



UNIVERSIDAD  
NACIONAL  
AUTÓNOMA DE MÉXICO

**UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO**

---

---

**FACULTAD DE INGENIERÍA**

# TESIS

Para obtener el Título de Ingeniero Eléctrico  
Electrónico

**“PRUEBAS EXPERIMENTALES CON EL ALGORITMO  
SUBOPTIMAL DE MODOS DESLIZANTES EN UN MOTOR DE  
CORRIENTE DIRECTA”**



Presentado por:

Dávila Mérida Israel Alejandro

Director: Ing. Rafael Iriarte Balderrama

JUNIO 2006



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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



---

*Agradecimientos...*

L. Fridman y R. Iriarte hacen reconocimiento al Consejo Nacional de la Ciencia y la Tecnología (**CONACYT**) del apoyo financiero otorgado para el desarrollo del presente trabajo, en la concesión **no. 43807-Y**, de igual manera a la Universidad Nacional Autónoma de México (**UNAM**), por el apoyo a través de **PAPPIT 107006-2** y **PAPIME 101705**.



---

*Agradecimientos Especiales...*

*A la memoria de mi Madre y de mi Abue... por su infinito amor y por la herencia más valiosa que recibí de su parte: educación.*

*A mi familia... por creer en mí y por su constante e incondicional apoyo, recordando los momentos de felicidad y el cariño que hemos compartido.*

*A todos los demás seres queridos que me rodean y que forman parte de mi felicidad*

*A mi Facultad por su cobijo intelectual*



# ÍNDICE

## ÍNDICE DE FIGURAS INTRODUCCIÓN

<i>Presentación</i> .....	i
<i>Motivación</i> .....	iii
<i>Formulación del problema y contribuciones</i> .....	iii
<i>Objetivo general</i> .....	iv
<i>Objetivos específicos</i> .....	iv
<i>Contenido de la tesis</i> .....	v

## CAPITULO 1 CONTROL DE MODOS DESLIZANTES

<i>1.0.- Introducción</i> .....	1
<i>1.1.- Modos Deslizantes</i> .....	2
<i>1.1.1.- Concepto</i> .....	2
<i>1.1.2.- Control en Modo Deslizante</i> .....	3
<i>1.2.- Modos Deslizantes de Primer Orden (FOSM)</i> .....	3
<i>1.2.1.- Existencia del Modo Deslizante</i> .....	5
<i>1.3.- Consideraciones Generales</i> .....	5
<i>1.4.- El problema del Chattering</i> .....	7

## CAPITULO 2 MODOS DESLIZANTES DE ORDEN SUPERIOR (HOSM).

<i>2.0.- Introducción</i> .....	9
<i>2.1.- Concepto</i> .....	9
<i>2.2.- Algoritmo de Suboptimal de Modos deslizantes de segundo orden (2-SMC)</i> .....	11
<i>2.2.1.- Concepto</i> .....	11
<i>2.2.2.- Características Principales</i> .....	14

**CAPITULO 3 DIFERENCIADORES**

<i>3.0.- Introducción.....</i>	17
<i>3.1.-Análisis de los diferenciadores propuestos.....</i>	18
<i>3.1.1.-Diferenciador SM.....</i>	18
<i>3.1.2.-Diferenciador de Levant.....</i>	19
<i>3.1.3.-Diferenciador de Shtessel.....</i>	21
<i>3.2.-Simulaciones.....</i>	22
<i>3.2.1.-Diferenciador de Simulink para una señal cuadrada.....</i>	23
<i>3.2.2.-Diferenciador por Filtro Lineal para una señal cuadrada.....</i>	24
<i>3.2.3.-Diferenciador SM para una señal cuadrada.....</i>	24
<i>3.2.4.-Diferenciador de Levant para una señal cuadrada.....</i>	25
<i>3.2.5.-Diferenciador de Shtessel para una señal cuadrada.....</i>	26

**CAPITULO 4 SIMULACIONES EN EL MOTOR DE CD.**

<i>4.0.- Introducción.....</i>	28
<i>4.1.- El Motor de CD.....</i>	29
<i>4.2.-Control de velocidad con el algoritmo FOSM.....</i>	33
<i>4.2.1.-Análisis de los resultados obtenidos.....</i>	38
<i>4.2.2.-Respuesta ante una perturbación Senoidal.....</i>	40
<i>4.3.-Control de velocidad con el algoritmo Suboptimal.....</i>	43
<i>4.3.1.-Análisis de los resultados obtenidos.....</i>	48
<i>4.3.2.-Respuesta frente a una perturbación senoidal.....</i>	50
<i>4.4.-Control Suboptimal como filtro.....</i>	53
<i>4.5.-Control de velocidad con el algoritmo PID.....</i>	55
<i>4.5.1.-Respuesta ante una perturbación senoidal.....</i>	60

**CAPITULO 5 CONTROL DE VELOCIDAD EN TIEMPO REAL.**

<i>5.1.-Introducción.....</i>	63
<i>5.2.-Generalidades.....</i>	65
<i>5.3.-Sintonizado de parámetros interactivo en tiempo real.....</i>	67
<i>5.4.-Programación desde Matlab/Simulink.....</i>	68
<i>5.5.- Pruebas con el motor de CD.....</i>	70
<i>5.5.1.- Control de velocidad con el algoritmo FOSM .....</i>	71
<i>5.5.2.- Control de velocidad con el algoritmo Suboptimal.....</i>	75
<i>5.5.3.- Control de velocidad con el algoritmo Suboptimal como filtro.....</i>	79
<i>5.5.4.- Control de velocidad con el control PID.....</i>	80



<b>CONCLUSIONES</b> .....	85
<b>ANEXOS</b>	
<i>Control Proporcional Integral Derivativo</i> .....	91
<i>Introducción</i> .....	91
<i>Ajuste del PID</i> .....	94
<i>Tarjeta de Control dSPACE -DS 1103</i> .....	96
<i>Señales PWM (Pulse Width Modulation)</i> .....	97
<i>Encoder Incremental</i> .....	98
<i>Amplificador de Potencia</i> .....	100
<b>BIBLIOGRAFÍA.</b> ....	102



# ÍNDICE DE FIGURAS

	Pág.
<i>Figura 1.1</i> Trayectoria en el plano de fase de una estructura variable de segundo orden.....	2
<i>Figura 1.2</i> Trayectoria del FOSM en el espacio de estados.....	4
<i>Figura 1.3</i> Diagrama en Simulink del FOSM.....	6
<i>Figura 1.4</i> Representación del Fenómeno chattering de un sistema controlado por modos deslizantes.....	7
<i>Figura 1.5</i> Acción de los controles $u^+$ y $u^-$ .....	8
<i>Figura 2.1</i> Trayectoria de un Modo deslizante de segundo Orden (2-SM).....	10
<i>Figura 2.2</i> Trayectoria de Fase del algoritmo Suboptimal.....	12
<i>Figura 2.3</i> Representación del algoritmo Suboptimal como histéresis negativa.....	13
<i>Figura 2.4</i> Estructura interna del bloque Suboptimal en Simulink.....	14
<i>Figura 2.5</i> Estructura del algoritmo Suboptimal en Simulink.....	15
<i>Figura 3.1</i> Diagrama en Simulink del Diferenciador SM.....	19
<i>Figura 3.2</i> Diagrama en Simulink del Diferenciador de Levant.....	20
<i>Figura 3.3</i> Diagrama en Simulink del Diferenciador de Shtessel.....	21
<i>Figura 3.4</i> Diagrama en Simulink de los Diferentes Diferenciadores: a) diferenciador de simulink, b) diferenciador por filtro lineal, c) diferenciador SM, d) diferenciador de Levant e) diferenciador de Shtessel.....	22
<i>Figura 3.5</i> Señal aplicada por el Generador de Pulsos de Simulink.....	23
<i>Figura 3.6</i> Señal Diferenciada por Derivador de Simulink.....	23
<i>Figura 3.7</i> Señal Recuperada Derivador $1/0.001s + 1$ .....	24
<i>Figura 3.8</i> Señal Diferenciada por Derivador $1/0.001s + 1$ .....	24
<i>Figura 3.9</i> Señal Entrada Recuperada por SM.....	25
<i>Figura 3.10</i> Señal Diferenciada por Derivador SM.....	25
<i>Figura 3.11</i> Señal Diferenciada por Derivador de Levant.....	26
<i>Figura 3.12</i> Señal Entrada Recuperada por Levant.....	26
<i>Figura 3.13</i> Señal Diferenciada por Derivador Shtessel.....	27
<i>Figura 3.14</i> Señal Entrada Recuperada por Shtessel.....	27
<i>Figura 4.0</i> Lazo de Control para el Motor de CD.....	29
<i>Figura 4.1</i> Motor de CD.....	31
<i>Figura 4.2</i> Diagrama en Simulink del Motor de CD en lazo abierto.....	32
<i>Figura 4.3</i> Respuesta en lazo abierto del Motor de CD con una entrada de 24 V.....	33
<i>Figura 4.4</i> Lazo de control en Simulink del controlador FOSM.....	33
<i>Figura 4.5</i> Respuesta del lazo de control con el FOSM diferenciador de Simulink.....	34
<i>Figura 4.6</i> Respuesta del Control FOSM con diferenciador de Simulink.....	34
<i>Figura 4.7</i> Trayectoria del error en el espacio de estados.....	35
<i>Figura 4.8</i> Espacio de Estados del control FOSM.....	35
<i>Figura 4.9</i> Respuesta del FOSM con diferenciador SM.....	35
<i>Figura 4.10</i> Trayectoria del error en el espacio de estados con diferenciador SM.....	36
<i>Figura 4.11</i> Espacio de estados del control FOSM con diferenciador SM.....	36
<i>Figura 4.12</i> Respuesta del FOSM con diferenciador de Levant.....	36
<i>Figura 4.13</i> Trayectoria del error en el espacio de estados con diferenciador de Levant.....	36
<i>Figura 4.14</i> Espacio de estados del control FOSM con diferenciador de Levant.....	37
<i>Figura 4.15</i> Respuesta del FOSM con diferenciador de Shtessel.....	37
<i>Figura 4.16</i> Trayectoria del error en el espacio de estados con diferenciador de Shtessel.....	37
<i>Figura 4.17</i> Espacio de estados del control FOSM con diferenciador de Shtessel.....	38
<i>Figura 4.18</i> Control de velocidad con el algoritmo FOSM frente a perturbaciones.....	41
<i>Figura 4.19</i> Respuesta del lazo de control con el FOSM frente a la perturbación del Amplificador.....	42
<i>Figura 4.20</i> Respuesta del lazo de control con el FOSM frente a la perturbación de error de medición.....	42
<i>Figura 4.21</i> Respuesta del lazo de control con el FOSM frente a las 2 perturbaciones.....	43
<i>Figura 4.22</i> Control de velocidad con el algoritmo Suboptimal.....	44



<i>Figura 4.23</i>	<i>Respuesta del Suboptimal con diferenciador de Simulink.....</i>	<i>44</i>
<i>Figura 4.24</i>	<i>Respuesta del Control Suboptimal con diferenciador de Simulink.....</i>	<i>44</i>
<i>Figura 4.25</i>	<i>Trayectoria del error en el espacio de estados con diferenciador de Simulink.....</i>	<i>44</i>
<i>Figura 4.26</i>	<i>Espacio de estados del control Suboptimal con diferenciador de Simulink.....</i>	<i>44</i>
<i>Figura 4.27</i>	<i>Respuesta del Suboptimal con diferenciador SM.....</i>	<i>45</i>
<i>Figura 4.28</i>	<i>Trayectoria del error en el espacio de estados.....</i>	<i>45</i>
<i>Figura 4.29</i>	<i>Espacio de estados del control Suboptimal con diferenciador SM.....</i>	<i>45</i>
<i>Figura 4.30</i>	<i>Respuesta del Suboptimal con diferenciador Levant.....</i>	<i>46</i>
<i>Figura 4.31</i>	<i>Trayectoria del error en el espacio de estados con diferenciador Levant.....</i>	<i>46</i>
<i>Figura 4.32</i>	<i>Espacio de estados del control Suboptimal con diferenciador Levant.....</i>	<i>46</i>
<i>Figura 4.33</i>	<i>Respuesta del Suboptimal con diferenciador Shtessel.....</i>	<i>47</i>
<i>Figura 4.34</i>	<i>Trayectoria del error en el espacio de estados.....</i>	<i>47</i>
<i>Figura 4.35</i>	<i>Espacio de estados del control Suboptimal.....</i>	<i>47</i>
<i>Figura 4.36</i>	<i>Control de velocidad con el algoritmo Suboptimal frente a perturbaciones.....</i>	<i>51</i>
<i>Figura 4.37</i>	<i>Respuesta del lazo de control con el Suboptimal frente a la perturbación del Amplificador.....</i>	<i>51</i>
<i>Figura 4.38</i>	<i>Respuesta del lazo de control con el Suboptimal frente a la perturbación de error de medición.....</i>	<i>52</i>
<i>Figura 4.39</i>	<i>Respuesta del lazo de control con el Suboptimal frente a las 2 perturbaciones.....</i>	<i>52</i>
<i>Figura 4.40</i>	<i>Estructura interna del control Suboptimal como Filtro.....</i>	<i>53</i>
<i>Figura 4.41</i>	<i>Acción del control Suboptimal como Filtro.....</i>	<i>54</i>
<i>Figura 4.42</i>	<i>Respuesta del lazo de control con el Suboptimal como filtro.....</i>	<i>54</i>
<i>Figura 4.43</i>	<i>Respuesta del lazo de control con el Suboptimal como filtro frente a las 2 perturbaciones.....</i>	<i>55</i>
<i>Figura 4.44</i>	<i>Control de velocidad con el algoritmo PID.....</i>	<i>55</i>
<i>Figura 4.45</i>	<i>Curva de reacción para el ajuste del PID en el motor de CD.....</i>	<i>56</i>
<i>Figura 4.46</i>	<i>Respuesta del control PID.....</i>	<i>57</i>
<i>Figura 4.47</i>	<i>Trayectoria del error en el espacio de estados.....</i>	<i>57</i>
<i>Figura 4.48</i>	<i>Trayectoria en el espacio de estados del control PID.....</i>	<i>57</i>
<i>Figura 4.49</i>	<i>Respuesta del control PID con diferenciador SM.....</i>	<i>58</i>
<i>Figura 4.50</i>	<i>Trayectoria del error en el espacio de estados con diferenciador SM.....</i>	<i>58</i>
<i>Figura 4.51</i>	<i>Trayectoria en el espacio de estados del control PID con diferenciador SM.....</i>	<i>58</i>
<i>Figura 4.52</i>	<i>Respuesta del control PID con diferenciador de Levant.....</i>	<i>58</i>
<i>Figura 4.53</i>	<i>Trayectoria del error en el espacio de estados con diferenciador de Levant.....</i>	<i>59</i>
<i>Figura 4.54</i>	<i>Trayectoria en el espacio de estados del control PID con diferenciador de Levant.....</i>	<i>59</i>
<i>Figura 4.55</i>	<i>Respuesta del control PID con diferenciador de Shtessel.....</i>	<i>59</i>
<i>Figura 4.56</i>	<i>Trayectoria del error en el espacio de estados con diferenciador de Shtessel.....</i>	<i>59</i>
<i>Figura 4.57</i>	<i>Trayectoria en el espacio de estados del control PID con diferenciador de Shtessel.....</i>	<i>59</i>
<i>Figura 4.58</i>	<i>Diagrama del PID frente a perturbaciones.....</i>	<i>60</i>
<i>Figura 4.59</i>	<i>Respuesta del PID frente a la perturbación del Amplificador.....</i>	<i>61</i>
<i>Figura 4.60</i>	<i>Respuesta del PID frente a la perturbación de error de medición.....</i>	<i>61</i>
<i>Figura 4.61</i>	<i>Respuesta del PID frente a las 2 perturbaciones.....</i>	<i>61</i>
<i>Figura 5.1</i>	<i>Cadena de generación automática a partir de modelos Simulink.....</i>	<i>65</i>
<i>Figura 5.2</i>	<i>Ventana de Parámetros del Real Time Workshop.....</i>	<i>66</i>
<i>Figura 5.3</i>	<i>Mensajes emitidos por RTW durante la generación automática de un ejecutable.....</i>	<i>67</i>
<i>Figura 5.4</i>	<i>Pantalla de Interacción con Simulink del Programa ControlDesk.....</i>	<i>67</i>
<i>Figura 5.5</i>	<i>RTI del DSpace 1103.....</i>	<i>68</i>
<i>Figura 5.6</i>	<i>Bloques DAC y Enc. Position del Real Time Interface.....</i>	<i>68</i>
<i>Figura 5.7</i>	<i>Bloques PWM y Enc. Position del Real Time Interface.....</i>	<i>68</i>
<i>Figura 5.8</i>	<i>Forma de Trabajar para el Control del Velocidad en Tiempo Real.....</i>	<i>70</i>
<i>Figura 5.9</i>	<i>Motor Empleado.....</i>	<i>70</i>
<i>Figura 5.10</i>	<i>Fuente de Alimentación.....</i>	<i>70</i>
<i>Figura 5.11</i>	<i>Tarjeta DSPACE 1103.....</i>	<i>71</i>
<i>Figura 5.12</i>	<i>Lazo de Control para el Motor de CD.....</i>	<i>71</i>
<i>Figura 5.13</i>	<i>Lazo de control con el FOSM para el motor de CD con diferenciador aproximado.....</i>	<i>72</i>
<i>Figura 5.14</i>	<i>Respuesta del FOSM con diferenciador aproximado.....</i>	<i>72</i>
<i>Figura 5.15</i>	<i>Acción de control del FOSM con diferenciador aproximado.....</i>	<i>72</i>
<i>Figura 5.16</i>	<i>Trayectoria del error en el espacio de estados.....</i>	<i>72</i>
<i>Figura 5.17</i>	<i>Respuesta del FOSM con diferenciador Levant.....</i>	<i>73</i>



<i>Figura 5.18</i> Acción de control del FOSM con diferenciador Levant.....	73
<i>Figura 5.19</i> Trayectoria del error en el espacio de estados.....	73
<i>Figura 5.20</i> Lazo de control de velocidad con el algoritmo Suboptimal y un diferenciador lineal.....	74
<i>Figura 5.21</i> Respuesta del Suboptimal con diferenciador aproximado.....	74
<i>Figura 5.22</i> Acción de control del Suboptimal con diferenciador aproximado.....	74
<i>Figura 5.23</i> Trayectoria del error en el espacio de estados.....	75
<i>Figura 5.24</i> Respuesta del Suboptimal con diferenciador SM.....	75
<i>Figura 5.25</i> Trayectoria del error en el espacio de estados.....	76
<i>Figura 5.26</i> Respuesta del Suboptimal con diferenciador Levant.....	76
<i>Figura 5.27</i> Trayectoria del error en el espacio de estados.....	76
<i>Figura 5.28</i> Respuesta del Suboptimal con diferenciador Shtessel.....	77
<i>Figura 5.29</i> Trayectoria del error en el espacio de estados.....	77
<i>Figura 5.30</i> Acción de control del Suboptimal con diferenciador Shtessel.....	77
<i>Figura 5.31</i> Lazo de control de velocidad con el algoritmo Suboptimal y diferenciador de Levant.....	78
<i>Figura 5.32</i> Respuesta del Suboptimal como filtro con diferenciador Levant.....	79
<i>Figura 5.33</i> Acción de control del Suboptimal como filtro con diferenciador Levant.....	79
<i>Figura 5.34</i> Trayectoria del error en el espacio de estados.....	79
<i>Figura 5.35</i> Respuesta del Suboptimal como filtro con “diferenciador aproximado”.....	80
<i>Figura 5.36</i> Lazo de control de velocidad con el control PID y diferenciador lineal.....	80
<i>Figura 5.37</i> Estructura interna del acondicionador de señal PID.....	81
<i>Figura 5.38</i> Respuesta del control PID propuesto con diferenciador aproximado.....	81
<i>Figura 5.39</i> Acción del control PID propuesto con diferenciador aproximado.....	81
<i>Figura 5.40</i> Trayectoria del error en el espacio de estados.....	82
<i>Figura 5.41</i> Respuesta del control PID de Simulink con diferenciador aproximado.....	82
<i>Figura 5.42</i> Acción del control PID de Simulink con diferenciador aproximado.....	82
<i>Figura 5.43</i> Trayectoria del error en el espacio de estados.....	82
<i>Figura 5.44</i> Respuesta del control PID propuesto con diferenciador Levant.....	82
<i>Figura 5.45</i> Acción del control PID propuesto con diferenciador Levant.....	82
<i>Figura 5.46</i> Trayectoria del error en el espacio de estados.....	83
<i>Figura 5.47</i> Respuesta del control PID de Simulink con diferenciador Levant.....	83
<i>Figura 5.48</i> Acción del control PID de Simulink con diferenciador Levant.....	83
<i>Figura 5.49</i> Trayectoria del error en el espacio de estados.....	83
<i>Figura A.1</i> Diagrama del PID de Simulink.....	93
<i>Figura A.2</i> Diagrama del PID propuesto.....	93
<i>Figura A.3</i> Método de Oscilaciones sostenidas de Ziegler-Nichols.....	94
<i>Figura A.4</i> Método curva de reacción de Ziegler-Nichols.....	95
<i>Figura A.5</i> Forma de una PWM.....	97
<i>Figura A.6</i> Señales del Encoder con giro positivo.....	98
<i>Figura A.7</i> Señal del Encoder con giro negativo.....	99
<i>Figura A.8</i> Bloque de Configuración del Encoder.....	99
<i>Figura A.9</i> Diagrama base de la disposición en H de los transistores.....	100
<i>Figura A.10</i> Camino de Corriente para giro positivo.....	100
<i>Figura A.11</i> Camino de Corriente para giro negativo.....	100
<i>Figura A.12</i> Diagrama del inversor utilizado.....	101
<i>Tabla A.1</i> Sintonización PID en lazo cerrado.....	94
<i>Tabla A.2</i> Sintonización PID en lazo abierto.....	95
<i>Tabla C.1</i> Resultados de la simulación con el algoritmo FOSM.....	87
<i>Tabla C.2</i> Resultados de la simulación con el algoritmo Suboptimal.....	87
<i>Tabla C.3</i> Resultados de la simulación con el algoritmo Suboptimal como Filtro.....	88
<i>Tabla C.4</i> Resultados de la simulación con el control PID.....	88
<i>Tabla C.5</i> Resultados de las pruebas en tiempo real con el algoritmo FOSM.....	89
<i>Tabla C.6</i> Resultados de las pruebas en tiempo real con el algoritmo suboptimal.....	89
<i>Tabla C.7</i> Resultados de las pruebas en tiempo real con el algoritmo Suboptimal como filtro.....	90



**Tabla C.8** Resultados de las pruebas en tiempo real con el control PID propuesto.....90



# INTRODUCCIÓN

## *Presentación*

En los últimos años la ingeniería de control ha jugado un papel de gran importancia para el desarrollo de nuevas tecnologías, llegando a ser una de las herramientas indispensables para el mejoramiento de muchos procesos industriales, de manufactura, robótica, aeronáutica, comunicaciones, bioingeniería, etc.

Gracias a los avances en la teoría y práctica de la ingeniería de control, día a día se aportan nuevas formas para obtener mejores desempeños de los sistemas dinámicos, eficientar los equipos de trabajo, mejorar la productividad, alargar la vida útil de la maquinaria, aligerar la carga para realizar operaciones manuales y/o de monitoreo, entre otras cosas.

En el transcurso de la historia de la ingeniería de control la intensidad de investigación y desarrollo de sistemas con acciones de control discontinuo han sido mantenidas en un alto nivel, en particular la teoría de los sistemas de estructura variable<sup>i</sup> han despertado gran interés para el control de los equipos electrónicos de potencia y actuadores eléctricos. Las ventajas (en términos de robustez y eficiencia) ofrecidas por esta técnica son notables.

En sistemas con control en función del estado discontinuo llamados comúnmente modos deslizantes<sup>ii</sup>, son un ejemplo claro a la solución de problemas de matemática pura, hasta aspectos de aplicación, los sistemas con modos deslizantes tienen la capacidad para ser una herramienta poderosa y eficiente para el control complejo de plantas dinámicas de alto orden que operen bajo condiciones inciertas, así mismo, pueden emplearse para obtener los parámetros que se desconozcan del sistema o para la simple observación de la variación de los mismos.

Comparando con otras técnicas tradicionales o clásicas de control de las que ya se tiene conocimiento de su desempeño y limitaciones frente a la variación de sus parámetros<sup>iii</sup>, el control por modos deslizantes es una técnica moderna de control que es ampliamente utilizada para trabajar con sistemas con dinámicas no modeladas e incertidumbres paramétricas.



Esta técnica se basa en el cumplimiento de ciertas restricciones predefinidas mediante la conmutación en alta frecuencia (idealmente infinita) de una señal de control, sus ventajas principales son la convergencia en tiempo finito, la precisión, la insensibilidad frente a perturbaciones (externas o internas), la reducción del orden del sistema en estudio y la más interesante que consiste en que no necesario conocer el modelo completo de nuestro sistema a controlar.

Sin embargo no todo es perfecto y dado que no existen implementaciones de casos ideales, existe algo que limita la aplicación de este tipo de control: la frecuencia de conmutación que generan los actuadores producen el fenómeno conocido como *chattering* ó “castaño”, el cual se manifiesta como una oscilación de frecuencia y amplitud finita no deseable. Estas oscilaciones dan por resultado alta disipación de potencia en los circuitos de conmutación, baja precisión en el control e incluso el desgaste de los elementos mecánicos del sistema.

En las aplicaciones experimentales con modos deslizantes, hay que tomar en cuenta las dinámicas no modeladas y los retardos de los elementos del sistema como son sensores y accionadores en la etapa de diseño y ajuste (sintonización) del controlador para mejorar su eficiencia y disminuir la presencia del fenómeno de *chattering*. En la actualidad se han formulado una gran cantidad de soluciones para contrarrestar este fenómeno, una solución al efecto del *chattering*, es el diseño de modos deslizantes de orden superior (HOSM, por sus siglas en inglés), generalizando la idea de los modos deslizantes convencionales actuando en las derivadas de las variables del sistema, a su vez se han desarrollado una variedad de diferenciadores<sup>iv</sup> basados en la teoría de modos deslizantes, los cuales permiten obtener una aproximación más cercana al valor real de cada una de las derivadas.

El algoritmo *Suboptimal* pertenece al grupo de los **HOSM** aunque de forma específica a los algoritmos con modos deslizantes de segundo orden (**2-SMC**), conservando las características generales: convergencia en tiempo finito, robustez, insensibilidad frente a perturbaciones, reducción del orden del sistema y una presencia casi nula del efecto *chattering*.



## *Motivación*

La teoría de sistemas con control de modos deslizantes es un área que no está dentro del plan de estudios actual. Sin considerar lo complejo del fondo matemático con que se definen y a manera de visualizar su comportamiento, el presente trabajo pretende establecer, desde un punto de vista académico, de investigación y con un enfoque orientado hacia el aprendizaje; un ejemplo ilustrativo, de una serie de pruebas experimentales y de implementación al control de velocidad de un motor de corriente directa, con modos deslizantes y modos deslizantes de orden superior: algoritmo *Subóptimo*, y de la aplicación de los diferenciadores que se proponen en esta tesis para la obtención de señales diferenciadas.

## *Formulación del Problema y Contribuciones*

En el desarrollo del presente trabajo se estudiarán las características de la velocidad de salida del comportamiento de un motor de **CD** cuando se controla por medio del algoritmo *Subóptimo* en conjunto con los diferenciadores propuestos basados en la teoría de modos deslizantes.

Dentro de la teoría de modos deslizantes se destaca la ventaja de no tener que conocer el modelo de la planta a controlar debido a su robustez y eficiencia, pero sin embargo, como no se está diseñando un algoritmo que contemple los retardos de los elementos que actúan en nuestro sistema y los dispositivos accionadores no generan las frecuencias de conmutación ideales de operación del control con modos deslizantes, se verificará la presencia del fenómeno *chattering* en la prueba, así mismo los valores de los parámetros establecidos en la etapa de simulación computacional se mantendrán fijos al realizar la implementación para la prueba experimental en tiempo real, ya que de esta forma se entenderá de manera sencilla lo que ocurre en cada caso. El presente trabajo tratará de presentar pruebas en las que visualice una serie de variantes para corregir dichas imperfecciones y se logre apreciar las características generales de operación de los modos deslizantes de segundo orden a través del algoritmo *Subóptimo* en el control de velocidad del motor de **CD**.



## Objetivo General

Se realizarán pruebas experimentales del control de velocidad a un motor de corriente directa empleando el algoritmo *Suboptimal* de modos deslizantes, resaltando las características generales de operación con base a la teoría de funcionamiento de los modos deslizantes de segundo orden y se introducirá en el lazo de control la aplicación de una serie de diferenciadores basados en la teoría de modos deslizantes para verificar el desempeño de los mismos.

## Objetivos Específicos

- a) Conocer el funcionamiento del algoritmo *Suboptimal*
- b) Realizar simulaciones computacionales del modelo matemático del motor empleado, utilizando las siguientes leyes de control:
  - Control por Modos Deslizantes Clásicos (*SMC*).
  - Control por Modos Deslizantes de segundo orden (*Suboptimal*).
  - Control Proporcional Integral Derivativo (*PID*).
- c) Introducir en el lazo de control velocidad los diferenciadores de Modos Deslizantes.
  - Diferenciador por Modos Deslizantes Clásicos (*SM*).
  - Diferenciador por Modos Deslizantes de segundo orden (Super-twisting, *Levant*).
  - Diferenciador de *Shtessel* ó filtro *SSZI*
- d) Realizar la implementación en tiempo real de los diferentes controles estudiados en la simulación.
- e) Realizar conclusiones.



## Contenido de la Tesis

La presente tesis se divide en 5 capítulos y un apartado de anexos, en el primer Capítulo se hace una breve descripción del concepto y operación de los modos deslizantes de primer orden (**FOSM**, por sus siglas en inglés), se describe en que consiste el fenómeno de chattering y se plantea como se utilizará éste control para llevar a cabo el control de velocidad del motor de **CD**, el Capítulo dos hace mención de la definición de los algoritmos de orden superior y enfatiza en el funcionamiento del algoritmo *Subotimal* de modos deslizantes de segundo orden (**2-SMC**) y la importancia del mismo para contrarrestar el *chattering* inherente a los modos deslizantes. El Capítulo tres introduce a la aplicación de tres algoritmos de modos deslizantes, los cuales permiten la obtención de una derivada más cercana al valor real y que facilitarán el buen funcionamiento del controlador en estudio dadas las características que los definen y a la facilidad con la que se pueden implementar en el lazo de control. El Capítulo cuatro relata las pruebas en calidad de simulación computacional aplicadas al motor de **CD** de los controladores en estudio en conjunto con los diferenciadores propuestos, además de aplicar un control clásico de tipo **PID** para el control de velocidad del motor de **CD**, por último, el Capítulo cinco explica como opera un sistema **CACSD** (*Computer Aided Control System Design*) y el como se lleva a cabo la implementación del control de velocidad del motor de **CD** en tiempo real.

El apartado de Anexos describe de forma breve el control **PID** y hace mención de una propuesta alterna al bloque de simulink para este control, además de mencionar las características principales de la tarjeta **dSPACE 1103** y la descripción de las señales que maneja y que se utilizarán durante las pruebas, así mismo, se explica como opera la etapa de potencia del sistema para manejar al motor de **CD**.

---

<sup>i</sup> Para profundizar en el entendimiento de este tipo de sistemas, ver [5]

<sup>ii</sup> Ver capítulo 1 de [6]

<sup>iii</sup> El controlador PID, es un ejemplo.

<sup>iv</sup> Estos diferenciadores se estudiarán en el capítulo 3



# CAPÍTULO UNO

## *Control de Modos Deslizantes*

### **1.0.- Introducción**

El término modo deslizante aparece primero en el contexto de sistemas de la estructura variable, pronto el modo deslizante se volvió el modo operacional principal para esta clase de sistemas de control. Prácticamente todos los métodos para los sistemas de la estructura variable son basados en la introducción deliberada de modos deslizantes que han jugado, y todavía está jugando un papel excepcional en desarrollos teóricos y en aplicaciones prácticas.

El sistema de control de estructura variable (**VSC**, por sus siglas en inglés) con Modos Deslizantes (**SM**) se caracteriza por la facilidad con la que trata no linealidades e incertidumbres en los sistemas de control. Entre los sistemas susceptibles de aplicar la teoría del **VSC** se tienen: sistemas no lineales, sistemas con incertidumbres paramétricas, sistemas de gran escala, sistemas multivariables, etc. Aparte de su aplicación principal, que es la estabilización de sistemas, las técnicas del **VSC** comienzan a aplicarse al control de sistemas de seguimiento de movimientos, control óptimo y al control adaptativo.

El **SM** presenta ventajas notables en su aplicación como la insensibilidad a las incertidumbres de los sistemas. **SM** puede ser aplicado a sistemas lineales o sistemas que no lo sean; con incertidumbres, porque en él los estados del sistema siguen la ecuación de la superficie de conmutación  $s(x)=0$ . **SM** puede controlar el sistema mediante el uso de un sistema de orden reducido: ecuaciones de modos deslizantes. Dichas ecuaciones son invariantes frente a la presencia de incertidumbres. En el avance de éste capítulo se revisarán las propiedades del **VSC** y se comentará el procedimiento para elegir los parámetros de cada una de las dos estructuras que forman el sistema final y se explica como el sistema resultante tendrá propiedades muy importantes pues se puede diseñar un control que no depende de los parámetros del sistema original, sino de un modo totalmente distinto del sistema a controlar, lo que ofrece un control invariante a las perturbaciones e incertidumbres del sistema.

## 1.1.- Modos Deslizantes

### 1.1.1.- Concepto

Un sistema de estructura variable consiste en un conjunto de subsistemas continuos con una lógica de conmutación adecuada, que pretende llevar a la estabilidad el sistema a controlar.<sup>1</sup> Una propiedad importante de los sistemas de estructura variable es la posible existencia de una conmutación permanente a una muy alta frecuencia en las superficies de discontinuidad de la estructura. Si la trayectoria del sistema alcanza las proximidades de una región en la que existe esta conmutación, el movimiento ya no se aleja de la región excepto posiblemente en las fronteras de la región.

Por ejemplo supongamos una estructura variable de segundo orden como es el caso del motor a estudiar, donde se presenta una conmutación de alta frecuencia en un segmento del plano de fase,  $\mathbf{R}^2$ :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \varphi(t, x) + \gamma(t, x)u \end{cases}$$

Entonces, las trayectorias que tocan el segmento  $\mathbf{R}$ , permanecen sobre  $\mathbf{R}$  excepto si llegan a los extremos del segmento, donde, posiblemente puedan escapar de la región. (Figura 1.1)

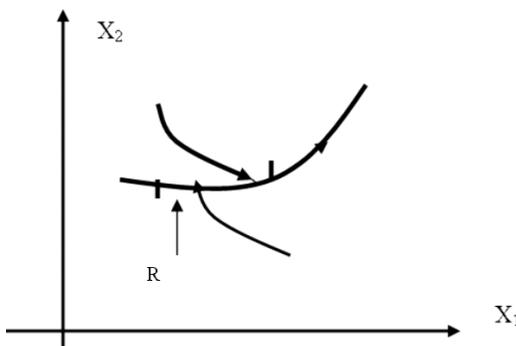


Figura 1.1 Trayectoria en el plano de fase de una estructura variable de segundo orden

<sup>1</sup> Ver [5]



A  $\mathbf{R}$  también se le llama región de deslizamiento. Cuando la trayectoria evoluciona sobre la región  $\mathbf{R}$  se dice que el sistema está en “**modo deslizante**”. Cuando se encuentra en modo deslizante un sistema puede tener cualquier comportamiento dinámico, como tender hacia un punto de equilibrio asintóticamente estable dentro de la región  $\mathbf{R}$  ó seguir una trayectoria de amplitud creciente que lo lleve a abandonar la región de deslizamiento.

Para que se presente un modo deslizante en una superficie de discontinuidad entre dos estructuras, las trayectorias a ambos lados de la superficie deben dirigirse hacia la superficie.

### 1.1.2.-Control en Modo Deslizante

Definimos la siguiente ley de control o lógica de conmutación

$$u = \begin{cases} u+ & \text{Si } s(x) > 0 \\ u- & \text{Si } s(x) < 0 \end{cases}$$

$$u \in \mathfrak{R}^m$$

La función  $s(x)$  se denomina función de conmutación y determina la siguiente superficie de dimensión  $n-1$ .

$$S = \{x \in \mathfrak{R}^n : s(x) = 0\}$$

La superficie  $S$  se denomina superficie de discontinuidad o superficie de conmutación.<sup>2</sup>

Sí hay alguna región  $\mathbf{R}$  donde aparece un modo deslizante, es un subconjunto de  $S$ ,  $R \subset S$ .

Se dice que el sistema está controlado en modo deslizante cuando hay una región  $\mathbf{R}$  donde existe un modo deslizante.

### 1.2.-Modos deslizantes de primer orden (FOSM)

Los modos deslizantes de primer orden (**FOSM**, por sus siglas en inglés), conservan todas las características de los Modos Deslizantes ideales, lo que se quiere decir es que tienen convergencia en tiempo finito, son robustos e insensibles a la variación de los parámetros de la planta e incertidumbres asociadas, aunque, dado que la frecuencia de conmutación que se puede obtener es finita y no infinita, como se supone en el caso ideal, poseen gran presencia del fenómeno *chattering* en su acción.

<sup>2</sup> La forma de la función de conmutación depende del tipo de algoritmo en modos deslizantes empleado.

Los modos deslizantes de primer se rigen bajo la siguiente ley de control:

$$u = -M \text{sign}(s)$$

donde  $u$  es el control,  $M$  es la ganancia máxima de operación y la variable de deslizamiento  $S$ , esta definida por

$$s = Cx + \dot{x}$$

lo que al encontrar el modo deslizante en alguna región  $R$  de la superficie  $S$ , la representación en el espacio de estados describirá una recta con pendiente  $C$ .

Ahora bien, la rapidez de acción de éste control depende de los parámetros  $M$  y el valor de  $C$ , así también, de las condiciones iniciales de nuestro sistema.

Aunque considerando que lo que se busca es hacer que el error del sistema a controlar sea nulo<sup>3</sup>, se realiza un cambio de variable en la definición de la superficie de deslizamiento, la cual queda expresada en función del error como:

$$s = Ce + \dot{e}$$

al realizar esta consideración se logra que se alcance la referencia ó valor deseado cuando el error sea nulo y, la representación en el espacio de estados de lo anterior será el origen.

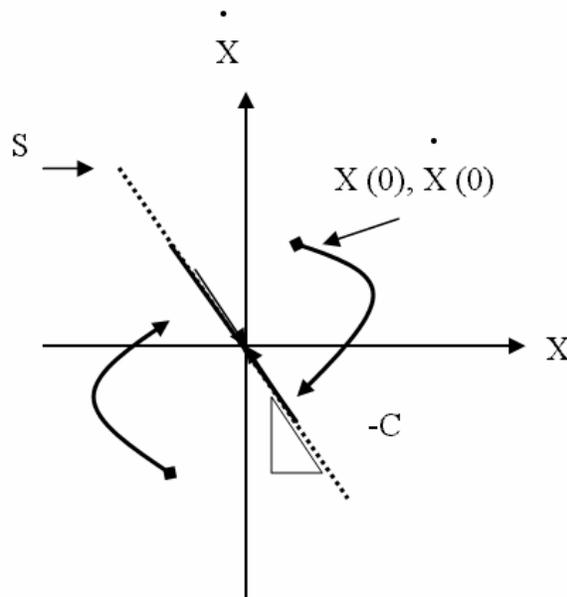


Figura 1.2 Trayectoria del FOSM en el espacio de estados

<sup>3</sup> Cuando la diferencia entre la salida del sistema y el valor de referencia en el lazo de control, sea igual a cero.



### 1.2.1.-Existencia del Modo Deslizante

En las proximidades de los puntos de  $\mathbf{R}$ , la función de conmutación  $s(x)$  y su derivada temporal deben tener signos contrarios para que las trayectorias tiendan hacia  $\mathbf{S}$ . las condiciones de existencia de modo deslizante se pueden expresar de la siguiente forma:

$$\begin{aligned} \dot{s} &= -\text{sign}(s) \\ s\dot{s} &< 0 \end{aligned}$$

En los puntos de  $\mathbf{S}$  donde se cumpla lo anterior será una región de deslizamiento.

### 1.3.-Consideraciones generales

En el  $\mathbf{VSC}$ , los algoritmos de diseño incluyen la selección de las superficies deslizantes  $s(x)$  deseadas, esto es, la elección de sus parámetros. Posteriormente se implementa un control discontinuo que asegura la existencia en cada punto de la región deslizante  $s(x)=0$ . En la etapa final, el control debe dirigir los estados del sistema hacia a superficie deslizante.

En general, las dinámicas del  $\mathbf{VSC}$  constan de dos condiciones: la condición de alcance y la condición deslizante. En la primera, el objetivo es llegar al plano de conmutación en un tiempo finito desde cualquier punto. También se denomina modo no deslizante. La segunda parte se denomina el modo deslizante, durante el cual la trayectoria tiende al origen del espacio de estados. Los parámetros de conmutación condicionan el comportamiento final del sistema durante el modo deslizante.

Se pueden observar cuatro características importantes de este método de control:

1.- Como el origen de estados representa el estado de equilibrio del sistema, el  $\mathbf{SM}$  representa el comportamiento del mismo durante el periodo transitorio. En otras palabras, la superficie que describe  $s(x)=0$  define la respuesta transitoria del sistema durante el  $\mathbf{SM}$ .

2.- Durante el  $\mathbf{SM}$ , las dinámicas de la trayectoria son de un orden menor que el orden del sistema original.

3.- Durante el  $\mathbf{SM}$ , la dinámica de la trayectoria esta gobernada solamente por los parámetros que describe la función deslizante.

4.- El  $\mathbf{SM}$  es una trayectoria que no es inherente a ninguna de las estructuras de control.

La ley de control del sistema de estructura variable se cambia cuando su estado cruza la línea de conmutación. En general, la entrada del control puede que no sea suave, e incluso a veces es discontinua durante la conmutación y puede excitar dinámicas no modeladas de alta frecuencia lo que origina la presencia de un fenómeno mejor conocido como *chattering*.<sup>4</sup> Durante las dos décadas anteriores se han propuesto una serie de métodos para suavizar la entrada de control y disminuir este efecto de aquí la aparición de los algoritmos de control por modos deslizantes de orden superior (*HOSM*).

El diagrama propuesto en Simulink de Mathworks<sup>5</sup>, para el desarrollo de las simulaciones y la implementación experimental que busca el objetivo de esta tesis para éste algoritmo es:

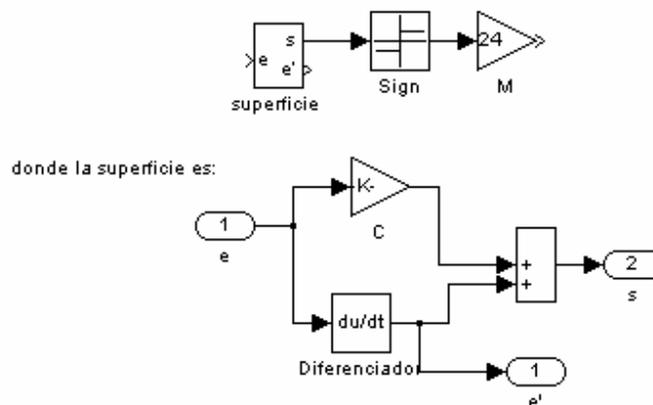


Figura 1.3 Diagrama en Simulink del FOSM

Como puede apreciarse en la figura anterior el diseño se realizó con el diferenciador de Simulink para la obtención de la derivada del error, este hecho, se estudiará en el Capítulo 3, donde se hará la propuesta de otros algoritmos que nos permiten realizar la diferenciación de una señal de tal forma que se permitirá hacer un estudio a detalle de la importancia de la obtención de una derivada aproximada al valor real y que mejora las características del algoritmo *FOSM*.

<sup>4</sup> La amplitud y frecuencia de este fenómeno indeseable no depende de los algoritmos en modos deslizantes, básicamente depende de las imperfecciones del sistema real.

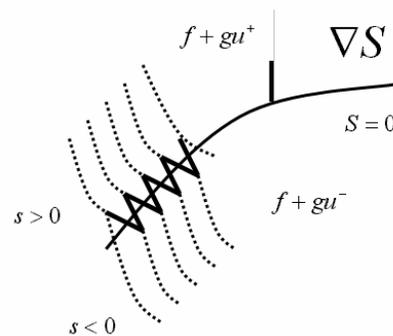
<sup>5</sup> Simulink/ Matlab es un paquete con gran cantidad de librerías y aplicaciones a nivel de bloques, que permite en éste trabajo el desarrollo de los algoritmos de control a estudiar.

### 1.4.-El problema del chattering

Sea un sistema dinámico estable en el origen a lazo abierto descrito por la ecuación:

$$\frac{dx}{dt} = f(x) + g(x)u \quad y(x) = s(x) \quad f, x \in \mathcal{R}^n, u \in \mathcal{R}^m$$

y se desea que la salida del mismo esté confinada en una superficie  $s(x)=0$  con un control de la forma  $u = -U \text{sign}(s)$ . Cuando el sistema se encuentra en la superficie de deslizamiento y el control trabaja a frecuencia finita, aparece el fenómeno de chattering que puede visualizarse en la figura.



*Figura 1.4 Representación del Fenómeno chattering de un sistema controlado por modos deslizantes*

La dinámica propia del sistema produce un desbalance en las acciones de control. Cuando un sistema estable en el origen, de interés en éste análisis, es llevado a un punto de operación cualquiera  $\bar{x}$ , la dinámica propia del sistema evoluciona hacia el origen, esto produce un desequilibrio de las acciones de control,  $u^+$  y  $u^-$ . Por ejemplo, la acción de la señal  $u^+$ , aplicada cuando  $s(x) > 0$  será mucho mayor, es decir, producirá un mayor apartamiento que la aplicación de la señal  $u^-$ , que es la señal que lo aleja del origen, aplicada cuando  $s(x) < 0$ . Este comportamiento es debido a que en un caso la dinámica propia del sistema aporta a dicho apartamiento y en el otro, la dinámica libre debe ser contrarrestada, en otras palabras:

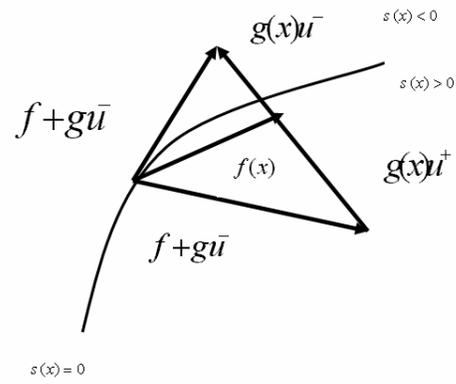


Figura 1.5 Acción de los controles  $u^+$  y  $u^-$



## CAPÍTULO 2

# *Modos Deslizantes de Orden Superior*

### **2.0.- Introducción**

Los algoritmos de modos deslizantes de orden superior (**HOSM**, por sus siglas en inglés), están basados en el principio de funcionamiento de los modos deslizantes de primer orden, pero diseñados como una solución al problema de *chattering*, permitiendo que el control del sistema en estudio sea más suave y con convergencia en tiempo finito, además de poseer la característica de poder reducir en su acción el orden del sistema a controlar.

Existe una gran variedad de algoritmos propuestos por varios autores basados todos en la teoría de modos deslizantes cumpliendo a su vez con la propiedad de robustez a la variación de los parámetros de la planta a controlar y dinámicas no modeladas, así como también, de insensibilidad frente a perturbaciones externas.

Algunos de ellos son el algoritmo *cuasi-continuo* [15], *twisting* [16], *supertwisting* [16] y el que nos interesa para su estudio el algoritmo *Subóptimo*. [8, 9,10]

### **2.1.- Concepto**

Los modos deslizantes de orden superior tienen la tarea de llevar a través del control conmutado a la variable de deslizamiento y a sus derivadas sucesivas al valor de cero en la vecindad de deslizamiento en tiempo finito, hablando en términos generales, llegar al origen del espacio de estados, el cual dependerá del orden del sistema a controlar, pues para esta clase de algoritmos basta con saber el grado relativo de la planta para efectuar el control.

Para el sistema con grado relativo  $r$  con respecto a la condición  $s = 0$ :

El dominio

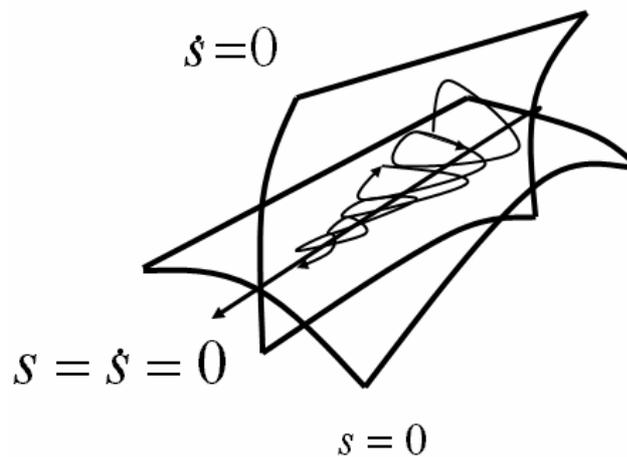
$$s = \dot{s} = \ddot{s} = \dots = s^{(r-1)} = 0$$

es el dominio de deslizamiento de orden  $r$

Para el caso de un sistema de segundo orden:

$$s = \dot{s} = 0$$

donde  $s$  es la variable de deslizamiento.



**Figura 2.1** Trayectoria de un Modo deslizante de segundo Orden (2-SM)

Los modos deslizantes de orden dos, actúan sobre la segunda derivada de la variable de deslizamiento y ocurre mientras se está llegando al origen de nuestro plano de estados, donde ocurrirá la conmutación de alta frecuencia (idealmente infinita) garantizando su existencia.

Ahora se definirán las condiciones de existencia de los modos deslizantes de segundo orden de la siguiente manera:

Considere el sistema

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -a_1 x_1 - a_2 x_2 + f(t) - bu\end{aligned}$$

donde la variable de deslizamiento es  $x_1$  y el control  $u$ , el cual actúa en la ecuación donde aparece la última derivada de grado relativo  $r=2$ , es decir, en  $\dot{x}_2 = \ddot{x}_1$ .



Ahora bien lo que garantiza que los modos deslizantes existan es la siguiente condición:

$$x_1 = \dot{x}_1 = 0$$

Cabe recordar que la dinámica del sistema estará regida por la superficie de deslizamiento dada por el control  $u$  y, el algoritmo de modos deslizantes empleado indicará la forma de onda que describe en su acción.

Por otro lado, lo que se busca en el control de un sistema es que el error sea nulo, de modo que si consideramos como variable de deslizamiento al error  $e^6$ , permite analizar que pasa en realidad en el espacio de estados del sistema a controlar y manejar de forma clara en que punto ocurre el modo deslizante.

En otras palabras, el modo deslizante existe en el punto:

$$e_1 = \dot{e}_1 = 0$$

## 2.2.- Algoritmo Suboptimal de Modos Deslizantes de Segundo Orden [9]

El algoritmo *Suboptimal* de modos deslizantes de segundo orden (2-SMC), pertenece a los algoritmos de orden superior (*HOSM*), conservando todas y cada una de las características que los definen como: convergencia en tiempo finito, robustez e insensibilidad frente a perturbaciones externas.

Fue diseñado a partir de la necesidad del problema de *chattering*, así como una herramienta a la teoría del control óptimo.

### 2.2.1.- Concepto

Este algoritmo de segundo orden fue desarrollado como una retroalimentación sub-óptima del funcionamiento del control clásico de tiempo óptimo del doble integrador.[8]

El grado relativo  $r$  del algoritmo es 2.

Considerando el siguiente sistema auxiliar:

$$\begin{cases} \dot{y}_1 = y_2 \\ \dot{y}_2 = \varphi(t, x) + \gamma(t, x)u \end{cases}$$

<sup>6</sup> Esta condición de cambio de variable es planteada de igual forma que en el caso FOSM

Las trayectorias en el espacio de estados definido por  $y_1$  vs  $y_2$ , describen una dinámica limitada por arcos parabólicos y el origen del plano, donde su comportamiento característico es dar vueltas y saltos (cuando  $y_1$  y  $y_2$  no cambian de signo). Así también, donde las coordenadas de la trayectoria intersectan al eje  $y_1$ , estas decrecen como progresión geométrica.

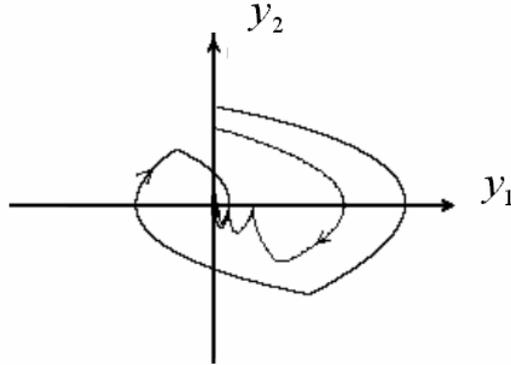


Figura 2.2 Trayectoria de Fase del algoritmo *Suboptimal*

Antes de iniciar su trayectoria en el plano de fase el algoritmo *Suboptimal* es regido por la siguiente ley de control:

$$u(t) = -\alpha(t)V_M \operatorname{sign}\left(y_1(t) - \frac{1}{2}y_{1M}\right),$$

$$\alpha(t) = \begin{cases} \alpha^* & \text{sí } \left[y_1(t) - \frac{1}{2}y_{1M}\right] [y_{1M} - y_1(t)] > 0 \\ 1 & \text{sí } \left[y_1(t) - \frac{1}{2}y_{1M}\right] [y_{1M} - y_1(t)] \leq 0 \end{cases}$$

Donde  $y_{1M}$  es el valor extremo de la función  $y_1(t)$ , tal que corresponda al valor de cero en  $\dot{y}_1 = y_2$ .

Los valores de  $\alpha$  y  $V_M$  son definidos como:

$$\begin{cases} \alpha^* \in (0,1] \cap \frac{3\Gamma_m}{\Gamma_M} \\ V_M > \max\left(\frac{\Phi}{\alpha^* \Gamma_m}, \frac{4\Phi}{3\Gamma_m - \alpha^* \Gamma_M}\right) \end{cases}$$

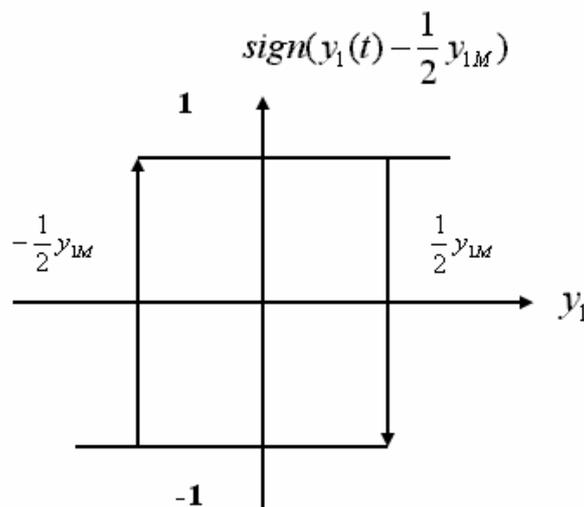
En otras palabras y para mayor simplicidad de lo anterior, podemos considerar a la variable  $\alpha=1$  y  $V_M$  como la ganancia del algoritmo.

Motivo por el cual, el algoritmo *Suboptimal* puede ser manejado con la siguiente expresión:

$$u(t) = -V_M \operatorname{sign} \left( y_1(t) - \frac{1}{2} y_{1M} \right)$$

Por otro lado, al realizar un análisis en el comportamiento de estado estable del algoritmo se asume que este presenta un movimiento periódico y simétrico, además, de que en ese punto existe un cambio de signo, es decir, existen valores positivos y negativos de la variable  $y_{1M}$  de igual magnitud, razón suficiente para considerar que el control Suboptimal es un control discontinuo con histéresis negativa.

Gráficamente de lo que se habla es lo siguiente:



**Figura 2.3** Representación del algoritmo Suboptimal como histéresis negativa

De lo anterior se sabe que una histéresis negativa promueve una alta frecuencia de conmutación, objetivo buscado en el control para aproximarse más al comportamiento real de nuestro sistema. Sin embargo el control con las características anteriores actuará de forma suave, por lo que el problema del *chattering* será disminuido en gran proporción.



Otra manera de realizar la misma operación pero con diferente representación es:

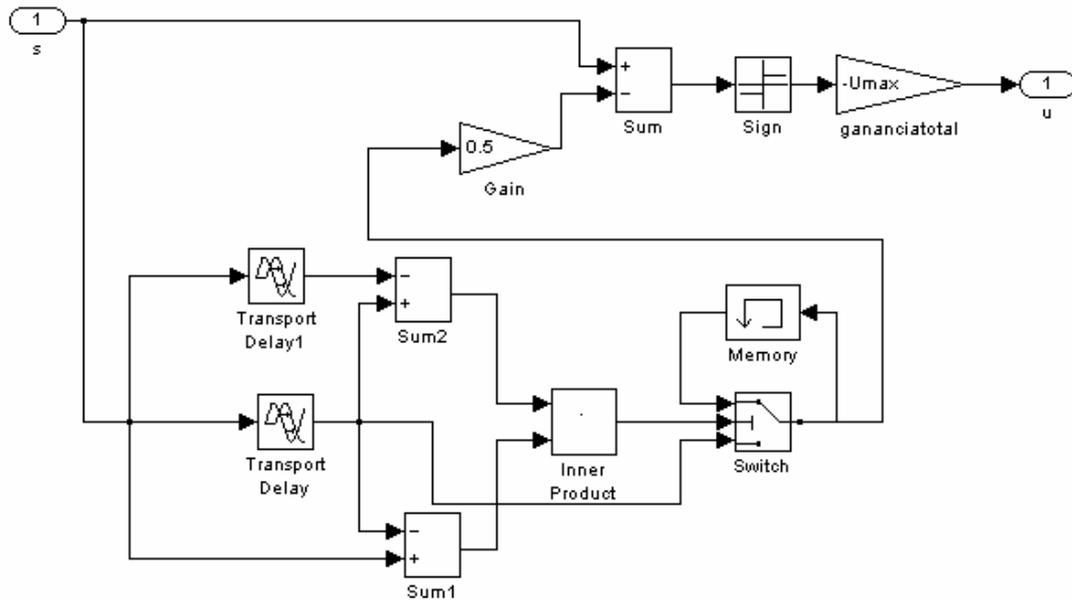


Figura 2.5 Estructura del algoritmo Suboptimal en Simulink.

La ventaja de éste diseño es que no se necesita conocer la derivada de la variable deslizante a la entrada del bloque, de lo que resultaría más sencillo desarrollar un sistema de control con esta estructura<sup>9</sup>. Más, sin embargo, el comportamiento de ambas estructuras es el mismo.

Para los fines que se tienen en la presente tesis, las simulaciones y la implementación experimental se llevarán a cabo con el primer bloque o estructura del algoritmo *Suboptimal* ya que permite introducir la aplicación de los diferenciadores basados en la teoría de modos deslizantes y analizar las características del desempeño del algoritmo con cada uno de ellos.

Por último, solo queda agregar que el algoritmo *Suboptimal* ofrece un control robusto con convergencia en tiempo finito, donde la velocidad de su respuesta radica en el valor de ganancia aplicado y, como ya se mencionó con anterioridad, su principio de funcionamiento es hacer que la variable deslizante y sus derivadas sucesivas tiendan al valor de cero.<sup>10</sup>

<sup>9</sup> Esta estructura se basa en el cálculo del valor máximo y mínimo de una señal

<sup>10</sup> en función de la superficie de conmutación predefinida.



## CAPÍTULO 3

# *Diferenciadores*

### *3.0.- Introducción*

Para la realización de un control *PID* es necesario conocer la derivada de la señal del error actuante para obtener una señal que sea proporcional a este y, esta señal se anticipe al comportamiento natural del error permitiendo una corrección significativa de la respuesta del sistema mejorando las características de estado estable y tiempo de asentamiento de nuestro sistema a controlar..

En el funcionamiento del algoritmo de modos deslizantes de primer orden (*FOSM*, por sus siglas en inglés) y de igual manera en la acción de control del algoritmo *Subóptimo* en su versión como filtro, al determinar la forma de la superficie de deslizamiento<sup>11</sup> se aprecia que es necesario encontrar la derivada del error, o bien, la señal a controlar para su funcionamiento. Así mismo, para la acción de control del algoritmo *Subóptimo*, es necesario conocer la derivada de la señal de control, debido al diseño del mismo.

Por tal motivo es importante tener un diferenciador que presente un buen desempeño al trabajar en tiempo real y que cumpla con las características de exactitud, precisión y convergencia en tiempo finito.

En este capítulo se estudiarán los resultados de una variedad de diferenciadores que cumplen con estas características para la realización de las simulaciones y experimentos de implementación desarrollados en la presente tesis.

Muchas son las formas de obtener una diferenciación, como es bien conocido existen una gran cantidad de métodos numéricos que nos permiten obtener la derivada de una señal, como también existe el diseño de filtros lineales que cumplen con este fin.

---

<sup>11</sup> Este tipo de control incorpora las características del FOSM, pero al final se integra el control, promoviendo una acción mas suave en su desempeño.



Algunos de los diferenciadores en estudio en el presente trabajo están basados en la teoría de modos deslizantes los cuales cumplen con las características antes mencionadas: precisión y convergencia en tiempo finito.

### 3.1.-Análisis de los diferenciadores propuestos

En esta sección se presentarán tres diferenciadores: *Diferenciador SM* [18], *Diferenciador de Levant* [19], *Diferenciador de Shtessel* [20].

La forma de onda del estimado de la señal de entrada, así como, la diferenciación dada una señal de entrada, será diferente según sea el algoritmo de modos deslizantes empleado.

La aproximación de la derivada al valor real y el nivel de *chattering* depende de una buena selección de los parámetros que conformen el algoritmo de diferenciación.

#### 3.1.1.-Diferenciador SM

El diferenciador *SM* (Sliding Modes, por sus siglas en inglés), esta basado en el principio de funcionamiento del algoritmo de modos deslizantes de primer orden.

Las ecuaciones que describen su funcionamiento son:

$$\begin{aligned}z &= v \\v &= \lambda \operatorname{sign}(x_1 - z) \\v_f &= a(v - v_f)\end{aligned}$$

La estructura anterior trata de estimar la derivada de la señal de entrada  $x_1$ , pero en presencia de la función “*sign*”,<sup>12</sup> la señal diferenciada  $v$  presenta un gran número de armónicas de alta frecuencia, un filtro lineal de primer orden con característica de filtro paso bajas es agregado al final de la estructura para disminuir el problema.

<sup>12</sup> La función sign regresa el signo de su argumento

La frecuencia de corte del filtro paso bajas esta determinada por el parámetro  $a$  y la salida del filtro es la variable  $v_f$ .

Para el ajuste del parámetro  $\lambda$  es necesario conocer el valor máximo de la derivada a estimar, pues en la práctica un valor muy bajo de  $\lambda$  provocaría que la señal diferenciada este muy lejos del valor real y, un valor muy grande de  $\lambda$  amplificaría el ruido de alta frecuencia provocado por la función *sign*.

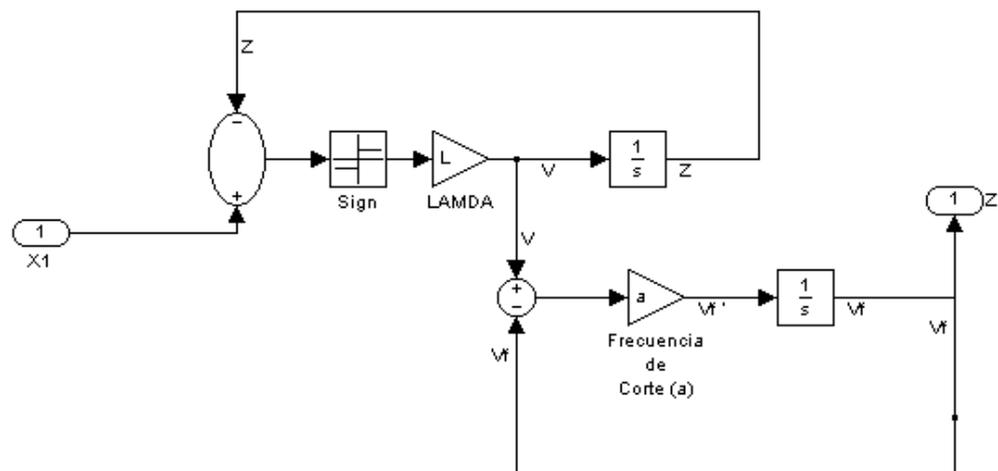


Figura 3.1 Diagrama en Simulink del Diferenciador SM

### 3.1.2.-Diferenciador de Levant

El siguiente diferenciador fue propuesto por Arie Levant y esta basado en el algoritmo *super-twisting* de modos deslizantes de orden superior (*HOSM*, por sus siglas en inglés), el cual queda definido por la siguiente ecuación:

$$\begin{aligned} \dot{z}_0 &= z_1 - \lambda_0 |z_0 - x_1|^{\frac{1}{2}} \text{sign}(z_0 - x_1) \\ \dot{z}_1 &= -\lambda_1 \text{sign}(z_0 - x_1) \end{aligned}$$

Las constantes  $\lambda_0$  y  $\lambda_1$  son los parámetros de ajuste de nuestro diferenciador para los que se ha planteado una técnica de sintonización dependiendo del valor máximo de la señal a estimar, es decir:

$$|f^{(n+1)}| \leq L \quad \text{sí} \quad n = 1 \quad |f'| \leq L$$

donde

$$\lambda_1 = 1.1L$$

$$\lambda_0 = 1.5L^{\frac{1}{2}}$$

La salida del diferenciador es la variable  $z_0$  la cual depende directamente de la discontinuidad del término  $\lambda_0 \left| (z_0 - x_1) \right|^{\frac{1}{2}} \text{sign}(z_0 - x_1)$  por lo que se sugiere analizar esta estructura como la que se propone en el observador lineal basado también en este algoritmo pero sin el término  $-ax_2+bu$ , es decir:

$$\begin{aligned} \dot{z}_0 &= z_1 - \lambda_0 (z_0 - x_1) \\ \dot{z}_1 &= -\lambda_1 (z_0 - x_1) \end{aligned}$$

De lo anterior se puede considerar que la variable  $z_1$  es la señal diferenciada y esta no depende del término de discontinuidad antes mencionado sino de la salida de un integrador, de lo que figura una atenuación considerable de la presencia de señales de alta frecuencia encontradas en el término  $z_0$  (*chattering*).

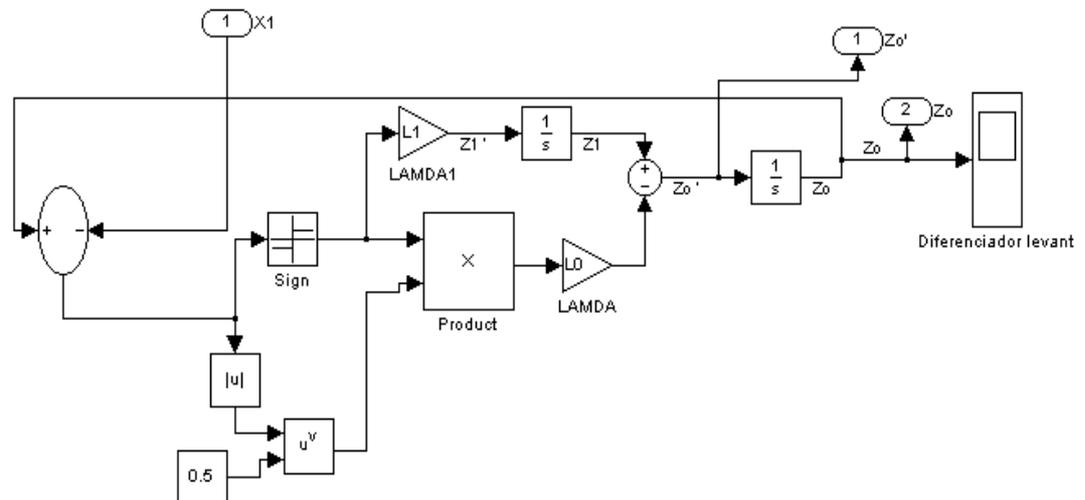


Figura 3.2 Diagrama en Simulink del Diferenciador de Levant

### 3.1.3.-Diferenciador de Shtessel

El tercer y último diferenciador en estudio es el diferenciador propuesto por Yuri Shtessel, también es conocido por el nombre de “*filtro SSZL*” el cual queda definido por las siguientes ecuaciones:

$$\begin{aligned} z_0 &= z_1 \\ z_1 &= -\rho_0 \operatorname{sign}(z_2 + e) \\ z_2 &= b_1 |e|^{\frac{1}{2}} \operatorname{sign}(e) - a_1 |(z_2 + e)|^{\frac{1}{2}} \operatorname{sign}(z_2 + e) \\ e &= (x_1 - z_0) \end{aligned}$$

Donde la señal diferenciada esta dada por la variable  $z_1$  y, como en el caso del diferenciador de Levant la salida proviene del integrador y no depende del término discontinuo por lo que también la característica de atenuación del *chattering* de alta frecuencia es notable. Cabe señalar que su comportamiento es similar al de un filtro lineal con característica de filtro paso-bandas, donde los parámetros de ajuste del diferenciador  $\rho_0$ ,  $a_1$  y  $b_1$  son muy difícil de sintonizar en la práctica por la naturaleza no lineal del mismo.

Para darnos una idea de cómo operan estos parámetros considere  $a_1$  como la frecuencia central del filtro paso-bandas y  $b_1$  como la calidad del filtro, por ultimo el parámetro  $\rho_0$  nos relaciona la velocidad de respuesta del filtro misma que depende del valor máximo de la señal estimada y del nivel de *chattering* que se desee que esta presente.

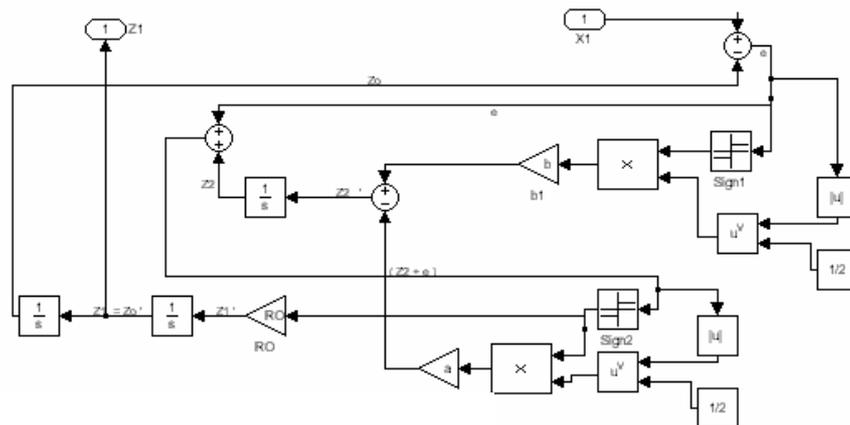


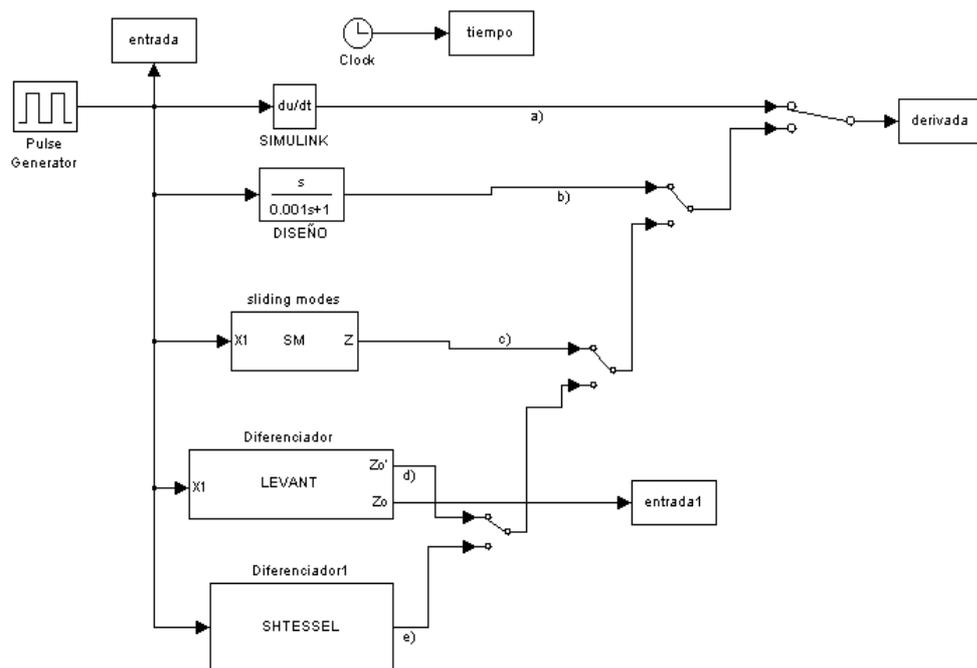
Figura 3.3 Diagrama en Simulink del Diferenciador de Shtessel

### 3.2.-Simulaciones<sup>13</sup>

En la siguiente sección se presentarán las simulaciones realizadas con los diferenciadores propuestos, así también un breve comparativo del bloque que Simulink de Mathworks, facilita para la diferenciación y otro más de la función de transferencia de un filtro lineal de primer orden el cual realiza la misma operación de diferenciación.

La duración de la simulación es de 1 [s], el paso de la muestra es de 10  $\mu$ s, el método de solución es por ODE1 (euler) y las variables de salida son graficadas con la misma base de tiempo.

En la siguiente figura se observa el diagrama en Simulink que se desarrolló para realizar las siguientes simulaciones.

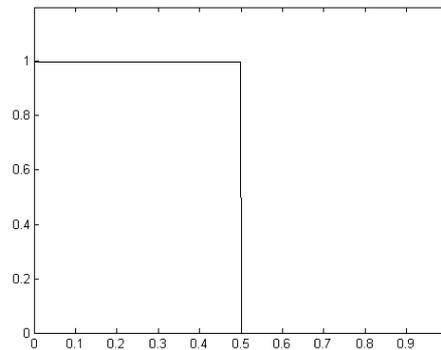


**Figura 3.4** Diagrama en Simulink de los Diferentes Diferenciadores: a) diferenciador de simulink, b) diferenciador por filtro lineal, c) **diferenciador SM**, d) **diferenciador de Levant**, e) **diferenciador de Shtessel**.

<sup>13</sup> Las simulaciones fueron desarrolladas a manera de bloque para facilitar su uso

Para el desarrollo de las siguientes simulaciones se aplicó a la entrada del bloque de diferenciación una señal cuadrada de amplitud 1 [V], periodo de 1 [s] y ancho del pulso del 50%.

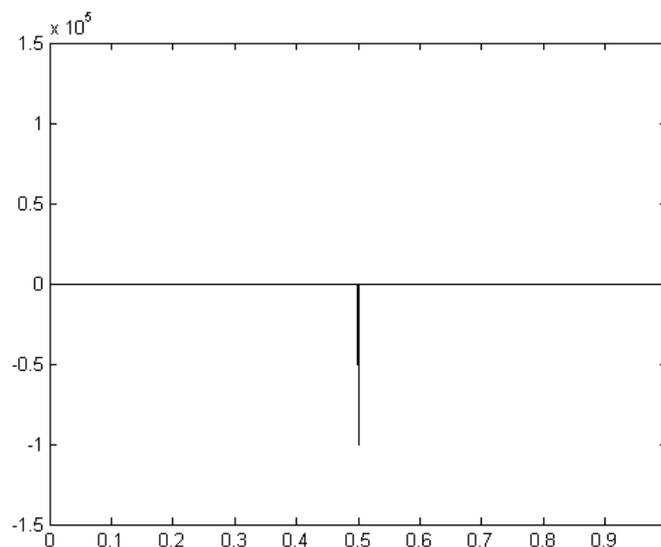
14



*Figura. 3.5 Señal aplicada por el Generador de Pulsos de Simulink*

### **3.2.1.-Diferenciador de Simulink para una señal cuadrada**

La señal que nos proporciona la diferenciación de este bloque nos da una idea de cual es la cota superior de nuestra derivada, dato de suma importancia para el resto de las simulaciones pues con el valor obtenido ajustaremos los parámetros del resto de los diferenciadores a estudiar para una mejor aproximación.



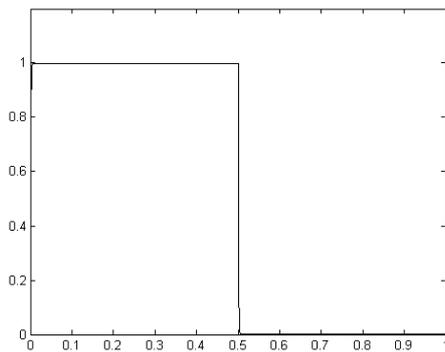
*Figura. 3.6 Señal Diferenciada por Derivador de Simulink.*

### **3.2.2.-Diferenciador por Filtro Lineal para una señal cuadrada**

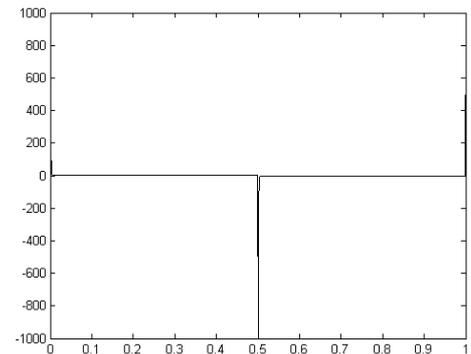
<sup>14</sup> Se escogió esta señal debido al gran número de armónicas que maneja para ver la respuesta de los diferenciadores en una amplia gama de frecuencias.

Para el diferenciador compuesto por la función de transferencia  $\frac{1000 s}{s + 1000}$  o bien,  $\frac{s}{0.001 s + 1}$  se puede apreciar que la señal recuperada es casi la misma aplicada a la entrada del bloque, con la presencia de un pequeño retardo.

Por lo que la señal estimada se puede considerar muy cercana al valor real.



**Figura 3.7** Señal Recuperada Derivador  $s / (0.001s + 1)$



**Figura. 3.8** Señal Diferenciada por Derivador  $1 / 0.001s + 1$

### 3.2.3.-Diferenciador SM para una señal cuadrada

La señal estimada de la entrada por el Diferenciador SM es un poco más lenta y esto se puede aclarar por la interacción del filtro paso-bajas y los parámetros seleccionados para su funcionamiento.

Los parámetros utilizados para esta simulación son: el valor de  $\lambda$  es de 100 y la frecuencia de corte del filtro paso-bajas es de 300, su selección fue experimental dado que a mayor valor del parámetro  $\lambda$  el nivel de *chattering* en la señal diferenciada aumentaba y la frecuencia de corte se tomo con ese valor ya que era más que suficiente para reconstruir la señal de entrada al ser pasada nuevamente por un integrador.

A continuación se muestra la señal diferenciada obtenida con estos parámetros, cabe señalar que el nivel de ruido de alta frecuencia tiene una amplitud no mayor a  $\pm 0.15$ [V].

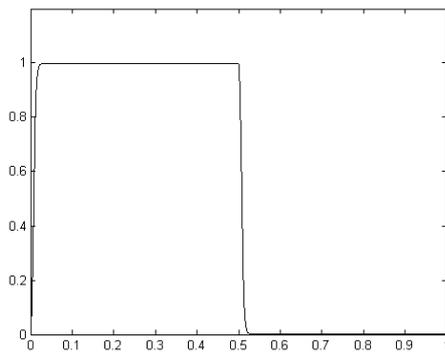


Figura. 3.9 Señal Entrada Recuperada por SM

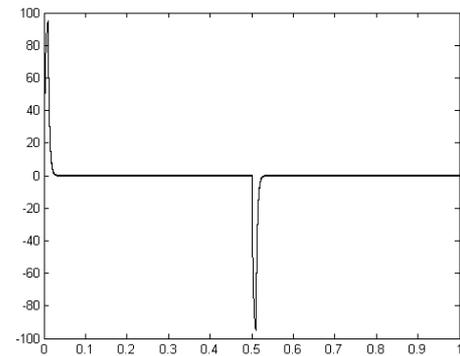


Figura. 3.10 Señal Diferenciada por Derivador SM

### 3.2.4.-Diferenciador de Levant para una señal cuadrada

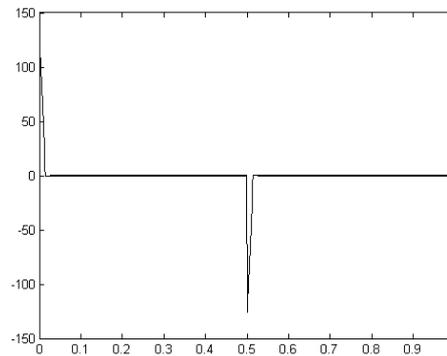
Ahora bien si observamos en la siguiente figura el desempeño del diferenciador de Levant se alcanza a apreciar que se tiene un tiempo de levantamiento menor en la señal diferenciada, mismo que provoca una mayor aproximación al valor real en la recuperación de la señal de entrada. Así también el nivel de *chattering* no era mayor a  $\pm 0.07$  [V].

Para que los parámetros empleados en el diferenciador presentaran una buena recuperación de la señal de entrada después de aplicado un integrador, un nivel de *chattering* bajo y mejor aproximación al valor real de la señal estimada se consideró el valor de  $1e^5$  como el valor de  $L$ , es decir:<sup>20</sup>

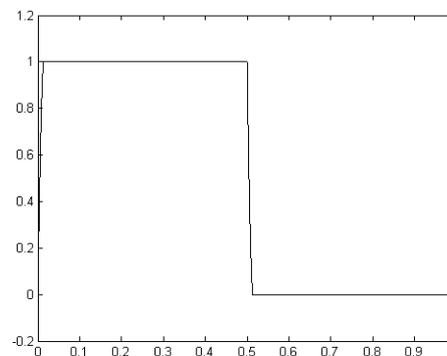
$$|f^{(n+1)}| \leq L \quad \text{sí} \quad n = 1 \quad |f'| = 100000$$

$$\text{donde} \quad \lambda_1 = \frac{1}{5} \lambda_0 \approx 126 \quad \lambda_0 = 2 * (100000)^{\frac{1}{2}} \approx 630$$

<sup>20</sup> Esta sintonización esta basada en el cálculo desarrollado en [18] para este diferenciador.



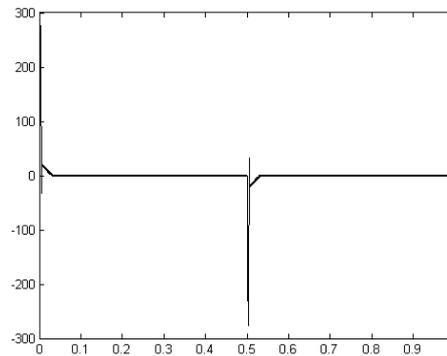
*Figura. 3.11 Señal Diferenciada por Derivador de Levant*



*Figura. 3.12 Señal Entrada Recuperada por Levant*

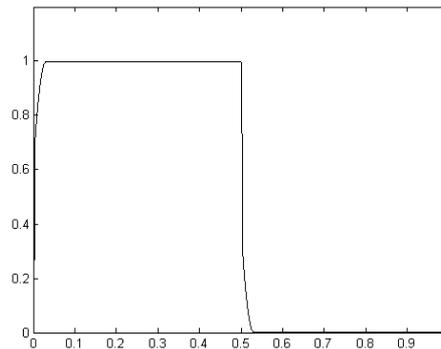
### **3.2.5.-Diferenciador de Shtessel para una señal cuadrada**

En esta última parte se presenta la señal diferenciada empleando al derivador de Shtessel el cual dada la complejidad de su estructura y, a un mayor número de parámetros de ajuste presenta gran dificultad para sintonizarlo, sin embargo, la recuperación de la señal de entrada después de aplicado un integrador, es muy aproximada al valor real, por lo que la derivada se puede considerar que también lo es. En la siguiente figura se presenta la diferenciación donde la amplitud del *chattering* es mayor que los anteriores pero, esto se debe a una no muy buena selección de los parámetros que lo rigen.



**Figura. 3.13** Señal Diferenciada por Derivador Shtessel.

En la siguiente figura se ve claramente como la señal de entrada es casi recuperada considerando que se emplearon los valores para  $\rho_0$  de 100000 ,  $a_I=400$  y  $b_I= 40$ , mismo que se seleccionaron con base a la cota máxima del diferenciador de Simulink.<sup>21</sup>



**Figura. 3.14** Señal Entrada Recuperada por Shtessel.

Para finalizar este capítulo y después de presentar una serie de ejemplos de operación de los distintos diferenciadores en estudio, se aclara que después de esta breve explicación, en el resto de esta tesis se trabajará con aplicaciones orientadas al objetivo principal dónde, todos y cada uno de los diferenciadores presentados en esta parte formaran parte esencial de los experimentos que se desarrollen en el motor de corriente directa, ya que una mejor aproximación de la señales diferenciadas que se necesitan, promueven un mejor control de nuestro sistema y así también, dadas las imperfecciones de operación del sistema en tiempo real, una disminución significativa de errores, reflejará un mejor desempeño de los algoritmos de control aplicados.

<sup>21</sup> Este valor corresponde a su vez al inverso del tiempo de muestreo del sistema.



## CAPÍTULO 4

# *Simulaciones en el Motor de CD*

### **4.0.- Introducción**

Una simulación de un modelo Simulink puede llevarse a cabo mediante la invocación de comandos en modo texto desde la ventana de MATLAB o bien en modo interactivo desde los menús del interfaz gráfico del propio Simulink. Describiremos esta segunda opción, ya que es más intuitiva y es suficiente para nuestro propósito.

Antes de ordenar a Simulink que simule nuestro modelo de sistema, es necesario configurar una serie de opciones que le indiquen cómo hacerlo. Las únicas que nos van a interesar en principio son las que se encuentran en la pestaña *solver* del menú *simulation/parameters*. En esta pantalla elegimos básicamente dos cosas: el tiempo de simulación y el resolutor (o motor de simulación) a utilizar, junto con los valores de los parámetros asociados al resolutor elegido (el conjunto de parámetros es distinto para cada uno).

El tiempo de simulación (*simulation time*) establece el tiempo de comienzo y fin de la simulación. Tiempo en términos de la propia simulación, que no coincide con el transcurso del tiempo real: si los cálculos implicados en la simulación son sencillos, simular 20 segundos de experimento puede tardar tan sólo 1 segundo. O al revés si los cálculos son muy complejos.

En cuanto a la elección del resolutor, ésta dependerá del tipo concreto de modelo con que estemos tratando. Debido a la diversidad de comportamientos de los sistemas dinámicos, Simulink nos da a elegir entre una amplia gama de resolutores especializados en un tipo de comportamiento particular. Una primera cuestión a elegir sobre el resolutor es si será de paso fijo o de paso variable. Los de paso variable pueden modificar durante la simulación su paso, lo que les permite ciertas ventajas como la posibilidad de detectar los cortes con cero de una señal. Los de paso fijo no aportan esas posibilidades, pero son los únicos que podemos elegir si queremos traducir el modelo a un programa ejecutable, por lo que nos centraremos en ellos. Para un resolutor de paso fijo sólo hay un

parámetro asociado a elegir: el tamaño de paso fijo (*fixed step size*) el cual depende del error máximo que se quiera tener también.

Al margen de que usemos un resolutor de paso fijo o variable, nuestro modelo puede tener o no estados continuos. Si no los tiene podemos usar el resolutor *discrete*, que no lleva a cabo ningún tipo de integración, y que está disponible en las dos versiones (paso fijo y paso variable). Si nuestro modelo sí contiene estados continuos, sí que necesita integración, por lo que será necesario elegir un resolutor no discreto.

Dentro de éstos se nos ofrecen múltiples opciones, entre las cuales hay uno genérico, que funciona bien con la mayoría de los problemas, que se llama *ode45* en el caso de paso variable y *ode5* en el caso de paso fijo. Está basado en la fórmula explícita Runge-Kutta . *ODE* es el acrónimo de *Ordinary Differential Equations* (resolutor de Ecuaciones Diferenciales Ordinarias).

#### 4.1.- El Motor de CD

En este capítulo se desarrollarán las simulaciones correspondientes al control de velocidad en el motor de corriente directa.

El lazo de control empleado para realizar esta tarea se puede esquematizar de la siguiente manera:

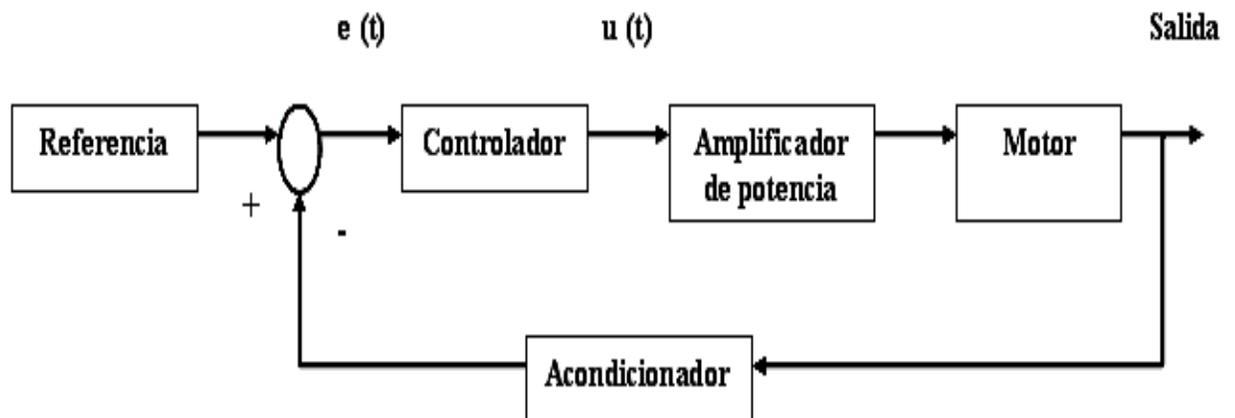


Figura 4.0 Lazo de Control para el Motor de CD



Este lazo de control se utilizará para ejemplificar las diferentes acciones de control descritas en los primeros tres capítulos, donde se tratará de resaltar las propiedades y características de cada controlador, así también, se introducirá la aplicación de los diferenciadores descritos en el capítulo anterior para apreciar su desempeño en las diferentes pruebas.

Los términos que se utilizaran de referencia para la realización de las comparaciones entre los algoritmos de control que se estudien serán: el tiempo de levantamiento, el tiempo de asentamiento y el porcentaje de sobrepaso.<sup>22</sup>

Se considerará como tiempo de levantamiento: el tiempo requerido para que la respuesta del sistema pase del 0 al 100% de su valor final; el tiempo de asentamiento como el tiempo que se requiere para que la curva de respuesta alcance un rango entre el 2 y el 5% del valor final y permanezca dentro de él y, el porcentaje de sobrepaso: como la relación del valor pico máximo de la curva de respuesta medido a partir del valor de estado estable de la respuesta, por cien.

Los parámetros aproximados del motor<sup>23</sup> con el que se trabajo son:

#### *Eléctricos*

$$R_a = 69.7 \text{ } [\Omega]$$

$$K_{bf} = 0.1111 \text{ } [V / (\text{rad /s})]$$

$$L_a = 4.458 \cdot 10^{-3} \text{ } [H]$$

#### *Mecánicos*

$$K_{bf} = 0.1111 \text{ } [N \cdot m / A]$$

$$J_{eq} = 8.86 \cdot 10^{-4} \text{ } [Kg \cdot m^2]$$

$$\omega = [\text{rad /s}]$$

---

<sup>22</sup> Ver [3]

<sup>23</sup> Son aproximados ya que fueron calculados de forma experimental ver [7]

Y la función de transferencia se puede obtener partiendo de la representación del motor de CD, para conseguirlo nos apoyaremos de la siguiente figura:

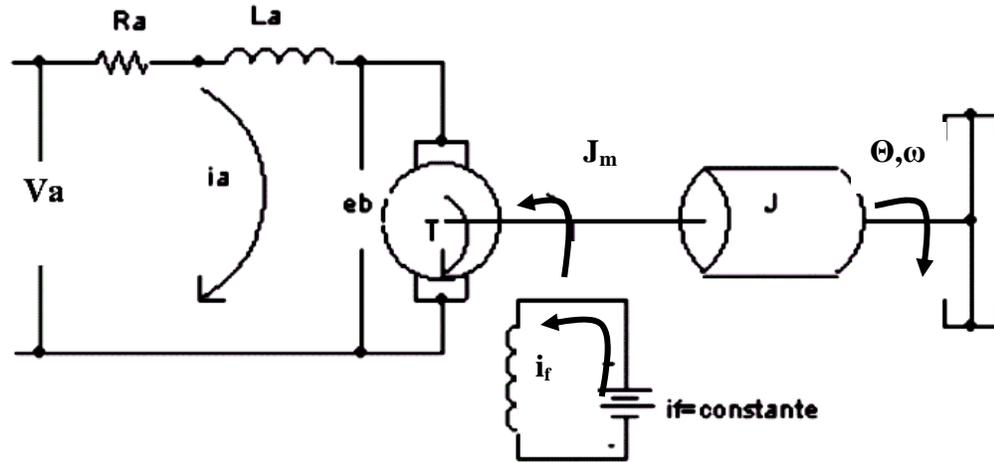


Figura 4.1 Motor de CD

donde

$R_a$  = Resistencia del devanado de inducido.

$L_a$  = Inductancia del devanado de inducido.

$i_a$  = Corriente del devanado de inducido.

$i_f$  = Corriente de campo.

$V_a$  = Tensión aplicada a la armadura.

$e_b$  = Fuerza contra - electromotriz.

$T$  = Par desarrollado por el motor.

$J_{eq}$  = Momento de inercia equivalente del motor y carga con referencia al eje del motor.

$f_{eq}$  = Coeficiente de fricción viscosa equivalente del motor y carga referido al eje del motor.

El modelo generalizado de un motor de **CD** queda expresado en las siguientes ecuaciones:

$$L_a \frac{d i_a(t)}{dt} = -R_a i_a - k_{bf} \omega(t) + v_a(t)$$

$$J_{eq} \frac{d \omega(t)}{dt} = k_{bf} i_a(t) - f_{eq} \omega(t)$$

y en términos de Laplace como

$$L_a s i_a(s) = -R_a i_a(s) - k_{bf} \omega(s) + v_a(s)$$

$$J_{eq} s \omega(s) = k_{bf} i_a(s) - f_{eq} \omega(s)$$

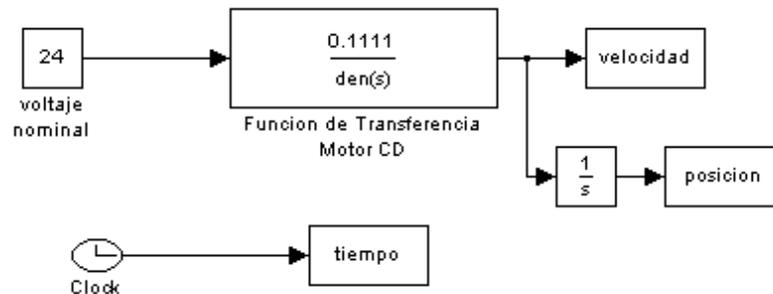
Ahora bien, solo queda definir la función de transferencia de interés

$$\frac{\omega(s)}{v_a(s)} = \frac{k_{bf}}{L_a J_{eq} s^2 + s(R_a J_{eq} + L_a f_{eq}) + f_{eq} R_a + k_{bf}^2}$$

sustituyendo los valores de nuestro motor y considerando que la  $f_{eq} = 0$

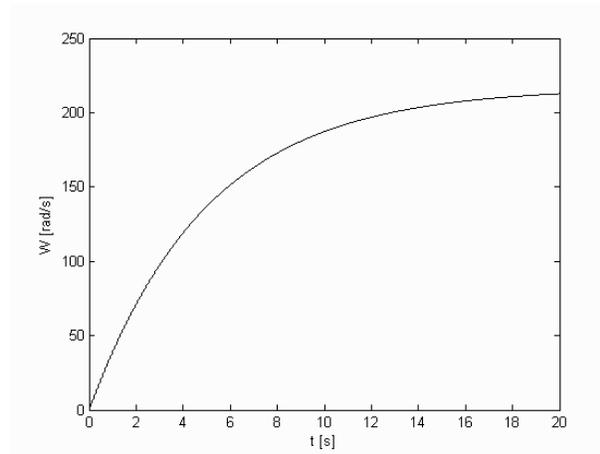
$$\frac{\omega(s)}{v_a(s)} = \frac{0.1111}{3.95 \times 10^{-6} s^2 + 0.0618 s + 0.0123}$$

La respuesta en lazo abierto de nuestro motor con una entrada de **24V**, el cual es el voltaje nominal que puede recibir, se puede apreciar en la Figura 4.3.



*Figura 4.2 Diagrama en Simulink del Motor de CD en lazo abierto*

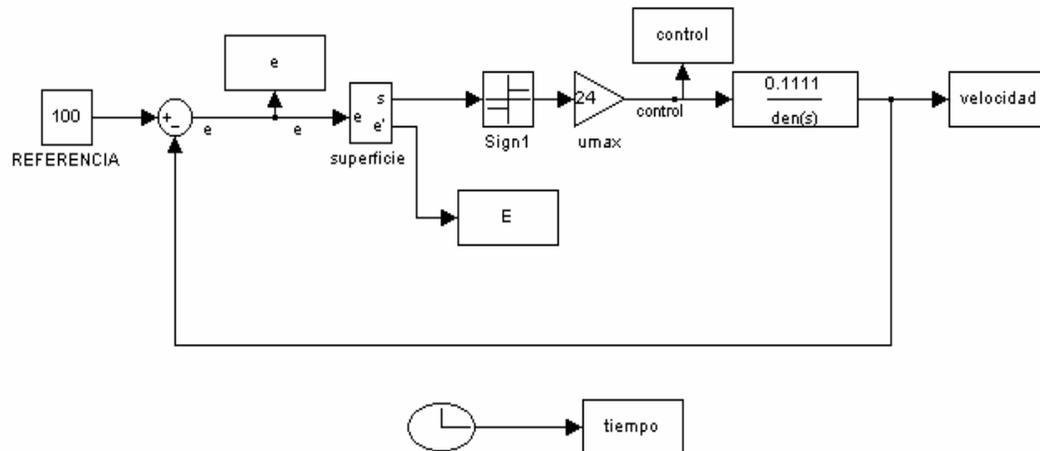
El desarrollo de esta simulación se realizó con el método de solución ODE 1 (euler), con una duración de 20 [s] y un tiempo de muestreo de 10 [μs]. De lo anterior obtendremos una gráfica que nos indicará la velocidad máxima en [rad /s] que el motor puede alcanzar bajo estas condiciones de operación.



**Figura 4.3** Respuesta en lazo abierto del Motor de CD con una entrada de 24 V.

## 4.2.-Control de velocidad con el algoritmo FOSM

El control de velocidad con el algoritmo de modos deslizantes de primer orden (*FOSM*, por sus siglas en inglés), queda representado por el siguiente diagrama en Simulink:



**Figura 4.4** Lazo de control en Simulink del controlador *FOSM*

Puede apreciarse que el Amplificador de potencia no está contemplado dentro de la representación de este lazo de control, esto es debido a que se considera para las simulaciones que existe una relación 1:1 en la acción del mismo.

Como se menciona en el primer capítulo, este algoritmo depende de la definición de la superficie de deslizamiento,  $s$ . Para lo cual  $e = \omega - \omega^*$  donde  $\omega$  es la velocidad de referencia y  $\omega^*$  es la velocidad estimada.

La expresión que define a  $s$  en función del error es  $s = C e + \dot{e}$ , superficie que define una línea recta en el espacio de estados, el valor de  $C$  para las siguientes simulaciones se considerara como de **10 000**, este valor nos dará una rapidez de convergencia razonable para el estudio que se quiere realizar con el algoritmo.

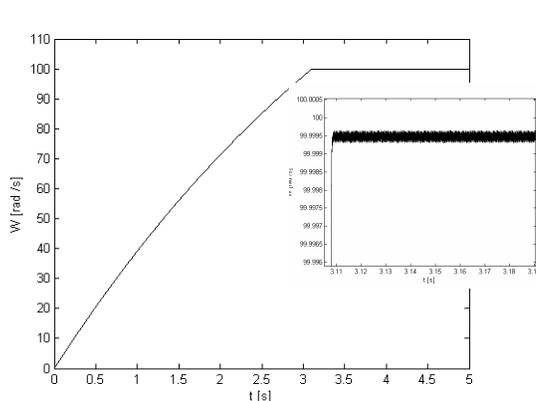
El control  $u$  aplicado será:

$$u = -24 \operatorname{sign}(e)$$

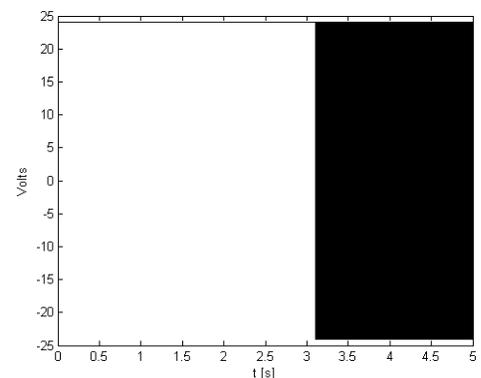
Por otro lado se aplicara una referencia a seguir de 100 [rad / s] y una ganancia máxima de 24 V al controlador pues este es el voltaje nominal que soporta nuestro motor, también se manejara un método de solución ODE 1(euler), un tiempo de muestreo de 10 [ $\mu$ s] con una duración máxima de la simulación de 5 [s].

Cabe aclarar que la primer prueba que se hace es utilizando como diferenciador el bloque proporcionado por Simulink para la obtención de la derivada del error para cumplir con la superficie  $s$  propuesta, el cual será cambiado con posterioridad para ver si existe alguna mejora en la respuesta de el sistema de control.

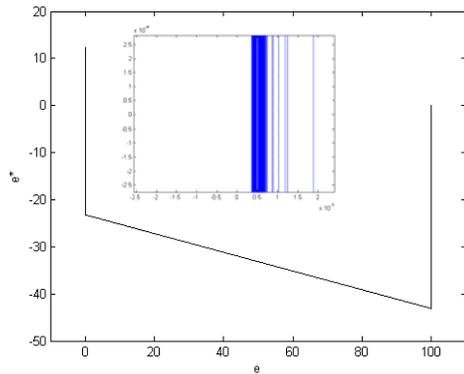
A continuación se mostraran las gráficas obtenidas de esta simulación dentro de las que encontraremos: la respuesta del sistema y el plano de fase, la trayectoria del error en el espacio de estados, la respuesta del control y, el fenómeno del *chattering* presente en el sistema.



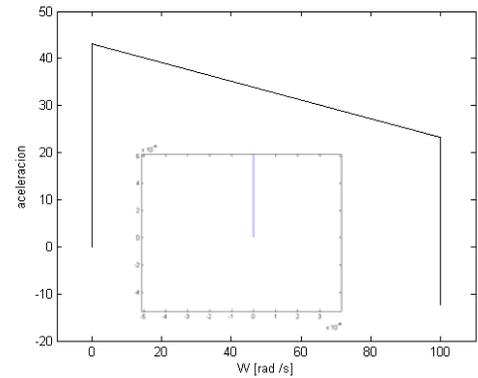
**Figura 4.5** Respuesta del lazo de control con el FOSM diferenciador de Simulink



**Figura 4.6** Respuesta del Control FOSM con diferenciador de Simulink



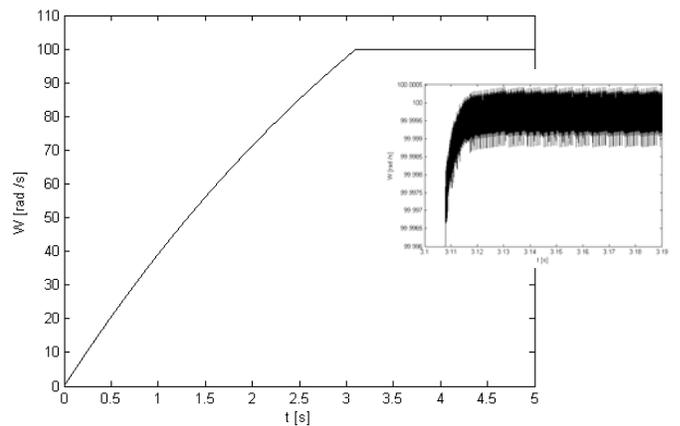
**Figura 4.7** Trayectoria del error en el espacio de estados



**Figura 4.8** Espacio de Estados del control FOSM

Ya concluida esta prueba ahora cambiemos el diferenciador de Simulink por uno de los basados en la teoría de modos deslizantes, conocido como el diferenciador **SM**.

El diferenciador conservara los parámetros que se trabajaron en el capítulo anterior para ver su desempeño con el algoritmo FOSM.



**Figura 4.9** Respuesta del FOSM con diferenciador SM

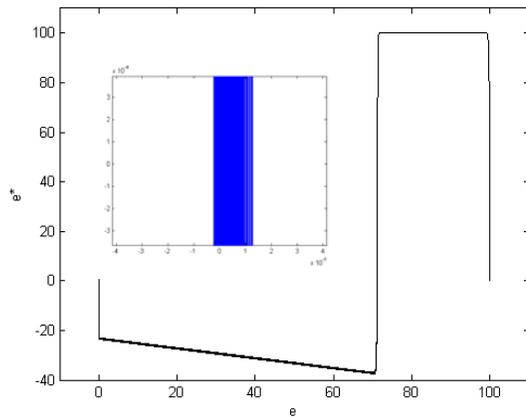


Figura 4.10 Trayectoria del error en el espacio de estados con diferenciador SM

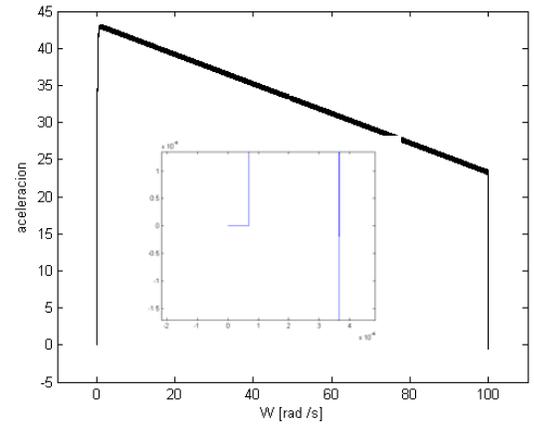


Figura 4.11 Espacio de estados del control FOSM con diferenciador SM

Cambiando nuevamente de diferenciador, al diferenciador de Levant obtenemos las siguientes gráficas:

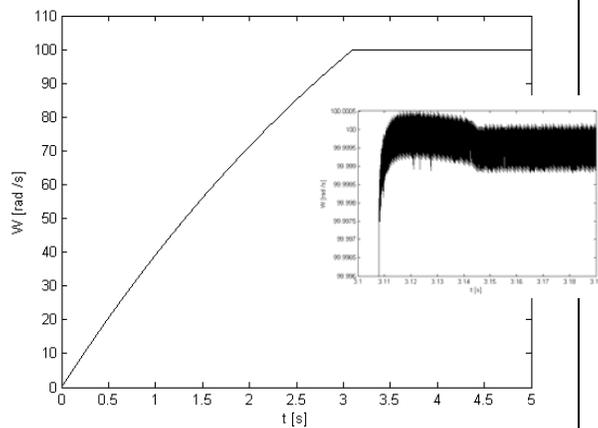


Figura 4.12 Respuesta del FOSM con diferenciador de Levant

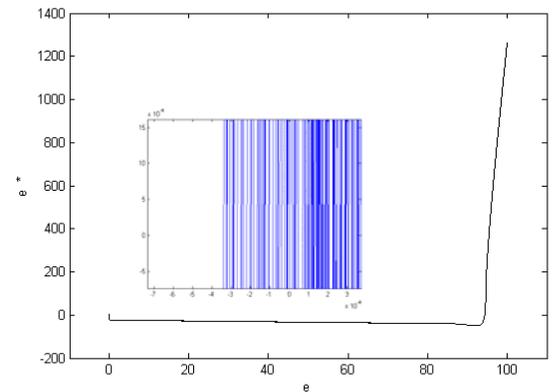


Figura 4.13 Trayectoria del error en el espacio de estados con diferenciador de Levant

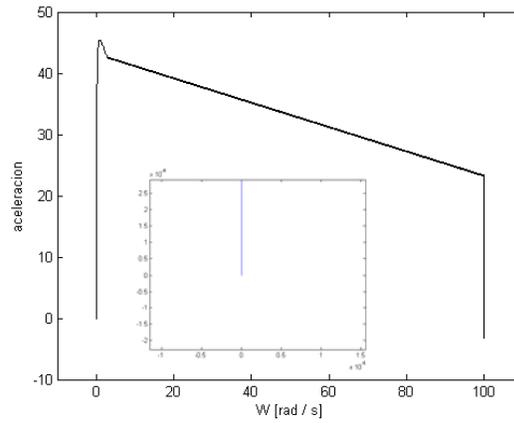


Figura 4.14 Espacio de estados del control FOSM con diferenciador de Levant

Una última prueba aplicada al **FOSM** se realizara con el diferenciador de **Shtessel** también llamado como filtro **SSZL**.

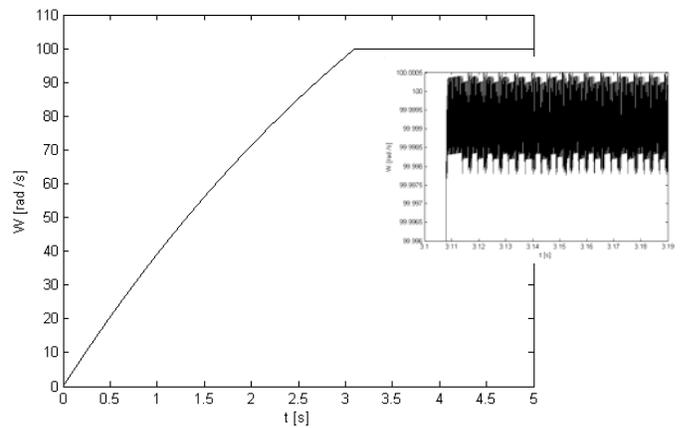


Figura 4.15 Respuesta del FOSM con diferenciador de Shtessel

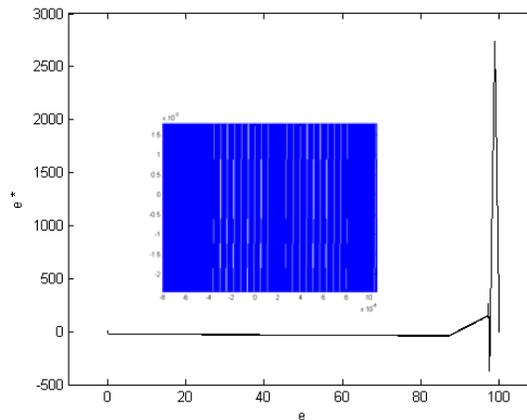


Figura 4.16 Trayectoria del error en el espacio de estados con diferenciador de Shtessel

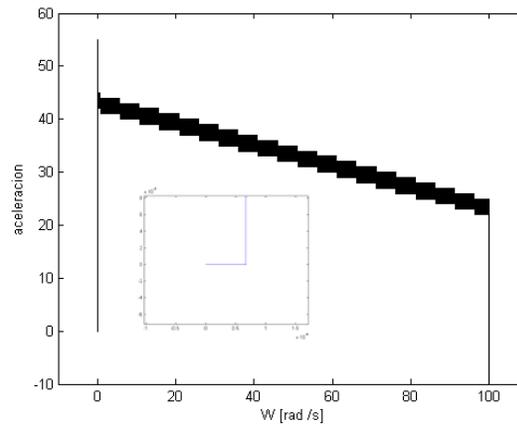


Figura 4.17 Espacio de estados del control FOSM con diferenciador de Shtessel

#### 4.2.1.-Análisis de los resultados obtenidos

De las gráficas anteriores se hará un análisis general del comportamiento de las trayectorias definidas en cada una de ellas, partiendo de que el objetivo principal es el control de velocidad y, así también, apreciar las características del desempeño del algoritmo *FOSM* y de los diferenciadores propuestos basados en la teoría de modos deslizantes.

En la **Figura 4.5, 4.9, 4.12 y 4.15** se tiene la respuesta del algoritmo *FOSM* donde la referencia de 100[rad /s] (misma que esta dentro del rango de velocidades alcanzado por el motor de CD) es alcanzada sin problemas a simple vista y en un tiempo finito y, que el tiempo de levantamiento es de aproximadamente 3 [s], en estas graficas no es evidente la presencia de algún sobrepaso y no existe la presencia a primera inspección de un error de estado estable.

De la **Figura 4.6**, se ve que la acción de control en alta frecuencia en efecto se alcanza cuando se llega a la superficie de deslizamiento y en un principio solo mantiene el voltaje nominal fijo en un valor positivo hasta que se empieza a deslizar en el valor dado como referencia, por otro lado se omitieron las gráficas de este tipo de respuesta para las otras pruebas ya que coincidían todas y cada una con lo anterior. La conmutación del control esta dada por un valor positivo de 24V y uno negativo de -24V, lo que físicamente nos dice que el motor cambiaria de sentido de giro a una muy alta frecuencia (teóricamente infinita) para mantener la estabilidad en el valor de referencia dado.

De las gráficas correspondientes a la representación del *chattering* presente en el sistema, **Figuras 4.5, 4.9, 4.12, 4.15**, notamos que en la **Figura 4.5** la amplitud de este fenómeno es muy



pequeña, con una amplitud de  $\pm 0.00025$  [rad/s], esto debido a que el tiempo de muestreo es de  $10[\mu\text{s}]$  y que la excitación de éste ocurre en aproximadamente  $3.11[\text{s}]$ , también es evidente que el valor de referencia de  $100[\text{rad/s}]$  no es alcanzado dada las imperfecciones de la aproximación real de nuestra derivada aunque de forma practica se considera que el promedio de la amplitud de este fenómeno nos dice el valor real logrado por la acción de control y, para este caso es de  $99.9995$  lo que nos dice que existe un error de  $0.0005\%$  con el valor establecido como referencia de velocidad.

Ahora bien, en la **Figura 4.9**, el fenómeno esta montado sobre el valor de referencia con una amplitud de  $\pm 0.0035$  [rad /s], manteniendo un valor promedio de  $99.99975$  originando un error del  $0.00025\%$  y excitando las dinámicas de alta frecuencia de nuestra planta poco antes de los  $3.1$  [s], esto debido a los parámetros con los que se permitió la operación del diferenciador **SM** pues si se hiciera un nuevo ajuste la amplitud de este fenómeno puede disminuir mejorando aun más la respuesta del sistema.

De la que se puede decir que al igual que en el caso anterior la respuesta esta montada sobre la referencia con un ligero sobrepaso dado por lo parámetros de ajuste de nuestro diferenciador y a la sensibilidad del mismo es la **Figura 4.12**, la cual muestra una amplitud en estado estable de  $\pm 0.0035$  [rad /s] y el sobrepaso de aproximadamente  $0.0005\%$  con un tiempo de asentamiento de no mas de  $3.13[\text{s}]$  permitiendo así una referencia alcanzada de  $99.99975$  y un error por las imperfecciones de nuestro ajuste del diferenciador de  $0.00025\%$  y, para el caso de la **Figura 4.15** solo nos que decir que aunque la amplitud es de  $0.0025$  [rad /s], la respuesta también esta montada en el valor de referencia de  $100$  [rad /s] y el valor alcanzado por el mismo es de  $99.9996$  originando un error de  $0.0004\%$  y nos garantiza, así al igual que los otros dos anteriores, que funciona muy bien el diferenciador bajo estas condiciones de operación aunque su ajuste no es el optimo dada la complejidad de su estructura.

De las trayectorias del error en el espacio de estados se puede apreciar como el algoritmo FOSM trata de llevar a la variable de deslizamiento y a su derivada al origen de nuestro plano sin importar las condiciones iniciales en las que se encuentre, aunque la forma de la trayectoria varia según el tipo de diferenciador empleado y la precisión de cada uno para el caso de la **Figura 4.7** el valor de  $(0,0)$ , es logrado en apariencia pero no se visualiza una estabilidad del mismo, caso contrario del resto de las graficas **Figuras 4.10, 4.13 y 4.16** don de el origen es logrado en tiempo finito debido a las propiedades de los diferenciadores empleados.



Por último solo queda comentar las graficas donde aparece la trayectoria de estado de la velocidad, es decir, donde se grafica a la (velocidad vs aceleración) **Figuras 4.8, 4.11, 4.14 y 4.17**, es muy fácil ver la convergencia en tiempo finito al origen del espacio de estados en la acción del algoritmo de control **FOSM** pero la importancia de estas graficas radica en ver cuales son las condiciones en las que trabajo nuestro controlador, en otras palabras, si en su acción de control se alcanzo la superficie de deslizamiento de forma brusca o en por el contrario de forma suave, asi mismo una valoración del nivel de chattering presente o si es que se llega solo a una vecindad de estabilidad o al punto fijado. Y solo se estudiaron 2 graficas en particular ya que en todas se ve que la acción del control de velocidad es de forma suave y por tiempo finito al origen del espacio de estados pero en la trayectoria del diferenciador **SM** (Figura 5.13) se nota una ligera presencia de ruido de alta frecuencia en su acción, donde para el caso donde se aplico el diferenciador de Levant este ruido es mas suave aun. (Figura 5.17)

De todo lo expuesto se puede decir que el control con el algoritmo **FOSM** que mejor desempeño tuvo es el trabajo en conjunto con el diferenciador de **Levant** para esta serie de pruebas por lo que solo quedaría agregar dos pruebas más, donde se aplicará únicamente este diferenciador y estas dos constan de probar que tan sensible es el controlador **FOSM** frente a perturbaciones externas o internas en el sistema y medir de esta manera su robustez la cual es una propiedad de los algoritmos de control por modos deslizantes.

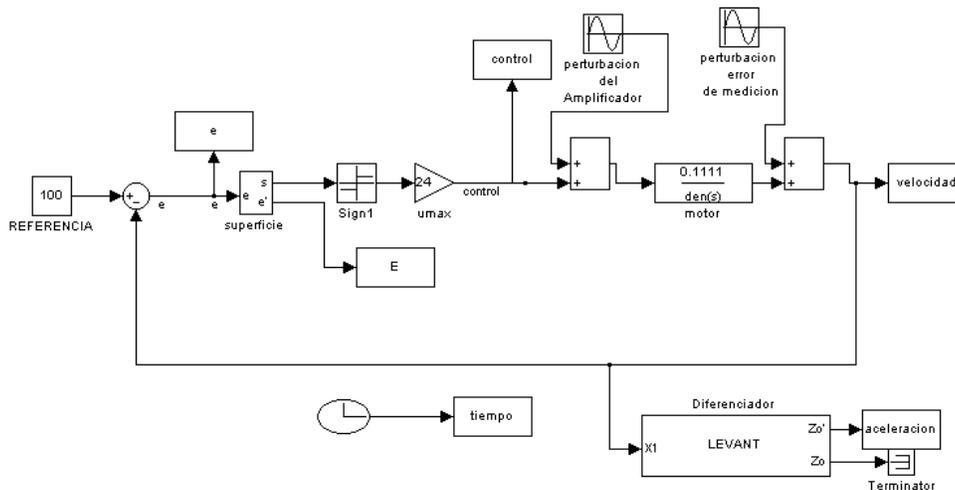
#### ***4.2.2.-Respuesta ante una perturbación Senoidal***

Las perturbaciones son señales no deseadas, que fácilmente se agregan a la dinámica de nuestro sistema de control provocando en su adherencia un comportamiento diferente del mismo, ya que son señales que modifican la acción de control cuando este es muy sensible, es de vital importancia tratarlas para nuestro estudio ya que se notará la robustez del control frente a estas.

Las dos pruebas que hacen falta por completar para el algoritmo **FOSM** son: que sucede cuando el amplificador de potencia en nuestro sistema real agregue una perturbación y la otra que sucede cuando existe un error en la medición de la variable de salida del motor de CD.

Para estas pruebas se manejará el siguiente diagrama en Simulink, donde la perturbación agregada después de la acción de control representará la perturbación de nuestro amplificador de

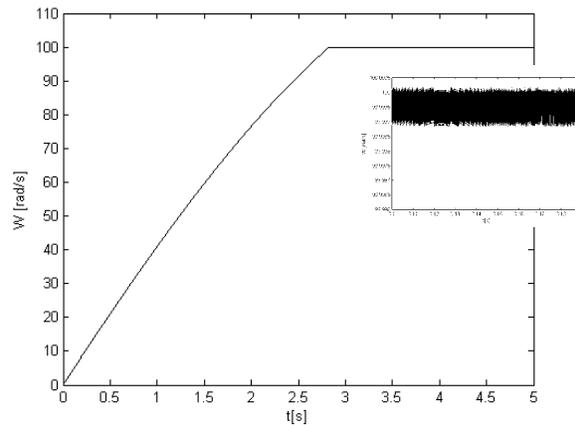
potencia y la perturbación añadida después de la planta nos relacionara el error de medición de la variable de salida que para nuestro caso es la velocidad del motor de CD.



**Figura 4.18** Control de velocidad con el algoritmo FOSM frente a perturbaciones

Si consideramos que la perturbación provocada por el amplificador de potencia es debida a una variación de su salida en un valor de  $\pm 10\%$  y que el motor se comporta como filtro paso bajas, podemos definir que la perturbación puede ser de 2.4V y una frecuencia de 1[rad /s], (ya que una frecuencia mayor de esta perturbación puede ser considerara como *chattering* provocado por el algoritmo de control) para ver su efecto en el lazo de control.

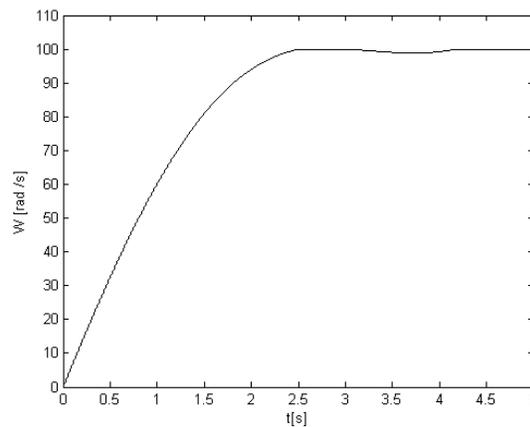
Sin la presencia de la segunda perturbación por lo que se considera que esta vale 0. Encontramos la respuesta del sistema a controlar como se muestra en la **Figura 4.23 y 4.24** donde se aprecia que el nivel de *chattering* aumento considerablemente y que el tiempo de levantamiento disminuyo dado que la energía aplicada a la planta excede el valor nominal de 24V de las pruebas anteriores y así mismo, que la referencia es alcanzada sin problemas y mantenida en un valor de 99.9995[rad /s], valor que se fijo como el promedio de nivel de ruido de alta frecuencia, por lo se puede decir que el control contrarresto la perturbación agregada de forma artificial, pero redujo su precisión en 0.00025% a la respuesta del sistema anterior.



**Figura 4.19** Respuesta del lazo de control con el FOSM frente a la perturbación del Amplificador

Si se aplica ahora la perturbación producida por un error de medición de  $\pm 25\%$  en la variable de salida de nuestro motor lo que en otras palabras sería una variación de la velocidad como de 25 [rad /s] y una frecuencia de 1 [rad /s] y que la primera perturbación vale cero.

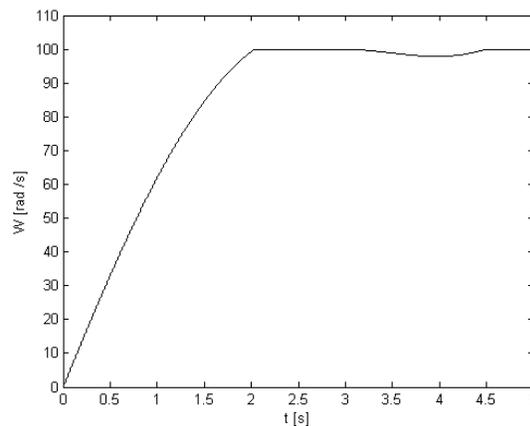
Se tiene que :



**Figura 4.20** Respuesta del lazo de control con el FOSM frente a la perturbación de error de medición

no existe sobrepaso en la respuesta, aunque el tiempo que le toma al control para contrarrestar la perturbación es mayor que en la prueba anterior y la referencia es llevada al estado estable por tiempo finito y alcanzando este valor ya no es sensible a la perturbación aplicada.

Para dar por terminada esta prueba se aplicarán las dos perturbaciones al mismo tiempo, este caso es el mas parecido a una aplicación practica ya que por lo regular los dos fenómenos tienen presencia en el control de velocidad dadas las imperfecciones de manejar el tiempo real y las características de nuestros equipos para realizar el control, además de que servirá como referencia el resultado obtenido para pruebas posteriores, así también, se verá con mayor claridad la robustez del lazo de control con el algoritmo *FOSM*.



*Figura 4.21* Respuesta del lazo de control con el FOSM frente a las 2 perturbaciones

### ***4.3.-Control de velocidad con el algoritmo Suboptimal***

Como ya se mencionó en un principio, lo que es de nuestro gran interés para este trabajo es ver las características de operación del algoritmo Suboptimal de modos deslizantes de segundo orden en su acción en el lazo de control de velocidad propuesto en la **Figura 4.0**, y su equivalente **Figura 4.27** en el que se harán los cambios correspondientes según sea la prueba para la obtención de las señales diferenciadas como fue realizada en la sección anterior con cada uno de los diferenciadores propuestos basados en la teoría de modos deslizantes y apreciar su desempeño en conjunto con el *Suboptimal* y en su versión *Suboptimal como filtro*.

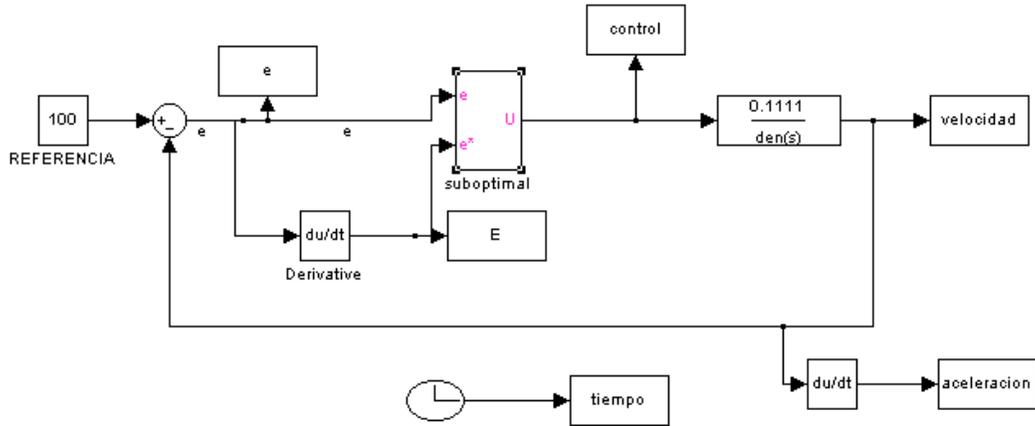


Figura 4.22 Control de velocidad con el algoritmo Suboptimal

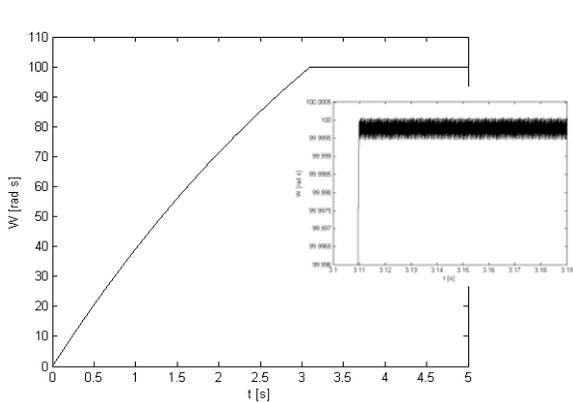


Figura 4.23 Respuesta del Suboptimal con diferenciador de Simulink

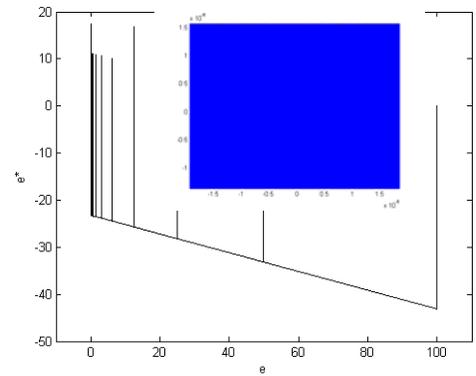


Figura 4.25 Trayectoria del error en el espacio de estados con diferenciador de Simulink

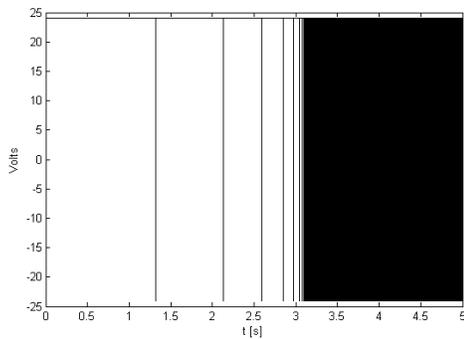


Figura 4.24 Respuesta del Control Suboptimal con diferenciador de Simulink

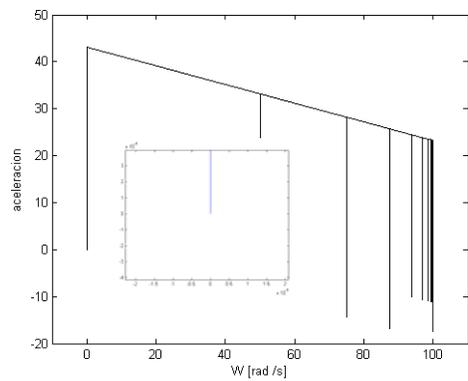


Figura 4.26 Espacio de estados del control Suboptimal con diferenciador de Simulink

Si cambiamos el bloque de diferenciación de Simulink por el diferenciador *SM*.

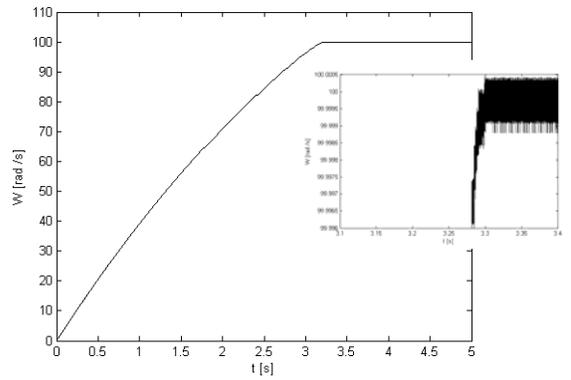


Figura 4.27 Respuesta del Suboptimal con diferenciador *SM*

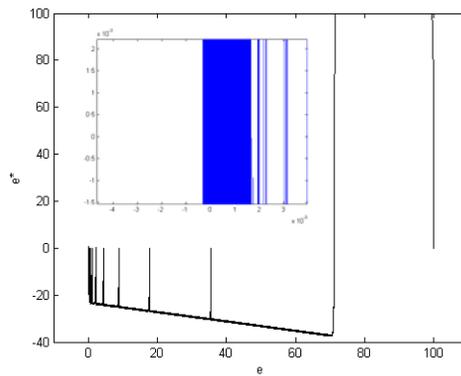


Figura 4.28 Trayectoria del error en el espacio de estados

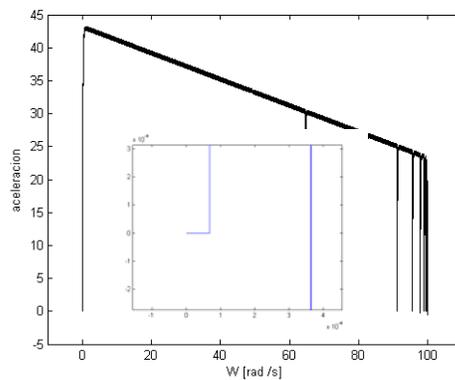
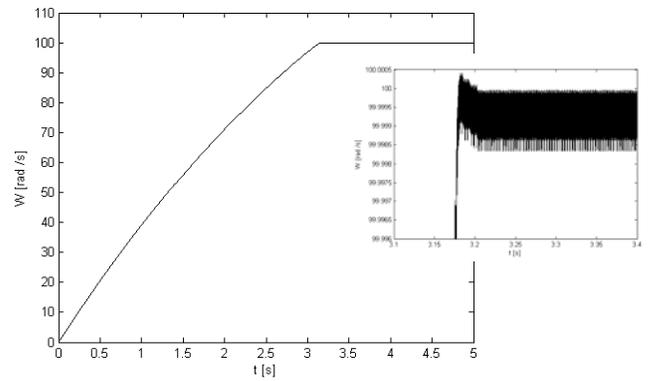
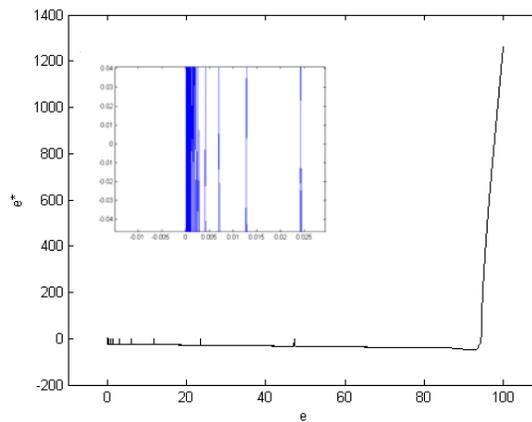


Figura 4.29 Espacio de estados del control Suboptimal con diferenciador *SM*

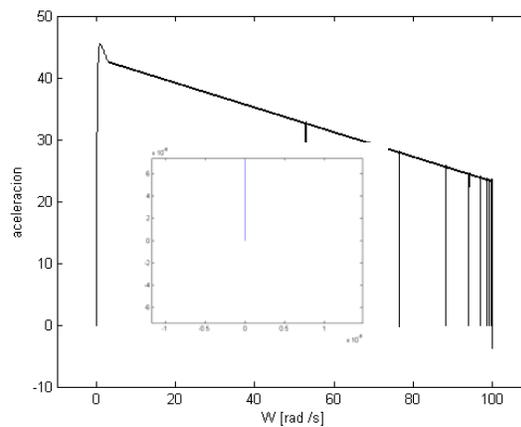
Modificando una vez mas el diferenciador para ver su desempeño con el algoritmo, se probará el diferenciador de *Levant*.



**Figura 4.30** Respuesta del Suboptimal con diferenciador Levant



**Figura 4.31** Trayectoria del error en el espacio de estados con diferenciador Levant



**Figura 4.32** Espacio de estados del control Suboptimal con diferenciador Levant

Por último aplicando el filtro *SSZL* o diferenciador de *Shtessel*

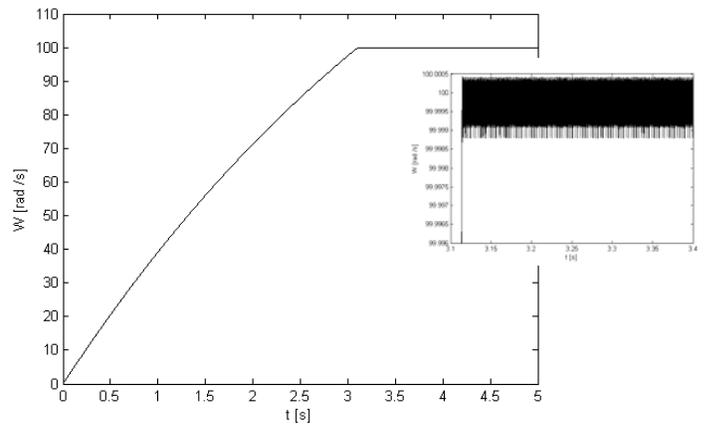


Figura 4.33 Respuesta del Suboptimal con diferenciador *Shtessel*

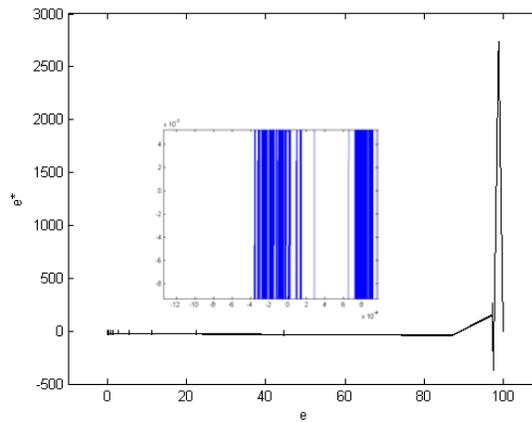


Figura 4.34 Trayectoria del error en el espacio de estados

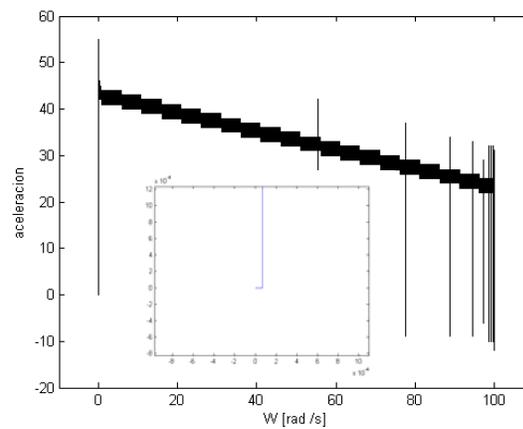


Figura 4.35 Espacio de estados del control Suboptimal



#### 4.3.1.-Análisis de los resultados obtenidos

De las gráficas anteriores se hará un análisis general del comportamiento de las trayectorias definidas en cada una de ellas, partiendo de el objetivo principal es el control de velocidad y catalogar el desempeño del algoritmo *Suboptimal* y de los diferenciadores propuestos basados en la teoría de modos deslizantes.

En las **Figuras 4.23, 4.27, 4.30 y 4.33** tenemos la respuesta del algoritmo *Suboptimal* donde la referencia de 100[rad /s] es alcanzada sin problemas a simple vista y en un tiempo finito y, que el tiempo de levantamiento varia de acuerdo al bloque de diferenciación empleado y este oscila entre aproximadamente de 3 a 4 [s], según sea el caso aunque, para el diferenciador de Shtessel este tiempo es mínimo con respecto a los demás 3.1 [s] aprox. Así mismo en todas las gráficas no es evidente la presencia de algún sobrepaso y no existe la presencia a primera inspección de un error de estado estable.

De la **Figura 4.24**, la acción de control en alta frecuencia en efecto se alcanza cuando se llega a la superficie de deslizamiento y en un principio solo mantiene la dinámica que caracteriza al algoritmo *Suboptimal* según la sección donde se estudio su comportamiento, variando la aplicación del voltaje nominal , hasta que se empiece a deslizar en el valor dado como referencia, por otro lado se omitieron las gráficas de este tipo de respuesta para las otras pruebas ya que coincidían todas y cada una con lo anterior. La conmutación del control esta dada por un valor positivo de 24V y uno negativo de -24V, lo que físicamente nos dice que el motor cambiaria de sentido de giro a una alta frecuencia, pero a tramos para mantener la estabilidad en el valor de referencia dado, lo que lo hace un control suave.

De las gráficas correspondientes a la representación del *chattering* presente en el sistema, **Figuras 4.23, 4.27, 4.30, 4.33**, de la **Figura 4.23** la amplitud de este fenómeno es muy pequeña, con una amplitud de  $\pm 0.0005$  [rad/s], esto debido a que el tiempo de muestreo es de 10[ $\mu$ s] y a las imperfecciones del bloque de Simulink para la obtención de la derivada, la excitación de éste fenómeno ocurre en aproximadamente 3.11[s], también es evidente que el valor de referencia de 100[rad /s] no es alcanzado del todo quedando por debajo sin tocar el valor fijado, de forma práctica se considera que el promedio de la amplitud de este fenómeno nos dice el valor real logrado por la



acción de control y, para este caso es de 99.99975 lo que nos dice que existe un error de 0.00025% con el valor establecido como referencia de velocidad.

Ahora bien, en la **Figura 4.27**, el fenómeno está montado sobre el valor de referencia con una amplitud de  $\pm 0.001$  [rad /s], manteniendo un valor promedio de 99.999625 originando un error del 0.00037% y excitando las dinámicas de alta frecuencia de nuestra planta poco antes de los 3.1 [s], y logrando el estado estable hasta los 3.3[s], esto debido a los parámetros con los que se permitió la operación del diferenciador **SM** pues si se hiciera un nuevo ajuste la amplitud de este fenómeno puede disminuir mejorando aun más la respuesta del sistema.

De la **Figura 4.30** se puede decir que al igual que en el caso anterior la respuesta está montada sobre la referencia con un ligero sobrepaso dado por los parámetros de ajuste de nuestro diferenciador y a la sensibilidad del mismo, la cual muestra una amplitud en estado estable de  $\pm 0.001$  [rad /s] y el sobrepaso de aproximadamente 0.0005% con un tiempo de asentamiento de no más de 3.3[s] permitiendo así una referencia alcanzada de 99.9995 y un error por las imperfecciones de nuestro ajuste del diferenciador de 0.0005% y, para el caso de la **Figura 4.33** solo nos queda decir que aunque la amplitud es de 0.00125 [rad /s], la respuesta también está montada en el valor de referencia de 100 [rad /s] y el valor alcanzado por el mismo es de 99.99965 originando un error de 0.00035% y garantizando, así al igual que los otros dos anteriores, su buen funcionamiento en estas condiciones de operación aunque su ajuste no es el óptimo dada la complejidad de su estructura, pero así también es el que menos error de estado estable presenta.

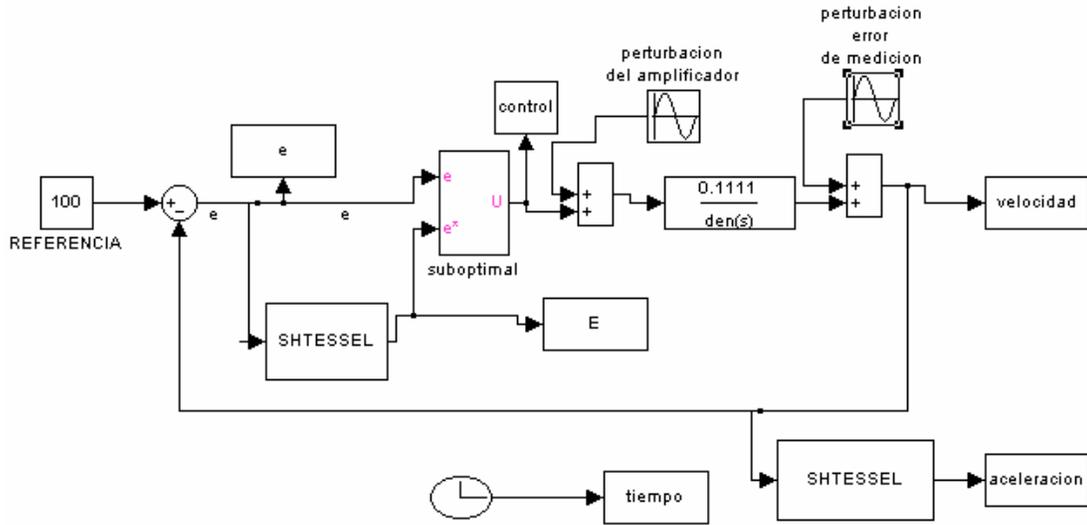
De las trayectorias del error en el espacio de estados podemos apreciar como el algoritmo **Suboptimal** trata de llevar a la variable de deslizamiento y a su derivada al origen de nuestro plano sin importar las condiciones iniciales en las que se encuentre, aunque la forma de la trayectoria depende directamente de la dinámica del algoritmo esta varía según el tipo de diferenciador empleado ya que en los gráficos es notable la precisión de cada uno y la manera de actuar es decir en el caso de la **Figura 4.25** el valor de (0,0), es logrado en apariencia pero no se visualiza una estabilidad del mismo, caso contrario del resto de las gráficas **Figuras 4.28, 4.31 y 4.34** donde el origen es logrado en tiempo finito debido a las propiedades de los diferenciadores empleados, como en la sección de las pruebas con el **FOSM**.



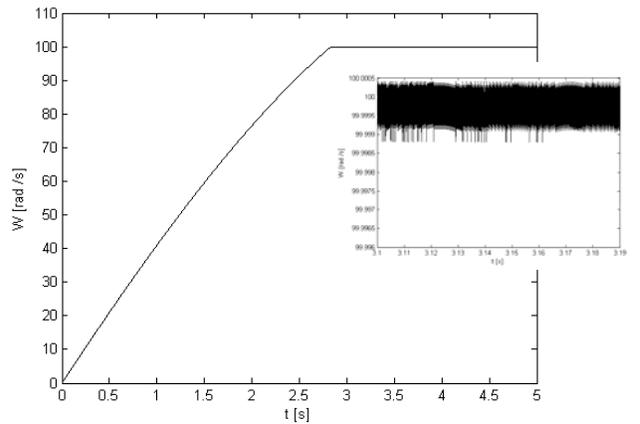
Por último solo queda comentar las gráficas donde aparece la trayectoria de estado de la velocidad, es decir, donde se grafica a la (velocidad vs aceleración) **Figuras 4.26, 4.29, 4.32 y 4.35**, es muy fácil ver la convergencia en tiempo finito al origen del espacio de estados en la acción del algoritmo de control *Suboptimal* pero la importancia de estas gráficas radica en ver cuales son las condiciones en las que trabajo el controlador, en otras palabras, si en su acción de control se alcanzó la superficie de deslizamiento de forma brusca o en por el contrario de forma suave, así mismo una valoración del nivel de *chattering* presente o si es que se llega solo a una vecindad de estabilidad o al punto fijado. Mencionado lo anterior se ve que la acción del control de velocidad es de forma suave y por tiempo finito al origen del espacio de estados pero en la trayectoria del diferenciador *Shtessel* (Figura 4.35) notamos una ligera presencia de ruido de alta frecuencia en su acción pero sin embargo, dadas las demás características que resaltan de este diferenciador y como lo que se busca es apreciar el desempeño de los mismos, este será el que elegiremos para la siguiente sección de pruebas del control *Suboptimal*.

#### **4.3.2.-Respuesta frente a una perturbación senoidal**

Realizando la prueba de la sensibilidad frente a perturbaciones equivalente a la empleada por el algoritmo *FOSM*, y utilizando el diferenciador *SSZL* de la sección anterior dado su desempeño para el algoritmo *Suboptimal* tenemos la siguiente prueba representada por el siguiente diagrama en Simulink donde primero se aplicará la perturbación provocada por el amplificador de potencia, es decir, la perturbación después del control misma que será de 2.4 volts y una frecuencia de 1[rad /s] dado el comportamiento de filtro paso bajas del motor y, posteriormente la perturbación agregada al sistema como error de medición con un valor de 25 [rad /s] a 1 [rad /s]. Finalmente la prueba más importante, las dos perturbaciones al mismo tiempo.

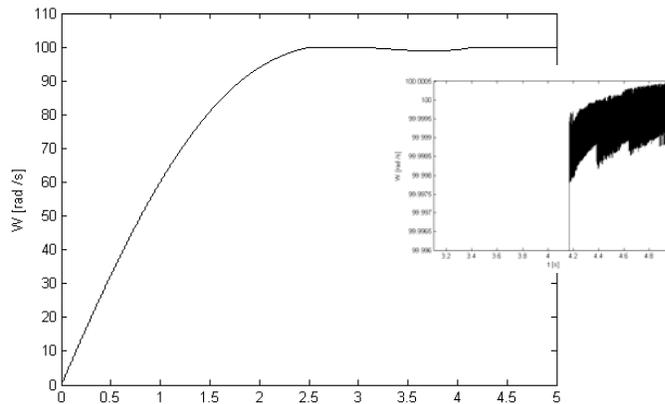


**Figura 4.36** Control de velocidad con el algoritmo Suboptimal frente a perturbaciones



**Figura 4.37** Respuesta del lazo de control con el Suboptimal frente a la perturbación del Amplificador

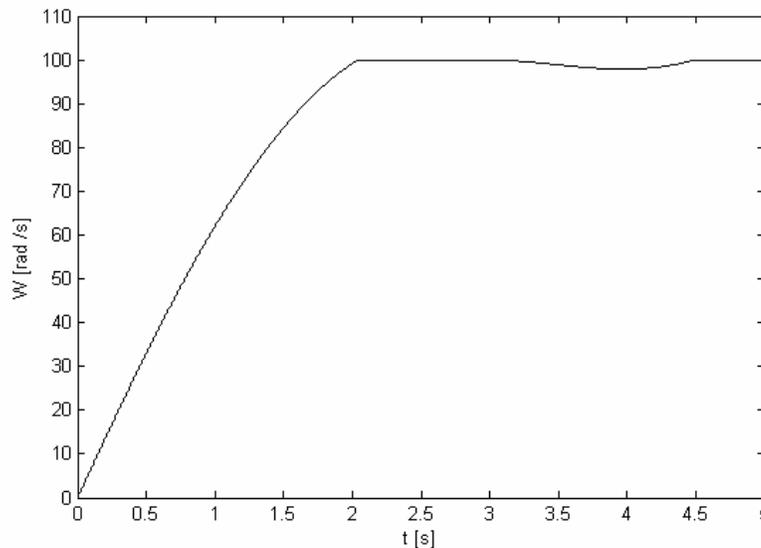
Aquí es claro que la perturbación del amplificador fue contrarrestada sin problemas y dado que se aplicó mayor cantidad de energía a la planta el tiempo de levantamiento es menor a los 3.1[s] de la prueba de la sección anterior.



**Figura 4.38** Respuesta del lazo de control con el Suboptimal frente a la perturbación de error de medición

Se ve que el tiempo que tarda el control en contrarrestar la perturbación de error de medición es mayor, mas sin embargo ya alcanzada la referencia esta ya no surte efecto en el lazo de control.

Ahora aplicando las dos perturbaciones al mismo tiempo caso de una situación de implementación, notaremos que el control es robusto y permite de igual forma contrarrestar el efecto de ambas volviéndose insensible después de un periodo de tiempo  $t$ .



**Figura 4.39** Respuesta del lazo de control con el Suboptimal frente a las 2 perturbaciones

#### 4.4.-Control Suboptimal como filtro

Una versión del control *Suboptimal* para mejorar sus condiciones de operación y disminuir en su acción el fenómeno de *chattering* es agregando un filtro lineal a la salida del control con característica de filtro paso-bajas, es decir, incrementando el orden del sistema de forma artificial ó bien integrando la conmutación de nuestra señal de control para que su efecto sea mas suave en su acción, esto es, si analizamos que la función *sign* es una función discontinua lo que se busca es cambiar el efecto de esta acción por una función continua más suave, como lo es una función de saturación.

Para fines prácticos se agregará solo un integrador a la salida del control para que funcione como filtro del mismo de ahí el nombre *Suboptimal como filtro* y las simulaciones estarán basadas en el bloque *Suboptimal* del **Capítulo 2** con la variante de que la señal de salida del bloque esta filtrada y además, se propone una superficie de deslizamiento  $s$ , basada en el error y que definirá una recta en el espacio de estados, como en el caso del control *FOSM*

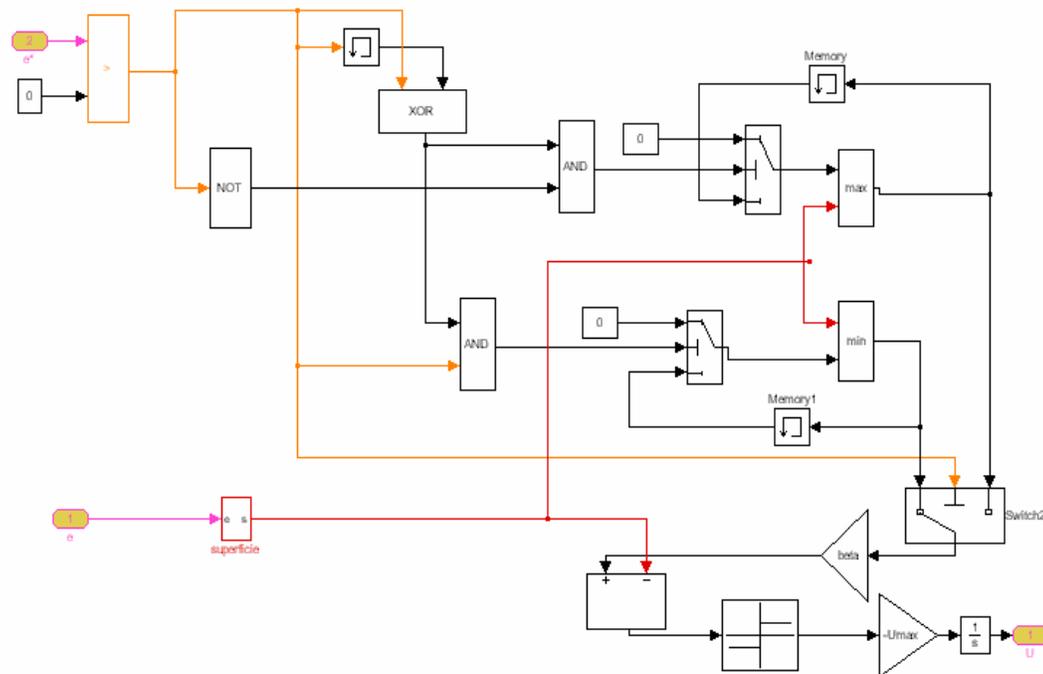
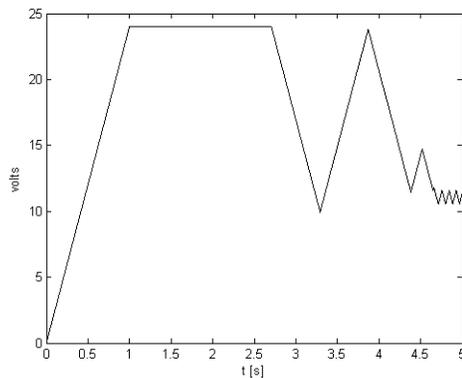


Figura 4.40 Estructura interna del control Suboptimal como Filtro

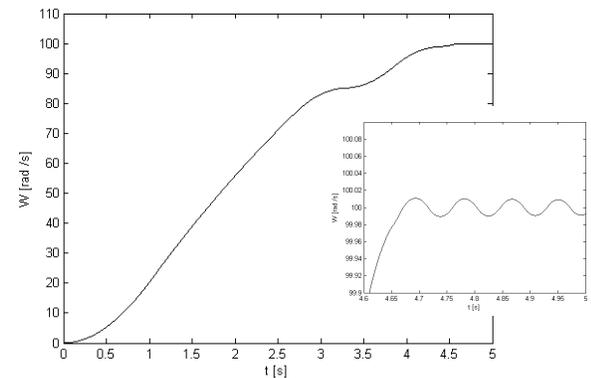
Las ventajas de este tipo de control son pocas ya que muchas de las características principales del control *Suboptimal* se pierden, algunas de estas son la reducción del orden, la insensibilidad frente a perturbaciones por mencionar algunas.

Las pruebas que se realizarán son precisamente las que muestren una disminución del *chattering* y la sensibilidad frente a perturbaciones, por lo que nos basaremos en el diagrama de la **Figura 4.0** y el equivalente en la **Figura 4.22**, así mismo, del diagrama de la **Figura 4.36** considerando en las pruebas que la señal de control esta filtrada.

Como puede verse en la figura siguiente si consideramos que antes teníamos pulsos cuadrados.



**Figura 4.41** Acción del control *Suboptimal* como Filtro

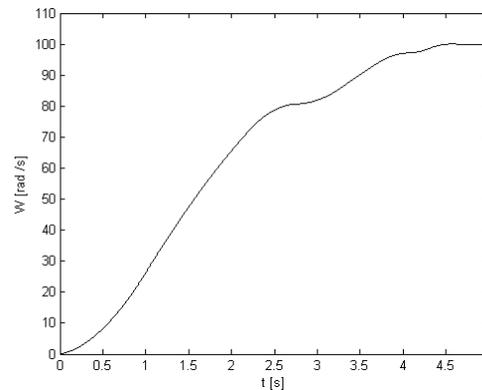


**Figura 4.42** Respuesta del lazo de control con el *Suboptimal* como filtro

Se puede ver que la respuesta es mas lenta y suave debido al efecto producido por el integrador y el valor de la constante  $C$  de la pendiente de la variable de deslizamiento, esto es importante ya que de esta manera no se excitan las componentes de alta frecuencia de nuestra planta, tales que provoquen el indeseable fenómeno de *chattering* en el sistema.

Como última prueba a esta versión del *Suboptimal* agregaremos las dos perturbaciones según el diagrama de la **Figura 4.36** y apreciaremos que al control de igual modo no le cuesta trabajo contrarrestar a ambas, pero con un pequeño detalle, la segunda perturbación fue de menor amplitud, es decir, que el error de medición que tolera este tipo de control es no mayor al 7%, de lo que se desprende que en efecto es mas sensible a perturbaciones.

La prueba fue hecha con las mismas constantes empleadas con el control *Suboptimal*, ya que en numerosas pruebas no expuestas, el valor de la primer perturbación no es significativo; caso opuesto para la segunda donde la amplitud fue de 5 [rad /s].

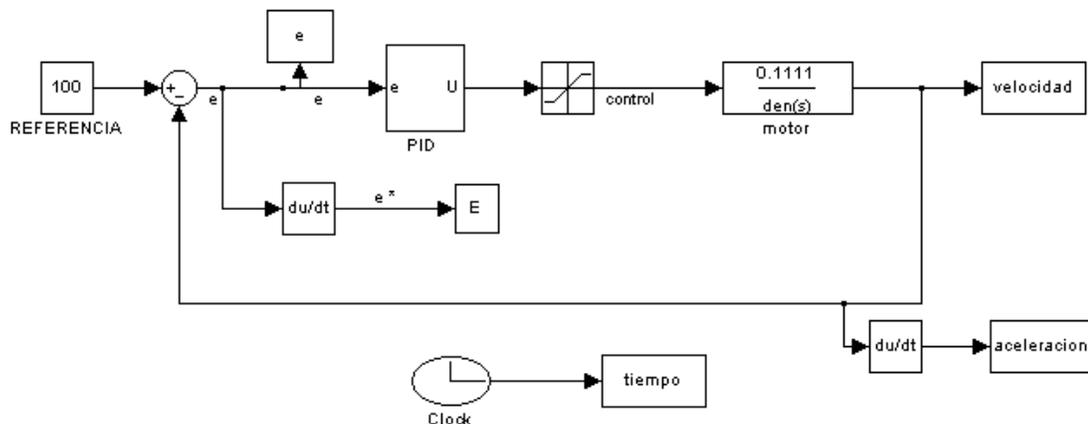


*Figura 4.43* Respuesta del lazo de control con el Suboptimal como filtro frente a las 2 perturbaciones

#### 4.5.-Control de velocidad con el algoritmo PID

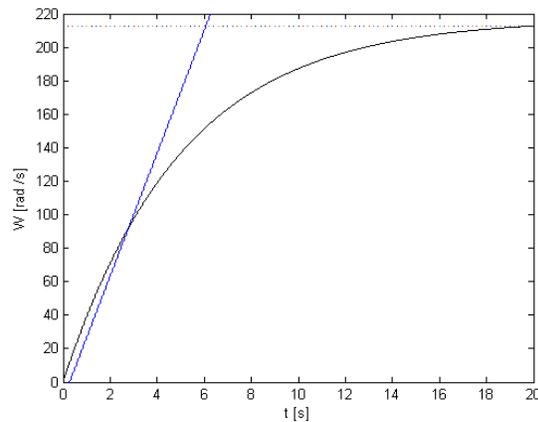
En esta sección se presentará el control de velocidad con el algoritmo PID, basado en la teoría de control clásico y con las características que se hablaron en el Capítulo 3.

Se tomará como base el diagrama de la Figura 4.0 en donde su equivalente para este control es el que se ve en la figura siguiente:



*Figura 4.44* Control de velocidad con el algoritmo PID

Para ajustar los parámetros de nuestro controlador utilizaremos la respuesta en lazo abierto de nuestro motor sin carga a una entrada escalón de 24[V] y basados en el segundo método de Ziegler-Nichols : la curva de reacción, donde se encontrará el punto de inflexión en la curva para la obtención de los valores de L y T, es decir:



*Figura 4.45* Curva de reacción para el ajuste del PID en el motor de CD

Los parámetros de sintonización, basados en el segundo método de ajuste de Ziegler-Nichols son:  $L = 0.25 [s]$ ,  $T = 6 [s]$ .

Y su vez el **PID** queda definido con las siguientes constantes según la **Tabla A.2**:

$$K_p = 28.8, T_i = 0.5 \text{ y } T_d = 0.125$$

Siguiendo con la serie de pruebas planteadas en este capítulo, empezaremos por analizar el comportamiento del bloque **PID** en el lazo de control a través de la respuesta del sistema , por lo que primero observaremos el desempeño del mismo en su acción con el diferenciador que proporciona Simulink y posteriormente se hará uso de los diferenciadores de modos deslizantes en estudio.

De la **Figura 4.46** se aprecia el desempeño de este control en términos generales, bondades e imperfecciones de su uso, ya que en comparación con los otros algoritmos empleados este es un control extremadamente suave y también, un poco mas lento en su acción, es decir, con otras palabras presenta un tiempo de levantamiento de 3.3 [s], también es notable es la presencia de un error en estado estable con un valor de 2 [rad /s], lo que equivale a un error del 2%, esto error tiende a disminuir de forma asintótica pero nunca será menor del 0.0003% dadas las condiciones de simulación.

El sobrepaso se considera del 2% y no existe ninguna presencia de componentes de alta frecuencia que este control haya excitado en la planta durante su acción.

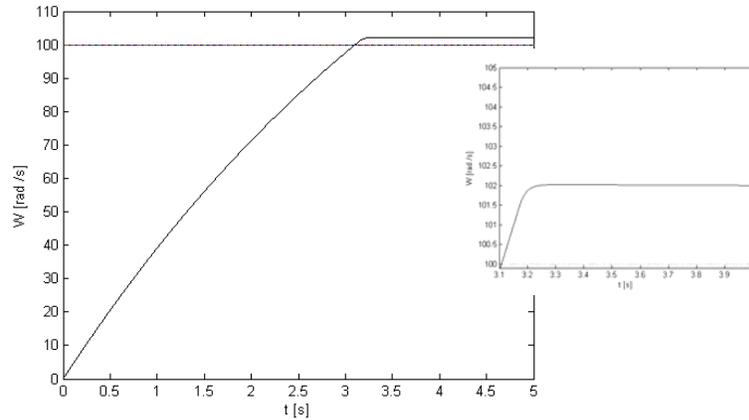


Figura 4.46 Respuesta del control PID

De las siguientes figuras, **Figuras 4.47 y 4.48**; de la primera observamos como el error tiende al origen de espacio de estados de forma asintótica dadas las condiciones iniciales de 100 [rad/s] como error máximo si se parte del reposo; de la segunda se aprecia como es el comportamiento del diferenciador en la obtención de la derivada y de cómo no se logra el origen del espacio de estados lo que indica la convergencia en el infinito de la acción de control.

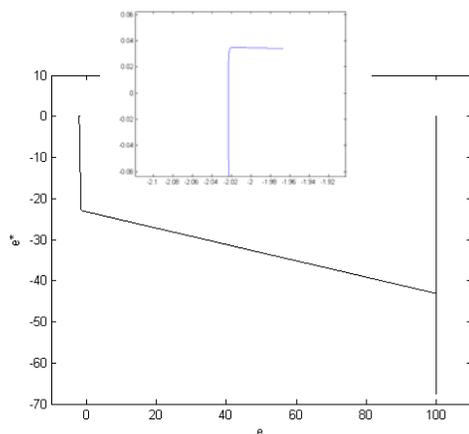


Figura 4.47 Trayectoria del error en el espacio de estados

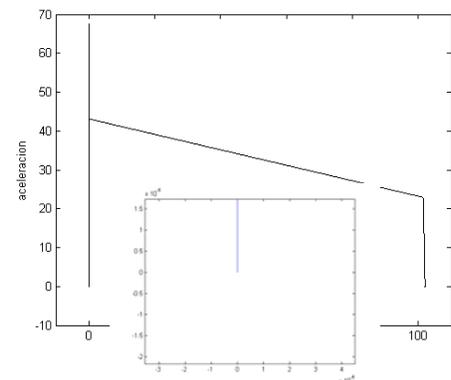
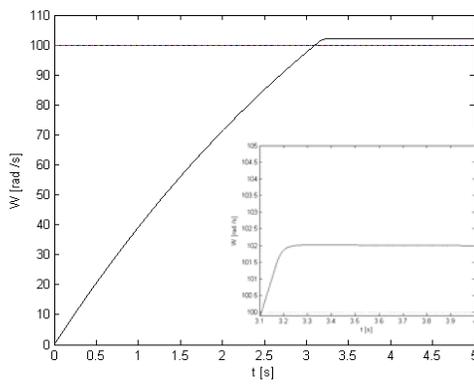


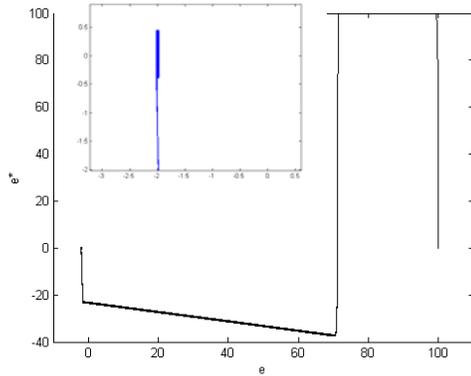
Figura 4.48 Trayectoria en el espacio de estados del control PID

En la siguiente sección solo se presentarán las respuestas del control **PID** con los diferentes diferenciadores propuestos basados en la teoría de modos deslizantes, mismas que responden al análisis anterior y que manifiestan un comportamiento muy parecido al que se presento en la sección de pruebas con el algoritmo **FOSM**, pero con la desventaja de presentar estabilidad asintótica y la no convergencia en tiempo finito.

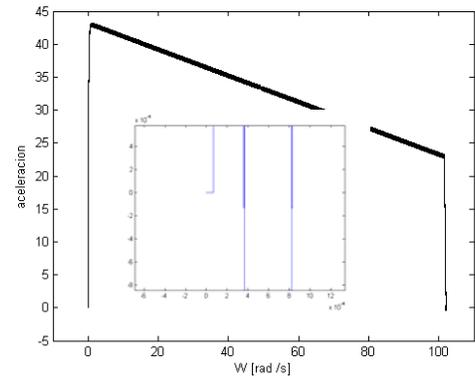
### Control **PID** con diferenciador **SM**



**Figura 4.49** Respuesta del control **PID** con diferenciador **SM**

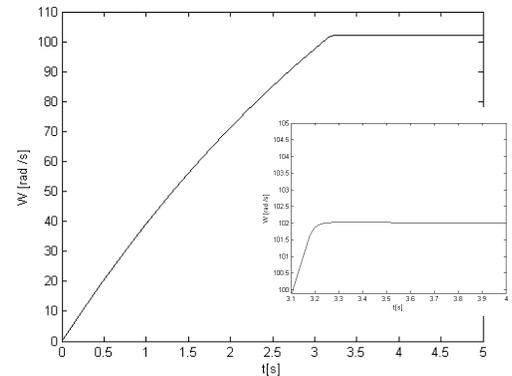


**Figura 4.50** Trayectoria del error en el espacio de estados con diferenciador **SM**



**Figura 4.51** Trayectoria en el espacio de estados del control **PID** con diferenciador **SM**

### Control **PID** con diferenciador de **Levant**



**Figura 4.52** Respuesta del control **PID** con diferenciador de **Levant**

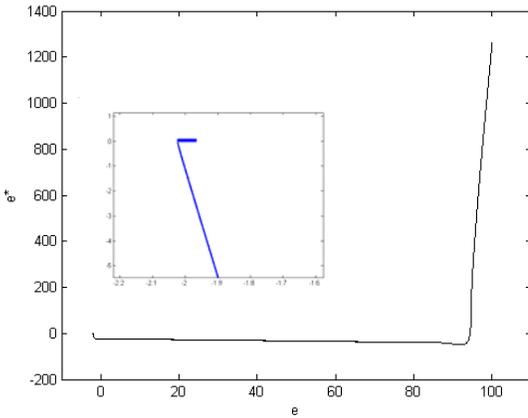


Figura 4.53 Trayectoria del error en el espacio de estados con diferenciador de Levant

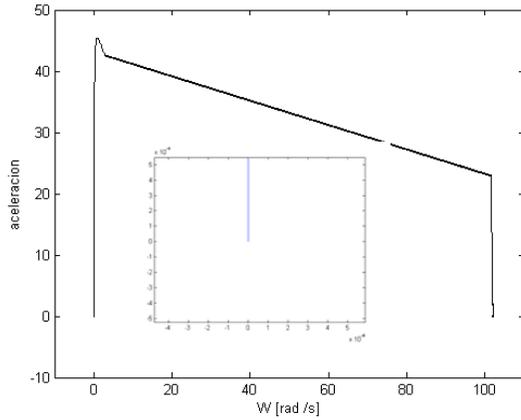


Figura 4.54 Trayectoria en el espacio de estados del control PID con diferenciador de Levant

Control PID con diferenciador de Shtessel

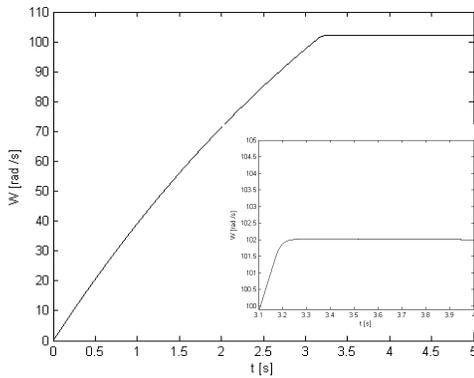


Figura 4.55 Respuesta del control PID con diferenciador de Shtessel

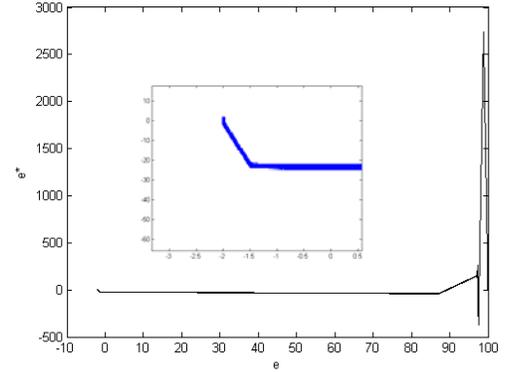


Figura 4.56 Trayectoria del error en el espacio de estados con diferenciador de Shtessel

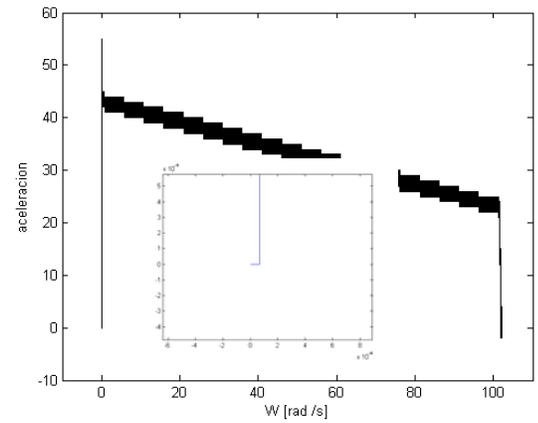


Figura 4.57 Trayectoria en el espacio de estados del control PID con diferenciador de Shtessel

Como puede apreciarse de lo anterior casi no existe una variación del comportamiento del **PID** con la aplicación de los diferenciadores comparado con lo sucedido en los casos anteriores por lo que para la última prueba se considerara el *diferenciador de Levant* para presentar las condiciones de operación del **PID** frente a perturbaciones.

#### 4.5.1.-Respuesta ante una perturbación senoidal

Para la realización de esta prueba como en casos anteriores se aplicará primero la perturbación del amplificador de potencia y después la perturbación generada por un error de medición en la variable de salida del motor de CD y por último se aplicarán las dos al mismo tiempo, según el diagrama siguiente:

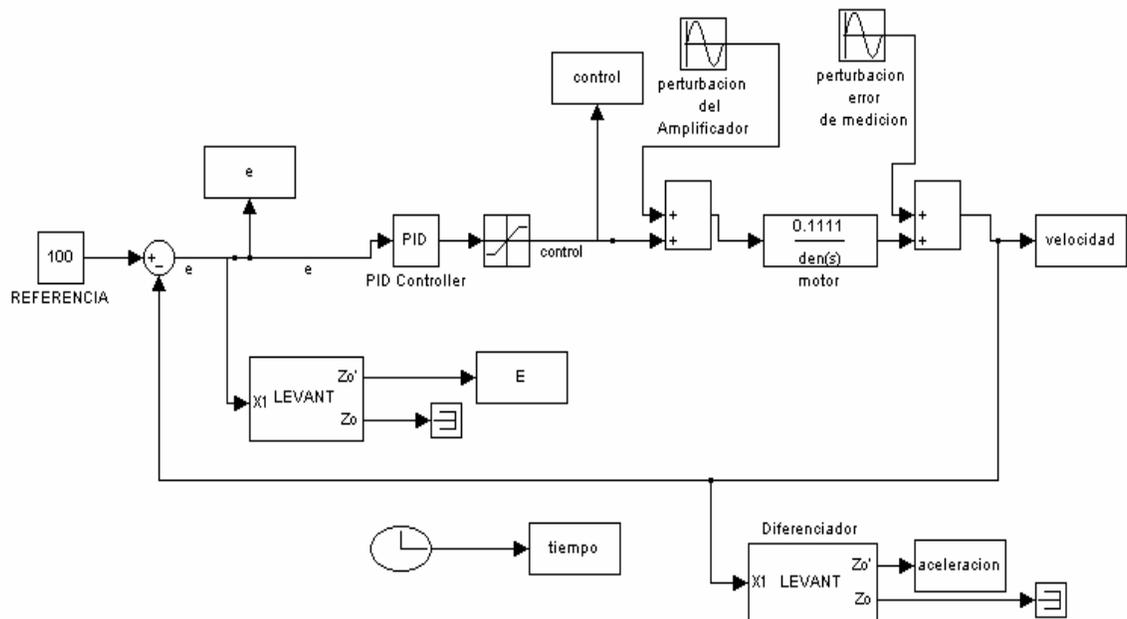
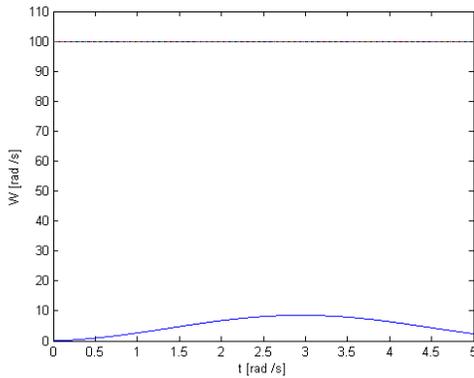
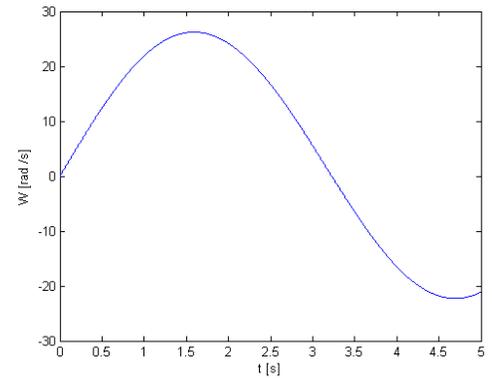


Figura 4.58 Diagrama del PID frente a perturbaciones



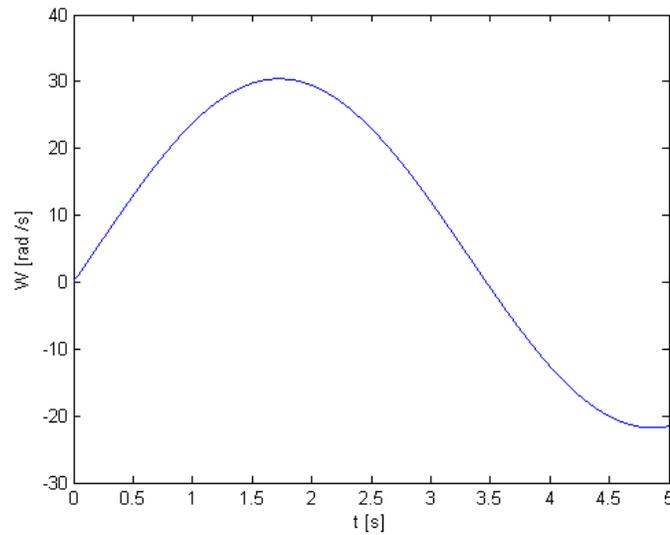
*Figura 4.59* Respuesta del PID frente a la perturbación del Amplificador



*Figura 4.60* Respuesta del PID frente a la perturbación de error de medición

De los gráficos anteriores se puede apreciar que el control **PID** no fue capaz de contrarrestar ninguna de las 2 perturbaciones por lo que juntas será aun más difícil como se ve en el último gráfico.

Aplicando las 2 perturbaciones al mismo tiempo.



*Figura 4.61* Respuesta del PID frente a las 2 perturbaciones



Para finalizar esta sección de pruebas se puede concluir que el control *PID* es muy sensible a la variación de los parámetros del lazo de control y poco robusto ante la presencia de perturbaciones llevando su comportamiento a una completa inestabilidad, y la única manera de realizar nuevamente un control adecuado a una referencia dada es contemplar estas variaciones en el modelado o por otro lado haciendo un ajuste en los parámetros de nuestro controlador, parámetros que deben disminuir este efecto. También cabe mencionar una forma adicional de mejorar la sensibilidad frente a perturbaciones misma que se explica brevemente en la sección de Anexos de este trabajo y consta de aplicar el control PID en su versión de PI-D.



## CAPÍTULO 5

# *Control de Velocidad en Tiempo Real*

### **5.1 Introducción**

En este Capítulo abordaremos el estudio de un entorno **CACSD** (*Computer Aided Control System Design*) para el diseño e implementación de sistemas de control.

Este entorno estará formado por un lenguaje de simulación dotado de una amplia biblioteca de bloques (**Simulink** [17]), una herramienta de generación automática de programas de control (**Real Time Workshop** [21]), y de una interfase para la adaptación de estos programas a un hardware específico (**Real Time Interface** [22]). El hardware sobre el que se van a implementar los controladores consiste de un procesador digital de señal, junto con canales analógicos/digitales y digitales/analógicos, así como otros subsistemas, que conforman la tarjeta de control **dSPACE 1103** ([23]).

El objetivo de este capítulo es la puesta en marcha de las diferentes herramientas en la aplicación del control de velocidad del motor de CD en estudio.

La herramienta **CACSD** facilita en teoría la labor al ingeniero de control, al permitirle automatizar diferentes pasos del proceso de diseño. Resaltaremos dos aspectos fundamentales, que han motivado la realización del experimento:

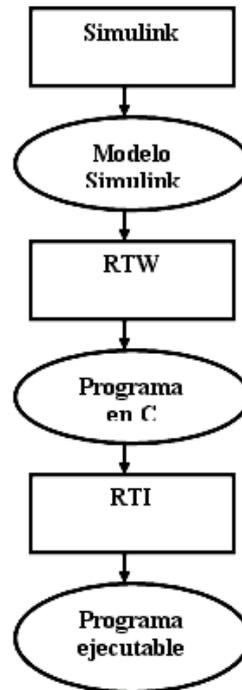
- El primero es que le permite describir y simular en un lenguaje de alto nivel la estructura de los sistemas con los que trabaja. Se trata de un lenguaje a nivel de diagramas de bloque, editable en un entorno visual, con todas las facilidades de cara a la edición que caracterizan a este tipo de entornos. Este lenguaje incorpora librerías de bloques básicos predefinidos utilizados con frecuencia, así como la posibilidad de crear nuevas librerías o expandir las ya existentes, de modo que se fomenta la reutilización de componentes, además permite la jerarquización de bloques, con lo que se facilita la estructuración de los modelos.



- El segundo, es que le permite la generación automática, a partir de la descripción de un controlador en ese lenguaje de alto nivel, del programa de control asociado, ejecutable directamente sobre un procesador específico. En particular, se puede generar un ejecutable para el **DSP** (Digital Signal Processor) que incorpora la tarjeta de control **dSPACE 1103**, que puede ser conectada directamente a la planta a controlar. Con esta generación automática de controladores ejecutables a partir de la descripción del control a implementar a nivel de diagrama de bloques se consigue un notable ahorro de tiempo y se evitan totalmente los errores de programación. Esta rapidez en la generación del programa permite al diseñador probar sobre la planta real sucesivas modificaciones (refinamientos) sobre el controlador en un tiempo mínimo, lo que se conoce como **RPC** (Rapid Control Prototyping). Además, la Tarjeta de Control puede interactuar con la PC a la vez que ejecuta un controlador, lo que permite realizar tareas de monitorización e incluso modificar parámetros del controlador desde la PC sin detener el control. Esto constituye una herramienta muy potente de cara a la sintonización de parámetros.

En otras palabras, con base en el modelo de una planta se podría diseñar y simular un controlador directamente en Simulink, sin necesidad de generar un programa a partir del controlador para realizar la simulación: Simulink, interactuando con MATLAB, se encarga de ello. Pero si además se desea probar el controlador sobre la planta real, es necesario generar un código que se ejecute sobre un procesador (en nuestro caso el de la Tarjeta de Control) para actuar como controlador sobre la planta. Es aquí donde entra en juego el proceso de generación automática de código mencionado. Su piedra angular el programa **RTW** (*Real Time Workshop*), capaz de generar a partir de un modelo en Simulink un programa ejecutable cuya ejecución reproduce en tiempo real el comportamiento del modelo Simulink. El código generado es específico del sistema sobre el que se vaya a ejecutar. **RTW** genera un archivo en lenguaje de programación **C** como paso intermedio entre el modelo Simulink y el ejecutable. La generación del ejecutable requiere de algunos archivos de configuración que indican a **RTW** las características del hardware destino, un compilador de **C** que genere código para ese hardware a partir del archivo intermedio, y posiblemente algunas librerías particulares de ese hardware que el compilador enlazará al programa principal para generar el ejecutable. Este esquema está abierto para la creación de código para numerosas plataformas distintas de destino, siempre que se suministren a **RTW** estos elementos específicos de esa plataforma.

A grandes rasgos, el proceso de generación automática de código descrito hasta ahora es de la siguiente manera



*Figura 5.1 Cadena de generación automática a partir de modelos Simulink*

No entraremos en una descripción detallada de todas sus posibilidades, pero sí ha de quedar claro que es un programa que traduce modelos Simulink/MATLAB a un programa ejecutable en un hardware específico. **RTW** está basado en una arquitectura abierta que permite que este hardware de destino sea configurable. Además, está diseñado para recibir una variedad amplia de aplicaciones, al margen de que en el presente trabajo nos centremos en el desarrollo de los controladores vistos con anterioridad.

## 5.2 Generalidades

Para generar un ejecutable con **RTW** a partir de un modelo Simulink es necesario configurar previamente una serie de parámetros. Esto se hace desde la ventana Simulink en que tenemos el modelo que vamos a compilar. En primer lugar, hay que indicar, de la misma forma que se hacía para una simulación, los parámetros de la pestaña *solver* de la ventana *simulation/parameters*. Estos parámetros afectan de igual forma a una simulación que se active a partir de ese momento

como al código generado si se invoca a **RTW**. Por otra parte, hay que indicar a **RTW** cómo generar el código para el hardware específico que elijamos. Esto se hace en la pestaña **RTW**. La instalación del *Real Time Interface* que acompaña a la tarjeta de control configura por defecto los parámetros de esta ventana para que el código generado sea para ejecutarse en la tarjeta:



Figura 5.2 Ventana de Parámetros del Real Time Workshop

Una vez configurados los parámetros necesarios, la generación de código es tan simple como elegir en el menú de nuestro modelo Simulink el comando *tools/rtwbuild*. Esto desencadena todo el proceso de traducción, que culmina con la carga y ejecución a la tarjeta. En la ventana de MATLAB van apareciendo los siguientes mensajes:

```
-----  
Starting build procedure with RTI 5.2 (RTI1103, 18-Mar-2005)  
Model: "probe_suboptimal_analog_tarjeta"  
(D:\MATLAB6p5\work\alex_experimentos_varios\probe_suboptimal_analog_tarjeta.mdl)  
-----  
*** Working directory: "D:\MATLAB6p5\work\alex_experimentos_varios"  
*** Optional User System Description File probe_suboptimal_analog_tarjeta_usr.sdf not available  
*** Initializing code generation  
### Starting Real-Time Workshop build procedure for model: probe_suboptimal_analog_tarjeta  
### Generating code into build directory: \probe_suboptimal_analog_tarjeta_rti1103  
### Invoking Target Language Compiler on probe_suboptimal_analog_tarjeta.rtw  
*** Generating Variable Description File probe_suboptimal_analog_tarjeta.trc  
*** Optional User Variable Description File probe_suboptimal_analog_tarjeta_usr.trc not available  
*** Found User-Code File probe_suboptimal_analog_tarjeta_usr.c from 07-Mar-2006 22:27:47  
*** Found User Makefile probe_suboptimal_analog_tarjeta_usr.mk from 07-Mar-2006 22:27:47  
### probe_suboptimal_analog_tarjeta.mk which is generated from d:\dSPACE\matlab\rti1103\m\rti1103.tmf is up to date  
### Building probe_suboptimal_analog_tarjeta: dsmake -f probe_suboptimal_analog_tarjeta.mk WORKINGBOARD=ds1103  
  
BUILDING APPLICATION (Single Timer Task Mode)
```



```

WORK DIRECTORY "D:\MATLAB6p5\work\alex_experimentos_varios"
BUILD DIRECTORY "D:\MATLAB6p5\work\alex_experimentos_varios\probe_suboptimal_analog_tarjeta_rti1103"
TARGET COMPILER "D:\PPCTools20m"

COMPILING probe_suboptimal_analog_tarjeta.c
COMPILING probe_suboptimal_analog_tarjeta_data.c
COMPILING D:\dSPACE\MATLAB\RTI1103\C\rti_external_sim.c
COMPILING D:\dSPACE\MATLAB\RTI1103\C\rti_sim_engine.c

USING LIBRARY "D:\dSPACE\MATLAB\RTI1103\C\Lib\rtwlib_r13_1_ds1103.lib"

LINKING APPLICATION ...
LINKING FINISHED

LOADING APPLICATION "probe_suboptimal_analog_tarjeta.sdf" ...
[#1] ds1103 - RTI:      Initializing ... (720)
[#2] ds1103 - RTI:      Initialization completed (721)
[#3] ds1103 - RTI:      Simulation state: RUN (700)
LOADING FINISHED

MAKE PROCESS SUCCEEDED

### Successful completion of Real-Time Workshop build procedure for model: probe_suboptimal_analog_tarjeta
*** Finished RTI build procedure for model probe_suboptimal_analog_tarjeta
    
```

Figura 5.3 Mensajes emitidos por RTW durante la generación automática de un ejecutable

### 5.3 Sintonización de Parámetros Interactivo en Tiempo Real

Se puede usar Simulink como un panel que nos permita interactuar con el programa en ejecución en tiempo real. Esto nos permite cambiar parámetros del programa en ejecución y observar directamente, en tiempo real, los efectos de esos cambios. En este proyecto no usaremos esta posibilidad de RTW, sino, en su lugar, se usará uno de los programas que acompañan a la tarjeta de control con la que se ha trabajado, **ControlDesk**, que nos proporciona una interfase sencilla agradable e intuitiva.

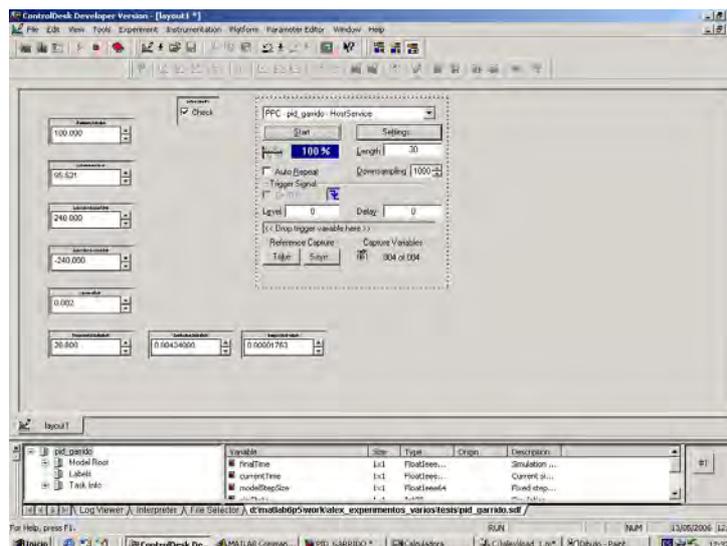


Figura 5.4 Pantalla de Interacción con Simulink del Programa ControlDesk

## 5.4 Programación desde Matlab/Simulink

El objetivo que se persigue es conseguir la traducción automática de un modelo Simulink a un programa que se ejecute sobre la tarjeta, para ello es necesario especificar en el modelo qué elementos funcionales de entrada / salida de la tarjeta vamos a utilizar en el modelo, especificando en el caso de aquellos que se encuentran duplicados en la tarjeta el número de elemento elegido, esto establece la correspondencia entre las señales del modelo Simulink y las entradas y salidas de la tarjeta.

Para conseguir esta finalidad el RTI (*Real Time Interface*) de la tarjeta proporciona una nueva librería de bloques Simulink que representan los distintos elementos de E/S de la tarjeta.

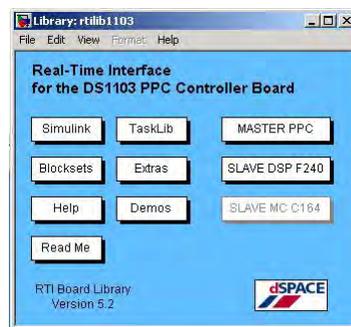


Figura 5.5 RTI del DSpace 1103

Por ejemplo los bloques que se emplearán para la realización de pruebas son:

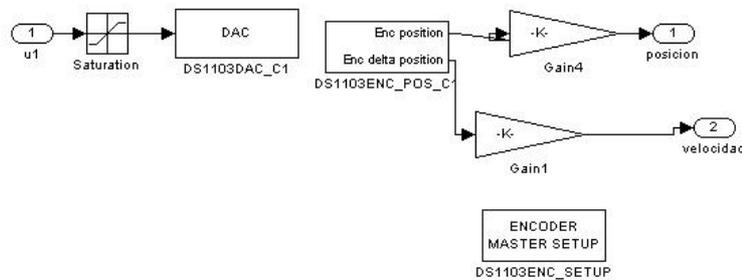


Figura 5.6 Bloques DAC y Enc. Position del Real Time Interface

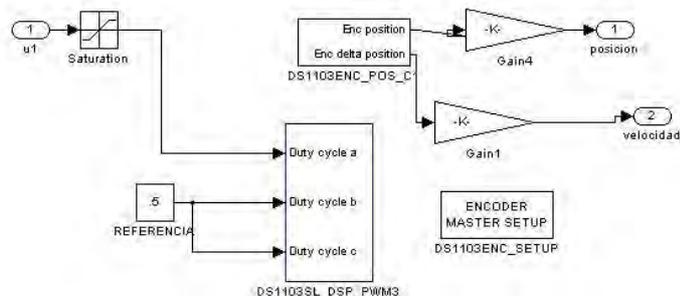


Figura 5.7 Bloques PWM y Enc. Position del Real Time Interface



**Bloque DAC:** este bloque nos permite la conversión de señales digitales en analógicas en un rango de  $\pm 10$  [V].

**Bloque PWM:** el subsistema de E/S digital de la tarjeta es capaz de generar, mediante la carga de un programa determinado en el procesador auxiliar hasta seis PWM's independientes (para una descripción de las señales PWM, ver sección de anexos). Esta funcionalidad queda recogida en el bloque PWM que añade el RTI a Simulink, este bloque recibe tres entradas, una por cada PWM a generar, para cada una de ellas, el valor que tome la señal de entrada indica el valor del ciclo de trabajo (un porcentaje del ciclo en que la señal está alta, en rango [0,1]), este bloque tiene varios parámetros configurables, individualmente para cada una de las señales, los valores inicial y final del ciclo de trabajo y de forma común a todas las señales, la frecuencia de generación del PWM.

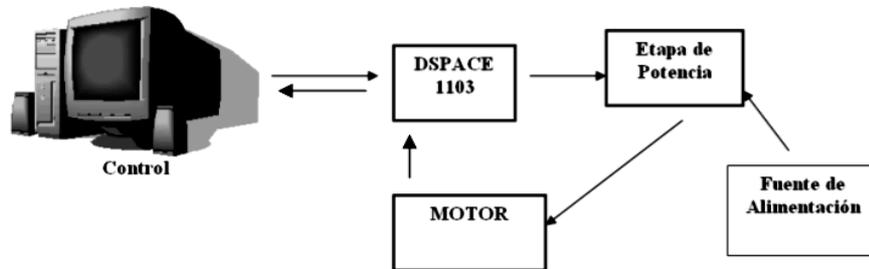
**Bloque ENC\_POS:** permite el acceso a los valores de ángulo generados por los interfases de *encoder* incremental de la tarjeta a partir de los trenes de pulsos defasados de un *encoder* del mismo tipo conectado a ellos (para una descripción de la medición de ángulos mediante *encoders* incrementales, referirse a la sección de anexos), este bloque no tiene parámetros. La conversión entre la salida de este bloque y el ángulo que representa, en radianes, obedece a la siguiente lógica: la interfase del *encoder* almacena el ángulo actual mediante un contador que se incrementa en una unidad por cada pulso que recibe en una dirección y se decrementa por cada pulso en la dirección contraria, el contador tiene 24 bits. La salida que proporciona el bloque tiene un rango [-1,1], que corresponde a la interpretación del valor del contador como un entero que varía en el rango  $[-2^{23}, 2^{23-1}]$ , y su correspondiente escalado.

Por todo ello, la conversión a realizar es la siguiente para obtener el ángulo real en radianes:

$$\text{ángulo [rad]} = \frac{2\pi}{\text{resolución del encoder}} = \frac{2\pi}{2500 \text{ pulsos}}$$

## 5.5 Pruebas con el motor de CD

Para la realización de las pruebas en el motor de CD, se trabajó de la siguiente forma:



*Figura 5.8* Forma de Trabajar para el Control del Velocidad en Tiempo Real

*Datos del motor empleado:*

### Datos de Placa

Motor de CD de escobillas  
Masa de la carga 1.3 Kgs  
Codificador óptico incremental de 2500 pulsos  
Velocidad nominal de 2000 RPM  
Corriente Nominal de 1.6 A  
Voltaje Nominal de 24 V  
Potencia Nominal de 38.4 VA  
I pico de 8 A



*Figura 5.9* Motor Empleado

*Datos sobre la fuente de alimentación:*

Esta fuente es de tipo no regulada de  $\pm 24$  [V] y 10 [A], tal que permite cubrir con las características de operación del motor en uso.



*Figura 5.10* Fuente de Alimentación

Datos sobre la tarjeta DSPACE 1103: Ver sección de anexos

Datos sobre la Etapa de Potencia: Ver sección de anexos



Figura 5.11 Tarjeta DSPACE 1103

Datos acerca del control:

En esta etapa se adecuan las simulaciones mostradas en el capítulo anterior para realizar la tarea del control de velocidad en el motor a través de Simulink y el panel de interacción *ControlDesk* por medio de la PC, tratando de seguir con el diagrama de control.

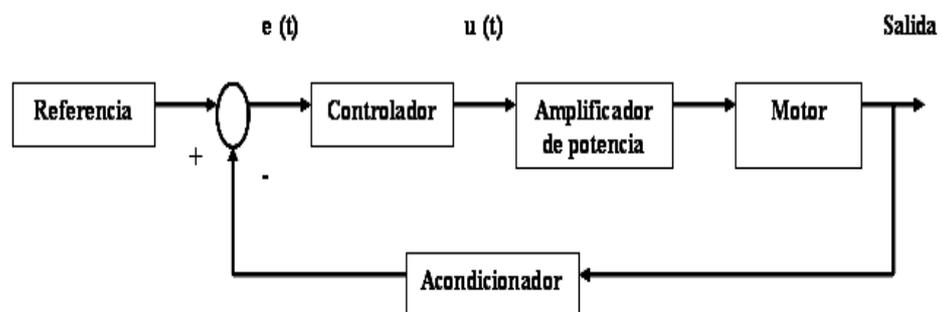


Figura 5.12 Lazo de Control para el Motor de CD

de tal manera que los bloques que representan al motor y al acondicionador de señal en las simulaciones se reemplazarán por los elementos reales.

### 5.5.1. Control de velocidad con el algoritmo FOSM

Para la prueba con el algoritmo de modos deslizantes se manejó un tiempo de muestreo de  $10[\mu\text{s}]$ , con solución a través de *ODE 1* para la compilación y descarga del programa en la tarjeta dSPACE, se fijó una referencia de  $100 [\text{rad} / \text{s}]$  y se hizo uso del subsistema de la **Figura 5.6**, así mismo, se empleó un diferenciador aproximado para la obtención de la derivada como se muestra en el siguiente diagrama.

RTI Data

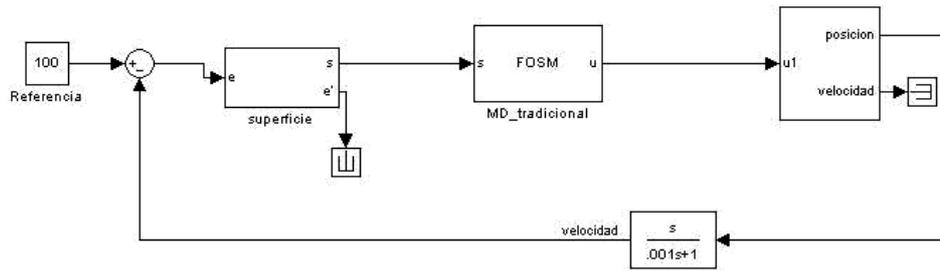


Figura 5.13 Lazo de control con el FOSM para el motor de CD con diferenciador aproximado

la respuesta de este sistema en tiempo real se puede observar en las siguientes figuras:

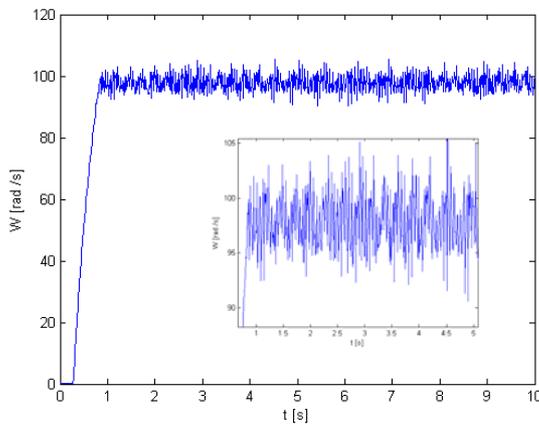


Figura 5.14 Respuesta del FOSM con diferenciador aproximado

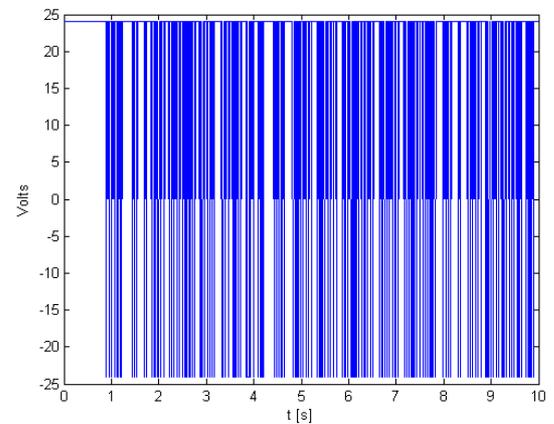


Figura 5.15 Acción de control del FOSM con diferenciador aproximado

Apartir de las dos figuras anteriores puede apreciarse el nivel de *chattering* presente y el nivel de referencia alcanzado por el motor, con tiempo de levantamiento en aproximadamente 0.9 [s], según la acción de control aplicada logrando un valor que oscila entre  $\pm 5$  [rad/s] del valor fijado.

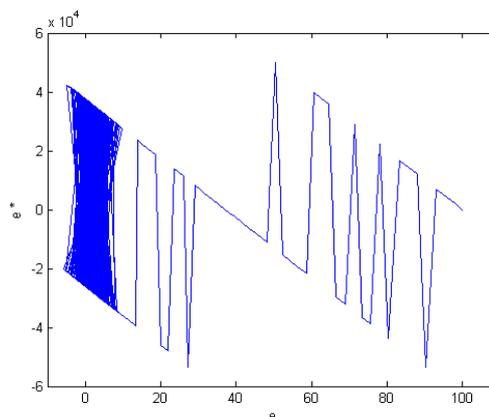
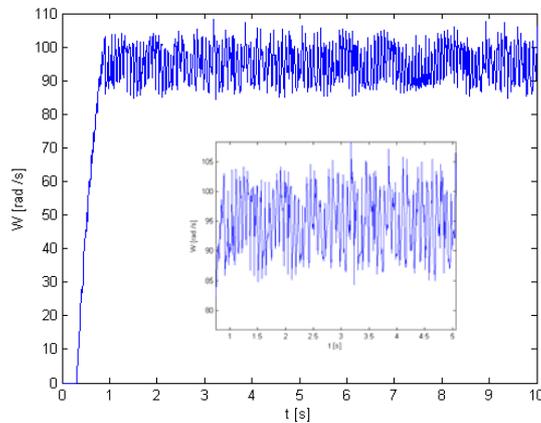
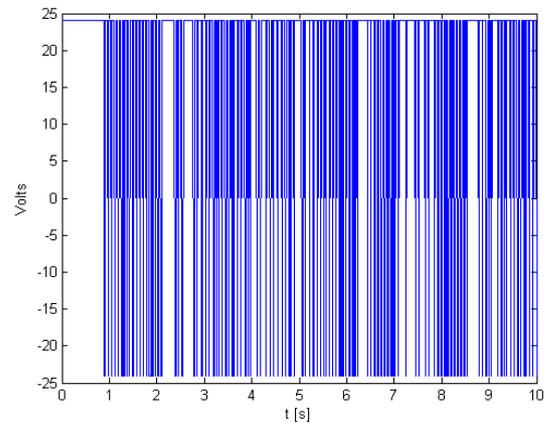


Figura 5.16 Trayectoria del error en el espacio de estados

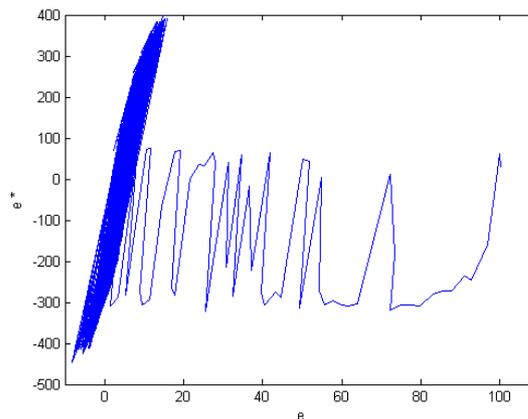
Si se cambia el tipo de diferenciador empleado por el diferenciador de Levant para la obtención de las derivadas involucradas en el sistema y respetando los parámetros de sintonización del mismo tal como se realizaron las simulaciones para esta prueba, es decir,  $\lambda_0=630$  y  $\lambda_1=126$  tenemos:



**Figura 5.17** Respuesta del FOSM con diferenciador Levant



**Figura 5.18** Acción de control del FOSM con diferenciador Levant



**Figura 5.19** Trayectoria del error en el espacio de estados

A partir de la **Fig. 5.18** se observa que el comportamiento del controlador cambia dejando ver una acción mas uniforme con este diferenciador, así mismo, la forma de llegar al origen en el plano de fase también cambia según se muestra en la **Fig. 5.19** y **Fig 5.16**, donde se aprecia una región de estabilidad (ciclo limite) provocada por los retardos intrínsecos de la discretización y las imperfecciones de operación de la etapa de potencia; más sin en cambio, las características de respuesta son muy similares según las **Fig. 5.17** y **5.14**.

Ahora bien, continuando con la prueba de control de velocidad en tiempo real apliquemos al experimento el control con el algoritmo Suboptimal.

### 5.5.2 Control de velocidad con el algoritmo Suboptimal

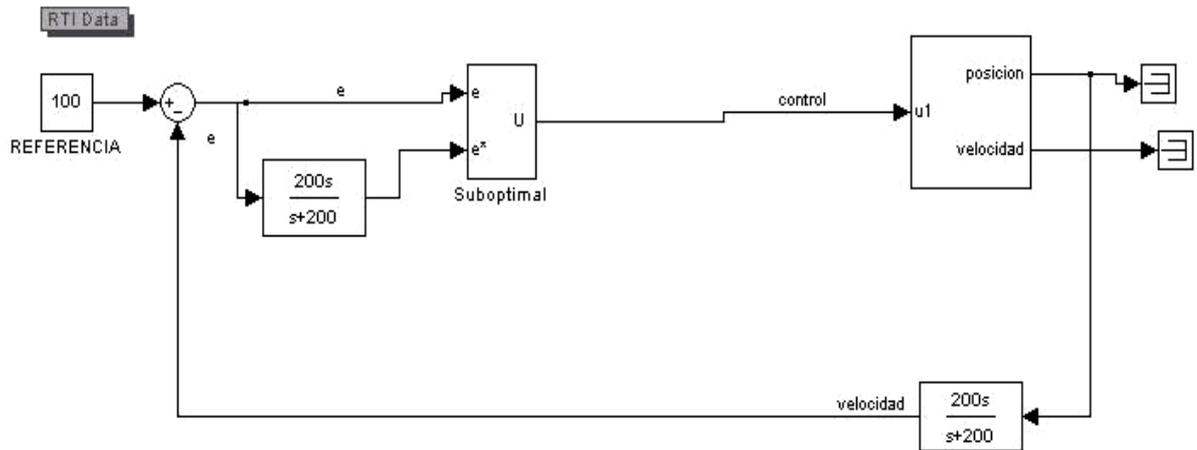


Figura 5.20 Lazo de control de velocidad con el algoritmo Suboptimal y un diferenciador lineal

Para este experimento primero se utilizó el algoritmo *Suboptimal* en conjunto con un diferenciador aproximado el cual permite la obtención de la derivada de manera significativa, con las características que representa el diagrama en simulink de la **Fig. 5.20**, conservando los parámetros empleados a su vez durante la simulación, ahora para la ejecución en tiempo real, es decir, solución del sistema a través de ODE1 y tiempo de muestro de 10 [μs] con duración de la prueba de 10 [s] y, utilizando una salida de la tarjeta como se muestra en el subsistema de la **Fig. 5.6**, los resultados obtenidos para esta prueba son:

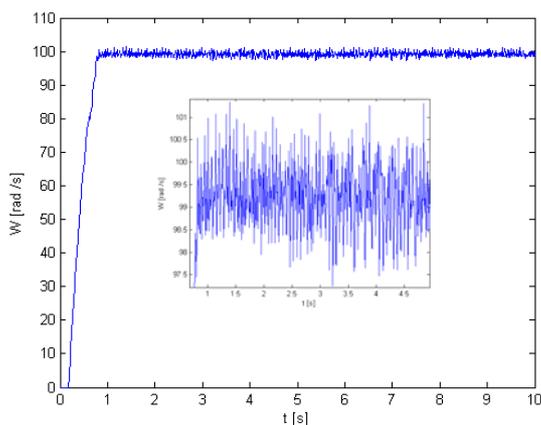


Figura 5.21 Respuesta del Suboptimal con diferenciador aproximado

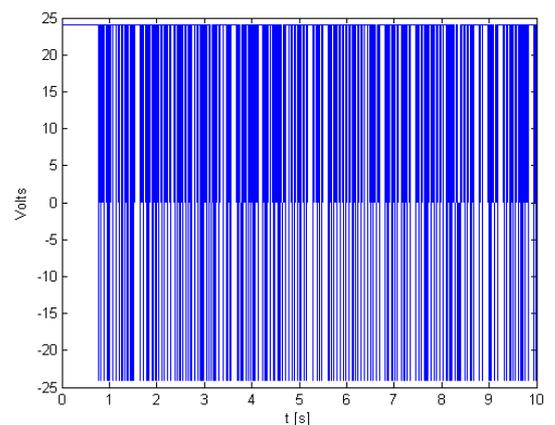


Figura 5.22 Acción de control del Suboptimal con diferenciador aproximado

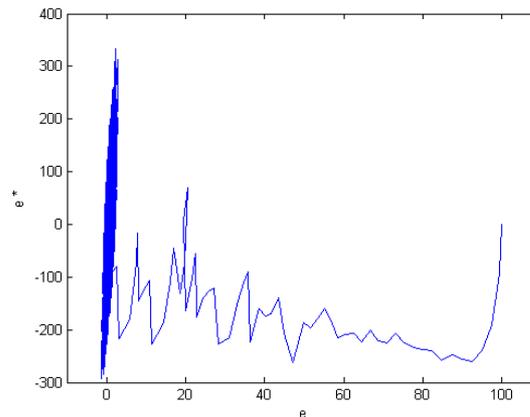


Figura 5.23 Trayectoria del error en el espacio de estados

De las figuras anteriores se aprecia que el nivel de *chattering* es menor en comparación con el algoritmo *FOSM*, alcanzando una amplitud máxima de  $\pm 1.5$  [rad] y que la respuesta del sistema no presenta sobrepaso como se vio durante las simulaciones, en la **Figura 5.22** vemos como la acción del control tiende a ser de muy alta frecuencia pero solo a tramos debido a las imperfecciones del sistema como puede verse también en la trayectoria del error según la **Figura 5.23** donde la convergencia se da en tiempo finito pero conservando una región de estabilidad.

Si se cambia el tipo de diferenciador del sistema por el de modos deslizantes (**SM**) se obtiene lo siguiente:

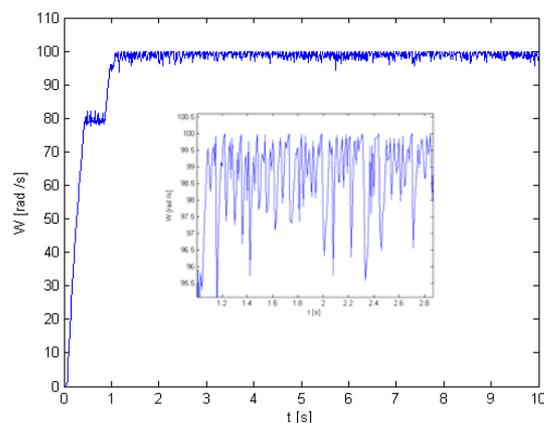
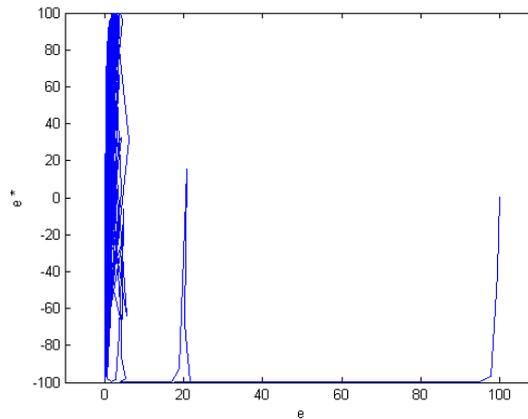


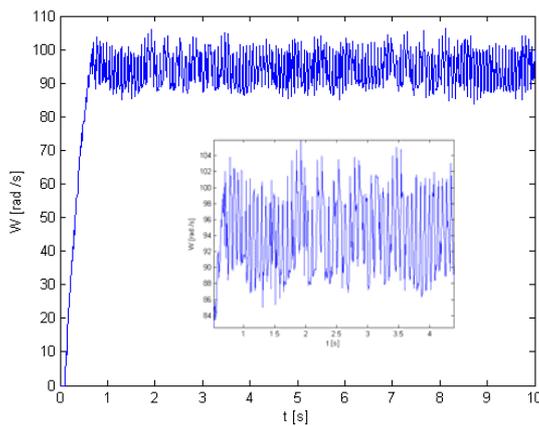
Figura 5.24 Respuesta del Subóptimo con diferenciador SM



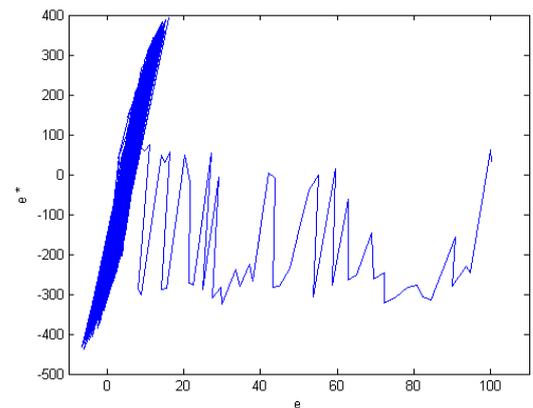
*Figura 5.25 Trayectoria del error en el espacio de estados*

Analizando la respuesta del control Suboptimal en su acción con el diferenciador **SM**, se observa claramente que: dada la información que recibe el control de la señal de salida del filtro paso-bajas del diferenciador **SM** este decide frenar la dinámica natural del sistema para garantizar la estabilidad en tiempo finito y sin sobrepaso a la referencia de 100 [rad /s] como puede también verse en la **Figura 5.25**.

Si se continua con las pruebas para este control se tiene que: una vez más el sistema conservará todo igual excepto que se cambiará el diferenciador por el diferenciador de Levant y una prueba mas sera cambiando el diferenciador por el diferenciador de Shtessel.

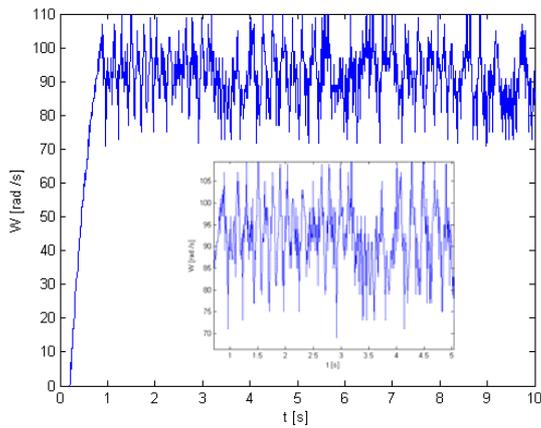


*Figura 5.26 Respuesta del Suboptimal con diferenciador Levant*

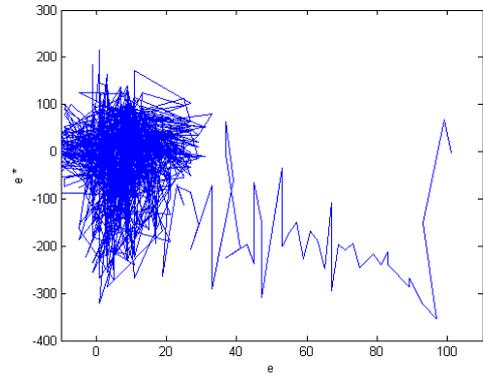


*Figura 5.27 Trayectoria del error en el espacio de estados*

Prueba para el diferenciador de Shtessel o filtro SSZL:

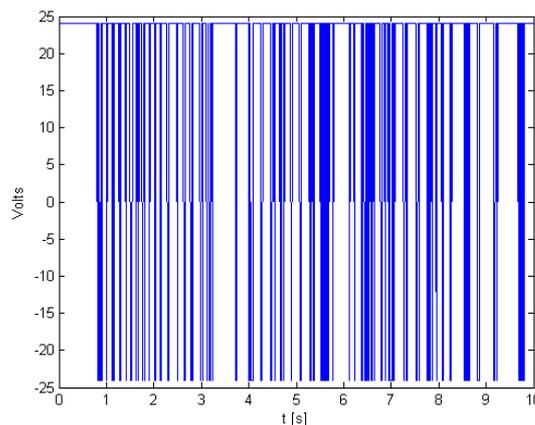


*Figura 5.28* Respuesta del Suboptimal con diferenciador Shtessel



*Figura 5.29* Trayectoria del error en el espacio de estados

De las cuatro figuras anteriores puede verse un mejor desempeño del algoritmo Suboptimal para el control de velocidad del motor de **CD** en tiempo real en su acción con los diferenciadores en estudio, aunque, en contraste con los resultados obtenidos durante la simulación, donde el **filtro SSZL** ofrecía mejores expectativas para su uso con este algoritmo, es evidente que este necesita ajustarse nuevamente para optimizar su funcionamiento ya que al parecer es más sensible a los retardos que se tienen en el sistema según se ve en la **Figura 5.29** y en la acción del control de la figura siguiente:

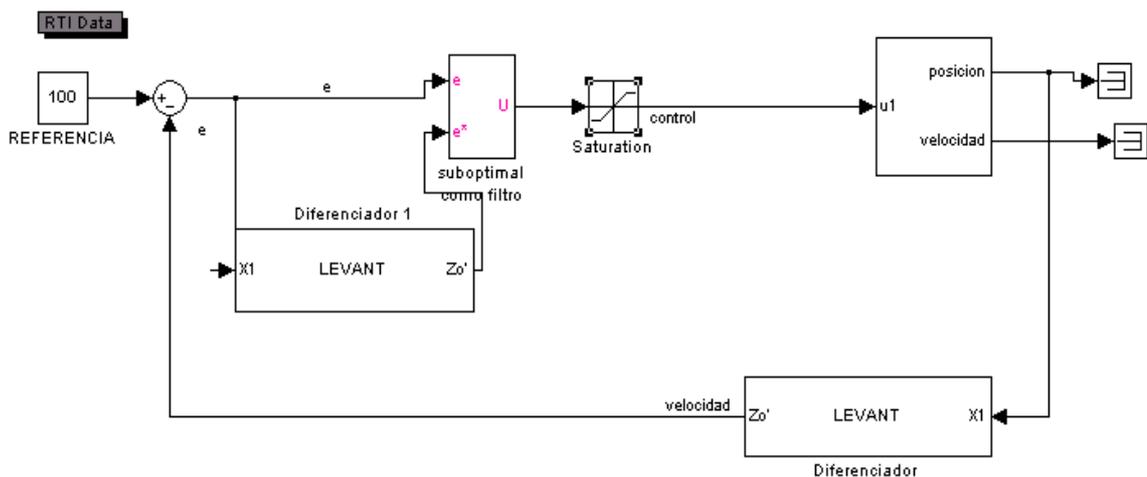


*Figura 5.30* Acción de control del Suboptimal con diferenciador Shtessel

Todas las figuras anteriores corresponden al comportamiento del motor cuando se controla su velocidad por medio del algoritmo **Suboptimal** de modos deslizantes, es fácil observar que en cada una de las pruebas el valor fijado como referencia en el sistema se alcanza sin sobrepaso y en tiempo finito; así mismo, el control contrarresta cualquier tipo de incertidumbre o perturbación

durante su acción, aunque también, es notable la aparición del fenómeno chattering en las pruebas, debido a las imperfecciones del sistema y este fue mucho más suave que para el caso del control de velocidad con el algoritmo FOSM, sin embargo, para disminuir la amplitud del chattering, se continuara con la prueba con el algoritmo Suboptimal en su versión como filtro para observar el desempeño de este en tiempo real.

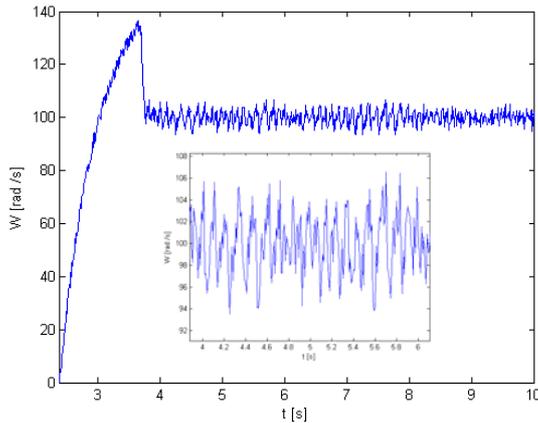
### 5.5.3 Control de Velocidad con el Algoritmo Suboptimal como Filtro



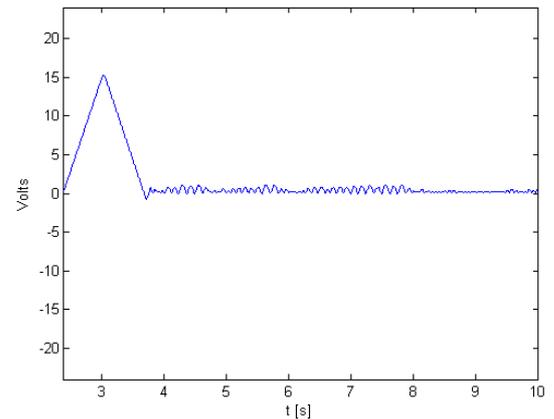
*Figura 5.31* Lazo de control de velocidad con el algoritmo Suboptimal y diferenciador de Levant

La prueba siguiente se basa en el diagrama descrito en la **Fig. 5.31**, donde claramente se ve que aparece un bloque que satura la acción del controlador, esto se debe a que el voltaje que debe recibir como máximo el motor de **CD** es de 24 V., los parámetros al igual que en los otros casos se conservarán con base a la simulación realizada en el capítulo anterior.

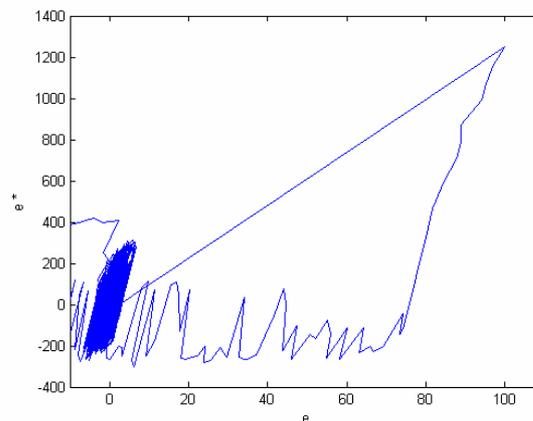
Los resultados de la aplicación de este algoritmo en su operación en conjunto con el diferenciador de Levant son según se muestra a continuación:



**Figura 5.32** Respuesta del Suboptimal como filtro con diferenciador Levant



**Figura 5.33** Acción de control del Suboptimal como filtro con diferenciador Levant



**Figura 5.34** Trayectoria del error en el espacio de estados

de lo anterior se puede concluir que el control Suboptimal como filtro si atenua el fenómeno del chattering que genera el sistema bajo estas condiciones de operación, aunque también es evidente en la **Fig. 5.32** el sobrepaso de casi 40% que provoca la acción del integrador a la salida del control y los retardos de discretización.

Si se cambia el diferenciador de Levant por un “diferenciador aproximado” para la obtención de la derivada, se obtienen la gráfica que se muestra en la **Figura 5.35**; tal que, al no tener un valor

de las derivadas más cercano al valor real y dadas las imperfecciones del sistema en la prueba provoca en este que su respuesta sea totalmente diferente aun teniendo la misma acción de control.

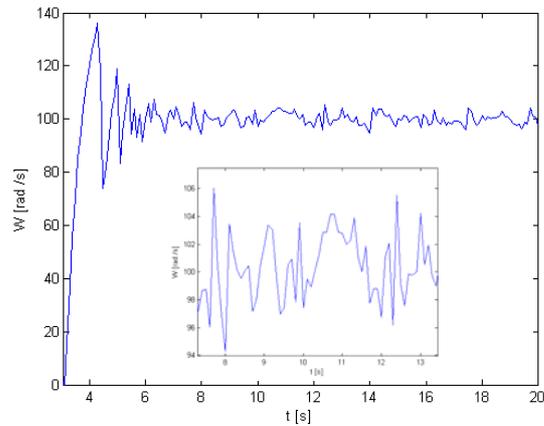


Figura 5.35 Respuesta del Suboptimal como filtro con “diferenciador aproximado”

#### 5.5.4 Control de Velocidad con el Control PID

Para esta sección emplearemos el control **PID** de Simulink y la propuesta de control según la **Fig. A.2** del apartado de anexos para la realización del control de velocidad del motor de CD en tiempo real con base al siguiente diagrama; los parámetros de sintonización del controlador se mantendrán de igual forma que en las simulaciones del capítulo anterior.

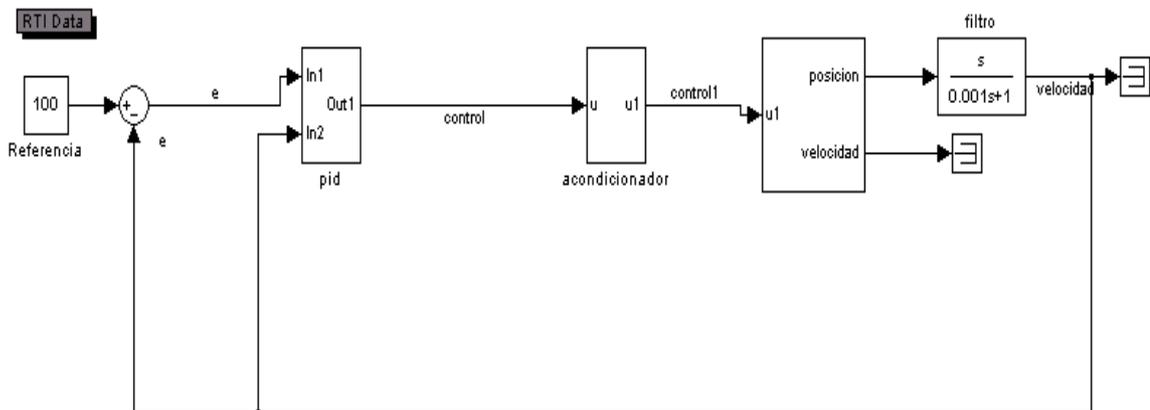
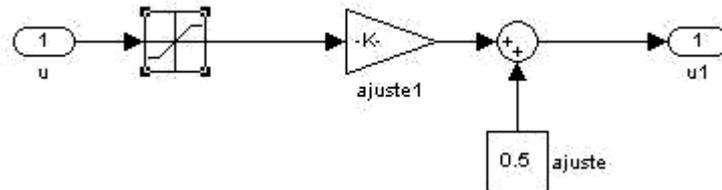


Figura 5.36 Lazo de control de velocidad con el control PID y diferenciador lineal

Se observa en el lazo de control de la figura anterior un bloque de acondicionamiento de señal el cual permite que la señal de control se la adecuada para la generación de los ciclos de trabajo correspondientes a la señal **PWM** generada por la tarjeta, según la salida especificada por el subsistema de la **Figura 5.7** y que recibirá la etapa de potencia para que el control opere de forma lineal.

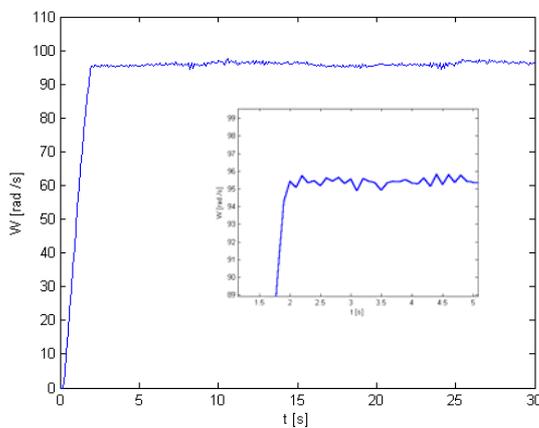


**Figura 5.37** Estructura interna del acondicionador de señal PID

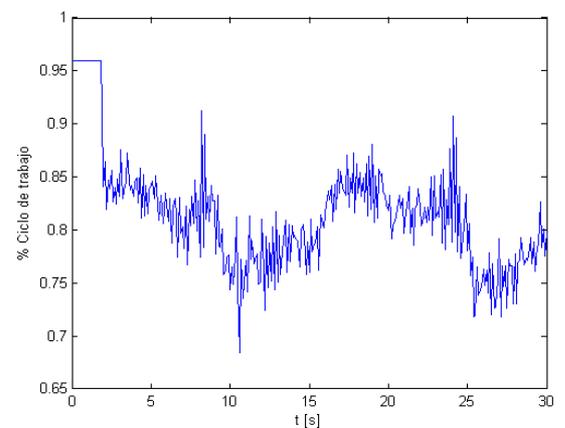
Así mismo cabe señalar que se limito el ciclo de trabajo en un rango de funcionamiento del 4 al 96 % dadas las imperfecciones del sistema.

Los resultados obtenidos para este controlador son...

Cuando aplicamos el PID propuesto:



**Figura 5.38** Respuesta del control PID propuesto con diferenciador aproximado



**Figura 5.39** Acción del control PID propuesto con diferenciador aproximado

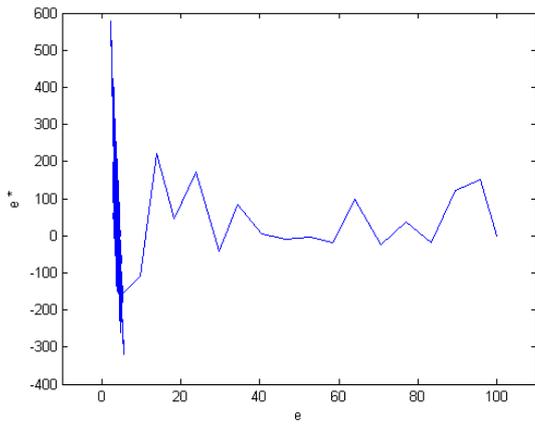


Figura 5.40 Trayectoria del error en el espacio de estados

Cuando se utiliza el bloque PID de Simulink

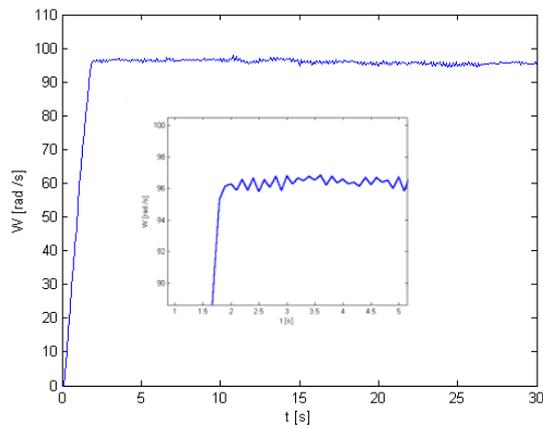


Figura 5.41 Respuesta del control PID de Simulink con diferenciador aproximado

Con PID propuesto

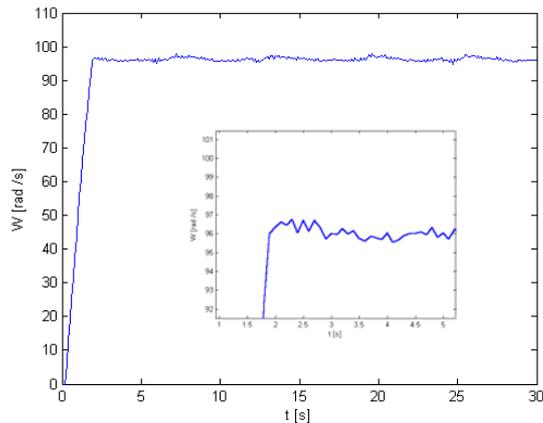


Figura 5.44 Respuesta del control PID propuesto con diferenciador Levant

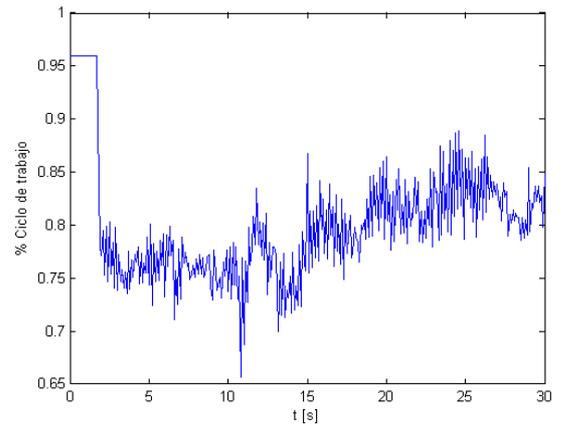


Figura 5.42 Acción del control PID de Simulink con diferenciador aproximado

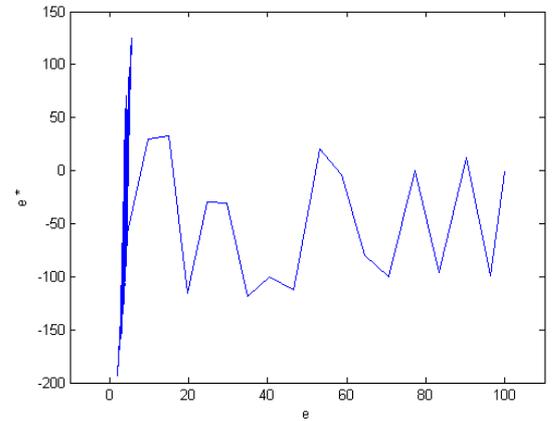


Figura 5.43 Trayectoria del error en el espacio de estados

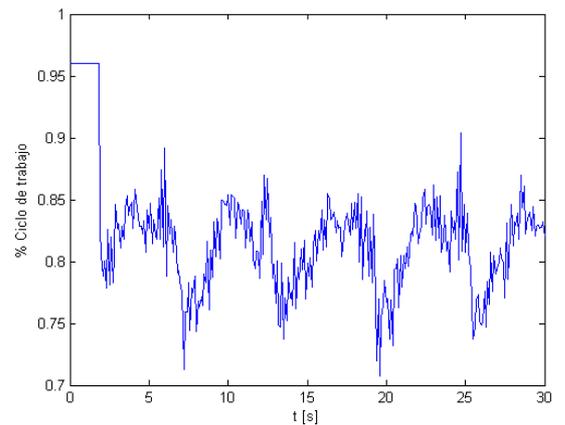
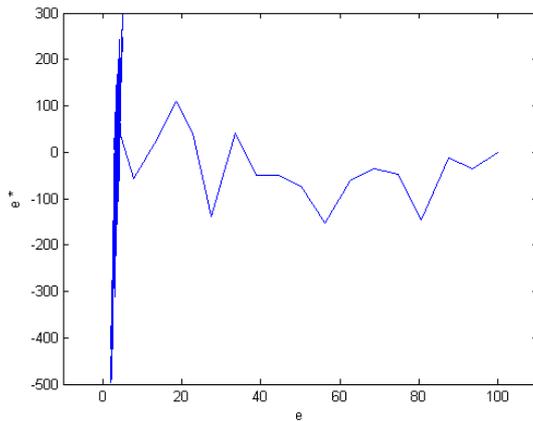
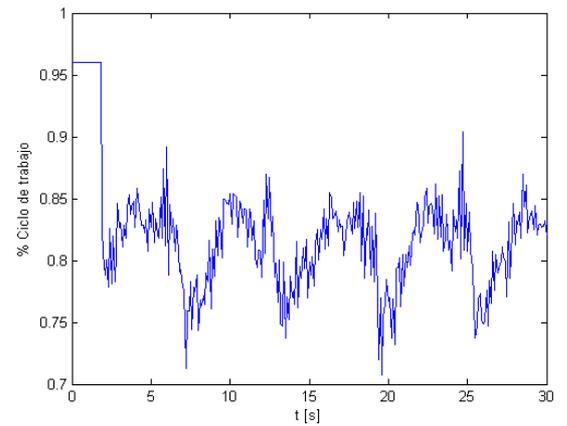


Figura 5.45 Acción del control PID propuesto con diferenciador Levant

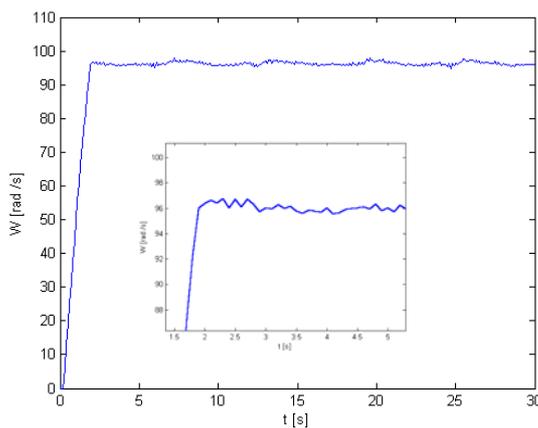


**Figura 5.46** Trayectoria del error en el espacio de estados

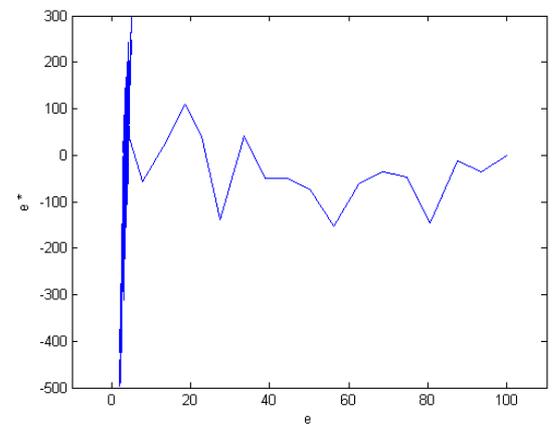


**Figura 5.48** Acción del control PID de Simulink con diferenciador Levant

Con PID de Simulink



**Figura 5.47** Respuesta del control PID de Simulink con diferenciador Levant



**Figura 5.49** Trayectoria del error en el espacio de estados

De las gráficas anteriores se observa el error de estado estable presente en cada una de las respuestas del sistema con valor aproximado de 4 [rad], se ve también que la referencia es alcanzada con tiempo de levantamiento aproximado de 1.7 [s] y que no presenta sobrepaso dada la sintonización de respuesta que se le asignó a este controlador, por otro lado se observa también la forma de actuar de este control al modular el ancho de pulso de una señal PWM con una frecuencia fija de 2 000 [Hertz] para hacer que el control de velocidad sea más suave pero así también, se aprecia la manera en que el error tiende al origen en el plano de fase y como la convergencia no se da en tiempo finito.



Sin embargo en las pruebas no se aprecia el efecto del control *PID* propuesto de manera significativa, es decir, en la optimización de la respuesta del sistema frente a incertidumbres o perturbaciones y, tampoco se nota una clara variación al intercambiar el diferenciador en uso, se hace la suposición de que esto se debe al ajuste del control para la generación del ciclo de trabajo o bien a la imperfección de la etapa de potencia con la que se trabajó.



## CONCLUSIONES

El proceso de diseño de sistemas de control se encuentra en la actualidad fuertemente automatizado. La utilización de herramientas *CACSD* ha permitido llegar al desarrollo y puesta a punto diferentes sistemas de control en tiempos verdaderamente cortos.

En este trabajo se ha analizado una herramienta de este tipo basada en el popular entorno de programación MATLAB. La herramienta está integrada por diferentes programas, cada uno con una determinada funcionalidad: **Simulink** para la definición y simulación de sistemas, **RTW/RTI** para la generación automática de código para la tarjeta **DSPACE 1103**.

También es destacable la buena integración entre **RTW** y **RTI**, que permite que el usuario puede abstraerse de los detalles de configuración de **RTW**. **RTI**, al instalarse, directamente configura todos los parámetros necesarios para la generación automática de código ejecutable en la tarjeta de control **dSPACE DS-1103**.

Se han descrito a grandes rasgos las características más importantes de cada uno de los pasos a seguir para la implementación de los controladores en estudio para su aplicación en tiempo real, de forma que esta memoria puede servir como un primer paso para el estudio de la herramienta **DSPACE 1103** y su aplicación a otros casos prácticos de control.

El objetivo inicial de la presente tesis era el análisis del comportamiento del control de velocidad de un motor de **CD** a través del algoritmo Suboptimal de modos deslizantes de segundo orden (2-SMC) y la implementación del mismo para su aplicación en tiempo real donde se puso en estudio el acondicionamiento de señales para los sensores y actuadores involucrados.

En particular, se ha diseñado electrónica de potencia con disposición en **H** de los transistores, para manejar al motor de corriente directa (véanse los Anexos).

Respecto al control del motor, se han realizado las pruebas con cada uno de los diferenciadores propuestos en este trabajo y se observó claramente en su acción la importancia de tener una derivada cercana al valor real en especial para trabajar en tiempo real, pues para valores aproximados y de baja precisión como es el caso del diferenciador de *Simulink* la magnitud del ruido indeseable aumento considerablemente, para lo cual, se pusieron en marcha una variedad de



## Conclusiones

---

pruebas con la finalidad de apreciar porque los diferenciadores basados en la teoría de modos deslizantes son una clara opción para su aplicación en tiempo real o bien para realizar cualquier otro tipo de operación, ya que se verificó en cierto modo, que la sintonización o ajuste realizado en una primera etapa para el cálculo de la derivada de una señal cuadrada era más que suficiente para continuar con el resto de las pruebas de simulación computacional y finalmente las pruebas en tiempo real para el control de velocidad del motor en estudio; desde luego es importante agregar que aun se pueden optimizar los resultados obtenidos con un nuevo ajuste en los parámetros que definen a cada uno de los diferenciadores, ya que las imperfecciones del sistema (retardos por discretización, retardos en la etapa de potencia), suelen ser representativos para las pruebas y en particular para el desempeño de los mismos, aunque lo que se busco en principio fue ver su funcionamiento partiendo de una base teórica. Por otro lado, al aplicar el algoritmo Suboptimal para el control de velocidad del motor de CD, primero en la etapa de simulación computacional, se verificó la convergencia en tiempo finito, la robustez frente a la variación de los parámetros, así como, la insensibilidad frente a perturbaciones internas y externas, además de un bajo nivel de *chattering*. Cuando se utilizó el Suboptimal como filtro, solo se comprobó que se vuelve mas sensible frente a perturbaciones y que su respuesta era mas lenta, sin perder las características de convergencia en tiempo finito y robustez ante la variación de parámetros de la planta, también se comprobó la suavidad de la señal de salida (velocidad). Al llevar estas pruebas a su operación en tiempo real, desafortunadamente y dadas las características en las que se trabajo (fuente, etapa de potencia, retardos de discretización), el nivel de *chattering* no fue tan evidente que se disminuyera pero si se comprobó la convergencia en tiempo finito, además de verificar la robustez y la insensibilidad frente a perturbaciones dejando notar diferencias claras sobre los otros algoritmos de control estudiados en este trabajo.

Como trabajo futuro se pueden mejorar todas las condiciones de operación del sistema, diseñando una nueva etapa de potencia con la capacidad de respuesta suficiente para efectuar el control en alta frecuencia, diseñar o agregar un tacogenerador para medir de forma directa la variable de interés, agregar algún observador para contemplar la variación de parámetros y la magnitud de las perturbaciones asociadas, realizar el ajuste de los diferenciadores de forma dinámica en tiempo real. Y lo más interesante seria probar otro algoritmo de modos deslizantes como el algoritmo Super-Twisting, algoritmo que puede ser empleado, como diferenciador, controlador u observador.



Para complementar lo anterior se presentan las siguientes tablas de resultados, basadas en las pruebas efectuadas al control de velocidad del motor de **CD** en estudio:

Tipo de Control	Diferenciador en uso	Tiempo de levantamiento [s]	Sobrepaso [%]	Tiempo de asentamiento [s]	% de error
<b>FOSM</b>	<i>Aproximado</i>	3	0	3.11	0.0005
	<i>SM</i>	3	0	3.1	0.00025
	<i>Levant</i>	3	0.0005	3.13	0.00025
	<i>Shtessel</i>	3.1	0	3.1	0.0004
	<b>Amplitud de Chattering [rad / s]</b>	<b>Valor Alcanzado [rad /s]</b>	<b>Convergencia en tiempo finito</b>		
	±0.00025	99.9995	si		
	±0.0035	99.99975	si		
	±0.0035	99.99975	si		
	±0.0025	99.9996	si		

*Tabla C.1 Resultados de la simulación con el algoritmo FOSM*

Tipo de Control	Diferenciador en uso	Tiempo de levantamiento [s]	Sobrepaso [%]	Tiempo de asentamiento [s]	% de error
<b>Suboptimal</b>	<i>Aproximado</i>	3.11	0	3.12	0.00025
	<i>SM</i>	3.25	0	3.3	0.00037
	<i>Levant</i>	3.15	0.0005	3.2	0.0005
	<i>Shtessel</i>	3.1	0	3.1	0.00035
	<b>Amplitud de Chattering</b>	<b>Valor Alcanzado [rad /s]</b>	<b>Convergencia en tiempo finito</b>		
	±0.0005	99.99975	si		
	±0.001	99.999625	si		
	±0.001	99.9995	si		
	±0.00125	99.99965	si		

*Tabla C.2 Resultados de la simulación con el algoritmo Suboptimal*



Tipo de Control	Diferenciador en uso	Tiempo de levantamiento [s]	Sobrepaso [%]	Tiempo de asentamiento [s]	% de error
Suboptimal como Filtro	<i>Shtessel</i>	4.6	0	4.8	0.02
	<b>Amplitud de Chattering</b>	<b>Valor Alcanzado [rad /s]</b>	<b>Convergencia en tiempo finito</b>		
	±0.001	100	si		

*Tabla C.3* Resultados de la simulación con el algoritmo Suboptimal como Filtro

Tipo de Control	Diferenciador en uso	Tiempo de levantamiento [s]	Sobrepaso [%]	Tiempo de asentamiento [s]	% de error
PID	<i>Aproximado</i>	3.2	2	3.3	2
	<i>SM</i>	3.2	2	3.3	2
	<i>Levant</i>	3.2	2	3.3	2
	<i>Shtessel</i>	3.2	2	3.3	2
	<b>Amplitud de Chattering</b>	<b>Valor Alcanzado [rad /s]</b>	<b>Convergencia en tiempo finito</b>		
No hay	102	no			
No hay	102	no			
No hay	102	no			
No hay	102	no			

*Tabla C.4* Resultados de la simulación con el control PID



Tipo de Control	Diferenciador en uso	Tiempo de levantamiento [s]	Sobrepaso [%]	Tiempo de asentamiento [s]	% de error
FOSM	<i>Aproximado</i>	0.8	0	1	2
	<i>Levant</i>	0.8	0	1	2
	<b>Amplitud de Chattering</b>	<b>Valor Alcanzado [rad /s]</b>	<b>Convergencia en tiempo finito</b>		
	± 8	98	si		
	± 15	98	si		

Tabla C.5 Resultados de las pruebas en tiempo real con el algoritmo FOSM

Tipo de Control	Diferenciador en uso	Tiempo de levantamiento [s]	Sobrepaso [%]	Tiempo de asentamiento [s]	% de error
Suboptimal	<i>Aproximado</i>	0.8	0	1	0.5
	<i>SM</i>	1	0	1.2	1
	<i>Levant</i>	0.8	0	1	3
	<i>Shtessel</i>	0.8	0	1	5
	<b>Amplitud de Chattering</b>	<b>Valor Alcanzado [rad /s]</b>	<b>Convergencia en tiempo finito</b>		
	± 3	99.5	si		
	± 3.5	99	si		
	± 14	97	si		
	± 35	95	si		

Tabla C.6 Resultados de las pruebas en tiempo real con el algoritmo suboptimal



Tipo de Control	Diferenciador en uso	Tiempo de levantamiento [s]	Sobrepaso [%]	Tiempo de asentamiento [s]	% de error
Suboptimal como Filtro	<i>Levant</i>	3	40	4	0
	<i>Aproximado</i>	3.8	38	6	2
	Amplitud de Chattering	Valor Alcanzado [rad /s]	Convergencia en tiempo finito		
	± 12	100	si		
	± 12	98	si		

Tabla C.7 Resultados de las pruebas en tiempo real con el algoritmo Suboptimal como filtro

Tipo de Control	Diferenciador en uso	Tiempo de levantamiento [s]	Sobrepaso [%]	Tiempo de asentamiento [s]	% de error
FOSM	<i>Aproximado</i>	2	0	2.5	4.5
	<i>Levant</i>	2	0	2.5	3.5
	Amplitud de Chattering	Valor Alcanzado [rad /s]	Convergencia en tiempo finito		
	No hay	95.5	no		
	No hay	96.5	no		

Tabla C.8 Resultados de las pruebas en tiempo real con el control PID propuesto



# ANEXOS

## Control Proporcional Integral Derivativo

### Introducción

El regulador más usado en control de procesos es el PID cuya representación es la siguiente:

$$u(t) = K \left[ e(t) + \frac{1}{Ti} \int_0^t e(\tau) d\tau + Td \frac{d e(t)}{dt} \right] \quad (1.1)$$

siendo  $u$  la variable de control y  $e$  la diferencia entre la referencia y la salida.

Se observan, dentro del corchete, tres términos representando la acción  $P$ ,  $I$  y  $D$  respectivamente.

La acción proporcional es el control por realimentación más simple que se pueda obtener. De todos modos ya incorpora ventajas en el comportamiento dinámico del sistema en lazo cerrado. Por ejemplo sea un sistema de primer orden

$$Y(s) = \frac{A}{1+sT} U(s) \quad (1.2)$$

Si se lo realimenta con un regulador  $P$  resulta

$$Y(s) = \frac{\frac{KA}{1+s\tau}}{1 + \frac{KA}{1+s\tau}} R(s) = \frac{\frac{KA}{1+KA}}{1+s \frac{\tau}{1+KA}} R(s) \quad (1.3)$$



La constante de tiempo en lazo abierto es  $\tau$ , en cambio en lazo cerrado es  $\frac{\tau}{1+KA}$

Al aumentar  $K$  el sistema se hace más rápido. Además la ganancia en lazo abierto es  $A$  y en lazo cerrado  $\frac{KA}{1+KA}$ . A medida que  $K$  aumenta, la ganancia tiende a uno, objetivo buscado en el control.

Sin embargo, solo con  $K = \infty$ , llegaríamos a ganancia uno, es decir, que no se tendría error de estado estable. En este ejemplo sencillo no habría problema en aumentar la ganancia. Pero en sistemas más complejos aumentar la ganancia implicaría transitorios oscilantes o directamente sistemas inestables.

De aquí surge la idea de la acción integral cuya función es eliminar el error de estado estable. El segundo término de la ecuación (1.1) muestra que esta acción es proporcional a la *integral* del error. Por lo tanto este factor dejará de integrar, es decir se estabilizará, solo cuando el error sea nulo que es el objetivo buscado.

Por último el término  $D$  es utilizado para mejorar los transitorios del sistema y el comportamiento frente a perturbaciones. Como lo indica su nombre, su tarea es oponerse a cualquier cambio brusco en el error. Por ejemplo, en los casos en que el error esté próximo al cero pero variando con una cierta velocidad, el término  $P$  no tendría casi efecto, pero sí el aporte derivativo.

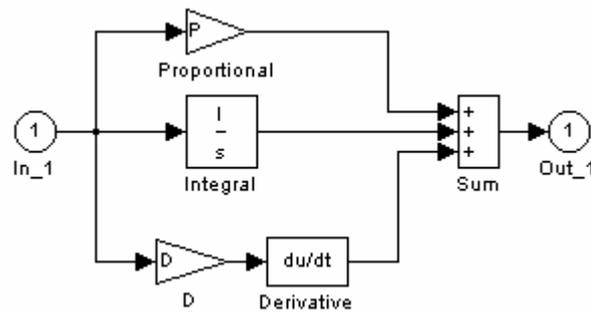
En muchos controladores comerciales se hace una modificación a este aporte definiéndolo como:

$$D = -K T_d \frac{d y(t)}{dt} \quad (1.4)$$

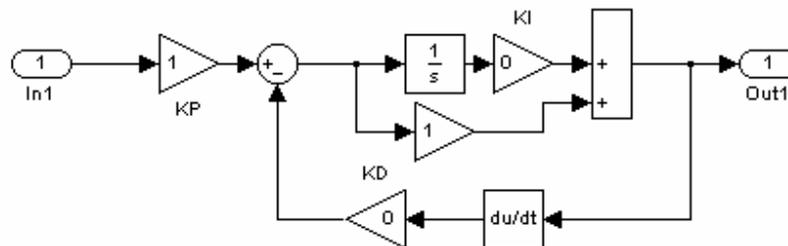
O sea que lo hace proporcional a la velocidad de cambio de la salida. Esto evita que cuando haya cambios en el valor de referencia se produzcan movimientos abruptos de la salida. De todos modos, en estado estable seguirá teniendo las mismas propiedades que el original.

Cabe notar que cuando la derivada de  $y$  es muy alta, por ejemplo en el caso de ruido de alta frecuencia, la acción derivativa puede tener un valor elevado indeseable.

Para el desarrollo de las simulaciones se realizó el siguiente modelo en Simulink de Mathworks, el cual presenta una ligera variación de la estructura ofrecida por el fabricante.



*Figura A.1 Diagrama del PID de Simulink*



*Figura A.2 Diagrama del PID propuesto*

La propuesta se basa en la consideración de que el diseño del PID ofrecido por el fabricante maneja señales en paralelo, por lo que un error grande a la entrada de nuestro controlador promueve una alteración en toda su operación, caso contrario en la otra estructura, pues depende directamente de la salida del mismo para su funcionamiento.

Otra cosa importante es que para la obtención de la derivada se hace uso del bloque de Simulink, motivo por el cual, se hará uso de los diferenciadores basados en la teoría de Modos Deslizantes para ver el desempeño de los mismo y del control con esta variación.

## Ajuste del PID

Existen muchos métodos para el cálculo de  $K$ ,  $T_i$  y  $T_d$  ya sea en forma empírica o analítica. Los más clásicos son los métodos de **Ziegler-Nichols (1942)** tanto a partir de la respuesta al escalón, como de la respuesta en frecuencia.

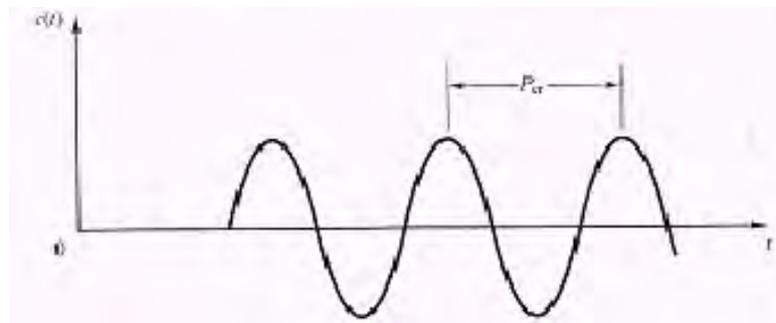
El método de Ziegler-Nichols basado en la respuesta en frecuencia o de lazo cerrado consiste en lo siguiente:

Se realimenta la planta con un regulador proporcional aumentando su ganancia hasta que el sistema presente una oscilación no amortiguada. Se mide la ganancia crítica  $K_c$  y el período  $t_c$  de las oscilaciones ajustando los parámetros del regulador según la tabla siguiente:

CONTROLADOR	$K_p$	$T_i$	$T_d$
P	$0.5 K_c$		
PI	$0.45 K_c$	$0.833 t_c$	
PID	$0.6 K_c$	$0.5 t_c$	$0.125 t_c$

*Tabla A.1 Sintonización PID en lazo cerrado*

Se observará que en la mayoría de las respuestas óptimas aparecerá una oscilación con amortiguamiento  $1/4$ , es decir, que la amplitud de las oscilaciones se reducirá a la cuarta parte en un período, lo que nos indica la presencia aun del error de estado estable.

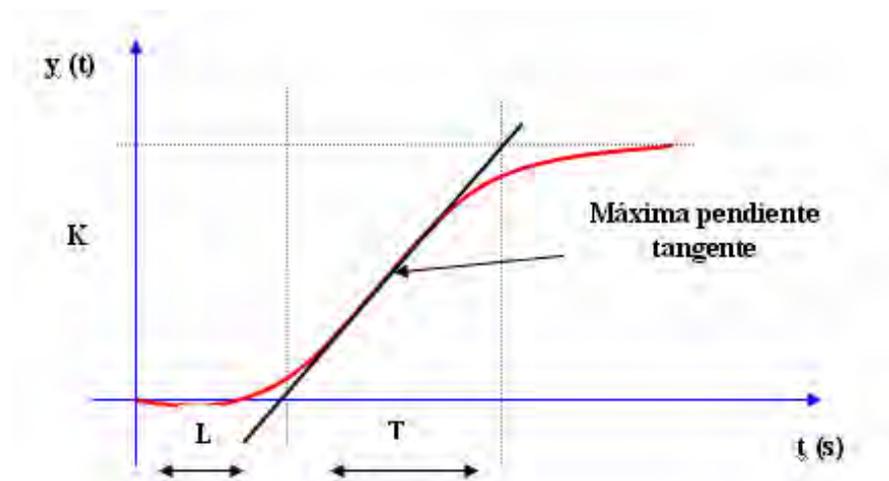


*Figura A.3 Método de Oscilaciones sostenidas de Ziegler-Nichols*

En la regla de Ziegler-Nichols de lazo abierto o de respuesta al escalón, se excita el proceso en lazo abierto con un escalón unitario y se miden sobre la respuesta el tiempo de retardo  $L$  y la constante de tiempo  $\tau$  según se ve en la figura y los parámetros del regulador se fijan de acuerdo a la tabla.

CONTROLADOR	$K_p$	$T_i$	$T_d$
<b>P</b>	$\frac{T}{L}$		
<b>PI</b>	$0.9 \frac{T}{L}$	$\frac{0.3}{L}$	
<b>PID</b>	$1.2 \frac{T}{L}$	$2L$	$\frac{L}{2}$

*Tabla A.2 Sintonización PID en lazo abierto.*



*Figura A.4 Método curva de reacción de Ziegler-Nichols*



# Tarjeta de Control dSPACE- DS 1103

La **DS 1103** [23] es una tarjeta específicamente diseñada para el desarrollo de controladores y simulaciones en tiempo real en campos diversos como robótica, actuadores servo hidráulicos y eléctricos, control de vehículos, etc. También se ajusta bien al procesado genérico de señales y tareas relacionadas.

Está basada en el **DSP** (Digital Signal Processor) de punto flotante Texas Instruments **PPC750GX**, que constituye su unidad principal de proceso. Este procesador proporciona un ciclo rápido de instrucción para algoritmos numéricos intensivos. La potencia del procesador se ve reforzada por una memoria externa que es lo suficientemente rápida para permitir que el procesador nunca deba esperarla ni un solo ciclo al realizar una lectura o escritura. Además esta memoria puede ser accedida desde el host mientras el programa de tiempo real está en ejecución, sin interrumpir la actividad del **DSP** ni interferir con ella, posibilitando de esta forma una fácil monitorización de interacción con el programa en ejecución. Esto es gracias a que se trata de memoria de doble puerto.

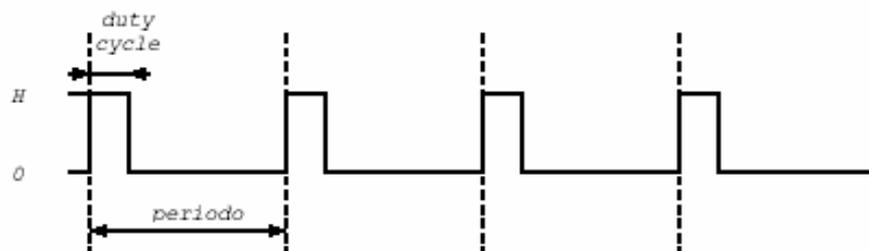
Alrededor del procesador, bajo su control, se sitúan los periféricos que posibilitan la interacción de la tarjeta con el sistema al que se conecta. Para empezar cuenta con ocho DAC's y veinte ADC's, directamente controlados por el procesador.

También cuenta con siete interfaces de encoder incremental, capaces de leer las señales en forma de tren de pulsos generadas por estos dispositivos y convertirlas directamente, mediante hardware, en un valor de ángulo. Utilizaremos estos interfaces en la aplicación de los diferentes controladores en estudio para el control de velocidad del motor de CD. Por último, cuenta con un subsistema de entrada salida digital gobernado por un procesador auxiliar, el Texas Instruments **TMS320F240**. El procesador auxiliar se encarga directamente de la gestión de los periféricos incluidos en este subsistema, descargando así al procesador principal de esta tarea. La forma en que actúa este procesador auxiliar viene determinada por el programa que ejecuta, que puede ser el básico (por defecto), que lee de una EPROM incluida en la tarjeta, o bien uno distinto. El programa básico permite el acceso a los periféricos (bits de E/S digital, un puerto serie, un subsistema de comparación de bits y captura de eventos) y poco más. Para conseguir

comportamientos más sofisticados se puede cargar sobre este procesador un programa que le haga realizar una tarea concreta más elaborada. La tarjeta viene acompañada de una serie de programas para el procesador auxiliar que implementan algunos de estos comportamientos, utilizados frecuentemente, como pueda ser la medición de frecuencias, la generación de PWM's, etc.

## Señales PWM (Pulse Width Modulation)

Una **PWM** es una señal periódica que se mantiene durante una parte del periodo en un cierto valor de tensión  $H$  constante, distinto de cero, y durante el resto del periodo a cero. La proporción del periodo total que la señal se mantiene en el valor distinto de cero se denomina *duty cycle*:



*Figura A.5 Forma de una PWM*

Regulando el *duty cycle* de una **PWM** podemos controlar la intensidad media que pasa a través del circuito alimentado por la **PWM**: esa intensidad media será proporcional al *duty cycle*. El rango de variación  $[0,1]$  para el *duty cycle* cubre el rango  $[0, \text{intensidad máxima}]$  para la intensidad.

El **RTI** de la tarjeta proporciona un bloque Simulink que nos da la posibilidad de generar hasta 6 **PWM**'s distintas, cada una con su *duty cycle*:

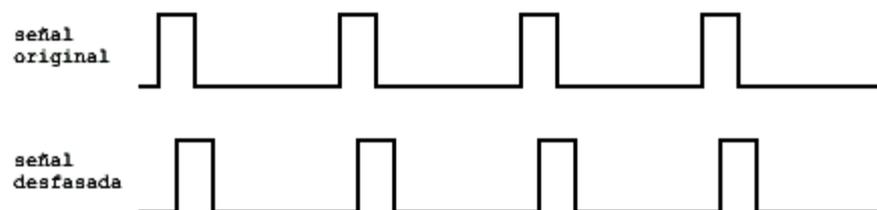
Cada una de las señales tiene asociada una entrada de este bloque. A través de esa entrada se indica dinámicamente para esa señal el *duty cycle*, en el rango  $[0,1]$ . Las señales son generadas a través del subsistema de entrada / salida digital de la tarjeta. Esto hace que los valores entre los

que varía la señal no sean modificables: corresponden a un uno y un cero lógicos, es decir, en esta tarjeta, a 5 y 0 Volts respectivamente.

Con una **PWM** que varía entre 0 y 5 Volts podemos controlar la velocidad de giro del motor en un sentido, pero no hacerlo girar en los dos sentidos. Para conseguir los dos sentidos de giro recurrimos a una electrónica de amplificación con una distribución en **H**, descrita mas adelante, que requiere la generación de dos **PWM's** idénticas, pero tales que cada una se activa para un sólo sentido de giro, aunque, dada la disposición de nuestra etapa de potencia para este trabajo, solo nos bastara con una señal **PWM** para efectuar la misma tarea.

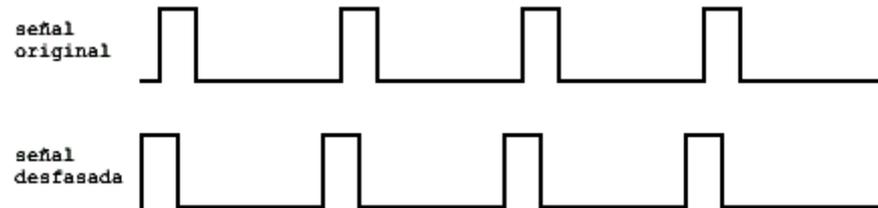
## Encoder Incremental

Un encoder incremental es un dispositivo utilizado para medir ángulos en el giro de un eje. No son capaces de leer directamente el valor absoluto del ángulo, sino que sólo pueden detectar que se ha producido un incremento discreto de cierta magnitud en éste, gracias a un sistema de marcas situadas a intervalos regulares a lo largo de una circunferencia que gira solidaria con el eje cuyo giro se quiere medir. El paso de una de estas marcas por cierta posición es detectable, y generará mientras sea detectada la marca un valor alto en una señal normalmente baja. Un giro mantenido es un tren de pulsos. Para determinar el sentido de giro se genera una segunda señal desfasada de la primera. Ante la detección de un pulso en la primera, la lectura del valor de la segunda determinará si el pulso detectado correspondía a un incremento del ángulo en uno u otro sentido de giro. En un giro en un sentido, tendríamos...



*Figura A.6 Señales del Encoder con giro positivo*

... cuando en la señal original se detecta el principio de una muestra, en la señal desfasada aún no se ha llegado a ella. En cambio, al girar en el sentido contrario...



**Figura A.7** Señal del Encoder con giro negativo

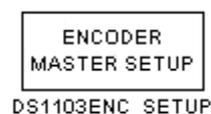
... cuando se detecta el principio de la muestra en la señal original, en la señal desfasada ya hay un nivel alto. Esto nos permite discriminar el sentido de giro.

El número de muescas en la circunferencia se denomina resolución, y va a determinar directamente cuál es el incremento mínimo de ángulo detectable por el encoder (precisión).

Dado que el encoder no detecta ángulos absolutos, se hace necesaria la posibilidad de detectar el paso por un ángulo concreto para determinar cuál es el ángulo cero, y así poder transformar la lectura de incrementos de ángulos en ángulos absolutos.

Esta señal se denomina *index* o microinterruptor.

Para la activación y uso de los Encoder's que dispone la dSPACE es necesario agregar el bloque *Encoder Master Setup*.



**Figura A.8** Bloque de Configuración del Encoder

# Amplificador de Potencia

Como ya se ha comentado en la presente memoria, se ha utilizado un amplificador de potencia con una disposición en **H** de los transistores, para permitir los dos sentidos de giro del motor con una sola fuente de alimentación:

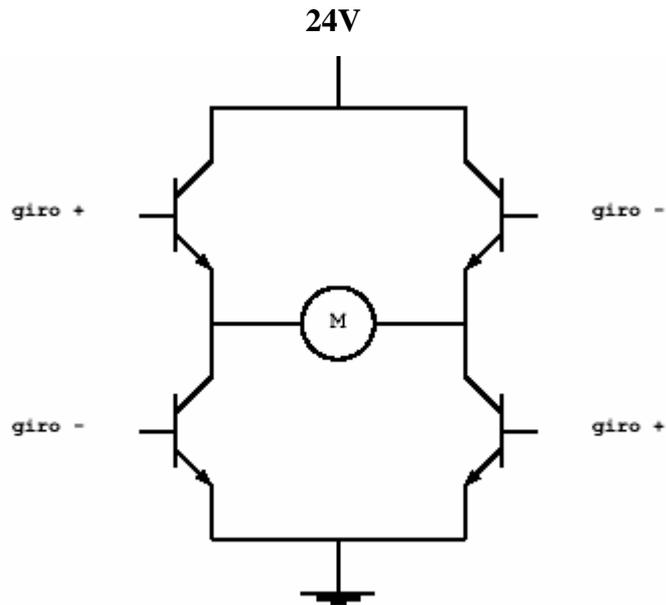


Figura A.9 Diagrama base de la disposición en H de los transistores

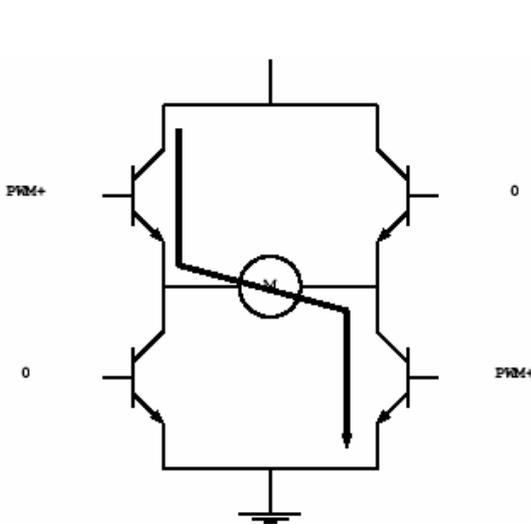


Figura A.10 Camino de Corriente para giro positivo

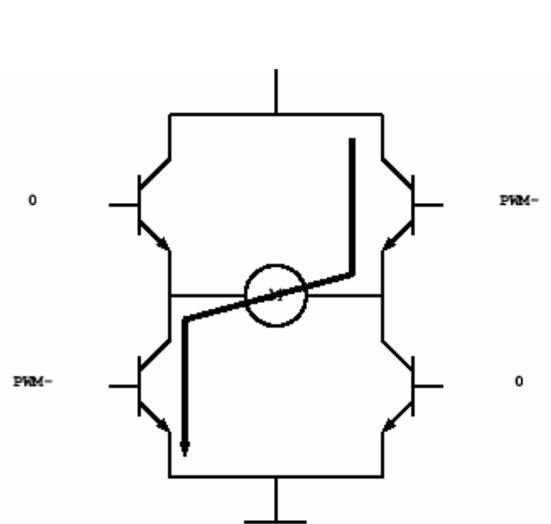
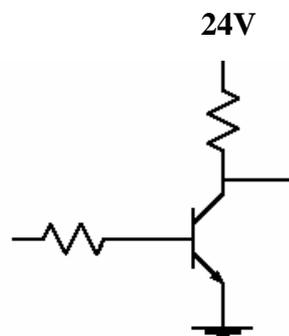


Figura A.11 Camino de Corriente para giro negativo

Dado que necesitamos un factor de amplificación muy alto (de la señal PWM generada por la tarjeta o bien de la señal analógica que proporciona el DAC) de acuerdo a lo que realmente necesita el motor para moverse, cada uno de los transistores que aparecen en las figuras anteriores son en realidad transistores de tipo *mosfet* **IRF-540**.

Pero para los transistores los 5 voltios generados por la tarjeta como uno lógico no son suficientes para que se saturen completamente, debido a la caída de tensión ocasionada por el motor, razón por la cual se hizo necesario elevar la tensión de los unos de la tarjeta hasta una ganancia permisible por la beta del transistor **2N2222**.

Y ello mediante el siguiente circuito inversor:



*Figura A.12* Diagrama del inversor utilizado

Otro problema que surgió en el diseño de esta etapa de amplificación es que un cero lógico de la tarjeta no son 0 Volts, sino un valor diferente. Esto hace que los transistores que reciben en su base ese supuesto cero lógico no estén realmente en corte cuando tendrían que estarlo. Para evitar ese problema se han añadido a la base de los transistores una etapa de lógica combinatorial con dispositivos **TTL**, tal que se cumpla el valor de 0 Volts en la base y que contenga una zona muerta entre cada conmutación, para evitar a su vez, que se encuentre un par de transistores encendido mientras que el otro par este activo, la disposición de los circuitos **TTL** cumple la tabla de verdad de la función **XOR** y sirven a su vez de protección al canal de salida de la tarjeta.



## BIBLIOGRAFÍA

- [1] Y. Takahashi, M.J. Robins, D.M. Auslander. “*Control and Dynamic Systems*”, Addison Wesley Publishing Company, U.S.A., 1972.
- [2] Utkin V.I., “*Sliding Mode Control design principle and applications to electric drives*”, IEEE Transactions on industrial Electronics, 1993.
- [3] Katsuhiko Ogata. “*Ingeniería de Control Moderna*”, Prentice Hall, México, 2003.
- [4] Edward W. Kamen. “*Introducción a Señales y Sistemas*”, CECSA, México, 1996.
- [5] P.H.Lewis and C.Yang. “*Sistemas de Control en Ingeniería*”, Prentice Hall, Madrid, 1999.
- [6] L.Fridman “*Introducción a los Modos Deslizantes*”, <http://verona.fi-p.unam.mx>,
- [7] A. A. Hernández López, “*Implementación del algoritmo cuasi-continuo de modos deslizantes en un motor de CD*”, Tesis, México, 2005.
- [8] G.Bartolini, et al, “*Robust Near- Optimal Control Via Unchattering Sliding Mode Control*”, Cagliari Italia, 1998.
- [9] G.Bartolini, et al, “*Sub-optimal Sliding Mode Control of Uncertain Second Order Dynamical Systems*”, 13<sup>th</sup> World Congress, International federation of Automatic Control, San Francisco, 1996.
- [10] G.Bartolini, et al, “*Second order VSC for non linear systems subject to a wide class of uncertainty conditions*”, IEEE Workshop on VSS, 1996.
- [11] Castellanos G.M.I “*Control por Modos Deslizantes de un motor de CD a pasos*”, Tesis, Chihuahua, Chih, 2003.
- [12] G.Bartolini, et al, “*Chattering Avoidance by second order Sliding Mode Control*”, Cagliari Italia, 1996.
- [13] G.Bartolini, et al, “*Variable Structure Control of Nonlinear Sampled Data Systems by Second Order Sliding Modes*”, Cagliari Italia.
- [14] L.Fridman, A. Levant “*Higher Order Sliding Modes*”, 2003.
- [15] A.Levant “*Quasi Continuous High- Order Sliding Mode controllers*”, IEEE Transactions on Automatic Control, Vol 50, No11, Noviembre 2005.



- [16] A.Levant “*Slidding Order and Slidding Accuracy in Slidding Mode Control*”, International Journal of Control, 1993.
- [17] The MathWorks Inc, “*Using Simulink*”, 1998
- [18] R.Garrido, Y.B.Shtessel, L.M. Fridmam, “*Application of VS differenciator to DC servomechanism*”
- [19] A.Levant “*Higher Order Sliding modes and arbitrary-order exact robust differentiation*”, Proc. Of the European Control conf.2001
- [20] I.A Shkolnikov, Y. B. Shtessel, P.Zarchan. D.P. Lianos “*Simulation study of the homing interceptor guidance loop with sliding mode observers versus kalman filters*”. Tech. Report. Department of Electrical and Computer Engineering. University of Alabama. Huntsville, Al, USA.
- [21] The Math Works Inc, “*Using Real Time Workshop*”, 1998.
- [22] Real-Time Interface (RTI and RTI-MP), “*Implementation Guide*”, dSPACE, Marzo 2004.
- [23] DS1103 PPC Controller Board, “*Hardware Installation and Configuration*”, dSPACE, Mayo 2004.
- [24] Boylestad R., Nashelsky L., “*Electrónica: teoría de circuitos y dispositivos electrónicos*”, Prentice Hall, Octava Edición.
- [25] Damiano A., Gianluca L., Ignacio Marongiu and Alessandro Pisano, “*Second Order Sliding-Mode Control of DC Drives*”, IEEE Transactions on Industrial Electronics, Abril 2004.