal como establece 2.58, por lo tanto, va a depender de estos valores que desde el inicio del experimento se pueda obtener una buena predicción. También se propone la siguiente distribución de probabilidad inicial para $X(k)$:

$$P(X(0)) = [1/14 \quad 1/14 \quad 1/14 \quad 1/14 \quad 1/14 \quad 1/14 \quad 4/7] \tag{3.35}$$

También se escogió una condición inicial para $X^*(0)$, más específicamente, $X^*(0) = x_7 \in \mathbb{X}$. Dado $X^*(0)$ es posible dar inicio al algoritmo de aprendizaje en línea para determinar a las matrices $A(k) \in \mathbb{R}^{7 \times 7}$ y $B(k) \in \mathbb{R}^{7 \times 7}$ con base en datos, por lo tanto se conocen todos los parámetros de la tabla 2.1 y el HMM queda especificado.

Para el sistema de predicción se adopta una ventana de tiempo deslizante de tamaño 10. Además, debido a que $T_y = T_u = 5$ se calcula al vector de probabilidad $P(Y(k+10)|O_{10} = \{o_{k-9}, o_{k-8}, \dots, o_{k-1}, o_k\}) \in \mathbb{R}^{1 \times 7}$. Con el algoritmo de avance se calcula

3. IMPLEMENTACIÓN DEL CONTROLADOR

$P(X(k) = x_p, o_{k-9}, o_{k-8}, \dots, o_k), p = 1 \dots 7$, para lo cual en cada iteración, $k = 1, \dots, \infty$, se tiene que calcular una matriz $\alpha \in \mathbb{R}^{7 \times 10}$, donde cada componente α_{pt} es de acuerdo a la ecuación 3.36.

$$\alpha_{pt} = \begin{cases} P(X(k-10) = x_p) b_{pq}(k), & \text{si } t = 1 \\ \left(\sum_{s=1}^{N_1} \alpha_{s(t-1)} a_{sp}(k) \right) b_{pq}(k), & \text{si } 2 \leq t \leq 10, \end{cases} \quad p = 1 \dots 7 \quad (3.36)$$

Donde q es de acuerdo a 2.82 y se tienen las condiciones iniciales:

$$\begin{aligned} o_{-8} = o_{-7} = o_{-6} = \dots = o_{-5} = o_0 = y_i \in \mathbb{Y} \\ P(X(-9)) = P(X(-8)) = \dots = P(X(-1)) = P(X(0)) \end{aligned} \quad (3.37)$$

Donde $P(X(0))$ es de acuerdo a 3.35 y $y_i = y_7 \in \mathbb{Y}$ se escogió de acuerdo al conocimiento previo del sistema, más específicamente, de acuerdo a las condiciones iniciales de este para obtener una buena predicción desde el inicio. Con las condiciones iniciales 3.37 es posible comenzar el algoritmo de avance y calcular en cada iteración del controlador $P(X(k) = x_j, O_{10}), j = 1, \dots, 7$. Una vez calculados estos valores con las ecuaciones 2.66, 2.67, 2.68 y 2.72 se calcula $P(Y(k+10)|O_{10})$. Por consiguiente, se predice la salida del sistema 10 periodos de muestreo adelante ($\hat{y}(k+10)$) dadas las últimas 10 observaciones de acuerdo a 2.77. Finalmente, con las ecuaciones 2.85, 2.86 y 2.87 se establece la ley del controlador neuronal predictivo para el ángulo *pitch*.

Como se puede apreciar en 3.5 y en la tabla 3.1 la dinámica del ángulo *roll* tiene la misma estructura y mismos valores numéricos que el ángulo *pitch* (sistema 3.4), por lo tanto la implementación del controlador neuronal predictivo es igual para el ángulo *roll*.

Empezando con la implementación para el ángulo *yaw* (sistema 3.6), con el método de Euler (dado el mismo h_m que para el *pitch*) se encontraron las siguientes ecuaciones en diferencias que modelan la dinámica del ángulo 3.6 en tiempo discreto.

$$\begin{aligned} x_\psi(k+1) &= \begin{bmatrix} x_1(k) + h_m x_2(k) \\ \left(1 - \frac{bh_m}{J_y}\right) x_2(k) - h_m U_3(k) \end{bmatrix} \\ z(k) &= [1 \quad 0] x_\psi(k) \end{aligned} \quad (3.38)$$

Como una observación importante, utilizando la función *c2d* del software *Matlab* (la cual utiliza el método del retenedor de orden cero y es solo útil para sistemas lineales) se obtiene exactamente la misma discretización que se obtuvo con el método de Euler para el sistema 3.6. Agregando a 3.38 el retardo de comunicación estocástico y ruido de observación se obtiene:

$$\begin{aligned} x_\psi(k+1) &= \begin{bmatrix} 1 & h_m \\ 0 & 1 - \frac{bh_m}{J_y} \end{bmatrix} x_\psi(k) + \begin{bmatrix} 0 \\ -h_m \end{bmatrix} u(k - \tau_u(k)) \\ y(k) &= [1 \quad 0] x_\psi(k - \tau_y(k)) + w(k - \tau_y(k)) \end{aligned} \quad (3.39)$$

Con base en 3.39, la ecuación 3.40 define al sistema nominal.

$$\begin{aligned} x_\psi(k+1) &= \begin{bmatrix} 1 & h_m \\ 0 & 1 - \frac{bh_m}{J_y} \end{bmatrix} x_\psi(k) + \begin{bmatrix} 0 \\ -h_m \end{bmatrix} u_{nom_1}(k - T_u) \\ y(k) &= [1 \ 0] x_\psi(k - T_y) \end{aligned} \quad (3.40)$$

Nuevamente se utilizan los mismo valores T_u y T_y que se utilizaron para el *pitch*. Note que el sistema nominal del *pitch* 3.11 y 3.40 son idénticos excepto por que J_y y J_p no son iguales. También note los siguientes valores propios:

$$\begin{aligned} \lambda_1 \left(\begin{bmatrix} 1 & h_m \\ 0 & 1 - \frac{bh_m}{J_y} \end{bmatrix} \right) &= \lambda_1(A_d) = 1 \\ \lambda_2 \left(\begin{bmatrix} 1 & h_m \\ 0 & 1 - \frac{bh_m}{J_y} \end{bmatrix} \right) &= \lambda_2(A_d) = 0.9953 \end{aligned} \quad (3.41)$$

Por lo tanto, el origen de 3.40, al igual que el origen del sistema nominal 3.11, es estable aunque no asintóticamente. El objetivo del control $u_{nom_1}(k)$ es hacer que el origen de 3.40 sea asintóticamente estable. De hecho, el origen del sistema original 3.6 (la dinámica del *yaw* en tiempo continuo) también es estable (no asintóticamente) pues 3.6 es un sistema lineal con los siguientes valores propios:

$$\begin{aligned} \dot{x}_\psi(t) &= \begin{bmatrix} 0 & 1 \\ 0 & \frac{-b}{J_y} \end{bmatrix} x_\psi(t) - \begin{bmatrix} 0 \\ 1 \end{bmatrix} U_3 \\ \lambda_1 \left(\begin{bmatrix} 0 & 1 \\ 0 & \frac{-b}{J_y} \end{bmatrix} \right) &= \lambda_1(A) = 0 \\ \lambda_2 \left(\begin{bmatrix} 0 & 1 \\ 0 & \frac{-b}{J_y} \end{bmatrix} \right) &= \lambda_2(A) = -0.7273 \end{aligned} \quad (3.42)$$

Note que a pesar de haber utilizado el método de Euler (una aproximación) para obtener el modelo en tiempo discreto 3.38 a partir del modelo en tiempo continuo 3.6 y no la solución exacta (solo para sistemas lineales): $A_d = e^{Ah_m}$, se sigue cumpliendo el mapeo de polos de tiempo continuo a tiempo discreto :

$$\begin{aligned} \lambda_1(A_d) &= e^{\lambda_1(A)h_m} = e^0 = 1 \\ \lambda_2(A_d) &= e^{\lambda_2(A)h_m} = e^{(-0.7273)(0.0065)} = 0.9953 \end{aligned} \quad (3.43)$$

Por lo tanto se puede concluir que el método de Euler es buena aproximación. Continuando con el diseño del controlador, se puede comprobar que el control 3.23 que estabiliza asintóticamente al sistema nominal del *pitch* también estabiliza asintóticamente a 3.40. Por lo tanto, el control $u_{nom}(k)$ (ecuación 3.23) del *pitch* es también el control nominal para el *yaw*.

3. IMPLEMENTACIÓN DEL CONTROLADOR

Debido a que el control $u_{nom}(k)$ de ambos ángulos es igual, el sistema aumentado para diseñar el control *reset* (3.30) también es igual para el *yaw* excepto por que J_y y J_p no son iguales. Sin embargo, se puede comprobar que se puede diseñar de la misma manera el control $\Phi\xi(k)$ de 2.4 para el *yaw*, por lo tanto, también se escogieron $\Phi = -5$ y $\phi = -3$. Para el control basado en redes neuronales se propone exactamente el mismo que para el *pitch*, es decir, un perceptrón. Finalmente, para el sistema de predicción se escogen los mismos espacios de estados 3.32, las mismas condiciones iniciales (3.34, 3.35, $X^*(0)$ y 3.37) y la misma ventana de tiempo deslizante. Con lo cual queda concluida la implementación del controlador neuronal predictivo propuesto al sistema 3.1 tipo cuadricóptero.

Antes de pasar al último capítulo se dan unos últimos comentarios sobre la implementación del control neuronal predictivo al sistema 3.1. Primero se discretizó el modelo en tiempo continuo utilizando el método de Euler. Con base en este modelo discreto se diseñó un control nominal considerando un retardo de tiempo constante (5 periodos de muestreo) y un sistema linealizado para el caso del *pitch*. En este caso, se consideró que no era necesario diseñar el observador de Luenberger pues la ecuación 3.16 es una aproximación bastante buena para estimar el estado a partir de la salida del sistema. Por la tanto, el diseño del controlador nominal fue sencillo, simplemente se planteó un sistema aumentado y se estabilizó por medio del control lineal de retroalimentación de estados y colocación de polos. Posteriormente, se planteó un control basado en *reset*, nuevamente para el diseño de este se debe plantear un sistema aumentado y estabilizarlo por medio del control de sistemas lineales discretos.

A continuación se propuso un control basado en un perceptrón con aprendizaje en línea; para este control hay que proponer las condiciones iniciales de la matriz de parámetros ($w_{11}(0)$ y $w_{12}(0)$), la condición inicial de la tasa de aprendizaje ($\eta_1(0)$), la función sigmoideal ($\varphi_1(\cdot)$) y a la función que cambia a la tasa de aprendizaje ($\epsilon(e(k))$ de 2.46). Todos estos parámetros y funciones se deben escoger por medio de la experimentación. Sin embargo, gracias a que la red neuronal presenta adaptabilidad, no hace falta hacer muchos experimentos para dar con una configuración en la que se tengan significativas mejoras en el desempeño, por lo que también se puede considerar un control fácil de implementar.

Finalmente, se planteó la estructura de un HMM, básicamente se escogieron los espacios de estados de las variables aleatorias con base en el conocimiento previo del rango de movimiento del sistema. También con base, en este conocimiento previo se escogió una distribución de probabilidad inicial para $X(k)$, de hecho, en el marco de referencia de la inferencia Bayesiana a este término se le conoce como el conocimiento *a priori*. Esta distribución de probabilidad inicial es importante para dar una buena predicción desde el inicio del experimento. La condición inicial para $X^*(k)$ también es fundamental que se elija con base en el conocimiento previo de las condiciones iniciales del sistema para asegurar un buen aprendizaje de los parámetros del HMM.

Por último, se escogieron condiciones iniciales para las matrices $M(k)$ y $N(k)$ (3.34). Estas condiciones iniciales se escogieron con base en el conocimiento previo del sistema, especialmente con base en el conocimiento previo concerniente al retardo de tiempo y al periodo de muestreo, pues de estos valores depende la relación (2.47) entre las variables aleatorias ocultas y las observables. También se escogieron las condiciones iniciales del método de avance (3.37), las cuales nuevamente es recomendable escogerlas de acuerdo al conocimiento previo de las condiciones iniciales del sistema para obtener una buena predicción desde el inicio. En el siguiente capítulo se presentan los resultados de las simulaciones y las conclusiones del presente trabajo.

Resultados

En esta capítulo se presentan los resultados de las simulaciones de la implementación del controlador neuronal predictivo propuesto al sistema dinámico continuo no lineal MIMO tipo cuadricóptero 3.1. Para las simulaciones se consideró un periodo de muestreo, un retardo de observación y un retardo de control distribuidos normalmente y de acuerdo a las definiciones 2, 3 y 4. También se consideró ruido blanco Gaussiano de acuerdo a 1.2. Los valores numéricos de los parámetros utilizados en la simulaciones se muestran en la tabla 4.1. Con el objetivo de que las figuras haya menos datos y por lo tanto sean más fáciles de comprender, los resultados se dividen en dos secciones; en la sección 4.1 se muestran los resultados del ángulo del *pitch* (sistema 3.4), mientras que la sección 4.2 se muestran los resultados del ángulo del *yaw* (sistema 3.6).

Nombre del parámetro	Símbolo	Valor numérico/Unidades
Valor esperado del periodo de muestreo	h_m	0.0065 [seg]
Desviación estándar del periodo de muestreo	σ_{h_m}	0.003 [seg]
Valor esperado del retardo de observación	$E(\tau_{y(s_k)})$	0.028 [seg]
Desviación estándar del retardo de observación	σ_{τ_y}	0.0032 [seg]
Valor esperado del retardo de control	$E(\tau_{u(t_k)})$	0.028 [seg]
Desviación estándar del retardo de control	σ_{τ_u}	0.0032 [seg]
Valor esperado del ruido de medición	w	0
Desviación estándar del ruido de medición	σ_w	$0.5 \left(\frac{\pi}{180}\right)$ [rad]

Tabla 4.1: Parámetros utilizados en las simulaciones.

En las figuras de este capítulo se muestran las salidas del sistema 3.1 al llegar controlador, es decir, las salidas después del muestreador, con ruido de medición y después

4. RESULTADOS

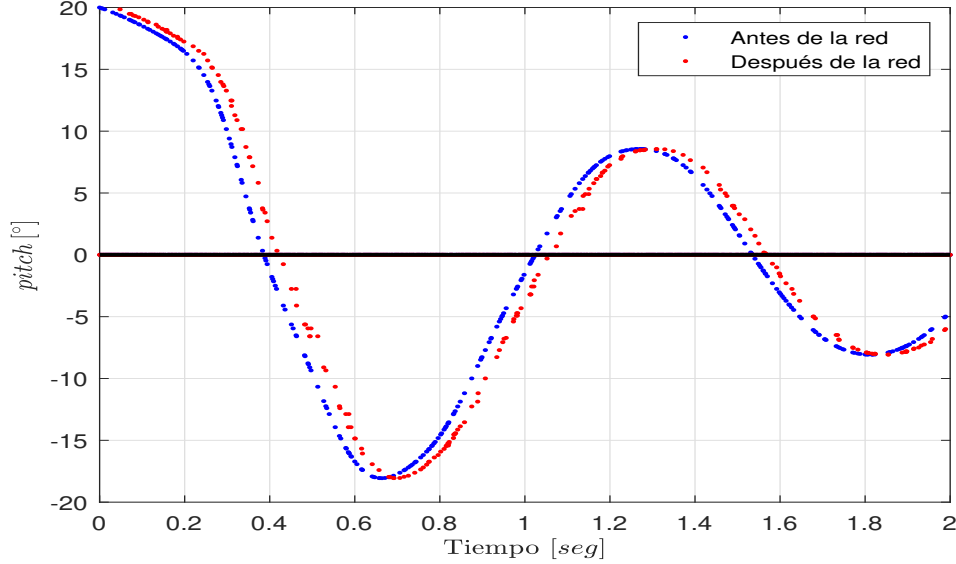


Figura 4.1: Salida del sistema (*pitch*) antes y después de la red de comunicación, sin ruido de medición y con el controlador nominal.

de la red de comunicación, es decir, con retardo de observación. En otras palabras, se muestra $y_c(t)$ de la figura 1.1, es decir, la evolución en tiempo continuo de $y(k)$ de la figura 2.6 y de las ecuaciones 3.9 y 3.39.

Para ejemplificar lo anterior, en la figura 4.1 se muestra la salida del sistema 3.4 (ángulo del *pitch*) al llegar al controlador (después de la red de comunicación), durante 2 segundos de simulación. Para esta simulación el sistema 3.4 tiene condiciones iniciales $x_1(0) = 20^\circ$ y $x_2(0) = -12^\circ$ y el controlador se compone únicamente de la ley del control nominal $u_{nom}(k)$ (ecuación 3.23). Además, para ejemplificar como funciona el retardo, en esta figura también se muestra la salida del sistema 3.4 después del muestreador y antes de la red de comunicación, es decir, sin retardo de observación; en un NCS real no se tiene acceso a esta señal (únicamente después de la red de comunicación), sin embargo, en el caso de simulaciones sí se puede conocer y sirve para mostrar como funciona el retardo de observación estocástico. Para que quede más claro como funciona el retardo, en esta simulación no se consideró al ruido de medición.

Como se puede apreciar en la figura 4.1 y como se explicó en la sección 1.5.1, la salida de un NCS al llegar al controlador es discontinua, de hecho, es una sucesión de puntos aislados, por lo que ni siquiera es continua a trazos. Asimismo, se aprecia que el instante que pasa entre dos muestras (el periodo de muestreo) no es constante y se puede observar que la salida al llegar al controlador está atrasada respecto a la salida después del muestreador, sin embargo, por momentos está más atrasada y por momentos menos,

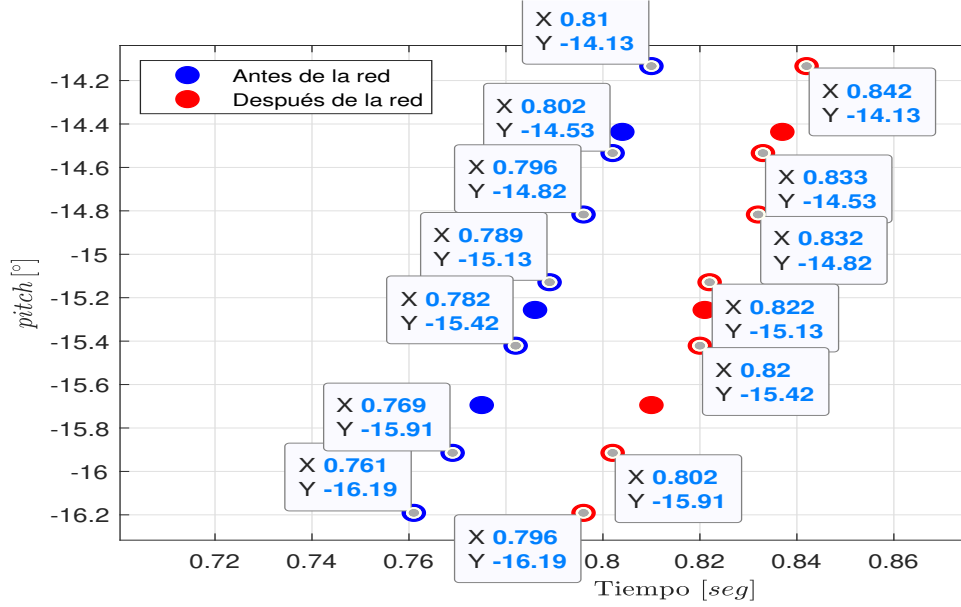


Figura 4.2: Acercamiento de la salida del sistema (*pitch*) antes y después de la red de comunicación, sin ruido de medición y con el controlador nominal.

lo cual se debe al retardo de observación estocástico utilizado en las simulaciones y de acuerdo al valor de la tabla 4.1. La figura 4.2 es un acercamiento de la figura 4.1, donde se especifica el valor de la salida $z(s_k)$ y el instante de muestreo s_k (definición 1) de 10 muestras. Con base en la información de esta figura, en la tabla 4.2 se especifica el valor del retardo de observación $\tau_{y(s_k)}$ (definición 3) y el periodo de muestreo T_k (definición 2) de cada muestra. Calculando los promedios ($E(T_k)$ y $E(\tau_{y(s_k)})$) y las desviaciones estándar (σ_{T_k} y $\sigma_{\tau_{y(s_k)}}$) del periodo de muestreo y el retardo de observación a partir de los valores de la tabla 4.1 se obtuvieron los siguientes resultados:

$$\begin{aligned}
 E(T_k) &= 0.0052[seg] \\
 \sigma_{T_k} &= 0.002[seg] \\
 E(\tau_{y(s_k)}) &= 0.0341[seg] \\
 \sigma_{\tau_{y(s_k)}} &= 0.002[seg]
 \end{aligned} \tag{4.1}$$

Como se esperaba, los valores obtenidos son similares a los de la tabla 4.1. Con este ejemplo de 10 muestras se mostró como funcionan el retardo y el periodo de muestreo estocásticos en las simulaciones realizadas para demostrar la efectividad del controlador neuronal predictivo, lo cual se asemeja a como funciona un NCS real.

Por último, a pesar de que las señales después del muestreador son discontinuas, con fines didácticos en el resto de las figuras de este capítulo las salidas del sistema 3.1 al llegar al controlador se mostrarán constantes e iguales al último dato leído hasta que

4. RESULTADOS

Instante de muestreo [seg] (s_k)	$z(s_k)$ [°]	Periodo de muestreo [seg] ($T_k = s_{k+1} - s_k$)	Retardo de observación [seg] (τ_{y_k})
0.761	-16.19	0.008 = 0.769 - 0.761	0.035 = 0.796 - 0.761
0.769	-15.91	0.006 = 0.775 - 0.769	0.033 = 0.802 - 0.769
0.775	-15.69	0.007 = 0.782 - 0.775	0.035 = 0.810 - 0.775
0.782	-15.42	0.004 = 0.786 - 0.782	0.038 = 0.820 - 0.782
0.786	-15.26	0.003 = 0.789 - 0.786	0.035 = 0.821 - 0.786
0.789	-15.13	0.007 = 0.796 - 0.789	0.033 = 0.822 - 0.789
0.796	-14.82	0.006 = 0.802 - 0.796	0.036 = 0.832 - 0.796
0.802	-14.53	0.002 = 0.804 - 0.802	0.031 = 0.833 - 0.802
0.804	-14.44	0.006 = 0.810 - 0.804	0.033 = 0.837 - 0.804
0.810	-14.13	0.003 = 0.813 - 0.810	0.032 = 0.842 - 0.810

Tabla 4.2: Comportamiento del retardo de observación y el periodo de muestreo durante 10 muestras.

llegue un nuevo dato al controlador, por lo tanto, las salidas aparecerán como señales continuas a trazos. Para las salidas del controlador, es decir, $u_c(t)$ en la figura 1.1 o $u(k)$ en la figura 2.6 y en las ecuaciones 3.9 y 3.39, se hace la misma consideración y por lo tanto también se muestran como señales continuas a trazos.

4.1. Pitch

En esta sección se muestran los resultados de las simulaciones de la implementación del controlador neuronal predictivo al sistema dinámico no lineal en tiempo continuo 3.4 de acuerdo a lo desarrollado en el capítulo 3. Las condiciones iniciales del sistema 3.4 se escogieron como $x_1(0) = 20^\circ$ y $x_2(0) = -12^\circ$. Como se explicó anteriormente, en las figuras de esta sección se muestra la salida del sistema 3.4 al llegar al controlador; llamaremos a esta señal $y_\theta(t)$ en el resto de la tesis. En las primeras simulaciones no se consideró al ruido de medición para dar un mayor énfasis a contrarrestar las perturbaciones causadas por el retardo de comunicación estocástico.

En la figura 4.3 se muestra $y_\theta(t)$ durante 12 segundos de simulación cuando el controlador se compone únicamente de la ley de control nominal $u_{nom}(k)$ (ecuación 3.23) considerando dos casos: el determinístico y el estocástico. En el caso estocástico se con-

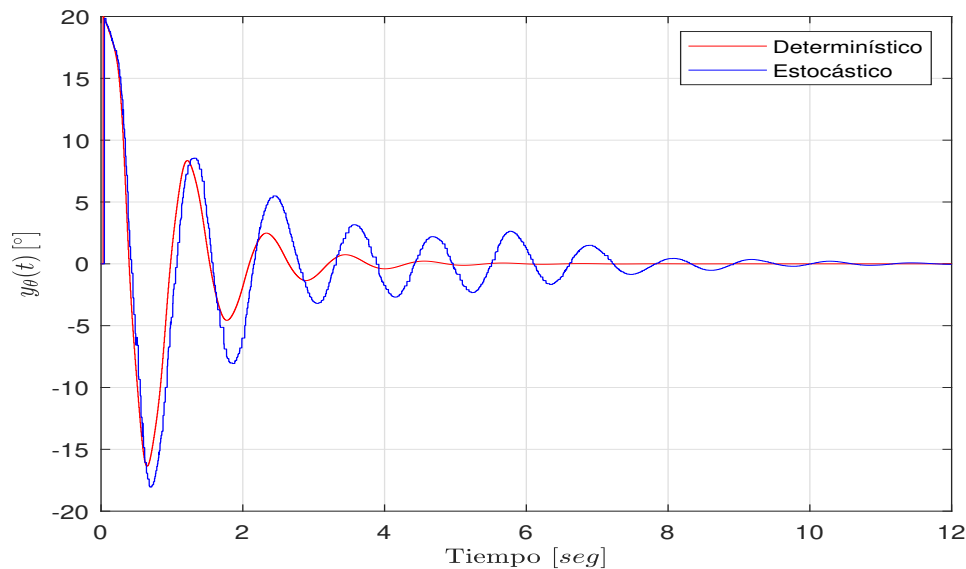


Figura 4.3: Respuesta del sistema en los casos determinístico (retardo y periodo de muestreo constantes) y estocástico (retardo y periodo de muestreo aleatorios), sin ruido de medición y con el controlador nominal.

sidera al retardo y al periodo de muestreo de acuerdo a los valores de la tabla 4.1, mientras que en el determinístico se considera el periodo de muestreo constante e igual a $0.006[seg]$ y los retardos de observación y de control también se consideran constantes e iguales a $0.03[seg]$ (5 periodos de muestreo).

Como se puede observar en la figura 4.3, a pesar de que en el caso determinístico el retardo es más grande que $E(\tau_{u(t_k)})$ y $E(\tau_{y(s_k)})$, el retardo de comunicación y el periodo de muestreo aleatorios afectan significativamente el desempeño del controlador nominal, sin embargo, este controlador sigue siendo capaz de cumplir el objetivo de control planteado a pesar de estos factores aleatorios. Además note que ambas señales inician en cero. Esto es debido a que al inicio de la simulación el controlador no recibe ningún dato hasta que pasa el tiempo equivalente a un retardo de observación; por lo tanto, este comportamiento se presentará en todas las figuras por el resto del capítulo.

A partir de este punto se va a dejar de considerar al caso determinístico y únicamente se va a considerar al estocástico. En la figura 4.4 se muestra $y_\theta(t)$ considerando dos casos. El primer caso es el caso estocástico de la figura 4.3. En el segundo caso en el controlador se tiene la siguiente ley de control:

$$u(k) = u_{nom}(k) + \Phi \xi(k) \quad (4.2)$$

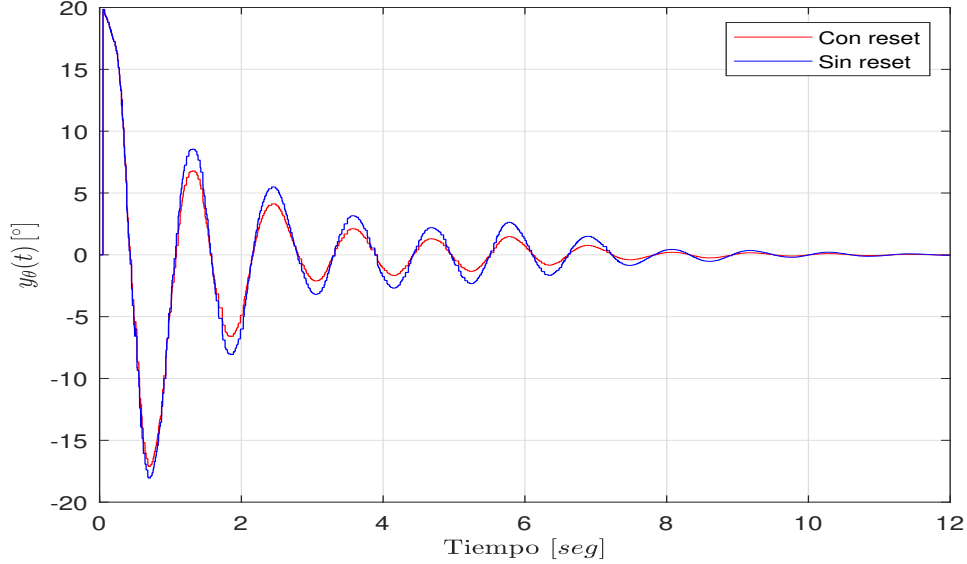


Figura 4.4: Respuesta del sistema en el caso estocástico (retardo y periodo de muestreo aleatorios) con y sin el control *reset*, sin ruido de medición y con el controlador nominal.

Donde $u_{nom}(k)$ es la ley de control nominal 3.23, $\Phi = -5$, $\xi(k)$ es de acuerdo a 3.26 con $\phi = -3$, es decir, se considera el control nominal y el control correctivo $v(k)$ 3.25 con $\Gamma\gamma_1(k) = 0$. En otras palabras, en el segundo caso se considera al control basado en *reset* sin considerar al control basado en redes neuronales.

Como se puede observar en la figura 4.4, al agregar el control basado en *reset* se tiene que el sistema sigue presentando las mismas oscilaciones, sin embargo, estas son de menor amplitud y por lo tanto se tiene un menor sobrepaso y menor tiempo de asentamiento. En la figura 4.5 se aprecian las evoluciones en tiempo continuo de las salidas de control totales calculadas en el controlador para ambos casos ($u(k)$ en la ecuación 3.9). Como se puede apreciar en la figura 4.5 la señal de control con el control *reset* es más pequeña en magnitud en comparación con la señal del control nominal. Esta diferencia causa que las oscilaciones sean más grandes si no se utiliza el control *reset*.

En la figura 4.6 se observar claramente el comportamiento basado en *reset* de la salida del controlador $\Phi\xi(k)$ de 4.2. Comparando las figuras 4.5 y 4.6, se puede apreciar que la salida del controlador *reset* siempre tiene el signo contrario a la salida del controlador nominal, es por eso que se reduce la magnitud de la señal de control utilizando el control *reset* y por lo tanto se reducen las oscilaciones en la respuesta del sistema. Note que la salida del controlador *reset* siempre mantiene el signo contrario a la del nominal gracias al comportamiento basado en reseteo, sin este comportamiento la señal de control total se haría más grande en lugar de reducir y por lo tanto se presentarían

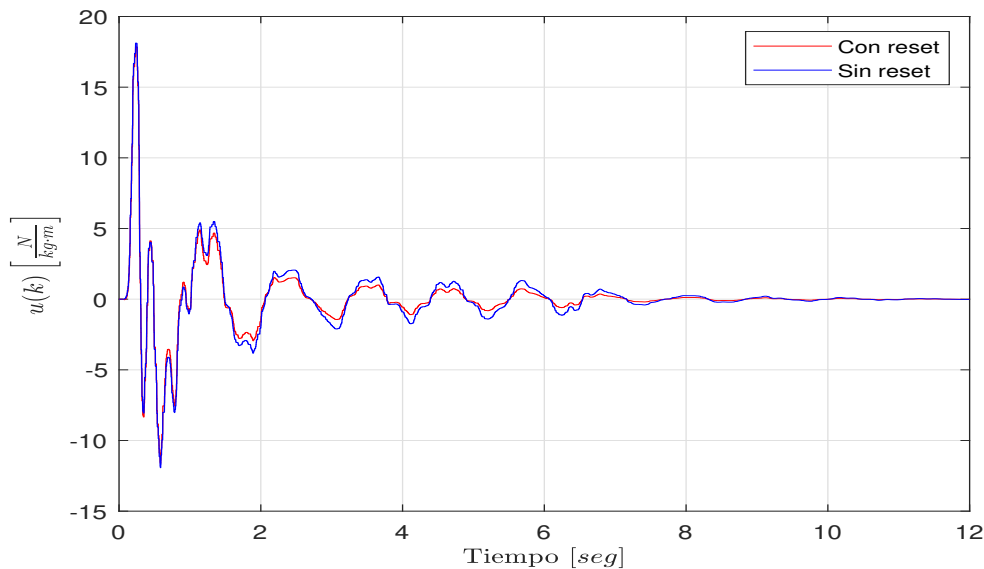


Figura 4.5: Salida del controlador nominal con y sin el control *reset*.

mayores oscilaciones.

En la figura 4.7 se muestra $y_\theta(t)$ durante 20 segundos de simulación en dos casos. El primer caso es el segundo caso de la figura 4.4, es decir, se considera ley de control 4.2. En el segundo caso se considera la ley de control 2.4 con el control correctivo $v(k)$ de 3.25. En otras palabras, el segundo caso es el primer caso con la adición del control basado en redes neuronales ($\Gamma\gamma_1(k)$) y aún sin utilizar la predicción hecha con el HMM. Para probar la robustez ante perturbaciones del sistema, a los 10 segundos se aplicó una perturbación, más específicamente, a los 10 segundos se le sumó una función de tipo impulso a la entrada de control $u(k)$ de 2.4.

Como se puede apreciar en la figura 4.7 la respuesta sin el control basado en redes neuronales presenta oscilaciones más grandes con un pico de sobrepaso de más de 17° y tiempo de asentamiento de aproximadamente 8 segundos. La respuesta con el control neuronal tiene un sobrepaso más pequeño (pico de menos de 8°) y tiempo de asentamiento de aproximadamente 3 segundos. También se observa una diferencia significativa entre las dos respuestas a la perturbación de tipo función impulso. Mientras que el control con redes neuronales tiene un tiempo de asentamiento de aproximadamente 2 segundos y pico de sobrepaso de 1.3° , el control sin redes neuronales tiene un tiempo de asentamiento de aproximadamente 5 segundos y un sobrepaso de 8.2° . Estos resultados demuestran que con el control basado en redes neuronales el sistema de control es más robusto a perturbaciones y presenta una mejor respuesta transitoria. En la figura 4.8 se muestra la evolución en el tiempo de los parámetros de la red neuronal. Como

4. RESULTADOS

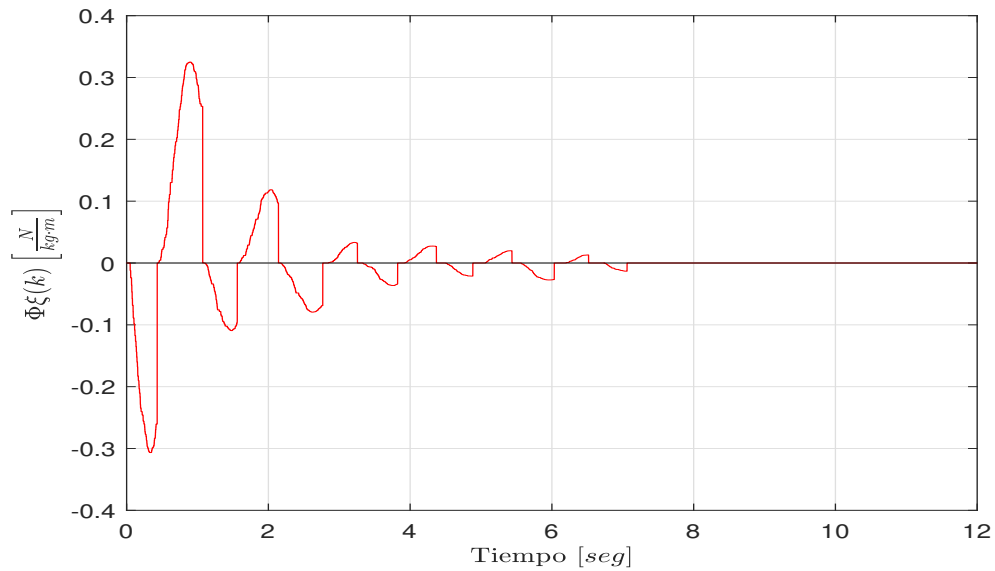


Figura 4.6: Salida del controlador *reset*.

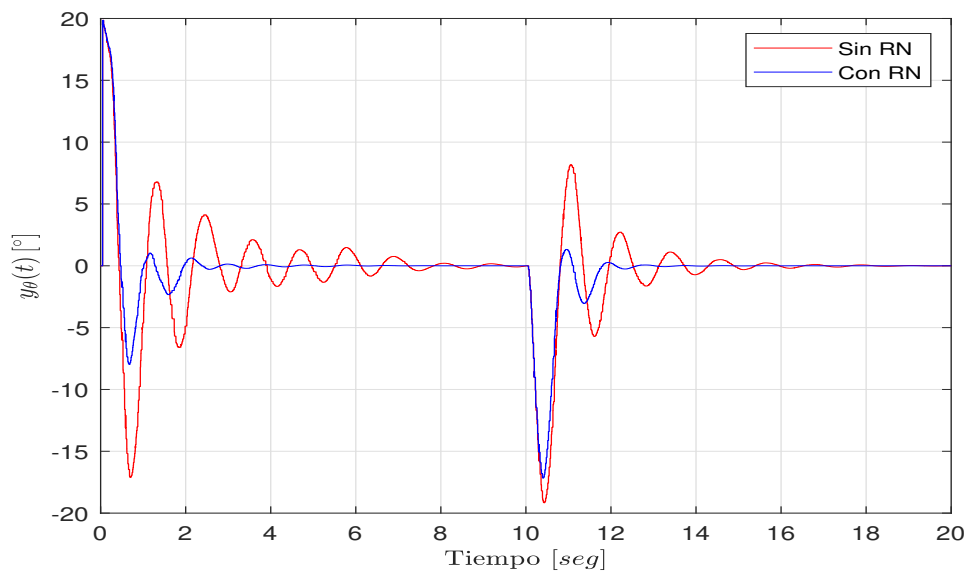


Figura 4.7: Respuesta del sistema con y sin el control basado en redes neuronales, con retardo y periodo de muestreo aleatorios, sin ruido de medición, con el controlador nominal y con el controlador *reset*.

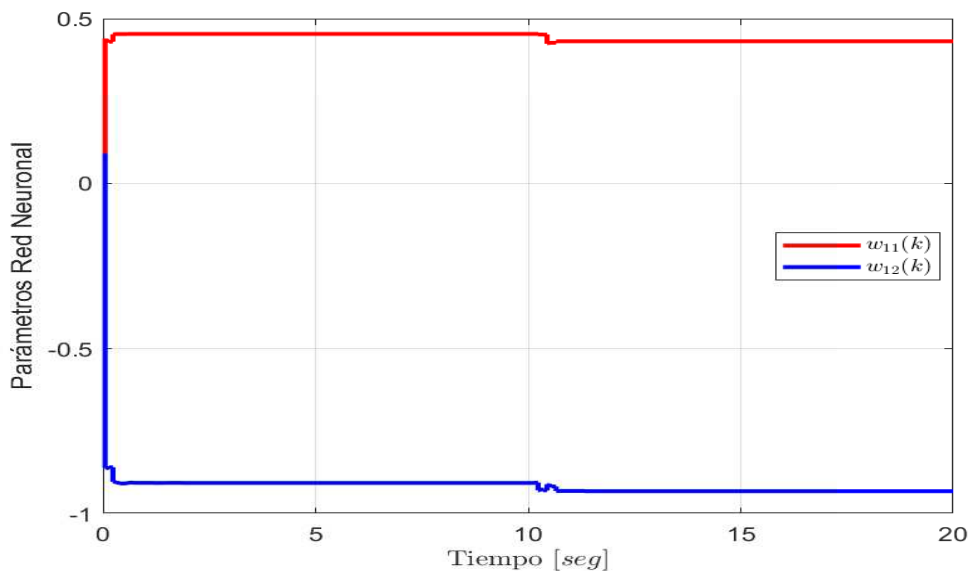


Figura 4.8: Evolución de los parámetros de la red neuronal cuando el sistema tiene retardo y periodo de muestreo aleatorios, sin ruido de medición y con el control nominal, el *reset* y el basado en redes neuronales.

se esperaba, los parámetros se adaptan para reducir el error y mejorar el desempeño del sistema de control, especialmente durante los dos periodos de respuesta transitoria, recordando que $w_{11}(0) = 0.44$ y $w_{12}(0) = 0.09$.

En la figura 4.9 se muestran la respuesta del sistema ($y_{\theta}(t)$) y la salida de la red neuronal ($\Gamma\gamma_1(k)$). Como se puede apreciar, cuando $y_{\theta}(t)$ está decreciendo $\Gamma\gamma_1(k)$ es negativo y cuando $y_{\theta}(t)$ está creciendo $\Gamma\gamma_1(k)$ es positivo, además, cuando $y_{\theta}(t)$ cambia más rápido $\Gamma\gamma_1(k)$ es más grande y justo en los instantes en que $y_{\theta}(t)$ tiene velocidad cero $\Gamma\gamma_1(k)$ es también cero. Por estas razones se concluye que la salida de control de la red neuronal es en términos generales proporcional a la velocidad de la salida del sistema. Por lo tanto, se puede interpretar que la red neuronal funciona como un control diferencial. Un control diferencial a grandes rasgos sirve para mejorar el desempeño pues puede reducir el sobrepaso, aumentar el amortiguamiento (menos oscilaciones) y reducir el tiempo de asentamiento. Por estas razones al agregar la red neuronal el desempeño del sistema de control mejoró significativamente.

En la figura 4.10 se observa la respuesta del sistema con el control basado en redes neuronales ($\Gamma\gamma(k)$) pero sin el control *reset* ($\Phi\xi(k)$). Como se puede apreciar el control *reset* ayuda a mejorar el desempeño, especialmente a reducir el sobrepaso, de la misma forma que lo hizo en los casos de la figura 4.4.

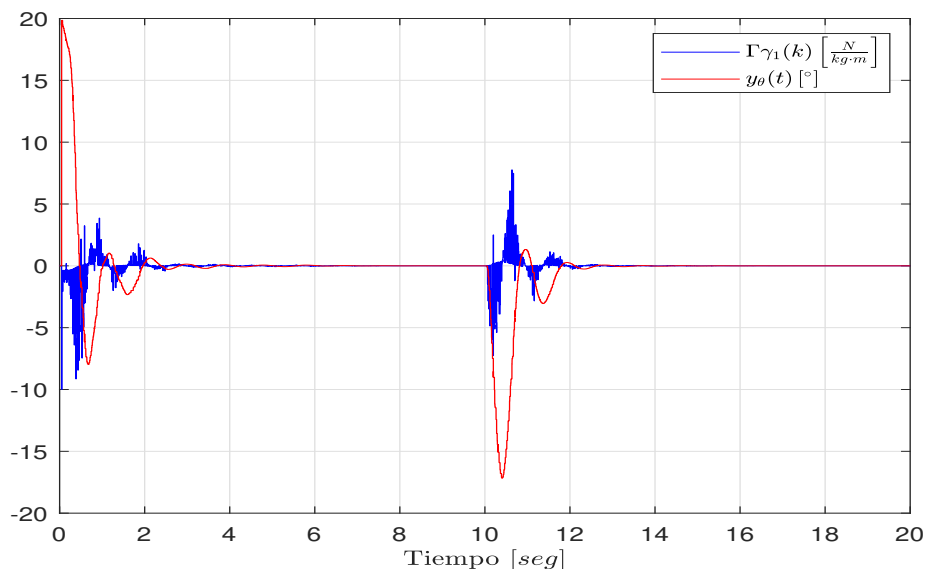


Figura 4.9: Salida del controlador basado en redes neuronales y respuesta del sistema con retardo y periodo de muestreo aleatorios, sin ruido de medición y con el control nominal, el *reset* y el basado en redes neuronales.

Ahora se considera el ruido de medición que se indica en la tabla 4.1. En la figura 4.11 se comparan las respuestas del sistema con el control neuronal con y sin la predicción hecha con el HMM. Es claro que el control neuronal es sensible al ruido de medición (al igual que un control diferencial) pues hay grandes desviaciones debido al ruido de medición, especialmente cuando la salida está cerca de cero. Sin embargo, la salida del sistema con el control neuronal predictivo es aproximadamente cero en estado estacionario. Además, como se aprecia en la figura 4.12 el control neuronal predictivo, gracias a la adaptabilidad de la red neuronal, sigue presentando un mejor desempeño respecto al control nominal. En la figura 4.13 se muestra la evolución en el tiempo de los parámetros de la red neuronal. Como se esperaba, nuevamente los parámetros se adaptan para reducir el error y mejorar el desempeño de especialmente durante los periodos de respuesta transitoria, además, la evolución de estos es muy similar a la presentada en el caso sin ruido (figura 4.8).

Estos resultados demuestran el cumplimiento del objetivo del controlador neuronal predictivo propuesto en la presente tesis, el cual era obtener una respuesta del sistema que tienda a cero a partir de condiciones iniciales distintas de cero y que mejore el desempeño respecto a otros controladores estabilizantes, en este caso, se mejora el desempeño respecto al controlador nominal que únicamente considera un retardo de comunicación

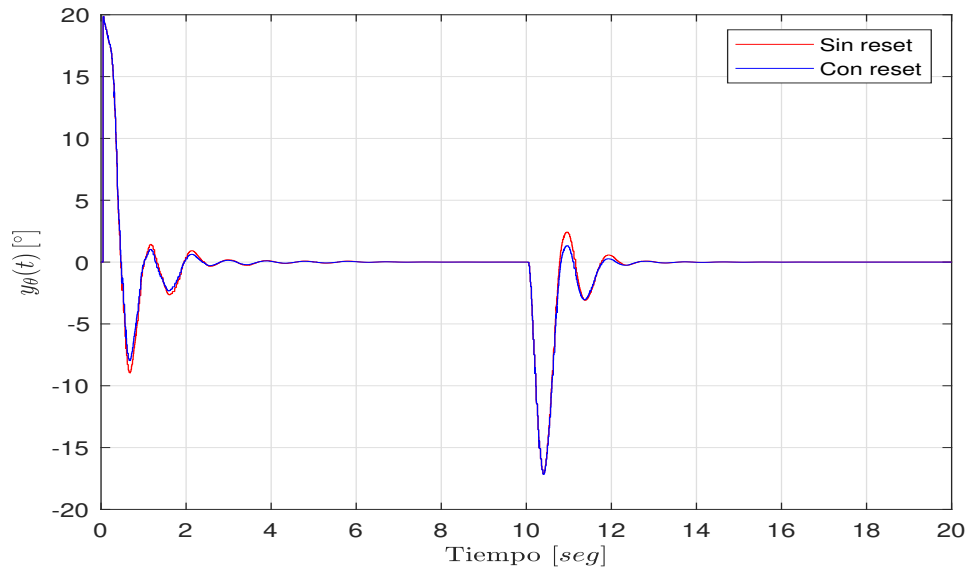


Figura 4.10: Respuesta del sistema con y sin el control *reset*, con retardo y periodo de muestreo aleatorios, sin ruido de medición y con el control nominal y el basado en redes neuronales.

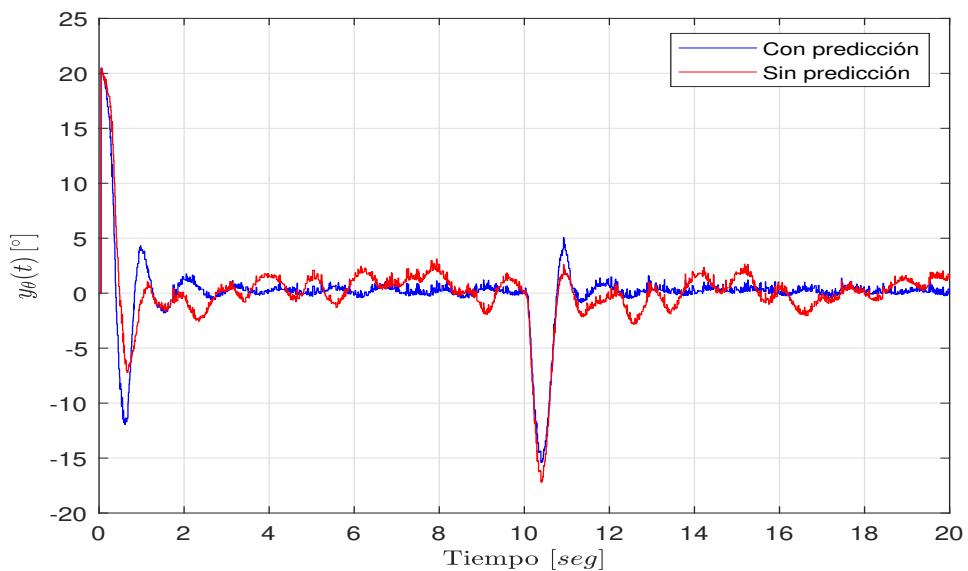


Figura 4.11: Respuesta del sistema (*pitch*) con y sin predicción, con retardo y periodo de muestreo aleatorios, con ruido de medición, con el control nominal, el *reset* y el basado en redes neuronales.

4. RESULTADOS

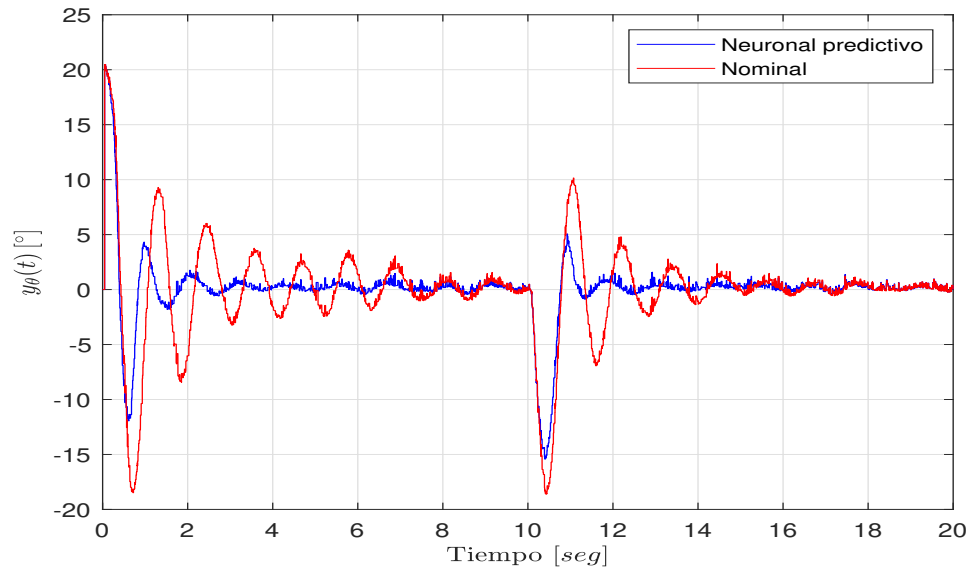


Figura 4.12: Respuesta del sistema (*pitch*) con el controlador nominal y con el controlador neuronal predictivo (nominal+reset+neuronal+predicción); con retardo y periodo de muestreo aleatorios y con ruido de medición.

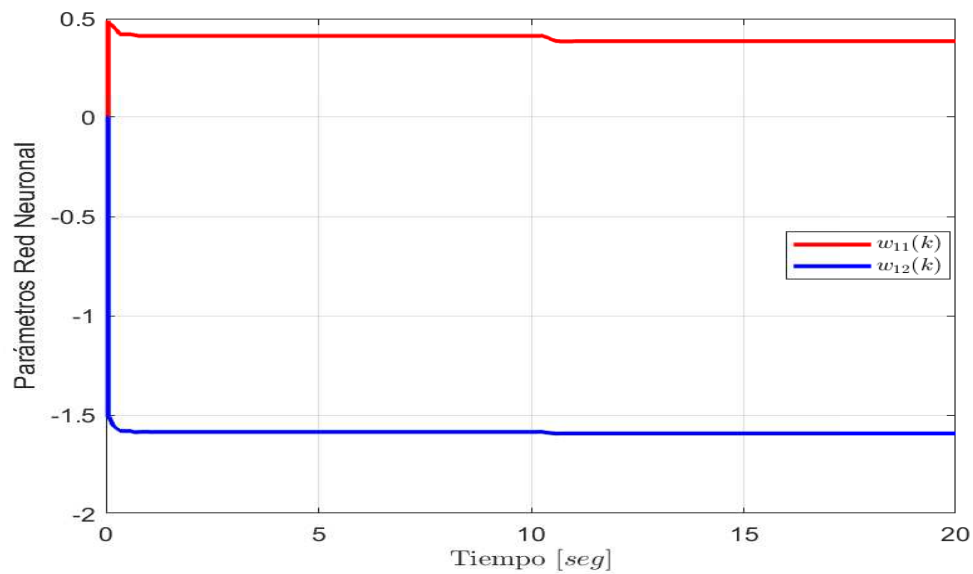


Figura 4.13: Evolución de los parámetros de la red neuronal cuando el sistema tiene retardo y periodo de muestreo aleatorios, ruido de medición y el controlador neuronal predictivo.

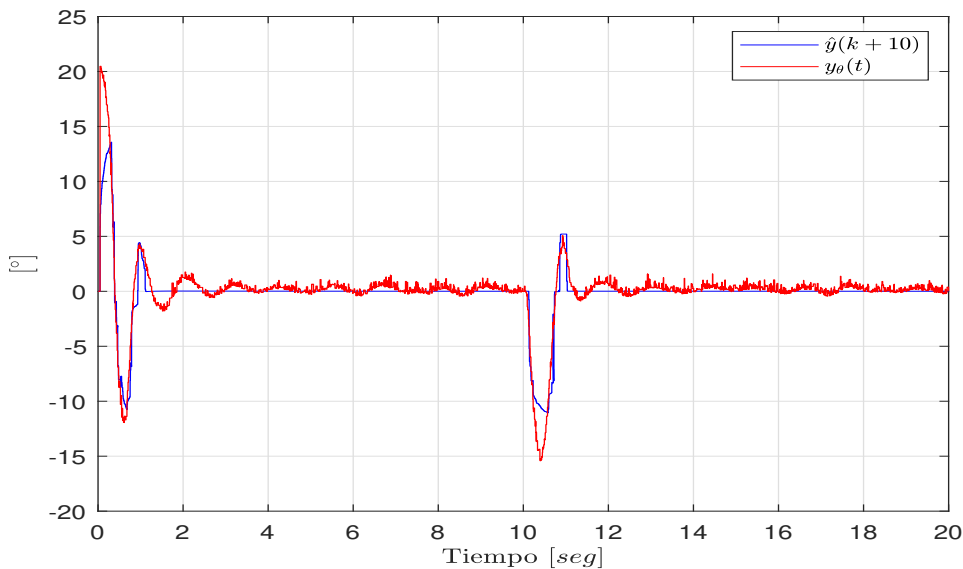


Figura 4.14: Predicción y respuesta del sistema (*pitch*) con el controlador neuronal predictivo, con retardo y periodo de muestreo aleatorios y con ruido de medición

constante. Las mejoras en el desempeño son significativas reducciones en el tiempo de asentamiento, en el sobrepaso y en las oscilaciones del sistema. En la figura 4.14 se aprecia la salida del HMM, es decir, la predicción de la salida del sistema ($\hat{y}(k+10)$) y se compara con la salida verdadera del sistema ($y_\theta(t)$). Como se puede observar se hace una predicción que permite el aprendizaje de una red neuronal y sobretodo resuelve el problema del ruido de medición.

La figura 4.15 muestra las probabilidades $P(Y(k+10)|O_{10})$ estimadas con el HMM y a partir de las cuales se obtiene la predicción $\hat{y}(k+10)$. Como se puede observar, la probabilidad $P(Y(k+10) = y_4|O_{10})$, la cual es la probabilidad de que el valor absoluto de la salida sea menor a 3° , empieza cerca de cero y crece hasta alcanzar un valor de más de 0.9 aproximadamente a los 5 segundos. En otro caso, la probabilidad $P(Y(k+10) = y_7|O_{10})$ (la probabilidad de que el valor de la salida sea mayor a 10°) empieza en 0.3 y crece rápidamente hasta llegar a 0.8, luego decrece continuamente hasta llegar a casi cero en 0.5 segundos. La probabilidad $P(Y(k+10) = y_1|O_{10})$ (la probabilidad de que el valor de la salida sea menor a -10°), por su parte, empieza en casi cero y crece continuamente hasta llegar a 0.8 a los 0.5 segundos, luego decrece nuevamente hasta llegar a cero en el segundo uno. Esto, como se puede observar en la figura 4.14, trae como resultado predecir que la salida del sistema sea cero a los dos segundos. Como también se puede observar en la figura 4.15, después de que se aplicó la perturbación a los 10 segundos, la probabilidad $P(Y(k+10) = y_4|O_{10})$ decrece abruptamente hasta llegar a casi cero y luego vuelve a crecer hacia uno. Por el contrario, las otras probabilidades,

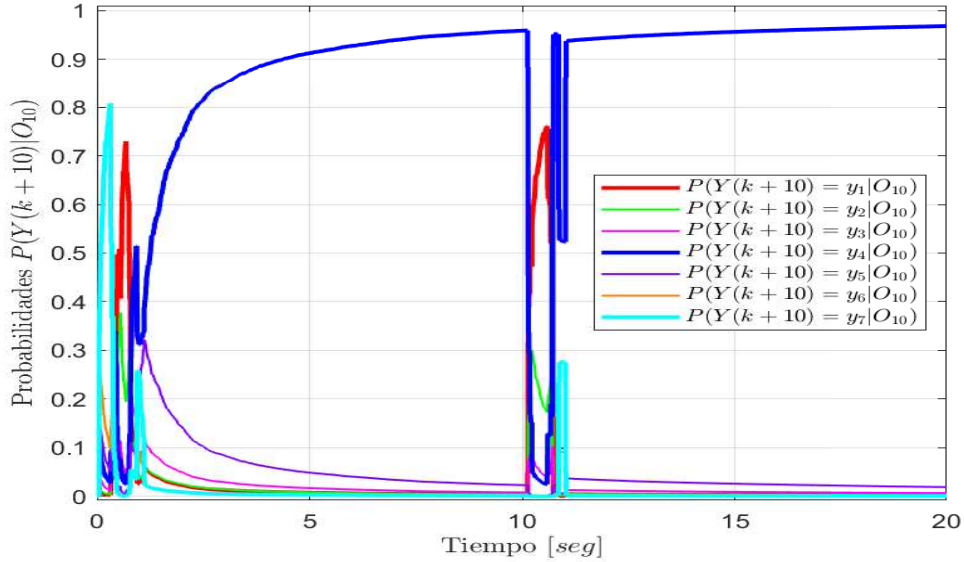


Figura 4.15: Estimaciones de las probabilidades $P(Y(k+10)|O_{10})$.

especialmente $P(Y(k+10) = y_1|O_{10})$, crecen rápidamente y luego decrecen conforme el sistema se acerca a estado estacionario. Esto demuestra que el sistema de predicción es capaz de capturar la naturaleza estocástica del sistema, incluyendo al retardo de comunicación variable en el tiempo, desde el comienzo hasta el final del experimento.

En la figura 4.16 se muestra la evolución en el tiempo de las probabilidades marginales $P(X(k))$. Estas probabilidades se calculan de manera recursiva de acuerdo a la primera ecuación de 2.54. Como se puede apreciar $P(X(k))$ guarda estrecha relación con $P(Y(k+10)|O_{10})$ (figura 4.15), especialmente en el comportamiento de $P(Y(k+10) = y_4|O_{10})$ y $P(X(k) = x_4)$, pues, por ejemplo, ambas probabilidades tienden a 1 y en el momento de la perturbación su valor se reduce. El hecho de que $P(X(k))$ y $P(Y(k+10)|O_{10})$ tengan el mismo comportamiento concuerda con el sentido común y con las observaciones del sistema, sin embargo, también llama mucho la atención, primero porque $P(Y(k+10)|O_{10})$ y $P(X(k))$ se calculan de manera independiente, pero sobre todo debido a que $P(X(k))$ solo depende de su valor inicial $P(X(0))$ y de $A(k)$, por lo tanto, si $A(k)$ estuviera fijo desde el inicio, la evolución de $P(X(k))$ no tendría que ser de acuerdo con las observaciones del sistema. Con lo cual, el hecho de que $P(X(k))$ se comporta de acuerdo con las observaciones del sistema se debe a que $A(k)$ cambia de acuerdo al algoritmo de aprendizaje en línea planteado en la sección 2.4. Por lo tanto, el hecho de que $P(X(k))$ cambie de acuerdo a las observaciones, es un buen indicio de que $A(k)$ también cambia de manera correcta de acuerdo a las observaciones. En otras palabras, la matriz $A(k)$ se actualiza de acuerdo a las observaciones y $P(X(k))$ se calcula de manera recursiva multiplicando $A(k)$ y $P(X(k-1))$, en conse-

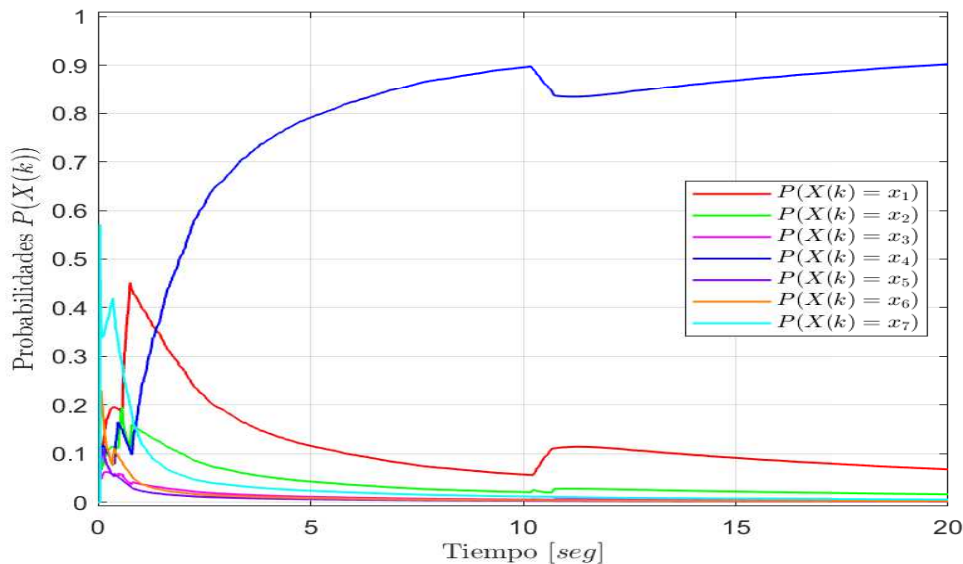


Figura 4.16: Estimaciones de las probabilidades $P(X(k))$.

cuencia se puede concluir que $P(X(k))$ también se calcula con base en las observaciones.

Utilizando $P(X(k))$ y las ecuaciones 2.56 y 2.57 es posible calcular las probabilidades $P(Y(k+10))$ cuya evolución en el tiempo se muestra en la figura 4.17. Como se esperaba estas probabilidades tienen estrecha relación con $P(X(k))$ (figura 4.16), lo cual nuevamente es un buen indicio de que el algoritmo de aprendizaje en línea para $A(k)$ y $B(k)$ funciona correctamente, además, como se explicó anteriormente, esto quiere decir que $P(Y(k+10))$ también se calcula con base en las observaciones del sistema. Con sustento en este razonamiento en Cho and Fadali (2011) y Cho and Lee (2008) se calcula la predicción de la salida del sistema a partir de $P(Y(k+T_\eta))$ con base en observaciones y utilizando la salida ponderada (ecuación 2.77), sin embargo, note la gran diferencia que existe entre $P(Y(k+10)|O_{10})$ (figura 4.15) y $P(Y(k+10))$ (figura 4.17).

Aplicando la salida ponderada a las probabilidades $P(Y(k+10))$ se obtiene la predicción de la salida mostrada en la figura 4.18. Como se puede observar la predicción con base en $P(Y(k+10))$ es muy distinta en comparación a la utilizada en el presente trabajo (figura 4.14), siendo esta última la que presenta mayor adaptabilidad a la salida actual del sistema. Con esto se demuestra que el enfoque utilizado para hacer la predicción en el presente trabajo, es decir, hacer la predicción con base en observaciones utilizando el algoritmo de avance, es mejor que el enfoque utilizado en Cho and Fadali (2011) y Cho and Lee (2008), en donde las observaciones se utilizan únicamente para aprender los parámetros del modelo probabilístico. En otras palabras, en Cho and Fadali (2011) y Cho and Lee (2008), a diferencia de la presente tesis, no se

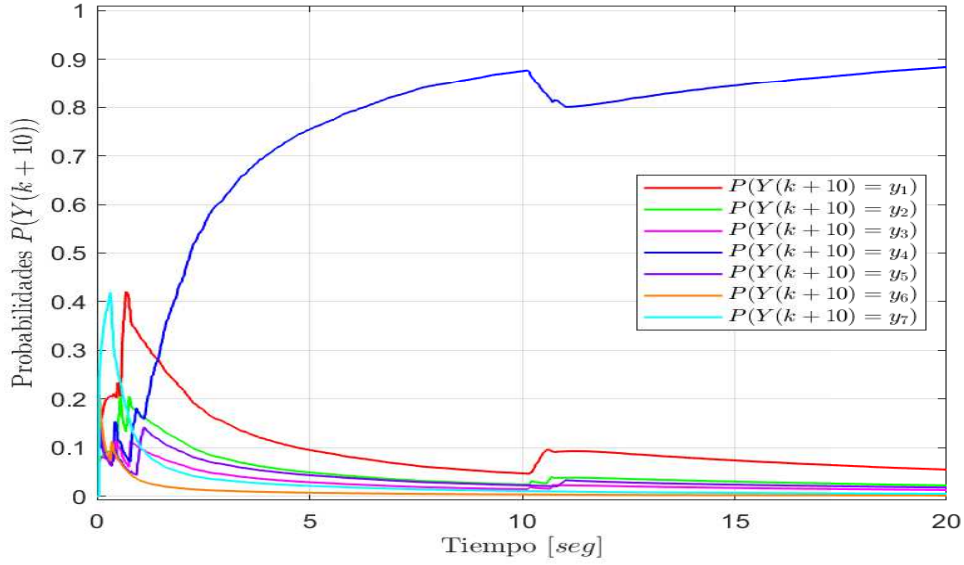


Figura 4.17: Estimaciones de las probabilidades $P(Y(k + 10))$.

calcula $P(Y(k + T_\eta)|O_{N_w})$, es decir, no se utiliza explícitamente las observaciones del sistema para hacer la predicción, únicamente para el aprendizaje de los parámetros del modelo probabilístico, lo cual como se observa en 4.18 puede servir como predicción, sin embargo, como se puede apreciar en 4.14, utilizar explícitamente las observaciones es mejor aproximación para hacer la predicción de la salida del sistema.

En la figura 4.19 se observa los valores que toma el estado $X^*(k)$ a largo del tiempo de simulación. $X^*(k)$ no puede tomar el valor $X^*(k) = 0$, sin embargo, hay que recordar que todas las figuras inician en cero debido a que al inicio de la simulación no se corre el controlador sino hasta que le llega el primer dato a este pues está presente el retardo de observación. $X^*(k)$ es un valor óptimo del valor verdadero de la variable aleatoria oculta $X(k)$ y es seleccionado de acuerdo a 2.64, es decir, $X^*(k)$ es el estado con la verosimilitud 2.63 máxima. $X^*(k)$ se calcula cada que llega un nuevo dato al controlador para actualizar las entradas de las matrices $A(k)$ y $B(k)$ con base en datos. Como se observa en la figura 4.19 $X^*(k)$ cambia de acuerdo con las observaciones del sistema ($y_\theta(t)$ de la figura 4.14), por lo tanto, este es un buen indicio de que el aprendizaje de las matrices $A(k)$ y $B(k)$ funciona correctamente. Lo cual es muy bueno pues si $A(k)$ y $B(k)$ representan cada vez mejor al sistema dinámico entonces la predicción será cada vez mejor.

En las figuras 4.20 y 4.21 se muestran la evolución en el tiempo del cuarto renglón de la matrices $A(k)$ y $B(k)$, respectivamente. Como se esperaba, los parámetros del modelo cambian desde su condición inicial a valores de acuerdo a los datos del sistema, pues las probabilidades $P(X(k) = x_4|X(k - 1) = x_4)$ y $P(Y(k) = y_4|X(k) = x_4)$ aumentan

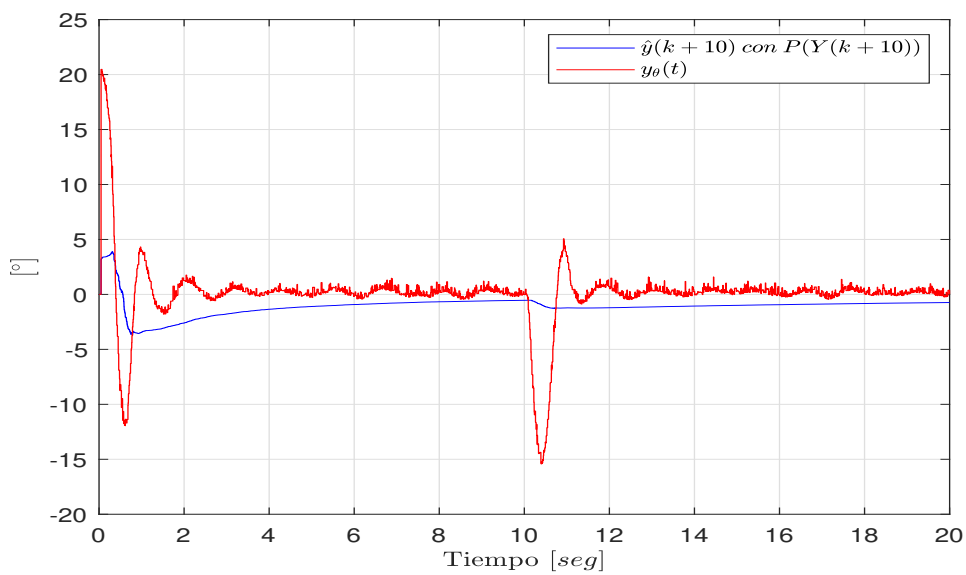


Figura 4.18: Predicción con base en $P(Y(k+10))$.

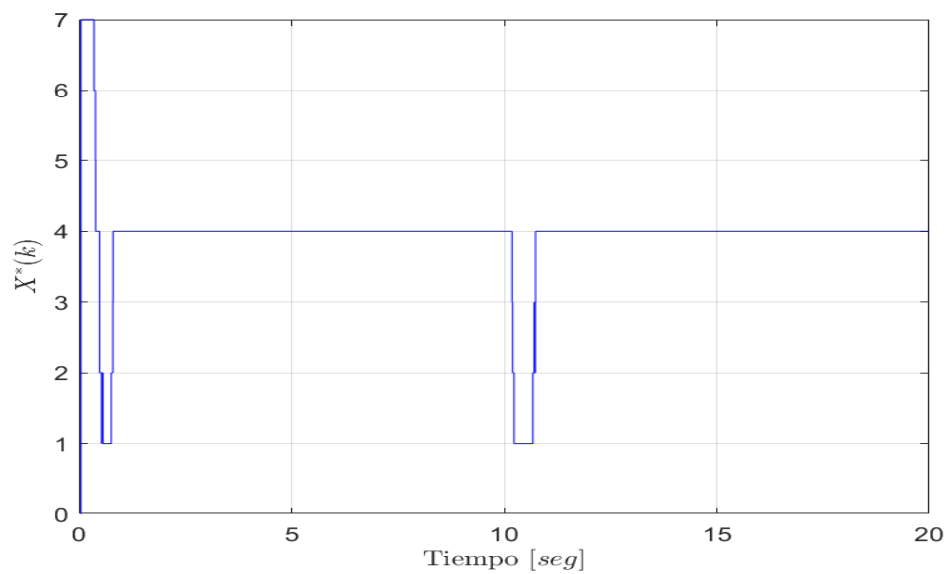


Figura 4.19: Valores que toma $X^*(k)$.

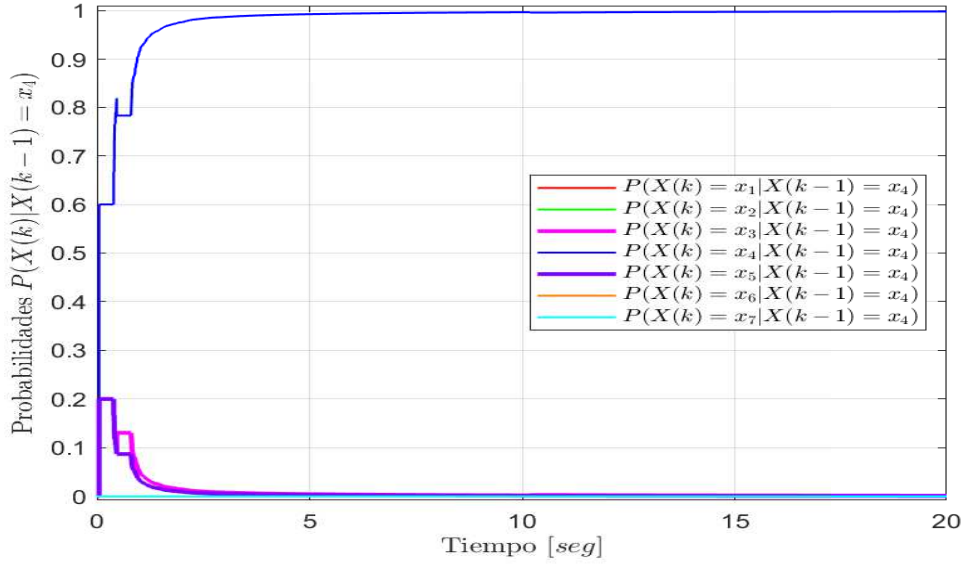


Figura 4.20: Estimaciones de las probabilidades $P(X(k)|X(k-1) = x_4)$.

continuamente y en el momento de la perturbación decrecen ligeramente, caso contrario para las demás probabilidades. También hay que resaltar que las probabilidades varían alrededor de las condiciones iniciales que se escogieron para las matrices de verosimilitudes $M(k)$ y $N(k)$ (3.34), es por eso que se tiene una buena predicción de la salida del sistema desde el inicio de la simulación. Las gráficas 4.20 y 4.21 confirman que el aprendizaje de los parámetros del HMM funciona correctamente.

En la siguiente sección se muestran los resultados de las simulaciones para el ángulo *yaw*. Los resultados de las simulaciones para el ángulo *roll* no se muestran debido a que son los mismos que los del *pitch*. Esto se debe a que, como se puede apreciar en 3.4 y en la tabla 3.1, la dinámica del ángulo *roll* (sistema 3.5) tiene la misma estructura y mismos valores numéricos que el ángulo *pitch*; además, como se mencionó al final del capítulo 3, el controlador neuronal predictivo se implementó de la misma manera para ambos ángulos.

4.2. Yaw

En esta sección se muestran los resultados de las simulaciones de la implementación del controlador neuronal predictivo al sistema dinámico no lineal en tiempo continuo 3.6 de acuerdo a lo desarrollado en el capítulo 3. Las condiciones iniciales del sistema 3.6 se escogieron nuevamente como $x_1(0) = 20^\circ$ y $x_2(0) = -12^\circ$. Como se explicó anteriormente, en las figuras de esta sección se muestra la salida del sistema 3.6 al llegar

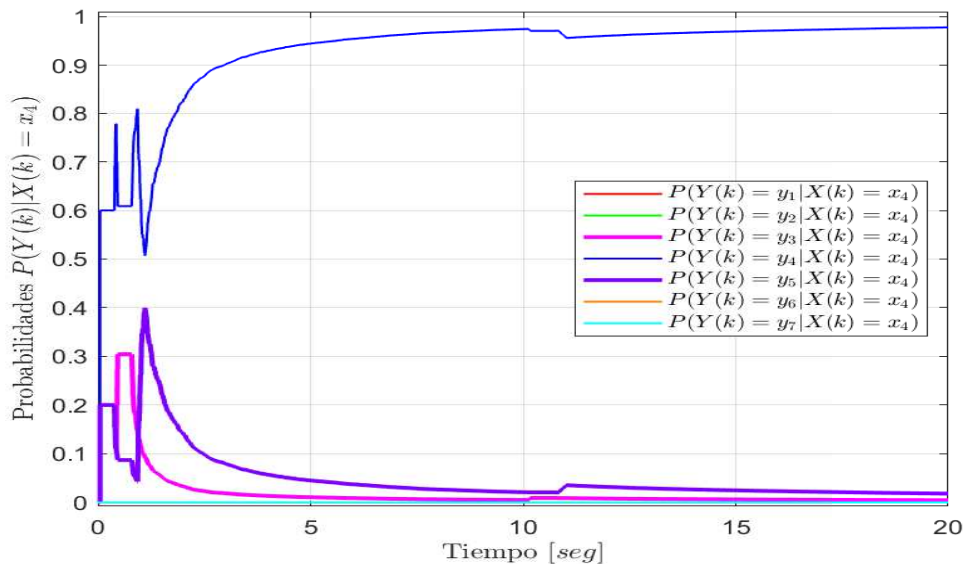


Figura 4.21: Estimaciones de las probabilidades $P(Y(k)|X(k) = x_4)$.

al controlador; llamaremos a esta señal $y_\psi(t)$ en el resto de la tesis. En las primeras simulaciones no se consideró al ruido de medición para dar un mayor énfasis a contrarrestar las perturbaciones causadas por el retardo de comunicación estocástico. En la figura 4.22 se muestra la salida del sistema para los casos de las figuras 4.4, 4.7 y 4.10, es decir, con el controlador nominal 3.23, con el controlador nominal y el controlador *reset* (control 4.2), con el controlador nominal y el controlador neuronal ($\Gamma\gamma_1(k)$ de 3.25), finalmente, con el controlador nominal y el control correctivo (3.25).

Como se puede apreciar en 4.22 se tiene resultados muy similares a los que se tuvieron con el *pitch*, pues al añadir el control *reset* disminuyen el sobrepaso, las oscilaciones y el tiempo de asentamiento, sin embargo, al añadir el control neuronal es cuando el desempeño mejora más significativamente. Ahora se considera el ruido de medición que se indica en la tabla 4.1. En la figura 4.23 se muestran las respuestas del sistema para los casos de la figura 4.11, es decir, con el controlador neuronal con y sin la predicción hecha con el HMM. Nuevamente queda claro que el control neuronal es sensible al ruido de medición, sin embargo, la salida del sistema con el control neuronal predictivo es aproximadamente cero en estado estacionario. Además, como se aprecia en la figura 4.24 el controlador neuronal predictivo presenta un desempeño significativamente mejor respecto al control nominal.

Estos resultados confirman el cumplimiento del objetivo del controlador neuronal predictivo propuesto en la presente tesis. En la figura 4.25 se aprecia la salida del HMM, es decir, la predicción de la salida del sistema ($\hat{y}(k + 10)$) y se compara con

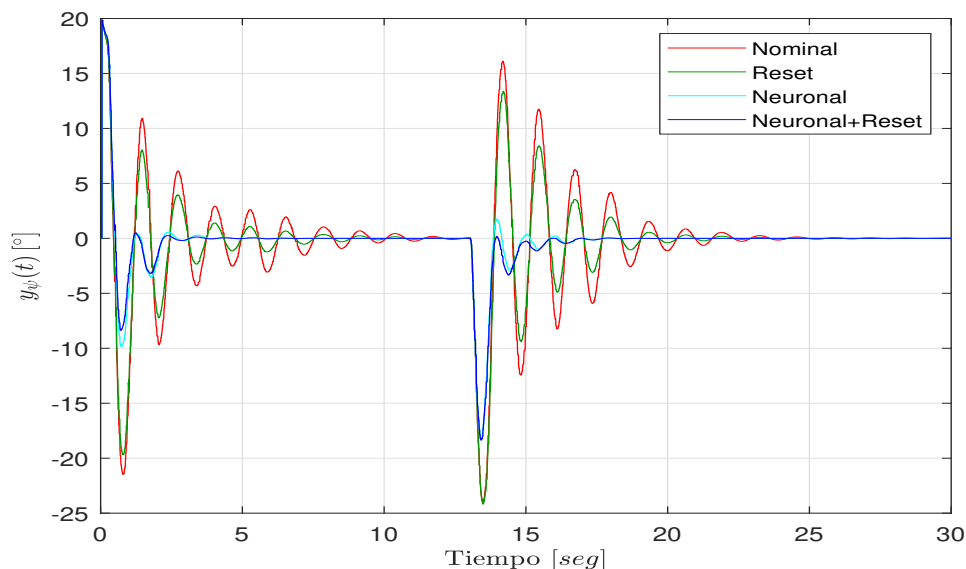


Figura 4.22: Respuesta del sistema con el controlador nominal, con el controlador nominal+reset, con el controlador nominal+neuronal y con el controlador nominal+reset+neuronal; con retardo y periodo de muestreo aleatorios y sin ruido de medición.

la salida verdadera del sistema ($y_\psi(t)$). Como se puede observar, nuevamente se hace una predicción que permite el aprendizaje de la red neuronal y sobretodo resuelve el problema del ruido de medición. En la siguiente sección se presentan las conclusiones finales del presente trabajo.

4.3. Comparación del desempeño

En las figuras 4.12 y 4.24 se puede observar claramente la superioridad del controlador neuronal predictivo respecto al control nominal, el cual se diseñó utilizando la teoría de sistemas discretos lineales con retardo de tiempo constante. Asimismo, se puede verificar la superioridad del esquema de control propuesto respecto a otros controladores. Tome como ejemplo al sistema no lineal 3.4 (dinámica del *pitch*) y sustituya la siguiente ley de control (linealización exacta (LE)):

$$U_1(t) = -1 \left(\frac{mgl}{J_p} \text{sen}(x_1(t)) + u_i(t) \right) \quad (4.3)$$

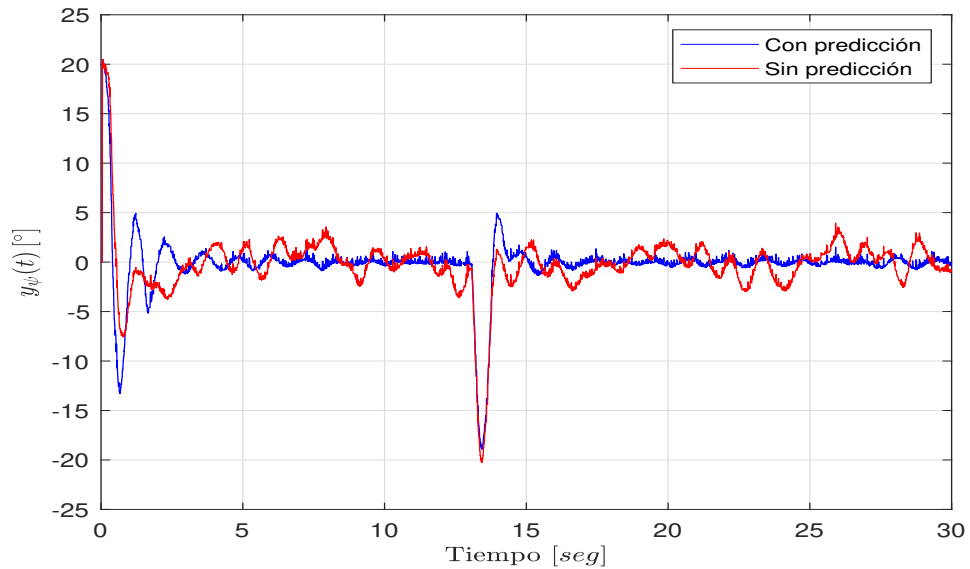


Figura 4.23: Respuesta del sistema (yaw) con y sin predicción, con retardo y periodo de muestreo aleatorios, con ruido de medición, con el control nominal, el *reset* y el basado en redes neuronales.

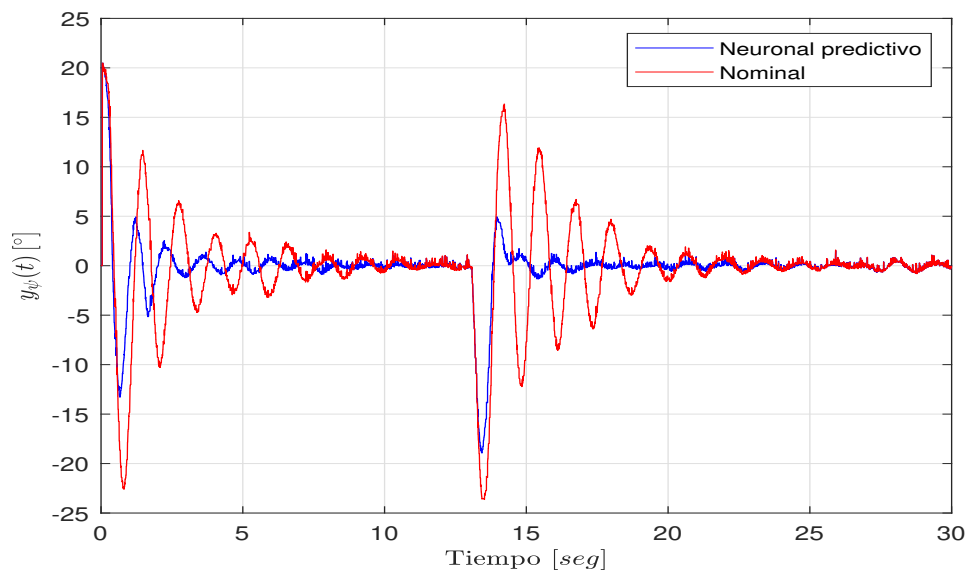


Figura 4.24: Respuesta del sistema (yaw) con el controlador nominal y con el controlador neuronal predictivo (nominal+*reset*+neuronal+predicción); con retardo y periodo de muestreo aleatorios y con ruido de medición.

4. RESULTADOS

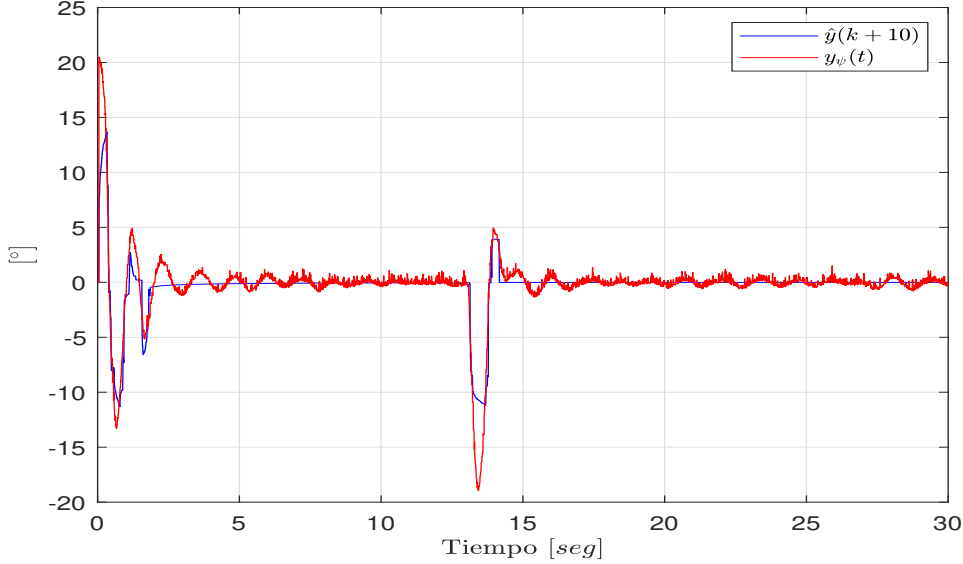


Figura 4.25: Predicción y respuesta del sistema (*yaw*) con el controlador neuronal predictivo, con retardo y periodo de muestreo aleatorios y con ruido de medición

El sistema de en lazo cerrado queda:

$$\begin{aligned} \dot{x}_\theta(t) &= \begin{bmatrix} x_2(t) \\ -\frac{b}{J_p}x_2(t) + u_l(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{b}{J_p} \end{bmatrix} x_\theta(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_l(t) = Ax_\theta(t) + Bu_l(t) \\ z(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_\theta(t) \end{aligned} \quad (4.4)$$

Si $u_l(t) = -Kx_\theta(t) = -[2 \ 0.5507]x(t)$ (considerando $z(t) = x_\theta(t)$), entonces 4.4 queda $\dot{x}_\theta(t) = (A - BK)x_\theta(t)$ con $A - BK$ Hurwitz, por lo tanto, el origen del sistema no lineal 3.4 es asintóticamente estable. Considerando que no hay retardo de comunicación, muestreador ni ruido de medición y las condiciones iniciales $x_1(0) = 20^\circ$ y $x_2(0) = -12^\circ$ se tiene la respuesta de la figura 4.26. Como se puede observar, se tiene una respuesta asintóticamente estable y con buen desempeño. Sin embargo, considerando al retardo de comunicación, al periodo de muestreo y al ruido de medición de acuerdo a los valores de la tabla 4.1 se obtuvo la respuesta de la figura 4.27. En esta figura también se muestra la salida con el controlador neuronal predictivo. Como se puede apreciar, con el controlador por linealización exacta se deja de cumplir el objetivo de estabilización pues existen desviaciones significativas de la salida con respecto de cero.

Ahora considere al sistema lineal 3.6 (dinámica del *yaw*) y sustituya la siguiente ley de

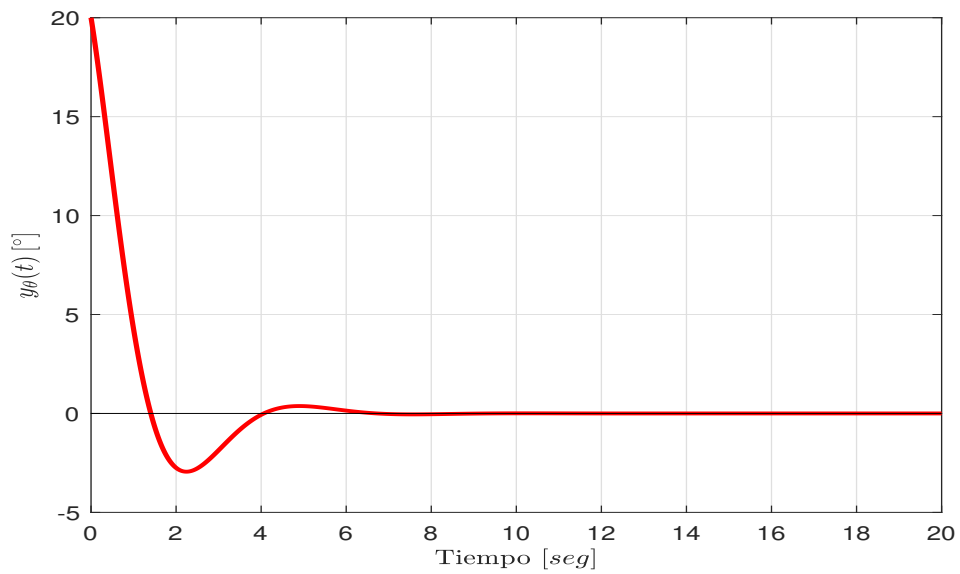


Figura 4.26: Respuesta del sistema (*pitch*) con el controlador por linealización exacta, sin muestreador (tiempo continuo), sin retardo de comunicación y sin ruido de medición.

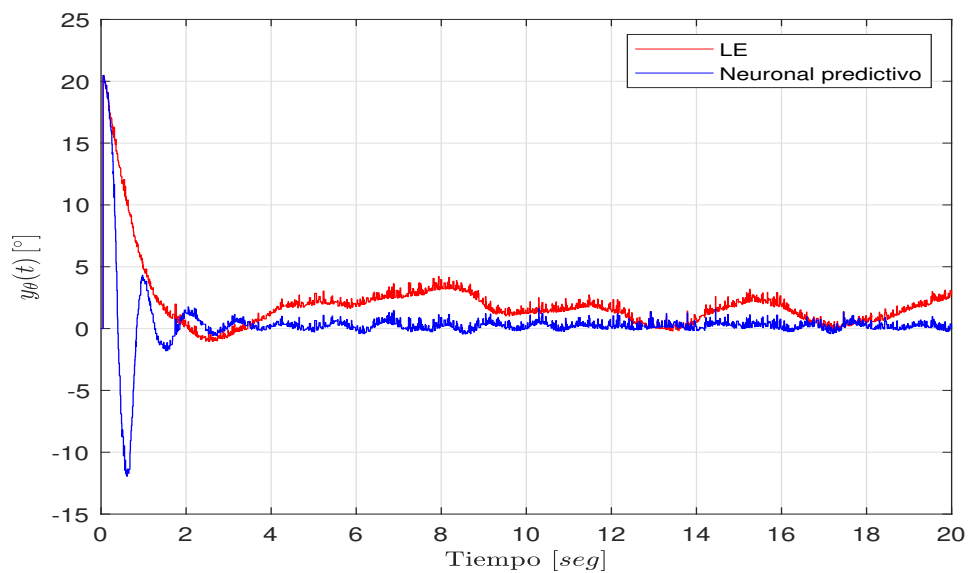


Figura 4.27: Respuesta del sistema (*pitch*) con el controlador por linealización exacta y con el neuronal predictivo; con retardo y periodo de muestreo aleatorios y con ruido de medición.

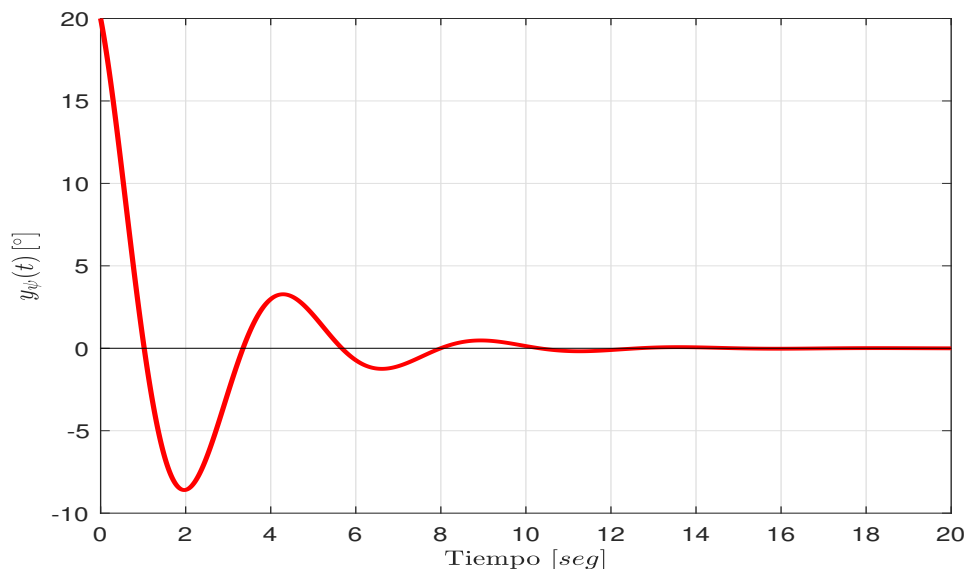


Figura 4.28: Respuesta del sistema (*yaw*) con el controlador por retroalimentación lineal de estados, sin muestreador (tiempo continuo), sin retardo de comunicación y sin ruido de medición.

control de retroalimentación lineal de estados (considerando $z(t) = x_\psi(t)$):

$$U_3(t) = -Kx_\psi(t) = -[-2 \quad -0.5507] x_\psi(t) \quad (4.5)$$

Con la retroalimentación lineal de estados (RE) el sistema en lazo cerrado queda $\dot{x}_\psi(t) = (A-BK)x_\psi(t)$ con $A-BK$ Hurwitz, por lo tanto, el sistema lineal 3.6 es asintóticamente estable. Considerando que no hay retardo de comunicación, muestreador ni ruido de medición y las condiciones iniciales $x_1(0) = 20^\circ$ y $x_2(0) = -12^\circ$ se tiene la respuesta de la figura 4.28. Como se puede observar, se tiene una respuesta asintóticamente estable y con buen desempeño. Sin embargo, considerando al retardo de comunicación, al periodo de muestreo y al ruido de medición se obtuvo la respuesta de la figura 4.29. En esta figura también se muestra la salida con el controlador neuronal predictivo. Como se puede apreciar, con el controlador por retroalimentación lineal de estados se deja de cumplir el objetivo de estabilización pues existe un error en estado permanente.

4.4. Análisis del retardo mínimo y máximo

Un aspecto muy importante de los NCSs son los retardos de comunicación máximo y mínimo que el sistema puede soportar sin que el lazo de control se haga inestable. Para hacer este análisis se probó el lazo de control con retardos de observación y de control

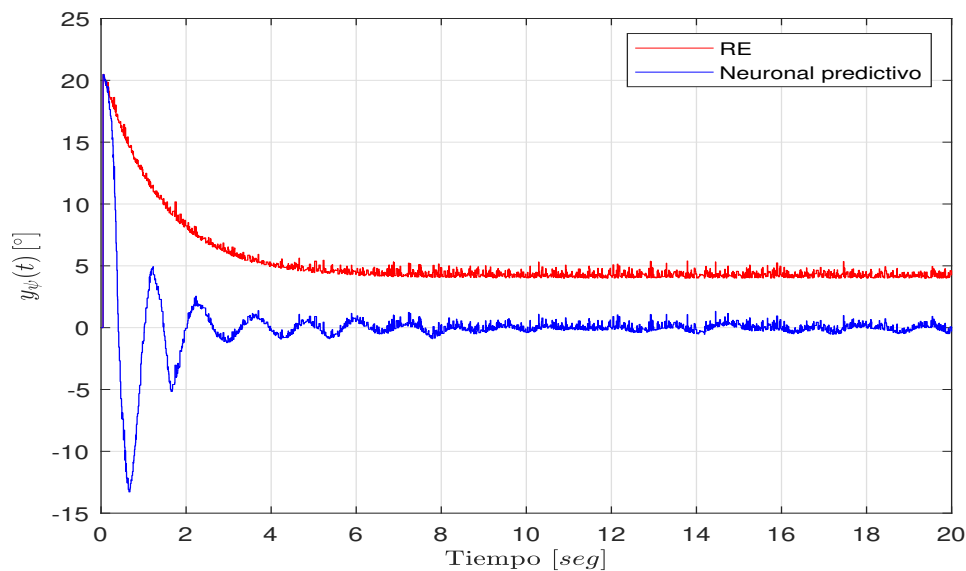


Figura 4.29: Respuesta del sistema (*yaw*) con el controlador por retroalimentación lineal de estados y con el neuronal predictivo; con retardo y periodo de muestreo aleatorios y con ruido de medición.

constantes e iguales a $0.05[\text{seg}]$ (1.8 veces los valores esperados $E(\tau_{u(t_k)})$ y $E(\tau_{y(s_k)})$ con los cuales se obtuvieron los resultados de las simulaciones en el capítulo 4). En la figura 4.30 se muestra la respuesta del ángulo *pitch* con el controlador neuronal predictivo y con el controlador nominal. En estas simulaciones se consideró al periodo de muestreo y al ruido de medición de acuerdo a los valores de la tabla 4.1, también se tomó en cuenta a la perturbación a la entrada a los 10 segundos de simulación. Como se puede observar, al aumentar el retardo de comunicación, el desempeño del controlador neuronal predictivo se ve afectado (en comparación de los resultados de la figura 4.12), sin embargo, sigue cumpliendo con el objetivo de control planteado (que las trayectorias en el espacio de estados se aproximen y permanezcan cerca de cero) y sigue presentando un buen rechazo de perturbaciones a la entrada. Por otra parte, también se observa que el desempeño con el controlador nominal se vió mucho más afectado ya que presenta muchas oscilaciones de gran amplitud.

En la figura 4.31 se muestra la respuesta del sistema considerando retardos de observación y de control constantes e iguales a $0.056[\text{seg}]$ durante 40 segundos y sin la perturbación a la entrada. Como se puede observar, al aumentar el retardo de comunicación, el degradamiento en el desempeño del controlador neuronal predictivo es más severo, además, debido a que existen grandes desviaciones de la salida respecto del cero en estado estacionario no se puede seguir considerando que se cumple con el objetivo

4. RESULTADOS

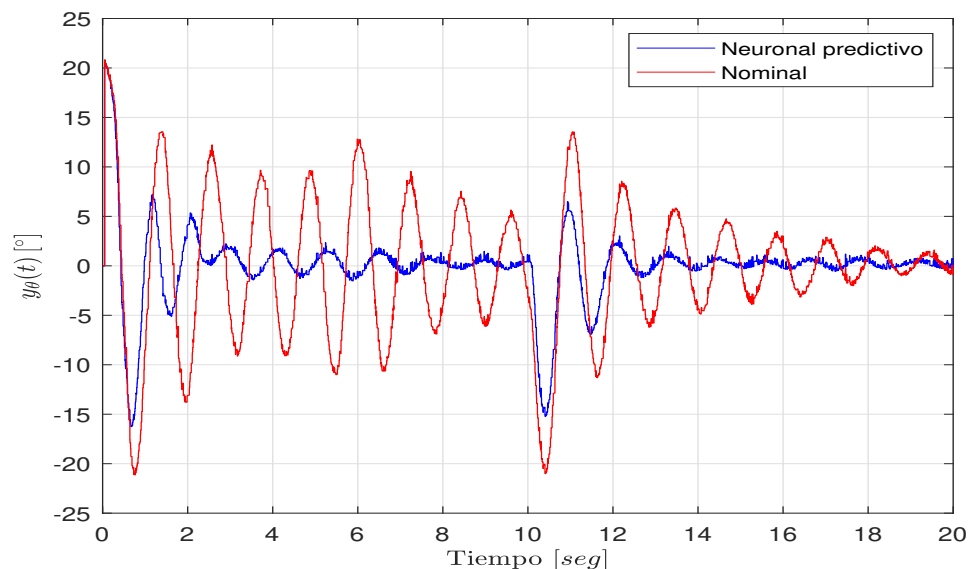


Figura 4.30: Respuesta del sistema (*pitch*) con el controlador neuronal predictivo y con el nominal; con periodo de muestreo aleatorio, con ruido de medición, con perturbación a la entrada a los 10[seg] y con retardos constantes de 0.05[seg].

de control planteado. A pesar de esto, en esta figura también se puede notar la gran diferencia con la respuesta del sistema con el controlador nominal, siendo esta última la que presenta peor desempeño y oscilaciones más grandes en tiempo estacionario. También es importante notar que, a pesar de que se incrementó el doble el valor del retardo respecto $E(\tau_{u(t_k)})$ y $E(\tau_{y(s_k)})$, la respuesta del sistema se puede seguir considerando estable. Para mostrar que aún bajo estas condiciones la predicción funciona correctamente, en la figura 4.32 se muestra la salida del sistema junto con la predicción.

Incrementando más el retardo la amplitud de las oscilaciones en estado permanente aumenta gradualmente hasta que con retardos de control y de observación constantes iguales a 0.07[seg] se tiene una respuesta que se aleja indefinidamente del origen. En la figura 4.33 se muestra la respuesta del sistema para el caso en el que no hay retardo de observación ni retardo de control. Como se puede apreciar, sin retardo de comunicación el desempeño del sistema de control es muy superior a cuando existe el retardo, lo cual reafirma la importancia de considerarlo en el diseño del controlador. Además, se observa nuevamente la superioridad del controlador neuronal predictivo sobre el nominal, algo que fue una constante a lo largo de todas las simulaciones.

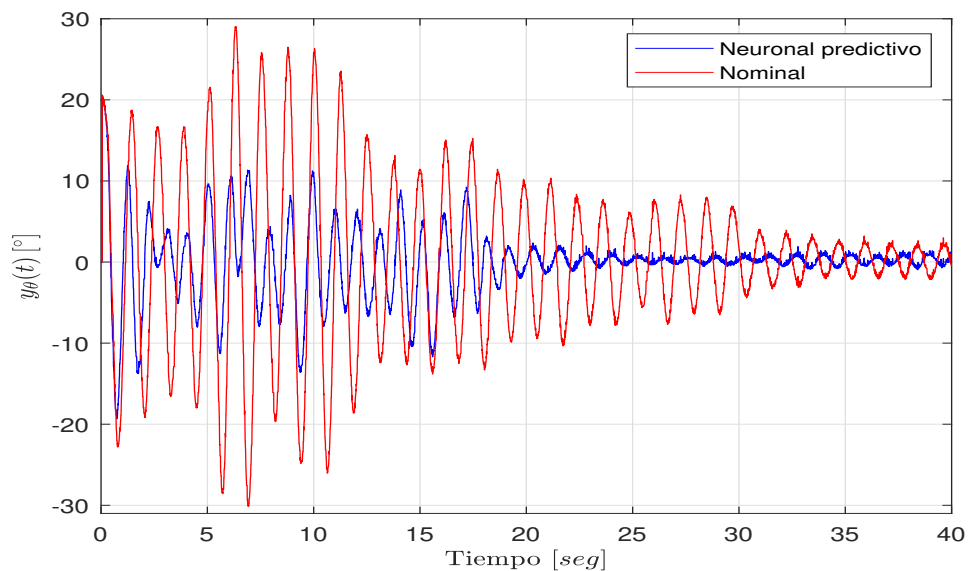


Figura 4.31: Respuesta del sistema (*pitch*) con el controlador neuronal predictivo y con el nominal; con periodo de muestreo aleatorio, con ruido de medición, sin perturbación a la entrada y con retardos constantes de 0.056[seg].

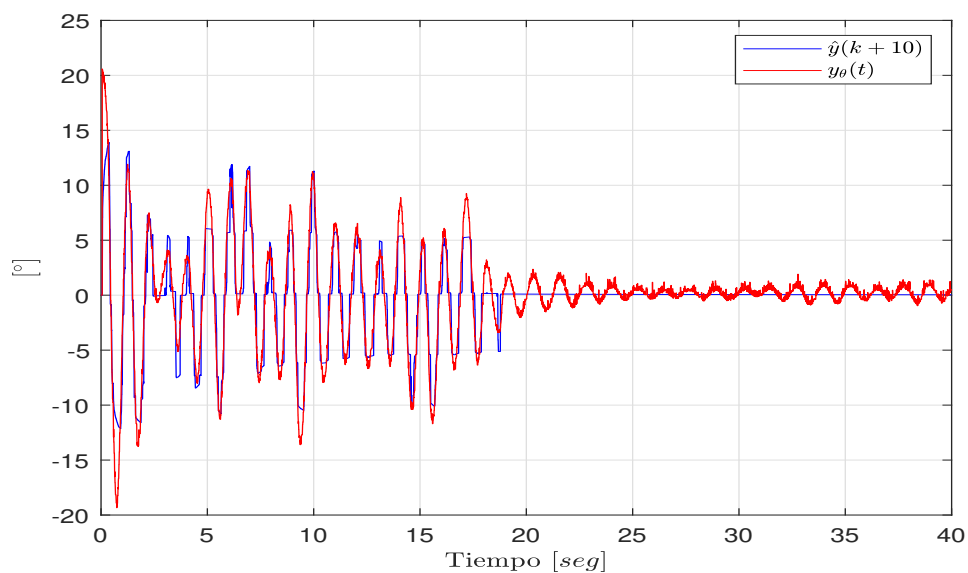


Figura 4.32: Predicción y respuesta del sistema (*pitch*) con el controlador neuronal predictivo, con periodo de muestreo aleatorio, con ruido de medición, sin perturbación a la entrada y con retardos constantes de 0.056[seg].

4. RESULTADOS

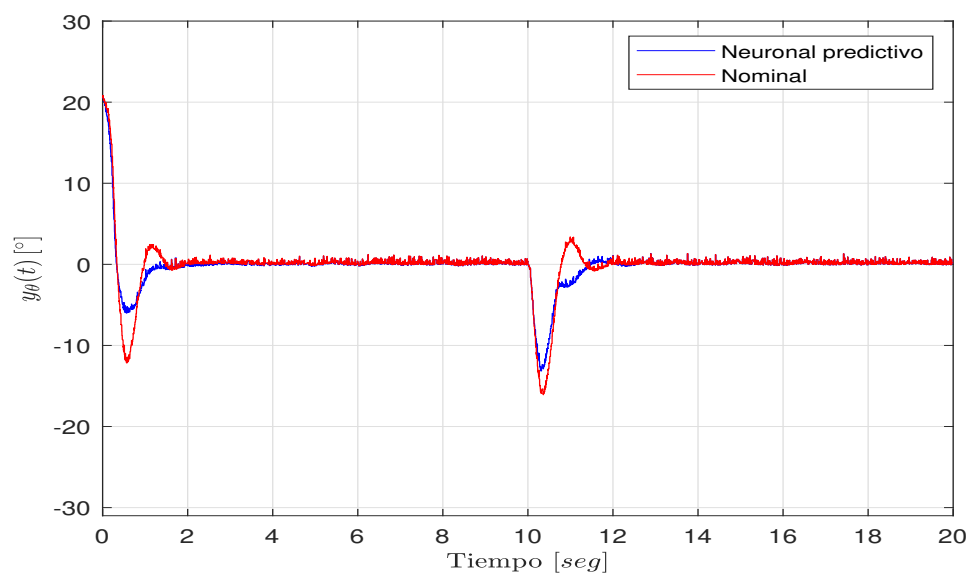


Figura 4.33: Respuesta del sistema (*pitch*) con el controlador neuronal predictivo y con el nominal; con periodo de muestreo aleatorio, con ruido de medición, sin perturbación a la entrada y sin retardo de comunicación.

Conclusiones

En el presente trabajo se propuso un controlador neuronal predictivo novedoso para NCSs en los que están presentes un retardo de comunicación y un periodo de muestreo aleatorios. El controlador se logró combinando control de sistemas lineales discretos con retardo de control constante, control basado en *reset*, control basado en redes neuronales y un modelo oculto de Markov, el cual se utiliza para predecir el error del sistema de control. Implementando el controlador a un sistema no lineal MIMO tipo cuadricóptero y por medio de simulaciones se demostró el cumplimiento del objetivo planteado al inicio de la presente tesis, pues el controlador propuesto estabiliza al sistema y presenta un mejor desempeño en comparación a otros controladores (linealización exacta, retroalimentación lineal de estados y control de sistemas lineales discretos con retardo de tiempo constante).

Por otra parte, a pesar de que la estabilidad no se demostró en un sentido riguroso y cuantitativo, como lo es en el sentido de Lyapunov o en el sentido de la estabilidad estocástica (Verriest (2002)), lo cual puede llegar a ser muy complicado de abordar debido a las perturbaciones estocásticas que presenta un NCS, con base en los resultados obtenidos se deja en claro que el controlador neuronal predictivo es útil para que las trayectorias en el espacio de estados se aproximen y permanezcan cerca de cero dada una condición inicial distinta de cero y obtener un buen desempeño, tanto en la respuesta transitoria como en la de estado estacionario, pues con este se reduce el sobrepaso y el tiempo de asentamiento. Asimismo, se demostró que el controlador mejora la robustez del sistema, pues se reducen las oscilaciones y provee un excelente rechazo a perturbaciones abruptas a la entrada. La demostración de la estabilidad en algún sentido riguroso se deja como trabajo a futuro.

Asimismo, se pudo comprobar que el controlador nominal propuesto funciona correctamente pues, a pesar de que su desempeño se ve afectado por el ruido de medición, el retardo de comunicación y el periodo de muestreo estocásticos, este siempre cumple con el objetivo de estabilización, a diferencia de los controladores por retroalimentación lineal de estados (RE) y por linealización exacta (LE). También se puede concluir la im-

5. CONCLUSIONES

portancia de tomar en cuenta al retardo de tiempo y al periodo de muestreo en el diseño de un controlador para NCSs, incluso si se trata solo del controlador nominal, pues si para el control nominal del controlador neuronal predictivo propuesto se hubiera considerado el controlador RE o el LE, se hubiera obtenido un desempeño tan deficiente que probablemente no se hubiera podido corregir ni siquiera añadiendo el control correctivo.

También se demostró la funcionalidad del controlador neuronal predictivo tanto para sistemas lineales como no lineales. Además, se concluye que el controlador neuronal predictivo presenta gran adaptabilidad y rechazo a perturbaciones provenientes de las no lineales del sistema, pues el controlador neuronal predictivo es exactamente el mismo para el *pitch* y para el *yaw*. También se pudo verificar que utilizar un modelo discreto para el diseño del controlador es mejor aproximación que un modelo en tiempo continuo en el que no se toman en cuenta el periodo de muestreo ni la presencia de señales discontinuas en el lazo de control, lo cual concuerda con muchos análisis en los que se ha demostrado que no tomar en cuenta estos factores puede provocar un degradamiento en el desempeño del controlador e incluso la inestabilidad del sistema de control.

Asimismo, se concluye que la discretización propuesta con el método de Euler, además de que es fácil de implementar, funcionó correctamente (tanto para sistemas lineales como no lineales) pues los controladores nominal y *reset* funcionaron adecuadamente y fueron diseñados con base en el modelo discreto obtenido con este método. También se pudo verificar la funcionalidad del método de Euler para un sistema lineal y un valor de periodo de muestreo específicos, pues para ese caso en particular se cumplió el mapeo de polos de tiempo continuo a tiempo discreto y se obtuvo exactamente el mismo modelo que el que se obtiene utilizando la función *c2d* del software *Matlab*, la cual utiliza el método del retenedor de orden cero, lo cual concuerda con el hecho de que ambas técnicas son equivalentes.

Sin embargo, tanto el correcto mapeo de polos, así como el correcto funcionamiento del controlador diseñado a partir del método de Euler, se deben a que el periodo de muestreo escogido fue pequeño (del orden de milésimas de segundo), pues el método de Euler es de primer orden, es decir, es un método en el que el error en un tiempo específico de la aproximación con respecto a la solución verdadera del sistema es proporcional al tamaño del periodo de muestreo. En la mayoría de las aplicaciones de control el periodo de muestreo es pequeño, por lo que se conjetura que el método de Euler es suficiente para obtener buenos resultados con el esquema de control propuesto, sin embargo, para sistemas en los que no se pueda hacer esta consideración se deben utilizar métodos de orden mayor (métodos con menor error que el método de Euler dado el mismo periodo de muestreo) para su discretización. Sin embargo, se debe tomar en cuenta que utilizar un método de orden mayor para obtener el modelo discreto implica mayor complejidad desde el punto de vista algebraico. En caso de que el periodo de muestreo sea significativamente más grande que el utilizado en las simulaciones de la presente tesis, se recomienda utilizar el método de Heun. Con este método se puede obtener un modelo

discreto, tanto para sistemas lineales como no lineales, a partir del cual se puede aplicar el esquema de control propuesto. Además, el método de Heun es de segundo orden, por lo tanto, el error es proporcional al cuadrado del periodo de muestreo (el cual generalmente es menor a 1[seg]), con lo cual, se obtendría un modelo discreto con menos error que el obtendría con el método de Euler; en adición, el desarrollo algebraico utilizando el método de Heun no es tan complicado en comparación de otros métodos de orden mayor.

Tomando como referencia los resultados de la sección 4.4 se llegan a las siguientes conclusiones; si los retardos de control y de observación se mantienen por debajo de 0.05[seg] se asegura el cumplimiento del objetivo de control planteado, sin embargo, estos pueden ser mayores a 0.05[seg] por algunos instantes y aún así se puede asegurar el cumplimiento del objetivo de control planteado, con un buen desempeño e incluso con robustez ante perturbaciones. Por otro lado, si estos retardos se mantienen constantemente arriba de 0.05[seg] no se puede asegurar el cumplimiento del objetivo de control planteado pues se pueden presentar oscilaciones en estado permanente y un degradamiento muy severo en el desempeño. Además, debido a que se cumple el objetivo de control sin retardo de comunicación y debido 0.05 segundos es prácticamente el doble de los valores esperados $E(\tau_{y(s_k)})$ y $E(\tau_{u(t_k)})$ con los cuales se diseñó el controlador, se concluye que el sistema de control es robusto a grandes variaciones en el retardo de tiempo. Lo cual se confirma con el hecho de que es necesario que los retardos de comunicación se mantengan constantemente arriba de 0.07[seg], 2.5 veces el valor de $E(\tau_{y(s_k)})$ y $E(\tau_{u(t_k)})$, para que la respuesta del sistema sea inestable. Con estos resultados también se pudo confirmar la superioridad del control neuronal predictivo respecto del nominal pues siempre se obtuvo un mejor desempeño; se concluye entonces que el control neuronal predictivo se adapta al retardo de tiempo para generar una ley de control correctivo que compense sus efectos y es gracias a esta adaptabilidad que el controlador es robusto a un gran rango de valores que pueda tomar el retardo.

Por otra parte, también se concluye que el controlador neuronal predictivo es de práctica implementación ya que el control basado en redes neuronales y el HMM presentan adaptabilidad y no son planteados con base en un modelo; de manera similar, el control basado en *reset* es de fácil implementación pues el diseño de este es con base en la teoría de sistemas lineales en tiempo discreto. Por estas razones, se concluye que el controlador neuronal predictivo puede ser implementado en NCSs reales en los que se tienen muchas perturbaciones e incertidumbres y obtener mejoras significativas en el desempeño. La obtención de resultados en sistemas reales se deja como trabajo a futuro.

Asimismo se puede concluir que las modificaciones propuestas a la red neuronal planteada en (Cho and Fadali (2011)) funcionaron correctamente pues esta genera un control correctivo adaptable que ayuda a mejorar significativamente el desempeño del controlador. Las modificaciones propuestas fueron plantear la entrada de la red neuronal en tiempo discreto y considerar solo dos parámetros de aprendizaje por cada perceptrón. Se demostró que el parámetro que se eliminó no es necesario para obtener mejoras

5. CONCLUSIONES

significativas en el desempeño cuando el objetivo es la estabilización. Sin embargo, como trabajo a futuro se plantea estudiar a la red neuronal con el parámetro eliminado cuando el objetivo es el seguimiento. De manera similar, como trabajo a futuro se tiene estudiar y analizar el uso de otras configuraciones de redes neuronales a NCSs, por ejemplo, las redes neuronales convolucionales y las redes neuronales recurrentes.

Además, se pudo verificar la utilidad del control basado en *reset* a NCSs, más específicamente, la utilidad del elemento reseteado de primer orden (FORE), pues al añadirlo al sistema de control no se afectó la velocidad con la que la salida del sistema se aproxima a cero por primera vez, sin embargo, siempre se obtuvo como respuesta oscilaciones de menor amplitud, un menor sobrepaso y menor tiempo de asentamiento. Estos resultados son similares a los obtenidos en el ejemplo planteado en [Barreiro and Baños \(2012\)](#) donde a un sistema de control lineal que cumple con regulación se le añadió el FORE y como resultado se mantuvo la rapidez del sistema (tiempo de levantamiento), sin embargo, las oscilaciones se redujeron significativamente, mejorando la robustez del sistema y reduciendo el tiempo de asentamiento. Asimismo, en los resultados se pudo observar claramente el comportamiento basado en *reset* de la salida de este controlador, por lo tanto, se concluye que la discretización del FORE propuesta utilizando el método de Euler funciona correctamente con la ventaja de que es un método fácil de implementar. Como trabajo a futuro se propone estudiar otras configuraciones de controladores basados en *reset*, por ejemplo, el controlador reseteado de orden arbitrario.

Los resultados de las simulaciones también demuestran que el control neuronal predictivo es superior al control neuronal (sin la predicción), pues este último es sensible al ruido de medición. Utilizando la predicción es posible seguir contando con el control correctivo de la red neuronal sin que el desempeño se vea afectado por el ruido de medición. Por lo tanto, se concluye que la predicción hecha con el HMM y la ley de control predictivo planteada son útiles para mejorar el desempeño de los NCSs. Además, debido a la mejora significativa del desempeño del controlador neuronal predictivo respecto al nominal se puede concluir que la predicción hecha con el HMM funciona correctamente, pues si la predicción no fuera correcta entonces la red neuronal daría una señal de control que en lugar de mejorar el desempeño lo empeoraría.

Por otra parte, debido a que las probabilidades marginales $P(X(k))$ y $P(Y(k+10))$ evolucionaron de acuerdo a los datos medidos de la salida del sistema, se concluye que el algoritmo de aprendizaje en línea para los parámetros del HMM funcionó correctamente. Esto se puede confirmar con la evolución de las probabilidades $P(X(k)|X(k-1) = x_4)$ y $P(Y(k)|X(k) = x_4)$, pues estas se actualizan de acuerdo con los datos medidos de la salida del sistema. Además, debido a que la predicción al utilizar las probabilidades condicionales es mejor en comparación de la predicción al utilizar las marginales, se demostró la superioridad de la predicción hecha en la presente tesis en comparación de las predicciones hechas en los trabajos [Cho and Fadali \(2011\)](#) y [Cho and Lee \(2008\)](#), pues en estos, a pesar de que utilizan otros modelos probabilísticos gráficos, utilizan las

probabilidades marginales. Sin embargo, se debe tomar en cuenta que las probabilidades condicionales $P(Y(k+10)|O_{10})$ se calculan a partir del método de avance, lo cual provoca que el algoritmo de control sea computacionalmente más costoso, sin embargo, con base en los resultados obtenidos se puede concluir que el vale la pena utilizarlo pues la predicción es significativamente mejor.

Adicionalmente, en el ejemplo presentado las variables aleatorias del HMM únicamente podían tomar 7 estados diferentes, sin embargo, estos resultaron ser más que suficientes para obtener una buena predicción, lo cual deja ver otra ventaja de la predicción con el HMM y es que no se requiere que los espacios de estados sean muy numerosos para conseguir una buena predicción, lo cual, a su vez, hace que el sistema de predicción no sea tan costoso desde el punto de vista computacional. Sin embargo, es importante mencionar que el número de estados depende del rango dinámico del sistema y, por lo tanto, habrá aplicaciones en las que 7 estados sean insuficientes, por ejemplo, para problemas de seguimiento.

Debido a que la predicción obtenida con el HMM funciona correctamente se concluye que el HMM planteado es capaz de modelar la naturaleza estocástica de un NCS en donde están presentes los fenómenos aleatorios del retardo de comunicación, el periodo de muestreo y el ruido de medición. Además, gracias a la predicción fue posible plantear una ley de control predictivo con la cual el desempeño del controlador mejora significativamente, ayudando a compensar los efectos de las perturbaciones aleatorias presentes en estos sistemas. Por estas razones se concluye que se cumplieron los objetivos planteados al inicio de la presente tesis acerca del HMM.

También se concluye que el modo de predecir con el HMM (incluyendo el aprendizaje en línea) sienta las bases para plantear otras leyes de control predictivo en las que el efecto de la predicción en el sistema de control sea más significativo para compensar los efectos del retardo de comunicación. Por ejemplo, se puede adoptar un enfoque parecido al de [Liu and Rees \(2005\)](#), en el que se plantea un control nominal que no considera al retardo y se utiliza una predicción para que su desempeño no se vea afectado en la presencia del retardo de comunicación. Se deja como trabajo a futuro plantear esta y otras leyes de control predictivo.

En adición, en simulaciones que no se muestran en la presente tesis se escogieron las siguientes condiciones iniciales de las verosimilitudes promedio: $M(0) = N(0) = 0$; como resultado se obtuvo una predicción que no captura completamente la naturaleza estocástica del sistema y, por lo tanto, el sistema de control tuvo un peor desempeño en comparación al controlador con las condiciones iniciales [3.34](#). Con base en estos resultados, se concluye que es muy importante definir correctamente a las condiciones iniciales de las matrices $M(k)$ y $N(k)$, pues es fundamental contar con una buena predicción desde el tiempo inicial para permitir un buen aprendizaje de la red neuronal y, en consecuencia, obtener un buen desempeño. Se conjetura que dar un adecuado va-

5. CONCLUSIONES

lor inicial a $M(k)$ y $N(k)$ es especialmente importante cuando el objetivo de control es la estabilización, pues el tiempo en que se tiene una respuesta dinámica es finito y por lo tanto, no se puede asegurar que este sea tiempo suficiente para que el algoritmo de aprendizaje de las matrices $A(k)$ y $B(k)$ pueda capturar la naturaleza estocástica del sistema. Por otra parte, si el objetivo de control es el seguimiento de una señal de referencia, entonces se cuenta con más tiempo de aprendizaje y mayor excitación al sistema, por lo tanto, si se plantea el control neuronal predictivo, las condiciones iniciales de las matrices $M(k)$ y $N(k)$ no serían tan importantes y se puede sacar más provecho al algoritmo de aprendizaje de los parámetros del HMM. Además, con más tiempo de aprendizaje y mayor excitación al sistema la predicción puede mejorar significativamente, con lo cual se pueden sentar las bases para experimentar con otras leyes de control predictivo. Como trabajo a futuro se tiene plantear el control predictivo para cumplir con el seguimiento de una señal de referencia.

Por último, como se demostró con los resultados, la predicción obtenida con el HMM funciona correctamente y se obtienen buenos resultados con fines de control, sin embargo, se plantea como trabajo a futuro probar con otros algoritmos de aprendizaje, por ejemplo, un método de aprendizaje en línea basado en el algoritmo esperanza-maximización. Incluso se plantea probar con el algoritmo esperanza-maximización fuera de línea, es decir, primero resolver el problema de aprendizaje de los parámetros del HMM y asegurar una buena predicción antes de que el sistema de control empiece a funcionar.

Bibliografía

- [1] Barreiro, A. and Baños, A. (2012). Sistemas de control basados en reset. *Revista Iberoamericana de Automática e Informática Industrial*, 9:329–346. [5](#), [90](#)
- [2] Castillo, O. (2020). *Sistemas muestreados estabilizados mediante señal de control pulsada*. PhD thesis, Universidad Nacional Autónoma de México, Programa de maestría y doctorado en ingeniería. [7](#), [8](#)
- [3] Cho, H. C. and Fadali, M. S. (2011). Nonlinear network-induced time delay systems with stochastic learning. *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, 19:843–851. [3](#), [5](#), [15](#), [20](#), [22](#), [23](#), [24](#), [25](#), [27](#), [31](#), [32](#), [35](#), [73](#), [89](#), [90](#)
- [4] Cho, H. C. and Lee, K. S. (2008). Nonlinear networked control systems with random nature using neural approach and dynamic bayesian networks. *International Journal of Control, Automation, and Systems*, 6:444–451. [3](#), [4](#), [5](#), [13](#), [17](#), [24](#), [27](#), [35](#), [73](#), [90](#)
- [5] Cho, H. C. and Lee, K. S. (2009). Online learning algorithm of dynamic bayesian networks for nonstationary signal processing. *International Journal of Innovative Computing, Information and Control*, 5:1027–1041. [32](#), [35](#), [38](#), [39](#), [40](#)
- [6] Fridman, E. (2014). *Introduction to Time-Delay Systems*. Birkhauser, London. [3](#), [6](#), [7](#), [9](#), [11](#), [15](#), [18](#), [46](#)
- [7] Grossman, S. (2010). *Cadenas de Markov y Teoría de Juegos*. Mc Graw Hill, México. [30](#)
- [8] Jurafsky, D. and Martin, J. (2008). *Speech and Language Processing*. Pearson International, New Jersey. [28](#), [29](#), [30](#), [31](#), [38](#)
- [9] Liu, G. and Rees, D. (2005). Design, simulation and implementation of networked predictive control systems. *Measurement + Control*, 38:17–21. [1](#), [5](#), [91](#)
- [10] Ma, L. and Wang, Z. (2019). *Nonlinear Control and Filtering for Stochastic Networked Systems*. Taylor and Francis, New York. [4](#)
- [11] Quanser, I. (2013). Quanser 3 dof hover user manual. *www.quanser.com*. [41](#), [42](#), [44](#)

BIBLIOGRAFÍA

- [12] Theodoridis, S. (2015). *Machine Learning a Bayesian and Optimization Perspective*. Elsevier, London. [27](#), [30](#), [35](#)
- [13] Verriest, E. I. (2002). Stability of systems with state-dependent and random delays. *IMA Journal of Mathematical Control and Information*, 19:103–114. [4](#), [87](#)
- [14] Wanigasekara, C. and Almakhles, D. (2019). Performance of neural network based controllers and pid controllers for networked control systems: A comparative investigation. *IEEE International Conference on Environment and Electrical Engineering*. [4](#)
- [15] Yang, S.-H. (2011). *Internet-based Control Systems*. Springer, London. [1](#)