

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial



**Implementación y estudio de un sistema de
frenado regenerativo:
Aplicación como carga mecánica**

Titulación: Ingeniería Industrial

Intensificación: Sistemas Eléctricos

Alumno: Manuel Muñoz Arranz

Director: Ángel Molina García

Cartagena 18 de Junio de 2008

Índice

<u>1 Motivación y objetivos.....</u>	<u>11</u>
<u>1.1 Motivación.....</u>	<u>11</u>
<u>1.2 Objetivos.....</u>	<u>11</u>
<u>1.2.1 Resumen.....</u>	<u>11</u>
<u>1.2.2 Fases del proyecto.....</u>	<u>12</u>
<u>2 Problemática de la obtención de par resistente en laboratorio.....</u>	<u>13</u>
<u>2.1 Introducción</u>	<u>13</u>
<u>2.2 Dispositivos Mecánicos.....</u>	<u>14</u>
<u>2.3 Dispositivos Hidráulicos.....</u>	<u>15</u>
<u>2.4 Dispositivos Eléctricos.....</u>	<u>16</u>
<u>2.4.1 Introducción.....</u>	<u>16</u>
<u>2.4.2 Freno de polvo magnético.....</u>	<u>16</u>
<u>2.4.3 Freno de corrientes parásitas.....</u>	<u>17</u>
<u>2.4.4 Generadores de C.C.....</u>	<u>18</u>
<u>2.4.5 Generadores síncronos.....</u>	<u>18</u>
<u>2.4.6 Máquina síncrona controlada con electrónica de potencia.....</u>	<u>19</u>
<u>2.5 Alternativas comerciales.....</u>	<u>19</u>
<u>2.5.1 Frenos dinamómetros hidráulicos (Tecner Ingeniería S.A.).....</u>	<u>19</u>
<u>2.5.2 Freno electrónico (EDIBON).....</u>	<u>20</u>
<u>2.5.3 Puesto de pruebas para máquinas con sistema de servofreno (LUCAS-NÜELLE).....</u>	<u>21</u>
<u>2.6 Consideraciones finales.....</u>	<u>22</u>
<u>3 Alternativa propuesta: Descripción de componentes.....</u>	<u>23</u>
<u>3.1 Variador de frecuencia.....</u>	<u>24</u>
<u>3.1.1 Principio de funcionamiento.....</u>	<u>24</u>
<u>3.1.2 Solución comercial empleada.....</u>	<u>26</u>
<u>3.1.2.1 Variador de frecuencia UNI 1404.....</u>	<u>26</u>
<u>3.1.2.2 Variador de frecuencia SP1401.....</u>	<u>27</u>
<u>3.2 Módulo SM-Ethernet.....</u>	<u>30</u>
<u>3.3 Servo.....</u>	<u>32</u>
<u>3.3.1 Principio de funcionamiento.....</u>	<u>32</u>
<u>3.3.2 Solución comercial empleada.....</u>	<u>33</u>
<u>3.3.2.1 Encoder Incremental.....</u>	<u>34</u>
<u>3.4 Switch o conmutador de red.....</u>	<u>35</u>
<u>3.4.1 Principio de funcionamiento.....</u>	<u>35</u>
<u>3.4.2 Solución comercial empleada.....</u>	<u>36</u>

<u>4 Configuración y conexionado del equipo.....</u>	<u>37</u>
<u>4.1 Conexiones de control.....</u>	<u>37</u>
<u>4.1.1 Interconexión Variador-Servo.....</u>	<u>38</u>
<u>4.1.2 Interconexión variador-PC.....</u>	<u>39</u>
<u>4.1.3 Conexiones de control del variador de frecuencia (Motor).....</u>	<u>42</u>
<u>4.1.4 Conexiones de control del variador de frecuencia (Servo).....</u>	<u>43</u>
<u>4.2 Conexiones de potencia.....</u>	<u>44</u>
<u>4.2.1 Alimentación del motor asíncrono.....</u>	<u>44</u>
<u>4.2.2 Alimentación del servo.....</u>	<u>45</u>
<u>4.2.3 Evacuación de Energía.....</u>	<u>45</u>
<u>4.2.3.1 Resistencia de frenado.....</u>	<u>46</u>
<u>4.2.3.2 Devolución de energía a la red.....</u>	<u>47</u>
<u>4.2.3.2.1 Montaje Standard:.....</u>	<u>48</u>
<u>4.2.3.2.2 Sustitución de la resistencia de frenado:.....</u>	<u>48</u>
<u>4.2.3.2.3 Regeneración Múltiple.....</u>	<u>49</u>
<u>5 Configuración de componentes.....</u>	<u>54</u>
<u>5.1 Variador de Frecuencia (Motor).....</u>	<u>54</u>
<u>5.2 Variador de Frecuencia (Servo).....</u>	<u>55</u>
<u>5.3 Aplicación web.....</u>	<u>56</u>
<u>5.3.1 Introducción. Lenguajes de programación.....</u>	<u>56</u>
<u>5.3.1.1 HTML.....</u>	<u>57</u>
<u>5.3.1.2 Javascript.....</u>	<u>57</u>
<u>5.3.1.3 Editor de código fuente HTML y JavaScript (BlueFish).....</u>	<u>57</u>
<u>5.3.2 Desarrollo de las aplicaciones.....</u>	<u>58</u>
<u>5.3.3 Carga de la aplicación web en el módulo SM-Ethernet. El cliente FTP.....</u>	<u>63</u>
<u>5.3.3.1 Configuración del cliente FTP.....</u>	<u>64</u>
<u>5.3.4 Acceso y empleo de la aplicación web. El navegador web Firefox 2.0.....</u>	<u>65</u>
<u>5.3.5 Acceso a interfaz de control.....</u>	<u>65</u>
<u>6 Ensayos y resultados.....</u>	<u>72</u>
<u>6.1 Estudio de la potencia disipada en la resistencia de frenado.....</u>	<u>72</u>
<u>6.1.1 Introducción.....</u>	<u>72</u>
<u>6.1.2 Procedimiento de experimentación.....</u>	<u>73</u>
<u>6.1.3 Configuración de instrumentación para la medición de parámetros.....</u>	<u>74</u>
<u>6.1.4 Análisis de resultados.....</u>	<u>77</u>
<u>6.2 Otros ensayos.....</u>	<u>82</u>
<u>6.2.1 Análisis de las magnitudes de tensión e intensidad en el sistema.....</u>	<u>82</u>
<u>7 Conclusiones.....</u>	<u>86</u>

8 ANEXOS.....	88
8.1 ANEXO 1: Estructura de la página web por defecto instalada en el módulo SM-Ethernet....	88
8.2 ANEXO 2: Esquema de control por parámetros del variador de frecuencia SP1401.....	89
8.2.1 Diagrama de control de velocidad.....	89
8.2.2 Diagrama de control de par.....	90
8.3 ANEXO 3: Código fuente de las páginas web de control.....	91
8.3.1 Página index.htm.....	91
8.3.2 Página inicial.htm.....	94
8.3.3 Página servo_motor.htm.....	96
8.3.4 Página servo_par.htm.....	98
8.3.5 Página servo_veloc.htm.....	103
8.3.6 Página servo-gener.htm.....	107
8.3.7 Librería smethernet.htm.....	115
9 Bibliografía.....	160
10 Referencias web.....	161

Índice de ilustraciones

Figura 1 Par Constante.....	13
Figura 2 Par lineal con la velocidad.....	13
Figura 3 Par parabólico con la velocidad.....	14
Figura 4 Freno de Fricción. Principio de funcionamiento.....	15
Figura 5 Freno dinamómetro hidráulico.....	15
Figura 6 Freno de polvo magnético.....	17
Figura 7 Freno de corrientes parásitas.....	17
Figura 8 Generador de C.C. Excitación independiente.....	18
Figura 9 Freno Hidráulico. Serie V.....	20
Figura 10 Freno hidráulico Serie SV.....	20
Figura 11 Unidad de frenado (EDIBON).....	21
Figura 12 Puesto de pruebas LUCAS-NÜELLE.....	21
Figura 13 Servomáquina Asíncrona LUCAS-NÜELLE.....	22
Figura 14 Alternativa propuesta. Diagrama de Funcionamiento.....	23
Figura 15 Diagrama de bloque del Variador de Frecuencia.....	24
Figura 16 Esquema básico de un variador de frecuencia.....	25
Figura 17 Modulación por ancho de pulso (PWM).....	26
Figura 18. Variador de frecuencia UNI1404.....	27
Figura 19. Variador de frecuencia SP1401.....	27
Figura 20. Teclado extraíble del variador de frecuencia.....	28
Figura 21. Desplazamiento entre menus y parámetros del variador de frecuencia SP1401.....	29
Figura 22. Módulo SM-Ethernet.....	30
Figura 23. Ranuras para módulos de aplicaciones.....	31
Figura 24. Esquema de conexión y secuencia de conmutación de diodos.....	32
Figura 25. Curvas características de un servo.....	33
Figura 26. Servo.....	33
Figura 27. Servo-motor.....	34
Figura 28. Relación Par velocidad del servo motor.....	34
Figura 29. Disco interior de un encoder incremental.....	35
Figura 30. Conmutador de red SMC modelo EZ6508TX.....	36
Figura 31. Esquema general de conexionado.....	37
Figura 32. Cable de 17 vias.....	38
Figura 33. Conexión PC-Módulo SM-Ethernet.....	39
Figura 34. Conexión Ethernet y LEDs de estado del módulo SM-Ethernet.....	39
Figura 35. Información de conexión en sistema operativo Windows (MsDOS).....	41
Figura 36. Información de conexión en sistema operativo Linux (Ubuntu 7.10).....	42
Figura 37. Conexiones de control UNI1404.....	43
Figura 38. Conexiones de control SP1401.....	43
Figura 39. Configuración de conexión en estrella y en triángulo.....	44

Figura 40. Terminales de potencia del variador de frecuencia UNII404.....	44
Figura 41. Conexiones de potencia del variador de frecuencia SP1401.....	45
Figura 42. Diagrama del sistema resistencia de frenado.....	46
Figura 43. Conexión de la resistencia de frenado.....	47
Figura 44. Potenciómetro.....	47
Figura 45 Configuración Standard del sistema de regeneración.....	48
Figura 46. Configuración de sustitución de la resistencia de frenado del sistema de regeneración..	48
Figura 47. Configuración Regeneración Múltiple.....	49
Figura 48 Configuración Standard. Conexionado de potencia.....	50
Figura 49 Configuración Standard. Conexionado de control.....	51
Figura 50 Sustitución de la resistencia de Frenado. Conexionado de potencia.....	52
Figura 51 Sustitución de la resistencia de Frenado. Conexionado de Control.....	53
Figura 52. Placa de características de un motor asíncrono.....	54
Figura 53. Placa de características de un servomecanismo.....	55
Figura 54. Captura de pantalla del editor BlueFish.....	58
Figura 55. Esquema de desarrollo de la aplicación web.....	59
Figura 56. Esquema de manipulación de parámetros.....	61
Figura 57. Relación Par Velocidad mediante la aplicación web.....	62
Figura 58. Diagrama de velocidad en el tiempo en servo-gener.htm.....	63
Figura 59. Gráfica Par-Velocidad en servo-gener.htm.....	63
Figura 60. Captura de pantalla del programa Ws FTP.....	64
Figura 61. Inicio de sesión en el programa Ws FTP.....	65
Figura 62. Página principal del servidor.....	66
Figura 63. Pagina de registro de usuario.....	67
Figura 64. Diagrama de navegación por páginas web creadas.....	67
Figura 65. Página index.htm.....	68
Figura 66. Página inicial.htm.....	68
Figura 67. Página servo_veloc.htm.....	69
Figura 68. Página servo_par.htm.....	69
Figura 69. Advertencia inicial en servo_veloc.htm y servo_par.htm.....	69
Figura 70. Página servo_gener.htm.....	70
Figura 71. Detalle del acoplamiento motor asíncrono al servo.....	71
Figura 72. Onda de voltaje en la resistencia de frenado.....	72
Figura 73. Curva de la máquina asíncrona y regímenes de funcionamiento.....	73
Figura 74. Diagrama de toma de datos.....	74
Figura 75. Sondas de medida en el variador del motor.....	74
Figura 76. Placa de conexión a la tarjeta de adquisición de datos.....	75
Figura 77. Detalle de configuración de la pinza amperimétrica.....	75
Figura 78. Menú del programa de adquisición de datos.....	75
Figura 79. Pantalla de adquisición y representación de datos.....	76
Figura 80. Gráfica del voltaje de la resistencia en función del tiempo a distintos valores del par resistente.....	77

Figura 81. Gráfica de la relación entre el periodo de la onda de voltaje de la resistencia y la potencia disipada en la resistencia.....	78
Figura 82. Gráfica de la relación entre el periodo de la onda de voltaje de la resistencia y la potencia en el eje mecánico.....	78
Figura 83. Gráfica de la relación par resistente y velocidad del eje.....	79
Figura 84. Gráfica de la relación par resistente y potencia en el eje.....	80
Figura 85. Gráfica de la relación par resistente y rendimiento energético del sistema.....	81
Figura 86. Gráfica de la relación potencia disipada en la resistencia eléctrica y velocidad del eje...	82
Figura 87. Intensidad de línea del motor asíncrono.....	83
Figura 88. Intensidad de línea del servomecanismo.....	84
Figura 89. Voltaje de línea del motor asíncrono.....	84
Figura 90. Voltaje de línea del servomecanismo.....	85

Índice de tablas

Tabla 1. Parametros del variador de frecuencia SP1401.....	29
Tabla 2. Parámetros del módulo SM-Ethernet.....	31
Tabla 3. Conexiones del cable de conexión Variador-Máquina Síncrona.....	38
Tabla 4. Conexión de terminales RJ45.....	40
Tabla 5. Relación de octetos de configuración IP con parámetro del módulo SM-Ethernet.....	41
Tabla 6. Cable de Potencia de la máquina síncrona (lado servo).....	45

1 Motivación y objetivos

1.1 Motivación

La motivación por la realización de este Proyecto Fin de Carrera es la obtención del título de Ingeniero Industrial por la Universidad Politécnica de Cartagena.

1.2 Objetivos

El objetivo principal es el estudio y posterior implementación de un equipo para la generación de un par resistente variable y parametrizable por el usuario a través de un computador, con el fin de simular los diferentes tipos de cargas resistentes usuales en la industria, además de conseguir el aprovechamiento posterior de la energía producida en el frenado.

En concreto, se desarrollará un prototipo que permita la generación de una curva de par/velocidad a definir por el usuario, con el equipamiento disponible en el laboratorio de máquinas eléctricas del Departamento de Ingeniería Eléctrica, así como la experimentación del mismo bajo determinadas condiciones.

1.2.1 Resumen

En el apartado de las máquinas eléctricas, el establecimiento de pares resistentes en función de la velocidad, puede llevarse a cabo mediante distintos tipos de dispositivos. Sin embargo, el usuario puede encontrarse con dificultades a la hora de fijar dichas curvas, haciéndose necesario el uso de equipos de electrónica de potencia para su regulación y sistemas computacionales para su configuración.

El presente proyecto está basado en un servomecanismo que provoca el par resistente, fijando los parámetros mediante un variador de frecuencia, con el objetivo de regularlo en tiempo real en función de su velocidad, y cuyo manejo será vía PC.

1.2.2 Fases del proyecto

- Recopilación de información y estudio de los distintos dispositivos utilizados en el mercado relacionados con aquellos a utilizar en este Proyecto Fin de Carrera, realizando las pertinentes descripciones y comparativas de cada uno de ellos.
- Descripción de los equipos a utilizar
- Montaje y configuración de equipos
- Programación de la interfaz para la comunicación entre el usuario y el equipo
- Realización de ensayos, toma de datos y análisis de resultados. Problemática de la obtención de par resistente en laboratorio

2 Problemática de la obtención de par resistente en laboratorio

2.1 Introducción

Cuando un motor eléctrico desempeña una función dentro de una industria se ve sometido a vencer unas exigencias de par resistente. Dependiendo del tipo de instalación en la que funciona el motor, dicho par posee una dependencia diferente con la velocidad angular, las más usuales son:

1. **Par constante**, prácticamente independiente con la velocidad de rotación (Figura 1). Ejemplos: grúas, guindastes, transportadoras de correas bajo par constante.

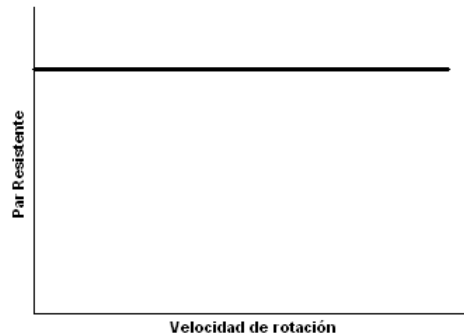


Figura 1 Par Constante

2. **Par que varía linealmente con la velocidad de rotación** (Figura 2). Algunos ejemplos de este tipo de curva Par-velocidad son molinos de rodillos, bombas de pistón, bombas de vacío y compresores

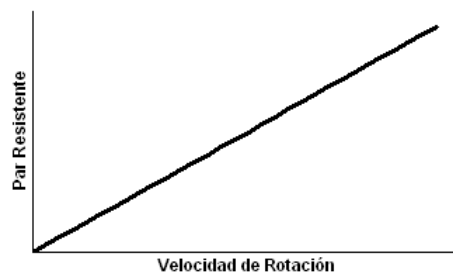


Figura 2 Par lineal con la velocidad

3. **Par que varía con el cuadrado de la velocidad de rotación** (variación parabólica) (Figura 3).
Ejemplos: Ventiladores, mezcladoras, centrifugadoras, bombas centrífugas.

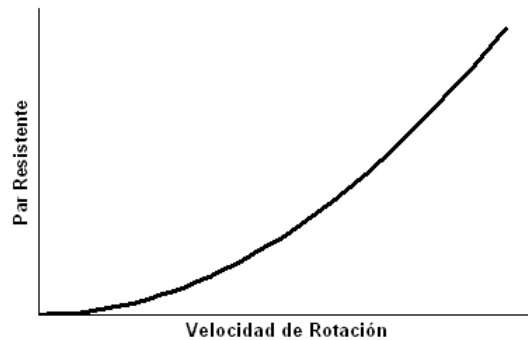


Figura 3 Par parabólico con la velocidad

Sin embargo en un entorno de laboratorio no es posible disponer habitualmente de este tipo de instalaciones, por lo que surge la necesidad de reproducir las condiciones de par resistente a través de dispositivos alternativos. Estos dispositivos se dividen en:

- Dispositivos mecánicos o de fricción.
- Dispositivos hidráulicos
- Dispositivos eléctricos

2.2 Dispositivos Mecánicos

El mecanismo por el cual está formado es relativamente básico (freno de fricción, Figura 4) consistente en dos piezas, una solidaria al eje (disco) y otra fija (zapata) a la que se aplica una fuerza perpendicular al eje para generar una fricción entre ambas piezas. Debido al rozamiento entre las piezas, se absorbe energía mecánica que el freno disipa en forma de calor.

Los inconvenientes de este sistema son su falta de fiabilidad, la dificultad de refrigeración así como la imposibilidad de conocer la dependencia real entre par y velocidad, además de un riesgo de fallo por desgaste de material que provoque la inutilidad del dispositivo. Su sencillez y su bajo precio son las principales ventajas que encontramos.

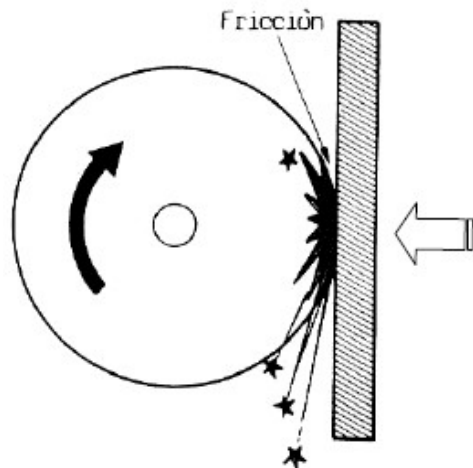


Figura 4 Freno de Fricción. Principio de funcionamiento

2.3 Dispositivos Hidráulicos

Durante el funcionamiento, el agua de los alvéolos del rotor es expulsada a gran velocidad por la acción de la fuerza centrífuga, introduciéndose en los alvéolos del estator por el perímetro externo. Estos últimos poseen una forma tal que hace que el agua retorne a los alveolos del rotor a menor velocidad por la parte más próxima al eje de rotación. De esta forma la trayectoria del agua es helicoidal, produciendo el torbellino de este movimiento un efecto de frenado entre el rotor y el estator. Como consecuencia del rozamiento entre las superficies del freno y el agua existente en su interior, la resistencia que el agua opone al giro del rotor reacciona sobre el estator, produciendo un par igual al par motor.

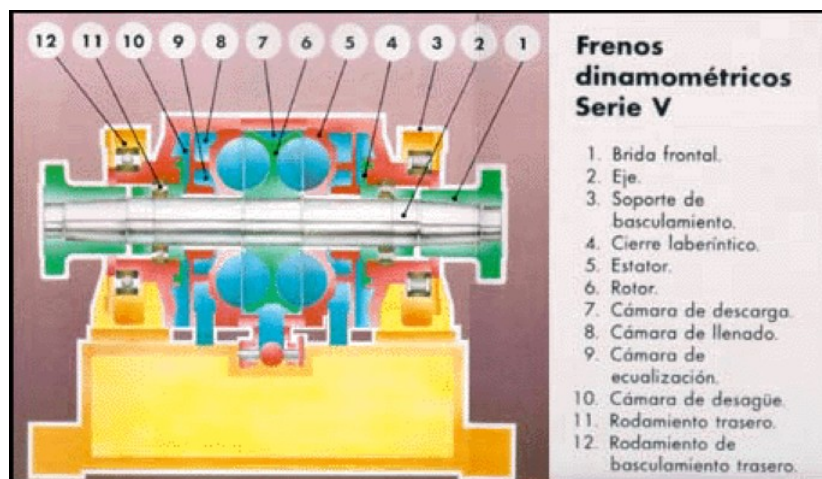


Figura 5 Freno dinamómetro hidráulico

Determinados modelos de los frenos hidráulicos Schenck tienen las partes móviles dispuestas exteriormente al estator, el cual está montado en forma basculante y unido al sistema de medida de fuerza. Esta disposición permite añadir un freno de fricción sobre la cara exterior del rotor para aumentar el par de frenado a bajo régimen. La regulación de la carga resistente se puede efectuar interponiendo unas compuertas en el espacio entre los alvéolos del estator y del rotor. Accionando un volante exterior al estator se aproximan o alejan del eje de rotación las dos compuertas, haciendo inactivo a los efectos de frenado un número menor o mayor de cavidades.

En la regulación por nivel de líquido, al girar el rotor el agua que hay dentro del freno sufre la acción de las fuerzas centrífugas formando un anillo, cuyo espesor se puede regular estrangulando la salida o la entrada de agua. (Figura 5). Sus principales ventajas son su fácil mantenimiento ya que no existen partes en rozamiento, alta velocidad de giro y que son compactos y ligeros.

Los inconvenientes que presenta son la necesidad de un aporte continuo de agua, para disipar el calor absorbido durante el frenado, y la dependencia de par de frenado con la presión de agua de abastecimiento.

2.4 Dispositivos Eléctricos

2.4.1 Introducción

Los dispositivos eléctricos basan su efecto en múltiples propiedades eléctricas y magnéticas de los componentes que forman parte de ellos, por lo que existe a su vez una clasificación de los mismos, descritos a continuación.

2.4.2 Freno de polvo magnético

Su principio de funcionamiento es el rozamiento entre partículas de polvo férrico sometidas a un campo magnético, el polvo férrico tiene un comportamiento similar a un líquido viscoso, siendo la viscosidad creciente con la intensidad del campo magnético. El freno consta de un rotor y un estator entre los cuales existe un entrehierro, dentro del cual se sitúa el polvo férrico.

En el estator se sitúa una bobina por la que circula una corriente de excitación que produce un campo magnético que se cierra entre el rotor, el entrehierro y el estator (Figura 6). Las ventajas de este sistema son que el par de giro es proporcional a la intensidad de excitación, e independiente de la velocidad, pudiéndose aplicar par desde velocidad cero.

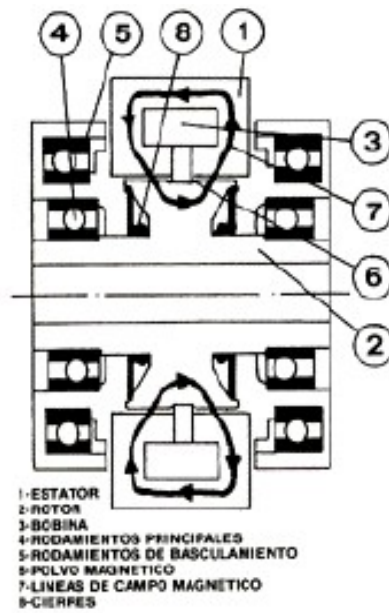


Figura 6 Freno de polvo magnético

2.4.3 Freno de corrientes parásitas

El principio de frenado es la creación de corrientes parásitas por un rotor dentado que gira en el seno de un campo magnético estacionario, producido por un electroimán anular y que constituye el estator del freno. Las corrientes parásitas se producen en la zona del estator enfrentada a los dientes del rotor, y se manifiestan en forma de calor, que es eliminado por el agua de refrigeración. El par frenante es regulado variando la excitación de la bobina del estator, con resolución prácticamente infinita y con una elevada velocidad de respuesta.

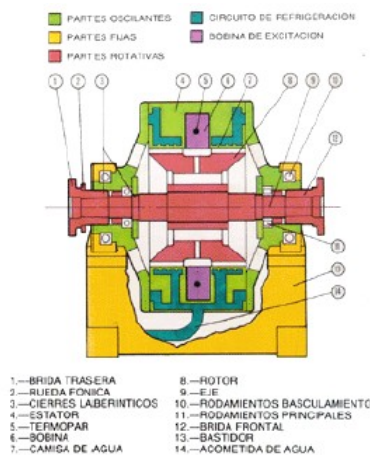


Figura 7 Freno de corrientes parásitas

2.4.4 Generadores de C.C.

Convierten la energía mecánica del eje en energía eléctrica en forma de corriente continua. Constan de un circuito inductor o de excitación y de un inducido giratorio provisto de un colector de delgas. Entre las formas de excitación posibles, la más recomendable es la excitación independiente, en ese caso el inducido gira dentro del campo magnético creado por el inductor, el cual es alimentado por una fuente externa y crea una f.em. aproximadamente proporcional a la velocidad de rotación y al flujo magnético existente entre los polos.

Como inconvenientes se tiene la existencia de escobillas, la necesidad de una fuente de energía externa, que ha de ser de corriente continua y regulable para generar el campo magnético, y la necesidad de cargas externas para disipar la energía obtenida durante el frenado del motor.

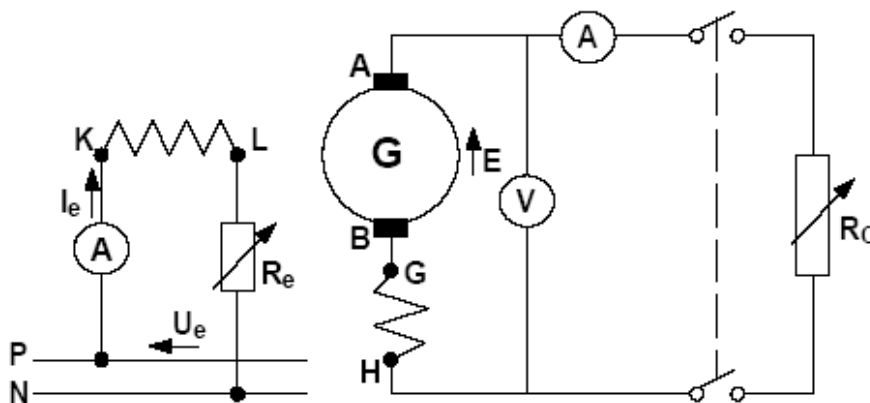


Figura 8 Generador de C.C. Excitación independiente

2.4.5 Generadores síncronos

Los generadores síncronos son máquinas eléctricas cuya frecuencia de salida está íntimamente relacionada con la velocidad de rotación del eje. Los sistemas de excitación más empleados son: la excitación con dinamo excitatriz y los sistemas de excitación electrónica. En nuestro caso, y debido a la pequeña potencia necesaria, se utilizaría la excitación independiente, suministrada por una fuente de corriente continua y regulable; o por medio de un inductor con imanes permanentes.

Al igual que la máquina de C.C. presenta los inconvenientes de necesitar cargas donde disipar la energía generada. La existencia de las escobillas en algunos sistemas de excitación son los mayores inconvenientes de su utilización.

2.4.6 Máquina síncrona controlada con electrónica de potencia

Consiste en una máquina síncrona trabajando como generador. En este caso se trata de un servo-motor brushless controlado mediante un variador de frecuencia. Este sistema, pese a ser más caro que los anteriores, es mucho más versátil; ya que permite que el servo-motor funcione en los cuatro cuadrantes. Además es posible fijar el valor del par resistente disponible, existiendo varias alternativas para evacuar la potencia absorbida en el frenado como pueden ser: disipar la energía en resistencias de frenado o inyectarla de nuevo a la red en forma de corriente alterna.

2.5 Alternativas comerciales

Existen alternativas comerciales de todos los dispositivos enumerados anteriormente. De ellos, los más destacables se resumen en los siguientes apartados.

2.5.1 Frenos dinamómetros hidráulicos (Tecner Ingeniería S.A.)

TECNER, ofrece frenos dinamómetros hidráulicos *Serie V* (Figura 9) y *Serie SV* (Figura 10). Existen frenos desde 2400 rpm hasta las 11000 rpm, y con una gama de potencias desde los 230 kW hasta los 3600 kW, constan de un regulador electrónico que recibe información de par y velocidad, y mantienen estos valores prefijados independientemente de las actuaciones del motor. Estos frenos ofrecen distintos modos de control para el control de la velocidad y del par, estos modos son:

- *Modo «n constante» velocidad constante:* La velocidad del sistema motor-freno se mantiene constante e independiente de las variaciones del par del motor. Se lee o registra la velocidad (consigna) y el par motor. Este modo cubre la prácticamente totalidad de los ensayos efectuados con motores térmicos y eléctricos. El regulador dispone de ajustes de ganancia PID (Proporcional, Integral y Derivativa) para optimización de la respuesta del control.
- *Modo «M constante» par constante:* El par motor se mantiene constante independientemente de las variaciones de velocidad del sistema freno-motor. Se lee o registra el par motor (consigno) y la velocidad de giro. Este modo puede ser útil en alguna aplicación específica con motores eléctricos que tengan una pendiente descendente muy fuerte par-velocidad (punto de corte de los reguladores de motores).
- *Otros modos:* Opcionalmente puede suministrarse respuesta cuadrática (modo n^2) o válvula fija (modo constante), si bien no son imprescindibles disponiendo el freno de modos «n» y «M» constante.

Control exterior: Los frenos Tecner Serie V están previstos para control directo exterior por señales eléctricas de tensión, que pueden provenir de controladores exteriores, como autómatas programables y ordenadores de cualquier tipo. Al estar incorporada la lógica del control en el regulador del freno no es preciso prever ningún bucle de control exterior.

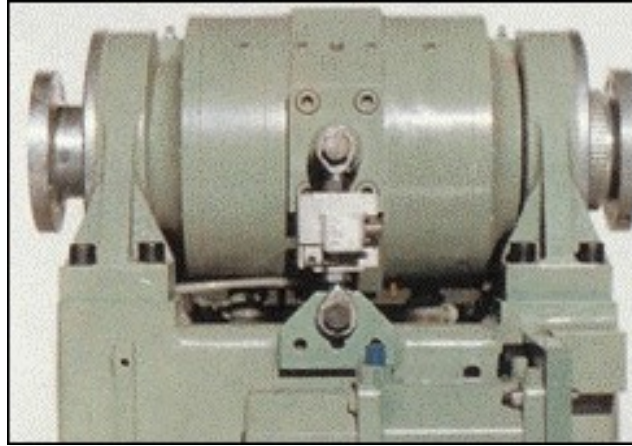


Figura 9 Freno Hidráulico. Serie V



Figura 10 Freno hidráulico Serie SV

2.5.2 Freno electrónico (EDIBON)

Este equipo es capaz de fijar un par de frenado ajustable en sentido positivo o negativo, lo que permite el funcionamiento del equipo en los 4 cuadrantes. La energía regenerada es devuelta directamente a la red con un factor de potencia ajustable. Está formado por una primera etapa que rectifica la tensión de red, un condensador de LINK que filtra la tensión obtenida de la primera etapa, y una tercera etapa que es la encargada de la conversión CC/CA ajustando el valor de tensión

y frecuencia a las exigencias de frenado. También puede funcionar como variador de frecuencia para un motor de CA reversible.(Figura 11)

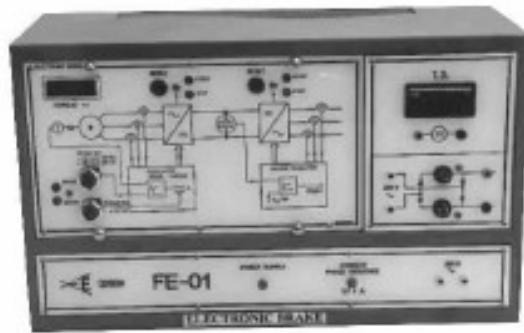


Figura 11 Unidad de frenado (EDIBON)

2.5.3 Puesto de pruebas para máquinas con sistema de servofreno (LUCAS-NÜELLE)

Este equipo consta de una unidad de control didáctica y una servomáquina asíncrona con resolver integrado, el sistema puede actuar en los cuatro cuadrantes, consta de instrumentos de control y medida de la velocidad y del par. Además dispone del modelo de trabajo de las seis máquinas de trabajo más usuales, las cuales son:

- Accionamiento elevador
- Calandria
- Ventilador / bomba
- Accionamiento bobinador
- Masa volante
- Compresor de émbolo



Figura 12 Puesto de pruebas LUCAS-NÜELLE



Figura 13 Servomáquina Asíncrona LUCAS-NÜELLE

2.6 Consideraciones finales

Revisando el apartado anterior, podemos ver que las distintas alternativas que se nos presentan no dan la suficiente flexibilidad buscada y que se ajuste a las exigencias deseadas en cuanto a control de par, a lo que habremos de sumar el elevado coste del equipo, haciendo en principio desaconsejable el uso de cualquier alternativa comercial expuesta. Sin embargo, un control adecuado del par puede ser realizado mediante el uso de la electrónica de potencia, y como se verá a continuación, un control gracias a la utilización de un dispositivo externo a la propia máquina que genere el par.

Para ello, la propuesta consta de una solución basada en equipos industriales. La flexibilidad del sistema se basa en que está constituido de manera modular, permitiendo que ciertos equipos, como el motor asíncrono, el servo o los variadores de frecuencia, puedan ser sustituidos por otros de prestaciones distintas, sin que ello afecte al sistema conjunto, más que en lo referente a la configuración de parámetros.

Además, esto desembocará en una transformación de conocimientos prácticos a los alumnos que reproduzcan dicho montaje y experimentación que sean llevados a cabo, ya que se podrá visualizar el uso de un servo para la generación de pares resistentes.

3 Alternativa propuesta: Descripción de componentes

Debido a que ninguna de las soluciones comerciales expuestas anteriormente satisface nuestras necesidades, decidimos formar nuestro propio banco de pruebas. Para ello optamos por conjugar dispositivos de uso industrial y de laboratorio, que ya están a nuestra disposición con el fin de obtener un par resistente que sea totalmente controlado por el usuario, estableciendo él mismo la relación Par-Velocidad que desee atendiendo a los perfiles típicos. Para esto partimos de un diagrama de funcionamiento:

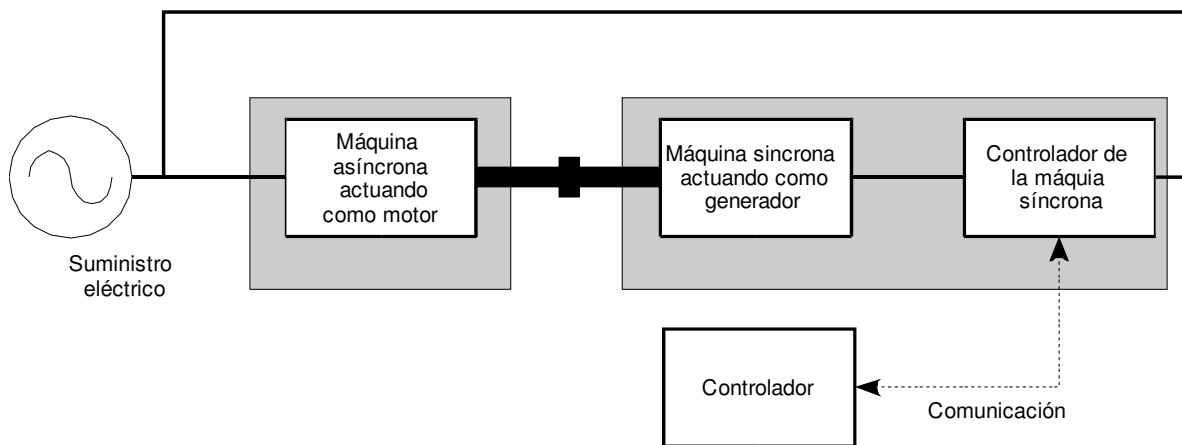


Figura 14 Alternativa propuesta. Diagrama de Funcionamiento.

En primer lugar la máquina asíncrona actuando como motor y accionada por cualquiera de los métodos existentes en la industria (arrancador electrónico, variador de frecuencia, etc...) es la encargada de proporcionar el par motor, el servo actuando como generador es la encargada de proporcionar el par resistente. El usuario, a través del controlador, configura el par resistente cuya orden pasa al controlador del servo, y este último modifica los parámetros eléctricos aplicados a la máquina síncrona para conseguir el par deseado. Los componentes necesarios para realizar este sistema son:

- Variador de frecuencia
- Modulo SM-Ethernet
- Servo
- Encoder incremental
- Switch o conmutador de red
- Dispositivo con navegador web

3.1 Variador de frecuencia

3.1.1 Principio de funcionamiento

El principio de funcionamiento de un variador de frecuencia es transformar la energía eléctrica de frecuencia y tensión RMS constantes en energía eléctrica de tensión y frecuencia variables, haciendo que el coeficiente V/f sea fijo para conseguir mantener constante el flujo magnético de la máquina. La misión principal de este tipo de dispositivos es la variación de velocidad en motores asíncronos, debido a las características de funcionamiento de los mismos, y, en servomotores, permite un control de la posición, la velocidad y del par suministrado, permitiendo su uso en múltiples operaciones.

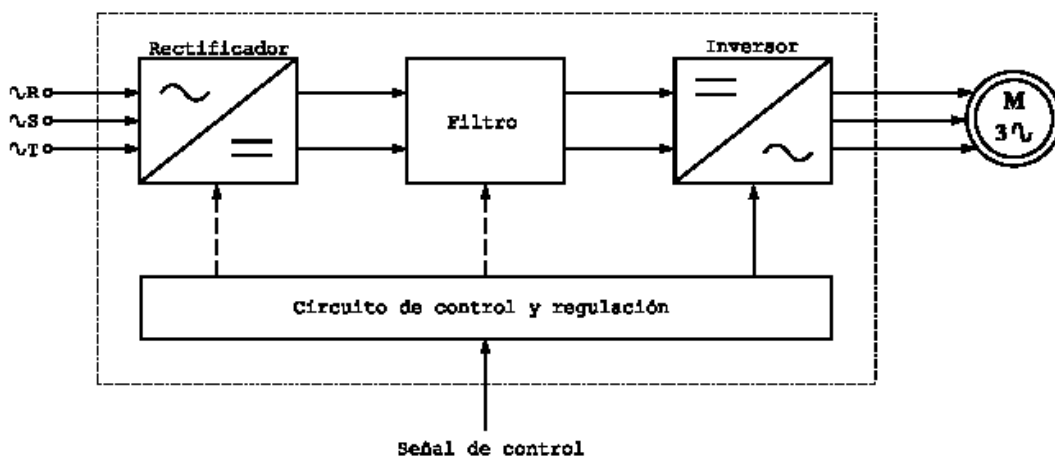


Figura 15 Diagrama de bloque del Variador de Frecuencia

El variador está formado por dos etapas en serie (Figura 15 y Figura 16), una etapa rectificadora que transforma la tensión alterna en continua, y que está formada principalmente por un puente de diodos; y otra etapa inversora que transforma la tensión continua en alterna de frecuencia y tensión regulables, los cuales dependerán de los valores de consigna introducidos a partir de señales de control.

Esta segunda etapa está formada por un puente de transistores de potencia como el IGBT (transistor bipolar de puerta aislada). En la etapa de corriente continua hay un condensador cuya misión es estabilizar la tensión, y un transistor IGBT encargado de habilitar la resistencia para disipar la energía absorbida por la máquina durante el frenado, evitando así sobrecalentamientos del dispositivo que pongan en riesgo al sistema.

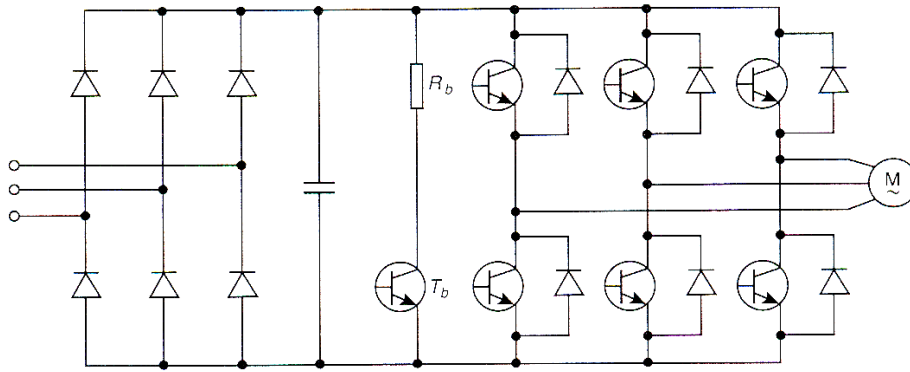


Figura 16 Esquema básico de un variador de frecuencia

El variador incluye una serie de protecciones para el motor, tales como fallo contra sobreintensidad, sobretensión, fallo contra desequilibrios entre fases, defectos a tierra, etc.. Además de ofrecer procesos de arranque y frenados suaves mediante rampas de aceleración y de frenado, lo que redundará en un aumento de la vida del motor y las instalaciones. El uso de variadores de frecuencia añade un enorme potencial para el ahorro de energía, disminuyendo la velocidad del motor en muchas aplicaciones, y evitando el uso de reductores mecánicos. Entre sus principales ventajas destacan las siguientes:

- Mejora el proceso de control y, por tanto, la calidad del producto.
- Posibilidad de programar un arranque suave, parada y freno (funciones de arrancador progresivo).
- Amplio rango de velocidad, par y potencia. (velocidades continuas y discretas).
- Control de varios motores.
- Factor de potencia unitario.
- Respuesta dinámica comparable con los drivers de DC.
- Capacidad de by-pass ante fallos del variador.
- Protección integrada del motor.

Para la obtención de una tensión de salida de amplitud y frecuencia variable es necesario utilizar la modulación por ancho de pulso (PWM). Este método utiliza una gran cantidad de pulsos por semiciclo, del orden de kilohertzios, la anchura de los pulsos se hace variar mediante una onda senoidal de referencia, de tal modo que los pulsos más próximos al pico de la onda son más anchos y los más cercanos al paso por cero son más estrechos.

Para conseguir este tipo de modulación, la onda de referencia debe de ser senoidal y de frecuencia igual a la que debe producir el inversor, siendo la onda portadora una serie de ondas triangulares. Los puntos de intersección de las dos ondas señalan los instantes de conexión y desconexión de los interruptores estáticos (IGBT).

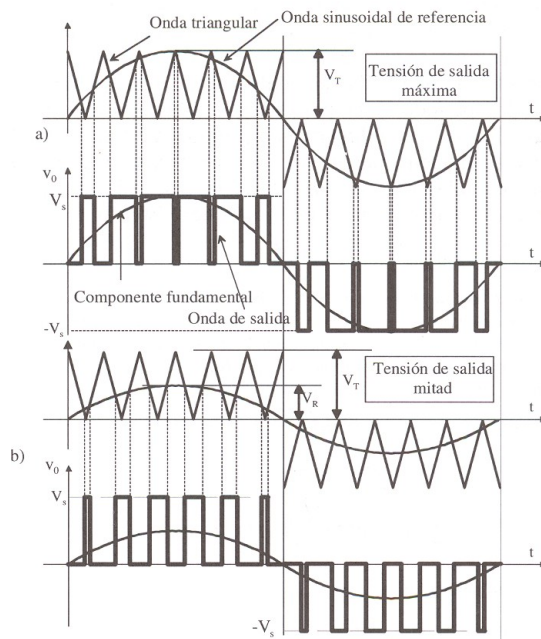


Figura 17 Modulación por ancho de pulso (PWM)

3.1.2 Solución comercial empleada

Dispondremos de dos tipos distintos de variadores de frecuencia en la instalación. El primero de ellos se encargará de la alimentación del motor asíncrono, el cual siempre actuará en el modo motor, mientras que el segundo controlará el servo de la instalación, regulando el par resistente generado cuando actúa como generador eléctrico y la velocidad o el par desarrollado cuando funciona en modo motor en solitario.

3.1.2.1 Variador de frecuencia UNI 1404

Variador de frecuencia de *Control Techniques*, modelo UNI 1404, con una potencia nominal de 4 CV y una intensidad máxima de 8,8 A. Las frecuencias de conmutación a las que el variador puede actuar son 3,4,5,6,9 o 12 kHz. Su funcionalidad en el sistema se limitará a la regulación de la velocidad de sincronismo del motor asíncrono, controlando dicho parámetro directamente desde el teclado de este.

El variador se compone de:

- **Display** dividido a su vez en dos partes, una superior y otra inferior. En el modo de programación, el display superior muestra el parámetro; y la inferior muestra el valor del parámetro. En el modo de funcionamiento, el display superior muestra la velocidad del motor o la causa del fallo en el funcionamiento, mientras que la inferior muestra el estado (según la visualización del parámetro 0.10: RUN= motor activado, STOP= motor parado,

TRIP= Variador en fallo)

- **Teclas (Pad)**, divididas en teclas de control para controlar la actuación del motor como el arranque, la parada y el cambio de sentido de giro; y teclas de programación que sirven para desplazarse por los menús y para cambiar los parámetros de cada parámetro.
- **Bornes de control**: Estos bornes sirven para controlar el variador mediante entradas externas, que pueden ser accionadas mediante un PLC, un contactor o mediante los pulsadores del 1 al 31.
- **Bornes de Potencia**: En estos bornes se realizan las conexiones entre el variador y la red, el motor o la resistencia de frenado (si existe).



Figura 18. Variador de frecuencia UNI1404

3.1.2.2 Variador de frecuencia SP1401

Variador de frecuencia de Control Techniques, modelo SP1401, con una potencia nominal de 5 CV y una intensidad máxima de 6,9 A. Las frecuencias de conmutación a las que el variador puede actuar son 3,4,6,8,12 o 16 kHz.



Figura 19. Variador de frecuencia SP1401

El variador constará de las siguientes partes:

- **Teclado** extraíble (Figura 20), el cual se compondrá:
 - Joypad: permite desplazarse por la estructura de parámetros y cambiar los valores de los parámetros.
 - Botón de modo: permite alternar entre los modos de visualización, la visualización de parámetros, la edición de parámetros y el estado.
 - Tres botones de control: si el modo de teclado se encuentra seleccionado, permiten controlar el accionamiento.

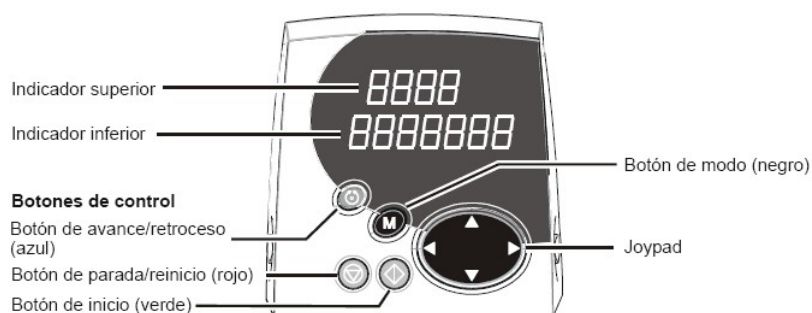


Figura 20. Teclado extraíble del variador de frecuencia

- **Bornes de control:** Estos bornes sirven para controlar el variador mediante entradas externas, que pueden ser accionadas mediante un PLC, un contactor o mediante pulsadores.
- **Bornes de Potencia:** En estos bornes se realizan las conexiones entre el variador y la red, el motor o la resistencia de frenado (si existe). A diferencia del anterior variador de frecuencia, tendremos que acceder a la parte posterior del variador de frecuencia para el montaje de la resistencia de frenado.
- **Ranuras para módulos:** El variador posee en su parte frontal de tres ranuras para el montaje de módulos que aumentarán la funcionalidad del mismo.

Para adecuar su funcionamiento a las necesidades del usuario el variador dispone de parámetros configurables. Estos parámetros se encuentran en grupos llamados *menús*, y a su vez cada menú comprende los parámetros dedicados a una parte del funcionamiento del variador. Cada parámetro tiene asignado una dirección dentro del variador, esta dirección esta compuesta por dos parejas de dígitos separados por un punto, la primera pareja indica el número del menú, y la segunda pareja indica la posición del parámetro dentro del menú, es decir, el parámetro 05.16 se encuentra en el menú 5 y es el parámetro 16. En el menú 0 se recogen los parámetros más importantes del resto de menús, siendo los más importantes para la configuración los que

encontramos en la Tabla 1

Parámetros	Funciones
0.00	Configuración y almacenamiento
0.01 ~ 0.02	Límites de velocidad
0.03 ~ 0.06	Rampas, selección de referencia de velocidad y límite de intensidad
0.10	Indicación de velocidad
0.14 ~ 0.15	Selector de modo de par y rampa
0.23 ~ 0.27	Referencias prefijadas
0.34	Código de seguridad del usuario
0.40	Autoajuste
0.41	Frecuencia de conmutación PWM
0.42 ~ 0.47	Parámetros del motor
0.48	Selección de modo de funcionamiento
0.49 ~ 0.50	Información de seguridad y versión de software

Tabla 1. Parámetros del variador de frecuencia SP1401

Podemos observar en la Figura 21 cómo se realiza el desplazamiento por los distintos menús y parámetros del variador de frecuencia, gracias al uso del *joypad*.

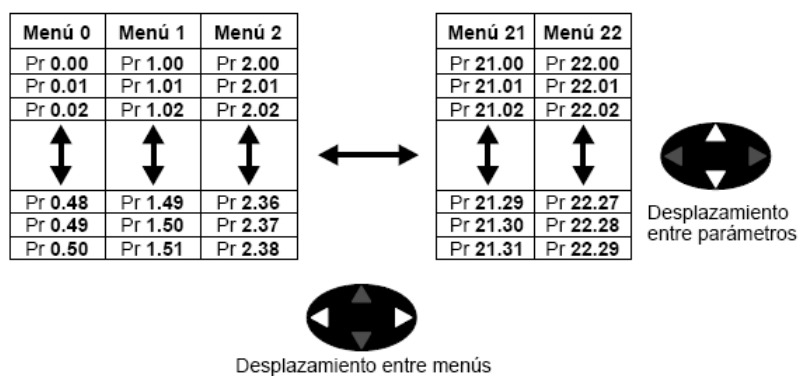


Figura 21. Desplazamiento entre menús y parámetros del variador de frecuencia SP1401

Sin embargo, para un control externo del variador de frecuencia SP1401, es necesario el uso de módulos de comunicaciones, lo cual posibilitará al usuario la configuración de parámetros sin tener que acceder a través del teclado del variador de frecuencia e, incluso, el desarrollo de aplicaciones automatizadas. En concreto, si dicha comunicación se realiza mediante un cableado de tipo RJ45, el módulo a conectar al variador de frecuencia es el tipo SM-Ethernet, explicado en el apartado a continuación.

3.2 Módulo SM-Ethernet

La función del módulo SM-Ethernet se basa en posibilitar la comunicación entre el PC y el variador de frecuencia, el cual se acopla sobre el propio variador. El módulo funciona de servidor web local, almacenando los valores de los distintos parámetros del variador de frecuencia y actualizándolos en ciclos temporizados.

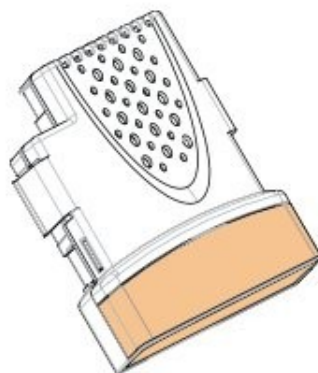


Figura 22. Módulo SM-Ethernet

La colocación en el variador se realizará en una de las ranuras disponibles del modelo SP1401, como se indica en la Figura 23. Dependiendo en qué ranura coloquemos el módulo, variará el número del menú, de forma que si es en la ranura 1, 2 o 3, el menú donde localizaremos los distintos parámetros serán el 15, 16 o 17 respectivamente.

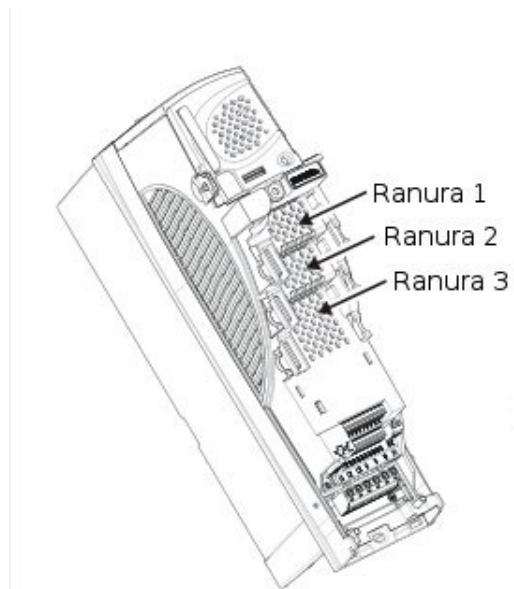


Figura 23. Ranuras para módulos de aplicaciones

Los parámetros que regulan el módulo controlan fundamentalmente las características de las comunicaciones entre dicho módulo y el PC a través del cable Ethernet. Algunos de ellos serán únicamente de lectura, mientras que otros podrán ser modificados, como los indicados en la Tabla 2.

Parámetros	Funciones
16.10	Velocidad de transmisión de datos
16.06	Diagnóstico de errores
16.10 ~ 16.13	Dirección IP
16.14 ~ 16.17	Máscara de Subred
16.18 ~ 16.21	Puerta de Enlace
16.30	Cargar opciones por defecto
16.31	Salvar parámetros de las opciones
16.32	Reinicializar
16.33	Restaurar parámetros de las opciones
16.35	Número de serie (Dirección MAC parcial)
16.44	Velocidad actual de transmisión de datos

Tabla 2. Parámetros del módulo SM-Ethernet

3.3 Servo

3.3.1 Principio de funcionamiento

Partiendo, como ya hemos comentado de dispositivos presentes en procesos industriales actuales, hemos escogido el servo como encargado de crear un par resistente para oponerse al par motor. Su constitución es similar a la de un generador síncrono con imanes permanentes, el inducido esta situado en el estator y compuesto por tres bobinados desfasados 120° , mientras el inductor, situado en el rotor, está compuesto por imanes permanentes. La alimentación del servo se realiza mediante la conmutación de los distintos transistores IGBT (Figura 24), estableciendo así un campo magnético giratorio.

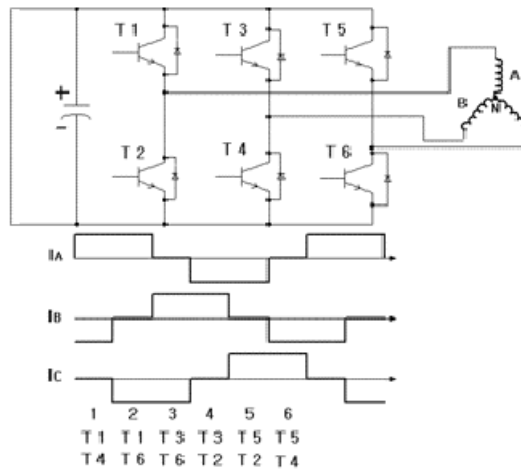


Figura 24. Esquema de conexión y secuencia de conmutación de diodos

Por su equivalencia con la máquina de corriente continua, estudiada por J.R. Hendershot en su libro *Desing of brushless permanet-magnet motors* y si partimos de la hipótesis que trabajamos en la parte lineal de la curva B-H, se establecen las ecuaciones que rigen el comportamiento de la máquina brushless. Estas ecuaciones son:

$$T \cdot \Omega = E \cdot I$$

Donde T es el par, Ω es la velocidad de giro del motor, E es la fuerza electro motriz, e I es la intensidad. Las ecuaciones que relacionan la fuerza electromotriz y la intensidad con el par y la velocidad de giro del motor son:

$$E = K_e \cdot \Omega \quad T = K_T \cdot I$$

Donde K_e y K_T son constantes características del servo. Resolviendo este sistema de ecuaciones se obtiene que el par es inversamente proporcional a la velocidad de giro, para una

tensión de alimentación constatare como se representa en la Figura 25.

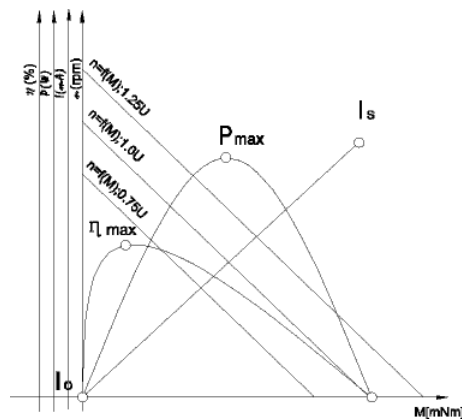


Figura 25. Curvas características de un servo

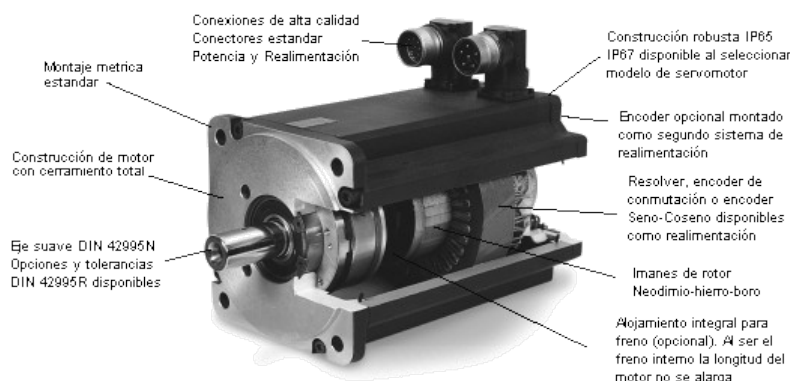


Figura 26. Servo

3.3.2 Solución comercial empleada

La solución comercial empleada es un motor trifásico sin escobillas, con rotor de imanes permanentes de metales alcalinos raros. La denominación comercial es **95UMB300CACAA**. A continuación se describe el significado de cada una de las partes que componen su denominación:

95: dimensión del chasis

UM: tipo de motor, en este caso para un accionamiento *Unidrive*.

B: Longitud del estator.

30: Velocidad nominal del motor; en nuestro caso 3000 rpm

0: Freno no disponible.

C: la conexión del motor se realiza mediante un *conector híbrido*.

A: conexión del eje, en este caso con llave.

C: el motor incorpora un encoder incremental.

A: la montura de la brida es del tipo IEC.

A: la inercia del motor: normal.

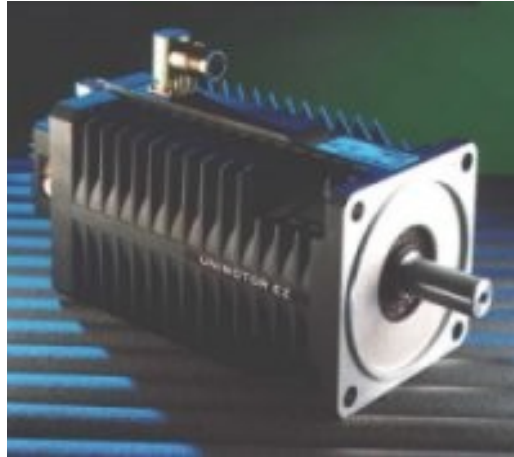


Figura 27. Servo-motor

El servo consta de 6 polos, y es capaz de proporcionar un par de forma continua de 4,3 Nm, consumiendo una intensidad de 3,1 Arms. La velocidad máxima es de 3000 rpm. La constante de par (K_t) es de 1,6 Nm/Arms. En la Figura 28 se muestra la relación Par/velocidad del servo, de forma continua (la zona verde oscura), y de forma discontinua (zona verde claro).

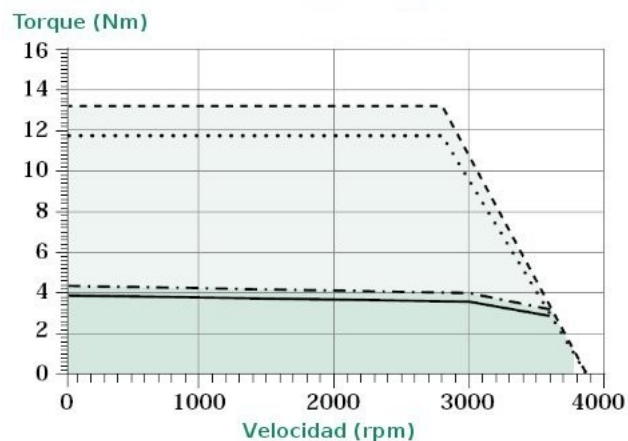


Figura 28. Relación Par velocidad del servo motor

El servo es alimentado por el variador con ondas sinusoidales obtenidas mediante la modulación por ancho de pulsos (PWM). El encoder incremental que incorpora el servo motor para la realimentación de velocidad es de 4096 pulsos por revolución, y consta de una conexión independiente, descrita más adelante.

3.3.2.1 Encoder Incremental

El servo lleva incorporado un encoder incremental encargado de medir la velocidad, siendo éste uno de los tipos de encoder más usados en la industria. Su principio de funcionamiento se basa en generar pulsos al girar el eje, empleando este número de impulsos por vuelta para determinar la

velocidad del eje, la posición de dicho eje o la medida de una longitud.

Se pueden diferenciar en:

- Unidireccionales: Se utilizan en aplicaciones donde no es necesario detectar el sentido de rotación.
- Bidireccionales: Se utilizan en aplicaciones donde es necesario discriminar el sentido de giro.

Cada encoder incremental dispone en su interior de un disco marcado con una serie de líneas uniformes a través de una única pista alrededor de su perímetro, con las líneas opacas a la luz y líneas transparentes de igual anchura que las opacas. También existe una unidad emisora de luz y una unidad de captación de la misma luz, por lo que al girar el disco se producen periodos en los que las líneas opacas interrumpen el haz de luz que llega a la unidad captadora. Así, dicha unidad genera una serie de pulsos correspondientes a las salidas del encoder.



Figura 29. Disco interior de un encoder incremental

3.4 Switch o conmutador de red

3.4.1 Principio de funcionamiento

El switch o conmutador de red es un dispositivo electrónico de interconexión de redes de ordenadores que opera en la capa 2 (nivel de enlace de datos) del modelo OSI (*Open Systems Interconnection*). Un conmutador interconecta dos o más segmentos de la red, pasando datos de un segmento a otro de la misma, por medio de la localización de dicho segmento a través de su dirección MAC.

Su función es la de conectar múltiples redes, fusionándolas en una sola. Los conmutadores poseen la capacidad de aprender y almacenar las direcciones de red de nivel 2 (direcciones MAC) de los dispositivos alcanzables a través de cada uno de sus puertos. Por ejemplo, un equipo conectado directamente a un puerto de un conmutador provoca que el conmutador almacene su dirección MAC. Esto permite que la información dirigida a un dispositivo vaya desde el puerto

origen al puerto de destino.

3.4.2 Solución comercial empleada

La solución comercial empleada es un switch de la marca SMC, modelo EZ6508TX,(Figura 30), con un total de 8 puertos Ethernet 10Base-T, también funcionales como 100Base_TX, con una velocidad de transferencia de datos de 100 Mbps y protocolo de interconexión de datos Ethernet o Fast Ethernet. La interconexión entre dispositivos se hace mediante cableado y su modo es mediante Duplex pleno. Posee indicadores de estado, como son de actividad de enlace, velocidad de transmisión del puerto y alimentación, a través de un panel LED. Posee un tamaño de tabla de direcciones MAC de 2000 entradas.



Figura 30. Conmutador de red SMC modelo EZ6508TX

Debido a su función Plug&Play, no es necesario el uso de drivers para su instalación y manejo. La fuente de alimentación se hará a través de un transformador alimentado de red y con una salida de 6 voltios, siendo el consumo eléctrico en funcionamiento entorno a 3,8 y 4,6 vatios.

4 Configuración y conexionado del equipo

Para el correcto funcionamiento del sistema es necesario realizar la interconexión entre los diferentes equipos, distinguiendo entre ellas las conexiones de potencia para la alimentación del variador y del servo, y las de control que se encargan de comunicar los distintos equipos y de proporcionar las señales digitales que habilitan el funcionamiento por parte del usuario. Pasamos a analizar por separado cada uno de ellas.

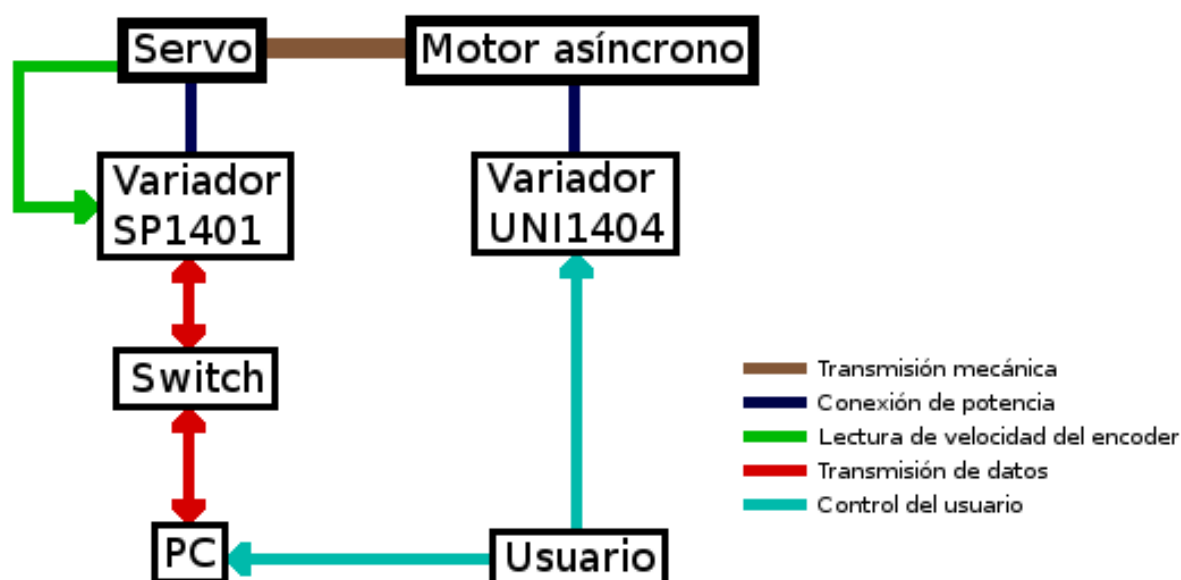


Figura 31. Esquema general de conexionado

4.1 Conexiones de control

Se pueden diferenciar entre dos tipos de conexión de control: en el primer grupo se engloban las conexiones digitales (0 ó 1) que son necesarias para habilitar el funcionamiento del variador y del servo; están situadas en el propio variador, tal y como hemos comentado, a través del teclado, y es necesario activarlas manualmente antes de probar el montaje. El segundo grupo abarca las conexiones de comunicación entre los distintos elementos, que son las encargadas de transmitir datos entre equipos, descritas a continuación.

4.1.1 Interconexión Variador-Servo

Es la encargada de transmitir la velocidad del eje común entre el motor y el servo (leída por el encoder incremental) y transmitirla al variador de frecuencia. El cable de conexión consiste en un terminal hembra de 17 vías (Figura 32) para la conexión en el extremo del servo y un conector tipo D macho de 15 terminales para la conexión al variador de frecuencia. La relación de conexiones puede visualizarse en la Tabla 3



Figura 32. Cable de 17 vías

Para el correcto funcionamiento del sistema es necesario realizar la interconexión entre los diferentes equipos, distinguiendo entre ellas las conexiones de potencia para la alimentación del variador y del servo, y las de control que se encargan de comunicar los distintos equipos y de proporcionar las señales digitales que habilitan el funcionamiento por parte del usuario.



Conector del variador	Nº en el conector del variador	Función	Nº en el conector del motor	Conector del motor
	1	Canal A	10	
	2	Canal A Inverso	13	
	3	Canal B	14	
	4	Canal B Inverso	15	
	5	Canal Z	11	
	6	Canal Z Inverso	12	
	7	S1	4	
	8	S1 Inverso	5	
	9	S2	6	
	10	S2 Inverso	7	
	11	S3	8	
	12	S3 Inverso	9	
	13	+ 5 VDC	16	
	14	0 V	17	
	15	No usado	3	
		Termistor +	1	
		Termistor -	2	

Tabla 3. Conexiones del cable de conexión Variador-Máquina Síncrona

4.1.2 Interconexión variador-PC

Para la conexión entre el PC y el módulo, se presentan dos opciones: La utilización de un cable RJ45 cruzado, con el cual podremos conectar directamente ambas partes o por medio de dos cables RJ45 no cruzados y un switch o conmutador de red. En este proyecto la opción elegida ha sido la segunda, debido a la posibilidad de ampliación de la instalación con otros módulos SM-Ethernet que puedan conectarse directamente al switch en aquellos puertos que están libres como podemos ver en la Figura 33

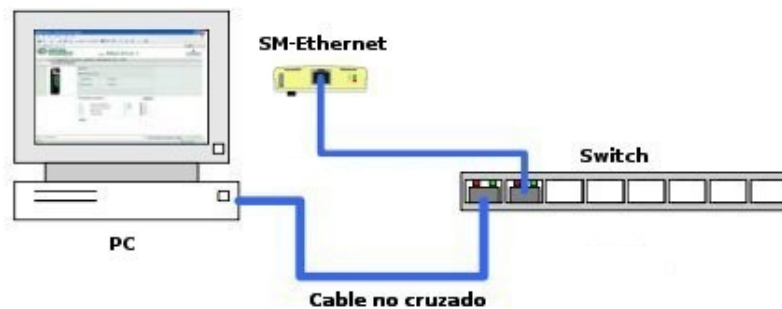


Figura 33. Conexión PC-Módulo SM-Ethernet

En la parte frontal del módulo podemos visualizar las siguientes conexiones y LEDs que nos indicarán el estado del módulo (Figura 34). En caso de un correcto funcionamiento del sistema, el color de los LEDs será amarillo, mientras que si surge un problema de conexión con el módulo, el color de los LEDs de estado (*Module status*) y acceso al servidor (*Flash access*) será rojo. Los terminales RJ45 estarán conectados en función del tipo de cable utilizado, descritos en la Tabla 4.

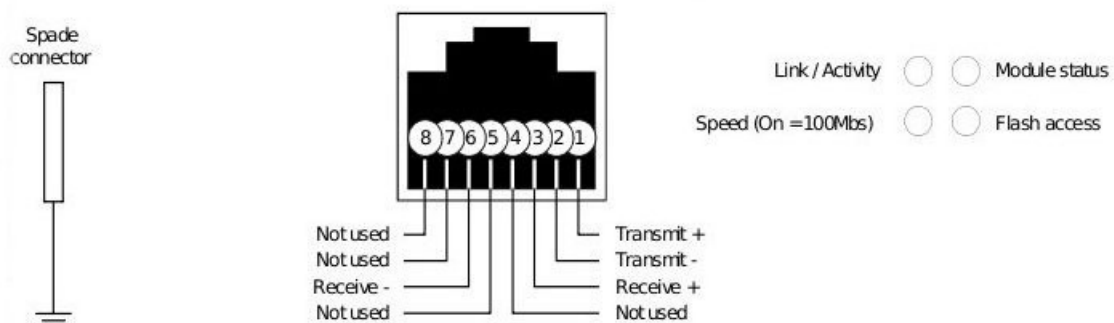


Figura 34. Conexión Ethernet y LEDs de estado del módulo SM-Ethernet

Terminales	Utilización de cable no cruzado	Utilización de cable cruzado
1	Transmite +Ve	Recibe +Ve
2	Transmite -Ve	Recibe -Ve
3	Recibe +Ve	Transmite +Ve
4	-	-
5	-	-
6	Recibe -Ve	Transmite -Ve
7	-	-
8	-	-

Tabla 4. Conexión de terminales RJ45

La configuración necesaria para un correcto funcionamiento de dicha conexión esta basada en los protocolos que rigen la comunicación TCP-IP. Para ello tendremos en cuenta los siguientes conceptos:

- Dirección IP: Código consistente en cuatro numeros de 8 bits cada uno (octetos) representados de la siguiente manera:

w,x,y,z como por ejemplo 192.168.0.1

- Máscara de subred: Código numérico que forma parte de la dirección de los Pcs, con el mismo formato que la dirección IP, y que afecta a un segmento particular de la red. Se utiliza para dividir grandes redes en redes menores, facilitando la administración y reduciendo el tráfico inútil, de tal manera que será la misma para ordenadores de una misma subred.

a,b,c,d como por ejemplo 255.255.255.0

- Ruta predeterminada: Al igual que la máscara de subred, la configuración debe ser igual que la utilizada por la red en que nos encontremos, es decir, tener el mismo código que el PC.

La dirección IP debe ser única, es por ello que la configuración de la red será local, sin una conexión directa a internet. Tanto la dirección IP y como la máscara de subred y la ruta predeterminada del módulo SM-Ethernet están almacenados en los parametros del variador de frecuencia, pudiendose modificar. Si definimos la dirección IP como w.x.y.z, la máscara de subred

como a.b.c.d y la ruta predeterminada como e.f.g.h, la distribución será la que figura en la Tabla 5.

Octeto	Parametro del variador	Octeto	Parametro del variador
w	16.10	c	16.16
x	16.11	d	16.17
y	16.12	e	16.18
z	16.13	f	16.19
a	16.14	g	16.20
b	16.15	h	16.21

Tabla 5. Relación de octetos de configuración IP con parámetro del módulo SM-Ethernet

En el caso de la máscara de subred, utilizaremos el código 255.255.255.0, mientras que para conocer la ruta predeterminada del PC, en caso de trabajar sobre el sistema operativo windows, pulsaremos el boton Inicio y a continuación en la opción de Ejecutar, introducimos el comando *cmd*, el cual da acceso a la consola de Ms-DOS y escribimos el comando *ipconfig* (Figura 35), tras lo cual el sistema nos mostrará la información necesaria para una correcta configuración.

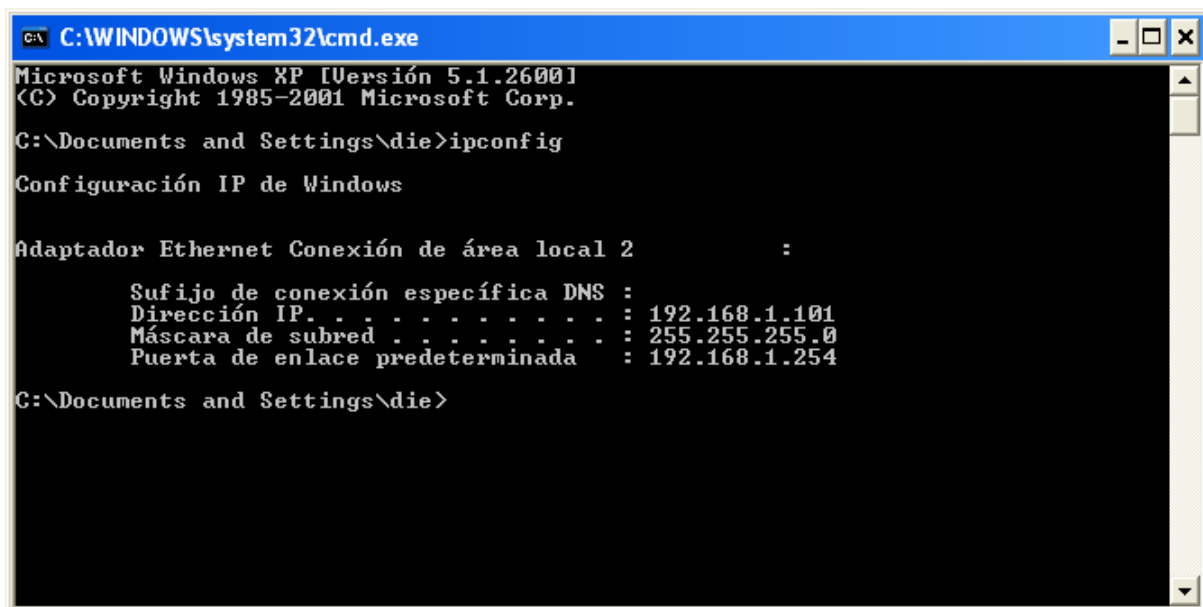


Figura 35. Información de conexión en sistema operativo Windows (MsDOS)

Si el sistema operativo en que nos encontramos está basado en linux, la información de conexión se obtendrá tras pulsar con el boton derecho del ratón sobre el icono de la conexión de red y a continuación elegir la opción de Características de red (Figura 36)

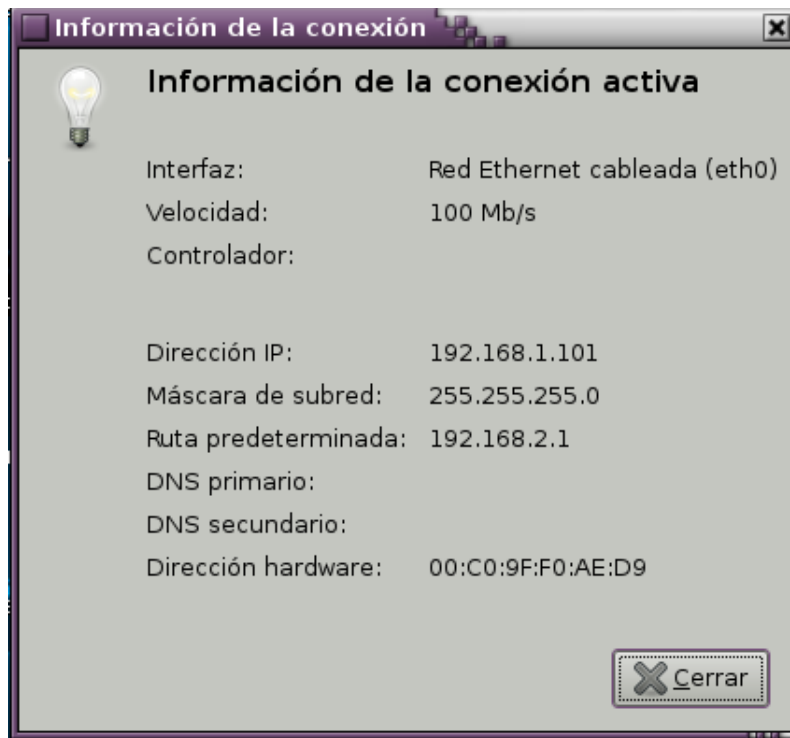


Figura 36. Información de conexión en sistema operativo Linux (Ubuntu 7.10)

El módulo SM-Ethernet almacena en memoria un servidor web, el cual posee por defecto paginas web que posibilitan el control del variador de frecuencia a través de navegadores de internet instalados en el PC. Entre las posibilidades que presenta dicho servidor, encontramos la de permitir al usuario la modificación de una de las subcarpetas del servidor web, mediante la creación de nuevas páginas web, con la ayuda de un programa cliente FTP.

4.1.3 Conexiones de control del variador de frecuencia (Motor)

Para que el variador pueda funcionar es necesario realizar las conexiones de control correspondientes. En nuestro caso al utilizar el variador en modo teclado no es necesario cablear los contactos correspondientes del sentido giro del motor, ya que se controla desde teclado, ni las entradas analógicas de referencia de velocidad. Sólo es necesario cablear los contactos correspondientes al termistor del motor y a la desconexión externa, señalados en la Figura 37

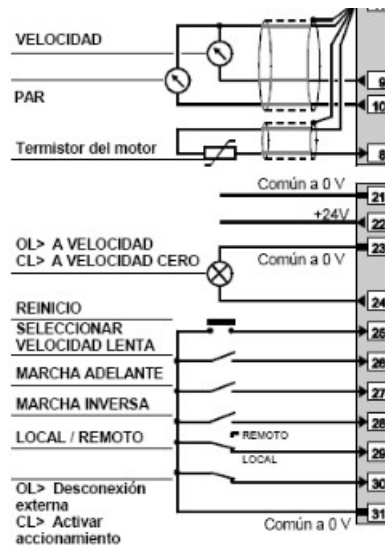


Figura 37. Conexiones de control UNI1404

4.1.4 Conexiones de control del variador de frecuencia (Servo)

Al igual que en el caso anterior, las conexiones de terminales del variador de frecuencia (Figura 38) se cablearán para la habilitación del propio servo y el termistor del motor. Como también se utilizará el servo en modo motor, se habilitará el contacto de marcha adelante.

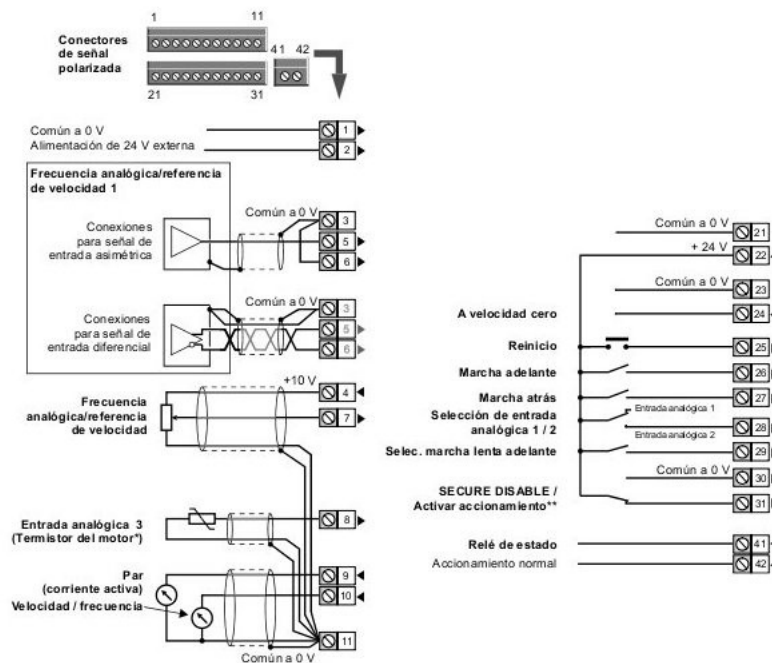


Figura 38. Conexiones de control SP1401

4.2 Conexiones de potencia.

4.2.1 Alimentación del motor asíncrono

Para la alimentación del motor asíncrono se realizará una conexión en triángulo entre los distintos terminales. Ya que se va a realizar un control de la velocidad de la máquina asíncrona, la línea trifásica provendrá, como ya se indicó, del variador de frecuencia UNI 1404, gracias a la variación de la frecuencia de la señal PWM, mediante la conexión a los terminales U, V y W.

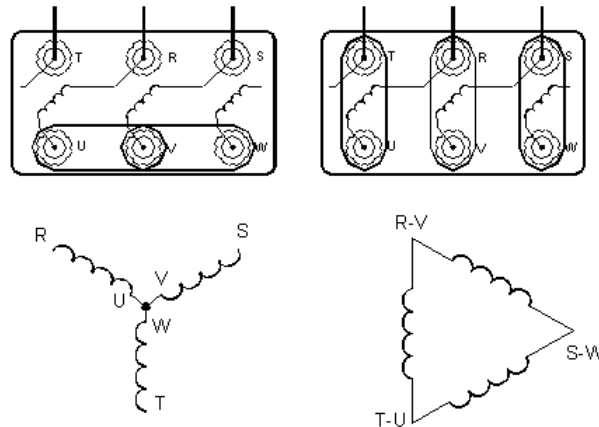


Figura 39. Configuración de conexión en estrella y en triángulo

A su vez, el variador de frecuencia será alimentado por su conexión directa con la red trifásica por medio de L1, L2 y L3, estando los terminales del variador configurados según se indica en la Figura 40

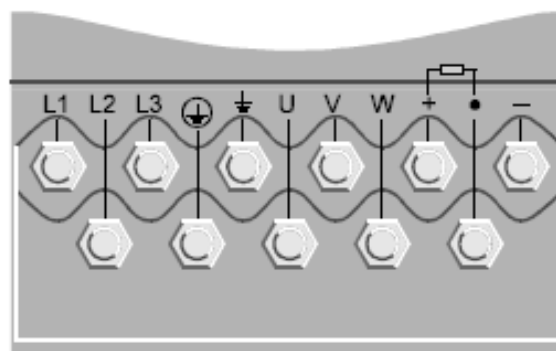


Figura 40. Terminales de potencia del variador de frecuencia UNI1404

4.2.2 Alimentación del servo

Las conexiones que se realizarán son las encargadas de transmitir la energía desde el variador hasta el servo, constituido por un conector hembra con seis terminales para la conexión a la máquina síncrona (Tabla 6), y en cuyo extremo quedan los cables con terminales aislados para la conexión a las bornas de potencia del variador: U ,V y W.

Pin	Función
1	Fase U
2	Fase V
3	Tierra
4	Fase W
5	-
6	-

Tabla 6. Cable de Potencia de la máquina síncrona (lado servo)

La configuración de conexión del variador de frecuencia SP1401, quedará según se indica en la Figura 41, donde podemos observar un terminal que se encargará de la toma de tierra para la protección eléctrica del sistema.

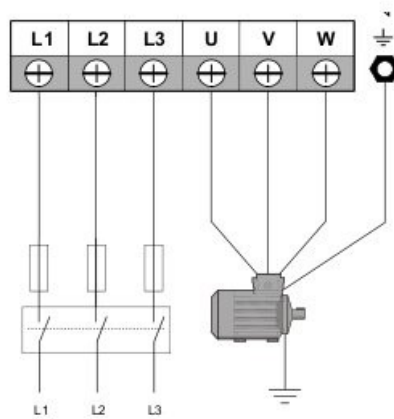


Figura 41. Conexiones de potencia del variador de frecuencia SP1401

4.2.3 Evacuación de Energía

Cuando el servo ejerce un par resistente que se opone al movimiento del motor asíncrono, se

produce una transformación de la energía mecánica del eje en energía eléctrica, es decir, el servo actúa como un generador síncrono, haciéndose necesario evacuar dicha energía. El variador de frecuencia está diseñado para evacuar pequeñas cantidades de energía, pero cuando la cantidad de energía es mayor, es necesario disponer de medios externos al propio variador para evacuar la energía restante, ya que de no hacerlo se produciría una desconexión del variador debido a la actuación de las protecciones térmicas.

Para permitir el correcto funcionamiento de equipo y evacuar esta energía se disponen de los siguientes métodos: Disipar energía con ayuda de una resistencia de frenado o devolver la energía a la red.

4.2.3.1 Resistencia de frenado

Este método consiste en conectar una resistencia en el bus de corriente continua del variador, entre la resistencia de frenado y el terminal negativo del bus de continua se sitúa internamente un transistor de potencia (IGBT). Este transistor es el encargado de conectar y desconectar la resistencia de modo periódico para la disipación de la energía.

Todo el proceso es monitorizado por el propio variador, modificando los tiempos de conexión y desconexión de la resistencia en función de la energía a disipar, siendo los tiempos de conexión prefijados y los de desconexión más cortos cuanto más energía es necesario evacuar.

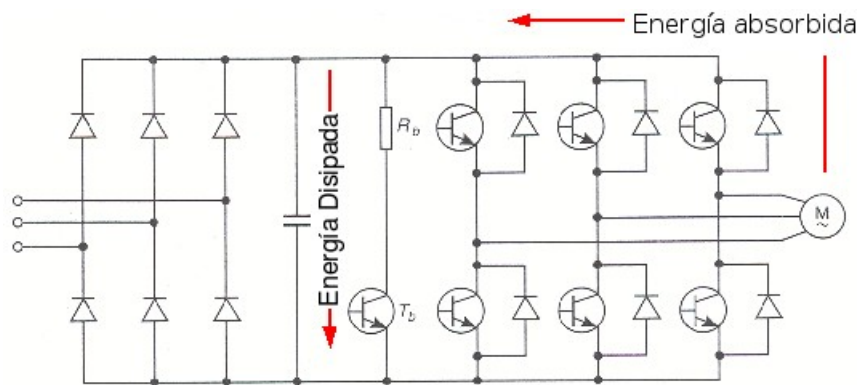


Figura 42. Diagrama del sistema resistencia de frenado

Para la instalación (Figura 43), haremos uso de un potenciómetro (elemento A, Figura 44) de 1 kW de potencia y hasta 100 Ω de resistencia eléctrica y filtro CEM (*compatibilidad electromagnética*, elemento B) con el fin de asegurar que el accionamiento funcione con fiabilidad y minimizar el riesgo de interferir en los equipos cercanos. La resistencia eléctrica se configurará para un valor de 64,5 Ω , ya que dicho valor es el utilizado por la resistencia que utiliza Control Techniques para el variador de frecuencia SP1401.

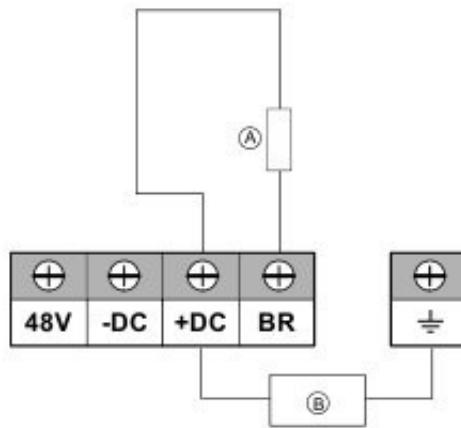


Figura 43. Conexión de la resistencia de frenado



Figura 44. Potenciómetro

4.2.3.2 Devolución de energía a la red

Consiste en transformar la energía de salida de la etapa de continua, disipada normalmente en la resistencia de frenado, para devolverla a la red de alimentación, de forma que esta energía no se pierde en forma de calor, sino que puede ser reutilizada por otros dispositivos eléctricos. Este sistema es más complejo que el anterior ya que requiere el uso de dos variadores, uno para el accionamiento del motor y otro conectado a la red, y diversos accesorios encargados de acondicionar la onda de tensión de salida del equipo (filtros, contactores, varistores, etc). En el caso del segundo variador, funcionando en modo regenerativo, el puente IGBT se utiliza como rectificador sinusoidal, encargándose este puente de transformar el voltaje del bus de C.C. en un voltaje sinusoidal de frecuencia y tensión a igual a la de la red.

La unidad regeneradora produce un voltaje PWM cuya tensión fundamental es igual a la de red, la diferencia entre esta dos tensiones es una tensión de alta frecuencia que es atrapada por el filtro inductivo, consiguiendo así una onda de tensión más pura, con menos armónicos. Para realizar la regeneración de energía a la red existen dos tipos de montaje dependiendo de las necesidades.

4.2.3.2.1 Montaje Standard:

Esta configuración es la más usual, en ella se asocian el variador del motor y el regenerador mediante el bus de continua, estos dos variadores son de igual tamaño ya que por ambos debe circular la misma potencia. En el variador regenerador la energía puede fluir tanto de la red a la máquina eléctrica como en sentido inverso, actuando dicha máquina como motor o como generador.

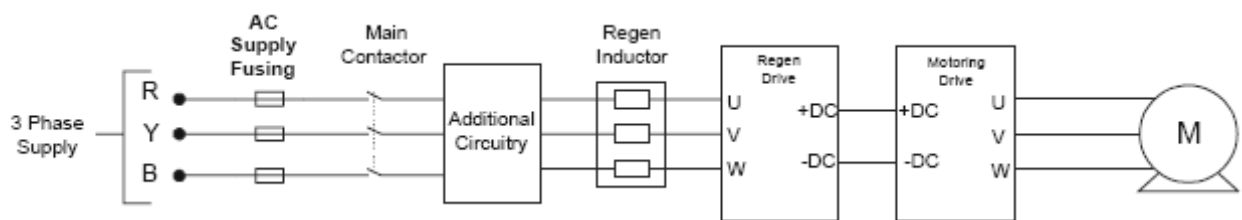


Figura 45 Configuración Standard del sistema de regeneración

4.2.3.2.2 Sustitución de la resistencia de frenado:

Este sistema se utiliza cuando la potencia a regenerar durante el frenado es mucho menor que la del motor, por lo que el variador encargado de la regeneración es de menor tamaño que el variador del motor, lo que supone un ahorro económico importante. En este caso, por el variador regenerador sólo fluye la energía de frenado del motor, mientras que el accionamiento funciona como motor la energía pasa a través del variador del motor.

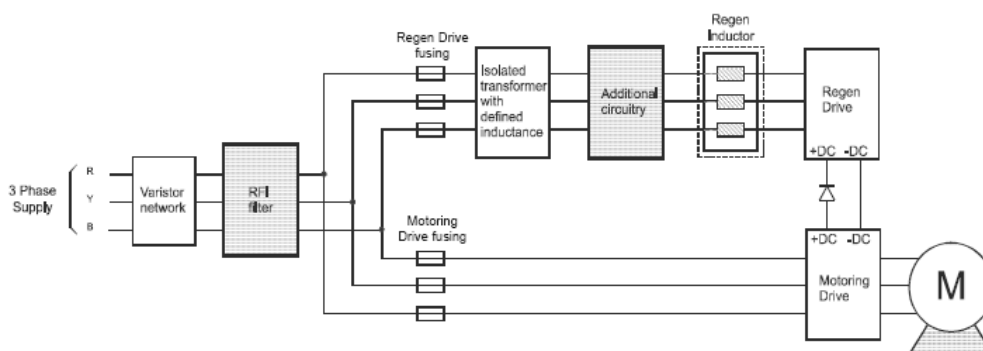


Figura 46. Configuración de sustitución de la resistencia de frenado del sistema de regeneración

4.2.3.2.3 Regeneración Múltiple

Este sistema es una variante de la configuración Standard, en él se combinan una serie de variador-motor de pequeña potencia, todos ellos unidos por el bus de continua y a su vez unidos a un variador regenerador de mayor potencia. La energía fluye de los motores a su respectivo variador, y de ahí al variador regenerador.

El principal inconveniente es que el variador regenerador debe ser de una potencia nominal suficiente para soportar toda la potencia del resto de variadores regeneradores lo cual encarece su precio. Por el contrario con este sistema de montaje es posible agrupar un gran número de motores. Este sistema es el más indicado para su instalación en laboratorios con un número alto de pequeños motores.

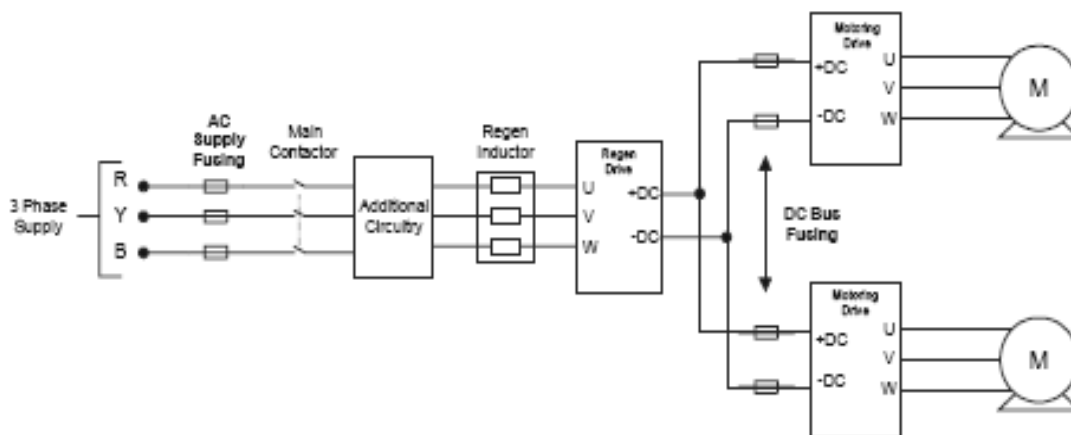


Figura 47. Configuración Regeneración Múltiple

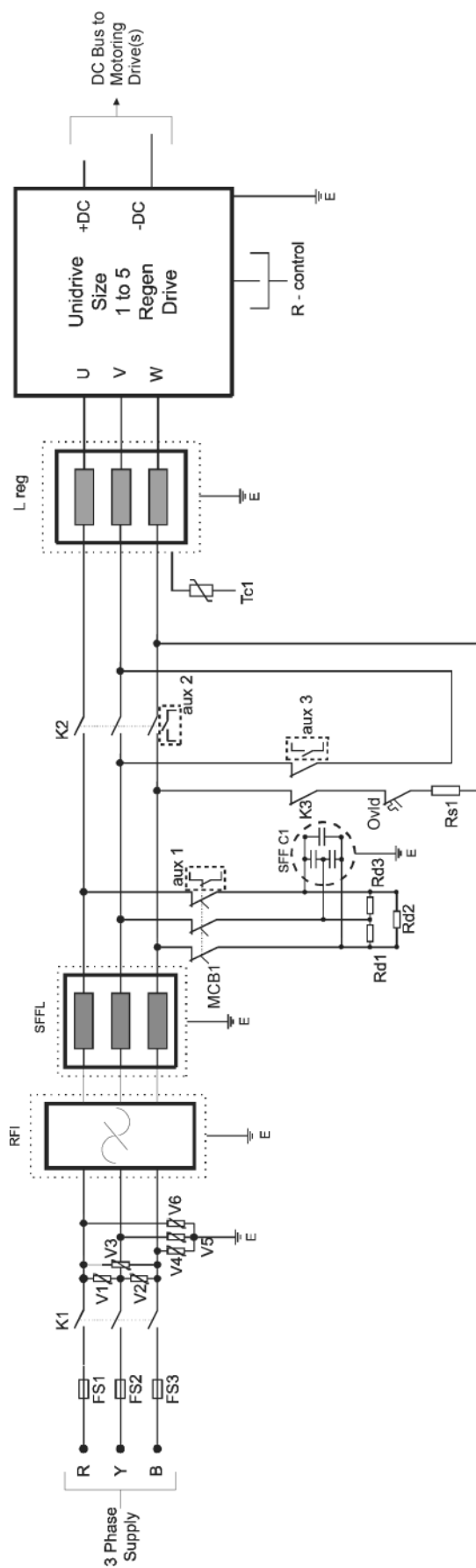


Figura 48 Configuración Standard. Conexión de potencia

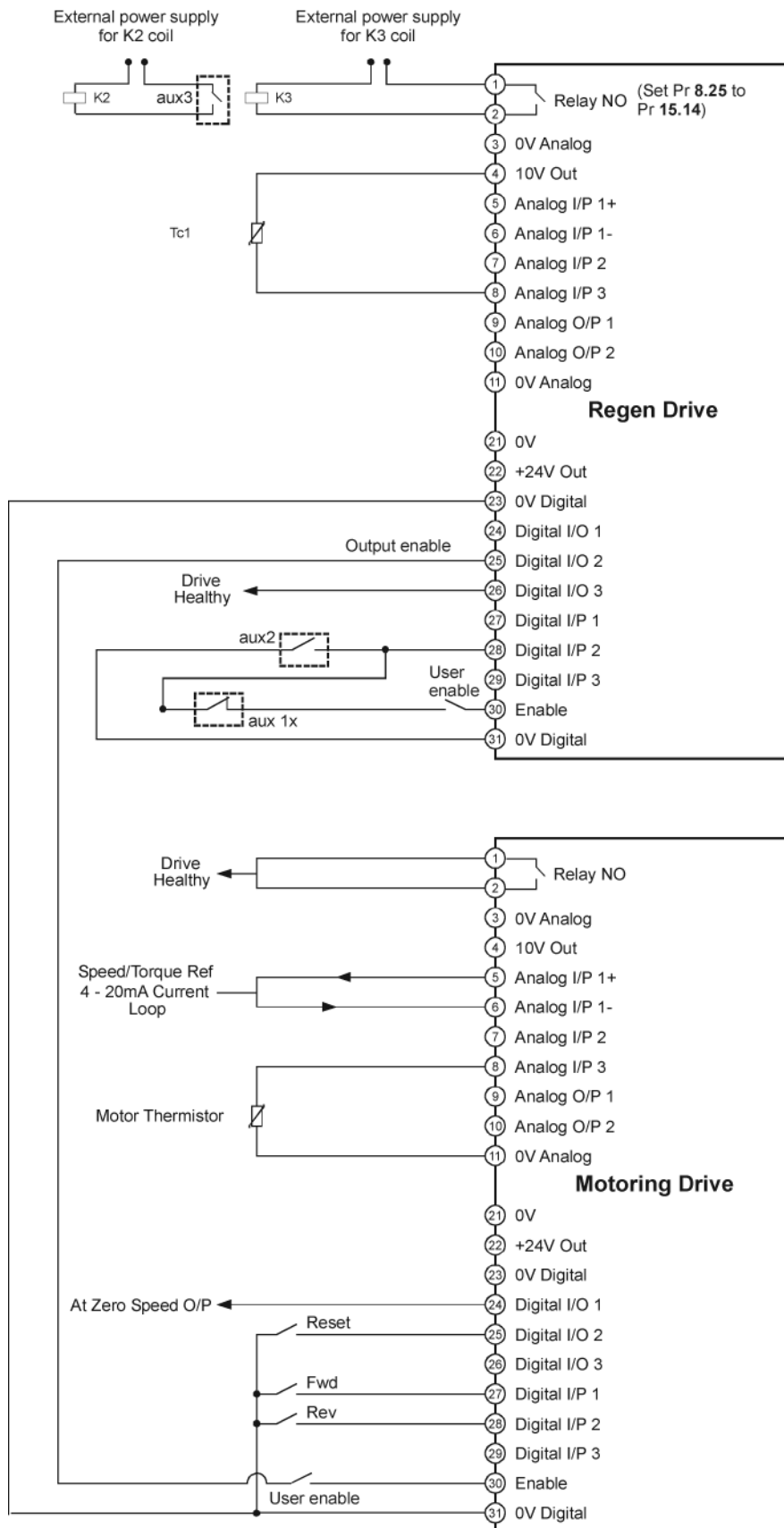


Figura 49 Configuración Standard. Conexión de control

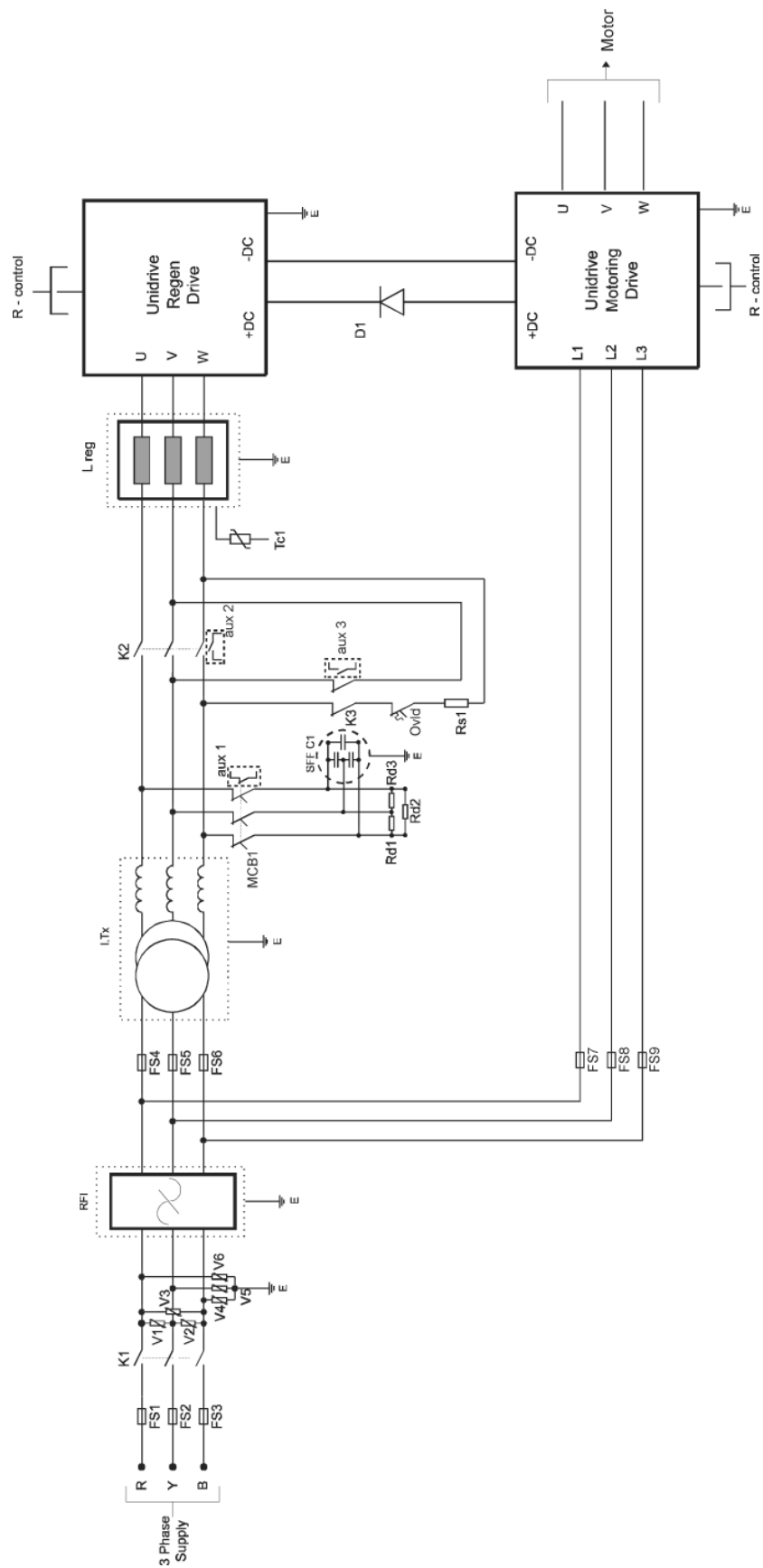


Figura 50 Sustitución de la resistencia de Frenado. Conexionado de potencia

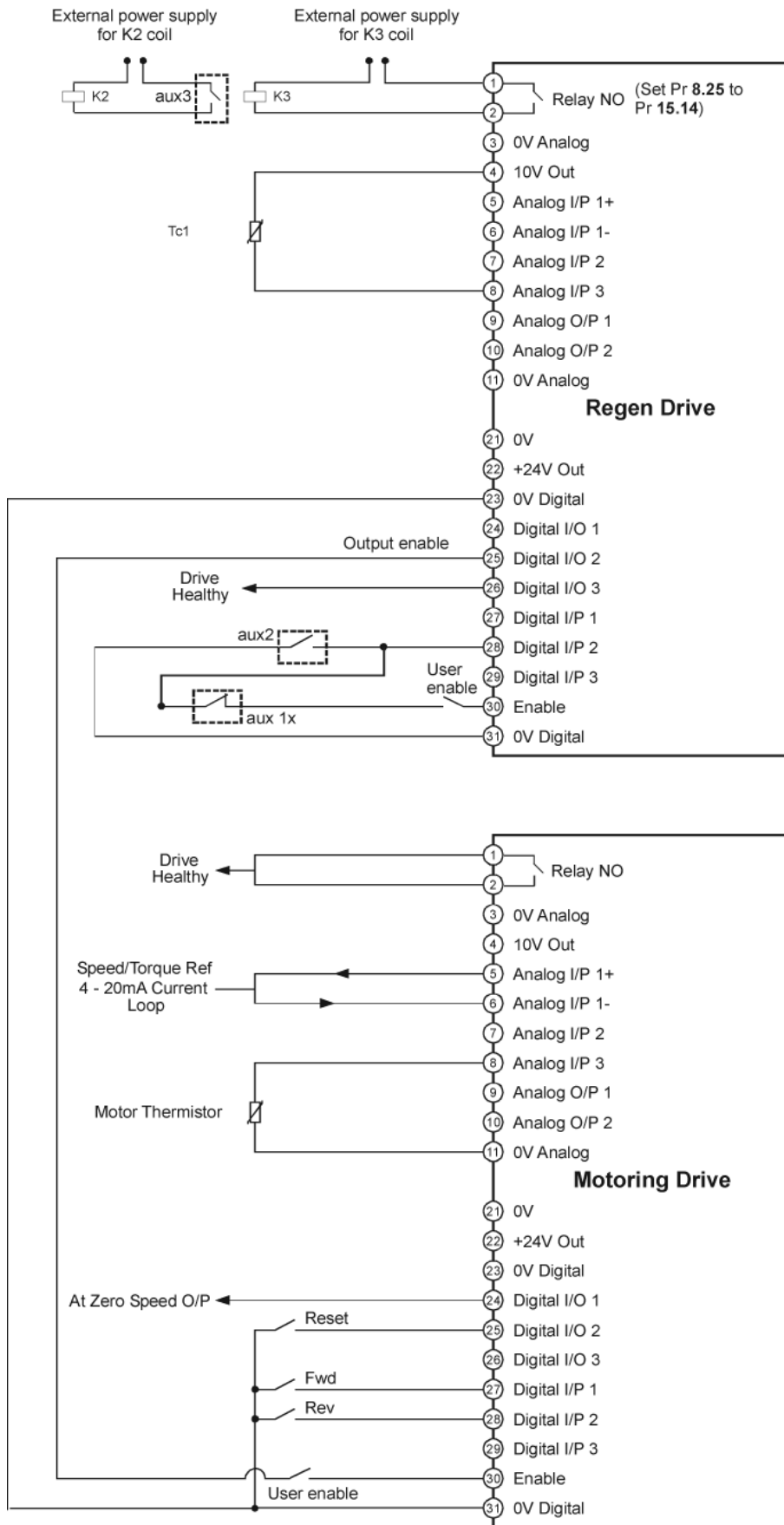
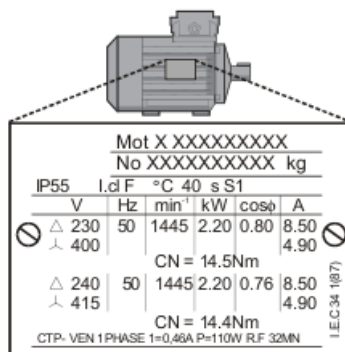


Figura 51 Sustitución de la resistencia de Frenado. Conexionado de Control.

5 Configuración de componentes

5.1 Variador de Frecuencia (Motor)

El motor es el encargado de proporcionar un par motor a dicho sistema, por lo que el funcionamiento de este determinará el de todo el sistema. Para ello será necesario fijar un valor de la frecuencia de sincronismo en su alimentación, el cual determinará el control del mismo.



Mot X XXXXXXXXXX						
No XXXXXXXXXX kg						
IP55 I.d.F °C 40 s S1						
	V	Hz	min ⁻¹	kW	cosφ	A
△	230	50	1445	2.20	0.80	8.50
λ	400					4.90
CN = 14,5Nm						
△	240	50	1445	2.20	0.76	8.50
λ	415					4.90
CN = 14,4Nm						
CTP- VEN 1 PHASE 1=0,46A P=110W R.F 32MN						
I.E.C.34 187						

Figura 52. Placa de características de un motor asíncrono

El motor asíncrono es controlado mediante un variador de frecuencia, el cual ya describimos con anterioridad. En primer lugar, y como ya se comentó en el apartado **Conexiones de control del variador de frecuencia (motor)**, se conectarán los terminales 8 y 11, que hacen referencia al termistor del motor, y 30 y 31 para la activación del accionamiento.

A continuación tenemos que seleccionar la macro a usar, en nuestro caso la macro 0, para realizar esto fijar el valor 2010 en el parámetro 0.00. Después seleccionaremos el tipo de referencia mediante el parámetro 0.05, asignando el valor 4 (control mediante teclado), para así poder seleccionar la velocidad del motor y el sentido de giro mediante el teclado. Después hay que configurar el variador para el motor que se va a utilizar.

Parámetro 0.42. Número de polos del motor: 2 polos

Parámetro 0.43. Factor de potencia: 0,84

Parámetro 0.44. Tensión nominal: 400V

Parámetro 0.45. Velocidad nominal del motor a plena carga: 2800 rpm

Parámetro 0.46. Intensidad del motor: 0.81 A

Parámetro 0.47. Frecuencia del motor: 50 Hz

Parámetro 0.48. Modo de funcionamiento: Lazo abierto

Finalmente, introducimos el valor 1000 en el parámetro 0.00, pulsamos M para memorizar y a continuación el botón rojo para guardar todos los parámetros introducidos. Para la marcha y paro del motor asíncrono, utilizaremos los botones verde y rojo respectivamente del variador de frecuencia, mientras que para aumentar o disminuir la velocidad, haremos uso de las teclas arriba y abajo .

5.2 Variador de Frecuencia (Servo)

En primer lugar, hay que cambiar el parámetro 0.48 a *SERVO* , para seleccionar el modo de funcionamiento *Servo-accionamiento de bucle cerrado* . Para que los cambios tengan efecto es necesario resetear el variador mediante la tecla RESET.

Para la configuración básica del codificador incremental, necesitaremos acceder a los menús auxiliares, por medio del valor 149 en el parámetro 0.00. Introducimos los siguientes parámetros en el variador:

Parametro 3.38. Tipo de codificador del accionamiento: Codificador en cuadratura con salidas de conmutación (Ab.Servo)

Parametro 3.36. Suministro de alimentación del codificador: 5V

Parametro 3.34. Impulsos por revolución del codificador del accionamiento: 4096 ppm

Parametro 3.39. Resistencia terminal del codificador del accionamiento: Todas activadas

El siguiente paso a llevar a cabo es configurar los parámetros de funcionamiento del servo, el parámetro 0.42 indica el número de polos del servo, en este caso se usa el valor por defecto, 6 indicando 6 polos. La intensidad Stall está en el parámetro 0.46, asignaremos el valor 4,3 A.

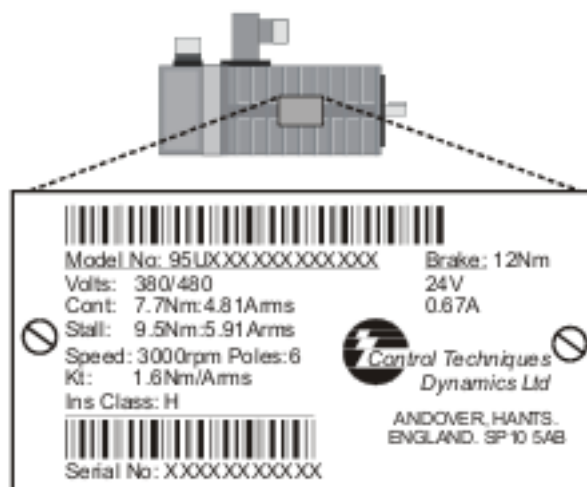


Figura 53. Placa de características de un servomecanismo

El siguiente parámetro que hay que modificar es la constante de tiempo térmica, que actúa como protección del motor, en el caso de que esta constante se determine incorrectamente el motor puede sufrir daños por calentamiento en los devanados. Esta constante es el parámetro 4.15, y para nuestro motor toma el valor de 58 segundos.

Los parámetros siguientes modifican la respuesta dinámica del motor, como son la ganancia proporcional (parámetro 0.07), la ganancia integral (parámetro 0.08), la ganancia de derivada (parámetro 0.09), y las rampas de aceleración y deceleración (parámetros 0.03 y 0.04). Estos parámetros no es necesario modificarlos, se dejan con el valor asignado por defecto.

Además, como este variador es el encargado de disipar la energía del sistema en la resistencia de frenado el parámetro 0.15 se ajusta en el valor *fast*. Para salvar los parámetros, introducimos el valor 1000 en el parámetro 0.00, pulsamos M y posteriormente el botón rojo del teclado.

No se hará uso de ninguna *Macro* predeterminada, debido a que la modificación en cuanto al funcionamiento del sistema se realizará mediante la variación directa de los parámetros correspondientes en los modos de control por velocidad y par, cuyos diagramas de funcionamiento están reflejados en el *Anexo 2*.

5.3 Aplicación web

5.3.1 Introducción. Lenguajes de programación

Como se indico en los capítulos iniciales, la finalidad del proyecto es la de reproducir en laboratorio las características reales a las que se ven sometidos los motores en la industria. Cuando la característica Par-Velocidad que se desea reproducir varia, en función de otros parámetros, es necesario un dispositivo flexible que establezca dichas relaciones y comande al resto, siendo el PC el dispositivo elegido para tal fin.

El interfaz utilizado será un navegador web, que facilitará la comunicación con el módulo SM-Ethernet que controla el variador de frecuencia. La elección de un navegador web como interfaz de control estriba en la gran variedad de plataformas (ordenadores de sobremesa, ordenadores portátiles, PDAs, teléfonos móviles,...) sobre las que podemos utilizar este tipo de programas.

Para un correcto funcionamiento se hace necesario una serie de herramientas de software, basado en el lenguaje de programación a utilizar para la administración de los parámetros del variador de frecuencia en los que se fundamenta el control del sistema, un editor de textos de dicho código de programación y un intérprete del código de programación que posibilite el funcionamiento de los programas desarrollados. Además, el código deberá ser alojado en el servidor web que posee internamente el módulo SM-Ethernet, por lo que necesitaremos de un cliente FTP

(Files Transfer Protocol) para el almacenamiento de dichos archivos.

5.3.1.1 HTML

HTML, siglas de **HyperText Markup Language** (*Lenguaje de Marcado de Hipertexto*), es el lenguaje de marcado predominante para la construcción de páginas web. El lenguaje de marcado es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de las llamadas etiquetas, rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script o conjunto de instrucciones (como es el ejemplo del lenguaje JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

5.3.1.2 Javascript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos y cuyos programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios, aunque será necesario indicar dicho código en el lenguaje de programación HTML mediante las etiquetas <script> y </script>.

5.3.1.3 Editor de código fuente HTML y JavaScript (BlueFish)

Aunque la edición del código fuente se puede realizar mediante cualquier tipo de editor de textos, incluso los más básicos como el Notepad del sistema operativo Windows, se ha utilizado el editor de páginas web BlueFish con licencia de código abierto GNU.

Cuenta con características tales como rapidez, posibilidad de abrir múltiples archivos simultáneamente, soporte multiproyecto, soporte para archivos remotos mediante gnome-vfs, marcado de sintaxis personalizable basado en expresiones regulares compatibles con Perl, soporte para sub-patrones y patrones predefinidos (para HTML, PHP, Javascript, JSP, SQL, XML, Python, Perl, CSS, ColdFusion, Pascal, R, Octave/Matlab), diálogos para etiquetas HTML, asistentes para creación fácil de documentos, creación de tablas, marcos (frames), soporte para múltiples

codificaciones, trabajo con diferentes juegos de caracteres, numeración de líneas, menús desplegables, barras de herramientas configurables, diálogo para insertar imágenes, buscador de referencia de funciones, Integración personalizable con varios programas (make, javac, etc), resaltado de sintaxis (C, Java, JavaScript, Python, Perl, ColdFusion, Pascal, R y Octave), traducciones completas a aproximadamente veintidós idiomas. Además funciona sobre múltiples sistemas operativos compatibles con POSIX (Portable Operating System Interface) tales como Linux, FreeBSD, MacOS-X, OpenBSD, Solaris and Tru64.

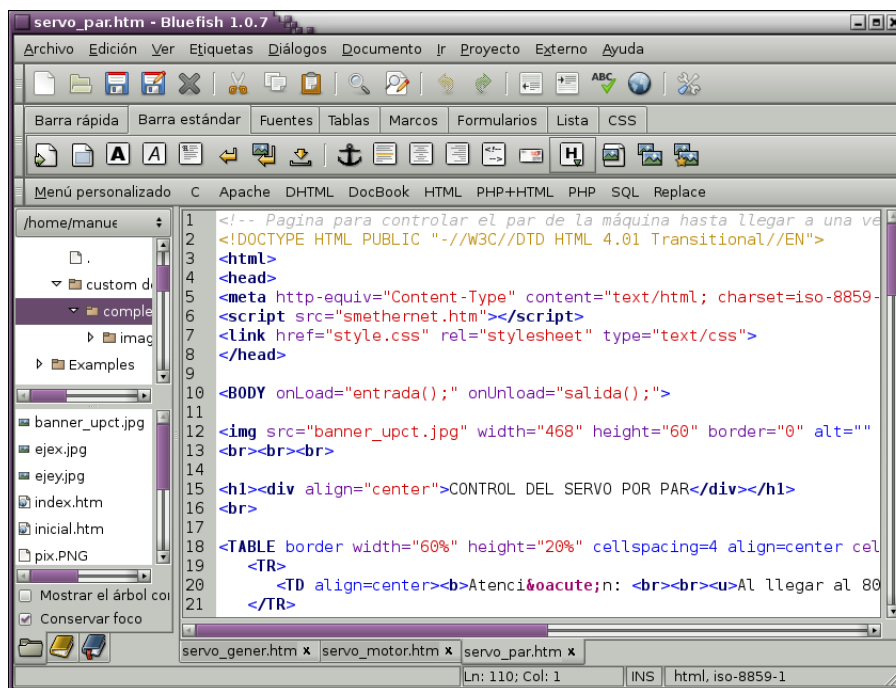


Figura 54. Captura de pantalla del editor BlueFish

5.3.2 Desarrollo de las aplicaciones

Para el desarrollo de la aplicación web de control del variador de frecuencia, se ha seguido un procedimiento de construcción del código fuente en HTML y JavaScript. Como vemos en la Figura 55, se ha procedido de manera ordenada para la consecución de las distintas etapas y poder así continuar con la siguiente en el diagrama.

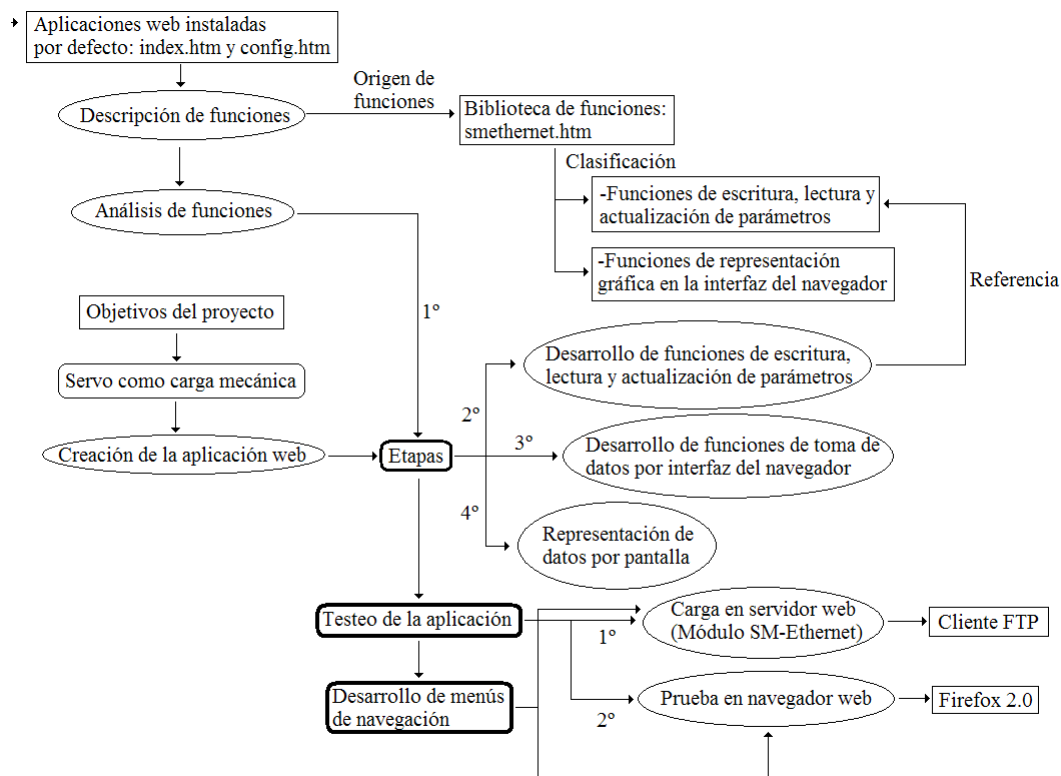


Figura 55. Esquema de desarrollo de la aplicación web

El módulo SM-Ethernet contiene por defecto, como servidor web, una serie de páginas web almacenadas en ella de nombres *config.htm* e *index.htm*, las cuales pueden ser descargadas por el usuario para su modificación y posterior actualización. El código fuente de las mismas sirve como referencia para el estudio de las distintas funciones que son llevadas a cabo por ellas. Es decir, el funcionamiento de dichas páginas web, podrá ser aplicado en aquellas que serán creadas por el usuario.

La estructura de la que hacen uso las páginas web instaladas por defecto en el modulo muestran una referencia web al archivo *smethernet.htm*. Este archivo almacena una gran variedad de funciones, por lo que actuará de biblioteca, las cuales son llamadas a ejecutar por el código fuente de las páginas web *config.htm* e *index.htm*. La biblioteca *smethernet.htm* está estructurada por una serie de funciones las cuales pueden ser clasificadas básicamente de dos tipo:

- Funciones para la lectura, escritura y actualización de parámetros del variador de frecuencia a través del módulo SM-Ethernet.
- Funciones de representación y captura de datos por medio de la interfaz gráfica ejecutado en el navegador web.

Las funciones que desempeñarán una mayor actividad en las aplicaciones web creadas son las consideradas en el primer grupo comentado anteriormente. Analizando más en profundidad las distintas funciones para la creación de objetos en páginas web, el manejo de los parámetros se

realizará con el módulo SM-Ethernet por medio del uso de la dirección (que en el caso de nuestro módulo es 192.168.1.100, introducido cada octeto mediante los parámetros 16.10, 16.11, 16.12 y 16.13), siendo para la lectura del valor de un parámetro del variador de frecuencia, para el caso del parámetro 1.21:

<http://192.168.1.100/US/1.21/dynamic/readparval.xml>

Cuya respuesta por parte del módulo SM-Ethernet al navegador web será, aplicado a un ejemplo:

```
<parameters>  
  
<parameter name="1.21" value="55" dp="1" text="5.5Hz" />  
  
</parameters>
```

Es decir, el parámetro 1.21 (el cual define la velocidad prefijada en el variador de frecuencia), tendrá un valor de escritura igual a 55 (value= 55), con un decimal (dp= 1), quedando el texto que define dicho valor igual a 5.5 Hz.

En cuanto a la escritura de un valor en un determinado parámetro del variador de frecuencia, el código utilizará el enlace web siguiente para dicha comunicación:

<http://192.168.1.100/US/parametro a escribir=valor a dar/dynamic/writeparval.xml>

En este caso, respuesta por parte del módulo será, si esta se haya llevado a cabo correctamente:

```
<status value= 0 text= ok />
```

Las funciones donde encontramos la utilización de los vínculos anteriormente mencionado serán `SMDData.getData`, para la lectura, y `SMDData.setParameter`, para la escritura, ambos localizados en el archivo `smethernet.htm`. Toda función que requiera del manejo de los parámetros del variador de frecuencia, ya sea para una representación gráfica o manipulación de sus valores, requerirá de la llamada de ambas funciones.

Para ello, se realiza la creación de una lista de parámetros mediante la función `SMDData.registerElement`, el cual pide un parámetro al módulo SM-Ethernet para ser continuamente monitoreado, actualizándolo de manera periódica en un tiempo definido por la variable *pollrate*. Tanto el nombre de los parámetros como su valor en las unidades correspondientes será almacenado en la lista creada, pudiéndose acceder a ellos por parte de aquellas funciones que necesiten utilizarlos.

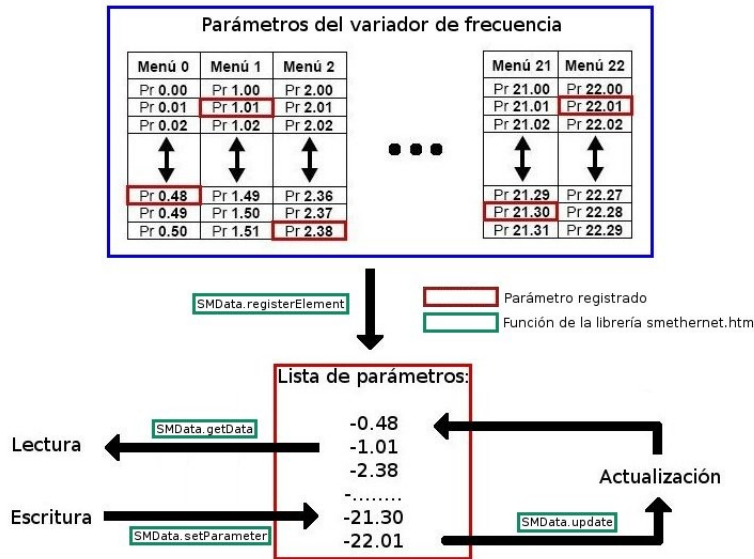


Figura 56. Esquema de manipulación de parámetros

Como se indico al inicio, el objetivo prioritario del proyecto es conseguir generar pares resistentes gracias a un servo, en función de la velocidad de giro del motor asíncrono. Por lo tanto, el sistema consistirá básicamente en un sistema realimentado en tiempo real, en el cual manejaremos fundamentalmente dos variables:

- La velocidad instantánea del sistema, reflejado en el parámetro 3.02, el cual se actualizará continuamente para su lectura.
- El par resistente impuesto, en el parámetro 4.08, el cual será fijado en función de la velocidad de giro.

La dependencia entre ambos parámetros se realizará con la ecuación siguiente:

$$T_R = A + B \cdot F + C \cdot F^2$$

Siendo T_R el valor del par resistente y F la relación entre la velocidad instantánea y un valor fijo de 3000 rpm. Los parámetros A , B y C serán fijados por el usuario, ya que dichos valores permitirán que el par resistente actúe en función de la velocidad en forma de constante, rampa o parábola. Para introducir el valor de A , B y C , se hará uso de la función en JavaScript `Number(prompt())` de la siguiente forma:

```
A=Number(prompt("Valor de A:",""));
```

Siendo A , B y C variables internas de la aplicación, por lo que no ralentizarán el uso de las

comunicaciones entre el PC y el módulo SM-Ethernet. Además, dicha aplicación debe poseer una serie de limitaciones en caso de error humano o funcionamiento del sistema fuera de los límites de seguridad del mismo. Es decir, en caso de introducir un par máximo excesivamente elevado, el propio sistema actuará modificando dichos valores directamente

```
if (par_res<-35) {par_res=-35;};  
if (par_res>35) {par_res=35;};
```

Además, el sistema, en caso de cambio del sentido de giro del motor, variará el signo del parámetro de par resistente, para adecuarlo a dicho sentido.

```
if (vel<0) {par_res=0-par_res;};
```

Como medida adicional de seguridad, a velocidades menores a 1000 rpm y para evitar que el eje cambie bruscamente su sentido debido al par que se opone al movimiento, el par resistente se hará nulo.

```
if ((vel<1000)&&(vel>-1000)) {par_res=0;};
```

A modo de ejemplo, en el caso de que configurásemos los parámetros de la siguiente forma, $A=20$, $B=0$ y $C=0$, la curva del par resistente en función de la velocidad quedaría según se refleja en la Figura 57

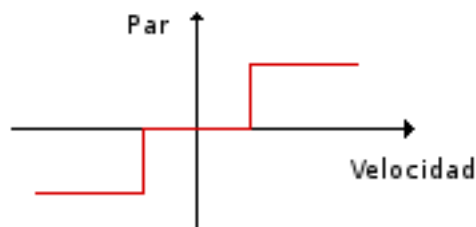


Figura 57. Relación Par Velocidad mediante la aplicación web

Para un mejor control de la aplicación web, se ha dotado a la interfaz una serie de gráficos donde se representarán aquellos datos de interés. Dicha representación de valores se podrá conseguir mediante el uso de marcos o tablas ya definidos por defecto en la librería que encontramos en el archivo smethernet.htm, como es el caso de el display el cual representa un parámetro durante un tiempo determinado (Figura 58), utilizado en la página servo_gener.htm

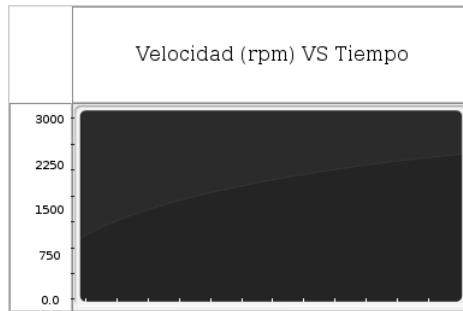


Figura 58. Diagrama de velocidad en el tiempo en servo-gener.htm

Además de la utilización de dichos marcos o tablas, se ha realizado un código para la representación gráfica del estado de funcionamiento del sistema de manera instantánea, donde podremos visualizar el valor del par resistente y la velocidad del eje (Figura 59), permitiendo al usuario un mayor control del sistema maquina asíncrona-servomecanismo.

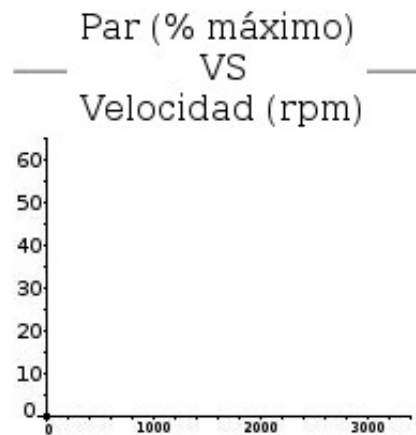


Figura 59. Gráfica Par-Velocidad en servo-gener.htm

5.3.3 Carga de la aplicación web en el módulo SM-Ethernet. El cliente FTP

FTP es un protocolo de transferencia de archivos entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor, de manera que desde un equipo cliente nos podemos conectar a un servidor para descargar archivos desde él o para enviarle nuestros propios archivos independientemente del sistema operativo utilizado en cada equipo.

Como ya se ha descrito con anterioridad, el control del variador de frecuencia será realizado desde un servidor web incluido en el modulo SM-Ethernet, el cual actualiza de manera permanente los distintos valores de los parámetros del variador. Es por ello necesario el uso de un cliente FTP,

cuya función se basa en poder alojar los archivos e imagenes creados por el usuario que posteriormente visualizaremos en el navegador web dirigiéndonos a la dirección deseada.

Entre los distintos clientes FTP del mercado, haremos uso del software Windows FTP, el cual presenta una gran sencillez en su manejo y que posibilita una funcionalidad ajustada a las necesidades de transferencia de archivos requerida en su versión limitada, sobre la cual trabajaremos.

Como podemos ver en la Figura 60, la pantalla del programa se dividirá fundamental en dos partes, cada una con un respectivo arbol de directorios, estando en la derecha de la pantalla la carpeta donde se alojan los programas a cargar en el servidor y que en ese instante se encuentran en el PC y en la izquierda de la pantalla la carpeta alojada en el servidor web, es decir, en el módulo SM-Ethernet.

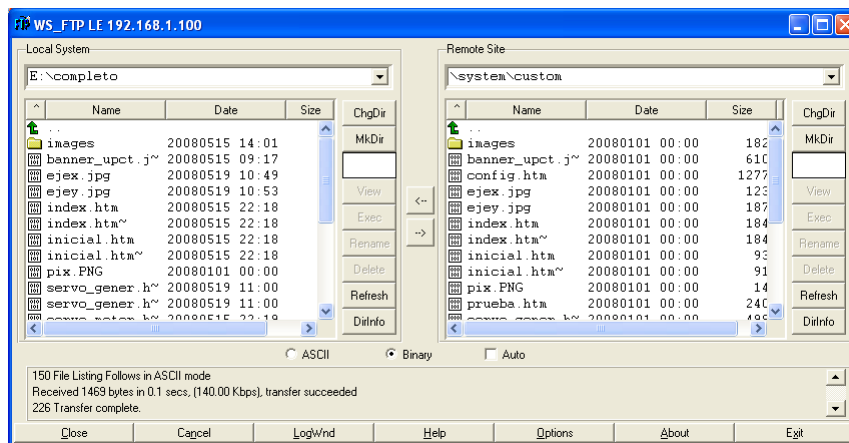


Figura 60. Captura de pantalla del programa Ws FTP

5.3.3.1 Configuración del cliente FTP

Para posibilitar el envío de archivos a la carpeta alojada en el servidor, se deben de seguir los siguientes pasos:

1. Abrir el programa FTP
2. Especificar la dirección del servidor
3. Indicar la carpeta donde están alojadas las páginas en el PC
4. Señalar el puerto como 21
5. Definir el servidor como de tipo FTP
6. Introducir el nombre de usuario como root y la contraseña ut72
7. Seleccionar modo activo
8. Seleccionar la carpeta remota en /system/
9. Conectar al módulo SM-Ethernet
10. Copiar los archivos deseados en la carpeta /system/custom/

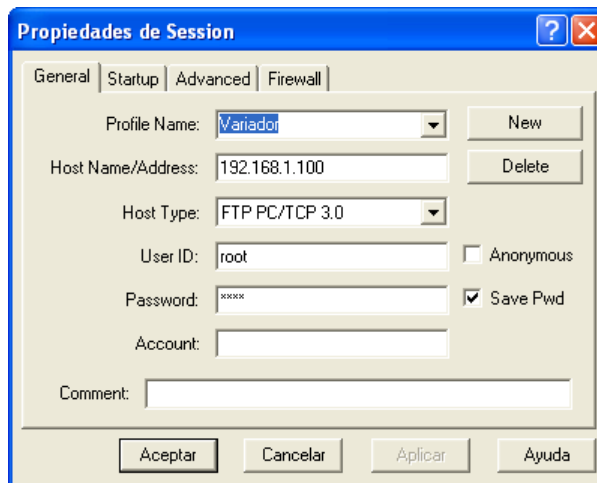


Figura 61. Inicio de sesión en el programa Ws FTP

5.3.4 Acceso y empleo de la aplicación web. El navegador web Firefox 2.0

Para el acceso a los archivos .htm para el control del variador de frecuencia, es necesario el uso de un navegador de internet, entre los que actualmente encontramos más utilizados en el mercado Internet Explorer, Mozilla Firefox, Netscape u Opera. La elección del navegador Mozilla Firefox se ha basado en las limitaciones que presenta el propio modulo SM-Ethernet, el cual solo posibilita un correcto uso con tres navegadores: Firefox, Netscape e Internet Explorer. Debido a la falta de servicio técnico de Netscape desde enero de 2008, dicho navegador es descartado para la utilización de este proyecto.

La elección del navegador Firefox se basa en el actual auge de programas de código abierto (a fecha de mayo de 2008, el uso de Firefox en sus múltiples versiones llega al 30% en Europa, colocándose segundo dentro del ranking de navegadores web más utilizados), cuyo interés en la empresa y primordialmente en el sector industrial se enmarca en un mayor control por parte de servicios técnicos propios para la resolución de errores en la programación.

5.3.5 Acceso a interfaz de control

Tras configurar correctamente las distintas conexiones TCP-IP entre PC y módulo SM-Ethernet y al encender el variador de frecuencia, podremos ejecutar el navegador web y dirigirnos a la dirección web <http://192.168.1.100> o a aquella que hayamos introducido en el módulo mediante la configuración este.

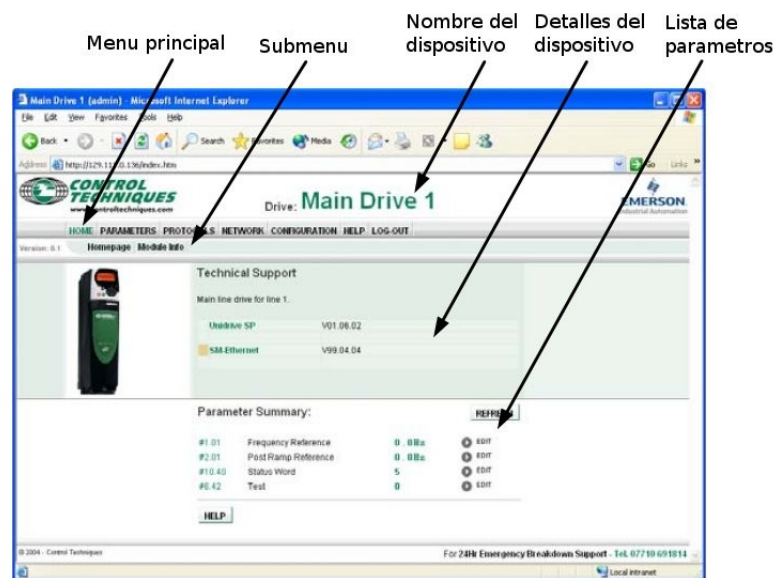


Figura 62. Página principal del servidor

Se nos presenta en el navegador la página principal, pudiendo definir distintos campos en la misma (Figura 62):

- Nombre del dispositivo: En el caso de que nos encontremos en una red con distintos módulos SM-Ethernet conectados a la misma red, nos ayudará a una rápida identificación del dispositivo utilizado en ese instante. Esto es gracias a que cada módulo SM-Ethernet poseerá su propia dirección IP dentro de la red local.
- Menú principal: Es el menú para la navegación principal por los distintos submenús que encontraremos por defecto instalados en el módulo. Para una navegación más avanzada, el usuario deberá registrarse pulsando sobre la opción **LOG IN**, lo que le permitirá el acceso a la escritura de los distintos parámetros del variador de frecuencia y por ende, el uso de las páginas web creadas para el control del servomecanismo. Por defecto, el nombre de usuario (*username*) será *root* y la contraseña a utilizar *ut72*.
- Submenús: Permite al usuario el acceso de las distintas opciones que ofrece la página web, como manipulación de parámetros, configuración de la conexión, priorización de comunicaciones, tiempo de desconexión y el acceso a las páginas web creadas.
- Detalles del dispositivo: Es la parte de la página donde vendrá representada la información del variador de frecuencia.
- Lista de parámetros: Información de los valores de los parámetros del variador de frecuencia.

Figura 63. Pagina de registro de usuario

Tras acceder a la pagina, pulsamos en el menú la opción *HOME* y en el submenú respectivo *CUSTOM PAGES*, trasladándonos a la página *index.htm*, la cual será nuestro acceso de inicio a los programas desarrollados según se indica en la Figura 64

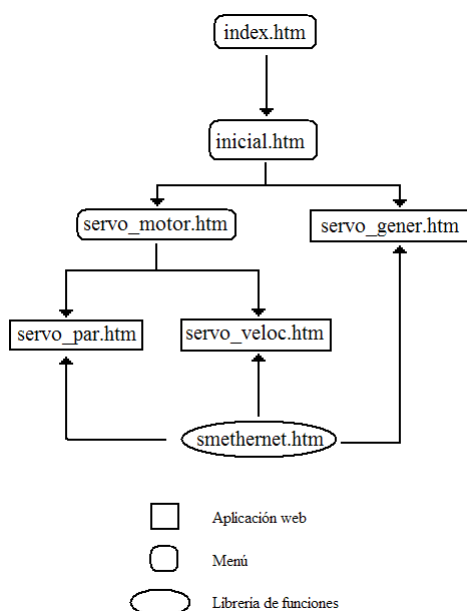


Figura 64. Diagrama de navegación por páginas web creadas

En la página *index.htm*, nombre elegido debido a que sustituirá a la instalada por defecto en el servidor web del módulo SM-Ethernet (Figura 65), visualizamos la opción de otorgar valores al servomecanismo para su configuración rápida o si estos ya han sido definidos con anterioridad por el usuario, pasar directamente a los menús de control.



Figura 65. Página index.htm

A continuación, accedemos a un nuevo menú, el cual nos da la opción de controlar el servomecanismo como un motor o entrar en la pagina de control del sistema de frenado regenerativo (inicial.htm, Figura 66).



Figura 66. Página inicial.htm

En la opción *Modo MOTOR*, tras ser pulsada, accederemos a un nuevo menú el cual da la posibilidad de un control mediante la configuración del parametro de velocidad (gracias a la página servo_veloc.htm, Figura 67) o del par motor (gracias a la página servo_par.htm, Figura 68) a generar por parte del servomecanismo.

El control por velocidad se activa mediante la configuración de los parametros 4.11, asignándole el valor 0 para permitir el control del servomecanismo por velocidad y no por par, y 1.14, asignando el valor 3, facilita el control externo gracias al parametro 1.21, el cual almacenará la velocidad deseada en revoluciones por minuto. En el caso de control por par, se habilita el parámetro 4.11 con el valor 1, para dicho control, mientras que el valor 4.08 almacenará el valor de par prefijado en función del porcentaje máximo a desarrollar por el servomecanismo.

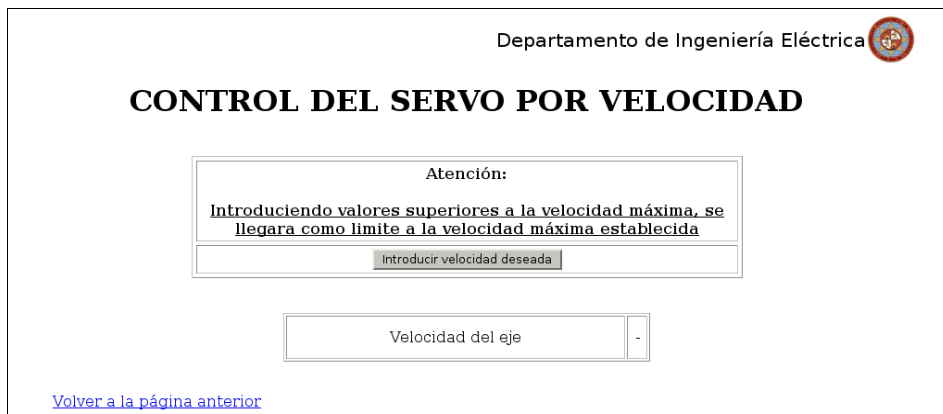


Figura 67. Página servo_veloc.htm

Como se observa, se limitará el funcionamiento debido a razones de seguridad del servo y del variador de frecuencia que lo controla, ya que en caso de sobreintensidades del sistema o velocidades críticas, el equipo correrá riesgo de avería.



Figura 68. Página servo_par.htm

Además, tras acceder a alguna de las dos páginas anteriores, el navegador web nos informará (Figura 69) de la necesidad de desacoplar el servomecanismo y el motor asíncrono para un correcto funcionamiento, ya que si se pusiera en marcha, el servomecanismo arrastraría el motor asíncrono durante el arranque.

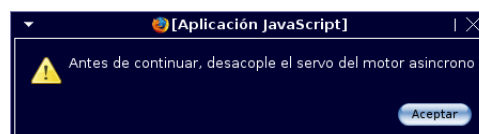


Figura 69. Advertencia inicial en servo_veloc.htm y servo_par.htm

Volviendo al menú que aparece en la página inicial.htm, al seleccionar la opción *MODO Generador*, a continuación el navegador nos mostrará la página servo_gener.htm (Figura 70), cuya función principal será la de generar un par resistente el cual depende de la velocidad del eje. La pantalla quedará dividida en tres zonas:

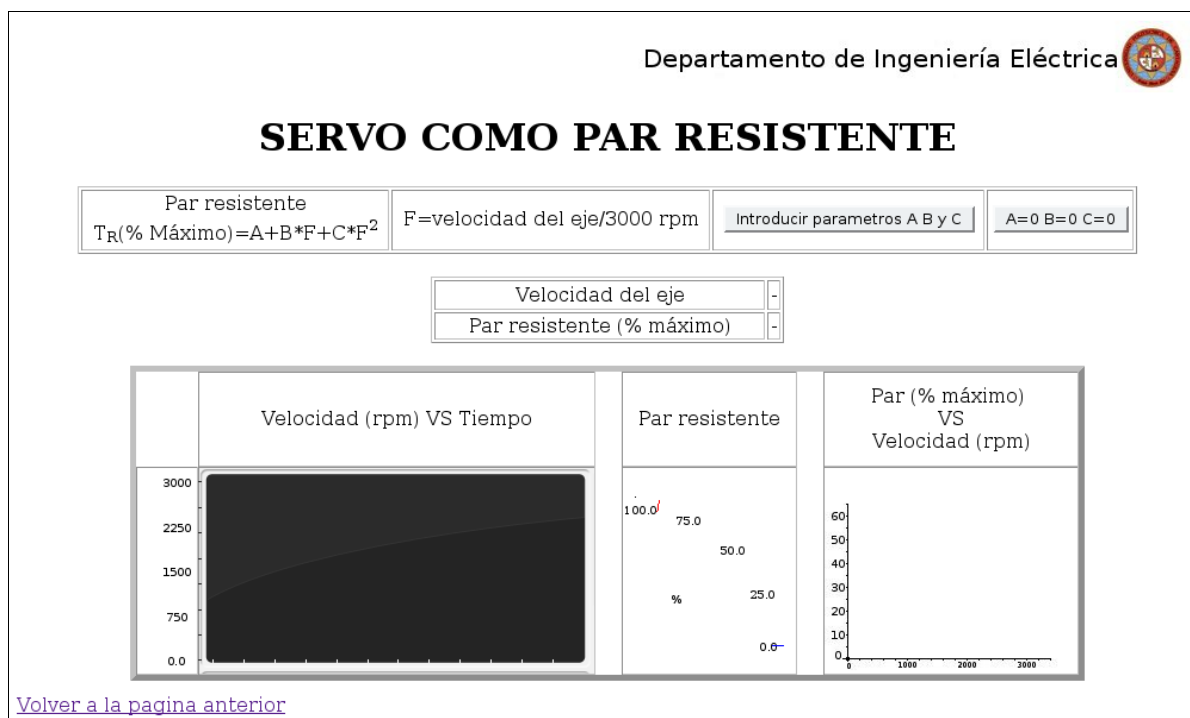


Figura 70. Página servo_gener.htm

- La superior, donde se informa al usuario de las fórmulas que relacionan la velocidad y el par impuesto, y pulsadores para posibilitar la introducción de los parámetros A, B y C
- La intermedia, donde se muestra los valores de velocidad y par resistente
- La inferior, con distintos displays de información, descritos a continuación:
 - Representación de la velocidad de giro del eje en función del tiempo
 - Porcentaje de carga o par resistente impuesto por el servo
 - Interpretación gráfica del valor del par resistente y velocidad de giro del eje en cada instante

El funcionamiento de la página se basa en que introduciremos inicialmente los valores A, B y C, parámetros que determinan la curva del par resistente. Se ha establecido un valor máximo del 35 %, ya que el motor asíncrono no será capaz de superar dicho par, provocando una inversión en el

sentido de movimiento. Como medida de seguridad, se han habilitado limitaciones a bajas revoluciones para evitar también dicho efecto en caso de fallo en las comunicaciones, ya que si se produjese una desaceleración de manera brusca, sin tiempo de reacción por parte del sistema para una correcta lectura de parámetros, el sistema actuaría automáticamente desconectando el par a generar, asignándole el valor cero, y permitiendo que el motor asíncrono funcione en vacío.



Figura 71. Detalle del acoplamiento motor asíncrono al servo

6 Ensayos y resultados

6.1 Estudio de la potencia disipada en la resistencia de frenado

6.1.1 Introducción

Como ya se comentó anteriormente, el servomecanismo evacuará la energía generada mediante el variador de frecuencia. Este, a través de una salida de voltaje cuya forma de onda se asemeja a la representada en la Figura 72, nos permitirá establecer mediante toma de datos el valor instantáneo del voltaje en dicha resistencia durante un tiempo determinado

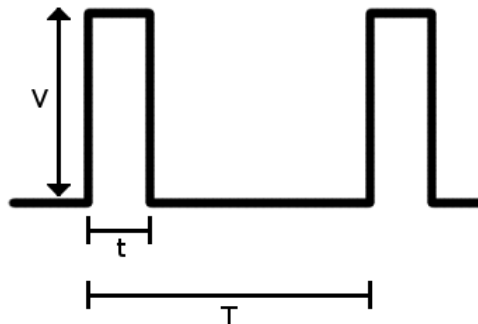


Figura 72. Onda de voltaje en la resistencia de frenado

Para el cálculo de la potencia disipada en la resistencia eléctrica (P_R) y extrañendo de la gráfica los valores de voltaje de pico (V), tiempo de conexión (t) y periodo de la onda (T), llegamos a la siguiente fórmula:

$$P_R = \frac{1}{T} \int_0^T V \cdot I dt = \frac{1}{T} \int_0^t \frac{V^2}{R} dt = \frac{t}{T} \cdot \frac{V^2}{R}$$

En cuanto a la potencia mecánica suministrada por el eje mecánico, utilizaremos la expresión, indicada en el capítulo:

$$P_{eje} = T \cdot \Omega$$

Donde P_{eje} será la potencia en el eje mecánico, T el par resistente del servomecanismo y Ω la velocidad del eje. En cuanto al rendimiento energético en cuanto a la relación entre la energía mecánica que suministra el eje y la energía disipada en la resistencia eléctrica, se expresará:

$$\eta = \frac{P_R}{P_{eje}}$$

Dichos cálculos serán necesarios para los posteriores análisis respectivos de evolución de la potencia disipada así de como varía el rendimiento energético en función de los distintos parámetros de la experimentación que se realizará

6.1.2 Procedimiento de experimentación

Como aplicación del sistema desarrollado, se ha llevado a cabo el estudio en un rango de pares resistentes del 10 % al 40 % del par máximo que es capaz de desarrollar el servomecanismo. Para ello, utilizaremos la página web de control remoto *servo-gener.htm*, y mediante la introducción de los parámetros A, B y C que controlan la función T_R (par resistente) según una parábola de la forma, al ser esta experimentación basada en valores constantes del par resistente (T_R), los valores de B y C serán iguales a 0, mientras que el valor de C variará en un rango del 10 al 40%. Además se ha elegido una velocidad determinada de sincronismo de la máquina asíncrona de 3000 rpm, es decir, suministraremos potencia por medio de un voltaje con forma de onda PWM, de frecuencia chopper igual a 3 kHz, cuyo armónico fundamental será de 50 Hz en un motor de dos polos

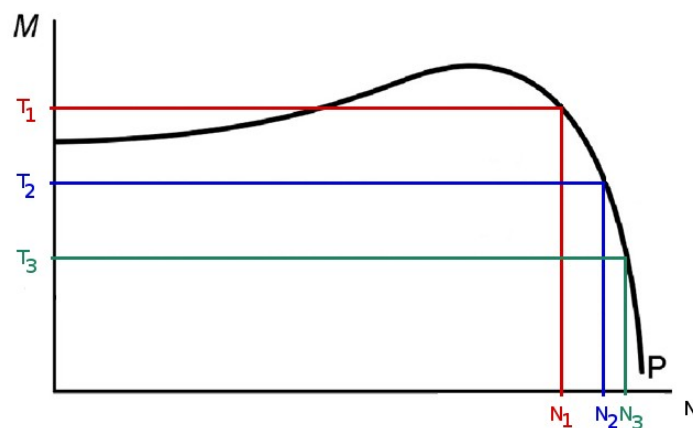


Figura 73. Curva de la máquina asíncrona y regímenes de funcionamiento

En cada uno de los ensayos, se tomarán los siguientes parámetros:

- Par resistente del servomecanismo
- Velocidad del eje
- Tensión respecto al tiempo en la resistencia de frenado
- Valor de la resistencia eléctrica, constante en todos los ensayos: 64,5 Ω .

6.1.3 Configuración de instrumentación para la medición de parámetros

Debido a la necesidad de la adquisición de datos de magnitudes como el voltaje y la intensidad eléctrica, se plantea el uso de instrumentación de medida. Para ello, definiremos los puntos más representativos del sistema, como se indica en la Figura 74. Dichos puntos se describen de la siguiente manera:

- A) Conexión de potencia del motor asíncrono. Medida de tensión de línea e intensidad de línea
- B) Conexión de potencia del servomecanismo. Medida de tensión de línea e intensidad de línea.
- C) Resistencia de frenado. Medida de la tensión en la resistencia de frenado y valor de dicha resistencia

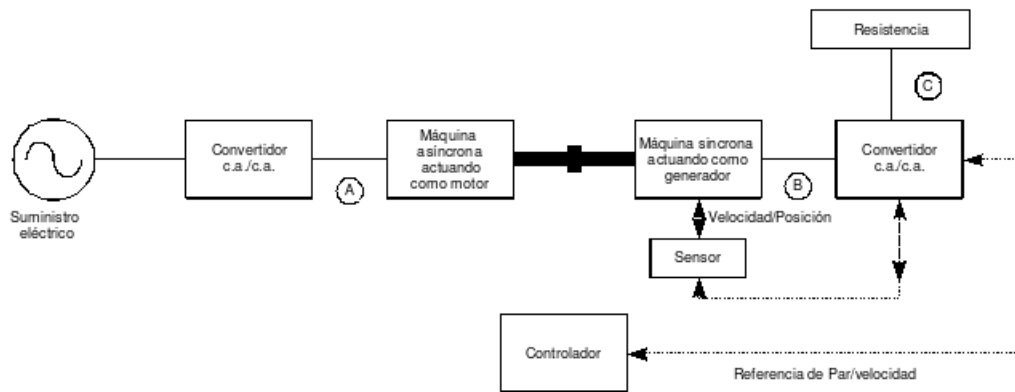


Figura 74. Diagrama de toma de datos

Para la realización de análisis y debido a la necesidad de simultaneidad en la toma de datos, además de las correspondientes pinzas amperimétricas para la medida de la intensidad y de sondas de voltaje, haremos uso de una tarjeta de adquisición de datos, la cual a su vez estará conectada al ordenador para poder almacenar los valores de las magnitudes a medir.

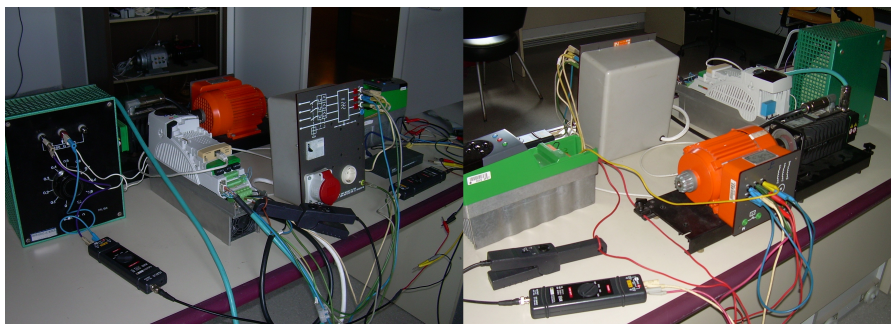


Figura 75. Sondas de medida en el variador del motor

En cuanto a configuración de los componentes de toma de medida, se ha optado por hacer pasar dos veces el cable sobre el cual queremos medir la intensidad a través de la pinza amperimétrica (Figura 77), para una reducción en el ruido que suele ser capturado durante la toma de medidas, por lo que en el momento de ser calculado el valor de la intensidad, se tendrá que dividir entre dos dicho valor.

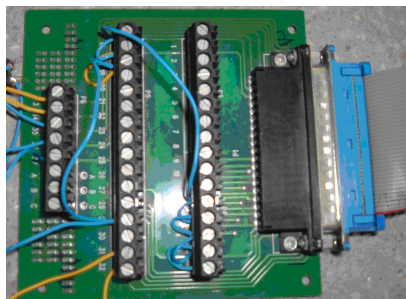


Figura 76. Placa de conexión a la tarjeta de adquisición de datos

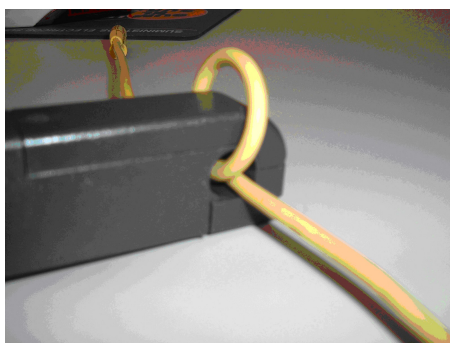


Figura 77. Detalle de configuración de la pinza amperimétrica

Los datos son capturados y almacenados en un archivo mediante el programa *Adquisición*, desarrollado previamente el Departamento de Ingeniería Eléctrica. Este programa presenta la primera pantalla (Figura 78) en que se configuran la frecuencia de muestreo, el tiempo de muestreo, el número de canales (entradas a la tarjeta), y el archivo de destino de los datos.

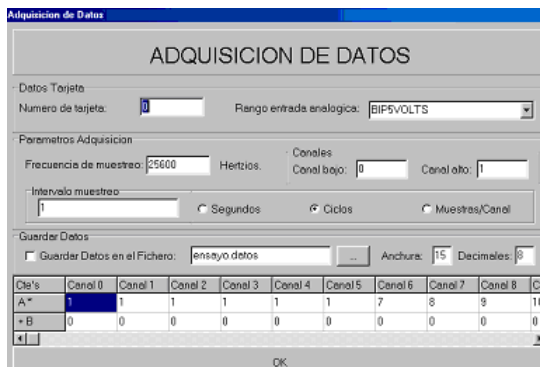


Figura 78. Menú del programa de adquisición de datos

La segunda pantalla (figura ...) es a través de la cual se capturan los datos, haciendo clic sobre **Tomar datos**, en ese momento los datos son capturados, en esta pantalla también se puede ver un extracto de la captura de la variable marcando la casilla correspondiente a cada canal. Los datos se almacenan en el correspondiente archivo haciendo clic en **Terminar**.

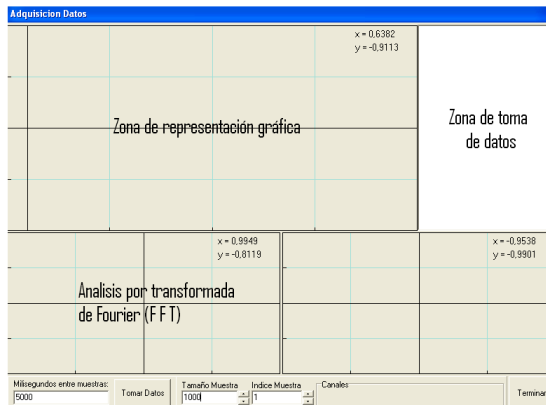


Figura 79. Pantalla de adquisición y representación de datos

Para un uso posterior del archivo, donde guardamos las serie de datos creadas a partir de la adquisición, se utiliza cualquier programa de hoja de cálculo (por ejemplo, Excel de Microsoft Office en caso de utilizar un sistema operativo Windows o la hoja de cálculo asociada a OpenOffice en caso de un sistema operativo Linux o Windows). En dichas hojas de cálculo, podremos procesar la información según las necesidades del usuario, ya sea por necesidades en cuanto al cambio de variables o la obtención de nuevas mágnitudes a partir de los datos adquiridos, así como representaciones gráficas de los mismos.

6.1.4 Análisis de resultados

En cuanto al tratamiento de datos, con los valores obtenidos y por medio de los procedimientos matemáticos explicados en apartados anteriores, pasaremos al análisis e interpretación de resultados gracias a la representación gráfica de estos.

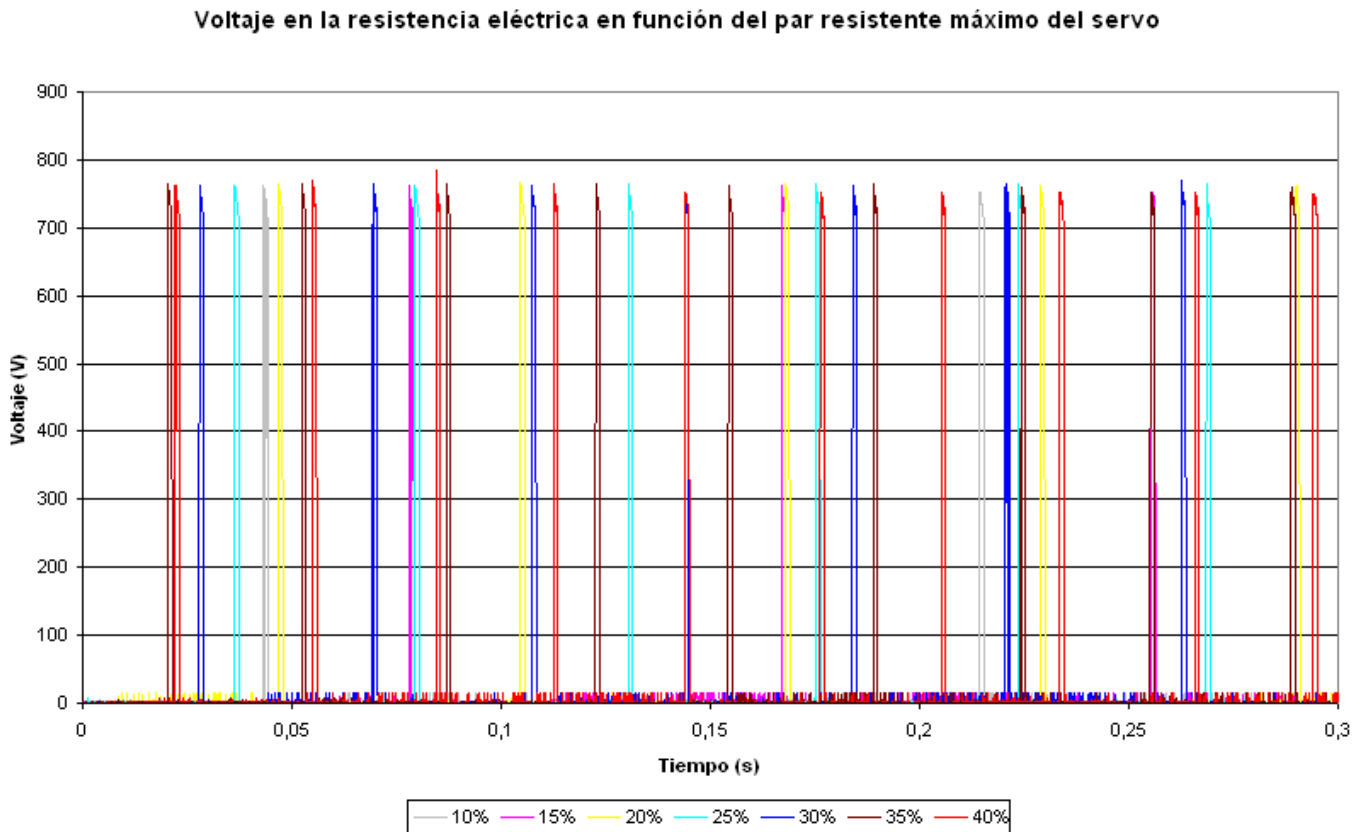


Figura 80. Gráfica del voltaje de la resistencia en función del tiempo a distintos valores del par resistente

Observando la gráfica representada en la Figura 80, vemos que en función de distintos pares resistentes, al ir aumentando el valor de estos, el periodo de la onda de voltaje disminuirá, es decir, si a esto se añade que el tiempo de duración del escalón en la onda de voltaje permanece constante, se deduce que la energía disipada en la resistencia de frenado será inversamente proporcional al par resistente impuesto. Una demostración de ello puede valorarse en la gráfica que relaciona el periodo de la onda de la resistencia eléctrica y la potencia disipada en la misma (Figura 81).

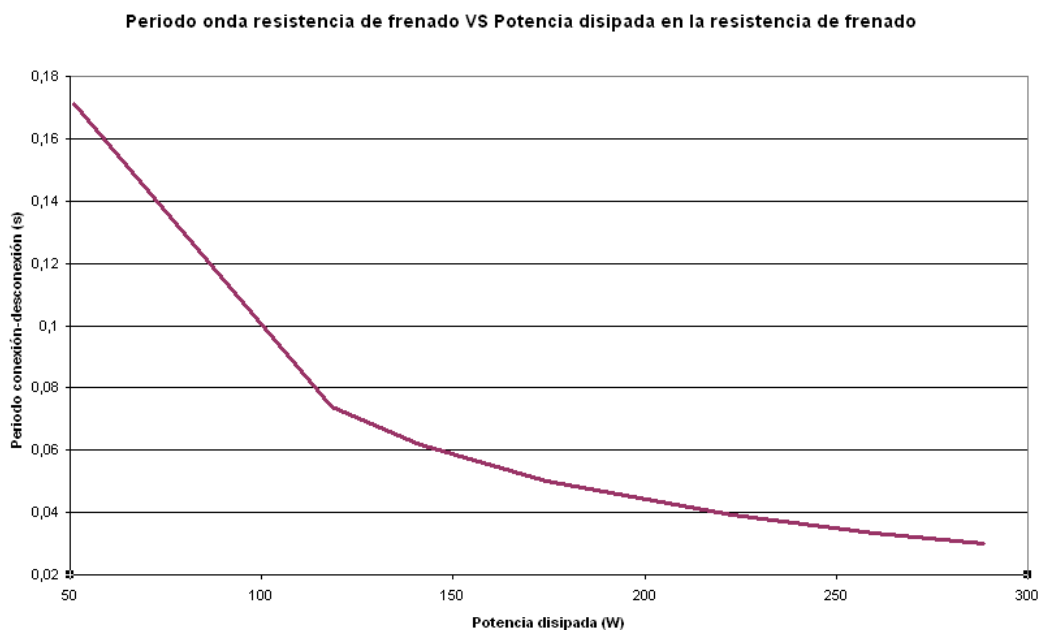


Figura 81. Gráfica de la relación entre el periodo de la onda de voltaje de la resistencia y la potencia disipada en la resistencia

De la misma forma, observamos en la Figura 82 una tendencia muy parecida al caso anterior si nos referimos a la relación que se establece entre el periodo de la onda de voltaje en la resistencia de frenado y la potencia en el propio eje, por lo que se puede considerar el origen de la forma de onda en cuanto a su periodo a los valores que se establecen en cuanto a la potencia en el eje, ya que dicha potencia mecánica pasará posteriormente a un potencia eléctrica que a continuación el variador de frecuencia tendrá que ser capaz de desalojar, en este caso, siendo consumida por una resistencia eléctrica.

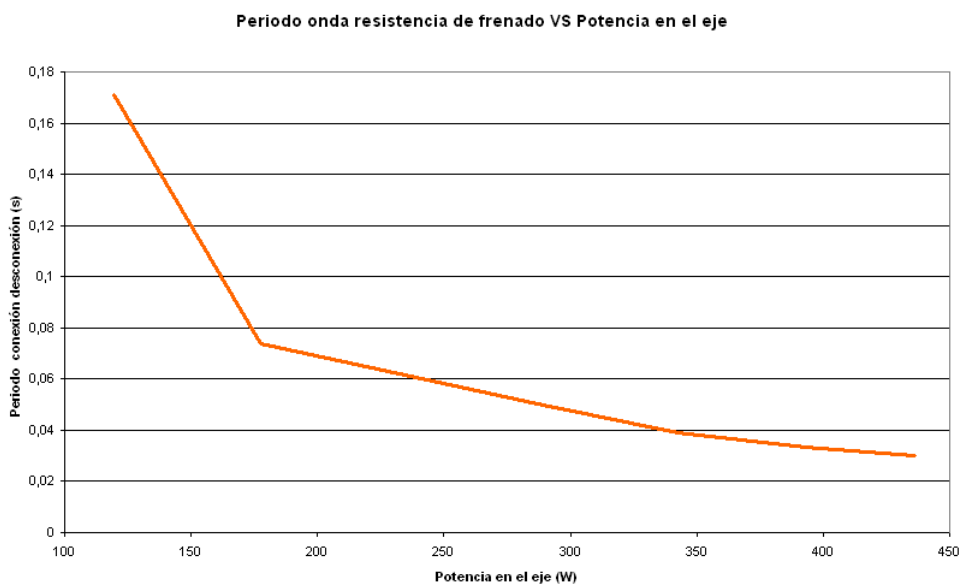


Figura 82. Gráfica de la relación entre el periodo de la onda de voltaje de la resistencia y la potencia en el eje mecánico

Al representar la gráfica (Figura 83) que relaciona el par resistente con la velocidad en el eje, se observa la misma forma de gráfica que en la zona estable de la curva característica de un motor asíncrono, debido a que la obtención de los distintos puntos se ha llevado a cabo por medio de distintos estados variando el par resistente, el cual anula al par motor de la máquina asíncrona.

De esta manera podemos observar que el sistema diseñado sirve para el análisis de curvas características de motores asíncronos, y al poder recurrir a un código de programación para la automatización de tareas, se puede diseñar un sistema el cual por si solo analizase todo el rango de valores de pares resistentes, siempre y cuando el comportamiento de tal sistema automatizado no comporte un riesgo debido a la parametrización de valores fuera de sus límites de seguridad. Dicho riesgo puede ser, gracias a la experiencia en el desarrollo de este proyecto, otorgar valores al par resistente por encima del valor de par motor capaz de desarrollar el propio motor asíncrono.

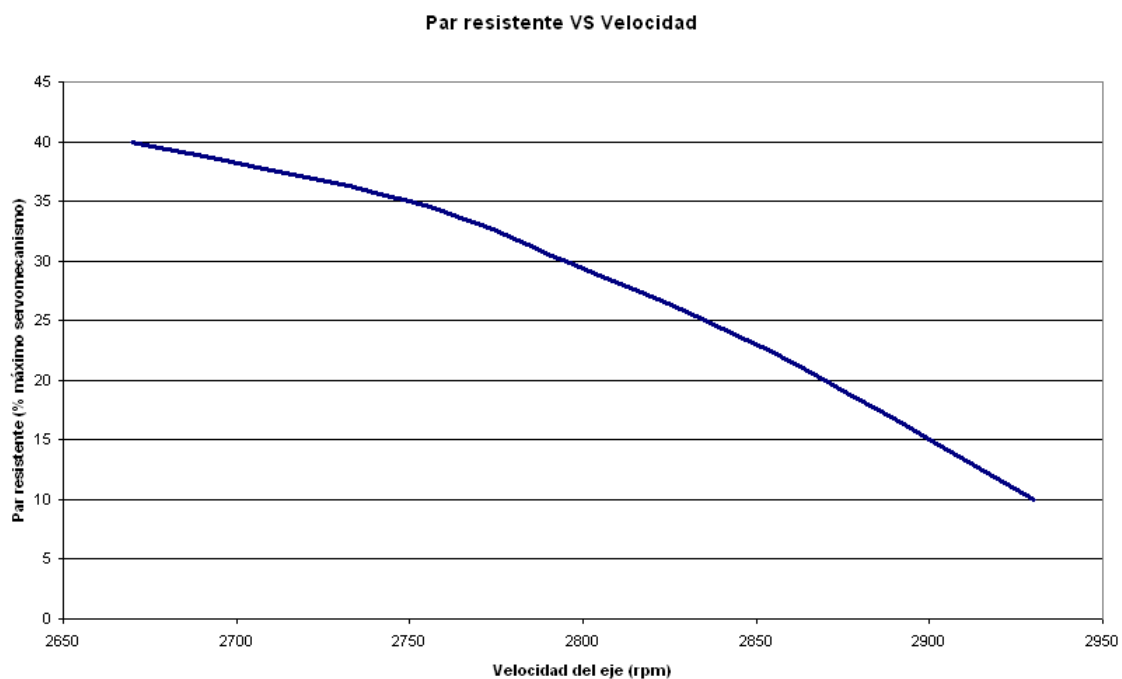


Figura 83. Gráfica de la relación par resistente y velocidad del eje

Como se observa en la Figura 84, la relación que se establece entre el par resistente y la potencia del eje se aproxima a una función lineal, directamente proporcional, ya si atendemos a los procedimientos de cálculo, la potencia en el eje ha sido obtenida mediante la ecuación:

$$P_{eje} = T \cdot \Omega$$

Lo que, gracias a considerar que los valores de deslizamiento, es decir, la relación entre la velocidad del eje y la velocidad de sincronismo (la cual permanecerá constante durante las distintas configuraciones de experimentación), en máquinas asíncronas es cercano a uno, el valor de la velocidad del eje no fluctuará más del 10%, y por lo tanto, el valor de la potencia del eje, como ya

se indico, variaría fundamentalmente por el par resistente, parámetro sobre el cual actuamos en la experimentación.

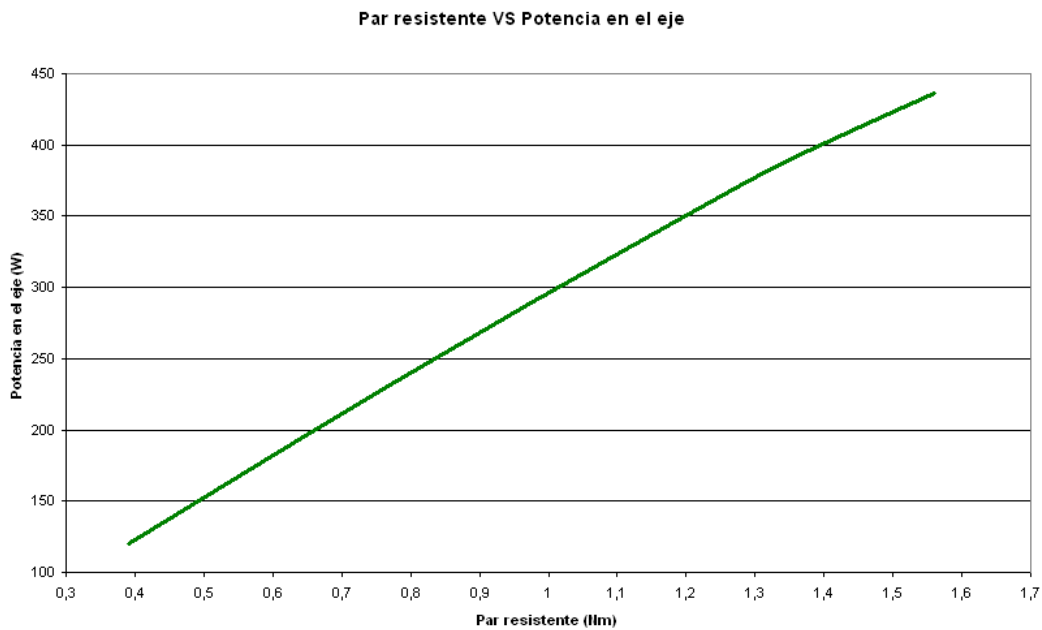


Figura 84. Gráfica de la relación par resistente y potencia en el eje

Pasando a analizar el rendimiento del sistema, teniendo en cuenta que la relación de dicho rendimiento se establece como el cociente entre la potencia eléctrica disipada en la resistencia eléctrica o, de manera más general, la potencia que evacua el variador de frecuencia conectado al servomecanismo, y la potencia mecánica en el eje, podemos observar (Figura 84) que por debajo de un cierto valor del par resistente impuesto, en torno a 0,5 Nm, decrecerá considerablemente dicho rendimiento. Sin embargo, para valores superiores del par resistente, el sistema presenta un rendimiento estable en un rango del 60% al 67%.

Hemos de considerar dichos resultados bajos si son comparados con la eficiencia que demuestran ciertas máquinas eléctricas, vease como ejemplo los transformadores eléctricos. Sin embargo, se ha de tener en cuenta que el conjunto del sistema si se analiza el proceso de conversión partiendo de la potencia suministrada por el eje hasta su dispersión en forma calorífica en la resistencia eléctrica, múltiples factores tomarán parte dando lugar a pérdidas en el sistema.

Los factores a los que hacemos referencia pueden ser divididos en dos clases:

- Pérdidas mecánicas, las cuales aparecen en el propio eje por fenómenos de rozamiento. Un estudio en vacío, es decir, sin acoplamientos adicionales, del servomecanismo mostró que para superar el par resistente en el arranque de la propia máquina era necesario un par motor superior al 3,5% de su par máximo (0,1365 Nm) mientras que para conseguir un valor permanente de velocidad del servomecanismo cercano a 3000 rpm, el servomecanismo necesitó un valor del par motor equivalente al 5% de su par máximo (0,195 Nm).

- Pérdidas eléctricas, que a su vez distinguiremos de dos tipos:
 - Debidas al consumo del variador de frecuencia que controla el servomecanismo, que, a pesar de que para un correcto funcionamiento de este es necesario su conexión a una red trifásica, la intensidad que por estos cables circula, se comprueba que es nula o prácticamente despreciable a efectos comparativos de potencia del sistema e incluso en relación con otros modos de funcionamiento, vease caso de funcionar el servomecanismo en modo motor, llegando a la conclusión que el consumo realizado es debido al suministro eléctrico del propio servomecanismo que actúa en modo generador. De una forma general, dicho consumo se distribuiría en refrigeración, funcionamiento de dispositivos electrónicos, etc...
 - Debidas a resistencia eléctrica que presenta los cables conductores, para la conexión de los distintos elementos que conforman el sistema. Dicho valor puede considerarse despreciable si es comparado con el del apartado anterior.
- El rendimiento de las máquinas educacionales es menor

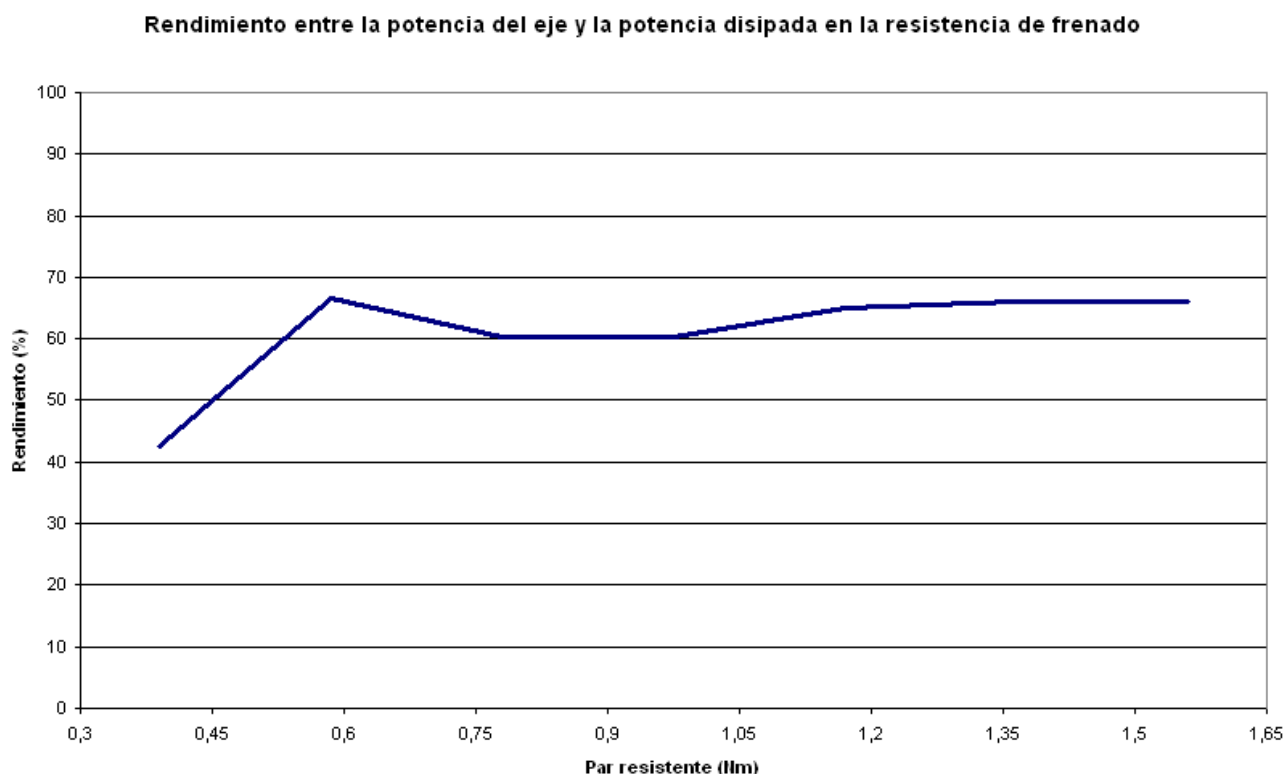


Figura 85. Gráfica de la relación par resistente y rendimiento energético del sistema

Por último en este apartado, se ha representado la relación entre la potencia disipada y la velocidad en el eje para cada instante (Figura 86). Tomando de partida los anteriores análisis, dicha

relación viene fundamentada en el hecho de que la velocidad es fijada en base a la curva característica del motor asíncrono, así de como el valor del par resistente establecido por el servomecanismo, es decir, dicha variable será dependiente a su vez de otras que han sido elegidas por el usuario, al igual que el término de potencia disipada, ya que este a su vez dependerá del valor del periodo de la onda anteriormente estudiada.

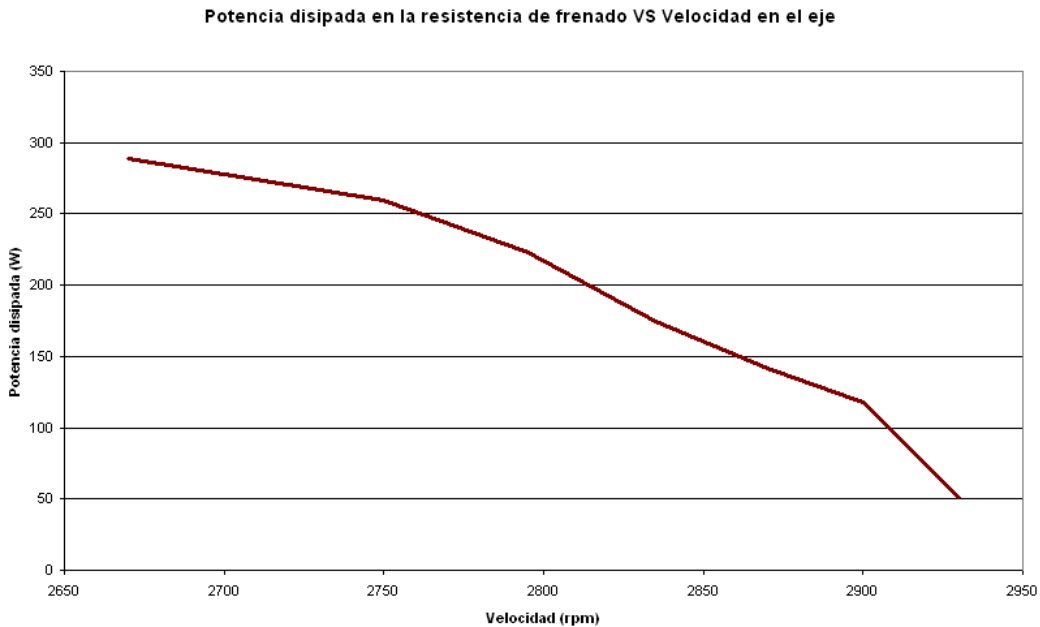


Figura 86. Gráfica de la relación potencia disipada en la resistencia eléctrica y velocidad del eje

6.2 Otros ensayos

6.2.1 Análisis de las magnitudes de tensión e intensidad en el sistema

Se ha realizado adicionalmente una serie de toma de datos a partir de las magnitudes de voltaje e intensidad eléctrica en distintos puntos del sistema, para una configuración determinada de funcionamiento. Para ello, se fijó como condiciones de la curva del par resistente dependiente de la velocidad del eje de la siguiente manera:

$$T_R = 35 \cdot F \quad F = \frac{\text{velocidad del eje (rpm)}}{3000 \text{ rpm}}$$

Siendo el par resistente igual a T_R medido en porcentaje respecto al par máximo a desarrollar por el servomecanismo y F una magnitud adimensional de velocidad. La velocidad de sincronismo

fijada para el motor asíncrono es de 2750 rpm. El funcionamiento del sistema se fija en una velocidad del eje igual a 2577 rpm en régimen permanente, es decir, el valor de F sería igual a 0.859 y por lo tanto, el par resistente impuesto en el servomecanismo es del 30.065 %.

A continuación se muestran las distintas gráficas de tensión e intensidad obtenidas bajo las condiciones descritas anteriormente. Empezaremos por analizar las curvas de intensidad del motor asíncrono y el servomecanismo. En la primera de ellas, intensidad de línea del motor asíncrono (Figura 87), se ha representado adicionalmente la onda correspondiente al armónico fundamental (color rojo), para una mejor visualización de los datos de la onda de intensidad. Sus características son un valor pico igual a 1,2 A y un periodo de 22,5 ms, equivalente a una frecuencia igual a 44,5 Hz, el cual si lo comparamos con la frecuencia de la velocidad de giro del motor asíncrono (45,83 Hz), comprobamos la similitud de ambos valores.

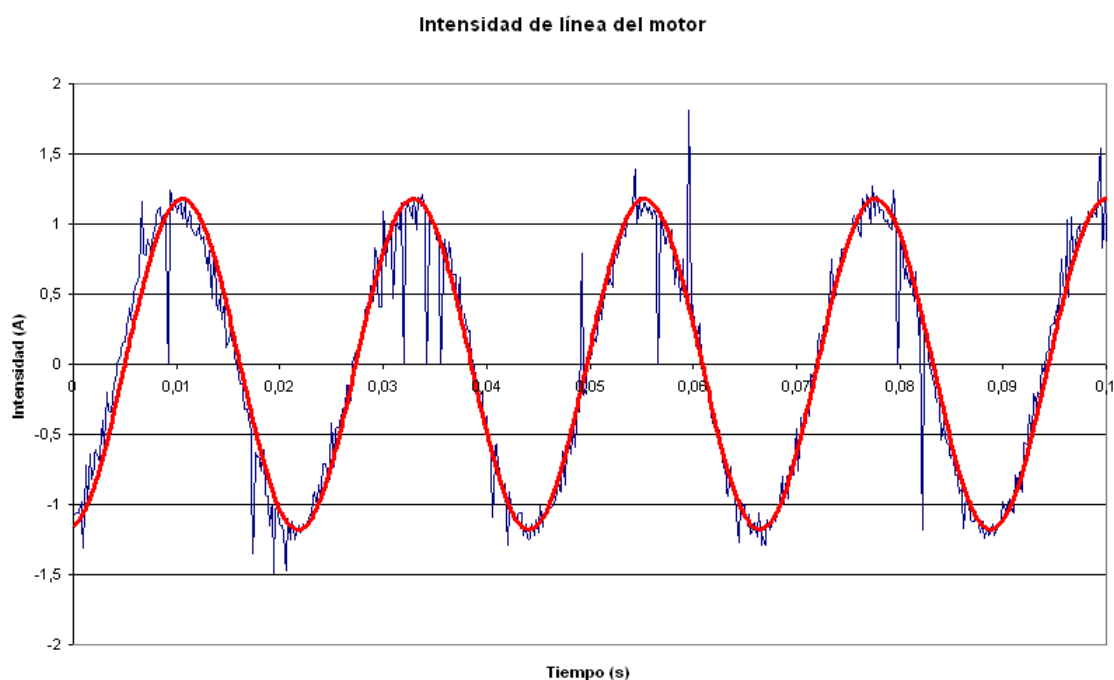


Figura 87. Intensidad de línea del motor asíncrono

El mismo planteamiento realizaremos para la intensidad de línea del servo (Figura 88), cuya forma sinusoidal presenta un mayor ruido, por lo que diferirá mas si la comparamos con su armónico fundamental. Un análisis de dicha onda y en particular, del armónico fundamental, establece unas características de intensidad de pico igual a 1,03 A y frecuencia igual a 123,46 Hz. El valor de pico del armónico fundamental, si es comparado con el que se obtendría por medio de la relación entre par resistente e intensidad eficaz, teniendo en cuenta que:

$$T = K_t \cdot I \qquad I = \frac{I_p}{\sqrt{2}}$$

Conocido el par impuesto, igual al 30,065% respecto al par máximo, es decir, 1,173 Nm, y K_t igual a 1,9 Nm/Arms se obtiene a su vez un valor de intensidad eficaz (I) igual a 0,73 A, muy cercano al valor del ensayo, el cual es igual a 0,728 A

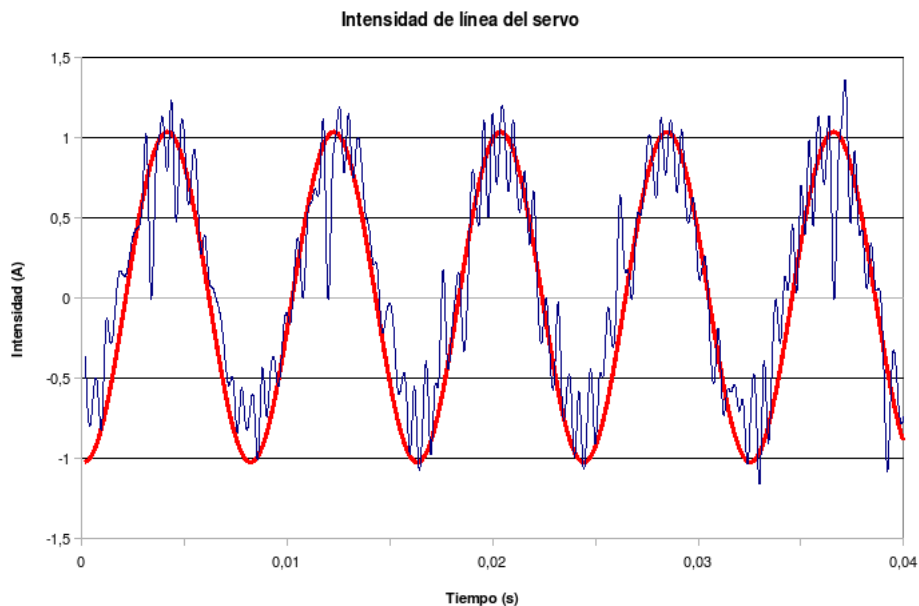


Figura 88. Intensidad de línea del servomecanismo

En cuanto al análisis de los voltajes de línea en el motor (Figura 89) y en el servomecanismo (Figura 89), se observa una forma de onda en el caso del motor igual a la característica de PWM explicada en apartados anteriores. Las frecuencias de las ondas de voltaje serán iguales a las que poseen las respectivas intensidades

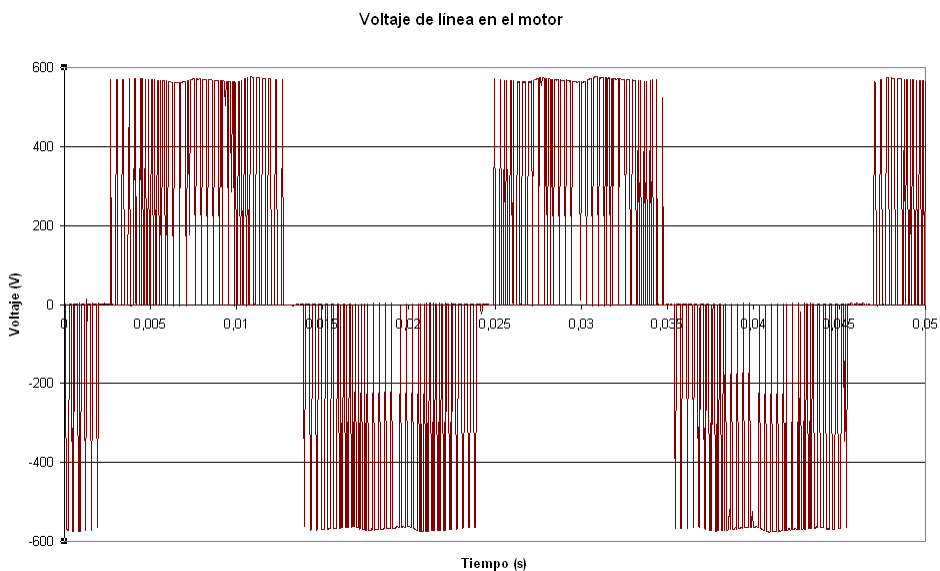


Figura 89. Voltaje de línea del motor asíncrono

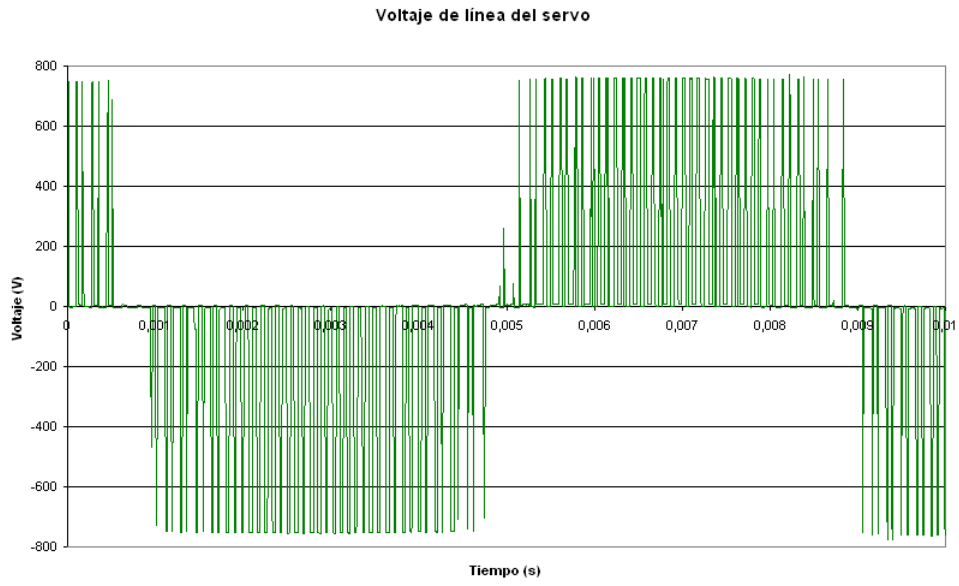


Figura 90. Voltaje de línea del servomecanismo

7 Conclusiones

En la elaboración de este proyecto, se ha planteado en un inicio los distintos componentes capaces de simular pares resistentes. Tras un análisis de cada uno de ellos, se ha proseguido con la elaboración de un prototipo de sistema cuya finalidad era la consecución de pares resistentes variables y configurables por el propio usuario, a partir de una serie de dispositivos habituales en entornos industriales (variadores de frecuencia, servos, computadores, ...).

Debido al uso de material disponible en el laboratorio del Departamento de Ingeniería Eléctrica, donde se ha desarrollado las experimentaciones, permite que otros usuarios puedan reproducir las experimentaciones aquí diseñadas y, además, gracias a que el sistema esta dividido en módulos capaces de ser sustituidos por otros de distintas características, vease motores asíncronos, servos, variadores de frecuencia, etc, dicho usuario podría trabajar con equipos de prestaciones e incluso rendimientos mejores.

En cuanto al lenguaje de programación desarrollado para el control del sistema, mejoras en el mismo pueden realizarse, ya que la complejidad del mismo vendrá dado por un aumento de los parámetros a controlar, siempre y cuando la cantidad de información a procesar no supere los límites del propio sistema de control, ya sea debido a la comunicación (límites de recepción y transferencia del conmutador de red) o al procesamiento de datos (límites por la CPU de la computadora).

También se ha desarrollado sistemas de control del servo de forma que actue como un motor. Dicho control se realiza de dos formas, fijando la velocidad o el par motor desarrollado. Esta aplicación resulta interesante pues posibilita la experimentación con el servo con finalidad educacional en el manejo de dicho equipo.

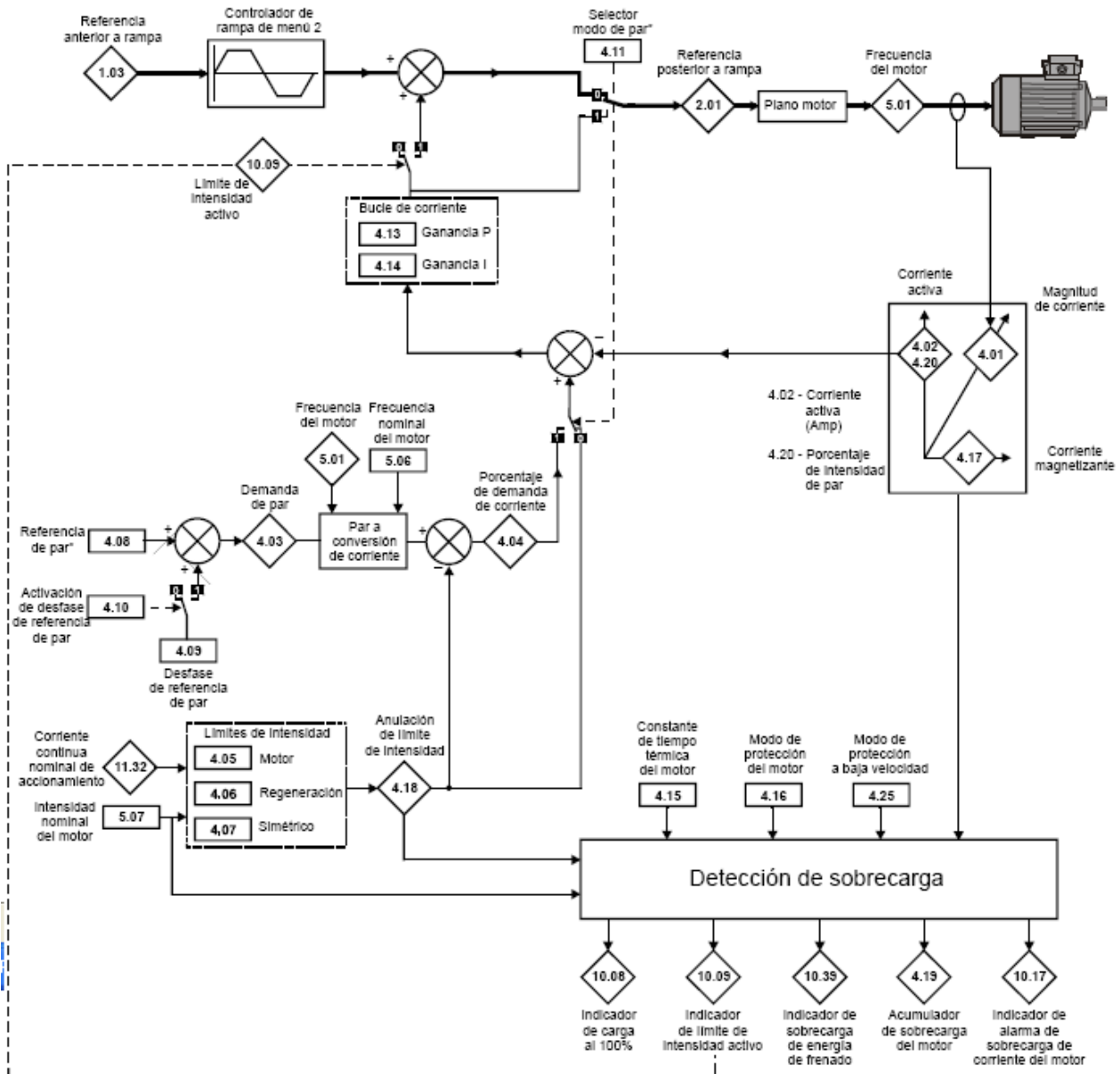
Además, el sistema en su conjunto no solo puede ser mejorado mediante la sustitución de los distintos módulos, sino que gracias a la red de comunicaciones basado en el PC, el conmutador de red y el módulo SM-Ethernet, dicho sistema podría aplicarse a uno basado en varios variadores de frecuencia con sus correspondientes módulos SM-Ethernet, controlados todos ellos a través del conmutador de red y un único PC. Cabe destacar adicionalmente, que entre las características adicionales, el sistema puede ser controlado remotamente a través de internet o gracias a una red local, la cual podría estar distribuida en un mismo laboratorio.

Finalmente, indicar que la captura de datos se ha desarrollado gracias al software

desarrollado por el Departamento de Ingeniería Eléctrica de la Universidad Politécnica de Cartagena, para los valores de tensión e intensidad de los dispositivos utilizados, con el fin de analizar en detalle el comportamiento del conjunto del sistema.

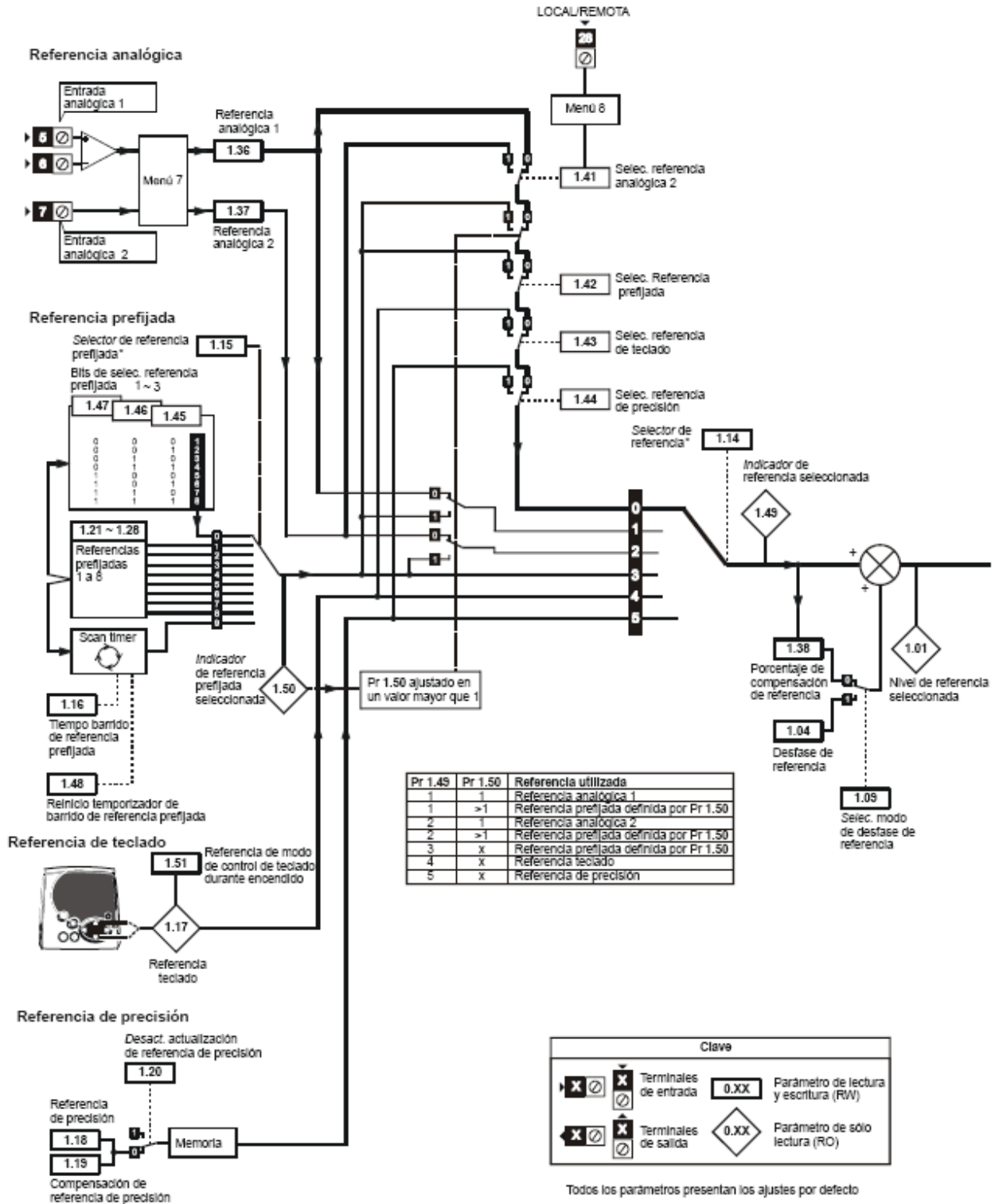
8.2 ANEXO 2: Esquema de control por parámetros del variador de frecuencia SP1401

8.2.1 Diagrama de control de velocidad



Todos los parámetros presentan los ajustes por defecto

8.2.2 Diagrama de control de par



8.3 ANEXO 3: Código fuente de las páginas web de control

8.3.1 Página index.htm

```
<!-- Página encargada de introducir en el variador de frecuencia que controla el
servomecanismo aquellos parametros necesarios para una configuración rápida del mismo
-->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" >
<script src="smethernet.htm"></script>
<link href="style.css" rel="stylesheet" type="text/css">
</head>

<body onLoad="SMData.update();" marginheight="10" marginwidth="10" leftmargin="10"
topmargin="10">


<br><br><br><br>

<table align="center" cellpadding="15" cellspacing="1" border="1">
<td>

<h1 align="center">Parámetros del servo</h1>

<br>
```

```
<div align="center"><b>Revise la placa de características del servo a utilizar</b></div>
```

```
<br>
```

```
<div align="center"><input align=center type="button" value="Introducir parametros"
onClick="param()"></div>
```

```
</td>
```

```
</table>
```

```
<script>
```

```
/* La función param se ejecuta tras haber sido pulsado el botón "Introducir parámetros"
```

```
En ella damos valores a la intensidad nominal, el numero de polos y la velocidad máxima
del servomecanismo */
```

```
function param()
```

```
{
```

```
int_nom=Number(prompt("Intensidad nominal:", ""));
```

```
n_pol=Number(prompt("Número de pares de polos:", ""));
```

```
vel_max=Number(prompt("Velocidad máxima:", ""));
```

```
if(confirm("Compruebe que son correctos\nIntensidad nominal: "+int_nom+" A\nNúmero
de pares de polos: "+n_pol+"\nVelocidad maxima: "+vel_max+" rpm"))
```

```
{
```

```
envio(int_nom,n_pol,vel_max);
```

```
}
```

```
else
```

```
{
```

```
document.location.href="";
```

```
}
```

```
}
```

```
function envio(int_nom,n_pol,vel_max)
```

```
{
```

```
    SMDData.setParameter(0.46,int_nom);
```

```
    //Envia los valores de la intensidad nominal al variador de frecuencia
```

```
    SMDData.setParameter(0.42,n_pol);
```

```
    //Envia los valores del numero de pares de polos al variador de frecuencia
```

```
    SMDData.setParameter(0.02,vel_max);
```

```
    //Envia los valores de la velocidad maxima al variador de frecuencia
```

```
    document.location.href="inicial.htm";
```

```
    //Redirecciona a la página inicial.htm
```

```
}
```

```
</script>
```

```
<br><br>
```

```
<div align=right><A HREF="inicial.htm">Pasar al men&uacute;</A></div>
```

```
</body>
```

```
</html>
```

8.3.2 Página inicial.htm

<!-- Página donde aparece el menú que da acceso al control del servomecanismo:

**Actuando como motor*

**Actuando como generador*

-->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<script src="smethernet.htm"></script>

<link href="style.css" rel="stylesheet" type="text/css">

</head>

<body>

**

*

*

<table align="center" cellpadding="1" cellspacing="1" border="1" width="70%" height="20%">

<td>

<div align="center"><h1>CONFIGURACION DEL SERVO</h1></div>

<table border width="60%" height="10%" cellspacing=4 align=center>

<TR>

```
        <td align="center"><A HREF="servo_motor.htm">Modo
MOTOR</A></Td>
```

```
        <td align="center"><A HREF="servo_gener.htm">Modo
GENERADOR</A></td>
```

```
    </TR>
```

```
</table>
```

```
<br>
```

```
</td>
```

```
</table>
```

```
<br><br>
```

```
<TD align=left><A HREF="index.htm">Reintroducir los par&acute;metros del
servo</A></TD>
```

```
</body>
```

```
</html>
```

8.3.3 Página servo_motor.htm

<!-- Página donde aparece el menú que da acceso al control del servomecanismo en modo motor:

**Controlando la velocidad*

**Controlando el par generado*

-->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<script src="smethernet.htm"></script>

<link href="style.css" rel="stylesheet" type="text/css">

</head>

<BODY>

**

*

*

<table cellpadding="1" cellspacing="1" border="1" align="center" valign="middle" width="70%" height="20%">

<td>

<div align="center"><h1>CONTROL DEL SERVO</h1></div>

<TABLE width="80%" height="10%" cellspacing=4 border="1" align=center>


```
<TR>
    <TD align=center><A HREF="servo_par.htm"><b>Control de
par</b></A></TD>
    <TD align=center><A HREF="servo_veloc.htm"><b>Control de
velocidad</b></A></TD>
</TR>

</TABLE>

<br>

</td>

</table>

<br><br>

<TR align=center><A HREF="inicial.htm">Volver a la p&aacute;gina anterior</A></TR>

</body>
</html>
```

8.3.4 Página servo_par.htm

```
<!-- Página para controlar el par de la máquina hasta llegar a una velocidad determinada -->
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

```
<script src="smethernet.htm"></script>
```

```
<link href="style.css" rel="stylesheet" type="text/css">
```

```
</head>
```

```
<BODY onLoad="entrada();" onUnload="salida();">
```

```

```

```
<br><br><br>
```

```
<h1><div align="center">CONTROL DEL SERVO POR PAR</div></h1>
```

```
<br>
```

```
<TABLE border width="60%" height="20%" cellspacing=4 align=center cellpadding="3">
```

```
<TR>
```

```
<TD align=center><b>Atención: <br><br><u>Al llegar al 80% de la velocidad máxima, el par sera cero</u></b></TD>
```

```
</TR>
```

```
<TR>
```

```
<TD align=center><input type="button" value="Introducir par motor" onClick="par()></TD>
```

</TR>

</TABLE>

<TABLE align=center border width="40%" height="10%" cellspacing=2>

<TR>

<TD align=center>Velocidad del eje</TD>

<TD><div align=center id="text1"></div></TD>

</TR>

<script language="javascript">

new TextDisplay("text1", "3.2", {textVersion:true});

</script>

</TABLE>

<TR align=center>Volver a la página anterior</TR>

<!--Encontramos la función par(), la cual envia el valor del par motor

introducido al variador de frecuencia, en caso de no haber escrito nada, volverá a pedir un valor.

-->

<script>

/*función registro: definida en el fichero smethernet.htm*/

```
new registro("text1", "3.2", {textVersion:true});
```

```
control=setInterval("cont()",500);
```

```
/* función cont: función llamada tras ejecutarse la actualización "control" por  
medio de la función setInterval de la línea de código anterior.  
Realiza la lectura del parametro 3.2 el cual define la velocidad actual del  
servomecanismo. En caso de superar las 2400 rpm, se frena al considerar un par  
motor (parámetro 3.8) igual a 0.  
*/
```

```
function cont()  
{  
  if (SMDData.parameters["3.2"]>2400)  
  {  
    SMDData.setParameter(4.08,0);  
  }  
}
```

```
/*función entrada: se activa tras la carga de la página web. En ella encontramos:  
- Un mensaje de alerta al usuario para un correcto funcionamiento del sistema  
- Una actualización del registro de parametros del variador de frecuencia  
- Modificación de parametros del variador de frecuencia  
*/
```

```
function entrada()  
{  
  alert("Antes de continuar, desacople el servo del motor asincrono");  
  SMDData.update();  
}
```

```

SMDData.setParameter(4.08,0);
//pone a cero el parametro que controla el par
SMDData.setParameter(4.11,1);
//activa el parametro que establece el control
}

/* función par: función que introduce el valor del parametro que define el par motor
del servomecanismo.
*/

function par()
{
par_motor=Number(prompt("Par motor:", ""));

if (par_motor==0)
{
if (!(confirm("El valor introducido es igual a 0. Pulse aceptar si es
correcto")))
{
alert("Por favor, introduzca un valor");
par();
}
}

SMDData.setParameter(4.08,par_motor);

/* Introduce el valor par_motor como valor de referencia del par que
ha de generar la máquina y lo guarda en el registro
*/

SMDData.setParameter(1.14,3);

```

```
}
```

```
function salida()
```

```
{
```

```
SMDData.setParameter(4.08,0);
```

```
//pone a cero el parametro que controla el par
```

```
SMDData.setParameter(4.11,1);
```

```
//desactiva el parametro que establece el control
```

```
SMDData.setParameter(1.21,0);
```

```
SMDData.setParameter(1.14,4);
```

```
}
```

```
</script>
```

```
</BODY>
```

```
</HTML>
```

8.3.5 Página servo_veloc.htm

```
<!-- Pagina para controlar la velocidad de la máquina-->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script src="smethernet.htm"></script>
<link href="style.css" rel="stylesheet" type="text/css">
</head>

<BODY onLoad="entrada();" onUnload="salida();">


<br><br><br>

<h1><div align="center">CONTROL DEL SERVO POR VELOCIDAD</div></h1>
<br>

<TABLE border width="60%" height="20%" cellspacing=4 align=center cellpadding="3">

    <TR>

        <TD align=center><b>Atenci&oacute;n:<br><br> <u>Introduciendo valores
superiores a la velocidad m&aacute;xima, se llegara como limite a la velocidad m&aacute;xima
establecida</u></b></TD>

    </TR>

</TR>

<TR>
```

```
<TD align=center><input type="button" value="Introducir velocidad deseada"
onClick="veloc()"></TD>
```

```
</TR>
```

```
</TABLE>
```

```
<br><br>
```

```
<TABLE align=center border width="40%" height="10%" cellspacing=2>
```

```
<TR>
```

```
<TD align=center>Velocidad del eje</TD>
```

```
<TD><div align=center id="text1"></div></TD>
```

```
</TR>
```

```
<script>
```

```
new TextDisplay("text1", "3.2", {textVersion:true});
```

```
</script>
```

```
</TABLE>
```

```
<br><br>
```

```
<div align=center><A HREF="servo_motor.htm">Volver a la p&acute;gina
anterior</A></div>
```


<script>

*/*función entrada: se activa tras la carga de la página web. En ella encontramos:*

- Un mensaje de alerta al usuario para un correcto funcionamiento del sistema*
- Una actualización del registro de parametros del variador de frecuencia*
- Modificación de parametros del variador de frecuencia*

**/*

function entrada()

{

alert("Antes de continuar, desacople el servo del motor asíncrono");

SMDData.update();

SMDData.setParameter(1.21,0);

//pone a cero el parametro que controla la velocidad

SMDData.setParameter(1.14,3);

//activa el parametro que establece el control por la variación del parametro 1.21

};

/ función veloc: se activa tras pulsar el botón "Introducir velocidad deseada".*

En ella se limita el valor a introducir en un rango de 0 a 3000 rpm

**/*

function veloc()

{

vel=Number(prompt("Velocidad:", ""));

if (vel<0)

{

alert("La velocidad introducida es negativa");

alert("Nueva velocidad introducida: 0 rpm");

```

        vel=0
    };
    if (vel>3000)
    {
        alert("La velocidad introducida es superior al valor maximo");
        alert("Nueva velocidad introducida: 3000 rpm");
        vel=3000;
    };

    SMDData.setParameter(1.21,vel);    //fija el valor de la velocidad
    SMDData.setParameter(1.14,3);    //establece control por velocidad
    prefijada en el parametro 1.21
    SMDData.setParameter(4.11,0);    //establece control de la velocidad

}

function salida()
{
    SMDData.setParameter(1.21,0);
    //pone a cero el parámetro que controla la velocidad
    SMDData.setParameter(1.14,4);
    //desactiva el parámetro que establece el control devolviendolo al teclado
}

</script>

</BODY>

</HTML>

```

8.3.6 Página servo-gener.htm

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script src="smethernet.htm"></script>
<link href="style.css" rel="stylesheet" type="text/css">
</head>

<body onLoad="carga()"
onUnLoad="SMDData.setParameter(4.8,0);SMDData.setParameter(4.11,0);">

<br>

<br><br>

<div align="center"><h1>SERVO COMO PAR RESISTENTE</h1></div>

<TABLE align=center border width="90%" height="10%" cellspacing=2>

<TD align=center>Par resistente<br> $T_{R}(\% \text{ Mximo})=A+B \cdot F$ 
 $+C \cdot F^2$ </TD>

<TD align=center> $F=\text{velocidad del eje}/3000 \text{ rpm}$ </TD>

<!-- Definimos el objeto button (boton) junto con el evento onClick (presin del objeto
button) para definir las acciones que queremos que lleve a cabo el sistema y definidas
por las funciones que encontramos en cdigo JavaScript -->

<TD align=center><input type="button" value="Introducir parametros A B y C"
```

```
onClick="coef_curva()"></TD>
<TD align=center><input type="button" value="A=0 B=0 C=0"
onClick="A=0;B=0;C=0;"></TD>
```

```
</TABLE>
```

```
<script>
```

```
var A,B,C;
```

```
/* La función carga se encargará de transmitir los valores para la inicialización
del ensayo, como son informar al usuario el acoplamiento de la máquina asíncrona
con el servomecanismo, dar el valor cero al par resistente de manera inicial y
controlar el motor por medio de su par, el cual convertiremos en par resistente
del sistema */
```

```
function carga()
```

```
{
alert("Antes de continuar, acople el eje del servo con el del motor asíncrono");
SMDData.update();
SMDData.setParameter(4.8,0);
SMDData.setParameter(4.11,1);
}
```

```
/* Reiniciamos las variables A, B, C */
```

```
function restaurar_parametros()
```

```
{
SMDData.setParameter(4.8,0);
SMDData.setParameter(4.11,0);
clearInterval("intervalo");
}
```

```

function coef_curva()
{
A=Number(prompt("Valor de A:",""));
B=Number(prompt("Valor de B:",""));
C=Number(prompt("Valor de C:",""));

Total=(A)+(B)+(C);
alert("Los valores introducidos son A="+A+" B="+B+" C="+C);
discr=B*B-4*C*(A-35);
if (discr<0)
{
alert("Valores no validos");
document.location.href=""; // Reiniciamos la página
return true;
}
else
{
F_1=parseInt(1000*(-B+Math.sqrt(discr))/(2*C))/1000; // escogemos tres
cifras significativas

/* Se informa al usuario que si durante un instante se supera el límite de par
resistente del 35%, este quedará fijado para dicho valor.
En caso contrario, alerta del máximo par resistente posible que puede
darse lugar
a una velocidad igual a 3000 rpm */

if (Total>35)
{

alert("Para velocidades mayores a "+3000*F_1+" rpm el par
resistente sera igual al par maximo del servo");
}
}
}

```

```

        }
    else
    {
        alert("No se llega a utilizar el par resistente máximo en el
rango de valores de la velocidad.\nComo máximo " +Total+" %");
    }

};

}

```

/ Utilizamos la función setInterval para ejecutar la función actualizar de manera repetida en un cierto intervalo de tiempo, en este caso, cada 300 ms*/*

```
intervalo=setInterval("actualizar()",300);
```

/ En la función actualizar encontramos el feed-back o realimentación del sistema, por el cual, se hacen lecturas periódicas de la velocidad y de manera instantánea es calculado el par resistente a ejecutar por parte del servo-mecanismo */*

```
function actualizar()
```

```

{
    vel=SMDData.parameters["3.2"];
    F=Math.abs(vel/3000);
    par_res=-(A+B*F+C*F*F);

```

/ El par resistente no puede sobrepasar del 35% debido al motor utilizado en este proyecto, ya que valores superiores harían entrar en una desaceleración del mismo puesto que el motor asíncrono no desarrollará un par motor capaz de igualar a dicho valor */*

```

if (par_res<-35) {par_res=-35;};
if (par_res>35) {par_res=35;};
if (vel<0) {par_res=0-par_res;};
if ((vel<1000)&&(vel>-1000)) {par_res=0;};

par_res=parseInt(par_res*100)/100; //redondeamos a dos decimales

/*      Fijamos en el variador de frecuencia el valor del par
        resistente a aplicar por el servo */

```

```

SMDData.setParameter(4.8,par_res);

```

Velocidad

```

/*      Comunicamos al objeto "punto" donde debe situarse en el gráfico Par-
        para la representación gráfica sobre el mismo de manera instantánea */

```

```

punto.style.top=0-2*Math.abs(par_res);
punto.style.left=8+Math.abs(vel/20);
}

```

```

</script>

```

```

<br>

```

```

<!-- Definimos los objetos donde estarán situados los valores de la velocidad del eje
        y el par resistente, mediante la función TextDisplay, la cual encontraremos en el
        archivo smethernet.htm -->

```

```

<TABLE align=center border width="30%" height="10%" cellspacing=2>

```

```

<TR><TD align=center>Velocidad del eje</TD>

```

```

<TD><div align=center id="text1"></div></TD></TR>

```

```
<TR><TD align=center>Par resistente (% máximo)</TD>
<TD><div align=center id="text2"></div></TD></TR>
```

```
<script language="javascript">
    new TextDisplay("text1", "3.2", {textVersion:true});
    new TextDisplay("text2", "4.8", {textVersion:true});
</script>
```

```
</TABLE>
```

```
<br>
```

```
<table align="center" width="800" height="100" border="5" cellpadding="10"
cellspacing="0">
    <tr>
        <td></td>
        <td align="center"><div>Velocidad (rpm) VS Tiempo</div></td>
        <td></td>
        <td align="center"><div>Par resistente</div></td>
        <td></td>
        <td width="800" height="80" align="center"><div>Par (% máximo)
<br>VS<br> Velocidad (rpm)</div></td>
    </tr>
    <tr>
        <td width="30"></td>
        <!-- Graph Background -->
        <td width="330" align="center"
background="images/backgroundgraph.gif">
            <div id="graph1" style="width:310px; height:150px;"></div>
```



```

<script language="javascript">
    new TimeGraph("graph1", "3.2", {maxValue: 2800, rate:1, fill:true,
color:"#999933", labelWidth:60, label0:"0.0", label25:"750", label50:"1500", label75:"2250",
label100:"3000", marker:"#FFFFFF", ticks:8});
</script>

```

```
</td>
```

```
<td>
```

```
<td align="left" valign="middle">
```

```
<div id="meter1" style="width:125px; height:125px;"></div>
```

```
<script language="JavaScript">
```

```

    new Meter("meter1", "4.8", {maxValue: 100, label0:"0.0", label25:"25.0",
label50:"50.0", label75:"75.0", label100:"100.0", units:"%", maxIndicator:90, minIndicator:0});

```

```
</script>
```

```
</td>
```

```
<td></td>
```

```
<td>
```

```

<!-- Dibujamos el punto definido por los parámetros de velocidad y par resistente
en tiempo real -->

```

```
<div style="position:relative; left:-10;top:11;visibility:visible;">
```

```
</div>
```

```
<div style="position:relative; left:9;top:10;visibility:visible;">
```

```
</div>
```

```
<div id="punto" style="position:relative; left:8;top:-1;visibility:visible;">  
</div>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
<br>
```

```
<div align=left><A HREF="inicial.htm">Volver a la pagina anterior</A></div>
```

```
</body>
```

```
</html>
```

8.3.7 Librería smethernet.htm

*/** SMDData - Una variable clase para manejar la comunicación con el módulo.*

** SMDData.update() function: Utilizado para iniciar las comunicaciones con el modulo. e.g. in onload event*

***/*

function SMDData() {;
añadiremos

/ define la funcion SMDData al cual
atributos posteriormente */*

SMDData.pollRate = 500;
medido en milisegundos

// tiempo de actualización de parametros

SMDData.parameters = new Object();

// creación del objeto SMDATA.parameters

SMDData.elements = new Array();

// definición de SMDData.elements como array

SMDData.parameterslist = "";

SMDData.indicator="none";

*/**

SMDData.registerElement: pide un parametro del modulo para ser monitoreado

Variables :

param - el parametro

elem - el elemento a registrar cuyo valor es el leído del modulo

*textVersion - booleano que especifica si el valor dado al elemento es el
correspondiente a la version del texto. Por defecto es falso*

*/

SMDData.registerElement = function(param, elem, textVersion) // llamada a la función de lectura

{

this.parameters[param]=null;

var sep="";

this.parameterslist="";

for (var p in this.parameters)

{

this.parameterslist+=sep+p;

sep=" _";

}

if (textVersion) //dependiendo del booleano escribira de distinta forma el parametro

{

p=param+"t";

}

else

{

p=param;

}

this.elements.push({parameter: p, element: elem});

}

*/**

SMDData.registerIndicator - Parpadea cada vez que un elemento ha sido actualizado

Solo puede haber uno por página

**/*

SMDData.registerIndicator = function(elem) {

this.indicator = elem;

}

*/**

SMDData.update - Llamado cuando la pagina empieza a pedir actualizaciones al modulo

Preferible llamarlo en los eventos de carga de la pagina web

**/*

SMDData.update = function()

{

setInterval("SMDData.getData()", this.pollRate);

}

*/**

SMDData.getData - Actualización de los valores provenientes del modulo

*Convierte un archivo XML del modulo y lo llama SMDData.xmlLoaded cuando
esta listo de ser actualizado*

**/*

SMDData.getData = function()

{

//CODIGO DE LLAMADA - var theurl = "getvalue.php?parameters="+this.parameterslist;

//función de llamada al modulo para obtener un parametro

var theurl = "/US/" + this.parameterslist + "/dynamic/readparval.xml";

```

if (document.implementation && document.implementation.createDocument)
{
var xmlDoc = document.implementation.createDocument("", "", null);
xmlDoc.onload = function() { SMDData.xmlLoaded(xmlDoc); }
xmlDoc.load(theurl)
}

```

```

else if (window.ActiveXObject)

```

```

{
var xmlDoc = new ActiveXObject("Microsoft.XMLDOM");

xmlDoc.async=false;
xmlDoc.load(theurl);
SMDData.xmlLoaded(xmlDoc);
}

```

```

else

```

```

{
//El navegador no soporta el metodo XML
alert("Lo siento, su navegador no es soportado");
}

```

```

}

```

```

/*

```

SMDData.xmlLoaded - Llamado cuando un nuevo XML esta disponible

Actualiza cada uno de los elementos registrado mediante la llamada de la función de actualización con el valor apropiado

```

*/

```

```

SMDData.xmlLoaded = function(xmlDoc)

```

```

{

```

//Actualiza cada función

```
var pElements = xmlDoc.getElementsByTagName('parameter')
```

```
if (pElements.length==0) return;
```

```
for (var i=0; i<pElements.length; i++)
```

```
{
```

```
var parn = pElements[i].getAttribute("name");
```

```
if (pElements[i].getAttribute("text")== "Error - No Parameter")
```

```
{
```

```
alert("Error - parameter " + parn + " doesn't exist");
```

```
}
```

//Almacena parametros numericos, convirtiendo enteros + dp a valores decimales

```
this.parameters[parn] = pElements[i].getAttribute("value")/Math.pow(10,  
pElements[i].getAttribute("dp"));
```

//Almacena parametros con texto

```
this.parameters[parn + "t"] = pElements[i].getAttribute("text");
```

```
}
```

```
for (var j=0; j<this.elements.length; j++)
```

```
{
```

```
this.elements[j].element.update(this.parameters[this.elements[j].parameter]);
```

```
}
```

//Si un indicador es usado, se ilumina para indicar que la actualización se ha llevado a cabo

```

    if (this.indicator != "none") SMDData.indicator.update(true);

}

/*
SMDData.setParameter - Llamado por los elementos para actualizar un valor de un parametro del
modulo.

Parameters : parameter - El parametro a actualizar. Ejemplo:
                10.1, 10.1b4 (bit 4 of 10.1)
                value - el valor a establecer
                Por compatibilidad con el modulo 1.1 es enviado como 11dp1

*/

SMDData.setParameter = function(parameter, value)
{

    //CODIGO DEL TEXTO
    //alert("Parameter " + parameter + " set to " + value);

    //CODIGO PARA ESTABLECER VALORES EN EL MÓDULO
    //convierte el parametro para contar los numeros decimales
    //e.g. 10.1 is 101dp1

    value="" + value;           //el valor se convierte en una cadena de caracteres

    if (value.indexOf(".")!=-1)
    {
        var dp=value.length-value.indexOf(".");
        value=value * Math.pow(10, dp);           //el valor se convierte en

```


un numero

```
value=value + "dp" + dp;
```

//el valor se convierte en una

cadena

```
}
```

```
else value=value+"dp0";
```

//escribe el valor del parametro al modulo

```
var fn = "/US/" + parameter + "=" + value + "/dynamic/writeparval.xml";
```

```
if (document.implementation && document.implementation.createDocument)
```

```
{
```

```
var xmlDoc = document.implementation.createDocument("", "", null);
```

```
xmlDoc.load(fn);
```

```
}
```

```
else if (window.ActiveXObject)
```

```
{
```

```
var xmlDoc = new ActiveXObject("Microsoft.XMLDOM");
```

```
xmlDoc.load(fn);
```

```
}
```

```
}
```

```
/******
```

```
****
```

FUNCIONES PARA LA REPRESENTACIÓN DE DISPLAYS EN PANTALLA

El compilador debera de hacer una llamada a la función y registrar el elemento creado mediante la función SMDData.registerelement. La actualización se llevará a cabo en cada ciclo de reloj definido por la variable pollrate.

```
*****  
*****/
```

```
/** TextDisplay - Función de representación de un parametro en pantalla
```

```
Variables:    divid - Nombre de la identificación del elemento creado  
             parameter - Parametro a representar  
             opts -        Opciones de representación en el display  
  
*/
```

```
function TextDisplay(divid, parameter, opts) {  
    if (!opts) opts={};  
    this.container = document.getElementById(divid);  
    this.textnode = document.createTextNode("-");  
    this.container.appendChild(this.textnode);  
  
    if (opts.textVersion && opts.textVersion===true) {  
        SMData.registerElement(parameter, this, true);  
    } else {  
        SMData.registerElement(parameter, this);  
    }  
}
```

```
/* Función de actualización del elemento TextDisplay */
```

```
TextDisplay.prototype.update = function(value) {  
    var newNode = document.createTextNode(value);  
    this.container.replaceChild(newNode, this.textnode);  
    this.textnode = newNode;
```

```
}
```

```
/** registro - Función que crea un registro del parametro a actualizar en la página */
```

```
function registro(divid,parameter, opts) {  
    if (!opts) opts={};  
    if (opts.textVersion && opts.textVersion===true) {  
        SMDData.registerElement(parameter, this, true);  
    } else {  
        SMDData.registerElement(parameter, this);  
    }  
  
}
```

```
}
```

```
/*Funcion de actualizacion del elemento registro */
```

```
registro.prototype.update = function(value)  
    {}
```

```
/** Bar - Función que crea una barra a partir de la lectura de un parametro */
```

```
function Bar(divid, parameter, opts) {  
    var container = document.getElementById(divid);  
    var bar=document.createElement('div');  
  
    container.style.position = "relative";  
    bar.style.position = "absolute";  
    bar.style.width = container.style.width;  
    bar.style.height = 1;
```

```

bar.style.fontSize="1px";

if(!opts) opts= {};
if (typeof opts.maxValue != "undefined") this.maxValue=opts.maxValue;
if (typeof opts.minValue != "undefined") this.minValue = opts.minValue;
    else this.minValue = 0;

if (opts.barImage) bar.style.backgroundImage="url(" + opts.barImage + ")"; //carga de la
imagen que hara de barra

    this.containerHeight = parseInt(container.style.height); //obtiene el valor entero de
container.style.height

    this.barInc = this.containerHeight / (this.maxValue - this.minValue);

this.container = container;
this.bar = bar;
container.appendChild(bar);

//dibuja las etiquetas

if(!opts.labelHeight) opts.labelHeight=10;
if(!opts.labelWidth) opts.labelWidth=20;

var lbl = document.createElement('div');
lbl.style.position = "absolute";
lbl.style.height=opts.labelHeight + "px";
lbl.style.width=opts.labelWidth + "px";
lbl.style.fontSize=opts.labelHeight + "px";
lbl.style.fontFamily="Arial, Helvetica, sans-serif";
lbl.style.left= -opts.labelWidth + "px";
var lbly;

```

```

for (var lblpos=0; lblpos<=100; lblpos+=25) {
    if (opts['label' + lblpos]) {
        var l=lbl.cloneNode(true);
        lbly = Math.ceil(this.containerHeight - (lblpos/100 * this.containerHeight +
opts.labelHeight/2));

        l.style.top = lbly + "px";
        l.innerHTML = opts['label' + lblpos];
        container.appendChild(l);
    }
}

```

//dibuja las marcas numéricas

```

var point = document.createElement('div');
point.style.position="absolute";
point.style.fontSize="1px";

```

```

if (typeof opts.ticks != "undefined" && opts.ticks > 0) {
    var tickinterval = (this.maxValue - this.minValue) / (opts.ticks - 1);
    for (var t=0; t<opts.ticks; t++) {

        var m = point.cloneNode(true);
        m.style.background="#242424";
        m.style.height="1px";
        m.style.width="3px";
        m.style.left = "-12px";
        m.style.top = Math.ceil(this.barInc * t * tickinterval) + "px";
        container.appendChild(m);
    }
}

```

```
}
```

```
}
```

```
SMDData.registerElement(parameter, this);
```

```
}
```

```
Bar.prototype.update = function(value) {  
    if (value < this.minValue) value = this.minValue;  
    if (value > this.maxValue) value = this.maxValue;  
    var containerHeight = this.container.height;  
    var barHeight = this.barInc * (value - this.minValue);  
    this.bar.style.top = (this.containerHeight - barHeight) + "px";  
    this.bar.style.height = barHeight + "px";  
}
```

```
/** Slider - Función que crea un deslizador para seleccionar el valor de un parámetro, subiéndolo  
o bajándolo*/
```

```
function Slider(divid, parameter, opts) {  
    if (!opts) opts = {};  
    this.parameter = parameter;  
    this.container = document.getElementById(divid);  
    this.container.style.position = "relative";  
    this.slider = document.createElement('div');  
    this.slider.style.position = "absolute";  
  
    if (opts.width) this.slider.style.width = opts.width + "px";  
        else this.slider.style.width = "20px";  
    if (opts.height) this.slider.style.height = opts.height + "px";  
        else this.slider.style.height = "15px";
```

```

if (opts.image) this.slider.style.backgroundImage="url(" + opts.image + ")";
    else this.slider.style.background="black";
if (opts.minValue) this.minValue = opts.minValue;
    else this.minValue=0;
if (opts.maxValue) this.maxValue = opts.maxValue;
    else this.maxValue = 100;
if (opts.accuracy) this.accuracy = opts.accuracy;
    else this.accuracy=0;
this.currentvalue = this.minValue;
if (opts.submit) {
    var cont = this;
    var sub = document.getElementById(opts.submit);
    sub.onclick = function() {
        SMDData.setParameter(parameter, cont.currentValue);
        return false;
    }
}

this.slider.minY=-parseInt(this.slider.style.height)/2
this.slider.maxY=parseInt(this.container.style.height)-parseInt(this.slider.style.height)/2;
this.slider.controller = this;
this.valueStep = (this.maxValue - this.minValue)/ (this.slider.maxY-this.slider.minY);
this.slider.style.top=(parseInt(this.container.style.height)+this.slider.minY)+"px";
this.slider.style.left="0px";

this.slider.style.fontSize="1px";

this.slider.style.cursor="move";

```

```
this.slider.theElement=this.slider;
```

```
this.slider.controller = this;
```

```
this.slider.onmousedown= function(e) {
```

```
    if (typeof event != 'undefined') {
```

```
        Slider.beginDrag(this, window.event);
```

```
    } else {
```

```
        Slider.beginDrag(this, e);
```

```
    }
```

```
}
```

```
this.container.appendChild(this.slider);
```

```
//Crea la etiqueta que señala el valor del parametro.
```

```
this.tooltip = document.createElement("div");
```

```
this.tooltip.style.position="absolute";
```

```
this.tooltip.style.background = "#FFFFFF";
```

```
this.tooltip.style.visibility = "hidden";
```

```
this.tooltip.style.fontFamily="Arial, Helvetica, sans-serif";
```

```
this.tooltip.style.fontSize="11px";
```

```
this.tooltip.style.left = (parseInt(this.slider.style.width) + 10) + "px";
```

```
this.container.appendChild(this.tooltip);
```

```
}
```

```
Slider.prototype.update = function(value) {
```



```
}
```

```
Slider.beginDrag=function(theElement, event) {
```

```
    var deltaX = event.clientX - parseInt(theElement.style.left);
```

```
    var deltaY = event.clientY - parseInt(theElement.style.top);
```

```
    if (document.addEventListener) {
```

```
        document.addEventListener("mousemove", moveHandler, true);
```

```
        document.addEventListener("mouseup", upHandler, true);
```

```
    }
```

```
    else if (document.attachEvent) {
```

```
        document.attachEvent("onmousemove", moveHandler);
```

```
        document.attachEvent("onmouseup", upHandler);
```

```
    }
```

```
    var y=(event.clientY - deltaY)
```

```
    var currentvalue = roundValue((theElement.maxY - y) * theElement.controller.valueStep +  
theElement.controller.minValue);
```

```
    theElement.controller.tooltip.innerHTML = currentvalue;
```

```
    theElement.controller.tooltip.style.visibility = "visible";
```

```
    theElement.controller.tooltip.style.top = y + "px";
```

```
    if (event.stopPropagation)
```

```
        event.stopPropagation();
```

else

event.cancelBubble = true;

if (event.preventDefault)

event.preventDefault();

else

event.returnValue = false;

function moveHandler(e) {

if (!e) e = window.event;

var y=(e.clientY - deltaY)

if (y<theElement.minY) y=theElement.minY;

if (y>theElement.maxY) y=theElement.maxY;

theElement.style.top = y + "px";

*var currentvalue = roundValue((theElement.maxY - y) * theElement.controller.valueStep + theElement.controller.minValue);*

theElement.controller.tooltip.innerHTML = currentvalue;

theElement.controller.tooltip.style.top = y + "px";

theElement.controller.currentValue = currentvalue;

if (e.stopPropagation) e.stopPropagation();

else e.cancelBubble = true;

}

```

function upHandler(e) {
    if (!e) e = window.event;

    var y=(e.clientY - deltaY)
    if (y<theElement.minY) y=theElement.minY;
    if (y>theElement.maxY) y=theElement.maxY;

    theElement.style.top = y + "px";

    theElement.controller.tooltip.style.visibility = "hidden";

    var currentvalue = roundValue((theElement.maxY - y) * theElement.controller.valueStep +
theElement.controller.minValue);
    theElement.controller.currentValue = currentvalue;

    if (document.removeEventListener) {
        document.removeEventListener("mouseup", upHandler, true);
        document.removeEventListener("mousemove", moveHandler, true);
    }
    else if (document.detachEvent) {
        document.detachEvent("onmouseup", upHandler);
        document.detachEvent("onmousemove", moveHandler);
    }

    if (e.stopPropagation) e.stopPropagation();
        else e.cancelBubble = true;
}

```

```

function roundValue(num) {
    var factor = Math.pow(10, theElement.controller.accuracy);
    return Math.round(num*factor)/factor;
}
}

```

*/** TimeGraph - Representación gráfica de los valores del parametro en los últimos 60 segundos*/*

```

function TimeGraph(divid, parameter, opts) {
    var container=document.getElementById(divid);
    var point = document.createElement('div');
    container.style.position="relative";
    point.style.position="absolute";
    point.style.fontSize="1px";

    if(!opts) opts={};
    if(opts.maxTime) this.maxTime = opts.maxTime;
        else this.maxTime=60;
    if(opts.maxValue) this.maxValue = opts.maxValue;
        else this.maxValue=100;
    if(opts.minValue) this.minValue = opts.minValue;
        else this.minValue=0;
    if(opts.color) point.style.background=opts.color;
        else point.style.background="#FF0000";
    if(opts.fill) this.fill= opts.fill;
        else this.fill= false;
    if(opts.rate) this.rate=opts.rate;
        else this.rate=1;
    this.samples=0;
}

```

```

this.containerHeight = parseInt(container.style.height);
this.valueInc = this.containerHeight/(this.maxValue-this.minValue);

this.originPos = (-this.minValue * this.valueInc);
this.containerWidth = parseInt(container.style.width);
this.timeInc = this.containerWidth/this.maxTime;

point.style.width = Math.ceil(this.timeInc) + "px";
point.style.height = Math.ceil(this.timeInc) + "px";

this.points = [];
this.markers = [];

for (var x=0; x<this.maxTime; x++) {
    this.points[x] = point.cloneNode(true);
    if (opts.marker && x%5==0) {
        var pointmarker = document.createElement('div');
        pointmarker.style.position="absolute";
        pointmarker.style.width="1px";
        pointmarker.style.height="3px";
        pointmarker.style.top=this.containerHeight + "px";
        pointmarker.style.left=Math.ceil(x * this.timeInc) + "px";
        pointmarker.style.background=opts.marker;
        pointmarker.style.fontSize="1px";
        this.markers[x]=pointmarker;
        container.appendChild(pointmarker);
        this.points[x].pointmarker=true;
    } else {

```

```

        this.points[x].pointmarker=false;
        this.markers[x]=null;
    }
    this.points[x].style.visibility = "hidden";
    this.points[x].style.left = Math.ceil(x * this.timeInc) + "px";
    this.points[x].style.top = this.containerHeight + "px";
    container.appendChild(this.points[x]);
}

```

```

//marcas numéricas del eje

```

```

if(!opts.labelHeight) opts.labelHeight=10;
if(!opts.labelWidth) opts.labelWidth=20;

```

```

var lbl = document.createElement('div');
lbl.style.position = "absolute";
lbl.style.height=opts.labelHeight + "px";
lbl.style.width=opts.labelWidth + "px";
lbl.style.fontSize=opts.labelHeight + "px";
lbl.style.fontFamily="Arial, Helvetica, sans-serif";
lbl.style.left= -opts.labelWidth + "px";
var lbly;

```

```

for (var lblpos=0; lblpos<=100; lblpos+=25) {

```

```

    if (opts['label' + lblpos]) {

```

```

        var l=lbl.cloneNode(true);

```

```

        lbly = Math.ceil(this.containerHeight - (lblpos/100 * this.containerHeight +
opts.labelHeight/2));

```

```

        l.style.top = lbly + "px";

```

```

        //l.style.left = lblx + "px";

```

```

        l.innerHTML = opts['label' + lblpos];
        container.appendChild(l);
    }
}

//Dibujar marcas numéricas indicadoras

if (typeof opts.ticks != "undefined" && opts.ticks > 0) {
    var tickinterval = (this.maxValue - this.minValue) / (opts.ticks - 1);
    for (var t=0; t<opts.ticks; t++) {

        var m = point.cloneNode(true);
        m.style.background="#242424";
        m.style.height="1px";
        m.style.width="3px";
        m.style.left = "-12px";
        m.style.top = Math.ceil(this.valueInc * t * tickinterval) + "px";
        container.appendChild(m);
    }

}

SMDData.registerElement(parameter, this);
}

//Función de actualización del elemento TimeGraph

TimeGraph.prototype.update = function(value) {
    this.samples++;
    if (this.samples % this.rate!=0) return;

```

```

if (value<this.minValue) value=this.minValue;
if (value>this.maxValue) value=this.maxValue;

var p = this.points.shift();
var tmpm = this.markers.shift();

for (var x=0; x<this.maxTime-1; x++) {
    this.points[x].style.left = Math.ceil(x * this.timeInc) + "px";
}
p.style.visibility= "visible";
var py = Math.abs(Math.ceil(value * this.valueInc));
p.style.left = Math.ceil((this.maxTime-1) * this.timeInc) + "px";
if (value>=0) {
    p.style.top = (this.containerHeight - (this.originPos + py)) + "px";
    if (this.fill) {
        p.style.height = py + "px";
    }
} else {
    p.style.top = (this.containerHeight - this.originPos) + "px";
    if (this.fill) {
        p.style.height = py + "px";
    }
}
this.points.push(p);
this.markers.push(tmpm);

for (var x=0; x<this.maxTime; x++) {
    if (this.points[x].pointmarker) this.markers[x].style.left = this.points[x].style.left;
}
}

```


*/** SelectorSwitch - Función que crea una ruleta para seleccionar multiples opciones */*

```
function SelectorSwitch(divid, parameter, opts) {  
  
    if (!opts) opts={};  
    if (opts.selectorImage) this.selectorImage=opts.selectorImage;  
    if (opts.selectorWidth) this.selectorWidth=opts.selectorWidth;  
        else this.selectorWidth=10;  
    if (opts.selectorHeight) this.selectorHeight=opts.selectorHeight;  
        else this.selectorHeight=10;  
  
    if (opts.positions) {  
        this.positions = opts.positions.length;  
        this.labelvalues = opts.positions;  
    }  
    if (opts.labelHeight) this.labelHeight = parseInt(opts.labelHeight);  
        else this.labelHeight = 12;  
    if (opts.labelWidth) this.labelWidth = parseInt(opts.labelWidth);  
        else this.labelWidth = 20;  
  
    this.parameter= parameter;  
    var container = document.getElementById(divid);  
    var selector = document.createElement('div');  
    container.style.position="relative";  
    selector.style.position="absolute";  
    selector.style.width=this.selectorWidth + "px";  
    selector.style.height=this.selectorHeight + "px";  
  
    selector.style.fontSize="1px";
```

```

selector.style.backgroundImage="url(" + this.selectorImage + ")";
selector.style.visibility="hidden";
container.innerHTML = '<map id="map1" name="test"></map>'
var map=document.getElementById('map1');

container.appendChild(selector);

var img = document.createElement('img');
img.src="images/spacer.gif";
img.position="absolute";
img.style.width=container.style.width;
img.style.height=container.style.height;
img.style.top="0px";
img.style.left="0px";
img.border="0";
img.useMap="#test";

var containerR = parseInt(container.style.width) / 2;
var selectorR = parseInt(selector.style.width) / 2;
var sfo = containerR-selectorR-5;

var angleStep = 2 * Math.PI/this.positions;
this.selectorPositions = new Array();
for (var p=0; p<this.positions; p++) {

    var tmpx, tmpy
    tmpx = sfo * Math.cos(p*angleStep);
    tmpy = sfo * Math.sin(p*angleStep);

    tmpx = tmpx - selectorR + containerR;

```

```
tmpy = tmpy - selectorR + containerR;
```

```
this.selectorPositions[p] = {x: Math.ceil(tmpx), y: Math.ceil(tmpy)}
```

```
//Dibujar marcas numéricas
```

```
var lblx, lbly;
```

```
lblx = Math.ceil( (containerR + 10) * Math.cos(p*angleStep) ) + containerR -  
this.labelWidth/2;
```

```
lbly = Math.ceil( (containerR + 10) * Math.sin(p*angleStep) ) + containerR -  
this.labelHeight/2;
```

```
var lbl = document.createElement('div');
```

```
lbl.style.position = "absolute";
```

```
lbl.style.left = lblx + "px";
```

```
lbl.style.top = lbly + "px";
```

```
lbl.style.width=this.labelWidth + "px";
```

```
lbl.style.height=this.labelHeight + "px";
```

```
lbl.style.fontSize=this.labelHeight + "px";
```

```
lbl.style.fontFamily="Arial, Helvetica, sans-serif";
```

```
lbl.style.textAlign= "center";
```

```
lbl.innerHTML = this.labelvalues[p].label;
```

```
lbl.onclick = function() {SelectorSwitch.click(this)}
```

```
lbl.selector = this;
```

```
lbl.area = p;
```

```
container.appendChild(lbl);
```

```
//Definir las áreas de pulsación
```

```
var x0, x1, x2, x3, y0, y1, y2, y3;
```

```
x0 = Math.ceil(0);
```

```
y0 = Math.ceil(0);
```

```
x1 = Math.ceil(containerR * Math.cos(p*angleStep - angleStep/2) );
```

```
y1 = Math.ceil(containerR * Math.sin(p*angleStep - angleStep/2) );
```

```
x2 = Math.ceil(containerR * Math.cos(p*angleStep) );
```

```
y2 = Math.ceil(containerR * Math.sin(p*angleStep) );
```

```
x3 = Math.ceil(containerR * Math.cos(p*angleStep + angleStep/2) );
```

```
y3 = Math.ceil(containerR * Math.sin(p*angleStep + angleStep/2) );
```

```
x0+=containerR;y0+=containerR;x1+=containerR;y1+=containerR;x2+=containerR;y2+=containerR;x3+=containerR;y3+=containerR;
```

```
var area = document.createElement("area");
```

```
area.shape="poly";
```

```
area.coords=x0 + "," + y0 + "," + x1 + "," + y1 + "," + x2 + "," + y2 + "," + x3 +  
"," + y3;
```

```
area.onclick = function() {SelectorSwitch.click(this)}
```

```
area.selector = this;
```

```
area.area = p;
```

```
map.style.position = "absolute";
```

```
map.appendChild(area);
```

```
}
```

```
container.appendChild(img);
```

```
this.container = container;
```

```
this.selector= selector;
```

```
SMDData.registerElement(parameter, this);
```

```
}
```

```
//Función de actualización del elemento SelectorSwitch
```

```
SelectorSwitch.prototype.update = function(value) {
```

```
    for (var x=0; x<this.labelvalues.length; x++) {
```

```
        if (this.labelvalues[x].value == value) {
```

```
            this.select(x);
```

```
            return;
```

```
        }
```

```
    }
```

```
this.selector.style.visibility="hidden";
```

```
}
```

```
SelectorSwitch.prototype.select = function(value) {
```

```
    this.selector.style.visibility="visible";
```

```
    if (this.selectorPositions[value]) {
```

```
        this.selector.style.left= this.selectorPositions[value].x;
```

```

        this.selector.style.top= this.selectorPositions[value].y;
    }

}

SelectorSwitch.click= function(a) {
    a.selector.select(a.area);
    SMData.setParameter(a.selector.parameter, a.selector.labelvalues[a.area].value);
}

/** Meter - Display que representa de manera analógica el valor de un parametro*/

function Meter(divid, parameter, opts) {
    var container = document.getElementById(divid);
    container.style.position="relative";

    var point = document.createElement('div');
    point.style.position="absolute";
    point.style.fontSize="1px";
    point.style.width = "1px";
    point.style.height = "1px";

    if (!opts) opts = {};

    if (typeof opts.maxValue != "undefined") this.maxValue = opts.maxValue;
        else this.maxValue = 100;
    if (typeof opts.minValue != "undefined") this.minValue = opts.minValue;
        else this.minValue = 0;
    if (opts.color) point.style.background = opts.color;
        else point.style.background = "black";
}

```

```

this.rad = parseInt(container.style.height)

this.lineSteps = this.rad;

this.points = [];

for (var r=0; r<this.lineSteps; r++) {
    this.points[r] = point.cloneNode(true);
    container.appendChild(this.points[r]);
}

this.angleStep = -(Math.PI / 2) / (this.maxValue - this.minValue);
this.angleStart = 0
this.originx = 0;
this.originy = parseInt(container.style.height);

//Dibujar etiquetas de valor del parámetro

if (!opts.labelHeight) opts.labelHeight=10;
if (!opts.labelWidth) opts.labelWidth=20;

var lbl = document.createElement('div');
lbl.style.position = "absolute";
lbl.style.height=opts.labelHeight + "px";
lbl.style.width=opts.labelWidth + "px";
lbl.style.fontSize=opts.labelHeight + "px";

var lblx, lbly;
var lblr = parseInt(container.style.height) - opts.labelHeight;

```

```

for (var lblpos=0; lblpos<=100; lblpos+=25) {
    if (opts['label' + lblpos]) {
        var l=lbl.cloneNode(true);
        lblx = Math.ceil( lblr * Math.cos(-(lblpos/100) * (Math.PI/2)) -
opts.labelWidth/2);
        lbly = Math.ceil( lblr * Math.sin(-(lblpos/100) * (Math.PI/2)) -
opts.labelHeight/2 + this.originy);

        l.style.top = lbly + "px";
        l.style.left = lblx + "px";
        l.innerHTML = opts['label' + lblpos];
        container.appendChild(l);
    }
}

if (opts.units) {
    var l= lbl.cloneNode(true);
    lblx = Math.ceil( .5 * lblr * Math.cos(-(Math.PI/4)) - opts.labelWidth/2);
    lbly = Math.ceil( .5 * lblr * Math.sin(-(Math.PI/4)) - opts.labelHeight/2 +
this.originy);
    l.innerHTML=opts.units;
    l.style.top = lbly + "px";
    l.style.left = lblx + "px";

    container.appendChild(l);
}

//Dibujar indicadores de max/min

```



```

    if (typeof opts.maxIndicator !== "undefined" && opts.maxIndicator <= this.maxValue &&
    opts.maxIndicator >= this.minValue) {
        for (var r=this.originy -10 ; r<this.originy; r++) {
            var m = point.cloneNode(true);
            m.style.background="#FF0000";
            var x = r * Math.cos(this.angleStep * (opts.maxIndicator - this.minValue));
            var y = r * Math.sin(this.angleStep * (opts.maxIndicator - this.minValue));
            m.style.left = Math.ceil(x) + "px";
            m.style.top = Math.ceil(y+this.originy) + "px";

            container.appendChild(m);

        }
    }

    if (typeof opts.minIndicator !== "undefined" && opts.minIndicator <= this.maxValue &&
    opts.minIndicator >= this.minValue) {
        for (var r=this.originy -10 ; r<this.originy; r++) {
            var m = point.cloneNode(true);
            m.style.background="#0000FF";
            var x = r * Math.cos(this.angleStep * (opts.minIndicator - this.minValue));
            var y = r * Math.sin(this.angleStep * (opts.minIndicator - this.minValue));
            m.style.left = Math.ceil(x) + "px";
            m.style.top = Math.ceil(y+this.originy) + "px";

            container.appendChild(m);

        }
    }

    if (typeof opts.ticks !== "undefined" && opts.ticks > 0) {
        var tickinterval = (this.maxValue - this.minValue) / (opts.ticks - 1);
        for (var t=0; t<opts.ticks; t++) {

```

```

    for (var r=this.originy -5 ; r<this.originy+5; r++) {
        var m = point.cloneNode(true);
        m.style.background="#CCCCCC";
        var x = r * Math.cos(this.angleStep * (tickinterval * t));
        var y = r * Math.sin(this.angleStep * (tickinterval * t));
        m.style.left = Math.ceil(x) + "px";
        m.style.top = Math.ceil(y+this.originy) + "px";

        container.appendChild(m);

    }
}

```

```

    SMDData.registerElement(parameter, this);
}

```

//Función de actualización del elemento Meter

```

Meter.prototype.update = function(value) {
    if (value<this.minValue) value=this.minValue;
    if (value>this.maxValue) value=this.maxValue;

    for (var r=0; r<this.lineSteps; r++) {
        var x = r * Math.cos(this.angleStep * (value - this.minValue));
        var y = r * Math.sin(this.angleStep * (value - this.minValue));
        this.points[r].style.left = Math.ceil(x) + "px";
        this.points[r].style.top = Math.ceil(y+this.originy) + "px";
    }
}

```

```

    }

}

/** Light - Función que indica el estado de un parametro (on/off) */

function Light(divid, parameter, opts) {
    if (parameter.indexOf("b")!=-1) {
        var bit = parameter.substr(parameter.indexOf("b")+1)
        this.bitmask = 1<<bit;
        parameter=parameter.substr(0, parameter.indexOf("b"));
    } else this.bitmask = 0;

    this.container = document.getElementById(divid);
    if (!opts) opts={};
    if (opts.imageOn) this.imageOn = opts.imageOn;
    if (opts.imageOff) this.imageOff = opts.imageOff;

    SMDData.registerElement(parameter, this);

}

//Función de actualización del elemento Light

Light.prototype.update = function(value) {
    var v=value;
    if (this.bitmask!=0) {

```

```

        v = parseInt(value) & this.bitmask;
    }

    if (v==0) {
        this.container.style.backgroundImage = "url(" + this.imageOff + ")";
    } else {
        this.container.style.backgroundImage = "url(" + this.imageOn + ")";
    }
}

/* MomentarySwitch - Función que, tras pulsar sobre el, escribe el valor de un parámetro*/

function MomentarySwitch (divid, parameter, opts) {
    if (!opts) opts={};

    this.container = document.getElementById(divid);

    var value;
    if (typeof opts.value != "undefined") value = opts.value;
        else value = 1;

    var readParameter=parameter;

    if (parameter.indexOf("b")!=-1) {
        var bit = parameter.substr(parameter.indexOf("b")+1)
        this.bitmask = 1<<bit;
        readParameter=parameter.substr(0, parameter.indexOf("b"));
    } else this.bitmask = 0;
}

```

```
if (opts.imageOn) this.imageOn = opts.imageOn;
if (opts.imageOff) this.imageOff = opts.imageOff;
if (opts.imagePressOn) this.imagePressOn = opts.imagePressOn;
    else this.imagePressOn=this.imageOn;
if (opts.imagePressOff) this.imagePressOff = opts.imagePressOff;
    else this.imagePressOff=this.imageOff;
```

```
//precarga de imagenes
```

```
this.preloaderOn = new Image;
this.preloaderOn.src = this.imageOn;
this.preloaderOff = new Image;
this.preloaderOff.src = this.imageOff;
this.preloaderPressOn = new Image;
this.preloaderPressOn.src = this.imagePressOn;
this.preloaderPressOff = new Image;
this.preloaderPressOff.src = this.imagePressOff;
```

```
this.pressed=false;
this.currentValue=0;
```

```
SMDData.registerElement(readParameter, this);
```

```
if (opts.setParameter) {
    parameter = opts.setParameter;
}
```

```
this.container.onclick = function() {
    SMDData.setParameter(parameter, value);
}
```

```

this.container.controller=this;
this.container.onmousedown= function(e) {
    if (typeof event != 'undefined') {
        MomentarySwitch.beginPress(this, window.event);
    } else {
        MomentarySwitch.beginPress(this, e);
    }
}
}

```

```

MomentarySwitch.prototype.update = function(value) {
    this.currentValue=value;

    var v=value;

    if (this.bitmask!=0) {
        v = parseInt(value) & this.bitmask;
    }

    var imageOn= this.imageOn;
    var imageOff=this.imageOff;

    if (this.pressed) {
        imageOn=this.imagePressOn;
        imageOff=this.imagePressOff;
    }
}

```

```

    if (v==0) {
        this.container.style.backgroundImage = "url(" + imageOff + ")";
    } else {
        this.container.style.backgroundImage = "url(" + imageOn + ")";
    }
}

```

//Función de actualización del elemento MomentarySwitch

```

MomentarySwitch.beginPress=function(theElement, event) {

    theElement.controller.pressed=true;
    theElement.controller.update(theElement.controller.currentValue);

    if (document.addEventListener) {
        document.addEventListener("mouseup", upHandler, true);
    }
    else if (document.attachEvent) {
        document.attachEvent("onmouseup", upHandler);
    }

    if (event.stopPropagation)
        event.stopPropagation();
    else
        event.cancelBubble = true;

    if (event.preventDefault)
        event.preventDefault();
    else

```

```
event.returnValue = false;
```

```
function upHandler(e) {
```

```
  if (!e) e = window.event;
```

```
  theElement.controller.pressed = false;
```

```
  theElement.controller.update(theElement.controller.currentValue);
```

```
  if (document.removeEventListener) {
```

```
    document.removeEventListener("mouseup", upHandler, true);
```

```
  }
```

```
  else if (document.detachEvent) {
```

```
    document.detachEvent("onmouseup", upHandler);
```

```
  }
```

```
  if (e.stopPropagation) e.stopPropagation();
```

```
    else e.cancelBubble = true;
```

```
  }
```

```
}
```

```
/* KeyPad - Función que crea un teclado en pantalla para introducir valores a determinados  
parametros */
```

```
function KeyPad(divid, parameter, opts) {
```

```
  var container = document.getElementById(divid);
```

```
  var keypadHTML = [];
```

```
  keypadHTML.push('<table width="322" border="0" cellspacing="0" cellpadding="0">');
```



```

keypadHTML.push('<tr>');
keypadHTML.push('<td width="10"></td>');
keypadHTML.push('<td background="images/digital.gif" width="302"><table
width="302" border="0" cellspacing="0" cellpadding="0">');
keypadHTML.push('<tr>');
keypadHTML.push('<td align="right" class="Digia">');
keypadHTML.push('<div id="" + divid + '_text"></div>');
keypadHTML.push('</td>');
keypadHTML.push('<td width="20"></td>');
keypadHTML.push('</tr>');
keypadHTML.push('</table></td>');
keypadHTML.push('<td width="10"></td>');
keypadHTML.push('</tr>');
keypadHTML.push('<tr><td height="10" colspan="3"></td></tr>');
keypadHTML.push('<tr>');
keypadHTML.push('<td>&nbsp;</td>');
keypadHTML.push('<td align="right" valign="top">');
keypadHTML.push('<table width="165" border="0" cellpadding="0" cellspacing="5"
class="NAME">');
keypadHTML.push('<tr align="left" valign="middle">');
keypadHTML.push('<td width="30" height="30" align="left"><a href="#"
onmouseover="KeyPad.mouseover(this, 1);" onmouseout="KeyPad.mouseout(this, 1);"
onclick="return KeyPad.pressed(\' + divid + \', 1);"></a></td>');
keypadHTML.push('<td width="30" height="30" align="left"><a href="#"
onmouseover="KeyPad.mouseover(this, 2);" onmouseout="KeyPad.mouseout(this, 2);"
onclick="return KeyPad.pressed(\' + divid + \', 2);"></a></td>');
keypadHTML.push('<td width="30" height="30" align="left"><a href="#"
onmouseover="KeyPad.mouseover(this, 3);" onmouseout="KeyPad.mouseout(this, 3);"

```

```
onclick="return KeyPad.pressed(\" + divid + '\', 3);"></a></td>');
```

```
keypadHTML.push('<td width="30" height="30" align="left"><a href="#"
onmouseover="KeyPad.mouseover(this, \'c\');" onmouseout="KeyPad.mouseout(this, \'c\');"
onclick="return KeyPad.pressed(\" + divid + '\', \'c\');"></a></td>');
```

```
keypadHTML.push('</tr>');
```

```
keypadHTML.push('<tr align="center" valign="middle">');
```

```
keypadHTML.push('<td width="30" height="30" ><a href="#"
onmouseover="KeyPad.mouseover(this, 4);" onmouseout="KeyPad.mouseout(this, 4);"
onclick="return KeyPad.pressed(\" + divid + '\', 4);"></a></td>');
```

```
keypadHTML.push('<td width="30" height="30" ><a href="#"
onmouseover="KeyPad.mouseover(this, 5);" onmouseout="KeyPad.mouseout(this, 5);"
onclick="return KeyPad.pressed(\" + divid + '\', 5);"></a></td>');
```

```
keypadHTML.push('<td width="30" height="30" ><a href="#"
onmouseover="KeyPad.mouseover(this, 6);" onmouseout="KeyPad.mouseout(this, 6);"
onclick="return KeyPad.pressed(\" + divid + '\', 6);"></a></td>');
```

```
keypadHTML.push('<td rowspan="3" width="30" height="100" ><a href="#"
onmouseover="KeyPad.mouseover(this, \'submit\');"
onmouseout="KeyPad.mouseout(this, \'submit\');" onclick="return KeyPad.pressed(\" + divid +
\', \'submit\');"></a></td>');
```

```
keypadHTML.push('</tr>');
```

```
keypadHTML.push('<tr align="center" valign="middle"> ');
```

```
keypadHTML.push('<td width="30" height="30" ><a href="#"
onmouseover="KeyPad.mouseover(this, 7);" onmouseout="KeyPad.mouseout(this, 7);"
onclick="return KeyPad.pressed(\" + divid + '\', 7);"></a></td>');
```

```
keypadHTML.push('<td width="30" height="30" ><a href="#"
onmouseover="KeyPad.mouseover(this, 8);" onmouseout="KeyPad.mouseout(this, 8);"
onclick="return KeyPad.pressed(\" + divid + '\', 8);"></a></td>');
```

```
keypadHTML.push('<td width="30" height="30" ><a href="#"
onmouseover="KeyPad.mouseover(this, 9);" onmouseout="KeyPad.mouseout(this, 9);"
```

```

onclick="return KeyPad.pressed(\" + divid + '\', 9);"></a></td>');

    keypadHTML.push('</tr>');

    keypadHTML.push('<tr align="center" valign="middle"> ');

    keypadHTML.push('<td width="30" height="30" ><a href="#"
onmouseover="KeyPad.mouseover(this, \'point\');"
onmouseout="KeyPad.mouseout(this, \'point\');" onclick="return KeyPad.pressed(\" + divid +
\', \'point\');"></a></td>');

    keypadHTML.push('<td width="30" height="30" ><a href="#"
onmouseover="KeyPad.mouseover(this, 0);" onmouseout="KeyPad.mouseout(this, 0);"
onclick="return KeyPad.pressed(\" + divid + '\', 0);"></a></td>');

    keypadHTML.push('<td width="30" height="30" ><a href="#"
onmouseover="KeyPad.mouseover(this, \'plusmini\');"
onmouseout="KeyPad.mouseout(this, \'plusmini\');" onclick="return KeyPad.pressed(\" + divid +
\', \'plusmini\');"></a></td>');

    keypadHTML.push('</tr>');
    keypadHTML.push('</table>');
    keypadHTML.push('</td>');
    keypadHTML.push('<td> &nbsp; </td>');
    keypadHTML.push('</tr>');
    keypadHTML.push('</table>');

    container.innerHTML = keypadHTML.join("\n");
    this.parameter = parameter;
    this.currentValue="0";
    this.divid=divid;
    this.lastkey="";
    container.controller = this;
    document.getElementById(divid + "_text").innerHTML = "0";

}

```

```

Keypad.mouseover = function(a, img) {
    var theimg=a.childNodes[0];
    theimg.src="images/calc/" + img + "_over.gif";
}

```

```

Keypad.mouseout = function(a, img) {
    var theimg=a.childNodes[0];
    theimg.src="images/calc/" + img + ".gif";
}

```

```

Keypad.pressed = function(divid, key) {

    var container = document.getElementById(divid);
    var controller = container.controller;
    var lastkey = container.lastkey;

    if (key=="submit") {
        if (controller.currentValue != "") controller.update(controller.currentValue);
    } else {
        if (lastkey=="submit") controller.currentValue="0";
        //limpia el texto si la ultima acción fue de enviar el dato
    }

    if (controller.currentValue=="0") controller.currentValue="";
    switch(key) {

        case(1): if (controller.currentValue.length<20) controller.currentValue += "1";
break;

        case(2): if (controller.currentValue.length<20) controller.currentValue += "2";

```

```

break;
    case(3): if (controller.currentValue.length<20) controller.currentValue += "3";
break;
    case(4): if (controller.currentValue.length<20) controller.currentValue += "4";
break;
    case(5): if (controller.currentValue.length<20) controller.currentValue += "5";
break;
    case(6): if (controller.currentValue.length<20) controller.currentValue += "6";
break;
    case(7): if (controller.currentValue.length<20) controller.currentValue += "7";
break;
    case(8): if (controller.currentValue.length<20) controller.currentValue += "8";
break;
    case(9): if (controller.currentValue.length<20) controller.currentValue += "9";
break;
    case(0): if (controller.currentValue != "") controller.currentValue += "0"; break;
    case("00") : if (controller.currentValue.length<19) if (controller.currentValue !=
"") controller.currentValue += "00"; break;
    case("c") : controller.currentValue = "0"; break;
    case("point") : if (controller.currentValue.length<20) if
(controller.currentValue.indexOf(".") == -1) controller.currentValue+="."; break;
    case("plusmini"): if (controller.currentValue!="") {
        if (controller.currentValue.substr(0,1) == "-")
            controller.currentValue=controller.currentValue.substr(1);
        else
            controller.currentValue="-" +
controller.currentValue;
    }
    break;
}
}

```

```

if (controller.currentValue=="") controller.currentValue="0";

document.getElementById(divid + "_text").innerHTML = controller.currentValue;

container.lastkey = key;
return false;

}

//Función de actualización del elemento KeyPad

KeyPad.prototype.update = function(value) {
    SMDData.setParameter (this.parameter, value);
}

/* Indicator - Indicador luminoso en pantalla que parpadea al ser actualizado el registro de
parametros */

function Indicator(divid, imageOn, imageOff) {
    this.container = document.getElementById(divid);
    this.imageOn = imageOn;
    this.imageOff = imageOff;
    this.container.style.backgroundImage="url(" + imageOff + ")";

    this.preloader = new Image;
    this.preloader.src = imageOn;

    SMDData.registerIndicator(this);

}

```

//Función de actualización del elemento Indicator

```
Indicator.prototype.update = function(value) {  
    if (value) {  
        this.container.style.backgroundImage="url(" + this.imageOn + ")";  
        setTimeout("SMDData.indicator.update(false);", 200);  
    } else {  
        this.container.style.backgroundImage="url(" + this.imageOff + ")";  
    }  
}
```

9 Bibliografía

- Máquinas eléctricas Fraile Mora, Jesús *Colegio de Ingenieros de Caminos, Canales y Puertos*
- Brushless permanent-magnet and reluctance motor drives T.J.E. Miller *Clarendon Press Oxford*
- The Control Techniques Drives and Controls Handbook Control Techniques
- Design of brushless permanent-magnet J.R. Hendershot *Magna Physics publicaciones*
- Installation Guide: Unidrive Regen Control Techniques
- User Guide: Unidrive Control Techniques
- Guía de instalación Unidrive Regen Control Techniques
- User Guide SM-Applications Control Techniques
- User Guide SM-Ethernet Control Techniques
- Advanced User Guide Unidrive SP Control Techniques
- Guía rápida Unidrive SP Control Techniques
- HTML & Web design [electronic resource] : tips & techniques , Jamsa, Kris A., *McGraw-Hill/Osborne*,
- HTML dinamico, ASP y JavaScript a través de ejemplos Bobadilla Sancho, Jesús *Madrid : Ra-Ma*

10 Referencias web

- Introducción a lenguaje Javascript: <http://www.librosweb.es/javascript/>
- Manual de lenguaje HTML: <http://www.desarrolloweb.com/html/>
- Utilización del navegador web Firefox en países europeos: <http://www.xitimonitor.com/fr-fr/barometre-des-navigateurs/firefox-mars-2008/index-1-1-3-127.html?xtor=11>,
- http://es.wikipedia.org/wiki/C%C3%B3digo_HTML
- <http://es.wikipedia.org/wiki/Javascript>
- http://es.wikipedia.org/wiki/Conmutador_de_red
- http://es.wikipedia.org/wiki/Cliente_FTP

