

VIRGINIA SOLARTE MUÑOZ

EFECTO DE LA LIMITACIÓN DE VALORES DEL  
ALGORITMO ACELERADOR REGRESIVO VERSIÓN  $\gamma$  ( $AR\gamma$ )

Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Maestría en Ingeniería  
Área Electrónica y Telecomunicaciones

Popayán  
2013



VIRGINIA SOLARTE MUÑOZ

EFFECTO DE LA LIMITACIÓN DE VALORES DEL  
ALGORITMO ACELERADOR REGRESIVO VERSIÓN  $\gamma$  (AR $\gamma$ )

Tesis presentada a la Facultad de Ingeniería  
Electrónica y Telecomunicaciones de la  
Universidad del Cauca para la obtención del  
Título de Magíster en Ingeniería

Área:  
Electrónica y Telecomunicaciones

Director:  
Pablo Emilio Jojoa Gómez

Popayán  
2013



*A mis padres por su amor y  
ante todo por ser un ejemplo de vida.  
A Carlos Eduardo  
por ser mi apoyo incondicional y  
a mis hijos,  
por ser mi mayor motivación.*

*Virginia*



# **Agradecimientos**

En especial, agradezco a mi tutor, Ph.D Pablo Emilio Jojoa, quien además de compartir sus conocimientos, me guió y acompañó para alcanzar las metas propuestas.

Igualmente, agradezco a aquellos compañeros y sobretodo amigos de la FIET que siempre brindaron su apoyo a nivel personal y profesional.



# Resumen

Cada vez que existe la necesidad de procesar señales resultantes de ambientes cuyas estadísticas son desconocidas, la utilización de un filtro adaptativo constituye una solución atractiva con respecto a los sistemas de métodos convencionales, ya que con ellos se logran nuevas capacidades de procesamiento de señal que no serían posibles de otra forma. Los algoritmos adaptativos son parte fundamental de estos filtros. Este trabajo presenta el papel que juegan los algoritmos adaptativos en la implementación de los filtros y particularmente enfoca su análisis sobre el Algoritmo Acelerador Regresivo versión  $\gamma$  (AR $\gamma$ ) y su implementación en sistemas de punto fijo.

**Palabras Clave:** *Procesamiento Digital de Señales, Filtros Adaptativos, Algoritmos Adaptativos, Efectos de Precisión Finita.*



# Abstract

Whenever there is a need to process signals resulting from environments whose statistics are unknown, the use of an adaptive filter is an attractive solution, over conventional methods, since they provide new signal processing capabilities that would not be possible otherwise. The adaptive algorithms are an important part of such filters. This document presents the role of adaptive algorithms in the implementation of filters and it particularly focuses its analysis on the Gamma-version Regressive Algorithm Accelerator (AR $\gamma$ ) and its implementation in fixed point systems.

**Palabras Clave:** *Digital Signal Processing, Adaptive Filters, Adaptive Algorithms, Finite Precision Effects.*



# Contenido

	Pág
Lista de Figuras	vii
Lista de Tablas	ix
<b>Capítulo 1. Introducción</b>	<b>1</b>
<b>1.1 Justificación y Motivación</b>	<b>1</b>
<b>1.2 Contribución Original</b>	<b>2</b>
<b>1.3 Organización de la Tesis</b>	<b>3</b>
<b>1.4 Publicaciones</b>	<b>3</b>
<b>Capítulo 2. Algoritmos Adaptativos y Efectos de Precisión Finita</b>	<b>5</b>
<b>2.1 Sistemas Adaptativos</b>	<b>5</b>
2.1.1 Definición de un Sistema Adaptativo	5
2.1.2 Filtros Adaptativos	6
2.1.3 Aplicaciones de los Filtros Adaptativos	8
<b>2.2 Algoritmos Adaptativos</b>	<b>9</b>
2.2.1 Algoritmo LMS	10
2.2.2 Algoritmo NLMS	11
2.2.3 Algoritmo RLS	12
2.2.4 Algoritmo Acelerador Regresivo Versión $\gamma$ ( $AR_{\gamma}$ )	13
<b>2.3 Efectos de Precisión Finita en los Algoritmos Adaptativos</b>	<b>14</b>
2.3.1 Errores de Cuantización	15
2.3.2 Efectos de Precisión Finita en el Algoritmo LMS	17
2.3.3 Efectos de Precisión Finita en el Algoritmo RLS	18
2.3.4 Efectos de Precisión Finita en el Algoritmo $AR_{\gamma}$	20

<b>Capítulo 3.</b>	<b>Implementación del Algoritmo Acelerador Regresivo versión <math>\gamma</math></b>	23
<b>3.1</b>	<b>Definición del Caso de Estudio</b>	23
3.1.1	Caso de estudio 1: Sistema de Identificación usando un Filtro de Respuesta Finita al Impulso (FIR)	24
3.1.2	Caso de estudio 2: Sistema de Identificación usando un Filtro de Respuesta Infinita al Impulso (IIR)	24
<b>3.2</b>	<b>Selección de la Herramienta de Simulación</b>	26
<b>3.3</b>	<b>Implementación del AR<math>\gamma</math> en Simulink</b>	27
3.2.1	Caso de estudio 1: Sistema de Identificación usando un Filtro de Respuesta Finita al Impulso (FIR)	27
3.2.2	Caso de estudio 2: Sistema de Identificación usando un Filtro de Respuesta Infinita al Impulso (IIR)	30
<b>Capítulo 4.</b>	<b>Análisis y Comparación de Resultados</b>	35
<b>4.1</b>	<b>Experiencia 1. Determinación del mínimo número de bits requerido para el funcionamiento del AR<math>\gamma</math></b>	35
4.1.1	Caso de estudio 1: Sistema de Identificación usando un Filtro de Respuesta Finita al Impulso (FIR)	36
4.1.2	Caso de estudio 2: Sistema de Identificación usando un Filtro de Respuesta Infinita al Impulso (IIR)	47
<b>4.2</b>	<b>Experiencia 2. Estudio del efecto de la longitud de palabra en la media y la varianza del error de predicción del AR<math>\gamma</math></b>	55
4.2.1	Caso de estudio 1: Sistema de Identificación usando un Filtro de Respuesta Finita al Impulso (FIR)	55
4.2.2	Caso de estudio 2: Sistema de Identificación usando un Filtro de Respuesta Infinita al Impulso (IIR)	59
<b>4.3</b>	<b>Experiencia 3. Análisis del efecto de la longitud de palabra en los parámetros de ajuste del AR<math>\gamma</math></b>	62
<b>4.4</b>	<b>Experiencia 4. Análisis del error de cuantización en el AR<math>\gamma</math></b>	64
<b>Capítulo 5.</b>	<b>Conclusiones y Recomendaciones</b>	67
<b>5.1</b>	<b>Conclusiones</b>	67
<b>5.2</b>	<b>Recomendaciones</b>	68
<b>Bibliografía</b>		69

<b>Anexo 1</b>	<b>Algoritmo Acelerador Regresivo versión <math>\gamma</math> (AR<math>\gamma</math>)</b>	<b>71</b>
<b>Anexo 2</b>	<b>Sistema de Identificación implementado en Matlab</b>	<b>75</b>
<b>Anexo 3</b>	<b>Códigos de Implementación en Simulink</b>	<b>80</b>
<b>Anexo 4</b>	<b>Parámetros Óptimos</b>	<b>85</b>
<b>Anexo 5</b>	<b>Cuantización</b>	<b>87</b>



## Lista de Figuras

		Pág
Figura 2.1	Arquitectura de un Sistema Adaptativo.	6
Figura 2.2	Elementos de un Filtro Adaptativo.	7
Figura 2.3	Aplicaciones de los Filtros Adaptativos.	9
Figura 2.4	Característica entrada salida de un cuantizador uniforme.	16
Figura 2.5	Diagrama en bloques que representa los efectos de precisión finita del algoritmo LMS.	17
Figura 2.6	Diagrama en bloques que representa los efectos de precisión finita del algoritmo RLS.	17
Figura 3.1	Sistema de Identificación usando un filtro FIR	24
Figura 3.2	Sistema de Identificación usando un filtro IIR	25
Figura 3.3	Sistema de Identificación que compara el AR $\gamma$ con los algoritmos LMS y RL, usando filtros FIR	28
Figura 3.4	Implementación del RLS	29
Figura 3.5	Implementación del LMS	29
Figura 3.6	Implementación del AR $\gamma$	30
Figura 3.7	Vector de Regresión usado en el Caso de Estudio 1	31

Figura 3.8	Sistema de Identificación que compara el AR $\gamma$ con los algoritmos LMS,y RLS, usando filtros IIR	31
Figura 3.9	Vector de Regresión usado en el Caso de Estudio 2	32
Figura 4.1	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.8.56 y sin ruido, usando un filtro FIR	36
Figura 4.2	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.8.56 y con ruido 0.001, usando un filtro FIR	37
Figura 4.3	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.8.56 y con ruido 0.1, usando un filtro FIR	38
Figura 4.4	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.6.58 y sin ruido, usando un filtro FIR	39
Figura 4.5	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.6.58 y con ruido 0.001, usando un filtro FIR	40
Figura 4.6	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.6.58 y con ruido 0.1, usando un filtro FIR	41
Figura 4.7	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.4.60 y sin ruido, usando un filtro FIR	42
Figura 4.8	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.4.60 y con ruido 0.001, usando un filtro FIR	43
Figura 4.9	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.4.60 y con ruido 0. 1, usando un filtro FIR	44
Figura 4.10	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.3.61 y sin ruido, usando un filtro FIR	47
Figura 4.11	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño	

	Q.1.8.56 y sin ruido, usando un filtro IIR	47
Figura 4.12	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.8.56 y con ruido 0.001, usando un filtro IIR	48
Figura 4.13	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.8.56 y con ruido 0.1, usando un filtro IIR	49
Figura 4.14	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.6.58 y sin ruido, usando un filtro IIR	50
Figura 4.15	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.6.58 y con ruido 0.001, usando un filtro IIR	51
Figura 4.16	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.6.58 y con ruido 0.1, usando un filtro IIR	52
Figura 4.17	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.4.60 y sin ruido, usando un filtro IIR	53
Figura 4.18	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.4.28 y sin ruido, usando un filtro FIR	56
Figura 4.19	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.4.12 y sin ruido, usando un filtro FIR	57
Figura 4.20	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.4.4 y sin ruido, usando un filtro IIR	58
Figura 4.21	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.6.26 y sin ruido, usando un filtro IIR	59
Figura 4.22	Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$ con una palabra de tamaño Q.1.6.10 y sin ruido, usando un filtro IIR	60
Figura 4.23	Región de convergencia de los coeficientes para los	



## Lista de Tablas

		Pág
Tabla 2.1	Valores de $H$ para diferentes condiciones de $\alpha, \gamma$ y $m_1$	14
Tabla 4.1	Media y varianza del error de predicción de los algoritmos LMS, RLS y $AR\gamma$ con una palabra de tamaño Q.1.8.56 para diferentes valores de ruido de medida, usando un filtro FIR	38
Tabla 4.2	Media y varianza del error de predicción de los algoritmos LMS, RLS y $AR\gamma$ con una palabra de tamaño Q.1.6.58 para diferentes valores de ruido de medida, usando un filtro FIR	41
Tabla 4.3	Media y varianza del error de predicción de los algoritmos LMS, RLS y $AR\gamma$ con una palabra de tamaño Q.1.4.60 para diferentes valores de ruido de medida, usando un filtro FIR	44
Tabla 4.4	Media y varianza del error de predicción de los algoritmos LMS, RLS y $AR\gamma$ con una palabra de tamaño Q.1.4.60 para diferentes valores de ruido de medida, usando un filtro FIR	46
Tabla 4.5	Media y varianza del error de predicción de los algoritmos LMS, RLS y $AR\gamma$ con una palabra de tamaño Q.1.8.56 para diferentes valores de ruido de medida, usando un filtro IIR	49
Tabla 4.6	Media y varianza del error de predicción de los algoritmos LMS, RLS y $AR\gamma$ con una palabra de tamaño Q.1.6.58 para diferentes valores de ruido de medida, usando un filtro IIR	52
Tabla 4.7	Media y varianza del error de predicción de los algoritmos LMS, RLS y $AR\gamma$ con una palabra de tamaño Q.1.4.60 para diferentes valores con ruido de medida 0.1, usando un filtro IIR	54
	Media y varianza del error de predicción de los algoritmos	

Tabla 4.8	LMS, RLS y $AR\gamma$ con palabras de longitud variable, sin ruido de medida y usando un filtro FIR	58
Tabla 4.9	Media y varianza del error de predicción de los algoritmos LMS, RLS y $AR\gamma$ con palabras de longitud variable, sin ruido de medida y usando un filtro IIR	62
Tabla 4.10	Optimización de Parámetros del $AR\gamma$	65
Tabla 4.11	Varianza del Error de Cuantización	67

## Lista de Abreviaturas

- AR $\gamma$ : *Algoritmo Acelerador Regresivo versión  $\gamma$*
- DSP: *Procesador Digital de Señales (Digital Signal Processor)*
- FIR *Filtro de Respuesta Finita al Impulso (Finite Impulse Response)*
- IIR: *Filtro de Respuesta Infinita al Impulso (Infinite Impulse Response)*
- LMS: *Algoritmo de Mínimos Cuadrados Medio (Least Mean Squared)*
- MSE *Error Cuadrático Medio (Mean Squared Error)*
- NLMS: *Algoritmo de Mínimos Cuadrados Medio Normalizado (Normalized Least Mean Squared)*
- RLS: *Recursivo de Mínimos Cuadrados (Recursive Least Squares)*



# Capítulo 1

## Introducción

### 1.1 Justificación y Motivación

Actualmente, en una sociedad sofisticada en la que la tecnología está ligada a la forma de vida, las señales se encuentran presentes en los dispositivos que se utilizan diariamente y éstas provienen de medios naturales donde las variables son el tiempo, la distancia, la temperatura, etc., y requieren de una transformación a una señal de tipo eléctrica para poder ser procesadas. De esta manera, aquella señal que contiene la información acerca de la entidad o fenómeno que la generó, requiere ser procesada, dando lugar al “Procesamiento de Señales”; área del conocimiento encargada de la interpretación matemática de las señales y del desarrollo de algoritmos para la extracción de la información contenida en las mismas (Frías, 2005).

Según la manera en la que se realice la implementación para el procesamiento, éste puede ser analógico ó digital, diferenciándose fundamentalmente en la forma de manipular la señal. Hoy, debido a las limitaciones del procesamiento analógico en cuanto al ruido, la temperatura y complejidad, el procesado digital ha tomado protagonismo en los últimos años gracias al auge de los circuitos digitales, permitiendo así la implementación de algoritmos complejos en un sinnúmero de sistemas que brindan ventajas en cuanto a confiabilidad, precisión, flexibilidad y tamaño.

Este este contexto, un aspecto trascendental en procesamiento de señales es la degradación del comportamiento de los sistemas cuando las condiciones del escenario cambian con el tiempo, situación que ha obligado a elaborar diseños de nuevos y mejores sistemas que prevengan dichos inconvenientes (Haykin, 2002). Como respuesta a estas necesidades se cuenta con los sistemas adaptativos, que como su nombre lo indica, son capaces de adaptarse automáticamente a entornos no estacionarios donde las estadísticas de la señal son desconocidas y además brindan una alternativa de solución debido a que proporcionan un mejoramiento significativo con relación a los sistemas invariantes con el tiempo en cuanto al rendimiento de las aplicaciones (Manolakis, 2000) (Widrow y Stearns, 1985). Los

filtros adaptativos son sistemas que con el apoyo de algoritmos adaptativos, resuelven los inconvenientes de aquellas aplicaciones que varían con el tiempo ya que se ajustan de manera automática (Haykin, 2002), brindando un mejor desempeño durante el proceso de filtrado (Manolakis, 2000); razón por la cual es indispensable recurrir a aquellos algoritmos cuya respuesta sea rápida (convergencia), que tengan un seguimiento adecuado de las variaciones de la señal (“tracking”) y un desajuste pequeño con respecto a la respuesta esperada (Widrow y Stearns, 1985).

La mayoría de las implementaciones se realizan con algoritmos adaptativos ampliamente conocidos, como son el Algoritmo de Mínimos Cuadrados Medio (LMS, *Least Mean Squared*), el Algoritmo de Mínimos Cuadrados Medio Normalizado (NLMS, *Normalized Least Mean Squared*), (Haykin, 2002), (Manolakis, 2000), (Widrow y Stearns, 1985) y el Algoritmo de Mínimos Cuadrados Recursivo (RLS, *Recursive Least Squares*) y sus variantes (Mathews, 1990). El Algoritmo Acelerador Regresivo Versión  $\gamma$  ( $AR\gamma$ ) (Pait, 1998), (Gerken, Pait y Jojoa, 2000), (Jojoa, 2003), constituye una propuesta que mejora el desempeño buscando un mayor compromiso entre la velocidad de convergencia y el desajuste debido al error de medida.

El análisis de estabilidad del Algoritmo Acelerador Regresivo versión  $\gamma$  ( $AR\gamma$ ) verificado a través de simulaciones en MATLAB, cumple con todas las características de simplicidad, rapidez y adaptabilidad (Jojoa, 2003). Sin embargo; cuando se piensa en utilizar el algoritmo en un sistema real como una FPGA (*Field Programmable Gate Array*) o un DSP (*Digital Signal Processor*), se presentan riesgos debido a que las necesidades comerciales demandan aplicaciones tecnológicas cada vez más rápidas y precisas, llevando al límite la capacidad de los sistemas y en dichos casos, el comportamiento del algoritmo puede variar, razón por la cual se hace indispensable realizar el análisis del efecto de la aritmética de punto fijo en la convergencia del Algoritmo Acelerador Regresivo Versión  $\gamma$  ( $AR\gamma$ ), con el fin de brindar aportes para su implementación en hardware.

## 1.2 Contribución Original

Los estudios existentes sobre el análisis de errores de precisión finita de acuerdo a diferentes criterios como modelos de cuantización, entorno estacionario o no estacionario, análisis transitorio o de estado estable, tipo de entrada, tipo de cuantización, etc.; se han centrado en los algoritmos tipo gradiente (LMS y NMLS) y el de mínimos cuadrados recursivos (RLS) (Haykin, 2002), (Manolakis, 2000).

Los procesos de investigación realizados con respecto al Algoritmo Acelerador Regresivo versión  $\gamma$  ( $AR\gamma$ ), no han incluido el análisis del efecto de la aritmética de punto fijo (Pait, 1998), (Gerken, Pait y Jojoa, 2000), (Jojoa, 2003). La falta de análisis sobre los efectos de la rapidez de convergencia y estabilidad del filtro, al igual que el valor del error cuadrático medio bajo limitaciones de tamaño de

---

palabra, son aspectos desconocidos para el Algoritmo Acelerador Regresivo versión  $\gamma$  ( $AR_{\gamma}$ ).

El análisis objeto de esta investigación es el estudio del efecto de la aritmética de punto fijo del Algoritmo Acelerador Regresivo versión  $\gamma$  ( $AR_{\gamma}$ ) en la convergencia del mismo; razón por la cual esta tesis tomó relevancia brindando un aporte fundamental para la implementación en hardware del algoritmo a partir de la realización de un diagnóstico mediante procesos de simulación.

Como referente para el estudio se utilizaron los típicos problemas de identificación de sistemas con filtros FIR (*Finite Impulse Response*) e IIR (*Infinite Impulse Response*) y para la realización de las pruebas mediante simulación, se implementaron los sistemas en Simulink, utilizando la aritmética de punto fijo, dando como resultados aspectos que permitieron corroborar las bondades del Algoritmo Acelerador Regresivo versión  $\gamma$  ( $AR_{\gamma}$ ) en cuanto a la velocidad de convergencia, la varianza del error y los parámetros de ajuste, contribuyendo de esta forma a la integración de un modelo estadístico de los errores de cuantización del algoritmo y a la posibilidad de una implementación en hardware del mismo.

### **1.3 Organización de la Tesis**

En el Capítulo 2 se presentan los conceptos fundamentales sobre los sistemas adaptativos, incluyendo una descripción de los principales algoritmos utilizados, así como la fundamentación teórica del Algoritmo Acelerador Regresivo Versión  $\gamma$  ( $AR_{\gamma}$ ) y el análisis preliminar desde el punto de vista teórico de los efectos de la utilización de valores finitos en los algoritmos.

El Capítulo 3 describe la forma como se realizó la implementación del Algoritmo Acelerador Regresivo Versión  $\gamma$  ( $AR_{\gamma}$ ) utilizando como aplicación un sistema de identificación en Simulink, tanto para punto flotante como punto fijo.

El Análisis y Comparación de Resultados se presenta en el Capítulo 4, observando el impacto de la utilización de valores finitos sobre la convergencia del algoritmo y finalmente en el Capítulo 5, se exponen las Conclusiones y Recomendaciones de la tesis.

### **1.4 Publicaciones**

Como resultado del trabajo de investigación se han realizado las siguientes publicaciones:

- “El Algoritmo Acelerador Regresivo versión  $\gamma$  ( $AR_{\gamma}$ ) y los efectos de cuantificación”, publicado en el Volumen 1, No.2 de 2012 de la Revista Universitaria en Telecomunicaciones, Informática y Control RUTIC, y
-

- “Efecto de la Limitación de Valores del Algoritmo Acelerador Regresivo Versión  $\gamma$  (AR $\gamma$ )”, presentado en el IV Congreso de Telecomunicaciones, Tecnologías de la Información y Comunicaciones, realizado en Quito, Ecuador, durante los días 6, 7 y 8 de Julio de 2011.
-

## Capítulo 2

# Algoritmos Adaptativos y Efectos de Precisión Finita

En este capítulo inicialmente se presentan los conceptos teóricos relacionados con los sistemas adaptativos, dando lugar a la utilización de filtros adaptativos en un amplio rango de aplicaciones, donde los algoritmos adaptativos hacen parte fundamental de los mismos. Posteriormente, ya en el contexto de los algoritmos, en este capítulo se estudian los más comunes incluyendo el Algoritmo Acelerador Regresivo Versión  $\gamma$  ( $AR\gamma$ ), objeto de este estudio. El capítulo finaliza con una exposición sobre los efectos de precisión finita en los algoritmos.

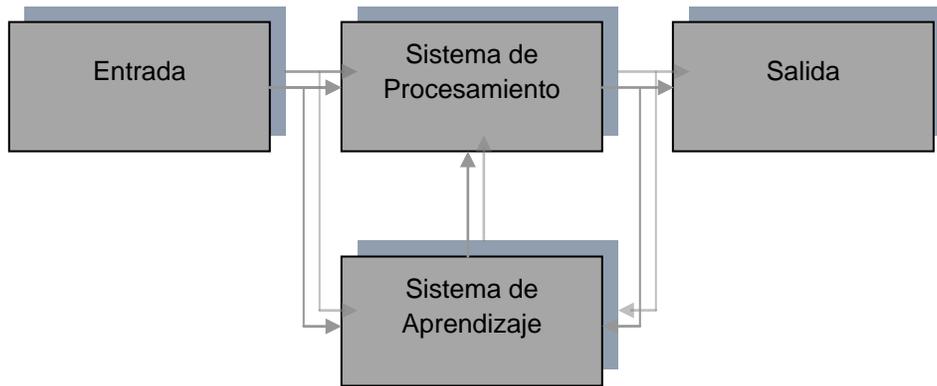
### 2.1 Sistemas Adaptativos

En aquellos escenarios en los cuales las características de las señales tienen un comportamiento aleatorio, aún los mejores diseños de sistemas de tratamiento de señales pueden dar lugar a degradaciones; situación que puede mejorarse mediante la utilización de sistemas capaces de adecuarse a un ambiente variable, aspecto importante en el momento de realizar un diseño.

#### 2.1.1 Definición de un Sistema Adaptativo

Un sistema adaptativo es aquel cuyos parámetros de diseño son alterables o ajustables de acuerdo a algún criterio deseado, de tal forma que su comportamiento o desempeño mejore, proporcionando nuevas capacidades que no se pueden lograr con sistemas de parámetros fijos (Manolakis, 2000).

En la Figura 2.1, se presenta la arquitectura genérica de un sistema adaptativo conformado por un sistema de procesamiento constituido generalmente por un sistema lineal al que se le superpone una estructura llamada sistema de aprendizaje, la cual observa las condiciones e introduce las modificaciones pertinentes en el sistema de procesamiento (Haykin, 2002), (Manolakis, 2000), (Widrow y Stearns, 1985).



**Figura 2.1 Arquitectura de un Sistema Adaptativo**

Los sistemas adaptativos surgen de la imitación de procesos naturales, como por ejemplo múltiples funciones presentes en nuestro propio organismo, o simplemente observando la evolución del hombre y los seres vivos y la capacidad de estos de adaptarse a su entorno o medio ambiente.

Entre las características de los sistemas adaptativos se pueden mencionar las siguientes (Widrow y Stearns, 1985):

- Se pueden adaptar automáticamente a ambientes y sistemas que varían con el tiempo.
- Pueden ser entrenados para desempeñar tareas específicas tales como filtraje o toma de decisiones.
- En general son más complejos y difíciles de analizar que aquellos sistemas que no poseen la capacidad de adaptarse a su entorno, pero ofrecen la posibilidad de presentar un desempeño mejor cuando las características de las señales de entrada son desconocidas o variantes con el tiempo.

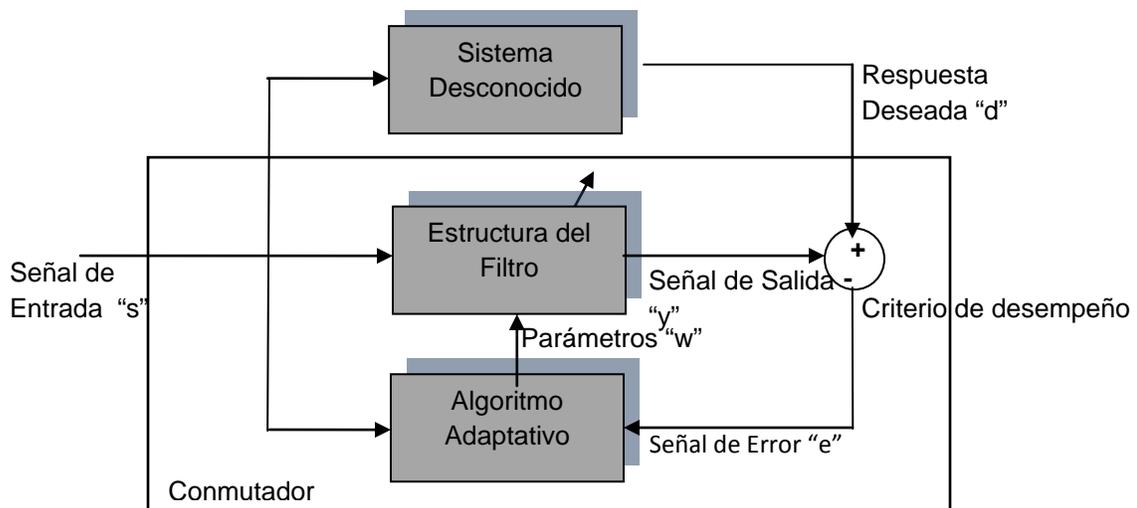
En aquellos ambientes en los cuales todas las condiciones en las que el sistema opera son conocidas, no es necesario implementar sistemas adaptativos ya que solo se requiere especificar un criterio de desempeño determinado para cada una de ellas y esperar así el mejor resultado. En contraste, en la mayoría de los problemas extraídos de la realidad, las condiciones de entrada no son totalmente conocidas y pueden cambiar o mutar con el tiempo. En estas circunstancias, un sistema adaptativo puede adecuarse continuamente y lograr así un mejor desempeño (Widrow y Stearns, 1985).

### 2.1.2 Filtros Adaptativos

En muchas aplicaciones se requiere el uso de filtros. De manera particular, los filtros digitales con coeficientes constantes no siempre brindan soluciones

adecuadas para sistemas variantes en el tiempo debido a que fueron implementados para condiciones específicas de diseño. En el contexto de los sistemas adaptativos, los filtros adaptativos resuelven los inconvenientes de aquellas aplicaciones que varían con el tiempo ya que se ajustan de manera automática, brindando un mejor desempeño durante el proceso de filtrado (Haykin, 2002), (Manolakis, 2000), (Marques, 1998).

Los filtros adaptativos son estructuras ajustables que tienen la capacidad de cambiar su respuesta automáticamente para mejorar su desempeño. Están compuestos básicamente por dos módulos fundamentales que son la estructura del filtro y el algoritmo (Manolakis, 2000), como se indica en la Figura 2.2.



**Figura 2.2 Elementos de un Filtro Adaptativo**

La estructura del filtro realiza el procesamiento de la señal de entrada según los coeficientes entregados por el algoritmo adaptativo y este algoritmo es el encargado de tomar las señales de salida del filtro y compararlas con la señal de respuesta deseada para evaluar los requerimientos de acuerdo a una aplicación en particular. La estructura del filtro es definida por el diseñador y puede ser una estructura del tipo FIR (*Filtro de Respuesta Finita al Impulso*) o de tipo IIR (*Filtro de Respuesta Infinita al Impulso*), y sus parámetros son ajustados por el algoritmo adaptativo (Manolakis, 2000).

El algoritmo adaptativo usa el valor del criterio de desempeño o alguna función de él, la señal de entrada y la respuesta deseada para decidir cómo se deben modificar los parámetros del filtro y mejorar su desempeño. La señal de salida del filtro y la señal de respuesta deseada son procesadas por un criterio de desempeño para evaluar la calidad respecto a los requerimientos de una aplicación en particular.

### 2.1.3 Aplicaciones de los Filtros Adaptativos

Los algoritmos adaptativos se pueden implementar generalmente en procesos relacionados con identificación de sistemas, inversión de sistemas, seguimiento de señales (predicción de señales) y cancelamiento (eliminación) de interferencias (Mulgrew y Cowan, 1998). Sus aplicaciones en ingeniería abarcan una amplia gama de campos como sistemas de comunicaciones, filtrado de interferencias, canceladores de eco telefónico y acústico, sistemas de control, identificación de sistemas, sistemas de navegación, sismología, ingeniería biomédica, entre otras (Marques, 1998).

A continuación se explican brevemente las aplicaciones de los filtros adaptativos y se ilustran en la Figura 2.3 (Haykin, 2002).

- **Identificación de Sistemas:** En la Figura 2.3.a) la señal de entrada  $s$  se introduce simultáneamente al filtro adaptativo y al sistema desconocido, el cual genera la señal  $d$ . La diferencia entre la señal  $d$  (señal deseada) y la señal  $y$  obtenida a través del filtro, produce un error  $e$ , que al ser considerado por el sistema de aprendizaje del algoritmo adaptativo que gobierna el filtro, modifica los coeficientes de éste de manera que produzca una señal  $y$  que minimiza el error. Cuando se obtiene el mínimo error, se puede afirmar que el filtro adaptativo realiza la misma tarea que la estructura desconocida y por lo tanto el sistema se ha identificado.
  - **Inversión de Sistemas:** Esta clase de aplicación mostrada en la Figura 2.3.b), se conoce comúnmente como modelación inversa de sistema. En este caso, el filtro adaptativo intenta recuperar mediante la minimización del error  $e$ , la señal  $s$  que fue alterada por un sistema desconocido y ruido aditivo.
  - **Predicción de señales:** En esta aplicación, el filtro adaptativo en el esquema de la Figura 2.3.c), tiene como función realizar un seguimiento de la señal de entrada  $s$  con el fin de minimizar al máximo el error de salida  $e$ , de manera que a partir de datos anteriores obtenidos de la señal  $x$ , se prediga la señal deseada  $d$ .
  - **Eliminación de Interferencias:** En el sistema de la Figura 2.3.d), el filtro adaptativo elimina las interferencias comparando la señal de entrada con una señal base. En este caso, la señal de entrada  $s$  está contaminada con ruido aditivo  $r$ , generando la señal deseada  $d$ , la señal  $r'$  es una aproximación del ruido aditivo  $r$ . El objetivo del filtro adaptativo es el de producir una señal  $y$  a partir de  $r'$ , de manera que el error  $e$  sea prácticamente igual a  $s$ , eliminando el ruido aditivo.
-

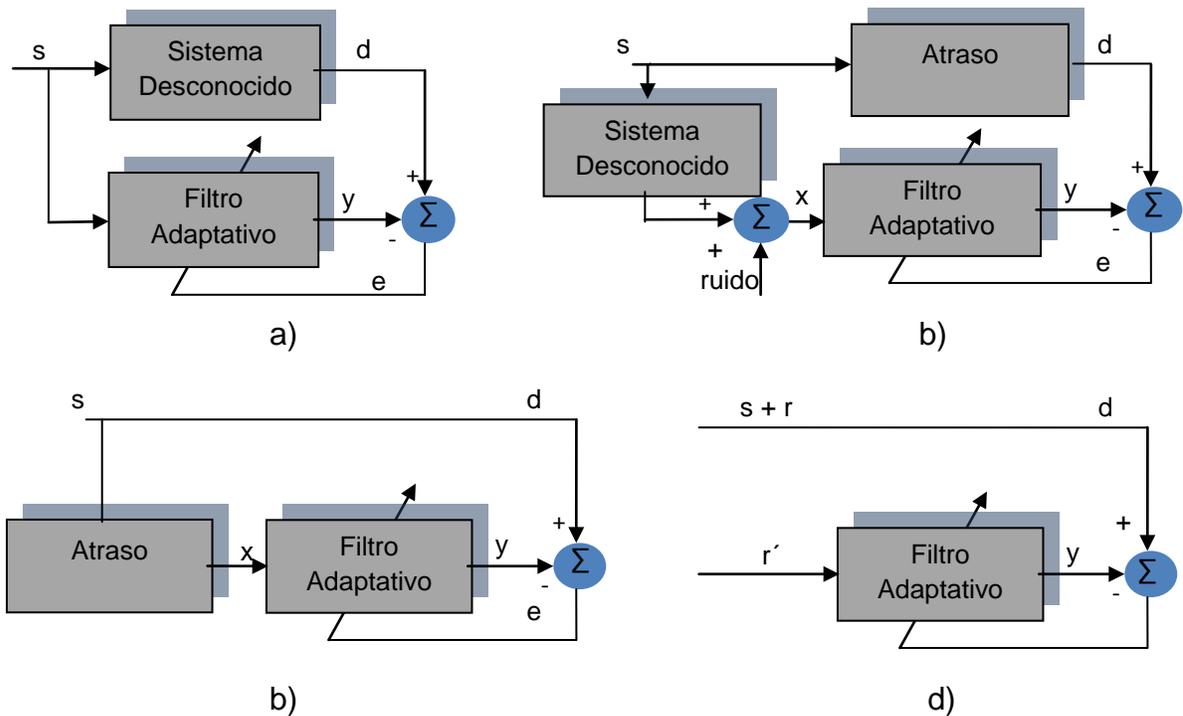


Figura 2.3 Aplicaciones de los Filtros Adaptativos (Haykin, 2002)

## 2.2 Algoritmos Adaptativos

Para la operación de un filtro adaptativo, se han desarrollado una gran variedad de algoritmos. Los factores determinantes para escoger uno u otro algoritmo son los siguientes (Haykin, 2002):

- **Velocidad de convergencia:** Definida como el número de iteraciones requeridas por el algoritmo en respuesta a las entradas estacionarias. Una rápida velocidad de convergencia permite que el algoritmo se adapte rápidamente a un entorno estacionario.
- **Desajuste:** Provee una medida cuantitativa de la desviación entre el valor final del error cuadrado medio y el error mínimo.
- **Tracking.** Relacionado con la capacidad del algoritmo de seguir las variaciones estadísticas del entorno.
- **Robustez:** Para que un filtro adaptativo sea robusto, pequeñas alteraciones pueden resultar de pequeñas estimaciones de error.
- **Requerimientos computacionales:** Corresponde al número de operaciones requeridas para hacer una iteración del algoritmo y el tamaño de localidades de memoria requeridas para almacenar el programa.
- **Estructura:** Se refiere a la estructura de información que fluye por el algoritmo, determinando la manera en la cual este es implementado en hardware.

- Propiedades numéricas: Cuando un algoritmo se implementa numéricamente, las imprecisiones producen cuantización de error, el cual se obtiene de conversiones analógico-digital. Se presentan entonces problemas de estabilidad numérica y la exactitud numérica, lo cual es determinado por el número de bits usados en la representación numérica de datos y coeficientes del filtro.

Por consiguiente, es indispensable recurrir a aquellos algoritmos cuya respuesta sea rápida (convergencia), que tengan un seguimiento adecuado de las variaciones de la señal (tracking) y un desajuste pequeño con respecto a la respuesta esperada.

En la actualidad existe una gran variedad de algoritmos adaptativos siendo los más conocidos los algoritmos LMS (*Least Mean Squared*) y NLMS (*Normalized Least Mean Squared*), el de Mínimos Cuadrados Recursivo (RLS, *Recursive Least Squares*) y sus variantes y el Algoritmo Acelerador Regresivo Versión  $\gamma$  (AR $\gamma$ ), los cuales se describen en las siguientes secciones.

### 2.2.1 Algoritmo LMS

El algoritmo LMS fue propuesto por Widrow y Hoff en el año 1960. Es un importante miembro de la familia de algoritmos de gradiente estocástico, ampliamente utilizado debido a su simplicidad.

Para ilustrar el comportamiento de este algoritmo se ha tomado una aplicación clásica del filtraje adaptativo, que consiste en la identificación de un sistema (Figura 2.3a), en donde a través del algoritmo adaptativo se busca determinar el modelo que rige el comportamiento de un sistema desconocido. Para tal fin, la señal de entrada  $x[n]$  se introduce simultáneamente a un sistema desconocido, el cual genera una señal deseada  $d[n]$  y al filtro adaptativo cuya función es realizar una combinación lineal entre  $x[n]$  y un conjunto de pesos denominados coeficientes del filtro adaptativo  $\mathbf{w}[n]$ , generando así una señal de salida  $y[n]$ .

Por medio de un proceso de adaptación, el filtro realiza el ajuste automático de los coeficientes, teniendo como base la señal de error  $e[n]$  que es obtenida a través de la comparación de la señal de salida  $y[n]$  con la señal de salida deseada  $e[n] = d[n] - y[n]$ . Con el objetivo de obtener un error  $e[n]$  mínimo, el algoritmo LMS procura minimizar la función  $J(\mathbf{w}[n])$ , denominada función de costo, la cual depende de la estimación del error por medio del cálculo del Error Cuadrático Medio (MSE con  $MSE = [E|e[n]|^2]$ ), ajustando sucesivamente los coeficientes del filtro en la dirección opuesta al vector gradiente  $\nabla J(\mathbf{w}[n])$  (Marqués, 1998), (Jojoa, 2003), (Realpe, 2009).

A continuación se presentan las ecuaciones que describen este algoritmo:

---

$$d[n] = \mathbf{x}^T [n] \mathbf{w}_o + r[n], \quad (2.1)$$

$$e[n] = d[n] - \mathbf{w}^T [n] \mathbf{x}[n], \quad (2.2)$$

$$\mathbf{w}[n + 1] = \mathbf{w}[n] + \mu \mathbf{x}[n] e[n] \quad (2.3)$$

donde (Haykin, 2002):

$\mathbf{x}[n]$ : Vector de la señal de entrada.

$\mathbf{w}[n]$ : Vector de coeficientes del filtro adaptativo.

$\mathbf{w}_o[n]$ : Vector de coeficientes (Sistema Desconocido).

$r[n]$ : Ruido de medida.

$d[n]$ : Señal deseada.

$e[n]$ : Error de medida.

$\mu$ : Parámetro de ajuste fijo.

El criterio de convergencia del algoritmo establece la siguiente condición para los valores de  $\mu$  (Haykin, 2002):

$$0 < \mu < \frac{2}{\lambda_{max}} \quad (2.4)$$

donde  $\lambda_{max}$  es el máximo valor de los autovalores de la matriz de autocorrelación de la señal de entrada  $\mathbf{x}[n]$  (Haykin, 2002).

La importancia práctica de los filtros LMS es debido a su simplicidad de implementación. Su mayor limitación es la relativa baja velocidad de convergencia, para mantener la estabilidad.

### 2.2.2 Algoritmo NLMS

En la forma estándar del LMS presentada anteriormente, el ajuste aplicado en la iteración  $n + 1$  depende del producto de tres términos, a saber: el parámetro de ajuste  $\mu$ , el vector de entrada  $\mathbf{x}[n]$  y el error de medida  $e[n]$ . Dado que el ajuste es directamente proporcional al vector de entrada  $\mathbf{x}[n]$ , cuando éste es extenso, el filtro LMS presenta problemas de amplificación del gradiente del ruido (Haykin, 2002). Para superar esta dificultad, y amortiguar las amplias variaciones de la señal de entrada, se utiliza la normalización del LMS, conocida como NMLS, en la que la actualización de los coeficientes está dada por la siguiente expresión:

$$\mathbf{w}[n + 1] = \mathbf{w}[n] + \frac{\hat{\mu}}{\epsilon + \|\mathbf{x}[n]\|^2} \mathbf{x}[n] e[n] \quad (2.5)$$

Con  $\epsilon > 0$  y como criterio de convergencia,  $\hat{\mu}$  debe satisfacer la siguiente condición:

$$0 < \hat{\mu} < 2 \quad (2.6)$$

Otra ventaja del NMLS con relación al LMS es que la velocidad de convergencia es potencialmente mayor.

### 2.2.3 Algoritmo RLS

A partir del análisis de mínimos cuadrados en el cual se minimiza de forma determinística la suma de los cuadrados de errores parciales, se deriva el algoritmo RLS (*Recursive Least Square*) (Marqués, 1998). La diferencia fundamental entre el algoritmo LMS y el RLS radica en que el parámetro de ajuste  $\mu$  en el LMS es reemplazado por la matriz de correlación inversa del vector de entrada  $x[n]$ , la cual tiene un efecto de limpieza en las entradas. Esta modificación tiene un profundo impacto en el comportamiento de la convergencia del algoritmo RLS en un entorno estacionario ya que la velocidad de convergencia es típicamente un orden de magnitud mayor que el algoritmo LMS. Sus desventajas son la mayor exigencia computacional y los problemas de inestabilidad (Haykin, 2002)

A continuación se indican las ecuaciones que describen la operación de filtraje del algoritmo donde se observa que el algoritmo RLS utiliza la ganancia de Kalman para hallar su solución, el cual se basa en la matriz de auto correlación de datos y de la matriz  $\mathbf{P}(n)$  (Hayquin, 2002), (Lennart, 1999), (Escobar, 2003):

$$\mathbf{w}[n] = \mathbf{w}[n - 1] + \mathbf{k}[n]\varepsilon^*[n] \quad (2.7)$$

$$\mathbf{k}[n] = \frac{\mathbf{P}[n-1]\mathbf{x}[n]}{\lambda + \mathbf{x}^H[n]\mathbf{P}[n-1]\mathbf{x}[n]} \quad (2.8)$$

$$\mathbf{P}[n] = \lambda^{-1}\mathbf{P}[n - 1] - \lambda^{-1}\mathbf{k}[n]\mathbf{x}^H[n]\mathbf{P}[n - 1] \quad (2.9)$$

donde

$$\varepsilon[n] = d[n] - \mathbf{w}^H[n - 1]\mathbf{x}[n] \quad (2.10)$$

y

$\mathbf{w}[n]$ : Vector de coeficientes del filtro.

$\mathbf{k}[n]$ : Vector de ganancia  $M \times 1$ , conocido como ganancia de Kalman.

$\mathbf{x}[n]$ : Vector de la señal de entrada.

$d[n]$ : Señal deseada.

$\varepsilon[n]$ : Error.

$\lambda$  : Factor de olvido

$\mathbf{P}[n]$ : Matriz  $M \times M$ , correspondiente a la Ecuación de Ricatti, usada para la actualización de la ganancia.

### 2.2.4 Algoritmo Acelerador Regresivo Versión $\gamma$ ( $AR_\gamma$ )

En 1998, fue propuesto por F. Pait (Pait, 1998), el Algoritmo Acelerador de Tiempo Continuo que consistió en el ajuste de la segunda derivada (aceleración) del error con respecto a los coeficientes del algoritmo (Ver Anexo 1). La principal característica analizada fue el mejor compromiso entre la velocidad de convergencia y la variación del error de estimación, en relación a los algoritmos de gradiente estocástico (Pait, 1998).

En 1999 se realizó la discretización del algoritmo acelerador, generando tres versiones del algoritmo en tiempo discreto (Jojoa, 1999). Estas versiones fueron obtenidas utilizando los métodos de discretización de Euler y la transformación bilineal. Los resultados obtenidos con los métodos de discretización de Euler arrojaron buenos resultados con respecto a desempeño y complejidad computacional. Los algoritmos generados fueron el Algoritmo Acelerador Progresivo Convencional Matricial (APCM), obtenido con el método de Euler progresivo y el Algoritmo Acelerador Regresivo Convencional Matricial (ARCM), obtenido con el método de Euler regresivo.

El algoritmo APCM presenta una baja complejidad computacional, en contraste con el buen desempeño y la estabilidad del algoritmo ARCM. (Jojoa, 1999)

Con el objetivo de disminuir la complejidad computacional del algoritmo ARCM, se llegó a obtener el Algoritmo Acelerador Regresivo Versión  $\gamma$ , el cual es una versión simplificada del algoritmo ARCM, cuya principal característica es la disminución del número de parámetros de ajuste de cuatro a tres, denominados  $\alpha$ ,  $\gamma$  y  $m_1$  por medio de los cuales se logra una buena velocidad de convergencia y paralelamente una considerable reducción del error de medida final (Gerken, et al., 2000).

Las ecuaciones que describen el Algoritmo Acelerador Regresivo Versión  $\gamma$  son:

$$e[n] = \mathbf{x}^T[n]\mathbf{w}[n-1] - d[n] \quad (2.11)$$

$$g[n] = \frac{e[n] + \gamma \mathbf{x}^T[n]\mathbf{q}[n-1]}{1 + \alpha \gamma \mathbf{x}^T[n]m_1 \mathbf{x}[n]} \quad (2.12)$$

$$\mathbf{q}[n] = \frac{\gamma}{\alpha + \gamma} \mathbf{q}[n-1] - \alpha g[n]m_1 \mathbf{x}[n] \quad (2.13)$$

$$\mathbf{w}[n] = \mathbf{w}[n-1] + \alpha \mathbf{q}[n] \quad (2.14)$$

$d[n]$  corresponde a la señal deseada obtenida de la siguiente forma:

$$d[n] = \mathbf{x}^T[n]\mathbf{w}_o[n] + r(n) \quad (2.15)$$

---

Con:

$x[n]$ : Señal de entrada

$\mathbf{w}[n]$ : Vector de coeficientes del filtro adaptativo.

$d[n]$ : Señal deseada.

$e[n]$ : Error de medida.

$g[n]$ : Escalar auxiliar en el instante  $n$ .

$\mathbf{q}[n]$ : Vector auxiliar de error.

$\mathbf{w}_o[n]$ : Vector de coeficientes óptimo.

$\alpha$ : Parámetro de ajuste fijo.

$\gamma$ : Parámetro de ajuste fijo.

$m_1$ : Parámetro de ajuste fijo.

La ventaja que presenta este algoritmo es que los parámetros de ajuste se reducen a cantidades escalares ( $\alpha, \gamma$  y  $m_1$ ). De acuerdo al análisis de convergencia con respecto a la media (Jojoa, 2003), se establece que el algoritmo converge para valores de  $\alpha, \gamma$  y  $m_1$  positivos. Así mismo del análisis de tracking realizado, en ambientes no estacionarios, se determinó que éste presenta un mínimo error de desajuste cuando  $\alpha\gamma m_1 \approx H$  (Criterio de Mínimo Error) en que  $H$  es una constante real positiva con un valor aproximado a 2. En la Tabla 2.1, se indican algunos valores de  $H$  obtenidos de manera teórica para diversas combinaciones de  $\alpha, \gamma$  y  $m_1$  con los que se alcanza un mínimo desajuste.

$m_1$	$\alpha$	$\gamma$	$H$
1	1.0	2.04	2.040
1	1.5	1.35	2.025
3	1.0	0.68	2.040
3	1.5	0.45	2.025
11	1.0	0.18	1.980
11	1.5	0.12	1.980

**Tabla 2.1 Valores de  $H$  para diferentes condiciones de  $\alpha, \gamma$  y  $m_1$  (Jojoa, 2003)**

### 2.3 Efectos de Precisión Finita en los Algoritmos Adaptativos

El análisis de aritmética presentado hasta el momento para los diferentes algoritmos, asume un modelo de tipo analógico, es decir, de precisión infinita tanto para los datos de entrada como para los cálculos internos del algoritmo. Sin embargo, esta teoría no es aplicable a la construcción de un filtro adaptativo de manera directa, ya que proporciona un marco idealizado.

El estudio sobre los algoritmos adaptativos quedaría incompleto sin una discusión sobre los efectos de cuantificación o redondeo; errores que surgen cuando los filtros se implementan digitalmente.

---

En particular, para la implementación digital de un filtro con un algoritmo adaptativo, los datos de entrada y los cálculos internos son cuantizados para efectos de precisión finita, situación determinante para consideraciones de diseño y costo. De esta forma, el proceso de cuantización tiene el efecto de generar un impacto en el rendimiento del algoritmo, inclusive; desviando los resultados de los obtenidos en la teoría, siendo los factores más trascendentales los detalles de diseño del algoritmo, el grado de efectos negativos a nivel de las características de la matriz de correlación de los datos de entrada y la forma de computación numérica empleada (por ejemplo punto fijo o punto flotante) (Haykin, 2002), (Nabil y Ali, 2003).

De esta manera, es importante entender las propiedades numéricas de los algoritmos adaptativos y en particular del Algoritmo Acelerador Regresivo versión  $\gamma$ , ya que así se pueden encontrar muchas especificaciones de diseño al analizar el efecto de la limitación de valores obteniendo patrones de comportamiento trascendentales para futuras implementaciones en hardware que pueden representar costos operacionales substancialmente menores y niveles de consumo de potencia pequeños.

A continuación y con el fin de entender los efectos de precisión finita, se examinan las fuentes de error de cuantización más comunes y sus efectos en la estabilidad numérica de los algoritmos.

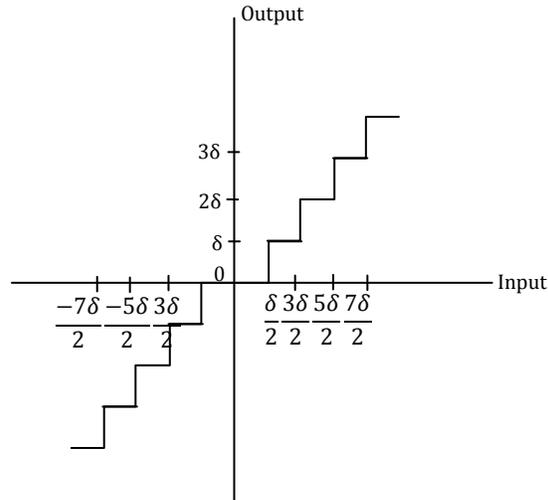
### 2.3.1 Errores de Cuantización

En la implementación de un filtro adaptativo se tienen dos fuentes de error de cuantización (Haykin, 2002):

- A. *Conversión analógica – digital.* Considerando que los datos de entrada son de forma analógica, se requiere de un conversor analógico digital para realizar la representación numérica. Para la presente discusión, se asume un proceso de cuantización con un tamaño de paso uniforme  $\delta$  y un conjunto de niveles de cuantización posicionados en  $0, \pm\delta, \pm2\delta, \dots$ . En la Figura 2.4 se ilustra la característica de un cuantizador uniforme típico.

Se considera una muestra particular en la entrada del cuantizador, con una amplitud que varía en el rango  $r\delta - (\delta/2)$  a  $r\delta + (\delta/2)$ , donde  $r$  es un entero (positivo, negativo o cero) y  $r\delta$  define la salida del cuantizador. De manera semejante, el proceso de cuantización introduce una región de incertidumbre de ancho  $\delta$ , centrado en  $r\delta$ . Definiendo  $\eta$  como el error de cuantización, la entrada del cuantizador será  $r\delta + \eta$ , donde  $-(\delta/2) \leq \eta \leq \delta/2$ . Mientras el cuantizador es lo suficientemente bueno (es decir que los niveles de cuantización son 64 o más) y el espectro de la señal es lo suficientemente rica, la distorsión producida por el proceso de cuantización puede ser modelada como una fuente aditiva independiente de ruido blanco con media cero y varianza determinada por el tamaño del paso  $\delta$  del cuantizador (Gray, 1990).

---



**Figura 2.4 Característica entrada salida de un cuantizador uniforme**

Es habitual asumir que el error de cuantización  $\eta$  es uniformemente distribuido para el rango  $-\delta/2$  a  $\delta/2$ . La varianza del error de cuantización está dada por:

$$\sigma^2 = \int_{-\delta/2}^{+\delta/2} \frac{1}{\delta} \eta^2 d\eta \quad (2.16)$$

Asumiendo que la entrada del cuantizador tiene una escala apropiada, tal que se encuentra en el intervalo  $(-1, +1]$ , con  $B$  bits, el paso del cuantizador es:

$$\delta = 2^{-B} \quad (2.17)$$

Reemplazando la ecuación (2.17) en la (2.16), el error de cuantización resultante de la representación de la entrada analógica de datos tiene la varianza

$$\sigma^2 = \frac{2^{-2B}}{12} \quad (2.18)$$

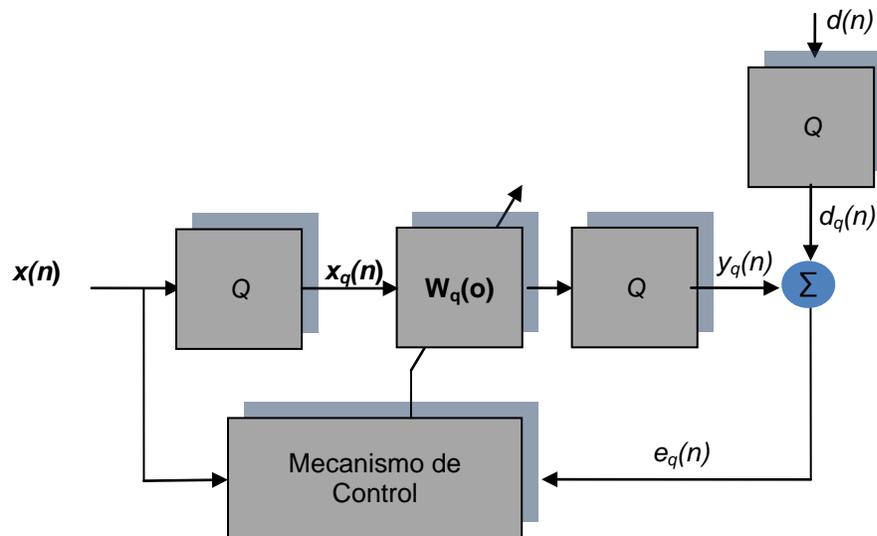
- B. *Aritmética de longitud de palabra finita.* En una máquina digital, la longitud de palabra finita es utilizada para almacenar los resultados de los cálculos internos. Asumiendo que no se presenta desbordamiento durante el cómputo, las sumas no introducen ningún error, utilizando aritmética de punto fijo, mientras que cada multiplicación introduce un error después de que cada producto es cuantizado.

La presencia de aritmética de palabra finita, representa un serio problema en la implementación de un filtro adaptativo, particularmente en lo relacionado con el peso de los coeficientes del filtro que son actualizados en una base continua. La versión digital del filtro muestra una respuesta específica de tales errores, causando una desviación de la respuesta ideal. Inclusive, el filtro puede llegar a una condición de “overflow”, generando un algoritmo numéricamente inestable. Otro aspecto importante en la implementación de un filtro adaptativo es la precisión, que depende del número de bits usados para implementar los cálculos internos del filtro.

A continuación se describen los efectos de precisión finita de manera general para los algoritmos LMS, RLS y a partir de ellos se definen para el algoritmo AR $\gamma$ .

### 2.3.2 Efectos de Precisión Finita para el Algoritmo LMS

En la figura 2.5 se presenta un diagrama en bloques de los efectos de precisión finita del LMS. Cada uno de los bloques etiquetados con Q representa un cuantizador, el cual introduce un error.



**Figura 2.5 Diagrama en bloques que representa los efectos de precisión finita del algoritmo LMS (Haykin, 2002)**

Las relaciones entrada salida de las operaciones de los cuantizadores de la figura se describen a continuación:

A la entrada del cuantizador se conecta  $x[n]$ , de donde se obtiene que

$$x_q[n] = Q[x[n]] = x[n] + \eta_x[n] \quad (2.19)$$

donde  $\eta_x[n]$  corresponde al vector de entrada de error de cuantización.

Para el cuantizador conectado a la respuesta deseada  $d[n]$ , se tiene que:

$$d_q[n] = Q[d[n]] = d[n] + \eta_d[n] \quad (2.20)$$

donde  $\eta_d[n]$  es el error de cuantización de la respuesta deseada.

Para la cuantización del vector  $\mathbf{w}_q[n]$ ,

$$\mathbf{w}_q[n] = Q[\mathbf{w}[n]] = \mathbf{w}[n] + \Delta\mathbf{w}[n] \quad (2.21)$$

Teniendo en cuenta que  $\mathbf{w}[n]$  es el vector en precisión infinita y  $\Delta\mathbf{w}[n]$  es el vector del error resultante de la cuantización.

Para el cuantizador a la salida del filtro, se considera que:

$$y_q[n] = Q[x_q^T[n]\mathbf{w}_q[n]] = x_q^T[n]\mathbf{w}_q[n] + \eta_y[n] \quad (2.22)$$

donde  $\eta_y[n]$  es el error de cuantización a la salida del filtro.

La precisión finita en el Algoritmo LMS está descrita entonces por las siguientes ecuaciones:

$$e_q[n] = d_q[n] - y_q[n] \quad (2.23)$$

y

$$\mathbf{w}_q[n+1] = \mathbf{w}_q[n] + Q[\mu e_q[n]x_q[n]] \quad (2.24)$$

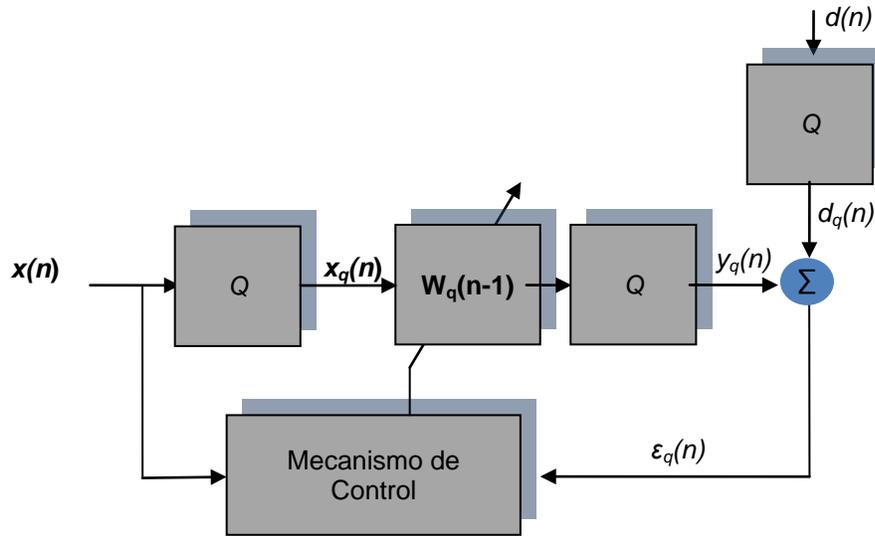
La operación de cuantización de la derecha de la ecuación (2.24), no se indica explícitamente en la figura 2.5, aunque es indispensable para la operación en precisión finita del algoritmo<sup>1</sup>.

### 2.3.3 Efectos de Precisión Finita para el Algoritmo RLS

El algoritmo RLS ofrece una solución alternativa al algoritmo LMS para la solución de problemas de filtraje adaptativo. Se conoce que el algoritmo RLS se caracteriza por una rápida velocidad de convergencia y un desajuste insignificante.

<sup>1</sup> Los primeros estudios de precisión finita del LMS fueron presentados por Gitlin (1963). Posteriormente, detalles sobre estos efectos los presentaron Weiss y Mitra (1979), Caraiscos y Liu (1984) y Alexander (1987). Un estudio más a fondo sobre la precisión finita fué presentado por Cioffi (1987), Sherwood y Bershad (1987).

En la Figura 2.6 se presenta un diagrama en bloques de los efectos de precisión finita para el RLS. Cada uno de los bloques etiquetados con Q representa un cuantizador, el cual introduce un error.



**Figura 2.6 Diagrama en bloques que representa los efectos de precisión finita del algoritmo RLS (Haykin, 2002)**

Las relaciones entrada salida de las operaciones de los cuantizadores de la figura se describen a continuación:

A la entrada del cuantizador se conecta  $x[n]$ , de donde se obtiene que

$$x_q[n] = Q[x[n]] = x[n] + \eta_x[n] \quad (2.25)$$

donde  $\eta_x[n]$  corresponde al vector de entrada de error de cuantización.

Para el cuantizador conectado a la respuesta deseada  $d[n]$ , se tiene que:

$$d_q[n] = Q[d[n]] = d[n] + \eta_d[n] \quad (2.26)$$

donde  $\eta_d[n]$  es el error de cuantización de la respuesta deseada.

Para la cuantización del vector  $\mathbf{w}_q[n]$ ,

$$\mathbf{w}_q[n] = Q[\mathbf{w}[n]] = \mathbf{w}[n] + \Delta\mathbf{w}[n] \quad (2.27)$$

Teniendo en cuenta que  $\mathbf{w}[n]$  es el vector en precisión infinita y  $\Delta\mathbf{w}[n]$  es el vector del error resultante de la cuantización.

Para el cuantizador a la salida del filtro, se considera que:

$$y_q[n] = Q[\mathbf{w}^H[n-1]x[n]] = \mathbf{w}_q^H[n-1]x_q[n] + \eta_y[n] \quad (2.28)$$

donde  $\eta_y[n]$  es el error de cuantización a la salida del filtro.

La precisión finita en el Algoritmo RLS está descrita entonces por las siguientes ecuaciones:

$$\varepsilon_q[n] = d_q[n] - y_q[n] \quad (2.29)$$

y

$$\mathbf{w}_q[n] = \mathbf{w}_q[n-1] + Q[k[n] \varepsilon^*[n]] \quad (2.30)$$

### 2.3.4 Efectos de Precisión Finita para el Algoritmo AR $\gamma$

Para iniciar el análisis sobre los efectos de precisión finita del Algoritmo Acelerador Regresivo, versión  $\gamma$ , se ha tomado como referencia lo expuesto en los numerales 2.3.2 y 2.3.3 con respecto a los algoritmos LMS y RLS; con el fin definir el Algoritmo bajo los efectos de la cuantización, partiendo de lo siguiente:

La señal de entrada cuantizada será:

$$x_q[n] = Q[x[n]] = x[n] + \eta_x[n] \quad (2.31)$$

donde  $\eta_x[n]$  corresponde al vector de entrada de error de cuantización.

La respuesta deseada  $d[n]$ , una vez cuantizada será de la forma:

$$d_q[n] = Q[d[n]] = d[n] + \eta_d[n] \quad (2.32)$$

donde  $\eta_d[n]$  es el error de cuantización de la respuesta deseada.

Para la cuantización del vector  $\mathbf{w}_q[n]$ ,

$$\mathbf{w}_q[n] = Q[\mathbf{w}[n]] = \mathbf{w}[n] + \Delta\mathbf{w}[n] \quad (2.33)$$

Teniendo en cuenta que  $\mathbf{w}[n]$  es el vector en precisión infinita y  $\Delta\mathbf{w}[n]$  es el vector del error resultante de la cuantización.

Y que la salida del filtro es

$$y_q[n] = Q \left[ x_q^T[n] \mathbf{w}_q[n-1] \right] = x_q^T[n] \mathbf{w}_q[n-1] + \eta_y[n] \quad (2.34)$$

De esta forma, la precisión finita en el Algoritmo  $AR_\gamma$  está descrita por las ecuaciones 2.35 a 2.38:

$$e_q[n] = y_q[n] - d_q[n] \quad (2.35)$$

y

$$\mathbf{w}_q[n] = Q\{\mathbf{w}[n-1] + \alpha \mathbf{q}[n]\} \quad (2.36)$$

Equivalente a decir que:

$$\mathbf{w}_q[n] = Q\{\mathbf{w}[n-1]\} + Q\{\alpha \mathbf{q}[n]\} = \mathbf{w}_q[n-1] + \eta_{wq(n-1)} + \alpha Q\{\mathbf{q}[n]\}$$

Ya que  $\alpha$  se considera constante.

$$Q[\mathbf{q}[n]] = Q \left\{ \frac{\gamma}{\alpha + \gamma} \mathbf{q}[n-1] - \alpha g[n] m_1 x[n] \right\}$$

$$\mathbf{q}_q[n] = \frac{\gamma}{\alpha + \gamma} Q\{\mathbf{q}[n-1]\} - \alpha m_1 Q\{g[n]x[n]\} \quad (2.37)$$

considerando que  $\alpha$ ,  $\gamma$  y  $m_1$ , son constantes y que

$$Q\{g[n]x[n]\} = Q\{g[n]\}Q\{x[n]\} + \Delta_{gx}[n] = g_q[n]x_q[n]\Delta_{gx}[n], \text{ con}$$

$$g_q[n] = Q \left\{ \frac{e[n] + \gamma \mathbf{x}^T[n] \mathbf{q}[n-1]}{1 + \alpha \gamma \mathbf{x}^T[n] m_1 \mathbf{x}[n]} \right\} \quad (2.38)$$

$$g_q[n] = Q\{e[n] + \gamma \mathbf{x}^T[n] \mathbf{q}[n-1]\} Q\{1 + \alpha \gamma \mathbf{x}^T[n] m_1 \mathbf{x}[n]\}^{-1}$$

$$g_q[n] = \{e_q[n] + \gamma Q\{\mathbf{x}^T[n] \mathbf{q}[n-1]\}\} \{1 + Q[\alpha \gamma \mathbf{x}^T[n] m_1 \mathbf{x}[n]]\}^{-1}$$

$$g_q[n] = e_q[n] + \gamma Q\{\mathbf{x}^T[n] \mathbf{q}[n-1]\} \{1 + \alpha \gamma m_1 Q[\mathbf{x}^T[n] \mathbf{x}[n]]\}^{-1}$$

Como se ha podido observar a lo largo del análisis realizado, cada operación de cuantización introduce un valor de ruido de cuantización que es acumulativo y que en el caso de las ecuaciones para el  $AR_\gamma$  genera desviaciones que deben ser tratadas con un detalle matemático profundo que está por fuera del alcance de este proyecto, con el fin de determinar la naturaleza del ruido. De manera experimental, a partir de la implementación mediante simulación del algoritmo que se realiza en los capítulos posteriores, se analizará también el ruido de cuantización.



## Capítulo 3

# Implementación del Algoritmo Acelerador Regresivo versión $\gamma$

Una vez realizado el estudio sobre los algoritmos adaptativos considerando de manera particular el Algoritmo Acelerador Regresivo versión  $\gamma$ , y los efectos de la utilización de aritmética de punto fijo de manera general; se procedió a desarrollar la implementación de los algoritmos LMS, RLS y AR $\gamma$  mediante el uso de simulación, con el objetivo de identificar las características del comportamiento del Algoritmo en estudio bajo los efectos de precisión finita.

En el presente capítulo inicialmente se define el caso de estudio para el análisis de los efectos de aritmética de punto fijo, luego se elige la herramienta de simulación, para finalmente realizar la implementación. Estos aspectos se exponen en las secciones siguientes.

### 3.1 Definición del Caso de Estudio

En la resolución de problemas en ingeniería, es indispensable contar con una estrategia consistente para resolver un problema que ya ha sido claramente definido.

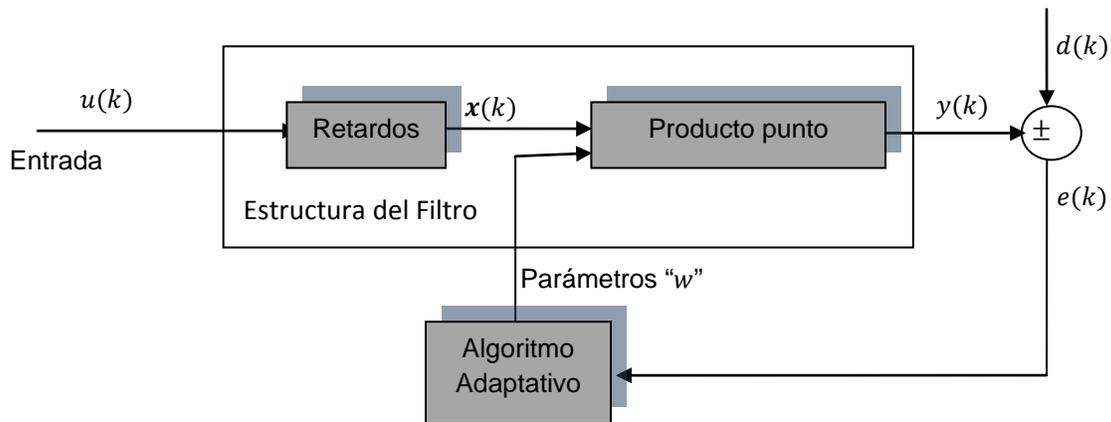
En este orden de ideas, a partir del objetivo general de la tesis que es “*Analizar el efecto de la aritmética de punto fijo en la convergencia del Algoritmo Acelerador Regresivo versión  $\gamma$* ”, se procedió a definir como caso de estudio un sistema de identificación como el que se indica en la Figura 2.3 a) y se explica en la Sección 2.1.3., con el fin de analizar la información tanto a nivel de entrada como de salida y realizar así una estimación de los parámetros.

Adicionalmente, el sistema de identificación definido como caso de estudio será analizado utilizando estructuras tanto FIR como IIR, debido a que su escogencia en un proceso de filtrado tiene un profundo efecto en la operación de los algoritmos.

En los Numerales 3.1.1 y 3.1.2 se presentan los diagramas del caso de estudio definido utilizando filtros FIR e IIR, respectivamente.

### 3.1.1 Caso de estudio 1: Sistema de Identificación usando un Filtro de Respuesta Finita al Impulso (FIR)

En la Figura 3.1 se muestra el sistema de identificación que utiliza un filtro tipo FIR. Una de las principales características de este filtro además de su respuesta finita al impulso, es que es inherentemente estable; razón por lo que es muy común que se encuentre en la representación básica cuando de diseño de filtros adaptativos se trata (Haykin, 2002).



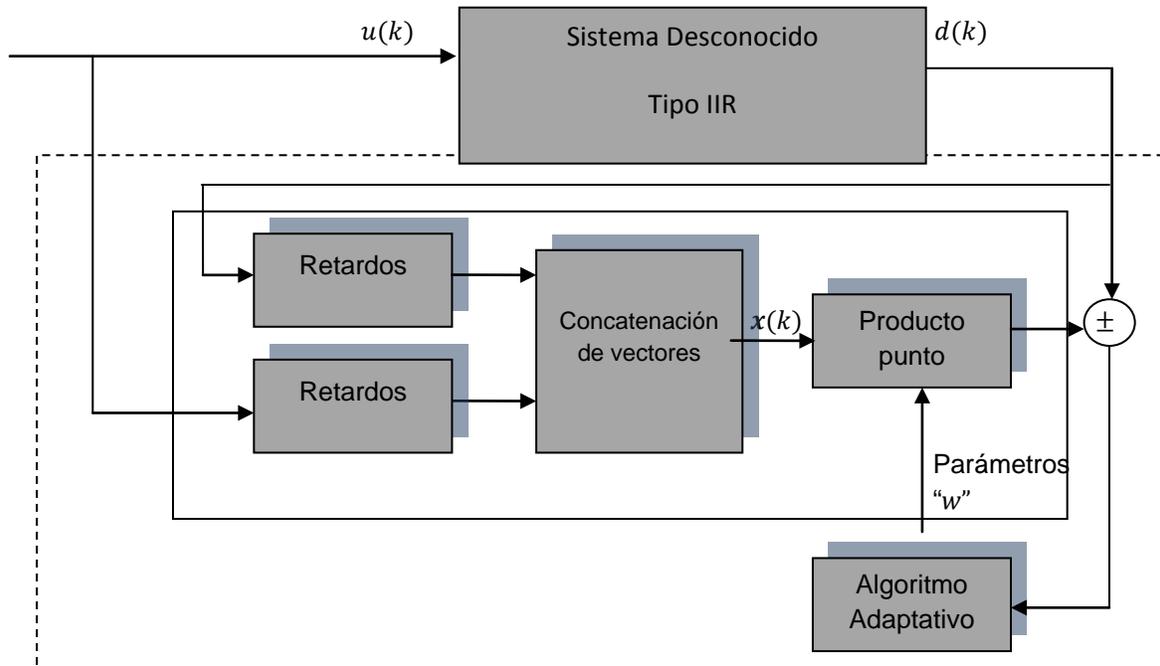
**Figura 3.1 Sistema de Identificación usando un Filtro FIR**

El vector  $x(k)$  es un vector columna que solamente depende de  $u(k)$  y  $w$  corresponde a los parámetros del sistema desconocido.

$$x[k] = \begin{bmatrix} u[k] \\ u[k-1] \\ \vdots \\ u[k-hb] \end{bmatrix}$$

### 3.1.2 Caso de estudio 2: Sistema de Identificación usando un Filtro de Respuesta Infinita al Impulso (IIR)

Considérese un filtro tipo IIR (ver Figura 3.2) en donde se observa la realimentación, la cual genera una respuesta al impulso infinita. El principal inconveniente de la realimentación es que puede introducir problemas de inestabilidad, haciendo más interesante el análisis.



**Figura 3.2 Sistema de Identificación usando un Filtro IIR**

El vector que contiene los parámetros estimados del sistema desconocido es  $w$ .

La definición del vector columna regresor  $x(k)$  está dada por

$$x(k) = \begin{bmatrix} d(k-1) \\ d(k-2) \\ \vdots \\ \vdots \\ d(k-ha) \\ u(k-1) \\ \vdots \\ \vdots \\ u(k-hb) \end{bmatrix}$$

Es importante recalcar que es la definición del vector  $x(k)$  la que permite estimar los parámetros de un sistema desconocido, bien sea de un modelo FIR o IIR. Así, el procedimiento propuesto no consiste en aproximar el IIR como un FIR con un mayor número de coeficientes.

Para la definición del vector de regresión se tuvo en cuenta la formulación que se realiza en el área de identificación de sistemas, en la cual un mismo algoritmo se utiliza para estimar parámetros de un sistema FIR o IIR, variando únicamente el vector de regresión  $x(k)$  (Lennatt Ljung, 1999).

### 3.2 Selección de la Herramienta de Simulación

En los trabajos realizados hasta el momento con relación al Algoritmo Acelerador Regresivo donde los estudios han evolucionado a partir de la misma propuesta del algoritmo (Pait, 1998), su discretización (Gerken et al., 2000) y la reducción de la complejidad numérica (Jojoa, 2003); la herramienta de simulación utilizada para verificar el análisis de estabilidad ha sido Matlab, permitiendo obtener resultados de relevancia académica e investigativa.

Inicialmente con el fin de entender el funcionamiento de Algoritmo Acelerador Regresivo y corroborar los resultados obtenidos en trabajos previos, se procedió a realizar la implementación de un sistema de identificación que utilizó los Algoritmos LMS y AR $\gamma$  bajo las mismas condiciones utilizadas en Jojoa (1999). Esta simulación se realizó en Matlab (Anexo 2) utilizando punto flotante.

Con el objetivo de entender de mejor manera el algoritmo y lograr visualizar claramente la representación gráfica de los algoritmos antes implementados en Matlab, se exploró Simulink que es un editor gráfico con las siguientes características y ventajas (<http://www.mathworks.com>):

- Brinda facilidades para crear y gestionar diagramas de bloques jerárquicos, permitiendo apreciar los sistemas implementados en un alto grado de abstracción.
- Cuenta con bibliotecas de bloques predefinidos para el modelado de sistemas de tiempo continuo y tiempo discreto.
- Permite la visualización de los datos de resultados de la simulación.
- Provee herramientas de gestión de datos y archivos
- Permite realizar el análisis para refinar la arquitectura del modelo y aumentar la velocidad de simulación.
- Brinda funciones para importar algoritmos desarrollados en Matlab.
- Cuenta con un soporte para construcción de prototipos que permite la traducción automática para un rápido desarrollo en hardware.

Analizadas las prestaciones de Simulink y sobretodo la ventaja de la representación jerárquica mediante bloques que permite un mejor entendimiento del funcionamiento de los algoritmos a través de la representación gráfica, se decidió utilizarlo como la herramienta de simulación para el desarrollo de la tesis.

---

### 3.3 Implementación del AR $\gamma$ en Simulink

A partir de los casos de estudio definidos en 3.1 se realizó la implementación en Simulink, donde se observa que para el análisis de punto fijo se ejecuta mediante una sentencia de decisión como se detalla en el Anexo 3.

Con el fin de establecer referentes y comparaciones sobre el comportamiento del algoritmo AR $\gamma$  en punto fijo, las implementaciones del sistema de identificación tanto en configuración FIR como IIR también incluyeron los algoritmos LMS y RLS

#### 3.3.1 Caso de estudio 1: Sistema de Identificación usando un Filtro de Respuesta Finita al Impulso (FIR)

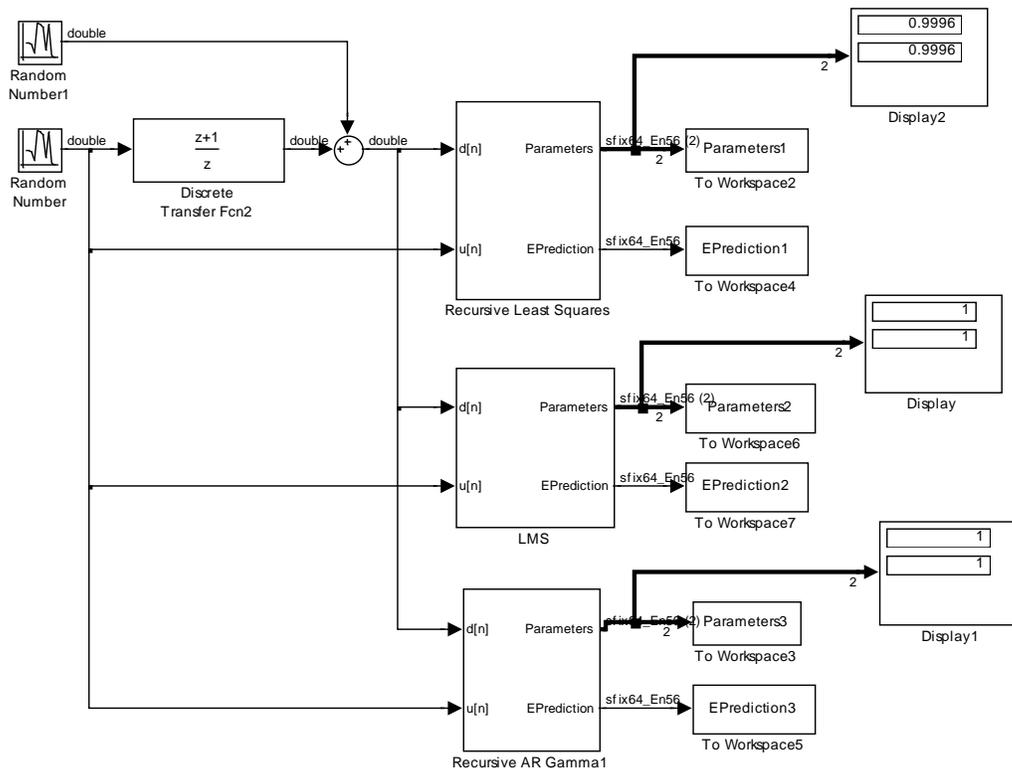
El objetivo en un sistema de identificación es estimar los parámetros del sistema a identificar. Se ha tomado como función de transferencia

$$H(z) = \frac{z+1}{z} \quad (3.1)$$

Correspondiente a la función de un filtro FIR.

Adicionalmente se introduce un ruido blanco gaussiano con media cero, generando de esta forma una señal contaminada que corresponde a la entrada del proceso adaptativo y el efecto del filtraje es estimar los parámetros mediante el uso de los algoritmos LMS, RLS y el AR $\gamma$ , como se muestra en la Figura 3.3.

---



**Figura 3.3 Sistema de Identificación que compara el AR $\gamma$  con los algoritmos LMS y RLS, usando filtros FIR**

En las Figuras 3.4, 3.5 y 3.6 se muestra la implementación de los algoritmos RLS, LMS y AR $\gamma$  en bloques de Simulink y en la Figura 3.7 se representa el vector de regresión. En el Anexo 3 se detallan los diagramas en Simulink para la implementación de cada ecuación de los algoritmos.

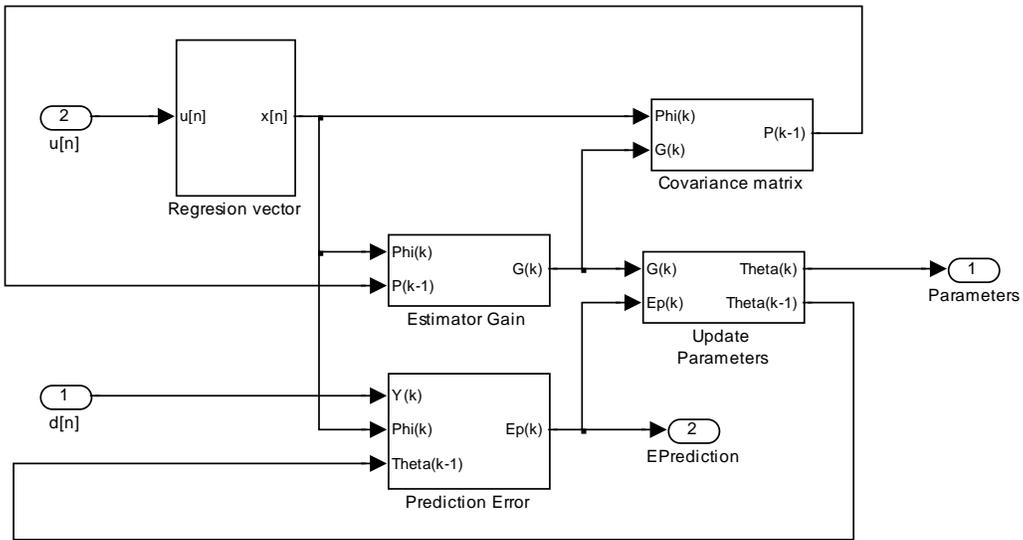


Figura 3.4 Implementación del RLS

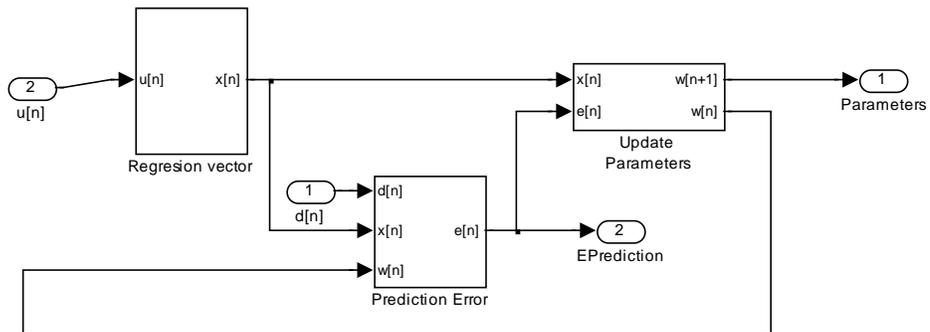


Figura 3.5 Implementación del LMS

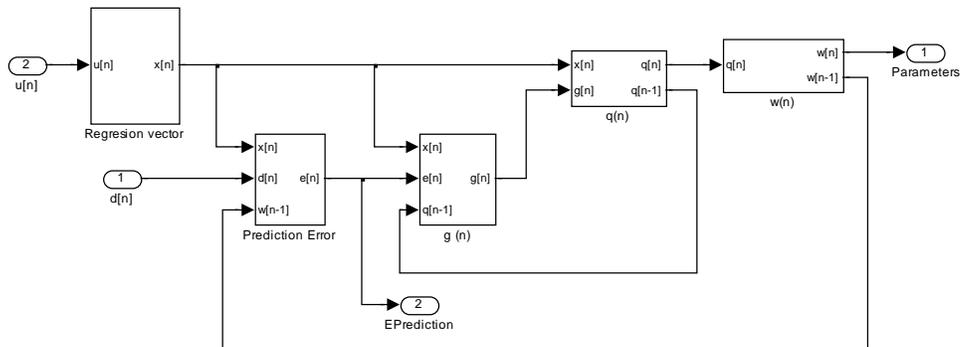


Figura 3.6 Implementación del AR $\gamma$

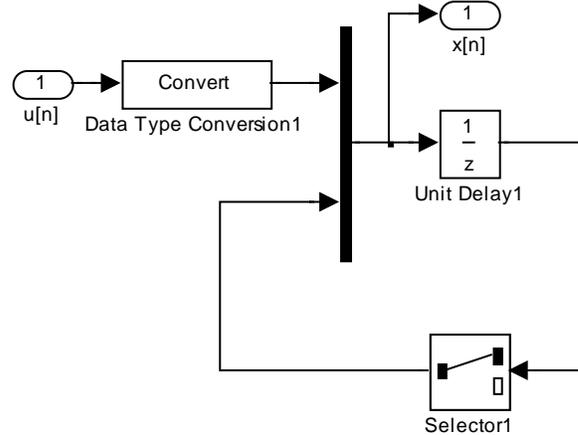


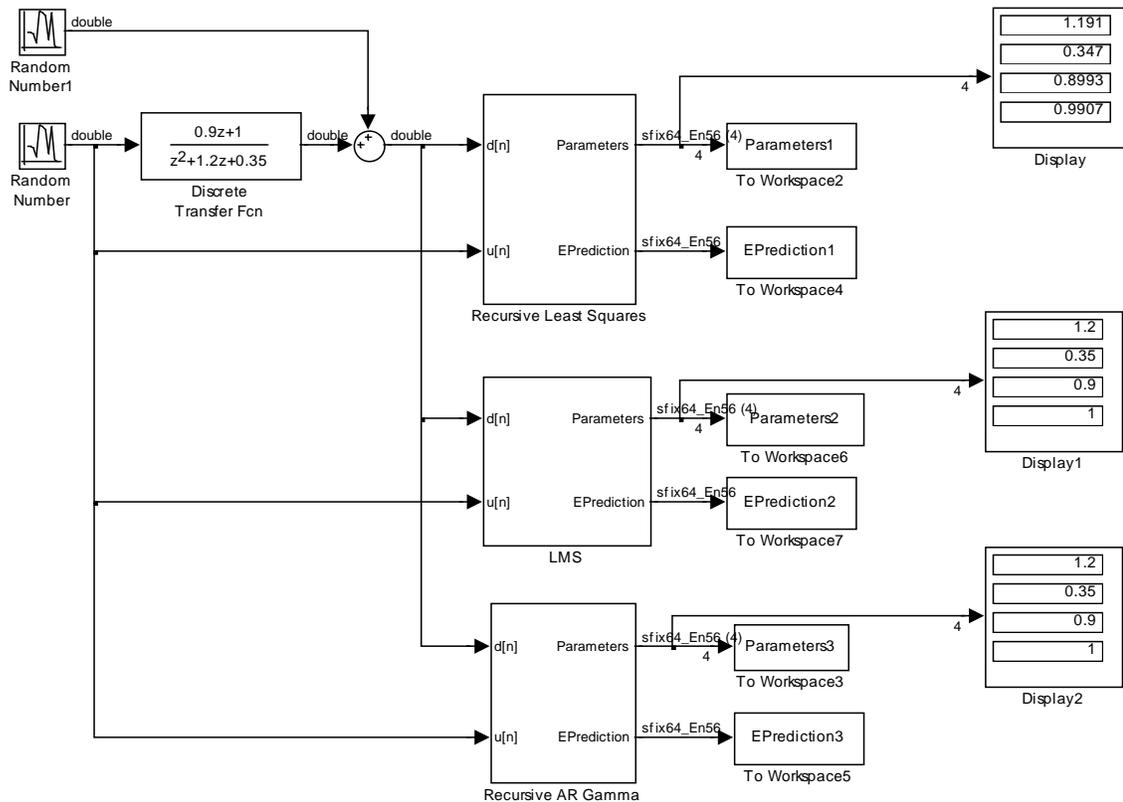
Figura 3.7 Vector de Regresión usado en el Caso de Estudio 1

### 3.3.2 Caso de estudio 2: Sistema de Identificación usando un Filtro de Respuesta Infinita al Impulso (IIR)

En la Figura 3.8 se muestra el caso de estudio donde se ha tomado el sistema descrito por la función de transferencia  $H(z)$  dada por

$$H(z) = \frac{0.9z+1}{z^2+1.2z+0.35} \quad (3.2)$$

función que en realidad tiene unos parámetros desconocidos y que debe cumplir con la condición de tener todos los polos al interior del círculo unitario.

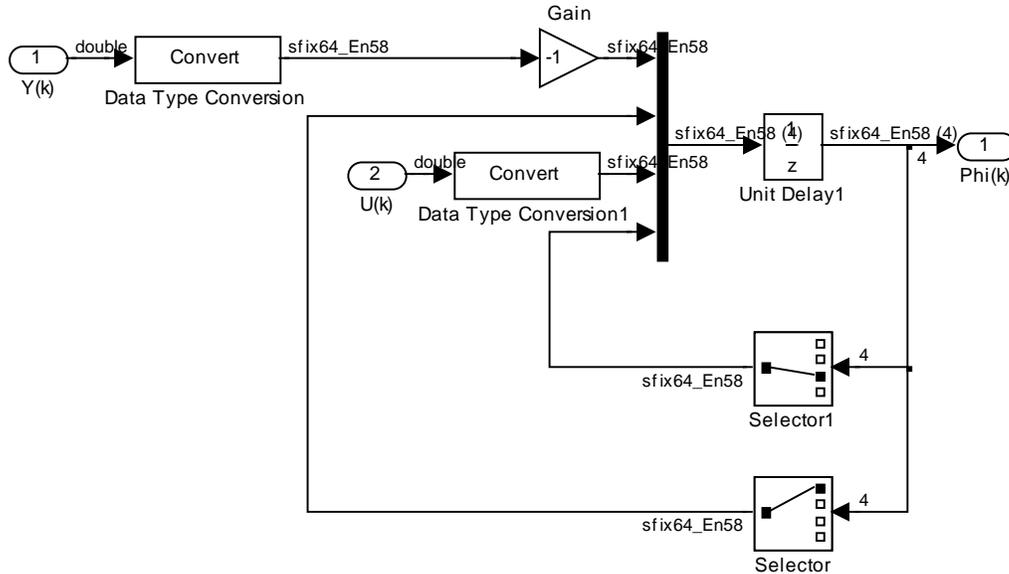


**Figura 3.8 Sistema de Identificación que compara el AR $\gamma$  con los algoritmos LMS y RLS, usando filtros IIR.**

Además se ha elegido un modelo de ruido blanco gaussiano con media cero. Así, la salida del sistema a identificar se suma con el ruido blanco, y esta señal contaminada junto con la entrada inicial del proceso se somete a un filtrado adaptativo que tiene como finalidad la estimación de los parámetros del sistema desconocido. Para efectos de comparación, el proceso de filtrado se ha realizado utilizando los algoritmos RLS, LMS y el algoritmo AR $\gamma$ .

En el campo llamado “Parameters”, indicado en la Figura 3.8 se guarda la evolución de los parámetros y también se calcula el error de predicción (EPrediction).

En la Figura 3.9 se muestra el detalle de la implementación del vector de regresión que se encarga de calcular las nuevas salidas a partir de los valores anteriores.



**Figura 3.9 Vector de Regresión usado en el Caso de Estudio 2.**

Para la implementación de los demás bloques ver el Anexo 3.

Después de la implementación de los casos de estudio, se mencionan como aportes o aspectos diferenciadores con relación a los trabajos realizados anteriormente con respecto al Algoritmo  $AR\gamma$  (Jojoa, 2003), (Realpe, 2009) los siguientes aportes:

- La implementación en Simulink permitió apropiar el conocimiento sobre los algoritmos adaptativos implementados gracias a la claridad de la representación gráfica de los mismos.
- Al comparar el algoritmo  $AR\gamma$  con el RLS y LMS, se cuenta con un nuevo referente ya que en Jojoa (2003) se utilizaron únicamente el LMS y el NMLS.
- A partir de la variación del vector de regresión  $x(k)$  se utilizaron los algoritmos para estimar parámetros de sistemas FIR e IIR, avanzando de esta manera en el estudio sobre el comportamiento del algoritmo.
- Dado que el ruido representa la imposibilidad de medir exactamente una señal, muchos algoritmos de estimación de parámetros funcionan adecuadamente sin su presencia. Al implementar el sistema con con ruido se buscó una mayor exigencia al algoritmo y se pudo determinar su comportamiento bajo condiciones más adversas.

A partir de las implementaciones en Simulink descritas, se realizaron diferentes experiencias mostradas en el Capítulo 4 de este documento, tendientes a analizar

el efecto de la aritmética de punto fijo en la convergencia del Algoritmo Acelerador Regresivo AR  $\gamma$ .

---



## Capítulo 4

### Análisis y Comparación de Resultados

A partir de la implementación del Algoritmo Acelerador Regresivo versión  $\gamma$  mediante simulación en el capítulo anterior, se presentan a continuación las diferentes experiencias realizadas para los casos de estudio definidos en la Sección 3.1, con el fin de determinar los aspectos de interés en el comportamiento del algoritmo bajo los efectos de precisión finita.

Los estudios realizados en el presente capítulo, fueron definidos teniendo en cuenta lo propuesto en los objetivos específicos de este trabajo de tesis con el fin de poder analizar del impacto del uso de valores finitos sobre la varianza del error de predicción, los parámetros de ajuste ( $m_1$ ,  $\alpha$  y  $\gamma$ ) y la velocidad de convergencia del  $AR_\gamma$ .

#### 4.1 Experiencia 1. Determinación del mínimo número de bits requerido para el funcionamiento del $AR_\gamma$

El objetivo de este experimento es determinar el número mínimo de bits que se requieren para que el algoritmo  $AR_\gamma$  sea estable.

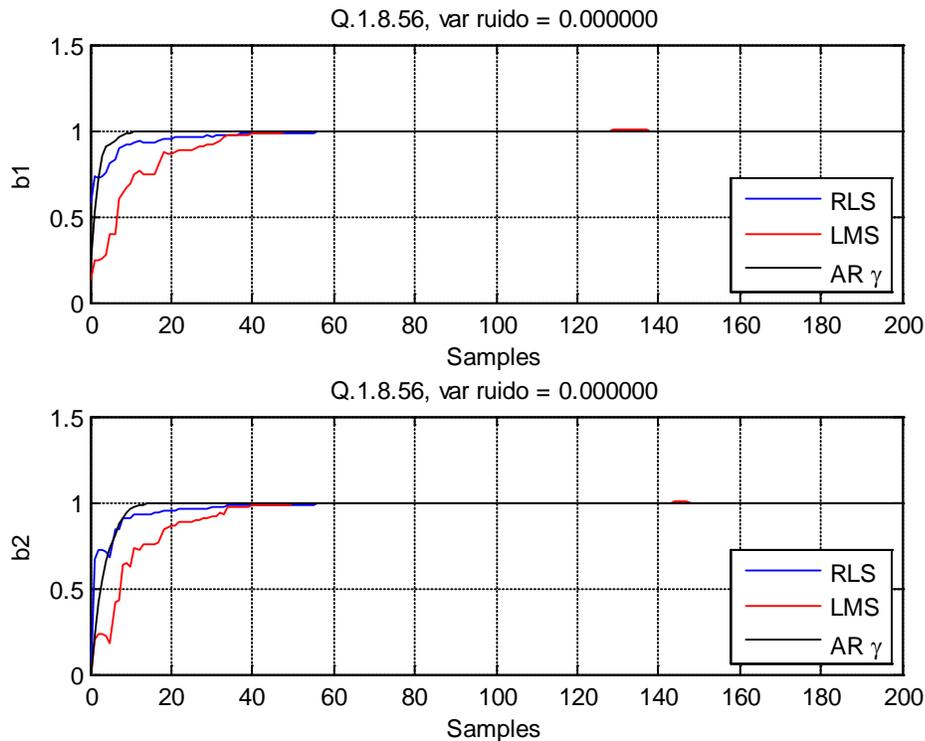
El procedimiento consiste en emplear diferentes cantidades de bits para la parte fraccionaria con una misma longitud de palabra. Se inicia con una longitud grande y una cantidad suficiente de bits en la parte entera con el fin de variar hasta llegar a un valor mínimo; garantizando así que no exista desbordamiento. En caso de que se presenten inconvenientes, éstos serán debidos a precisión.

Para indicar el número de bits de la parte fraccional, se especifica con la notación Q.signo.entera.fraccional. Así, si se coloca Q1.4.60 indica que los números son con signo y 60 bits para la parte fraccional y 4 bits para la parte entera, en total, la suma de estos números dice el número total de bits (64 bits para el ejemplo en mención).

Para la realización del experimento 1 se utiliza la simulación del Anexo 3 que implementa los sistemas de identificación usando filtro FIR e IIR, variando la longitud de la palabra, específicamente la parte entera. A continuación se presentan los resultados de las simulaciones realizadas, las gráficas que muestran la región de convergencia de los coeficientes y las tablas que resumen los resultados de la media y la varianza del error de predicción para los casos de estudio definidos en la Sección 3.1.

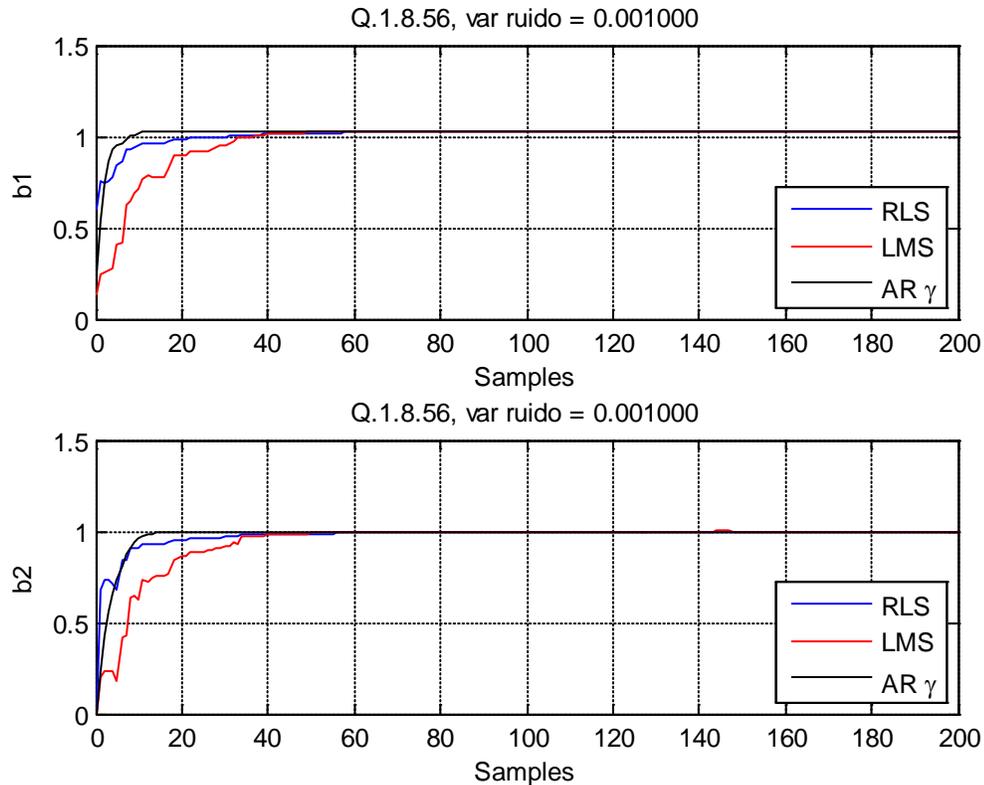
#### 4.1.1 Caso de estudio 1: Sistema de Identificación usando un Filtro de Respuesta Finita al Impulso (FIR)

- a) Tamaño de palabra: Q.1.8.56  
 Número de muestras: 200 muestras  
 Número de coeficientes: 2  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{z+1}{z}$   
 Varianza del ruido blanco de medición = 0



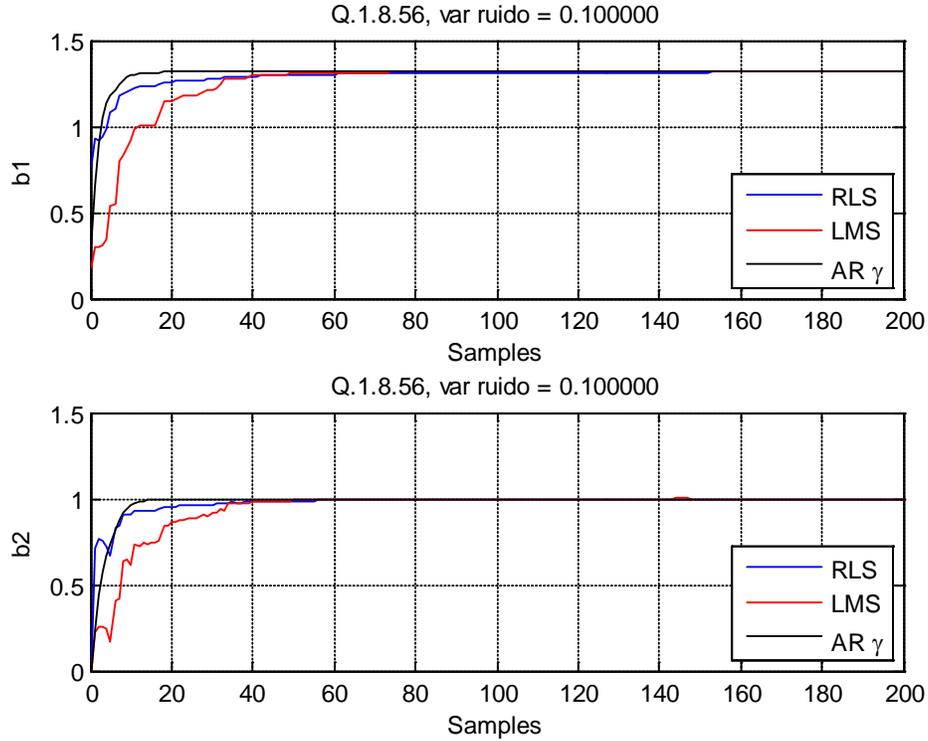
**Figura 4.1 Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$  con una palabra de tamaño Q.1.8.56 y sin ruido, usando un filtro FIR**

- b) Tamaño de palabra: Q.1.8.56  
 Número de muestras: 200 muestras  
 Número de coeficientes: 2  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{z+1}{z}$   
 Varianza del ruido blanco de medición = 0.001



**Figura 4.2 Región de convergencia de los coeficientes para los algoritmos LMS, RLS y  $AR \gamma$  con una palabra de tamaño Q.1.8.56 y con ruido 0.001, usando un filtro FIR**

- c) Tamaño de palabra: Q.1.8.56  
 Número de muestras: 200 muestras  
 Número de coeficientes: 2  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{z+1}{z}$   
 Varianza del ruido blanco de medición = 0.1



**Figura 4.3 Región de convergencia de los coeficientes para los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.8.56 y con ruido 0.1, usando un filtro FIR**

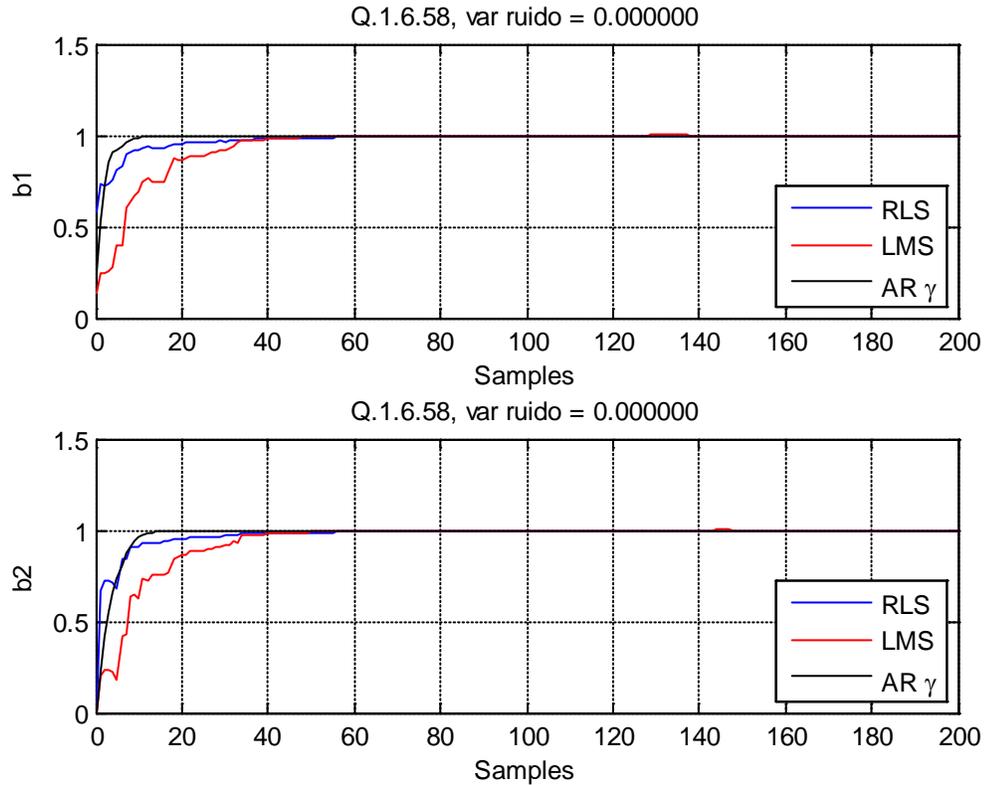
En la Tabla 4.1 se indican los valores obtenidos para la media y la varianza del error de predicción para los algoritmos estudiados y para diferentes valores de ruido de medición utilizando una longitud de palabra Q.1.8.56.

Q.1.8.56			
Media del Error de Predicción			
Ruido de Medida	LMS	RLS	$AR\gamma$
0	$2.785 \times 10^{-2}$	$1.412 \times 10^{-2}$	$1.552 \times 10^{-2}$
0.001	$2.820 \times 10^{-2}$	$1.425 \times 10^{-2}$	$1.569 \times 10^{-2}$
0.1	$3.139 \times 10^{-2}$	$1.543 \times 10^{-2}$	$1.723 \times 10^{-2}$
Varianza del Error de Predicción			
Ruido de Medida	LMS	RLS	$AR\gamma$
0	$4.930 \times 10^{-2}$	$1.380 \times 10^{-2}$	$1.625 \times 10^{-2}$
0.001	$5.067 \times 10^{-2}$	$1.403 \times 10^{-2}$	$1.650 \times 10^{-2}$
0.1	$6.491 \times 10^{-2}$	$1.635 \times 10^{-2}$	$1.900 \times 10^{-2}$

**Tabla 4.1 Media y varianza del error de predicción de los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.8.56 para diferentes valores de ruido de medida, usando un filtro FIR**

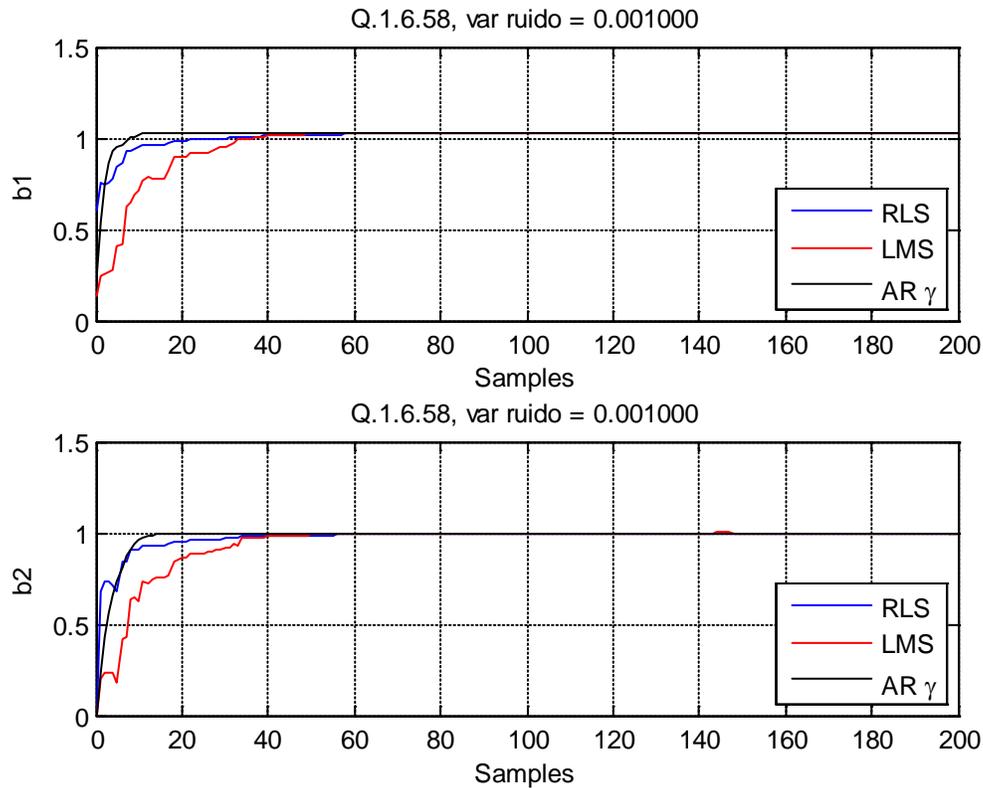
Se observa que el  $AR\gamma$  converge de manera rápida y precisa para longitudes de palabra en las que la parte entera tiene 8 bits. Los incrementos en el ruido de medición, generan inconvenientes en cuanto a la precisión del algoritmo.

- d) Tamaño de palabra: Q.1.6.58  
 Número de muestras: 200 muestras  
 Número de coeficientes: 2  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{z+1}{z}$   
 Varianza del ruido blanco de medición = 0



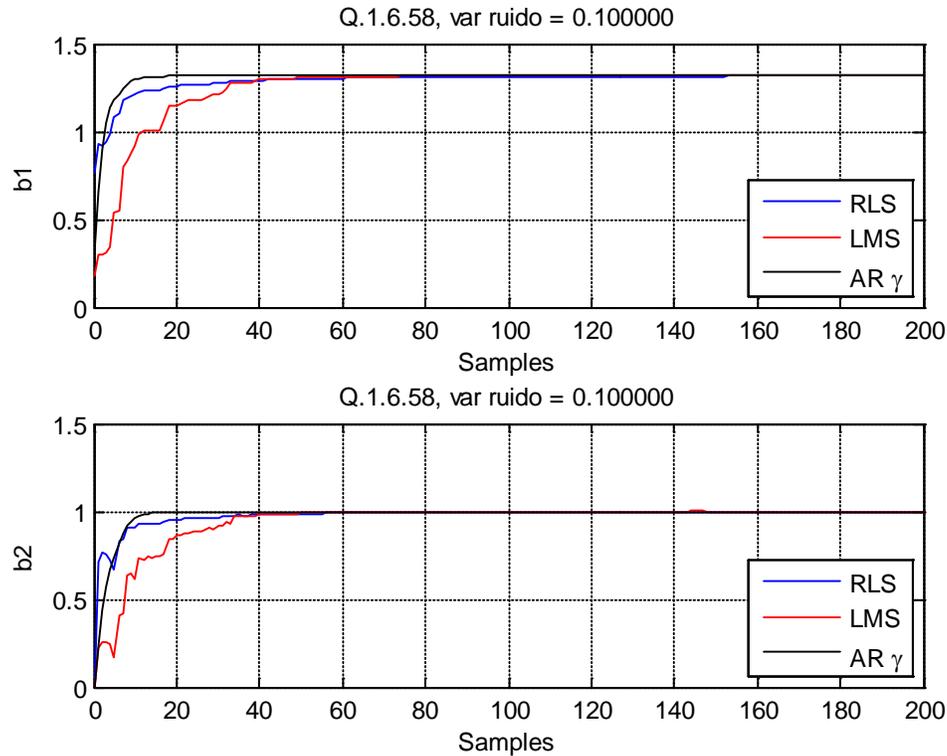
**Figura 4.4** Región de convergencia de los coeficientes para los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.6.58 y sin ruido, usando un filtro FIR

- e) Tamaño de palabra: Q.1.6.58  
 Número de muestras: 200 muestras  
 Número de coeficientes: 2  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{z+1}{z}$   
 Varianza del ruido blanco de medición = 0.001



**Figura 4.5** Región de convergencia de los coeficientes para los algoritmos LMS, RLS y  $AR_\gamma$  con una palabra de tamaño Q.1.6.58 y con ruido 0.001, usando un filtro FIR

- f) Tamaño de palabra: Q.1.6.58  
 Número de muestras: 200 muestras  
 Número de coeficientes: 2  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{z+1}{z}$   
 Varianza del ruido blanco de medición = 0.1



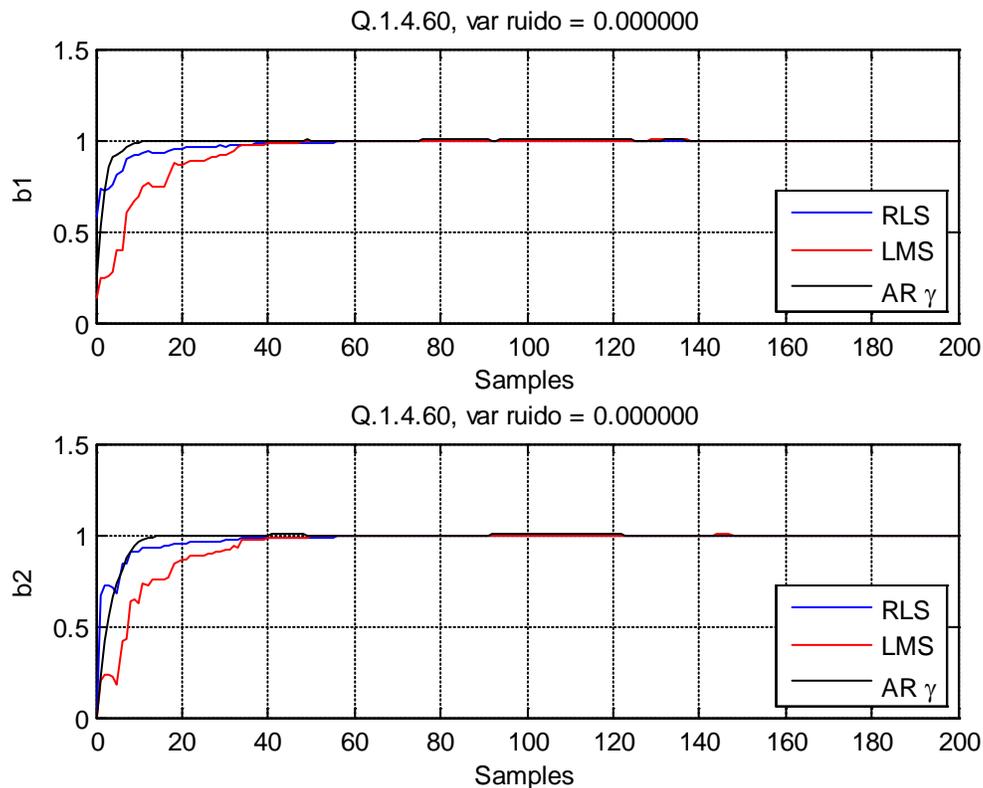
**Figura 4.6** Región de convergencia de los coeficientes para los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.6.58 y con ruido 0.1, usando un filtro FIR

En la Tabla 4.2 se indican los valores obtenidos para la media y la varianza del error de predicción para los algoritmos estudiados y para diferentes valores de ruido de medición y una longitud de palabra Q.1.6.58.

Q.1.6.58			
Media del Error de Predicción			
Ruido de Medida	LMS	RLS	$AR\gamma$
0	$2.785 \times 10^{-2}$	$1.412 \times 10^{-2}$	$1.552 \times 10^{-2}$
0.001	$2.820 \times 10^{-2}$	$1.425 \times 10^{-2}$	$1.569 \times 10^{-2}$
0.1	$3.139 \times 10^{-2}$	$1.543 \times 10^{-2}$	$1.723 \times 10^{-2}$
Varianza del Error de Predicción			
Ruido de Medida	LMS	RLS	$AR\gamma$
0	$4.930 \times 10^{-2}$	$1.380 \times 10^{-2}$	$1.625 \times 10^{-2}$
0.001	$5.067 \times 10^{-2}$	$1.403 \times 10^{-2}$	$1.650 \times 10^{-2}$
0.1	$6.491 \times 10^{-2}$	$1.635 \times 10^{-2}$	$1.900 \times 10^{-2}$

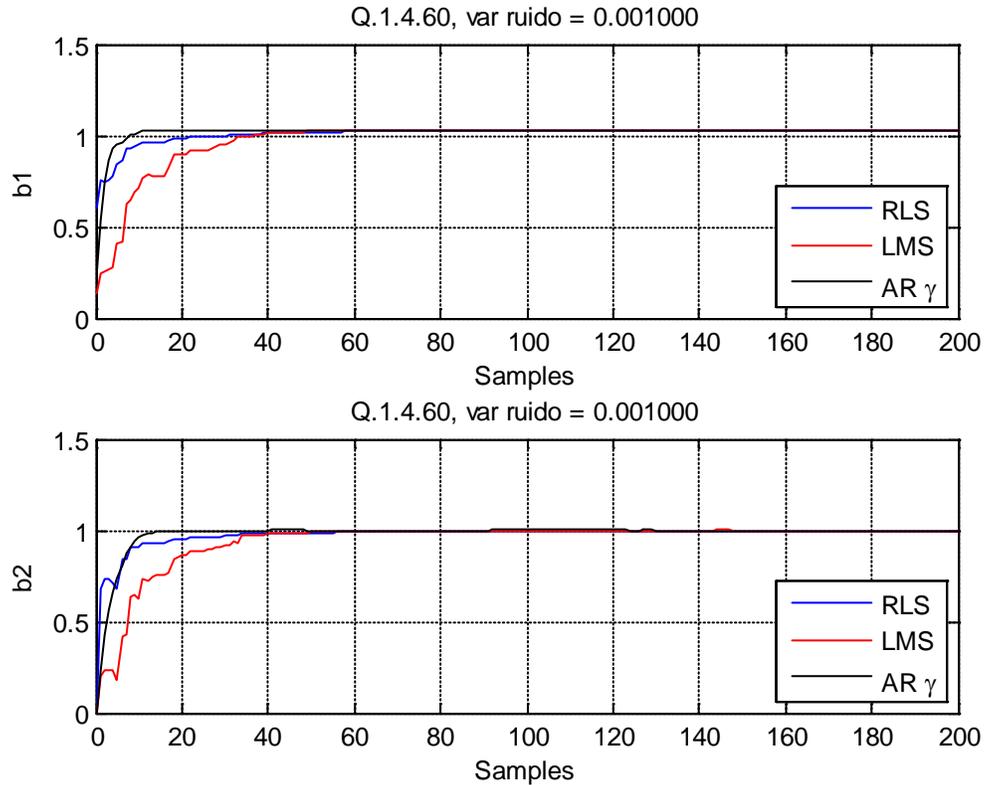
**Tabla 4.2** Media y varianza del error de predicción de los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.6.58 para diferentes valores de ruido de medida, usando un filtro FIR

- g) Tamaño de palabra: Q.1.4.60  
 Número de muestras: 200 muestras  
 Número de coeficientes: 2  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{z+1}{z}$   
 Varianza del ruido blanco de medición = 0



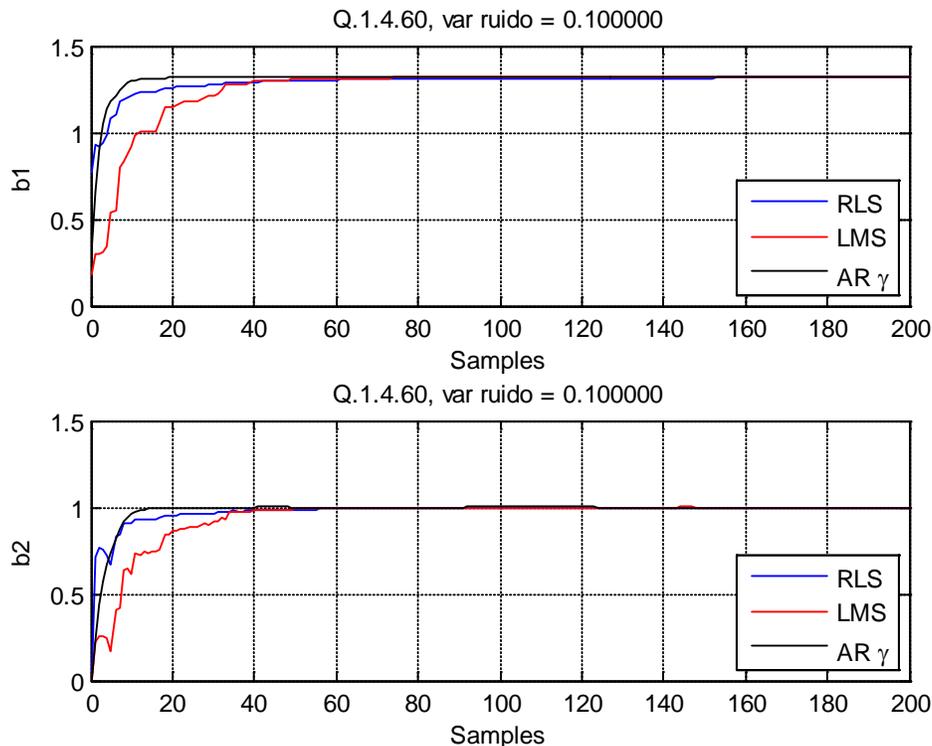
**Figura 4.7** Región de convergencia de los coeficientes para los algoritmos LMS, RLS y  $AR \gamma$  con una palabra de tamaño Q.1.4.60 y sin ruido, usando un filtro FIR

- h) Tamaño de palabra: Q.1.4.60  
 Número de muestras: 200 muestras  
 Número de coeficientes: 2  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{z+1}{z}$   
 Varianza del ruido blanco de medición = 0.001



**Figura 4.8 Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$  con una palabra de tamaño Q.1.4.60 y con ruido 0.001, usando un filtro FIR**

- i) Tamaño de palabra: Q.1.4.60  
 Número de muestras: 200 muestras  
 Número de coeficientes: 2  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{z+1}{z}$   
 Varianza del ruido blanco de medición = 0.1



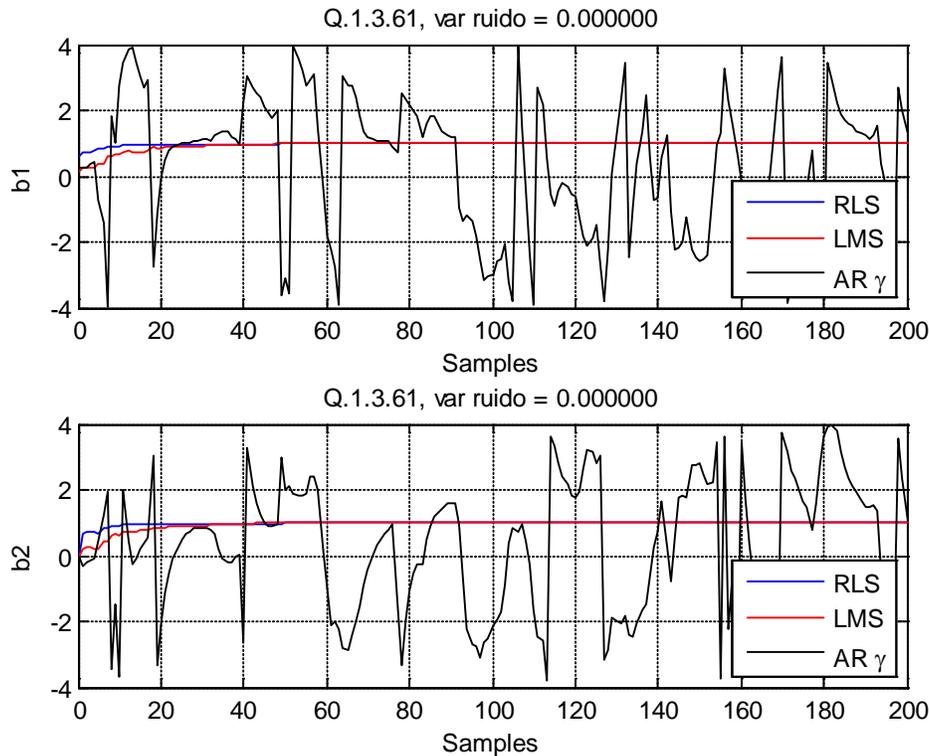
**Figura 4.9 Región de convergencia de los coeficientes para los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.4.60y con ruido 0.1, usando un filtro FIR**

En la Tabla 4.3 se indican los valores obtenidos para la media y la varianza del error de predicción de los algoritmos estudiados y para diferentes valores de ruido de medición y una longitud de palabra Q.1.4.60.

Q.1.4.60			
Media del Error de Predicción			
Ruido de Medida	LMS	RLS	$AR\gamma$
0	$2.785 \times 10^{-2}$	$1.412 \times 10^{-2}$	$1.552 \times 10^{-2}$
0.001	$2.820 \times 10^{-2}$	$1.425 \times 10^{-2}$	$1.569 \times 10^{-2}$
0.1	$3.139 \times 10^{-2}$	$1.543 \times 10^{-2}$	$1.723 \times 10^{-2}$
Varianza del Error de Predicción			
Ruido de Medida	LMS	RLS	$AR\gamma$
0	$4.930 \times 10^{-2}$	$1.380 \times 10^{-2}$	$1.625 \times 10^{-2}$
0.001	$5.067 \times 10^{-2}$	$1.403 \times 10^{-2}$	$1.650 \times 10^{-2}$
0.1	$6.491 \times 10^{-2}$	$1.635 \times 10^{-2}$	$1.900 \times 10^{-2}$

**Tabla 4.3 Media y varianza del error de predicción de los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.4.60 para diferentes valores de ruido de medida, usando un filtro FIR**

- j) Tamaño de palabra: Q.1.3.61  
 Número de muestras: 200 muestras  
 Número de coeficientes: 2  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{z+1}{z}$   
 Varianza del ruido blanco de medición = 0



**Figura 4.10** Región de convergencia de los coeficientes para los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.3.61 y sin ruido, usando un filtro FIR

En la Tabla 4.4 se muestran los valores obtenidos para la media y la varianza del error de predicción para los algoritmos estudiados con una longitud de palabra Q.1.3.61.

<b>Q.1.3.61</b>			
<b>Media del Error de Predicción</b>			
Ruido de Medida	<b>LMS</b>	<b>RLS</b>	<b>AR<math>\gamma</math></b>
0	$2.785 \times 10^{-2}$	$1.412 \times 10^{-2}$	$3.684 \times 10^{-2}$
<b>Q.1.3.61</b>			
<b>Varianza del Error de Predicción</b>			
Ruido de Medida	<b>LMS</b>	<b>RLS</b>	<b>AR<math>\gamma</math></b>
0	$4.930 \times 10^{-2}$	$1.380 \times 10^{-2}$	$3.413 \times 10^0$

**Tabla 4.4 Media y varianza del error de predicción de los algoritmos LMS, RLS y AR $\gamma$  con una palabra de tamaño Q.1.4.60 para diferentes sin ruido de medida, usando un filtro FIR**

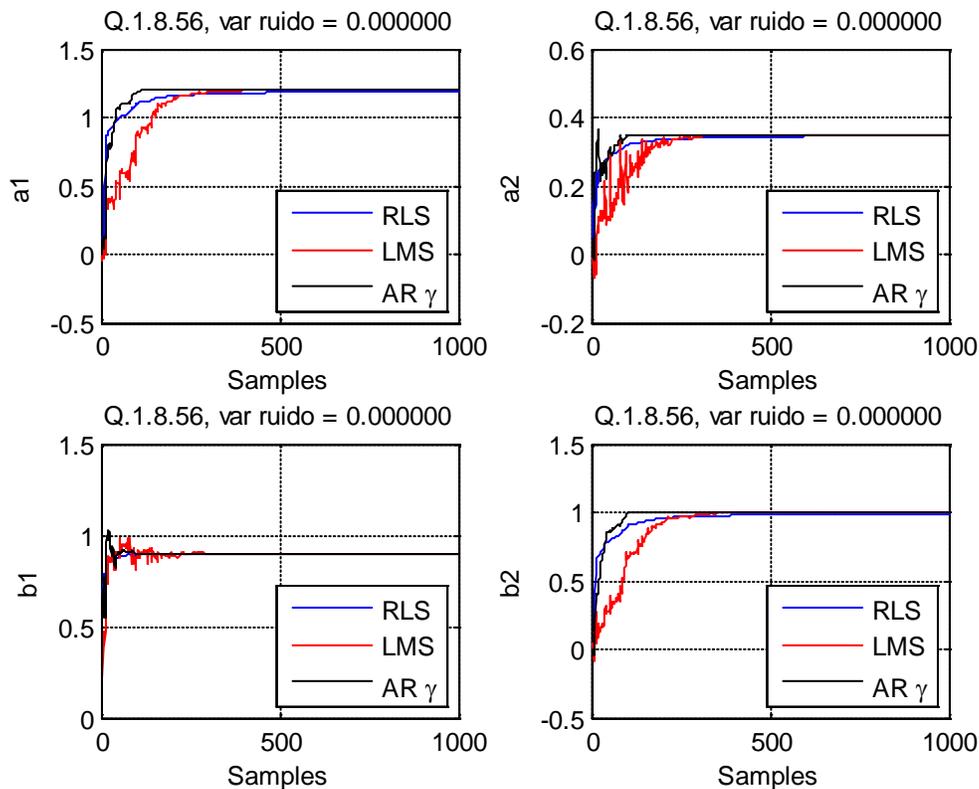
Del resultado obtenido en esta experiencia al utilizar 3 bits en la parte entera, se encontró que el AR $\gamma$  no logra la convergencia.

A partir de las experiencias antes realizadas se pueden sacar las siguientes conclusiones:

- En la proporción que el ruido de medición se aumenta, los algoritmos se hacen más sensibles, situación reflejada en la precisión como se puede observar en la Figuras 4.3, 4.6 y 4.9. De la misma forma se afecta la varianza del error de predicción como lo indican las tablas 4.1, 4.2 y 4.3, aumentando su valor de acuerdo a los incrementos en el ruido de medida.
- El Algoritmo AR $\gamma$  presenta una velocidad de convergencia mayor que el LMS.
- En la medida en que se limita la cantidad de bits en la parte entera la varianza del AR $\gamma$  se mantiene constante hasta que se realiza la prueba con 3 bits, momento en el que la varianza aumenta significativamente razón por la cual es válido afirmar que se requieren mínimo 4 bits para la parte entera para que el AR $\gamma$  sea estable.
- Al determinar un número mínimo de bits en la parte entera para que el algoritmo se comporte de manera adecuada, utilizando la configuración FIR, se garantiza que si hay problema de divergencia del AR $\gamma$ , se debe a insuficiencia en el número de bits utilizados para representar la parte entera.

#### 4.1.2 Caso de estudio 2: Sistema de Identificación usando un Filtro de Respuesta Infinita al Impulso (IIR)

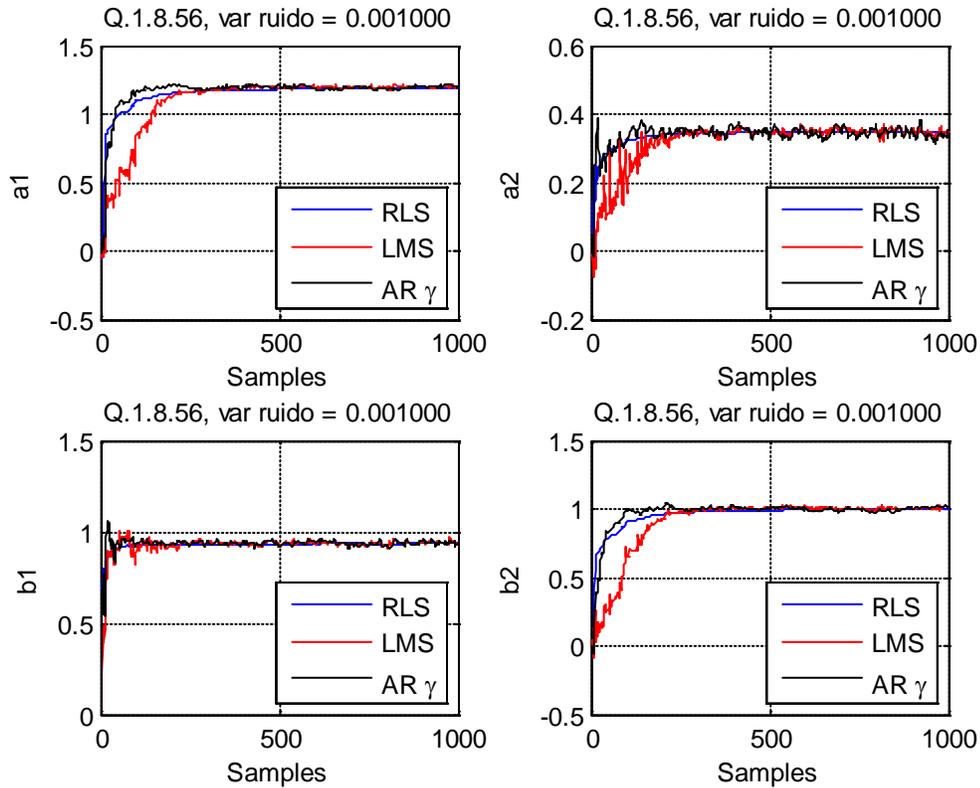
- a) Tamaño de palabra: Q.1.8.56  
 Número de muestras: 1000 muestras  
 Número de coeficientes: 4  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{0.9z+1}{z^2+1.2z+0.35}$   
 Varianza del ruido blanco de medición = 0



**Figura 4.11 Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$  con una palabra de tamaño Q.1.8.56 y sin ruido, usando un filtro IIR**

- b) Tamaño de palabra: Q.1.8.56  
 Número de muestras: 1000 muestras  
 Número de coeficientes: 4  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$

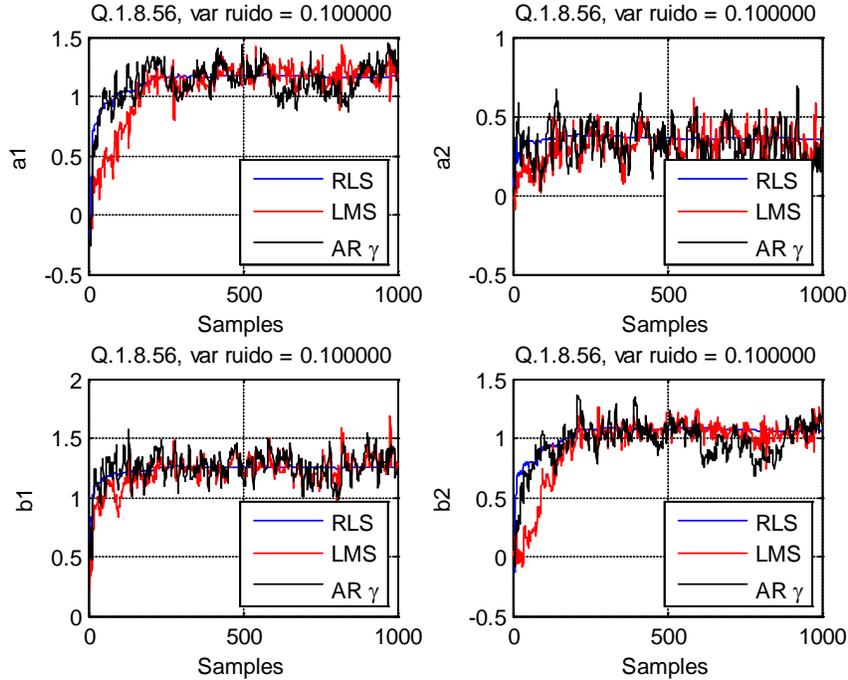
Sistema a Identificar:  $H(z) = \frac{0.9z+1}{z^2+1.2z+0.35}$   
 Varianza del ruido blanco de medición= 0.001



**Figura 4.12 Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$  con una palabra de tamaño Q.1.8.56 y con ruido 0.001, usando un filtro IIR**

- c) Tamaño de palabra: Q.1.8.56  
 Número de muestras: 1000 muestras  
 Número de coeficientes: 4  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$

Sistema a Identificar:  $H(z) = \frac{0.9z+1}{z^2+1.2z+0.35}$   
 Varianza del ruido blanco de medición= 0.1



**Figura 4.13 Región de convergencia de los coeficientes para los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.8.56 y con ruido 0.1, usando un filtro IIR**

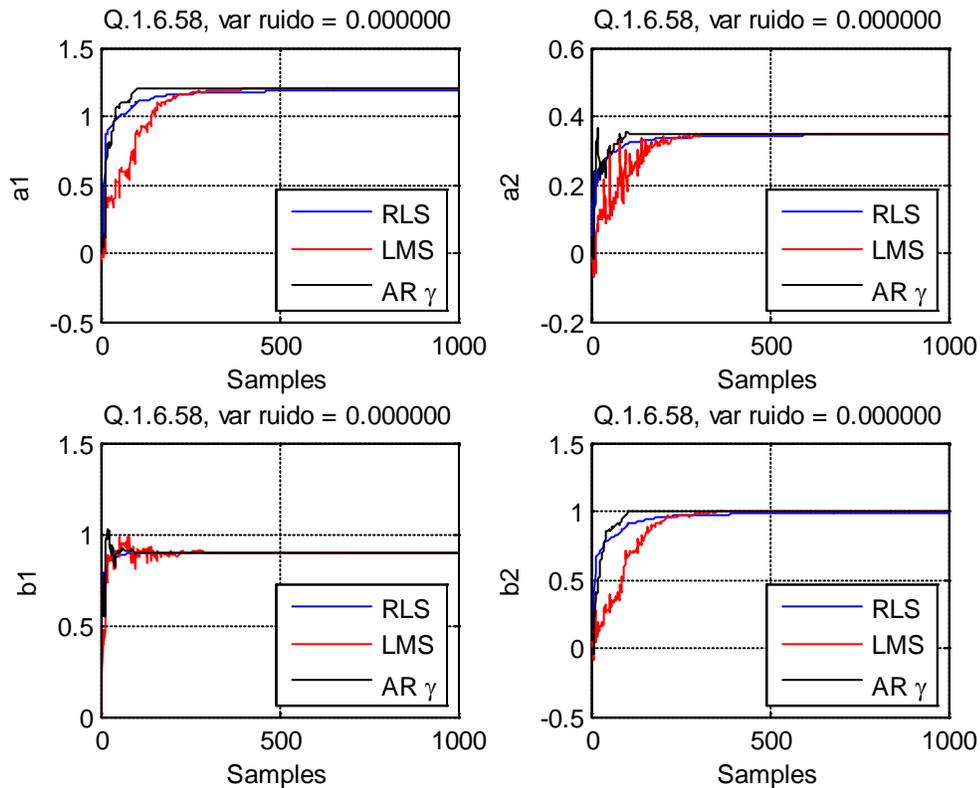
Los valores obtenidos para la media y la varianza del error de predicción en punto flotante para el  $AR\gamma$  son  $1.243 \times 10^{-3}$  y  $1.089 \times 10^{-2}$  respectivamente.

En la Tabla 4.5 se indican los valores obtenidos para la media y la varianza del error de predicción para los algoritmos estudiados y para diferentes valores de ruido de medición utilizando una longitud de palabra Q.1.8.56.

Q.1.8.56			
Media del Error de Predicción			
Ruido de Medida	LMS	RLS	$AR\gamma$
0	$2.795 \times 10^{-4}$	$1.548 \times 10^{-5}$	$1.751 \times 10^{-4}$
0.001	$1.570 \times 10^{-3}$	$1.179 \times 10^{-3}$	$1.378 \times 10^{-3}$
0.1	$1.296 \times 10^{-2}$	$9.879 \times 10^{-3}$	$1.449 \times 10^{-2}$
Varianza del Error de Predicción			
Ruido de Medida	LMS	RLS	$AR\gamma$
0	$4.483 \times 10^{-3}$	$1.151 \times 10^{-3}$	$1.696 \times 10^{-3}$
0.001	$7.861 \times 10^{-3}$	$3.667 \times 10^{-3}$	$5.823 \times 10^{-3}$
0.1	$2.937 \times 10^{-1}$	$2.081 \times 10^{-1}$	$3.509 \times 10^{-1}$

**Tabla 4.5 Media y varianza del error de predicción de los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.8.56 para diferentes sin ruido de medida, usando un filtro IIR**

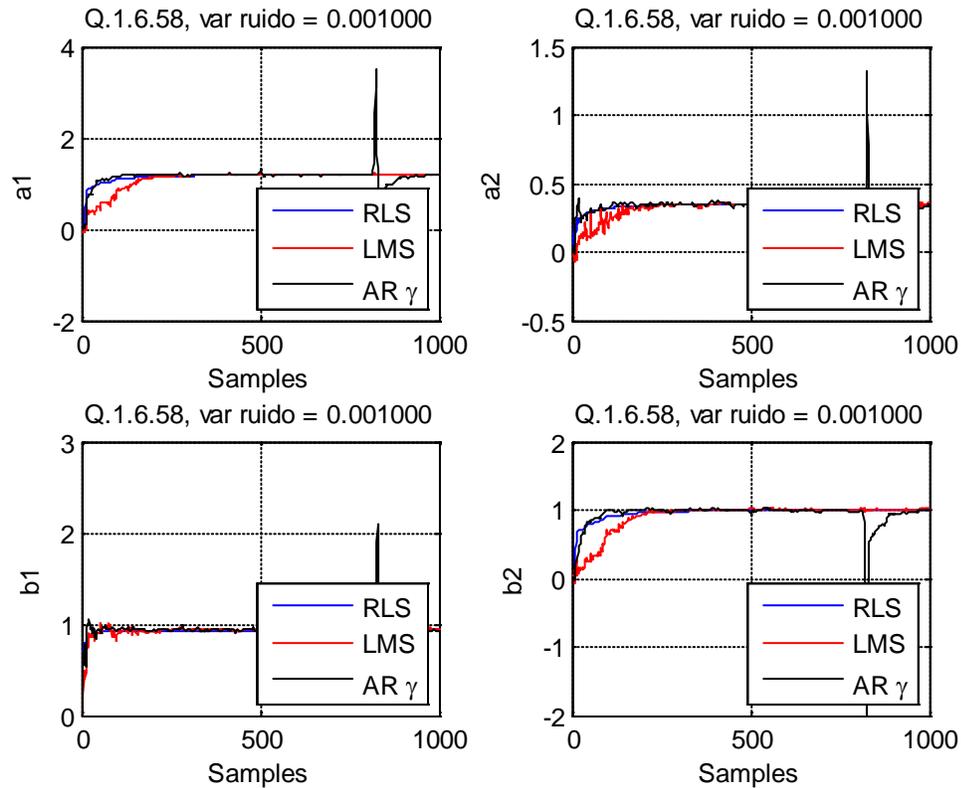
- d) Tamaño de palabra: Q.1.6.58  
 Número de muestras: 1000 muestras  
 Número de coeficientes: 4  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{0.9z+1}{z^2+1.2z+0.35}$   
 Varianza del ruido blanco de medición= 0



**Figura 4.14** Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$  con una palabra de tamaño Q.1.6.58 y con ruido 0, usando un filtro IIR

- e) Tamaño de palabra: Q.1.6.58  
 Número de muestras: 1000 muestras  
 Número de coeficientes: 4  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{0.9z+1}{z^2+1.2z+0.35}$

Varianza del ruido blanco de medición= 0.001



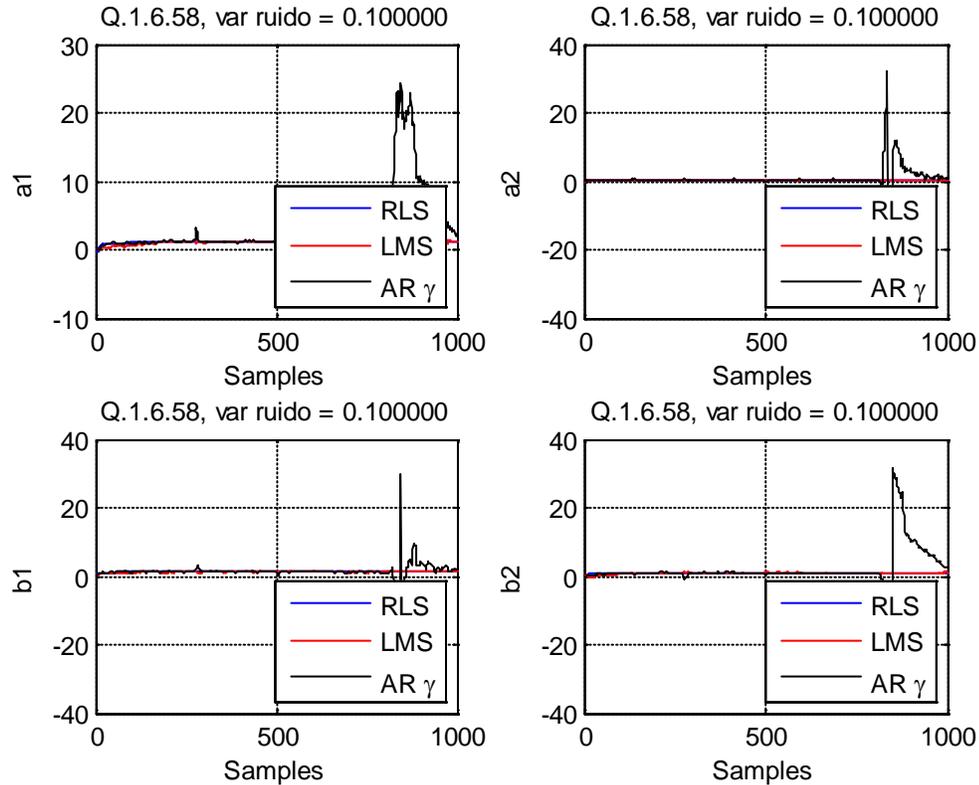
**Figura 4.15** Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$  con una palabra de tamaño Q.1.6.58 y con ruido 0.001, usando un filtro IIR

Al utilizar 6 bits en la parte entera, el AR $\gamma$  presenta mayor inestabilidad cuando hay presencia de ruido.

- f) Tamaño de palabra: Q.1.6.58  
 Número de muestras: 1000 muestras  
 Número de coeficientes: 4  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$

Sistema a Identificar:  $H(z) = \frac{0.9z+1}{z^2+1.2z+0.35}$

Varianza del ruido blanco de medición= 0.1



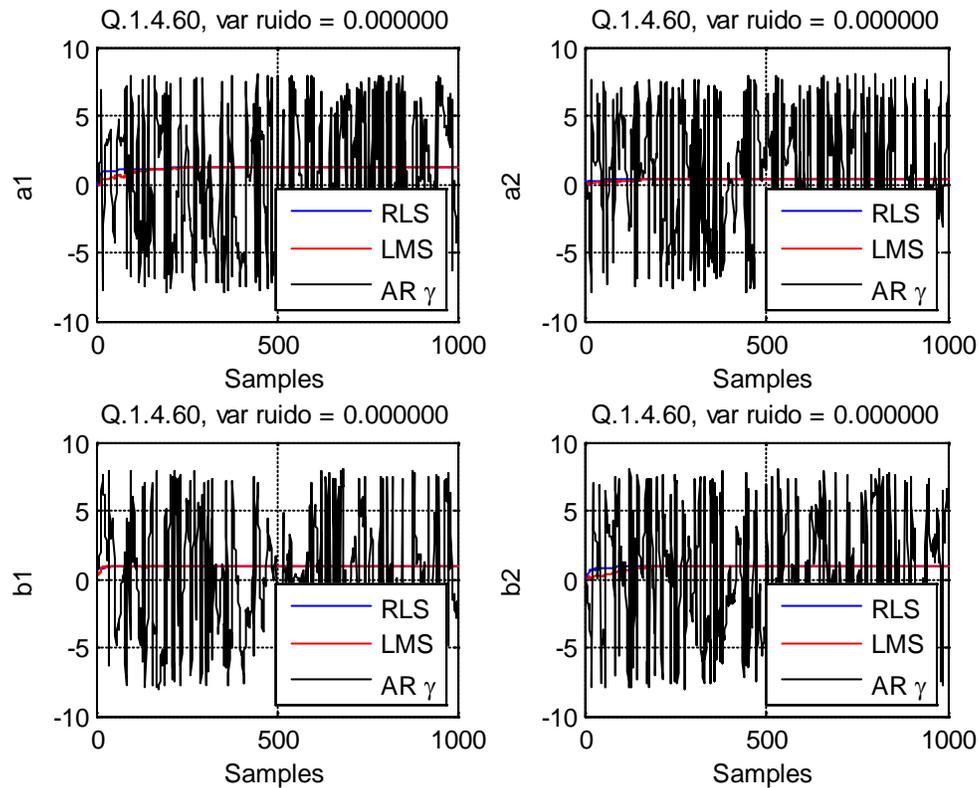
**Figura 4.16 Región de convergencia de los coeficientes para los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.6.58 y con ruido 0.1, usando un filtro IIR**

En la Tabla 4.6 se indican los valores obtenidos para la media y la varianza del error de predicción para los algoritmos estudiados y para diferentes valores de ruido de medición utilizando una longitud de palabra Q.1.6.58.

Q.1.6.58			
Media del Error de Predicción			
Ruido de Medida	LMS	RLS	$AR\gamma$
0	$2.183 \times 10^{-3}$	$1.629 \times 10^{-3}$	$1.262 \times 10^{-3}$
0.001	$2.236 \times 10^{-3}$	$1.714 \times 10^{-3}$	$1.345 \times 10^{-2}$
0.1	$3.445 \times 10^{-2}$	$4.474 \times 10^{-3}$	$8.475 \times 10^{-2}$
Varianza del Error de Predicción			
Ruido de Medida	LMS	RLS	$AR\gamma$
0	$2.139 \times 10^{-2}$	$5.462 \times 10^{-3}$	$1.089 \times 10^{-2}$
0.001	$2.336 \times 10^{-2}$	$6.675 \times 10^{-3}$	$1.806 \times 10^{-1}$
0.1	$1.823 \times 10^{-1}$	$1.145 \times 10^{-1}$	$1.599 \times 10^1$

**Tabla 4.6 Media y varianza del error de predicción de los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.6.58 para diferentes sin ruido de medida, usando un filtro IIR**

- g) Tamaño de palabra: Q.1.4.60  
 Número de muestras: 1000 muestras  
 Número de coeficientes: 4  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{0.9z+1}{z^2+1.2z+0.35}$   
 Varianza del ruido blanco de medición= 0.1



**Figura 4.17** Región de convergencia de los coeficientes para los algoritmos LMS, RLS y  $AR_\gamma$  con una palabra de tamaño Q.1.4.60 y sin ruido, usando un filtro IIR

En la siguiente tabla se presentan los valores obtenidos para la media y la varianza del error de predicción, con un valor de ruido de medición de 0.1:

<b>Q.1.4.60</b>			
<b>Media del Error de Predicción</b>			
Ruido de Medida	<b>LMS</b>	<b>RLS</b>	<b>AR<math>\gamma</math></b>
0.1	$2.183 \times 10^{-3}$	$1.629 \times 10^{-3}$	$1.759 \times 10^{-1}$
<b>Varianza del Error de Predicción</b>			
Ruido de Medida	<b>LMS</b>	<b>RLS</b>	<b>AR<math>\gamma</math></b>
0.1	$2.139 \times 10^{-2}$	$5.462 \times 10^{-3}$	$2.080 \times 10^1$

**Tabla 4.7 Media y varianza del error de predicción de los algoritmos LMS, RLS y AR $\gamma$  con una palabra de tamaño Q.1.4.60 para diferentes con ruido de medida de 0.1, usando un filtro IIR**

En esta experiencia, el algoritmo entra en “overflow” debido a que se limitó a 4 la cantidad de bits en la parte entera.

Con la implementación de los algoritmos utilizando una estructura IIR, y las experiencias antes realizadas se observa lo siguiente:

- Al utilizar un ruido de medición de 0.1, la varianza del error de predicción aumentó (Tablas 4.4, 4.5 y 4.6) hasta el punto que genera inconvenientes en la estabilidad del algoritmo.
- En la medida en que se limita la cantidad de bits en la parte entera, las varianzas aumentan.
- El número mínimo de bits en la parte entera para que el algoritmo se comporte de manera adecuada es de 6 usando una configuración IIR, así el problema de divergencia del AR $\gamma$  se debe a insuficiencia en el número de bits utilizados para representar la parte entera.

## 4.2 Experiencia 2. Estudio del efecto de la longitud de palabra en la media y la varianza del error de predicción del AR $\gamma$

El objetivo de este experimento es determinar como cambia la varianza en función de la longitud de palabra para los algoritmos implementados y en particular para el AR $\gamma$ .

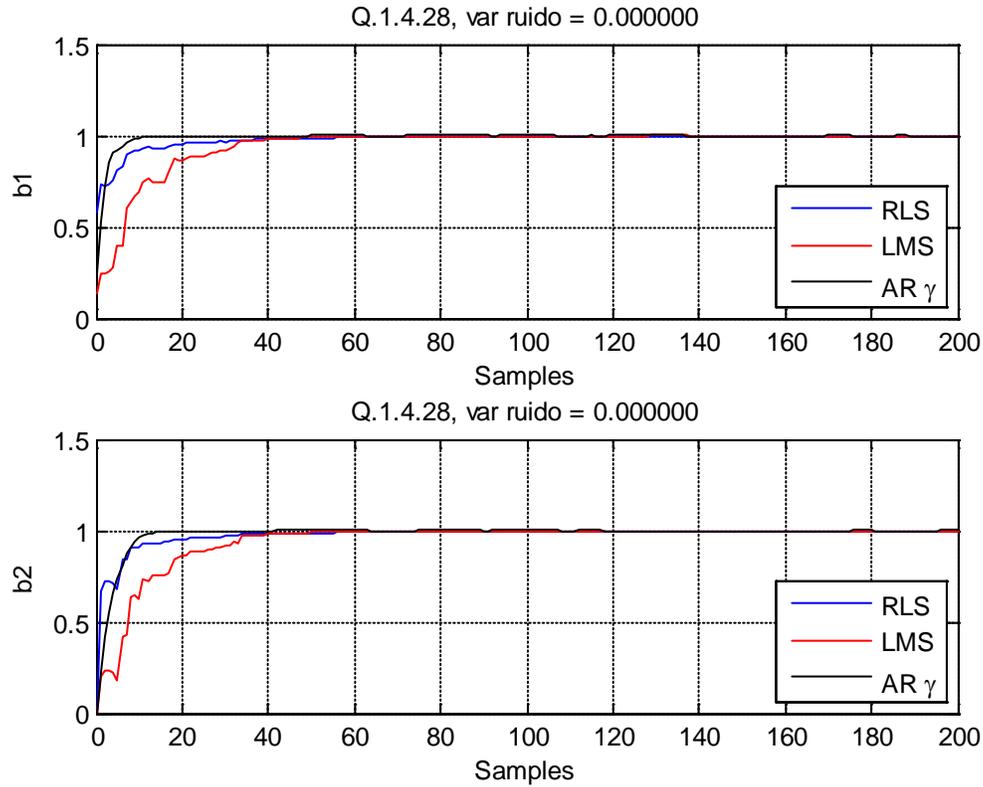
El procedimiento consiste en que a partir del número mínimo de bits requerido para la parte entera para un óptimo funcionamiento se emplean longitudes de 32, 16 y 8 bits para analizar el comportamiento del error de predicción. Se espera que en la medida que la longitud de palabra disminuya, la varianza del error de predicción debe aumentar hasta el punto en el cual el algoritmo no pueda operar.

Se utilizaron las simulaciones mostradas en el Anexo 3.

A continuación se presentan los resultados de las simulaciones realizadas, las gráficas que muestran la región de convergencia de los coeficientes y los resultados de la media y la varianza del error de predicción para los casos de estudio definidos.

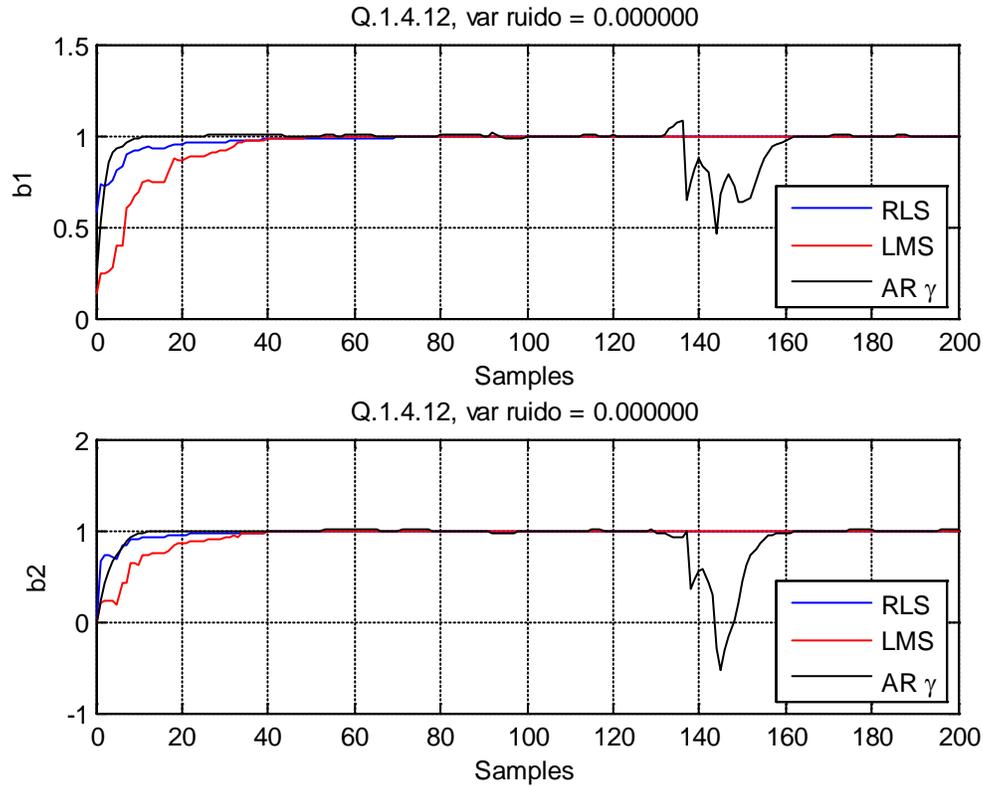
### 4.2.1 Caso de estudio 1: Sistema de Identificación usando un Filtro de Respuesta Finita al Impulso (FIR)

- a) Tamaño de palabra: Q.1.4.28
  - Número de muestras: 200 muestras
  - Número de coeficientes: 2
  - $m_1 = 1$
  - $\alpha = 1$
  - $\gamma = 2.04$
  - Sistema a Identificar:  $H(z) = \frac{z+1}{z}$
  - Varianza del ruido blanco de medición = 0
-



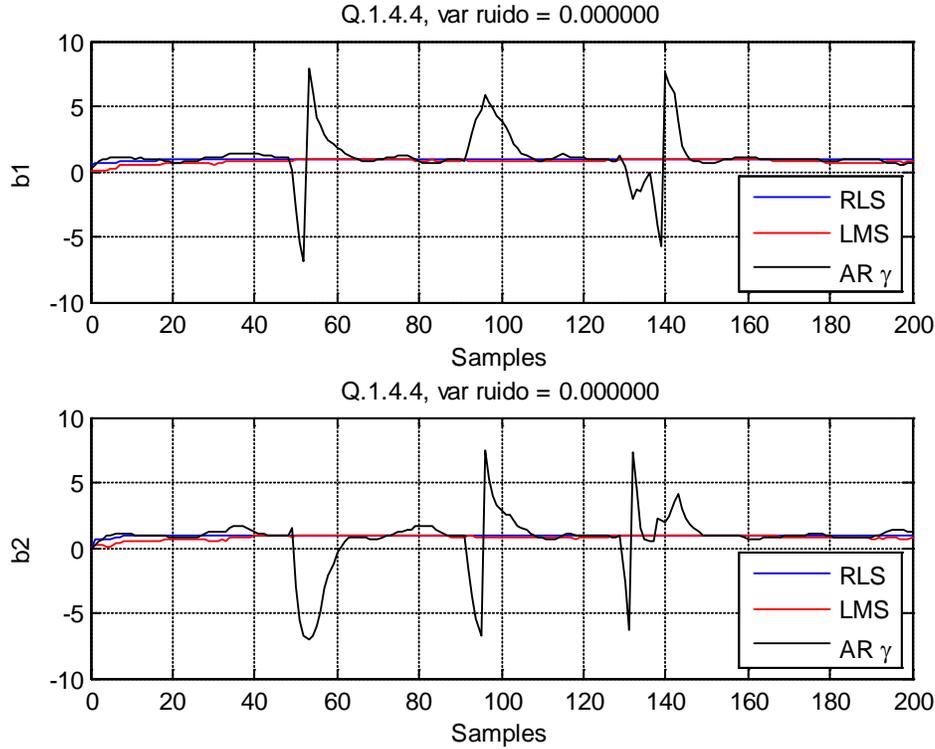
**Figura 4.18** Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR $\gamma$  con una palabra de tamaño Q.1.4.28 y sin ruido, usando un filtro FIR

- b) Tamaño de palabra: Q.1.4.12  
 Número de muestras: 200 muestras  
 Número de coeficientes: 2  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{z+1}{z}$   
 Varianza del ruido blanco de medición = 0



**Figura 4.19** Región de convergencia de los coeficientes para los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.4.12 y sin ruido, usando un filtro FIR

- c) Tamaño de palabra: Q.1.4.4  
 Número de muestras: 200 muestras  
 Número de coeficientes: 2  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{z+1}{z}$   
 Varianza del ruido blanco de medición = 0



**Figura 4.20** Región de convergencia de los coeficientes para los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.4.4 y sin ruido, usando un filtro FIR

El algoritmo  $AR\gamma$  presenta desbordamiento.

En la Tabla 4.8 se indican los valores obtenidos para la media y la varianza del error de predicción para los algoritmos estudiados con longitudes de palabra de 32, 16 y 8 bits.

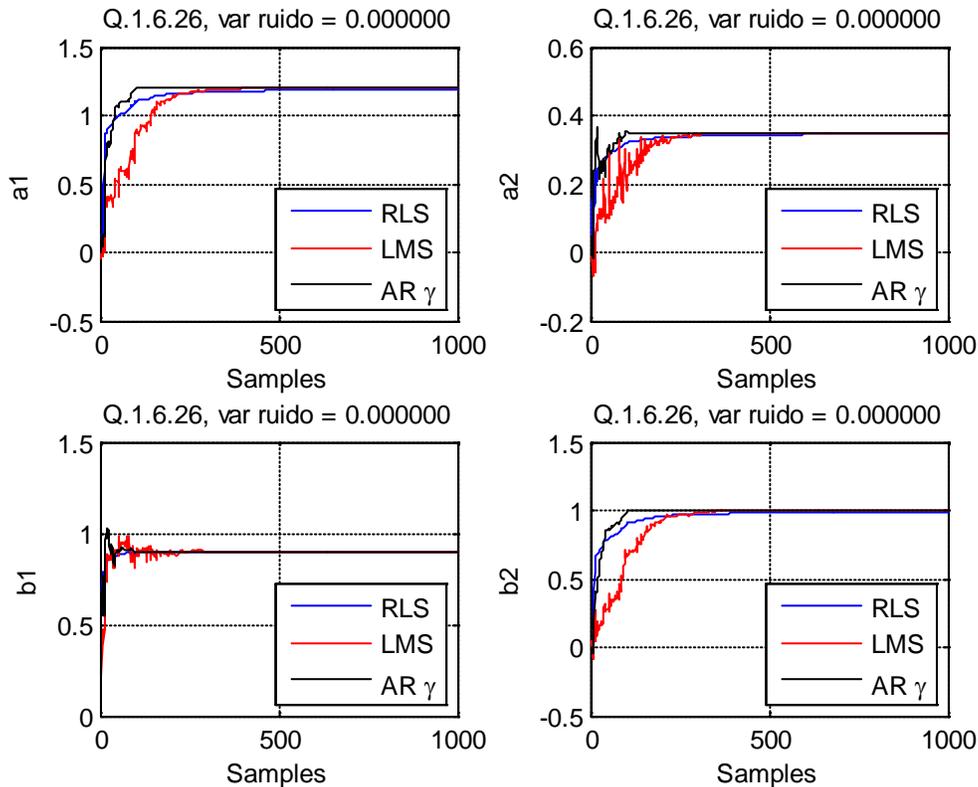
Tamaño de Palabra	Media del Error de Predicción		
	LMS	RLS	$AR\gamma$
Q.1.4.28	$2.785 \times 10^{-2}$	$1.412 \times 10^{-2}$	$1.552 \times 10^{-2}$
Q.1.4.12	$2.804 \times 10^{-2}$	$1.440 \times 10^{-2}$	$4.835 \times 10^{-2}$
Q.1.4.4	$5.312 \times 10^{-2}$	$4.406 \times 10^{-2}$	$2.021 \times 10^{-1}$
	Varianza del Error de Predicción		
	LMS	RLS	$AR\gamma$
Q.1.4.28	$4.930 \times 10^{-2}$	$1.380 \times 10^{-2}$	$1.625 \times 10^{-2}$
Q.1.4.12	$4.947 \times 10^{-2}$	$1.387 \times 10^{-2}$	$7.320 \times 10^{-2}$
Q.1.4.4	$1.703 \times 10^{-1}$	$3.838 \times 10^{-2}$	$3.585 \times 10^0$

**Tabla 4.8** Media y varianza del error de predicción de los algoritmos LMS, RLS y  $AR\gamma$  con palabras de longitud variable, sin ruido de medida y usando un filtro FIR

Es claro que en la medida que se limita el tamaño de la palabra, la media y la varianza del error de predicción aumentan. Para una longitud de palabra de 8 bits, el  $AR\gamma$  aumenta su varianza en el error de predicción de manera considerable. Así, es claro que el  $AR\gamma$  puede implementarse en hardware hasta de 16 bits.

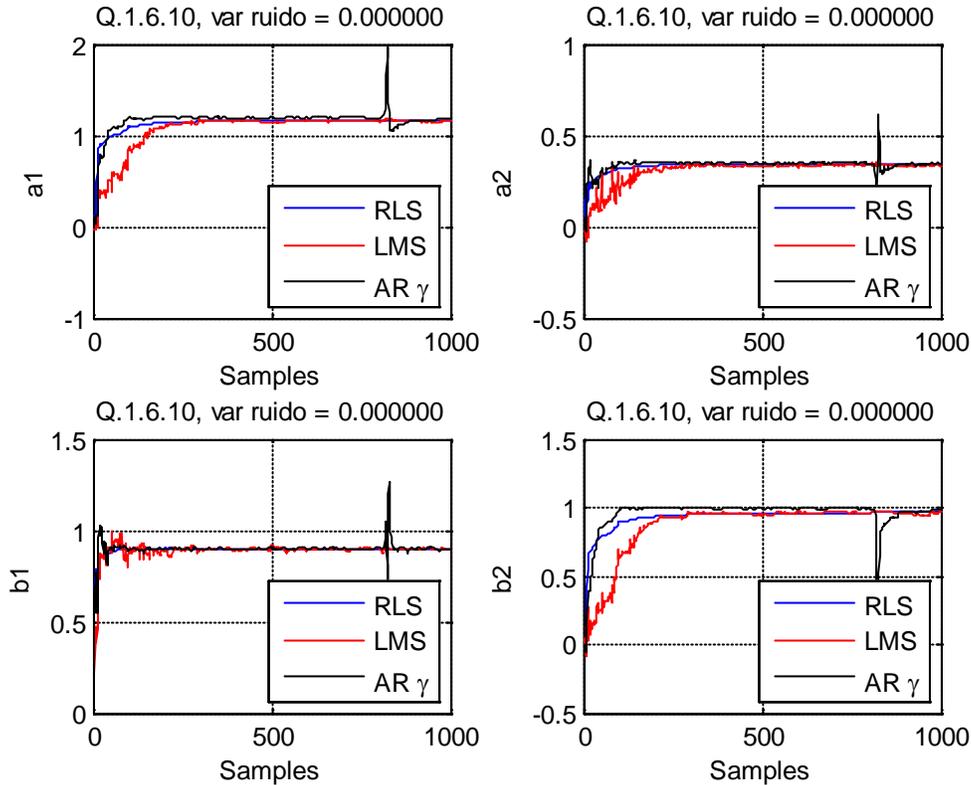
#### 4.2.2 Caso de estudio 2: Sistema de Identificación usando un Filtro de Respuesta Infinita al Impulso (IIR)

- a) Tamaño de palabra: Q.1.6.26  
 Número de muestras: 1000 muestras  
 Número de coeficientes: 4  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{0.9z+1}{z^2+1.2z+0.35}$   
 Varianza del ruido blanco de medición = 0



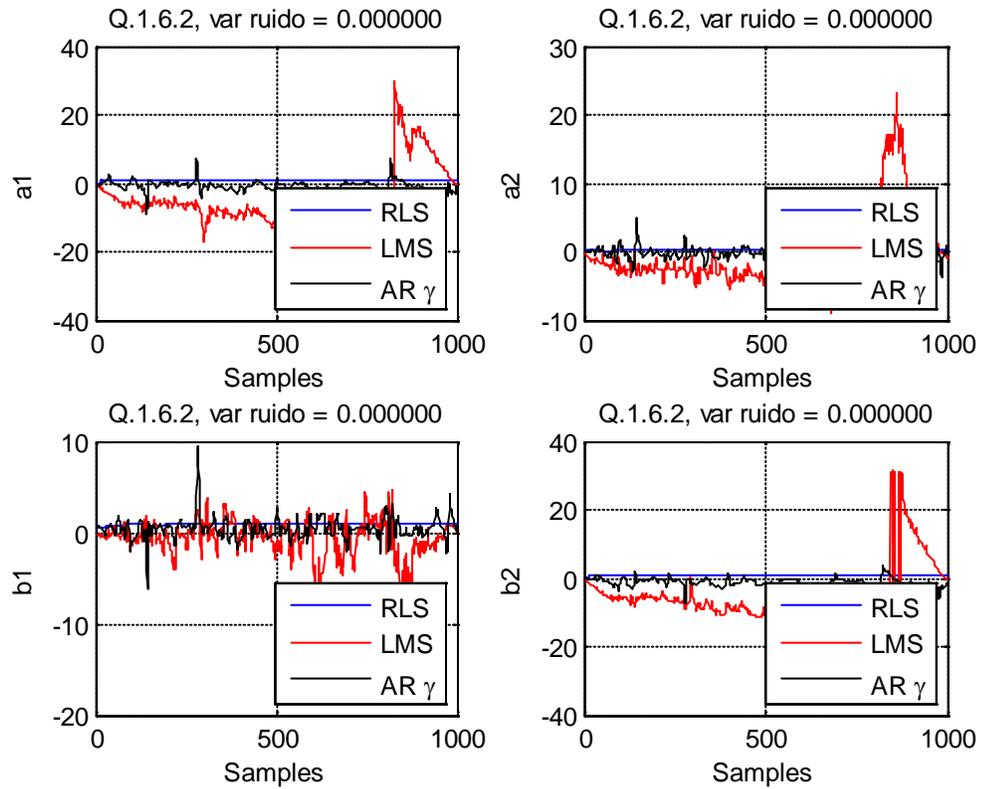
**Figura 4.21** Región de convergencia de los coeficientes para los algoritmos LMS, RLS y  $AR\gamma$  con una palabra de tamaño Q.1.6.26 y sin ruido, usando un filtro IIR

- b) Tamaño de palabra: Q.1.6.10  
 Número de muestras: 1000 muestras  
 Número de coeficientes: 4  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{0.9z+1}{z^2+1.2z+0.35}$   
 Varianza del ruido blanco de medición = 0



**Figura 4.22** Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR  $\gamma$  con una palabra de tamaño Q.1.6.10 y con ruido 0, usando un filtro IIR

- c) Tamaño de palabra: Q.1.6.2  
 Número de muestras: 1000 muestras  
 Número de coeficientes: 4  
 $m_1 = 1$   
 $\alpha = 1$   
 $\gamma = 2.04$   
 Sistema a Identificar:  $H(z) = \frac{0.9z+1}{z^2+1.2z+0.35}$   
 Varianza del ruido blanco de medición = 0



**Figura 4.23** Región de convergencia de los coeficientes para los algoritmos LMS, RLS y AR  $\gamma$  con una palabra de tamaño Q.1.6.2 y sin ruido, usando un filtro IIR

Se presenta desbordamiento, incluso para los algoritmos LMS y RLS.

En la Tabla 4.9 se indican los valores obtenidos para la media y la varianza del error de predicción para los algoritmos estudiados con longitudes de palabra de 32, 16 y 8 bits.

Tamaño de Palabra	Media del Error de Predicción		
	LMS	RLS	AR $\gamma$
Q.1.6.26	$2.183 \times 10^{-3}$	$1.629095 \times 10^{-3}$	$1.262606 \times 10^{-3}$
Q.1.6.10	$2.253 \times 10^{-3}$	$1.117187 \times 10^{-3}$	$8.750000 \times 10^{-4}$
Q.1.6.2	$1.002500 \times 10^{-1}$	$1.675000 \times 10^{-2}$	$3.442500 \times 10^{-1}$
	Varianza del Error de Predicción		
	LMS	RLS	AR $\gamma$
Q.1.6.26	$2.139398 \times 10^{-2}$	$5.462810 \times 10^{-3}$	$1.089822 \times 10^{-2}$
Q.1.6.10	$2.339936 \times 10^{-2}$	$5.912442 \times 10^{-3}$	$2.963450 \times 10^{-2}$
Q.1.6.2	$8.371160 \times 10^{-1}$	$1.237807 \times 10^{-1}$	$1.090809 \times 10^{-1}$

**Tabla 4.9 Media y varianza del error de predicción de los algoritmos LMS, RLS y AR $\gamma$  con palabras de longitud variable, sin ruido de medida y usando un filtro IIR**

Como conclusión, se nota que en la medida que se limita el tamaño de la palabra, la media y la varianza del error de predicción aumentan

De los experimentos anteriores, se tiene como conclusión que se determinó que para un sistema implementado con FIR el algoritmo requiere como mínimo 4 bits en la parte entera y para el caso de un IIR se requieren como mínimo 6 bits en la parte entera. Además la longitud mínima de palabra para una operación óptima del algoritmo es 16 bits.

De esta forma se contribuye a dar cumplimiento a los objetivos específicos 1 y 3 ya que se han comparado las propiedades de convergencia del Algoritmo Acelerador Regresivo, versión  $\gamma$  (AR $\gamma$ ) cuando éste se implementa mediante aritmética de punto fijo y punto flotante y se ha analizado el efecto de diferentes longitudes de palabra de la aritmética de punto fijo (16, 32, 64 bits) en la convergencia del Algoritmo Acelerador Regresivo.

### 4.3 Experiencia 3. Análisis del efecto de la longitud de palabra en los parámetros de ajuste del AR $\gamma$

El objetivo de esta experiencia es validar que los parámetros de ajuste ( $\alpha$ ,  $\gamma$  y  $m_1$ ) encontrados de manera teórica en (Jojoa, 2003) para condiciones de punto flotante, siguen siendo válidos cuando se limita el tamaño de la palabra en el AR $\gamma$ .

El procedimiento consiste en realizar un proceso de simulación cuya finalidad es obtener el valor de los parámetros óptimos mediante la implementación de una función de optimización (ver Anexo 5). Dicha función, tiene como datos iniciales

los parámetros determinados por (Jojoa, 2003). En primera instancia, se asigna una calificación (una combinación de valores) a la terna conformada por  $\alpha$ ,  $m_1$  y  $\gamma$  (que corresponde a un valor óptimo – mínimo local) y luego Matlab se encarga de modificarlos hasta minimizar la varianza del error de predicción en estado estacionario. Una vez obtenido el valor mínimo a nivel local, se despliegan los nuevos parámetros.

El experimento hizo uso de la implementación del sistema de identificación que utiliza un filtro IIR, como se muestra en la Figura 4.24 y se detalla en el Anexo 4, con las siguientes condiciones:

Tamaño de palabra: variable: Q.1.6.58, Q.1.6.26 y Q.1.6.10

Número de muestras: 1000 muestras

Número de coeficientes: 4

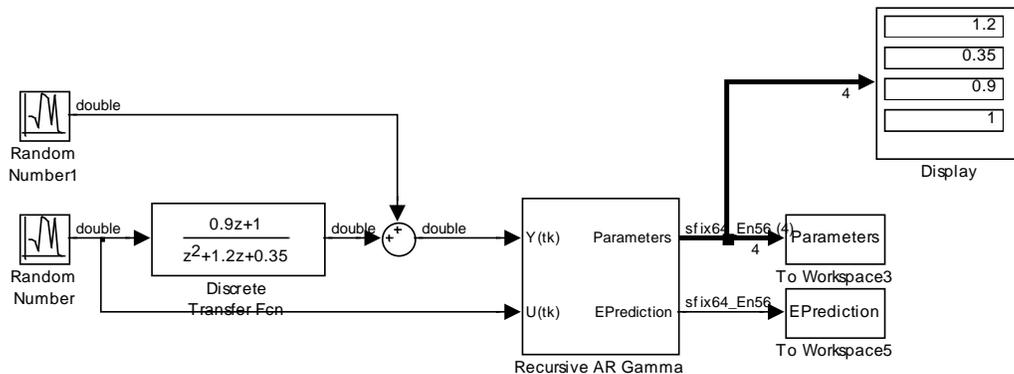
$m_1 = 1$

$\alpha = 1$

$\gamma = 2.04$

Sistema a Identificar:  $H(z) = \frac{0.9z+1}{z^2+1.2z+0.35}$

Varianza del ruido blanco de medición = 0



**Figura 4.24 Optimización de Parámetros usando un Sistema IIR**

Las simulaciones se realizaron para una cantidad de 6 bits en la parte entera y longitudes de palabra variable de 64, 32 y 16 bits, obteniendo los resultados tal y como se puede observar en la Tabla 4.10.

Parámetros Iniciales			Parámetros Óptimos (punto flotante)			Parámetros Óptimos Q.1.6.58			Parámetros Óptimos Q.1.6.26			Parámetros Óptimos Q.1.6.10		
$m_1$	$\alpha$	$\gamma$	$m_1$	$\alpha$	$\gamma$	$m_1$	$\alpha$	$\gamma$	$m_1$	$\alpha$	$\gamma$	$m_1$	$\alpha$	$\gamma$
1	1.0	2.04	1	1.0	2.04	1	1.0	2.04	1	1.0	2.04	1	1.0	2.04
1	1.5	1.35	1	1.5	1.35	1	1.5	1.35	1	1.5	1.35	1	1.5	1.35
3	1.0	0.68	3	1.0	0.68	3	1.0	0.68	3	1.0	0.68	3	1.0	0.68
3	1.5	0.45	3	1.5	0.45	3	1.5	0.45	3	1.5	0.45	3	1.5	0.45
11	1.0	0.18	11	1.0	0.18	11	1.0	0.18	11	1.0	0.18	11	1.0	0.18
11	1.5	0.12	11	1.5	0.12	11	1.5	0.12	11	1.5	0.12	11	1.5	0.12

**Tabla 4.10 Optimización de Parámetros del AR $\gamma$**

Con la anterior simulación se pudo verificar que los parámetros encontrados por Jojoa(2003) mediante un análisis matemático, corresponden a los arrojados por la simulación implementada en el presente trabajo, permitiendo concluir que no se requiere hacer modificaciones al Algoritmo Acelerador Regresivo al implementarlo en punto fijo, ya que los resultados se mantienen acordes al tipo de señales utilizadas. Con los tamaños de palabra utilizados se ha garantizado la convergencia a partir de los resultados obtenidos en el experimento 1, presentándose menor varianza en el error de predicción, sin embargo en la medida que se reduce el tamaño de la palabra a 16 bits el algoritmo converge pero se pierde precisión.

La mejor opción para la combinación de los parámetros es  $m_1 = 1$ ,  $\alpha = 2$ ,  $\gamma = 1$ , debido a que son números que se representan exactamente con dos bits y se está trabajando con un experimento que es el mínimo, por lo que se escogen los dos enteros más cercanos a 2.

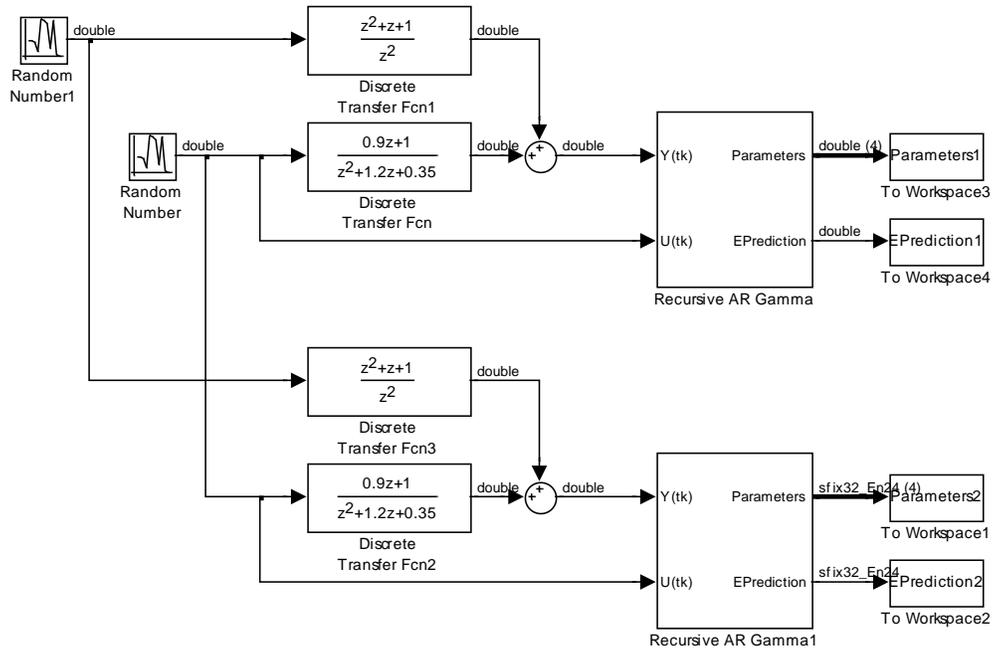
A colocar números que exijan más de la representación, como por ejemplo utilizar  $\sqrt[3]{2}$  para  $m_1$ ,  $\alpha\gamma\gamma$ ; se presentó corrección y se observó que el AR $\gamma$  converge con total precisión para una longitud de 32 bits. En 16 bits, se presentó convergencia pero disminuyó la precisión.

Dando cumplimiento al objetivo específico número 2 se validaron los parámetros de funcionamiento identificados para el algoritmo en punto fijo con diferentes longitudes de palabra (16, 32, 64 bits). y mediante la función de optimización se corroboraron los resultados teóricos obtenidos en Jojoa (2003).

#### 4.4 Experiencia 4. Análisis del error de cuantización en el AR $\gamma$

Este experimento tiene como finalidad determinar mediante un proceso de simulación el tipo de error que se introduce en el proceso de cuantización al utilizar los valores finitos en el AR $\gamma$ .

El procedimiento consiste en aplicar la misma señal de entrada al algoritmo AR $\gamma$  con condiciones de punto flotante y punto fijo y evaluar el valor del error de cuantización en la operación de división como se observa en la Figura 4.25 (ver Anexo5), para diferentes longitudes de palabra. La diferencia entre el error de cuantización para punto flotante en la división y el error de la misma operación en punto fijo es considerado como un valor de error introducido por la cuantización.



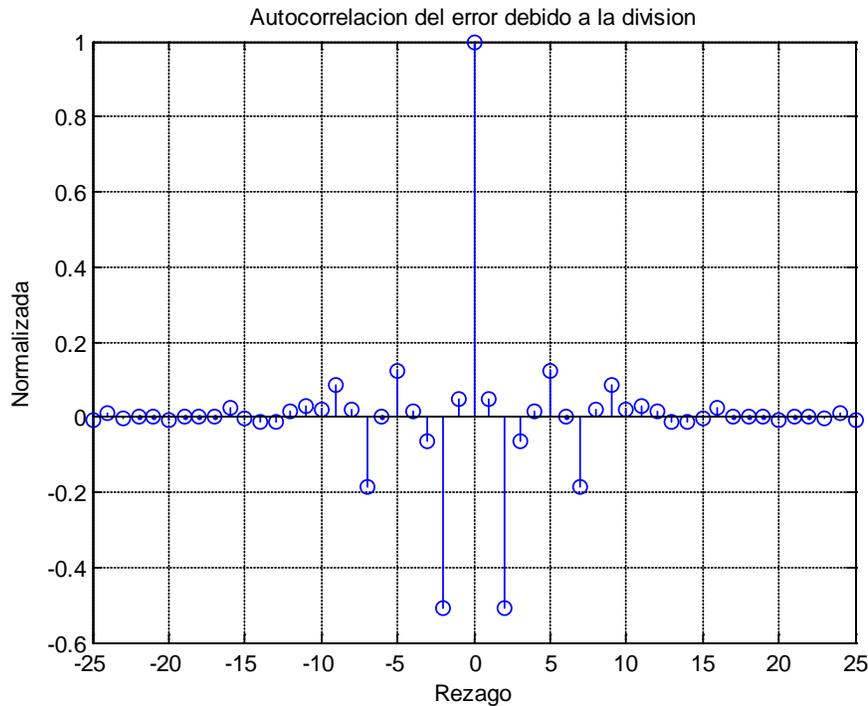
**Figura 4.25** Cuantización del AR $\gamma$

En la Tabla 4.11 se muestran los resultados de las varianzas del error de cuantización, donde se observa que en la medida que el tamaño de la palabra se reduce, la varianza de éste error aumenta.

Varianza del Error de Cuantización en el AR $\gamma$		
Q.1.6.58	Q.1.6.26	Q.1.6.10
$9.866 \times 10^{-27}$	$3.430 \times 10^{-10}$	$4.406 \times 10^{-1}$

**Tabla 4.11** Varianza del Error de Cuantización

Posteriormente, como se observa en el Anexo 5., con el fin de determinar la naturaleza del ruido, se calculó la función de autocorrelación y se graficó como se muestra en la Figura 4.26



**Figura 4.26 Autocorrelación del Error de Cuantización**

Es evidente que el error de cuantización en estado estacionario introducido por la división es una señal de ruido coloreado, donde el error actual depende del error anterior, observándose una fuerte correlación entre sus componentes.

De la experiencia realizada, se puede concluir que el ruido de cuantificación es acumulativo y en la medida que se limita el tamaño de la palabra, aumenta la varianza del error de cuantización, haciendo de la división la parte más crítica del algoritmo, brindando un elemento trascendental en el análisis del efecto de la aritmética de punto fijo en el algoritmo.

# Capítulo 5

## Conclusiones y Recomendaciones

A continuación se presentan las conclusiones y recomendaciones para trabajos futuros, obtenidos a partir del trabajo de investigación relacionado con el algoritmo acelerador regresivo versión  $\gamma$ .

### 5.1 Conclusiones

- El análisis del efecto de la aritmética de punto fijo en la convergencia del algoritmo realizado mediante procesos de simulación, permitió identificar que el algoritmo acelerador regresivo versión  $\gamma$  mantiene sus características de comportamiento en cuanto a rapidez y precisión para longitudes de palabra de 16 bits, utilizando mínimo 4 bits para estructuras FIR y 6 bits para configuraciones IIR; situación que hace muy viable su implementación en hardware.
- El comportamiento del algoritmo acelerador regresivo bajo el efecto de aritmética de punto fijo muestra que en la medida que se limita la longitud de la palabra, la varianza del error de predicción se hace más significativa.
- El análisis de los parámetros de ajuste en punto fijo para diferentes longitudes de palabra y la implementación de un proceso de optimización de parámetros mediante Simulink, ratificaron los resultados obtenidos de manera teórica en (Jojoa, 2003) manteniendo los resultados acordes al tipo de señales utilizadas avanzando de esta forma en el estudio del algoritmo y reuniendo así elementos para justificar su futura implementación en hardware.
- La formulación del efecto del error de cuantización sobre el algoritmo permitió identificar los puntos más críticos en las ecuaciones que lo conforman y la realización de las simulaciones orientadas a determinar el error de cuantización permitieron concluir que el ruido introducido en el algoritmo es acumulativo y coloreado.

- La definición e implementación de los casos de estudio en Simulink, permitieron obtener aportes adicionales como el probar el algoritmo en sistemas de identificación con filtros tipo IIR, contaminando la señal con ruido blanco y comparando el comportamiento del algoritmo con el algoritmo RLS y el LMS.

## **5.2 Recomendaciones**

- A partir de las simulaciones realizadas en Simulink es importante poder realizar la implementación hardware para verificar mediante experimentación los resultados obtenidos.
  - Con el fin de contribuir con el efecto de la limitación generar proyectos que analicen otras características del algoritmo como desajuste, tracking, robustez, etc.
  - Para construir el análisis de estabilidad numérica del algoritmo se pueden realizar estudios teóricos que bajo el efecto de valores finitos calculen los valores del error de cuantización y revisen las demás características a nivel de análisis matemático.
-

## Bibliografía

Escobar L. H., (2033). *Estudio comparativo y aplicaciones de algoritmos de filtrado adaptable*. Departamento de Electrónica, División de Ingeniería Eléctrica Facultad de Ingeniería, UNAM.

Frías, A., (2005). *Efectos de longitud finita de palabra en filtros digitales adaptivos*. Universidad de Guanajuato, México.

Gerken, M.; Pait, F.; Jojoa P.E. (2000). *An Adaptive Algorithm with Parameter Acceleration*. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Ed. Piscataway, N.J. : I.E.E.E.

Haykin, S., (2002). *Adaptive Filter Theory*, 4a. Ed., Upper Saddle River, Prentice Hall, USA.

Jojoa, P.E. (2003). Um Algoritmo Acelerador de Parâmetros, *Tesis de Doctorado*.  
*Escola Politécnica da Universidade de de Sao Paulo, Brasil.*

Jojoa, P.E. (1999). Análise de Algoritmo Adaptativo que Ajusta a Diferencia de Segunda Ordem dos Parametros, *Tesis de Maestrado*. *Escola Politécnica da Universidade de de Sao Paulo, Brasil.*

Jung, L. L., (1999). *System Identification – Theory for the user*. Second Edition, Linkping University Sweden Prentice Hall.

Manolakis, D, (2000). *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing*. McGraw Hill, USA.

Mathews, J., (1990). *Stochastic Gradient Adaptive Filters with Gradient Adaptive*. Step Size. IEEE Int. Conf. Acoust., Speech, Signal Processing, USA.

Nabil R., Ali H. (2003). *Fixed-point steady-state analysis of adaptive filters*. International Journal of Adaptive Control and Signal Processing.

Pait, F. (1998). *A Tuner that Accelerates Parameters, Systems and Control Letters*. vol. 35, n.1, pp. 65-68.

Realpe J. C., (2009). *Estudio del Efecto de la Variación Temporal de Parámetros del Algoritmo Acelerador Regresivo Versión gamma (AR gamma)*. Universidad del Cauca. Maestría en Ingeniería Electrónica y Telecomunicaciones.

Widrow, B., Stearns, S. (1985). *Adaptive Signal Processing, Englewood Cliffs*, Prentice Hall, USA.

<http://www.mathworks.com/products/simulink/description1.html>

---

## Anexo 1.

### Algoritmo Acelerador Regresivo Versión $\gamma$ (AR $\gamma$ )

En 1998, fue propuesto por F Pait (Pait, 1998), el Algoritmo Acelerador de Tiempo Continuo que consistió en el ajuste de la segunda derivada (aceleración) del error con respecto a los coeficientes del algoritmo. La principal característica analizada fue el mejor compromiso entre la velocidad de convergencia y la variación del error de estimación. Las ecuaciones que describen este algoritmo son:

$$\dot{\mathbf{w}} = -\mathbf{M}\mathbf{x}(t)e(t) = \mathbf{q}(t) \quad (1)$$

$$\dot{\mathbf{q}}(t) = -\mathbf{M}_1\mathbf{x}(t)e(t) + 2(\mathbf{M}_2 + \mathbf{x}(t)\mathbf{x}^T(t)\mathbf{M}_2\mathbf{M}_3)\mathbf{q}(t) \quad (2)$$

$$e(t) = \mathbf{x}^T\mathbf{w}(t) - d(t) \quad (3)$$

donde:  $\mathbf{w}(t)$  es el vector de coeficientes del filtro adaptativo,  $\mathbf{x}(t)$  corresponde a la señal de entrada,  $e(t)$  el error de estimación,  $d(t)$  la señal deseada,  $\mathbf{q}(t)$  la velocidad de adaptación de los coeficientes del filtro y  $\dot{\mathbf{q}}(t)$  la aceleración de adaptación de los coeficientes del filtro. Las matrices  $\mathbf{M}$ ,  $\mathbf{M}_1$ ,  $\mathbf{M}_2$  y  $\mathbf{M}_3$  son  $M$ -dimensionales, simétricas y definidas positivas ( $M$  es la dimensión del vector de la señal de entrada).

De acuerdo con el criterio de estabilidad de Lyapunov este sistema es estable cuando las matrices  $\mathbf{M}_1$ ,  $\mathbf{M}_2$  y  $\mathbf{M}_3$  satisfacen la condición de que: (Jojoa, 1999):

$$4\mathbf{M}_1\mathbf{M}_3\mathbf{M}_1\mathbf{M}_2 > \mathbf{I}_M \quad (4)$$

donde  $\mathbf{I}_M$  es la Matriz Identidad de dimensión  $M$  y

$$\mathbf{M}_2\mathbf{M}_1\mathbf{M}_3 + \mathbf{M}_1\mathbf{M}_3\mathbf{M}_2 > \mathbf{M}_1^{-1}/2 \quad (5)$$

Este algoritmo presenta un mejor compromiso entre la velocidad de convergencia y la variación del error de estimación, en relación a los algoritmos de gradiente estocástico. (Pait, 1998).

En 1999 se realizó la discretización del algoritmo acelerador, generando tres versiones del algoritmo en tiempo discreto (Jojoa, 1999). Estas versiones fueron obtenidas utilizando los métodos de discretización de Euler y la transformación bilineal. Los resultados obtenidos con los métodos de discretización de Euler arrojaron buenos resultados con respecto a desempeño y complejidad computacional. Los algoritmos generados fueron el Algoritmo Acelerador Progresivo Convencional Matricial (APCM), obtenido con el método de Euler progresivo y el Algoritmo Acelerador Regresivo Convencional Matricial (ARCM), obtenido con el método de Euler regresivo. Las ecuaciones de estos algoritmos se presentan a continuación:

*Algoritmo Acelerador Progresivo Convencional Matricial (APCM)*

$$e[n] = \mathbf{x}^T[n]\mathbf{w}[n-1] - d[n] \quad (6)$$

$$\mathbf{G}[n] = \mathbf{I}_M - \alpha^2 \mathbf{M}_1 \mathbf{x}[n] \mathbf{x}^T[n] + 2\alpha \mathbf{M}_1 (\mathbf{M}_2 + \mathbf{x}[n] \mathbf{x}^T[n] \mathbf{M}_1 \mathbf{M}_3) \quad (7)$$

$$\mathbf{q}[n] = (\mathbf{G}[n]^{-1}(\mathbf{q}[n-1] - \alpha \mathbf{M}_1 \mathbf{x}[n] e[n])) \quad (8)$$

$$\mathbf{w}[n] = \mathbf{w}[n-1] + \alpha \mathbf{q}[n] \quad (9)$$

donde los términos  $e[n]$  y  $\mathbf{w}[n]$  corresponden al error de medida y al vector de coeficientes del filtro;  $\mathbf{G}[n]$  y  $\mathbf{q}[n]$  son vectores auxiliares.

*Algoritmo Acelerador Regresivo Convencional Matricial (ARCM)*

$$e[n] = \mathbf{x}^T[n]\mathbf{w}[n-1] - d[n] \quad (10)$$

$$\mathbf{A} = (\mathbf{I}_M + 2\alpha \mathbf{M}_1 \mathbf{M}_2)^{-1} \quad (11)$$

$$\mathbf{B} = \alpha(\alpha \mathbf{I}_M + 2\mathbf{M}_1 \mathbf{M}_3) \mathbf{A} \mathbf{M}_1 \quad (12)$$

$$\mathbf{C}[n] = \frac{\alpha e[n] + \mathbf{x}^T[n] \mathbf{B} \mathbf{M}_1^{-1} \mathbf{q}[n]}{1 + \mathbf{x}^T[n] \mathbf{B} \mathbf{x}[n]} \mathbf{M}_1 \quad (13)$$

$$\mathbf{q}[n] = \mathbf{A}(\mathbf{q}[n-1] - \mathbf{C}[n] \mathbf{x}[n]) \quad (14)$$

$$\mathbf{w}[n] = \mathbf{w}[n-1] + \alpha \mathbf{q}[n] \quad (15)$$

nuevamente los términos  $e[n]$  y  $\mathbf{w}[n]$  corresponden al error de medida y al vector de coeficientes del filtro;  $\mathbf{I}_M$  es la matriz identidad y  $\mathbf{A}$ ,  $\mathbf{B}$  y  $\mathbf{C}[n]$  son vectores auxiliares.

El algoritmo APCM presenta una baja complejidad computacional, en contraste con el buen desempeño y la estabilidad del algoritmo ARCM. (Jojoa, 1999)

Con el objetivo de disminuir la complejidad computacional del algoritmo ARCM, se llegó a obtener el Algoritmo Acelerador Regresivo Versión  $\gamma$ , el cual es una versión simplificada del algoritmo ARCM, cuya principal característica es la disminución del número de parámetros de ajuste de cuatro ( $\alpha$ ,  $\mathbf{M}_1 = m_1 \mathbf{I}_M$ ,  $\mathbf{M}_2 = m_2 \mathbf{I}_M$  y  $\mathbf{M}_3 = m_3 \mathbf{I}_M$ ) a tres, denominados:  $\alpha$ ,  $\gamma$  y  $m_1$ , por medio de los cuales se logra una buena velocidad de convergencia y paralelamente una considerable reducción del error de medida final (Gerken, et al., 2000).

Las ecuaciones que describen el Algoritmo Acelerador Regresivo Versión  $\gamma$  son:

$$e[n] = \mathbf{x}^T[n] \mathbf{w}[n-1] - d[n] \quad (16)$$

$$g[n] = \frac{e[n] + \gamma \mathbf{x}^T[n] \mathbf{q}[n-1]}{1 + \alpha \gamma \mathbf{x}^T[n] \mathbf{M}_1 \mathbf{x}[n]} \quad (17)$$

$$\mathbf{q}[n] = \frac{\gamma}{\alpha + \gamma} \mathbf{q}[n-1] - \alpha g[n] \mathbf{M}_1 \mathbf{x}[n] \quad (18)$$

$$\mathbf{w}[n] = \mathbf{w}[n-1] + \alpha \mathbf{q}[n] \quad (19)$$

$d[n]$  corresponde a la señal deseada obtenida de la siguiente forma:

$$d[n] = \mathbf{x}^T[n] \mathbf{w}_o[n] + r(n) \quad (20)$$

Con:

$x[n]$ :	Vector de regresión (datos de entrada - salida)
$\mathbf{w}[n]$ :	Vector de coeficientes del filtro adaptativo.
$d[n]$ :	Escalar que corresponde a la señal deseada en el instante $n$ .
$e[n]$ :	Escalar que corresponde al error de medida en el instante $n$ .
$g[n]$ :	Escalar auxiliar en el instante $n$ .
$\mathbf{q}[n]$ :	Vector auxiliar.
$\mathbf{w}_o[n]$ :	Vector de coeficientes óptimo.
$\alpha$ :	Parámetro de ajuste fijo.
$\gamma$ :	Parámetro de ajuste fijo.
$m_1$ :	Parámetro de ajuste fijo.
$\mathbf{M}_1$ :	Matriz definida positiva de tal forma que $\mathbf{M}_1 = m_1 \mathbf{I}_M$

La ventaja que presenta este algoritmo es que los parámetros de ajuste se reducen a cantidades escalares ( $\alpha, \gamma$  y  $m_1$ ). De acuerdo al análisis de convergencia con respecto a la media (Jojoa, 2003), se establece que el algoritmo converge para valores de  $\alpha, \gamma$  y  $m_1$  positivos. Así mismo del análisis de tracking realizado, en ambientes no estacionarios, se determinó que éste presenta un mínimo error de desajuste cuando  $\alpha\gamma m_1 \approx H$  (Criterio de Mínimo Error) en que  $H$  es una constante real positiva con un valor aproximado a 2. En la Tabla 2.1, a continuación se indican algunos valores de  $H$  obtenidos de manera teórica y práctica para diversos valores de  $\alpha, \gamma$  y  $m_1$  con los que se alcanza un mínimo desajuste.

$m_1$	$\alpha$	$\gamma$	$H$
1	1.0	2.04	2.040
1	1.5	1.35	2.025
3	1.0	0.68	2.040
3	1.5	0.45	2.025
11	1.0	0.18	1.980
11	1.5	0.12	1.980

**Tabla 1. Valores de  $H$  para diferentes condiciones de  $\alpha, \gamma$  y  $m_1$  (Jojoa, 2003)**

## Anexo 2

# Sistema de Identificación implementado en Matlab

%programa para análisis de un identificador de sistema  
%uso de los algoritmos LMS, NLMS, RLS y AR\_gamma

```
%*****  
%CONDICIONES INICIALES  
%*****  
clear all;  
close all;  
N =1000;    %número de muestras  
M =2;      %orden del filtro adaptativo  
nM =M;     %orden del filtro a identificar  
n2 =400;   %posición inicial desde donde se calcula  
          %el error cuadrático medio  
n =1:N;  
K=1       %Número de experiencias  
  
%condiciones parámetro de ajuste algoritmo LMS  
mu=.1;  
  
%condiciones parámetro de ajuste algoritmo AR_gamma  
m1=2;  
alfa=0.6;  
gama=2/(m1*alfa);  
  
%valor inicial de los coeficientes para los algoritmos  
HE=zeros(M,1);
```

---

```

%condiciones para las señales aleatorias
sig=1;
sig2=0.05;

%otras inicializaciones
% estos variables almacenaran los valores
% cuadráticos de los errores medidos en cada
% experiencia
sel2=0;
sea2=0;
% *****
%SISTEMA QUE SE DESEA IDENTIFICAR
% *****
B=ones([1 nM]);%[-1 1]
A=1;
wo=B';

for k=1:K    %contador de n i 1/2 mero de experiencias

% *****
%SEÑAL DE ENTRADA (SEÑALES ALEATORIAS)
% *****
%senal de entrada
    x=sig*randn([N 1]);    %ruido blanco con desviaci i 1/2 n estandar=sig

%ruido de medida (se i 1/2 al gaussiana de variancia sig2^2
    v2 = sig2*randn(size(x));

% *****
%SEÑAL ENTREGADA POR EL SISTEMA A IDENTIFICAR
%EN ESTE CASO (IDENTIFICADOR) ES LA SEÑAL DESEADA
% *****
    yf=filter(B,A,x);
% d=yf;    %señal deseada sin ruido de medida
    d=yf+v2;    %señal deseada con ruido de medida

% *****
%REPUESTA ENTREGADA POR LOS ALGORITMOS ADAPTATIVOS:
%ERROR Y COEFICIENTES DEL FILTRO
% *****

[e_lms,w_lms]=lms(x,d,mu,M,HE);    %algoritmo LMS
[e_arg,w_arg,g]=alacel(x,d,alfa,gama,m1,M,HE);    %algoritmo AR_gamma

%cálculo del error cuadrático medio de la experiencia
    el2=e_lms.^2;

```

---

```
ea2=e_arg.^2;

%almacenamiento temporal del error cuadrático medio de las
%experiencias realizadas hasta ahora
    sel2=sel2+el2;
    sea2=sea2+ea2;
end %fin de la experiencia

%cálculo da media do erro cuadrático
%para obtener el error cuadrático medio final
mel2=sel2/K;
mea2=sea2/K;

%valor medio del error cuadrático medio en la región de
%convergencia
eqm_lms=mean(mel2(n2:N))
eqm_arg=mean(mea2(n2:N))

%grafica de los coeficientes
f1=plot(n,w_lms,'r',n,w_arg,'b');
    set(gca,'FontSize',14);
    set(gca,'Box','on');
legend([f1(1),f1(M+1)],'LMS','AR \gamma')
xlabel('n');
ylabel('p');
grid;

figure,
plot(n,g)
title('g')
axis([0 200 -0.5 0.5])
```

---

```
%LMS Algoritmo Least Mean Square
%
% versão simple
%
% [e,h]=lms(x,d,mu,M,HE)
%
% parâmetros entrada:
% x = sequencia de entrada
% d = sequencia desejada
% mu = passo de adaptação
% M = número de coeficientes
% HE = condição inicial dos coeficientes
%
% parâmetros de saída
% e = erro
% h = coeficientes do filtro LMS
```

```
function [epri,HE1]=lms(xx,dd,mu,M,HE);
```

```
N = length(xx);
XE=zeros(M,1);
```

```
for n=1:N,
    HE1(:,n) = HE;
    XE    = [xx(n);XE(1:M-1)];
    epri(n)= dd(n)-XE'*HE;
    HE    = HE+mu*epri(n)*XE;
end;
```

---

```
% alacel este..
%
% [e,w]=alacel(x,d,al,ga,m1,w1)
%
%
function [e,w,G]=alacel(x,d,al,ga,m1,M,w1)
N=length(x);
X=zeros(M,1);
q1=zeros(M,1);

for n=1:N
    X=[x(n);X(1:M-1)];
    e(n)=X'*w1-d(n);
    g=(e(n)+ga*X'*q1)/(1+al*ga*m1*X'*X);
    q=(ga/(al+ga))*(q1-al*g*m1*X);
    w(:,n)=w1+al*q;
    %actualizaciones
    q1=q;
    w1=w(:,n);
    G(n)=g;
end
```

---



## Anexo 3

### Implementación en Simulink

#### Caso 1. Caso de estudio 1: Sistema de Identificación usando un Filtro de Respuesta Finita al Impulso (FIR)

```
clc;
clear all;
close all;

% Call to simulink
NSamples    = 200;
Fixed_Point = 1;
if Fixed_Point == 1,
    % Fixed point
    WordLength    = 64;
    FractionLength = 60;
    DATA         = fixdt(1,WordLength,FractionLength);
else
    DATA         = 'double';
end
end
RNoise      = 0.001;
%VAlpha     = 0.0030;
%m1         = 13.2;
%VGamma     = 0.6;
m1          = 11;
VAlpha      = 1.5;
%VGamma     = 2/(m1*VAlpha);
%m1         = 3;
```

---

```

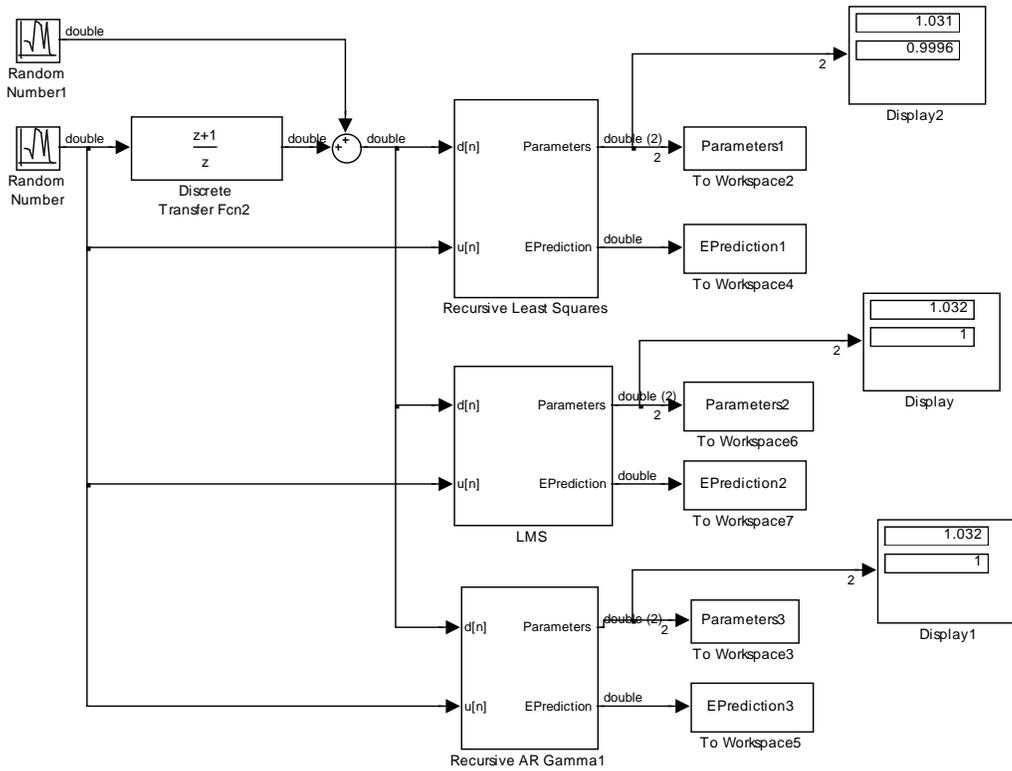
%VAlpha    = 1;
VGamma     = 0.12;
sim('recursive_ID_FIR',NSamples);

%% Plot
NMIn       = 1;
for i = 1:2
    subplot(2,1,i);
    plot(Parameters1.time(NMIn:end),Parameters1.signals.values(NMIn:end,i),'b-',...
        Parameters2.time(NMIn:end),Parameters2.signals.values(NMIn:end,i),'r-',...
        Parameters3.time(NMIn:end),Parameters3.signals.values(NMIn:end,i),'k-');
    grid on;
    legend('RLS','LMS','AR \gamma','Location','SouthEast');
    xlabel('Samples');
    ylabel(sprintf('b%d',i));
    title(sprintf('Q.1.%d.%d, var ruido = %f',WordLength-
FractionLength,FractionLength,RNoise));
end

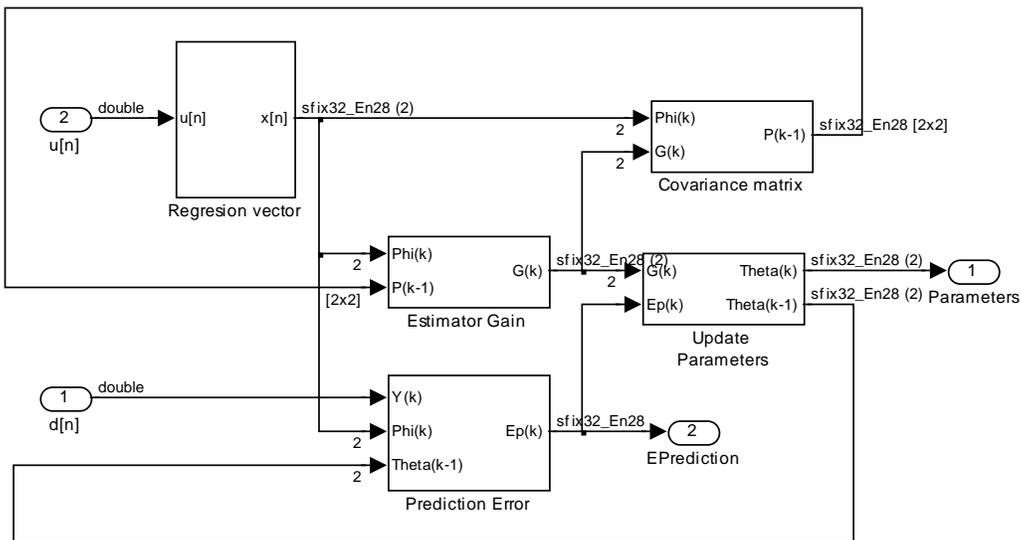
%% Varianza del error de prediccion
Var1       = var(EPrediction1.signals.values(2*NMIn:end,1));
Mean1      = abs(mean(EPrediction1.signals.values(2*NMIn:end,1)));
Var2       = var(EPrediction2.signals.values(2*NMIn:end,1));
Mean2      = abs(mean(EPrediction2.signals.values(2*NMIn:end,1)));
Var3       = var(EPrediction3.signals.values(2*NMIn:end,1));
Mean3      = abs(mean(EPrediction3.signals.values(2*NMIn:end,1)));
fprintf('Media RLS: %e\tVarianza RLS: = %e\n',Mean1,Var1);
fprintf('Media LMS: %e\tVarianza LMS: = %e\n',Mean2,Var2);
fprintf('Media ARG: %e\tVarianza ARG: = %e\n',Mean3,Var3);

```

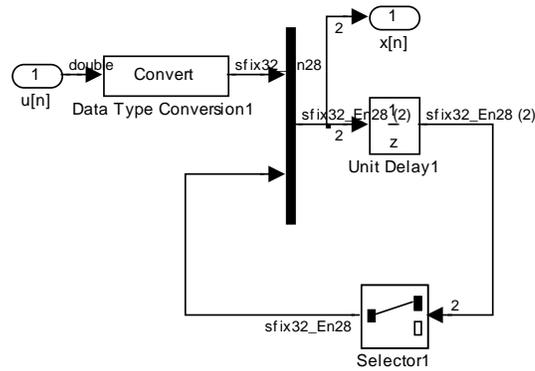
---



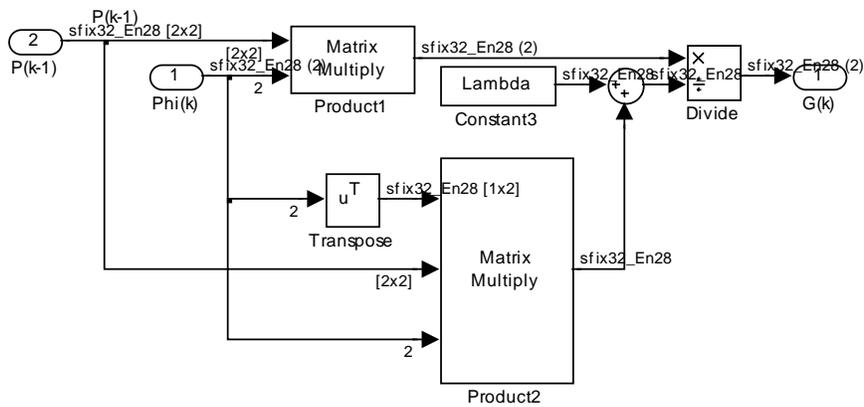
**a) Algoritmo RLS**



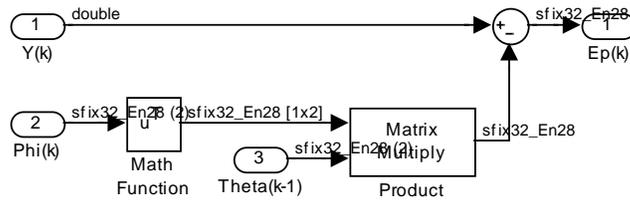
### Vector Regresor



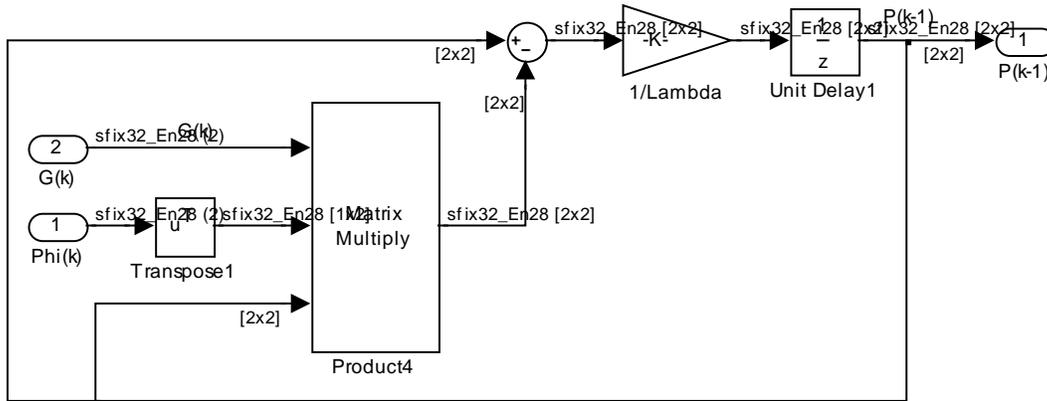
### Estimador de ganancia



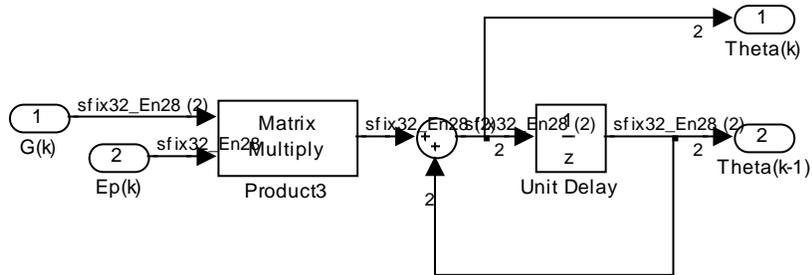
### Error de Predicción



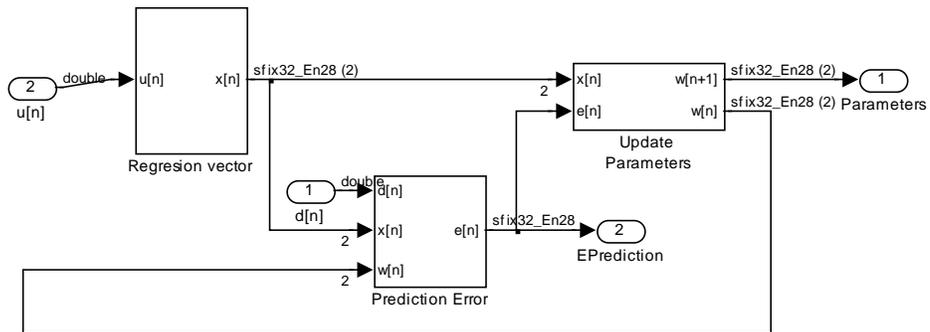
Matriz de Covarianza



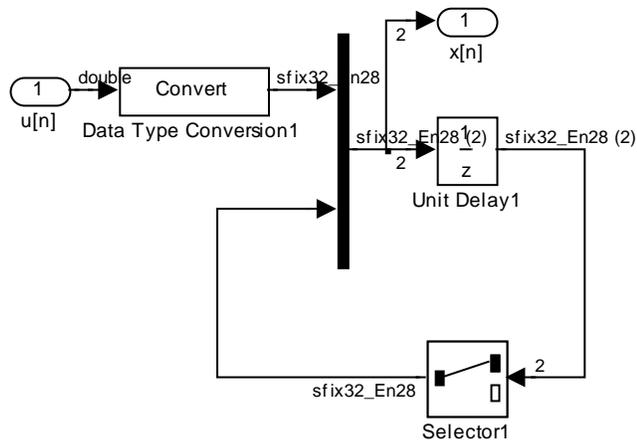
Actualización de parámetros



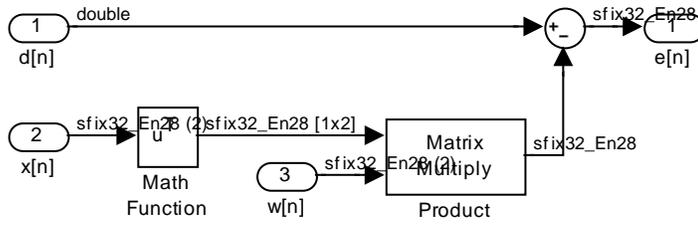
## b) Algoritmo LMS



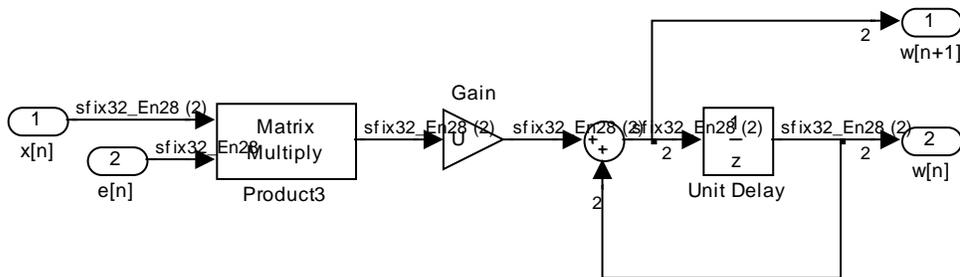
## Vector Regresor



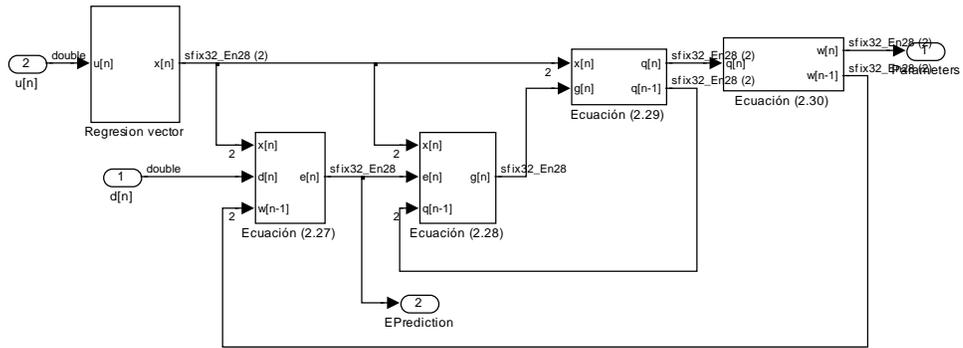
## Error de predicción



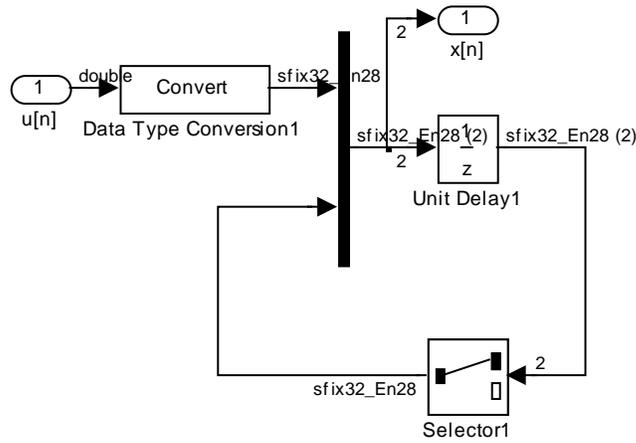
## Actualización de Parámetros



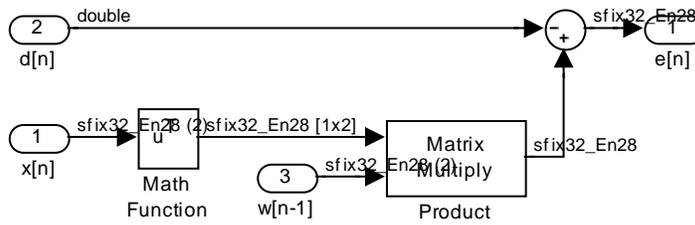
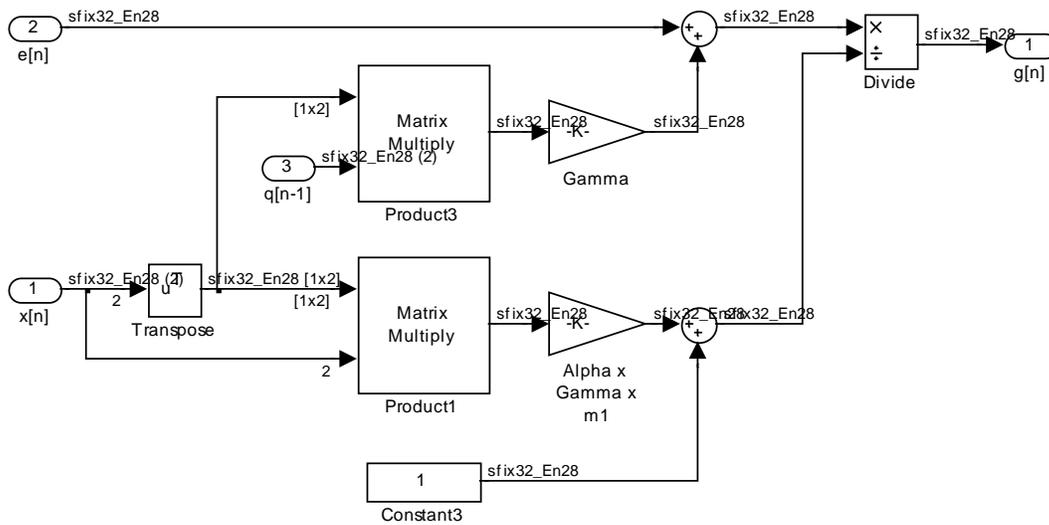
### c) Algoritmo Acelerador Regresivo



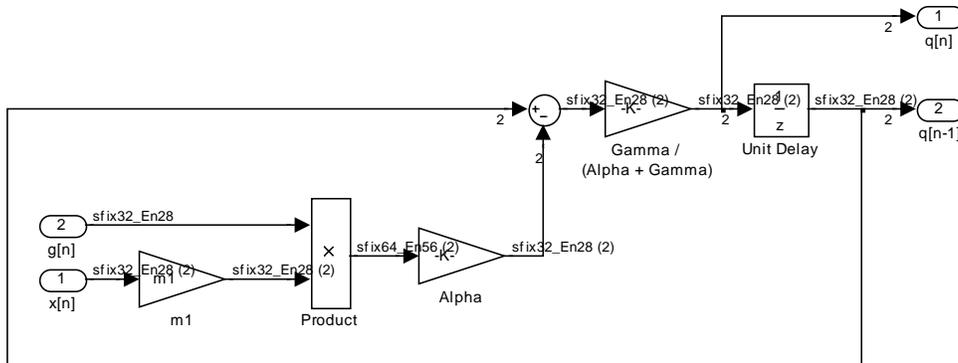
### Vector Regresor



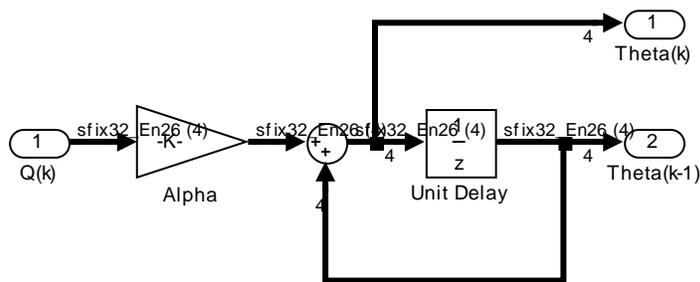
## Error de predicción

 $g(n)$ 

q(n)



Actualización de Parámetros



### 3.1.2 Caso de estudio 2: Sistema de Identificación usando un Filtro de Respuesta Infinita al Impulso (IIR)

```

clc;
clear all;
close all;

% Call to simulink
NSamples    = 10E3;
Fixed_Point = 1;
if Fixed_Point == 1,
    % Fixed point
    WordLength    = 64;
    FractionLength = 58;
    DATA         = fixdt(1,WordLength,FractionLength);
else
    DATA         = 'double';
end
RNoise      = 0.0;
VAlpha      = 0.0030;
m1          = 13.2;
VGamma      = 0.6;
%m1         = 2;
%VAlpha     = 0.6;
%VGamma     = 2/(m1*VAlpha);
sim('recursive_ID_IIR',NSamples);

%% Plot
NMIn        = 1000;
for i = 1:4
    subplot(2,2,i);
    plot(Parameters1.time(NMIn:end),Parameters1.signals.values(NMIn:end,i),'b-',...
        Parameters2.time(NMIn:end),Parameters2.signals.values(NMIn:end,i),'r-',...
        Parameters3.time(NMIn:end),Parameters3.signals.values(NMIn:end,i),'k-');
    grid on;
    legend('RLS','LMS','AR \gamma','Location','SouthEast');
    xlabel('Samples');
    switch i
        case 1
            ylabel('a1');
        case 2
            ylabel('a2');
        case 3
            ylabel('b1');
        case 4
            ylabel('b2');
    end
end

```

```

end
if Fixed_Point == 0,
    title(sprintf('double - var ruido = %f',RNoise));
else
    title(sprintf('Q.1.%d.%d, var ruido = %f',WordLength-
FractionLength,FractionLength,RNoise));
end
end
end

```

```

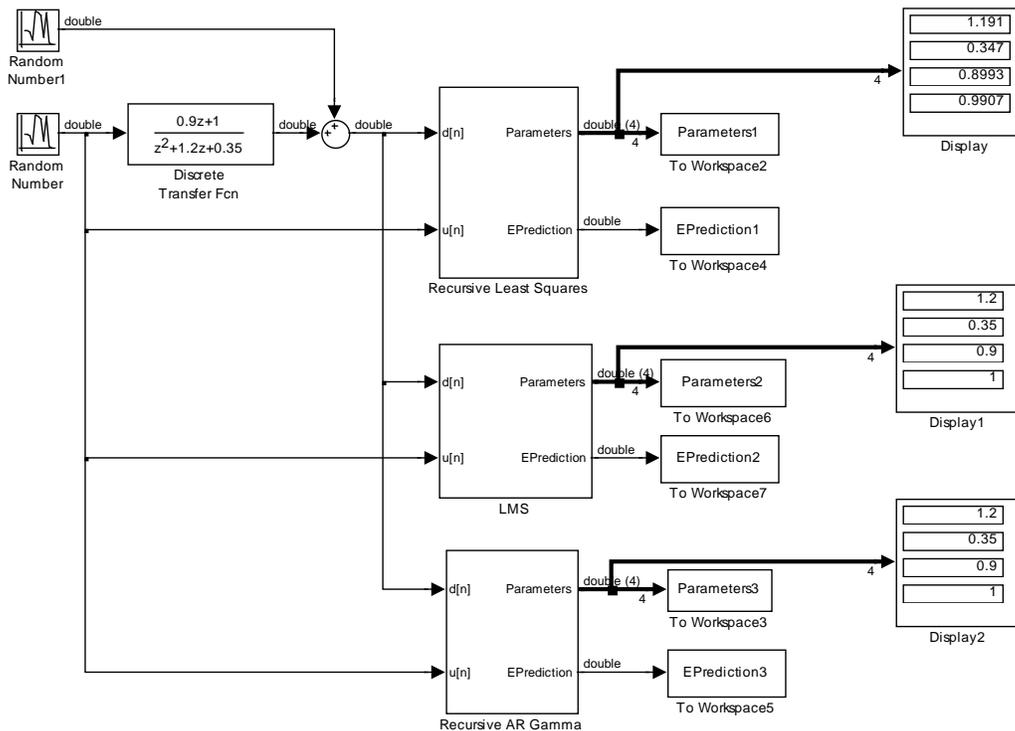
%% Varianza del error de prediccion

```

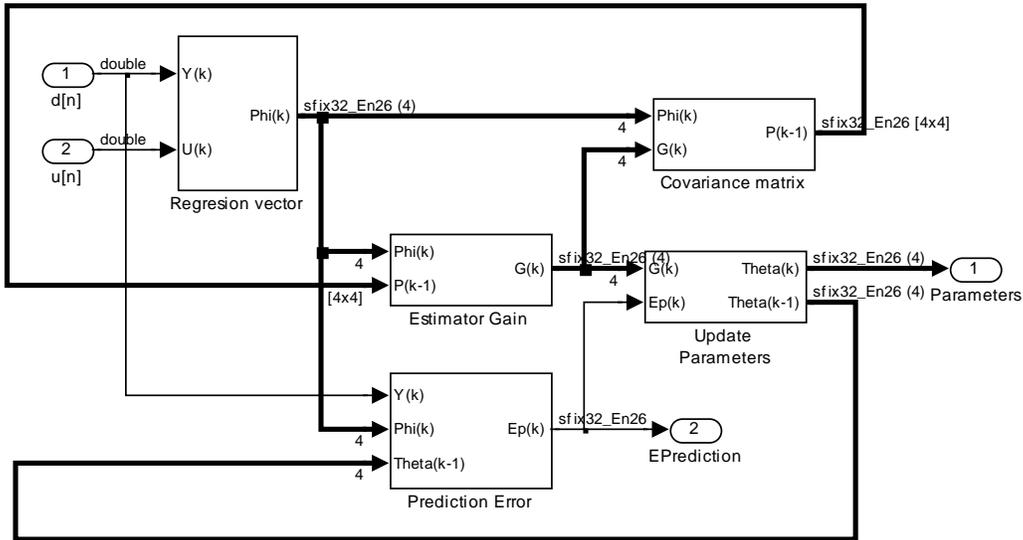
```

Var1      = var(EPrediction1.signals.values(2*NMIIn:end,1));
Mean1     = abs(mean(EPrediction1.signals.values(2*NMIIn:end,1)));
Var2      = var(EPrediction2.signals.values(2*NMIIn:end,1));
Mean2     = abs(mean(EPrediction2.signals.values(2*NMIIn:end,1)));
Var3      = var(EPrediction3.signals.values(2*NMIIn:end,1));
Mean3     = abs(mean(EPrediction3.signals.values(2*NMIIn:end,1)));
fprintf('Media RLS: %e\tVarianza RLS: = %e\n',Mean1,Var1);
fprintf('Media LMS: %e\tVarianza LMS: = %e\n',Mean2,Var2);
fprintf('Media ARG: %e\tVarianza ARG: = %e\n',Mean3,Var3);

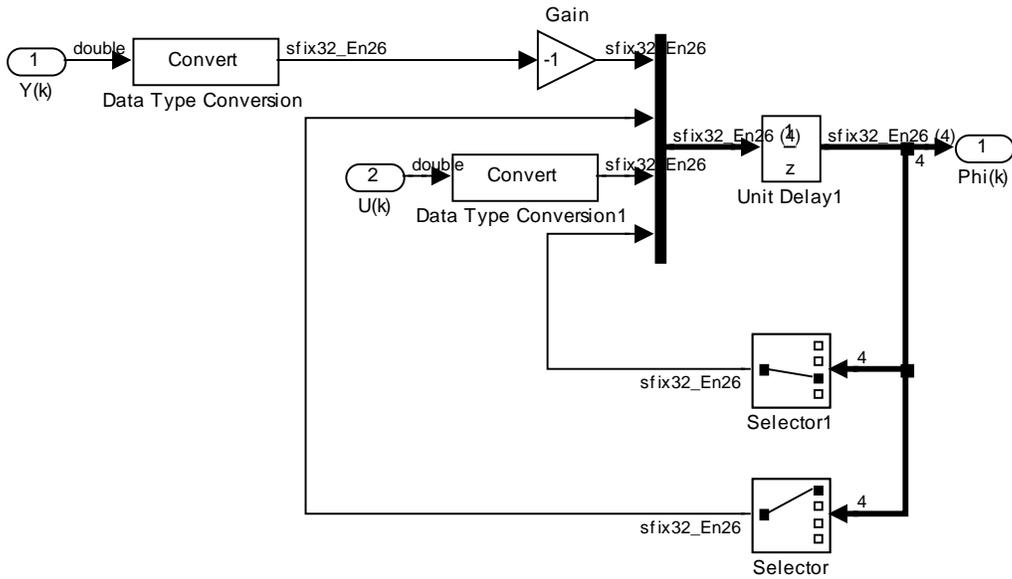
```



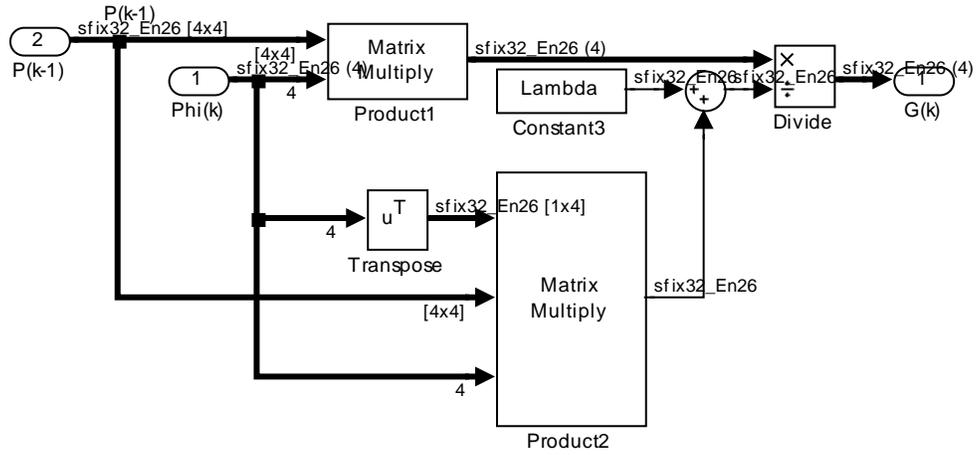
a) Algoritmo RLS



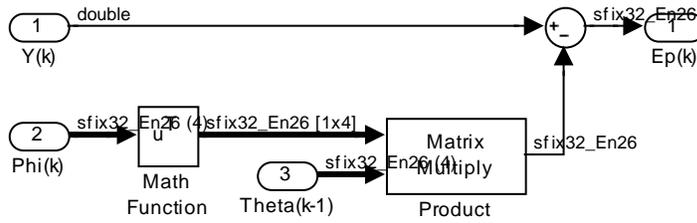
Vector de Regresion



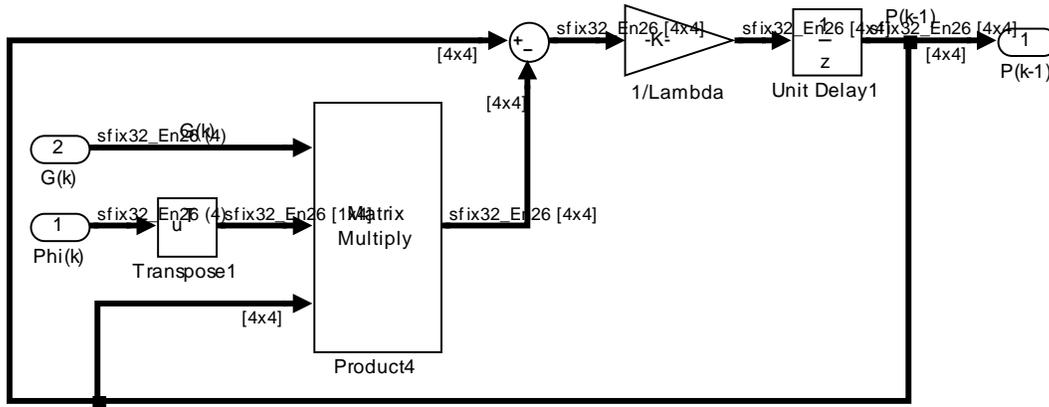
Estimador de Ganancia



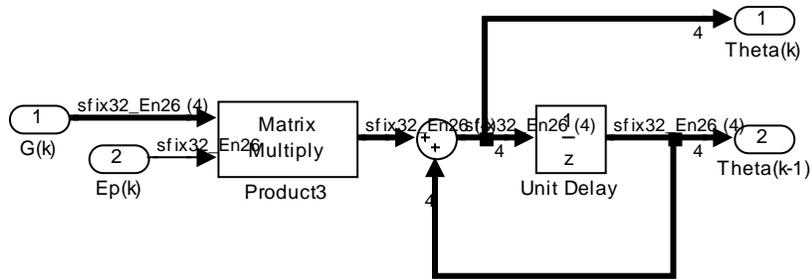
Error de Predicción



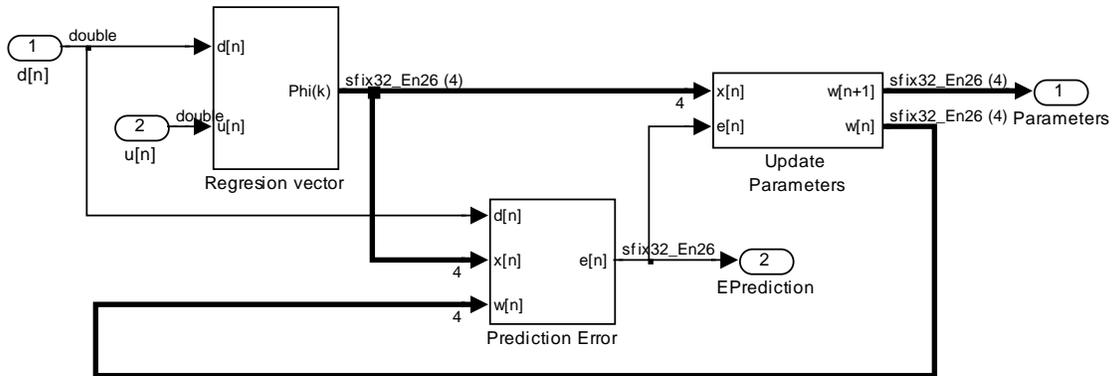
Covarianza



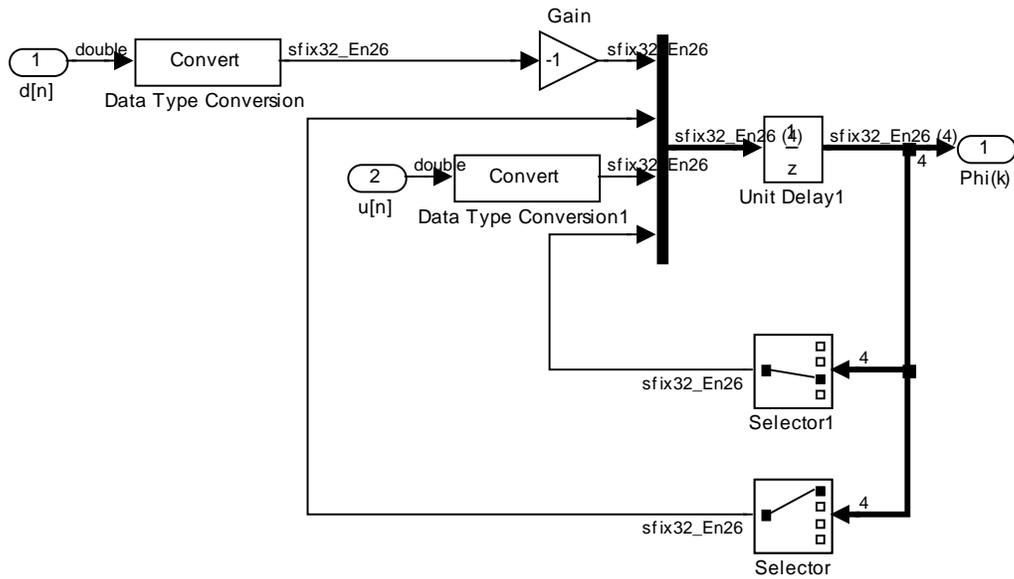
Actualización de parámetros



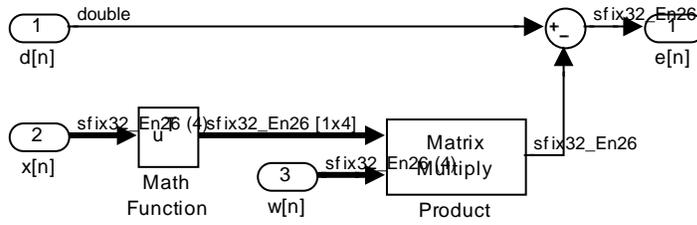
**b) Algoritmo LMS**



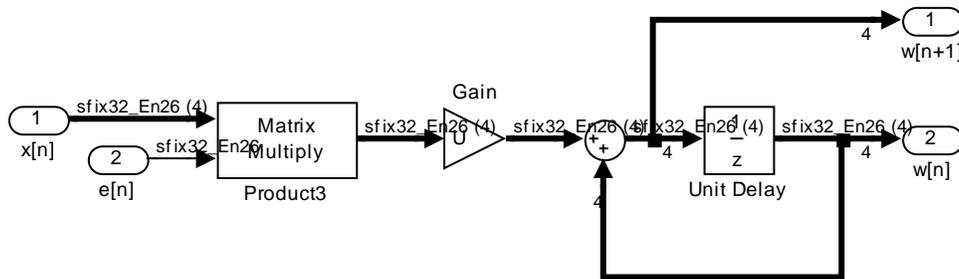
**Vector Regresor**



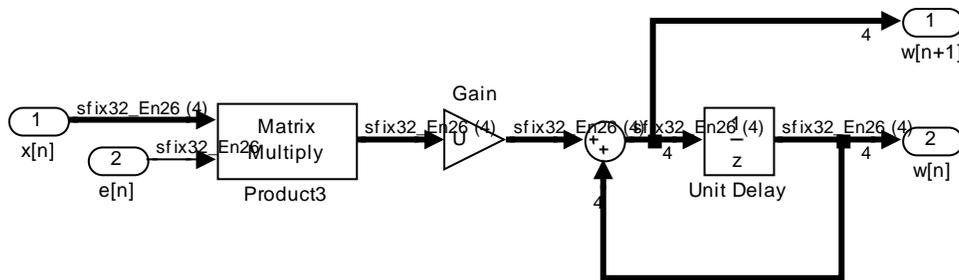
Error de Predicción



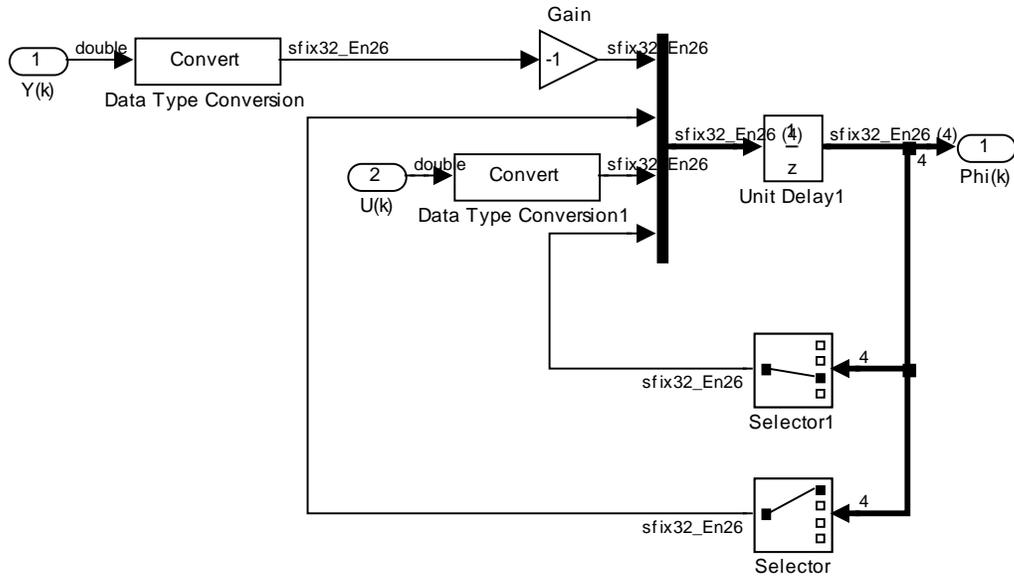
Actualización de parámetros



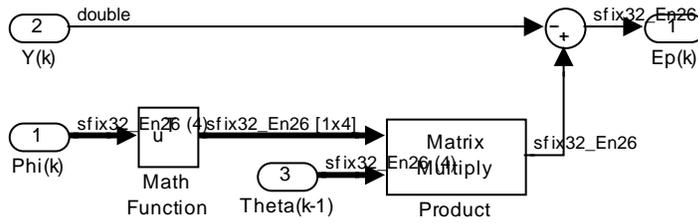
c) Algoritmo Acelerador Regresivo



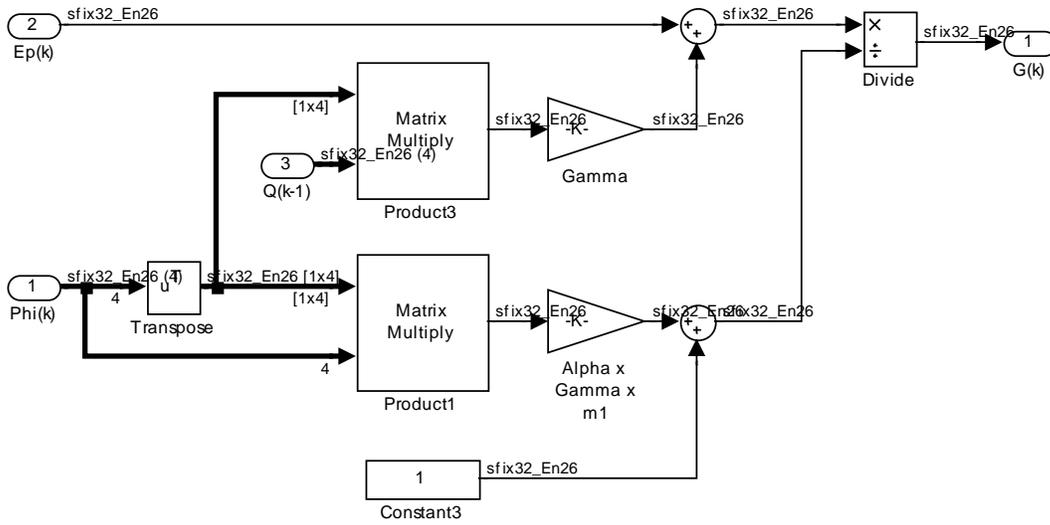
Vector Regresor



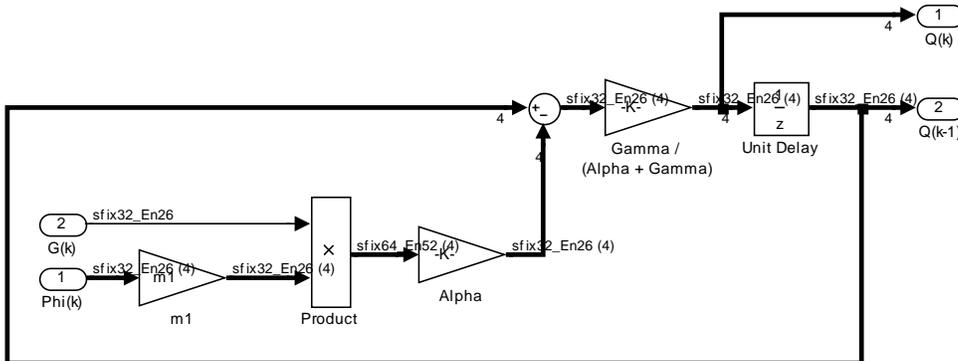
Error de Predicción



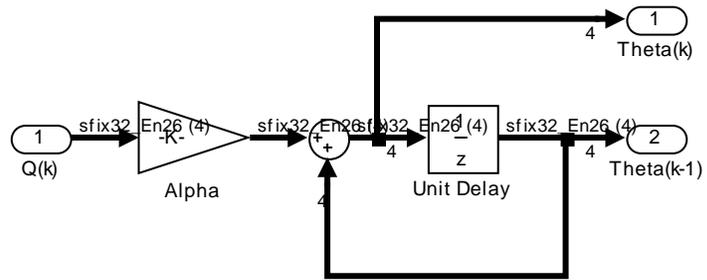
g(n)



q(n)



## Actualización de parámetros



## Anexo 4

### Parámetros Óptimos

```
clc;
clear all;
close all;

%% Verificacion
Fixed_Point = 1;
if Fixed_Point == 1,
    % Fixed point (Signed, WordLength = 16, FractionLength = 12)
    WordLength = 64;
    FractionLength = 56;
    DATA = fixdt(1,WordLength,FractionLength);
else
    DATA = 'double';
end

% Optimizacion
Current_Params = 4;
switch Current_Params
    case 1
        Test_VAlpha = 11;
        Test_m1 = 1.5;
        Test_VGamma = 0.12;
    case 2
        Test_m1 = 2;
        Test_VAlpha = 0.6;
        Test_VGamma = 2/(Test_m1*Test_VAlpha);
    case 3
        % Best params
        Test_m1 = 2;
```

---

```

    Test_VAlpha = 1;
    Test_VGamma = 2/(Test_m1*Test_VAlpha);
case 4
    % Equal params
    Test_m1 = nthroot(2,3);
    Test_VAlpha = nthroot(2,3);
    Test_VGamma = nthroot(2,3);
end
X0 = [Test_VAlpha; Test_m1; Test_VGamma];
[X,FX] = fminunc(@(X) criterio_optimizacion(X,DATA), X0);

NSamples = 10E3;
RNoise = 0.0;

% Parametros optimos
VAlpha = X(1);
m1 = X(2);
VGamma = X(3);
sim('recursive_IIR',NSamples);
Parameters1 = Parameters;
EPrediction1= EPrediction;

% Parametros originales
VAlpha = Test_VAlpha;
m1 = Test_m1;
VGamma = Test_VGamma;
sim('recursive_IIR',NSamples);
Parameters2 = Parameters;
EPrediction2= EPrediction;

% Plot
NMIn = 4000;
for i = 1:4
    subplot(2,2,i);
    plot(Parameters1.time(NMIn:end),Parameters1.signals.values(NMIn:end,i),'b.',...
        Parameters2.time(NMIn:end),Parameters2.signals.values(NMIn:end,i),'r-');
    grid on;
    legend('Opt','Iniciales','Location','SouthEast');
    xlabel('Samples');
    ylabel(sprintf('Parameter %d',i));
    title('32 Bits - sin ruido');
end

% Varianza del error de prediccion
Var1 = var(EPrediction1.signals.values(NMIn:end,1));
Var2 = var(EPrediction2.signals.values(NMIn:end,1));
fprintf('Varianza Optimos: = %e\n',Var1);

```

---

```
fprintf('Varianza Iniciales: = %e\n',Var2);

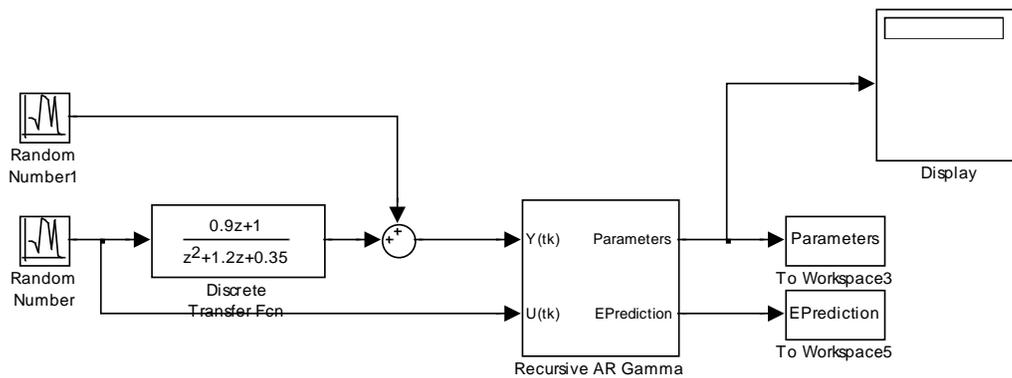
%% Comparacion de parametros
fprintf('Parametro\t Original\t Optimizacion\n');
fprintf('VAlpha\t\t %f\t\t %f\n',Test_VAlpha,X(1));
fprintf('m1\t\t\t %f\t\t %f\n',Test_m1,X(2));
fprintf('VGamma\t\t %f\t\t %f\n',Test_VGamma,X(3));
```

---

```

function [Eval] = criterio_optimizacion(Parametros,DATA)
    NSamples = 10E3;
    RNoise = 0.0;
    VAlpha = Parametros(1);
    m1 = Parametros(2);
    VGamma = Parametros(3);
    Options = simset('SrcWorkspace','current','DstWorkspace','current');
    sim('recursive_IIR',NSamples,Options);
    NMIN = 4000;
    Eval = var(EPrediction.signals.values(NMIN:end));
end

```



## Anexo 5

### Cuantización\_argamma

```
clc;
clear all;
close all;

% Call to simulink
NSamples    = 10E3;
Fixed_Point = 0;
DATA        = 'double';

% Fixed point (Signed, WordLength = 16, FractionLength = 12)
WordLength  = 32;

FractionLength = round(0.75*WordLength);
DATA_INTEGER   = fixdt(1,WordLength,FractionLength);
RNoise         = 0.00;
VAlpha         = 0.0030;
m1             = 13.2;
VGamma         = 0.6;
sim('recursive_ID_IIR',NSamples);

%% Plot
NMIn          = 1;
for i = 1:4
    subplot(2,2,i);
    plot(Parameters1.time(NMIn:end),Parameters1.signals.values(NMIn:end,i),'b-',...
         Parameters2.time(NMIn:end),Parameters2.signals.values(NMIn:end,i),'r-');
    grid on;
    legend('Double',sprintf('fix%d',WordLength),'Location','SouthEast');
    xlabel('Samples');
```

---

```

    ylabel(sprintf('Parameter %d',i));
end

%% Error de cuantizacion debido a la division
close all;
NMIIn      = 4E3;
Time       = Division1.time(NMIIn:end) - Division1.time(NMIIn-1);
Error_Division = Division1.signals.values(NMIIn:end,1)-
Division2.signals.values(NMIIn:end,1);
figure;
plot(Time,Error_Division);
title('Division');
xlabel('Samples');
ylabel('Error de cuantizacion');

% Funcion de autocorrelacion del error de prediccion
figure;
[X,Lags]    = xcorr(Error_Division,25,'coeff');
stem(Lags,X);
title('Autocorrelacion del error debido a la division');
xlabel('Rezago');
ylabel('Normalizada');
grid on;

%% Varianza del error de prediccion
Mean1      = mean(EPrediction1.signals.values(NMIIn:end,1));
Mean2      = mean(EPrediction2.signals.values(NMIIn:end,1));
Var1       = var(EPrediction1.signals.values(NMIIn:end,1));
Var2       = var(EPrediction2.signals.values(NMIIn:end,1));
fprintf('MEdias y varianzas del error de prediccion en PFijo y PFlotante\n');
fprintf('Media float64:   = %e\n',Mean1);
fprintf('Media fix%d:     = %e\n',WordLength,Mean2);
fprintf('Varianza float64:  = %e\n',Var1);
fprintf('Varianza fix%d:    = %e\n',WordLength,Var2);

```

---

