

UNIVERSIDAD NACIONAL DEL CALLAO

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



TESIS

**“MODELAMIENTO DE UNA UNIDAD DE
CONTROL PARA ENSAYOS DE MOTORES DE
COMBUSTION EN TIEMPO REAL”**

**PARA OBTENER EL TITULO PROFESIONAL DE INGENIERO
ELECTRÓNICO**

PRESENTADO POR LOS BACHILLERES:

- **YUCRA QUICHUA, LUIS ALEXANDER.**
- **PEÑA SHAPIAMA, JUAN CARLOS.**
- **MOSQUERA PEREZ, MARVIN.**

Callao, 2016

PERU

UNIVERSIDAD NACIONAL DEL CALLAO
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



TESIS

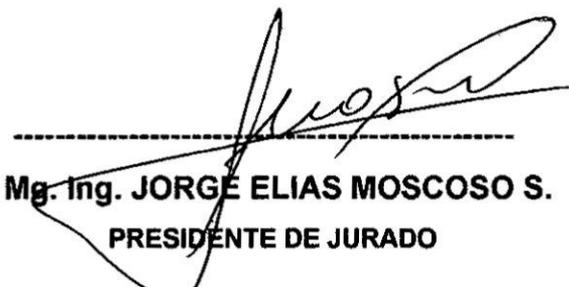
“MODELAMIENTO DE UNA UNIDAD DE CONTROL PARA ENSAYOS DE MOTORES DE COMBUSTION EN TIEMPO REAL”

PARA OBTENER EL TITULO PROFESIONAL DE INGENIERO ELECTRÓNICO

PRESENTADO POR LOS BACHILLERES:

- YUCRA QUICHUA, LUIS ALEXANDER.
- PEÑA SHAPIAMA, JUAN CARLOS.
- MOSQUERA PEREZ, MARVIN A.

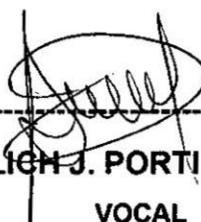
ASESOR: Mg. Ing. ABILIO BERNARDINO CUZCANO RIVAS
CALIFICACION: 15 (QUINCE)



Mg. Ing. JORGE ELIAS MOSCOSO S.
PRESIDENTE DE JURADO



Ing. LUIS E. CRUZADO MONTAÑEZ
SECRETARIO



Ing. ARLICH J. PORTILLO ALLENDE
VOCAL

Callao, PERU 2016

CARATULA

PAGINA DE RESPETO

HOJA DE REFERENCIA DEL JURADO Y APROBACION

INDICE

I. INDICE DE FIGURAS.....	06
II. RESUMEN.....	08
III. PLANTEAMIENTO DE LA INVESTIGACION.....	11
3.1 <i>Determinación del Problema.....</i>	11
3.2 <i>Formulación del Problema.....</i>	12
3.3 <i>Objetivos de Investigación.....</i>	12
3.4 <i>Justificación.....</i>	13
3.5 <i>Limitaciones y Alcances.....</i>	14
IV. FUNDAMENTO TEORICO.....	15
4.1 <i>Antecedentes del Estudio.....</i>	16
4.2 <i>Marco Teórico.....</i>	16
4.3 <i>Marco Metodologico.....</i>	42
V. VARIABLES E HIPOTESIS.....	97
5.1 <i>Variables de la Investigación.....</i>	97
5.1.1 <i>Variable independiente.....</i>	97
5.1.2 <i>Variables dependientes.....</i>	98
5.2 <i>Operacionalización de variables.....</i>	98
5.3 <i>Hipótesis.....</i>	99

5.3.1 Hipotesis general.....	99
5.3.2 Hipotesis especifica.....	99
VI. METODOLOGIA.....	99
6.1 Tipo de Investigacion.....	100
6.2 Diseño de la Investigación.....	100
6.2.1 Diseño arquitectónico.....	100
6.2.2 Diseño intermedio.....	102
6.2.3 Diseño detallado.....	109
6.2.4 Unidad de control secundario.....	109
6.2.5 Unidad de control primario.....	112
6.2.6 Programacion.....	114
6.2.7 Sistemas operativos.....	116
6.2.8 Implementacion.....	118
6.3 Poblacion y muestra.....	119
6.4 Tecnicas e instrumentos de recoleccion de datos.....	119
6.5 Procedimiento de recolección de datos.....	119
6.6 Procesamiento estadístico y análisis de datos.....	120
VII. RESULTADOS.....	120
7.1 Resultados parciales.....	120
7.2 Resultados finales.....	127
VIII. DISCUSION DE RESULTADOS.....	129
8.1 Contrastacion de hipótesis con los resultados.....	129

8.2	<i>Contrastación de hipótesis con otros estudios similares</i>	130
8.3	<i>Contrastación de impacto ambiental</i>	130
8.3.1	<i>Impacto ambiental de MCI tradicionales</i>	130
8.3.2	<i>Impacto ambiental de MCI aplicando el modelo de unidad de control</i>	131
IX.	CONCLUSIONES	131
X.	RECOMENDACIONES	131
XI.	REFERENCIAS BIBLIOGRÁFICAS	132
XII.	ANEXOS	134
	<i>Anexo A: Simbología</i>	135
	<i>Anexo B: Nomina de eventos</i>	136
	<i>Anexo C: Guía de encuesta</i>	137
	<i>Anexo D: Matriz de consistencia</i>	141

CAPITULO I

I. INDICE DE FIGURAS

Figura N° 01 Sistema híbrido de control	19
Figura N° 02 Esquema general de Control	20
Figura N° 03 Esquema de Control PID	22
Figura N° 04 Esquema de Control PI-D	22
Figura N° 05 Esquema de Control PID-D	23
Figura N° 06 Red de perceptrones con dos capas ocultas	30
Figura N° 07 Ajuste de pesos de la red.....	32
Figura N° 08 Método del gradiente descendente	32
Figura N° 09 Capacidad de generalización de las redes neuronales	36
Figura N° 10 Evolución del error en el proceso de entrenamiento.....	37
Figura N° 11 Aproximación de funciones.....	38
Figura N° 12 Ciclo de vida del desarrollo de software	41
Figura N° 13 El modelo en "V" de desarrollo de software	50
Tabla N° 01 Vocabulario del LEL	64
Tabla N° 02 Entradas al LEL y eventos	65
Figura N° 14 Diagrama de contexto del sistema	66
Tabla N° 03 Nómina de eventos.....	68
Cuadro N° 01 Ficha de evento	69
Cuadro N° 02 Ficha de evento	70
Cuadro N° 03 Ficha de evento	71
Cuadro N° 04 Ficha de evento	72
Cuadro N° 05 Ficha de evento	73
Cuadro N° 06 Ficha de evento	74
Cuadro N° 07 Ficha de evento	75

Cuadro N° 08 Ficha de evento	76
Cuadro N° 09 Ficha de evento	77
Figura N° 15 Modelo de eventos (diagrama "0").....	78
Tabla N° 04 Matriz de eventos	79
Figura N° 16 Diagrama de Entidad-Relación	80
Tabla N° 06 Relación de actividades y objetos.....	84
Figura N° 17 Diagrama de Clases / Contexto	93
Figura N° 18 Diagrama de Clases.....	94
Figura N° 19 Funcionalidades y componentes	98
Figura N° 20 Arquitectura del sistema.....	100
Figura N° 21 Intérprete de comandos	108
Tabla N° 07 Transiciones en el intérprete de comandos.....	108
Figura N° 22 interconexión de bus serial	109
Figura N° 23 Validación de mensaje	109
Tabla N° 08 Protocolo de comunicaciones	110
Tabla N° 09 Condiciones de Control.....	111
Figura N° 24 Selección de la lógica de control	112
Figura N° 26 Evolución del error con el intervalo de integración T3	115
Figuras N° 27 Extrapolación de un punto de la función.....	118
Figuras N° 28 Derivación numérica de la función extrapolada.....	122
Figura N° 29 Comportamiento de los errores con diferentes intervalos entre los puntos conocidos	124
Figura N° 30 Curvas teóricas del motor y freno.....	125
Figura N° 31 Evolución del error en el proceso de entrenamiento de redes con diferentes arquitecturas	127
Figura N° 32 Evolución del error en el proceso de entrenamiento con diferentes valores de β 128	
Figura N° 33 Reproducción por redes neuronales de las curvas del motor y freno	128
Figura N° 34 Respuesta del conjunto motor-freno a un cambio de posición del acelerador	129

Se identifica así, en el dominio del problema planteado, a un sistema que involucra la respuesta dinámica continua de un conjunto mecánico y la acción de una unidad digital de control que opera a partir del muestreo discreto de los valores reales de velocidad y carga. Más adelante, se describirán las características de este sistema de control y será necesario evaluar un modelo capaz de representar el comportamiento dinámico de un motor acoplado a un freno dinamométrico.

Estos sistemas de control, que tienen la finalidad de operar sobre dominios continuos, dan lugar a un tipo particular de sistema denominado "híbrido". Estos sistemas híbridos combinan, por lo tanto, componentes discretos y continuos que interactúan entre sí y responden a concepciones de naturaleza diferente, a pesar de que en última instancia el subsistema discreto no es más que una abstracción de la realidad. Estas dos dinámicas coexisten y es necesario representarlas con modelos que reflejen esta interacción, reconociéndose que las mayores dificultades se originan en la necesidad de sincronizar sus comportamientos. Podría afirmarse que en la actualidad los sistemas de control, en su mayoría, pueden ser reconocidos en alguna medida como híbridos ya que en realidad todo sistema de computación que interactúa con el mundo real queda encuadrado en esta categoría.

En este punto cabe destacar que los sistemas de control forman parte de las numerosas aplicaciones de tiempo real que son desarrolladas en la actualidad.

Éstas abarcan un amplio espectro de destinos, que incluye desde la industria bélica hasta equipos de uso doméstico, y es indudable que la atención de dicha demanda tiene un fuerte impacto en todos los campos de la informática. Estas aplicaciones se caracterizan tanto por sus particulares exigencias de confiabilidad, eficiencia y rápida respuesta como por la diversidad de disciplinas que involucran. Entre estas últimas, se incluyen los lenguajes de programación, protocolos de comunicaciones, arquitectura del hardware, modelos matemáticos, sistemas operativos y teoría de control.

De esta forma, los sistemas de tiempo real se han incorporado a la creciente demanda de software especializado que se verifica en la actualidad. Además, el abaratamiento del hardware y el incremento de la densidad de integración hace propicia la realización de sistemas cada vez más complejos y es de preverse que esta tendencia se acentúe en el futuro próximo.

Así, la necesidad de dar respuesta a esta demanda no ha hecho más que incorporar nuevas exigencias al gran esfuerzo que se viene realizando para superar definitivamente la llamada "crisis del software" y que ha justificado una muy importante inversión de recursos, tanto intelectuales como económicos. Estos esfuerzos han estado orientados a establecer nuevas técnicas, métodos y herramientas que permitan tanto mejorar la calidad de los desarrollos de software como así también a aumentar la productividad de los procesos conducentes a tales desarrollos, aumentando eficiencia y eficacia.

Sin embargo, en la actualidad resulta preocupante comprobar que, en muchos casos, son los intereses comerciales los que orientan el mercado aun en direcciones que se contraponen con lo que indicarían otros puntos de vista, incluido el técnico.

Se acentúa una visión simplista e ingenua de que el desarrollo de sistemas complejos se simplifica con la sola utilización de los métodos que han dado buenos resultados en otros campos de aplicación, o de que todo se resuelve incrementando los recursos de hardware.

Coexiste, felizmente, otra realidad determinada por el ya mencionado amplio espectro de buenas técnicas y herramientas de análisis y diseño de sistemas que han sido desarrolladas y están disponibles. El esfuerzo debe ahora orientarse a hacer que trasciendan los ámbitos de los centros de investigación y se incorporen al mercado cotidiano.

De una manera indudablemente muy modesta, este trabajo espera hacer su aporte en este sentido, demostrando que es posible desarrollar e implementar un simulador de tiempo real de una unidad de control a través de un procedimiento sistemático, racional y riguroso, a los fines de asegurar un modelo que sea a la vez correcto, completo y consistente.

CAPITULO III

III. PLANTEAMIENTO DE LA INVESTIGACION

3.1 Determinación del Problema

Las unidades de control analógicas se caracterizan por sobre las unidades de control digital por su rapidez y facilidad, ya que estos últimos siempre se han requerido de un ajuste muy laborioso para alcanzar buenos desempeños. Sin embargo, bajo un determinado sistema de ensayo para motores estarían destinados a la realización de pruebas en forma completamente automática, estamos convencidos que tiene margen para introducir variantes que posibiliten una calibración mas amigable y la obtención de aun mejores desempeños. Para ello, disponer de un modelo que permita evaluar alternativas de control, resultaría de un valor incalculable.

3.2 Formulación del Problema

Desarrollar un modelo de tiempo real de una unidad de control destinada a conducir ensayos de motores de combustión interna en forma completamente automática.

3.3 Objetivos de la Investigación

Objetivos generales:

Desarrollar, implementar y evaluar un simulador de tiempo real de una unidad de control destinada a conducir ensayos de motores de combustión interna en forma completamente automática.

Obejetivos especificos:

a. Organizacional o práctica.

b. Los resultados de la investigación serán aplicados en beneficio conducir ensayos de motores de combustión interna en forma completamente automática.

c. Economía y Socio-Política.

En el aspecto económico se tratará de minimizar costos y formular un modelo de presupuesto que servirá para futuros diseños conducir ensayos de motores de combustión interna en forma completamente automática.

d. Metodología.

Los procedimientos de recopilación de información, su compilación, análisis, comparación y evaluación serán sistematizados en el presente Proyecto de investigación conjuntamente con los métodos, técnicas, estrategias y métodos de medición de posibles impactos.

e. Magnitud.

En cuanto a la extensión geográfica donde se desarrollará el Diseño del plan de estudio, tomaremos como referencia la información recopilada en el sitio de ensayos de motores de combustión interna en forma completamente automática.

f. Intelectualidad.

Para nosotros es importante desarrollar esta investigación, porque nos permite aplicar nuestros conocimientos teóricos obtenidos en la Universidad e interrelacionarlos con la experiencia práctica lograda en la industria en beneficio de la colectividad mitigando posibles impactos ambientales.

3.5 Limitaciones y Alcances

Limitaciones

El modelamiento de una unidad de control para ensayos de motores de

combustión en tiempo real involucra la respuesta dinámica continua de un conjunto mecánico y la acción de una unidad digital de control que opera a partir del muestreo discreto de los valores reales de velocidad y carga. Estos sistemas de control, que tienen la finalidad de operar sobre dominios continuos, dan lugar a un tipo particular de sistema denominado "híbrido", que combinan componentes discretos y continuos que interactúan entre sí. Estas dos dinámicas coexisten y es necesario representarlas con modelos que refleje esta interacción, reconociéndose que las mayores dificultades se originan en la necesidad de sincronizar sus comportamientos.

Alcances

El presente estudio permite disponer de un modelo de una unidad de control para ensayos de motores de combustión: "unidad de control digital" de un valor incalculable.

Este modelo de unidad de control facilitaría la exploración de alternativas y permitiría, además de entender mejor algunos fenómenos de acomplamiento e interacción de difícil reproducción en instalaciones reales.

Predecir el desempeño de un conjunto motor-freno y evaluar el comportamiento de su unidad de control permitiendo estudiar condiciones de inestabilidad del sistema.

CAPITULO IV

IV. FUNDAMENTO TEORICO

4.1 Antecedentes del Estudio

Si bien la simulación de motores de combustión interna no es nueva, ya que desde los años '70 se vienen registrando experiencias en este campo (Borman, 1971), es a partir de los '90 que se produce un gran auge en este

tipo de modelos, de la mano de los avances de la tecnología informática y del interés comercial que despiertan. Los objetivos perseguidos con estos modelos son muy diversos, pero tienen en común la necesidad de aportar información adicional a la obtenida con ensayos experimentales (Rubin, 1997). Entre otros beneficios, esto permite reducir el ciclo de desarrollo de nuevos productos, explorar nuevas soluciones y evaluar performances en dominios inaccesibles o en circunstancias en que los ensayos son muy costosos. Más recientemente, son los estudios orientados a la reducción del consumo y las emisiones nocivas de motores los que se han convertido en uno de los principales objetivos de estos modelos (Brace, 2001).

Se identifica así, en el dominio del problema planteado, a un sistema que involucra la respuesta dinámica continua de un conjunto mecánico y la acción de una unidad digital de control que opera a partir del muestreo discreto de los valores reales de velocidad y carga. Más adelante, se describirán las características de este sistema de control y será necesario evaluar un modelo capaz de representar el comportamiento dinámico de un motor acoplado a un freno dinamométrico.

4.2 Marco Teórico

Se presenta a continuación una formulación teórica de los sistemas híbridos de control que respaldará el modelo a ser desarrollado y se comprueba la necesidad de otras herramientas numéricas, según el siguiente detalle:

- El fenómeno continuo será representado por un sistema de ecuaciones diferenciales ordinarias que debe ser resuelto en el tiempo, para lo cual se selecciona el método de integración numérica apropiado.
- Estas ecuaciones diferenciales incluyen entre sus términos las fuerzas que resultan de las acciones del motor y freno dinamométrico. Para su representación se recurre a redes neuronales artificiales de perceptrones multicapa.

- El sistema de control recibirá del sistema continuo un muestreo discreto y no sincronizado de las variables de estado. Para cumplir las acciones de control será necesario extrapolar y derivar numéricamente estos valores.

Por lo tanto, los sistemas híbridos de control, la integración numérica de ecuaciones diferenciales, la aproximación de funciones con redes neuronales artificiales y la diferenciación e interpolación numérica conforman el marco teórico de este trabajo.

Sistemas híbridos de control

Los sistemas dinámicos híbridos se caracterizan por la interacción entre procesos continuos y discretos. Se trata de una combinación de sistemas dinámicos continuos o discretos en el tiempo (time-driven) y sistemas dinámicos de eventos discretos SED's (event-driven), donde la característica distintiva está dada por su vector de estado conjunto w , que contiene variables tanto discretas como continuas.

Un sistema será entonces denominado híbrido si su vector de estado contiene al menos una de cada tipo de estas variables, es decir que $w \in \mathbb{R}^n \times \mathbb{Z}^m$ donde $n > 0$ y $m > 0$.

Aspectos generales

La necesidad de dar a estos sistemas un tratamiento especial fue reconocida hace más de treinta años por Fahrland, quien es considerado un pionero en este campo (Fahrland, 1970). Sin embargo, por mucho tiempo los sistemas híbridos no ocuparon la atención de los investigadores, hasta que en los últimos diez años estos sistemas comenzaron a ser el objeto de un estudio intensivo, tanto por parte de las comunidades de investigadores de Ciencia de la Computación como por las de la de Teoría del Control. En particular, puede comprobarse que se otorga un particular énfasis al objetivo de alcanzar una representación unificada de los sistemas híbridos que se apoye en fundamentos

matemáticos rigurosos (Balluchi, 2000). Sin embargo, debe advertirse que el campo de los sistemas híbridos es extremadamente amplio, incluyendo los problemas de control de sistemas continuos y de control de sistemas discretos como casos particulares. Por lo tanto, es de prever que no será fácil que se llegue a establecer una estrategia común para abordar todos estos casos.

Tradicionalmente, los sistemas híbridos han sido representados por diferentes formas de autómatas, cuyos estados son asociados con ciertas ecuaciones diferenciales y las transiciones son definidas a partir de condiciones impuestas sobre las variables del vector de estado. Las dos principales corrientes para estudiarlos se desarrollaron en torno a las Redes de Petri y a los Autómatas Finitos. La teoría de la Red de Petri fue desarrollada por Carl Adam Petri, Anatol Holt y otros, siendo el primero de ellos quién posteriormente la presentó en su tesis doctoral. Se trataba de un nuevo modelo, destinado al estudio de flujos de información en sistemas, que es aún hoy exitosamente utilizado para representar fenómenos asincrónicos, concurrencia, paralelismo y sincronización de actividades.

Estas Redes de Petri fueron luego extendidas para adecuar sus modelos a la representación de fenómenos de Tiempo Real, dando lugar a numerosas variantes que se designan bajo la denominación general de Redes de Petri Temporales (Murata, 1989).

Aproximadamente en esa misma época se introdujeron los modelos denominados Continuos (Alla y David, 1987), que forman parte de una evolución en la que se amplió el formalismo de las Redes de Petri para adecuarlas a la representación de Sistemas Híbridos. Se presentaron luego las Redes de Petri Híbridas (Le Bail, 1991) y las Redes de Petri Diferenciales (Demongodin, 1996 y Champagnat, 1998).

El enfoque alternativo para la representación de modelos de tiempo real surgió con la presentación de los Autómatas Híbridos (Alur, 1993 y 1995). Numerosas corrientes, que pueden ser agrupadas en dos clases, se derivaron de aquel trabajo de Alur. En la primera se encuentran los

regiones, es decir, los subdominios en que las ecuaciones diferenciales son aplicables.

El segundo caso corresponde al de controladores conmutables que operan sobre sistemas continuos. La principal razón para optar por controladores de este tipo es la obtención de la mejor performance posible en todo su rango de utilización; lo que implica buena respuesta en todos los regímenes estacionarios y una rápida transición al cambiar de una condición a otra. Además, es esencial un apropiado reconocimiento de su naturaleza híbrida en la obtención de un grado importante de autonomía. Estos sistemas conmutados son sistemas dinámicos híbridos que constan de una familia de subsistemas continuos o discretos en el tiempo y un conjunto de reglas que determinan la conmutación entre ellos. Cabe señalar que, en algunos casos, la conmutación entre controladores es necesaria para asegurar condiciones de estabilidad y que, en otros casos, se utiliza para cumplir eficientemente la secuencia de operaciones requeridas. En estos sistemas, un elemento supervisor selecciona el tipo de control apropiado a partir de la información recibida del proceso y de las señales de control requeridas para atender cada caso.

En aquellos casos donde intervienen distintos subsistemas suele diseñarse un autómata híbrido para cada subsistema, incluido el controlador y se obtiene luego mediante una composición paralela un autómata híbrido para el sistema completo.

Obsérvese que los sistemas híbridos de control aparecen con la incorporación de las computadoras en la supervisión de procesos continuos, como es el caso, por ejemplo, de los procesos químicos y de fabricación.

La consideración del problema desde el punto de vista de la modelización, permite señalar que, en muchos sistemas reales, sus principales componentes son claramente identificables. En estos casos la unidad de control, las interfaces y el subsistema controlado encuentran en los sistemas híbridos una representación funcional adecuada para su estudio formal, ya que cada componente está representado de manera natural. Desde ésta óptica puede ser interpretado el esquema de la Figura 1.

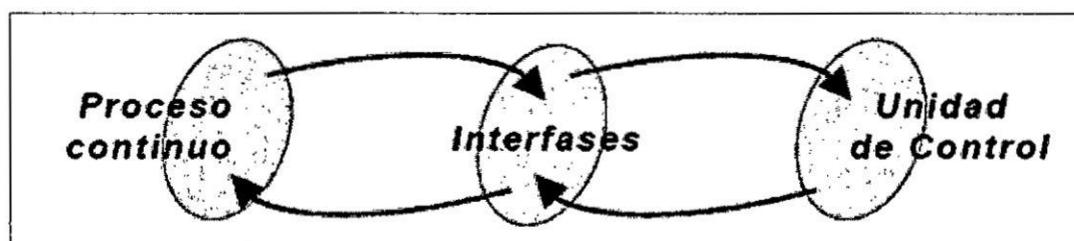


Figura N° 01 Sistema híbrido de control

Se presentan otros casos en los que tal separación no es tan clara y esta representación es una abstracción. En estos, el objetivo no suele ser la implementación de una estrategia de control, sino más bien la identificación de sus propiedades y la facilitación de su estudio. Aun en estos casos, una descripción de este tipo puede ser útil para una mejor comprensión de sus propiedades esenciales, las que están relacionadas con la interacción de las dinámicas continuas y discretas.

El desarrollo presentado permite orientar el estudio hacia el problema planteado, el control del ensayo de un motor de combustión interna, concentrándose la atención en la representación del objeto controlado, la unidad de control y sus interfases.

Para ello, se presenta a continuación la formulación teórica que respalda el estudio de estos elementos, para lo cual se tomaron como referencia los lineamientos propuestos en numerosos trabajos por los ya mencionados Michael Branicky (Branicky,1998), Raymond Decarlo (Decarlo,2000) y Xenofon Koutsoukos (Koutsoukos, 1999 y 2000) referidos a la supervisión y control de sistemas híbridos.

Lógicas de control

Se concentrará ahora la atención en las propias lógicas de control, cuya finalidad es generar las señales z destinadas a reducir y mantener estable el error representado por el vector e , tal como fue expresado por la Ecuación 17.

Con este fin, en la mayoría de los sistemas industriales de control, y en particular en el ensayo de motores, se implementan lógicas denominadas PID - Proporcional, Integral y Derivativa. Los conceptos asociados a este

tipo de control tienen antecedentes muy remotos, pero no fue hasta el trabajo de Minorsky (1922) referido a la conducción de barcos, que el control PID cobró verdadera importancia teórica. En la actualidad, el controlador PID es aún el más utilizado en la industria moderna, a pesar de la abundancia de sofisticadas herramientas y métodos avanzados alternativos de control. Es probable que esta gran difusión y popularidad en aplicaciones industriales, de la más variada naturaleza, se deba a la simplicidad, versatilidad y eficacia que las unidades de control PID han demostrado (Creus, 1987 y Goodwin, 2001). En la Figura 6, se representa un circuito realimentado básico que corresponde a una de las opciones de conmutación del sistema ya presentado en la Figura 4:

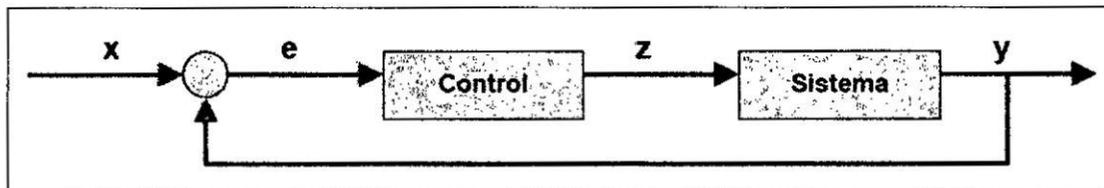


Figura N° 02 Esquema general de Control

La función de transferencia G queda definida como

$$G(t) = y(t) / z(t)$$

$$\text{y el error es } e(t) = x(t) - y(t)$$

$$\text{luego } e(t) = x(t) - G(t) \cdot z(t)$$

$$\text{donde } z(t) = K_p e(t) + (K_p / T_r) \int_0^t e(t) dt$$

$$+ K_p T_d de/dt$$

En la expresión (32), se observan tres términos que hacen que la señal de control "z" incluya componentes proporcionales al error (P), proporcionales a la integración del error en el tiempo (I) y finalmente proporcionales a su derivada (D), que se definen a continuación:

Acción proporcional (P): la acción de control está directamente relacionada con el error instantáneo medido. Esta acción proporcional

puede controlar cualquier sistema estable pero no es capaz de asegurar que la respuesta estará libre de errores estacionarios.

Acción integradora (I): la acción de control es proporcional al error acumulado en un cierto intervalo de tiempo, lo que implica una acción relativamente lenta. Como contrapartida, tiene la ventaja de asegurar un error estacionario nulo.

Acción derivativa (D): su acción es proporcional a la rapidez con que cambia el error y por esta razón su efecto suele ser definido como predictivo. En efecto, esta acción modera o revierte la acción proporcional cuando todavía no se alcanzó el nivel de referencia deseado pero el ritmo de aproximación es demasiado rápido. De esta forma, se "adelanta" la acción de la variable de control para obtener así una variable de proceso más estable. Esta acción derivativa es denominada a veces "rate action" por algunos fabricantes de controles porque considera la "razón de cambio" en la variable de proceso. Su mayor limitación es su tendencia a generar ajustes de elevada magnitud en respuesta a errores que cambian con mucha rapidez (Voda y Landau, 1995).

La lógica de control PID puede ser representada con un esquema como el presentado a continuación en la Figura 7:

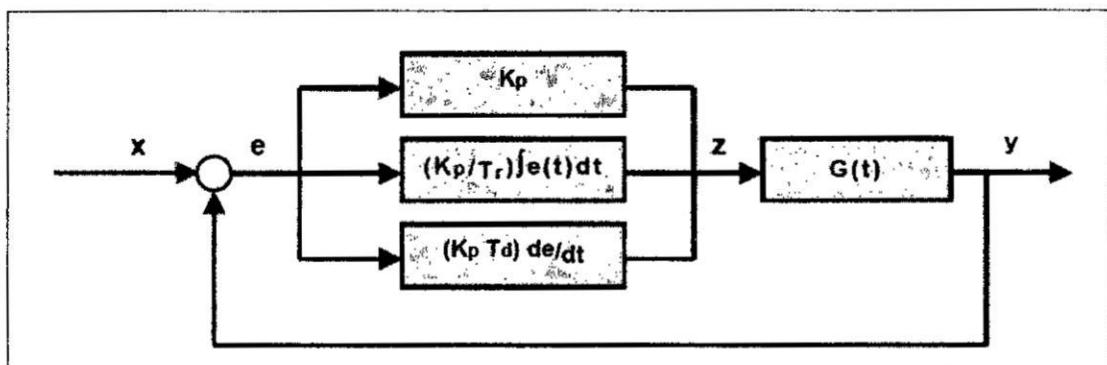


Figura N° 03 Esquema de Control PID

El ya señalado inconveniente que presenta la acción derivativa por su tendencia a generar ajustes de elevada magnitud se pone de manifiesto

ante los cambios bruscos en los elementos del vector x (set-point), casos que pueden ser asimilados a una entrada en escalón.

Una solución interesante para este problema se obtiene al trasladar el componente derivativo, de manera que sea calculado sobre la variable de salida del proceso (realimentada) y no sobre su error. Esta lógica de control es denominada PI-D y es una alternativa cierta cuando los cambios requeridos en las condiciones de operación son muy bruscos. En la Figura 8, se presenta un esquema de un sistema de este tipo.

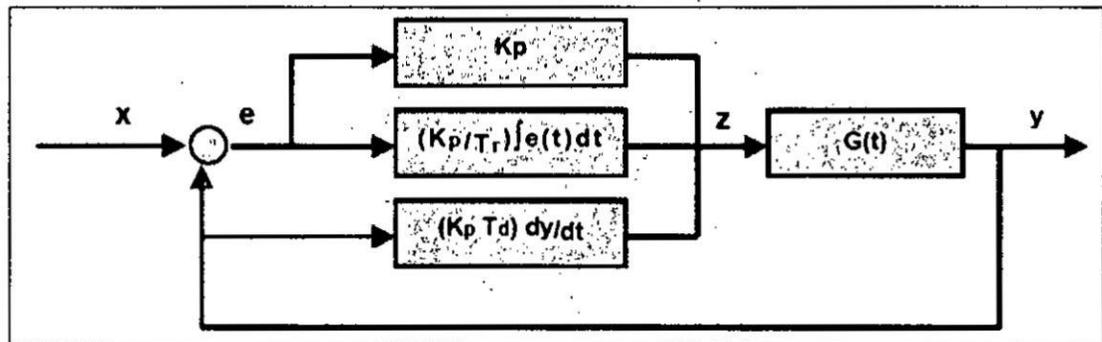


Figura N° 04 Esquema de Control PI-D

En base a ambas soluciones, surgió la idea de una unidad de control más general que incluya ambas componentes derivativas, es decir, que ajuste la variable de control tanto a partir de la realimentación de la variable de proceso (salida del objeto controlado) como también a partir de la realimentación de su error.

Este tipo de solución ofrece la posibilidad de regular en forma dinámica la incidencia de uno y otro componente de control, con el fin de lograr transiciones suaves y condiciones estacionarias más precisas. Obsérvese que el selector de lógicas de control (Figura 4) es el encargado de conmutar entre las diferentes opciones disponibles, incluyendo las opciones de variables de control sobre las que se actúa en cada instante y las realimentaciones que se consideran en cada caso.

Un esquema de un sistema de este tipo se presenta en la Figura 9.

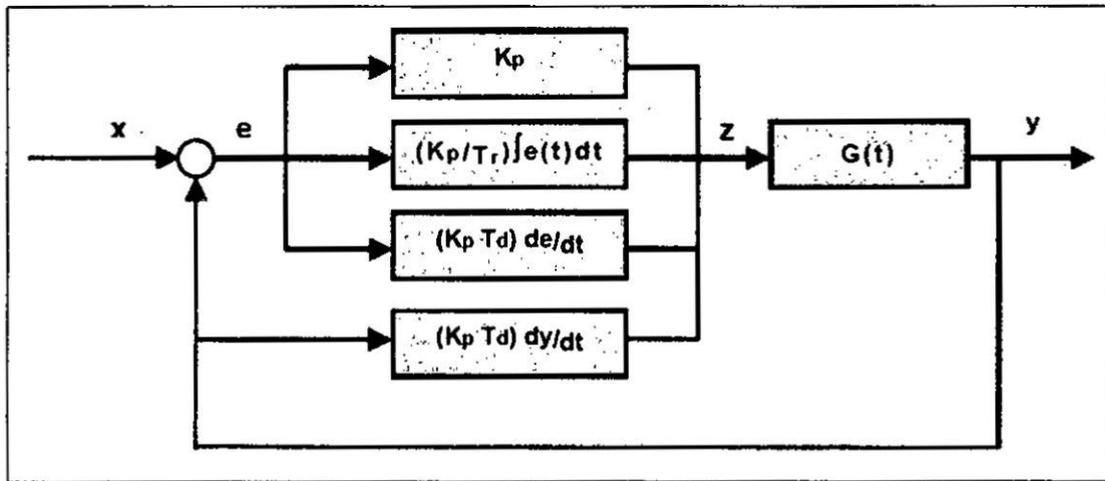


Figura N° 05 Esquema de Control PID-D

En particular, tanto los controles PID como sus variantes son apropiados para los casos en que se desconoce la función de transferencia de la Planta, ya que pueden ser ajustados a partir de técnicas que se apoyan en resultados experimentales. Los métodos más usados son los de Ziegler-Nichols (Ziegler,1942) y de Cohen-Coon. El primero permite ajustar los parámetros de la unidad de control a partir de la respuesta del sistema a lazo abierto y el segundo lo hace considerando la respuesta a una función escalón.

Integración de las ecuaciones del movimiento

Los modelos matemáticos apropiados para representar el comportamiento dinámico de elementos mecánicos quedan expresados por sistemas de ecuaciones diferenciales, denominadas ODE. En efecto, la teoría de las ecuaciones diferenciales permite estudiar los más diversos procesos dinámicos que cumplen con tres condiciones que forman la base de la mecánica de los sistemas discretos y son las siguientes:

- a. Que estén determinados: un proceso se denomina "determinado" si su estado pasado y futuro puede obtenerse a partir de su estado presente. Aquí cobran importancia los teoremas de existencia y unicidad, que postulan la existencia de una solución y que tal solución es única.

- b. Que tengan dimensión finita: el conjunto de todos los estados del proceso conforma su espacio de estados y se denomina de dimensión finita si lo es su espacio de estados.
- c. Que sean diferenciables: un proceso se llama diferenciable si sus estados tienen estructura de variedad diferencial.

A este problema se lo denominado "problema de Cauchy" o de "valor inicial" y es habitualmente resuelto en forma numérica, ya que estas ecuaciones pueden no tener una solución analítica conocida o no resultar práctica su determinación. Más aún, en muchos casos los métodos numéricos representan la única alternativa posible para la resolución de ciertos problemas analíticamente intratables, que normalmente representan fenómenos no-lineales.

Se presenta además otra razón que justifica la amplia difusión que han alcanzado estos métodos numéricos y está referida a la disponibilidad de medios apropiados de cálculo. En efecto, es necesario reconocer que, como ocurrió en todos los campos del conocimiento, los métodos de integración numérica han estado por mucho tiempo postergados hasta que el advenimiento del computador digital y su rápido desarrollo los trajo nuevamente al centro de la escena.

Sistemas rígidos

Sin embargo, el problema a ser resuelto en este trabajo posee una característica particular que debe ser considerada al momento de realizarse la selección del algoritmo. Las ecuaciones diferenciales a ser integradas corresponden a un sistema dinámico que exhibe tanto una respuesta de cuerpo rígido como también oscilaciones que son consecuencia de sus propiedades elásticas. Debe observarse que mientras el movimiento principal puede ser estacionario u oscilar a baja frecuencia, las vibraciones pueden incluir armónicas de frecuencias elevadas.

Inclusive, puede no sólo presentarse el caso en que ambas dinámicas estén presentes en forma simultánea, sino también en que en ciertos instantes de tiempo actúe sólo la componente rápida y en ciertos instantes de tiempo la componente lenta.

Estos sistemas cuya respuesta incluye componentes con muy diversas constantes de tiempo son denominados "rígidos" (stiff systems) por los matemáticos y su estudio ha despertado mucho interés en los últimos años por las dificultades que se presentan en los procesos de integración numérica. En efecto, si se considera la componente dinámica rápida se requiere un paso de integración pequeño para garantizar la estabilidad y exactitud de la solución de las ecuaciones diferenciales. Sin embargo, este paso puede ser demasiado pequeño para la solución de la componente dinámica lenta, y si se lo mantiene constante, llevará a extender innecesariamente la obtención de la solución en el intervalo de interés.

Para estos casos en que se presentan constantes de tiempo muy diferentes, son recomendables los métodos de paso variable, los cuales pueden ajustar el paso de integración de acuerdo a los requerimientos del sistema. Sin embargo, debe observarse que esta idea de utilizar un paso diferente según la naturaleza de la respuesta del sistema tiene un costo extra por la necesidad de efectuar evaluaciones y comparaciones adicionales para determinar su valor apropiado.

Métodos muy usados en la práctica para tratar estos casos son aquellos de pasos múltiples o multipaso de orden superior, tales como es el caso del método predictor-corrector de Milne o el método de Adams. Sin embargo, se ha comprobado que aún estas fórmulas tienen sus limitaciones y no son adecuadas cuando se trata de integrar algunos problemas particulares. Entre éstos, puede citarse el caso en que la matriz Jacobiana tenga algún autovalor con parte imaginaria grande (Alvarez Lopez, 2002). En un intento de superar estas dificultades en los

últimos años se han introducido algunos métodos más generales que combinan técnicas multipaso y de Runge-Kutta.

No obstante, todo lo expuesto, la alternativa de implementar en base al método Runge-Kutta tanto procedimientos de paso variable como de paso fijo demostraron en general muy buenos resultados y son las opciones seleccionadas para este trabajo.

Las constantes que definen todas las expresiones anteriores quedan resumidas en la siguiente tabla

	$x = 1$	2	3	4	5	6
a_{2x}	1 / 4					
a_{3x}	3 / 32	9 / 32				
a_{4x}	1932 / 2197	-7200 / 2197	7296 / 2197			
a_{5x}	439 / 216	-8	3680 / 513	-845 / 4104		
a_{6x}	-8 / 27	2	3544 / 2565	1859 / 4104	-11 / 40	
b_x		1 / 4	3 / 8	12 / 13	1	1 / 2
c_x	25 / 216	0	1408 / 2565	2197 / 4104	-1 / 5	
d_x	16 / 135	0	6656 / 12825	28561 / 56430	-9 / 50	2 / 55
e_x	1 / 360	0	128 / 4275	2197 / 75240	1 / 50	2 / 55

Como contrapartida a las innegables ventajas que ofrecen los métodos de pasos variables para tratar sistemas rígidos, debe advertirse las dificultades que presentan su implementación cuando no se dispone de libertad para alterar el período final de integración, como es el caso de las aplicaciones de tiempo real.

Redes neuronales artificiales

El interés que despiertan aquí las redes neuronales artificiales tiene dos razones, que son enunciadas a continuación.

En primer lugar, se reitera la necesidad de representar el comportamiento del motor y freno dinamométrico, que responden a funciones altamente no lineales y de las cuales sólo se conocen sus evaluaciones experimentales. En ambos casos, se trata de modelos que deben predecir el valor de

funciones a partir de dos parámetros de entrada y que deben cumplir las siguientes condiciones adicionales:

- a. Capacidad para reproducir el comportamiento observado de motores y frenos reales a través del ajuste de sus constantes.
- b. Posibilidad de extrapolar dentro de ciertos límites, en regiones donde se carece de lecturas.
- c. Rápida respuesta para hacer posible su implementación en simulaciones en Tiempo Real.
- d. Posibilidad de incorporar en el futuro nuevas entradas para contemplar otros parámetros (como es el caso de la temperatura del aire admisión).
- e. Tolerancia a cierta dispersión y algunas inconsistencias en los datos, es decir, diferentes valores para las mismas condiciones de entrada.

Esta última condición es particularmente importante ya que los datos disponibles provendrán de mediciones efectuadas en distintos ensayos, y, por lo tanto, es natural esperar que contengan ciertas discrepancias y componentes de error. Es necesario evitar que estos defectos en los datos produzcan oscilaciones o indeterminaciones en la respuesta de los modelos. Para implementar estos modelos se dispone de varias alternativas, tales como la interpolación sobre superficies bilineales, las superficies bicúbicas de Bézier, las superficies B-Spline o los mínimos cuadrados (Rogers y Adams, 1990). Más recientemente se vienen utilizando redes neuronales multicapa de perceptrones, por sus interesantes propiedades como aproximadores universales de funciones. En efecto, estos modelos, que han demostrado sus ventajas en la representación de fenómenos no-lineales y donde los datos disponibles presentan un nivel de incerteza importante, son utilizados exitosamente en la Universidad de Salerno para representar el comportamiento de motores (Arsie, 1998 y 2001) y en otros muchos centros de investigación (Xiao, 1996 y Yin, 2001).

La segunda razón para justificar el interés por las redes neuronales excede los límites previstos para este trabajo. Se refiere al potencial que estas redes ofrecen para implementar sistemas de control adaptativos, que, si bien no serán implementados aquí, ofrecen una promisorio alternativa que merece ser explorada en próximos trabajos.

El esquema clásico de control adaptativo, capaz de ajustar sus parámetros durante la operación del sistema (en línea) para brindar una mejor respuesta, se apoya en la hipótesis del comportamiento lineal del objeto controlado. Aquí las redes neuronales han demostrado ser una opción muy apropiada que permiten superar esta limitación y extender el alcance de estos sistemas a casos severamente no-lineales (Tan, 1997). Estas ventajas son aún más evidentes cuando se trata de sistemas MIMO (Multi Input-Multi-Output), que como se observará es el caso del control de ensayos de motores (Pham, 1995).

Un controlador neuronal de este tipo invariablemente involucra una red que debe ser entrenada para reproducir la dinámica inversa de un proceso (Isasi, 2004). Retornando a la Ecuación 29, puede expresarse la salida de un proceso dinámico a partir de la señal de control $z(t)$ y de la función de transferencia $G(t)$:

$$y(t+1) = G(t) \cdot z(t)$$

Luego, la relación inversa es una expresión de la forma:

$$z(t) = G^{-1}(t) y(t+1)$$

Por lo tanto, la red neuronal se utiliza para implementar una estrategia de control inverso que al aproximar la relación G^{-1} puede conocerse la señal de control necesaria para alcanzar cierta salida en el sistema controlado.

Red multicapa de perceptrones

La ya mencionada propiedad de "aproximador universal de funciones" de estas redes las hacen capaces de encontrar cualquier tipo de relación no

lineal múltiple entre las variables de un vector de entrada y los elementos de un vector de salida.

Esta propiedad que exhiben las redes de perceptrones, con estructuras de por lo menos tres capas, de ser capaces de reproducir cualquier función genérica continua fue formalmente demostrada por Kolmogorov y otros varios investigadores. Más aún, este autor estableció que una función entre dos vectores de dimensiones n (entrada) y m (salida) puede ser apropiadamente representada por una red con n unidades en la capa de entrada, un mínimo de $2n+1$ unidades en la capa oculta y m unidades en la capa de salida. Cabe acotar que la denominación de "oculta" obedece a que en esta capa sus unidades no están en contacto directo con las entradas ni con las salidas. Una de estas redes es presentada esquemáticamente a continuación en la Figura 10, donde se representa una arquitectura con dos unidades en la capa de entrada, una unidad en la capa de salida y dos capas ocultas. En lo sucesivo se denominarán N y M a las cantidades de unidades en estas capas.

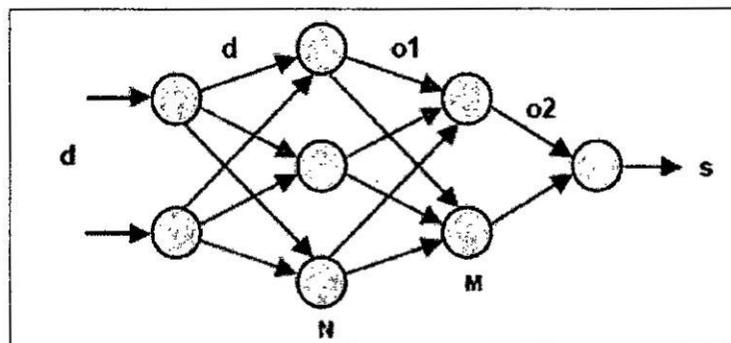


Figura N° 06 Red de perceptrones con dos capas ocultas

Arquitectura

La estructura mínima propuesta por Kolmogorov debe normalmente ser incrementada a efectos de lograr un proceso de convergencia más rápido para el entrenamiento de la red. No se presentan, sin embargo, para ello criterios formales ni prácticos de aplicación general y la estructura más conveniente para cada caso debe ser obtenida a partir de sucesivas pruebas.

Debe observarse que en las redes de perceptrones las conexiones siempre están dirigidas hacia adelante y todas las unidades de una capa están conectadas a todas las neuronas de la capa siguiente, prosiguiendo así hasta la capa de salida. No es posible demostrar formalmente que si se utilizan estructuras que escapen a esta regla se pueden obtener mejores resultados, aunque en algunos casos esto ha podido ser comprobado. Estas variantes se obtienen eliminando conexiones o incorporando conexiones que vinculen capas no contiguas. Por el contrario, nunca pueden implementarse vínculos hacia atrás o de realimentación ya que éstos desvirtuarían la naturaleza de este tipo de redes.

Entrenamiento de la red

Para lograr que la red de perceptrones produzca los resultados correctos es necesario que los pesos sinápticos tengan los valores apropiados y éstos deben ser obtenidos a través de un proceso de ajustes sucesivos, denominado "entrenamiento". Si bien desde los primeros años en que se presentaron estas redes se propusieron diversas variantes para ello, la verdadera revolución en el campo de las redes neuronales se produjo a mediados de los años '80 con la presentación que hizo Rumelhart del "Método de propagación hacia atrás" o "Backpropagation" (Rumelhart, Hinton & Williams, 1986). Aquí es necesario señalar que, al margen de los méritos de Rumelhart, la aplicación de este tipo de algoritmos se hizo posible recién cuando las computadoras alcanzaron la capacidad y rapidez necesaria.

El método de propagación hacia atrás prevé un proceso iterativo donde la diferencia entre la salida esperada (dato experimental) y la obtenida de la red es reconocida como un error que pone de manifiesto que los pesos sinápticos no son los apropiados para representar la función y que estos pesos deben ser ajustados. Este ajuste progresivo es realizado calculando el error a la salida y proyectando ese error hacia las capas anteriores,

pudiendo ser ejemplificado en el diagrama de la Figura 11 presentada a continuación:

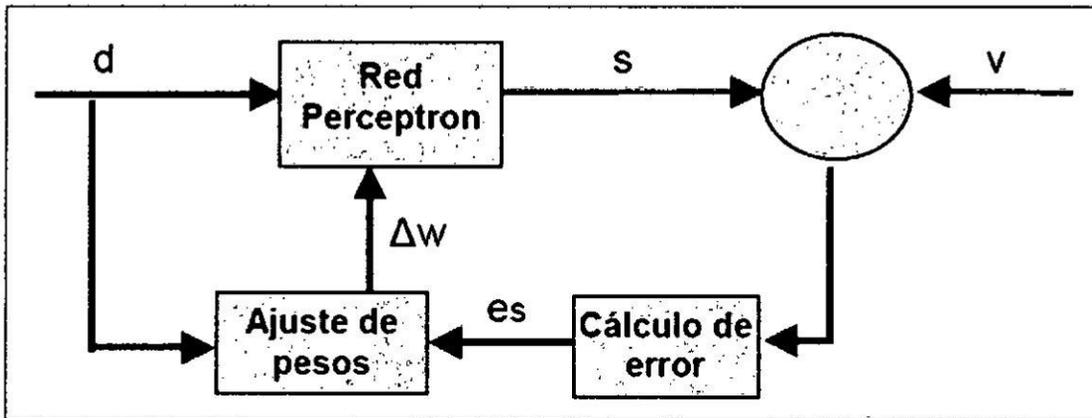


Figura N° 07 Ajuste de pesos de la red

La técnica clásica de "backpropagation" no es más que una implementación del método del gradiente descendente, en el que se procura encontrar el mínimo de una función error en el hiperespacio de los pesos sinápticos de la red, como se representa en la Figura 12.

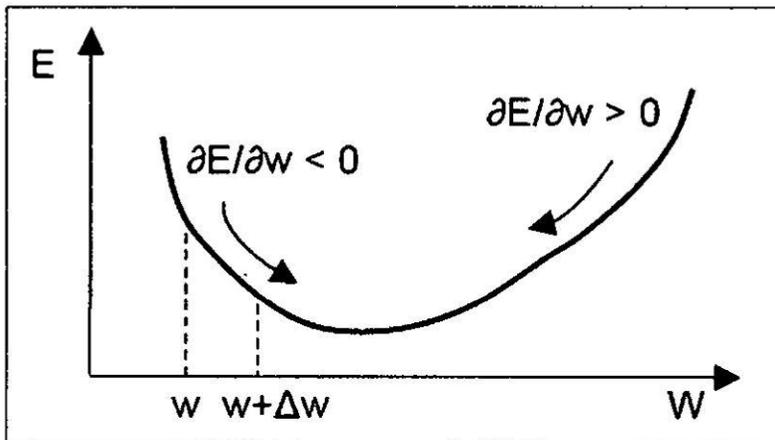


Figura N° 08 Método del gradiente descendente

Para ello se determina en primer lugar el error en la salida de la red, que a partir de la correspondiente formulación diferencial (Nilsson, 2001) queda expresada por:

$$\{es\} = (\{v\} - \{s\}) \cdot \{s\} \cdot (\{1\} - \{s\})$$

y se determinan luego los errores de los vectores salida de las sucesivas capas ocultas, desde la última hasta la primera. Volviendo al ejemplo de la Figura 10:

$$\begin{aligned} & \{u_2\} = [W_{os}]^T \{e_s\} \\ \text{y} & \{e_{o2}\} = [u_2] \cdot [o_2] \cdot (\{1\} - \{o_2\}) \\ & \{u_1\} = [W_{oo}]^T \{e_{o2}\} \\ \text{y} & \{e_{o1}\} = [u_1] \cdot [o_1] \cdot (\{1\} - \{o_1\}) \end{aligned}$$

Una vez determinados los errores de los vectores de salida, se ajustan las matrices de pesos sinápticos según las expresiones:

$$\begin{aligned} & [\Delta W_{os}] = \eta \{e_s\} \cdot \{o_2\}^T \\ \text{y} & [W_{os}] = [W_{os}] - [\Delta W_{os}] \\ & [\Delta W_{oo}] = \eta \{e_{o2}\} \cdot \{o_1\}^T \\ \text{y} & [W_{oo}] = [W_{oo}] - [\Delta W_{oo}] \\ & [\Delta W_{do}] = \eta \{e_{o1}\} \cdot \{d\}^T \\ \text{y} & [W_{do}] = [W_{do}] - [\Delta W_{do}] \end{aligned}$$

donde η , denominado "factor de aprendizaje", es un parámetro que determina la magnitud del desplazamiento sobre la superficie del error y define por lo tanto la velocidad de convergencia del algoritmo. Se presenta aquí uno de los puntos débiles más importantes de este método de aproximación de funciones. En efecto, para alcanzar la necesaria precisión durante el ajuste de los pesos sinápticos es necesario asignar valores bajos al factor de aprendizaje η . Esto permite evitar oscilaciones, pero conduce a procesos de convergencia muy lentos.

Procurando superar estas dificultades, se incorpora en cada ajuste de pesos un porcentaje del ajuste del ciclo anterior. Esta corrección incorpora cierta inercia en el proceso de ajuste y a este factor de incidencia se lo denomina "momentum" β . Sin embargo, aún así y dependiendo de las características de la superficie que representa la función error, el proceso de aprendizaje puede quedar atrapado en mínimo locales o planicies.

Además, la necesidad de asignar valores apropiados a los factores η y β incorpora nuevos parámetros que deben ser definidos mediante un proceso exploratorio, tal como ocurre con la selección de la arquitectura más conveniente. Esto dificulta el proceso de entrenamiento y hace imposible asegurar que se ha encontrado la solución óptima.

Otros métodos de entrenamiento

En los últimos años se vienen aplicando diversas técnicas del análisis numérico (Valishevsky, 1998) con el fin de superar las dificultades señaladas y acelerar los procesos de convergencia. Algunas de ellas, como el método "QuickProp", recurren a la curvatura de la función error determinando su segunda derivada. En este caso, las ventajas son discutibles, ya que no es posible asegurar que el esfuerzo de cálculo extra que implica la determinación del ajuste de pesos será compensado con una reducción en los ciclos de cálculo que son necesarios para entrenar la red.

Otro método destacable es el algoritmo "Rprop" cuyo nombre es un acrónimo de "Resilient backpropagation" (Riedmiller, 1994). Este método difiere de la técnica de propagación hacia atrás clásica en que las derivadas parciales de la función error sólo son usadas para determinar el sentido en que deben ser corregidos los pesos de la red, pero no las magnitudes de los ajustes. A estos ajustes se les asigna un valor inicial que es luego corregido en cada ciclo de cálculo con el siguiente criterio:

el ajuste de pesos en cada paso de cálculo "p" es: $\Delta w^p = \mu \cdot \Delta w^{p-1}$

donde si $\frac{\partial E^{p-1}}{\partial w} \frac{\partial E^p}{\partial w} < 0 \Rightarrow \mu = 0,5$ y si $\frac{\partial E^{p-1}}{\partial w} \frac{\partial E^p}{\partial w} > 0 \Rightarrow \mu = 1,2$

lo que significa que se reduce el tamaño del ajuste en caso de encontrarse un mínimo de la función error y aumentado en caso contrario (Figura 12).

Se presentan algunos métodos más recientes que proponen estrategias de ajuste no monótonas que permiten que la función error se incremente en ciertos puntos, conduciendo a procesos que, aunque moderadamente

oscilantes, siempre son estables, confiables y normalmente más rápidos (Plagianacos, 2002). En estas técnicas, se determina en forma dinámica el valor del factor de aprendizaje η más apropiado en cada caso, presentándose propuestas tales como el "BPVS" (Back Prop Variable Stepsize):

$$\eta = \frac{1}{2} \cdot |w^p - w^{p-1}| / |\Delta E(w^p) - \Delta E(w^{p-1})|$$

o el "ABP" (Adaptive Back Propagation) donde:

$$\eta^p = \begin{cases} \mu \cdot \eta^{p-1} & \text{si } E(w^p) < E(w^{p-1}) \quad ; \quad \mu > 1 \\ \delta \cdot \eta^{p-1} & \text{si } E(w^p) > E(w^{p-1}) \quad ; \quad 0 < \delta < 1 \\ \eta^{p-1} & \text{si } E(w^p) = E(w^{p-1}) \end{cases}$$

Por último, deben mencionarse las técnicas que trabajan sobre la definición de los pesos iniciales. Para la mayoría de los casos anteriores se deben adoptar valores aleatorios pequeños, mientras que una alternativa es encontrar la mejor distribución de pesos iniciales, según la naturaleza de la función reproducida y la arquitectura de la red. Estos pesos iniciales procuran que cada unidad tenga el mejor desempeño en el proceso de entrenamiento, acelerando la convergencia y evitando fenómenos de saturación prematura (Wlodzislaw, 1997). En algunos casos, se propone un incremento progresivo en las unidades de la red con determinación de los pesos iniciales de las nuevas unidades a partir de los pesos en la red anterior (Delashmit, 2002 y 2003).

Conducción del proceso de entrenamiento

Al evaluar el comportamiento de una red de perceptrones no sólo es necesario saber si la red fue exitosamente entrenada, también es indispensable comprobar el comportamiento de la red ante patrones no utilizados en el proceso de aprendizaje. Esto significa que, durante el entrenamiento, la red debe extraer de las muestras las características de la

función a representar y no memorizar los vectores utilizados en este proceso, como se ilustra en la Figura 13.

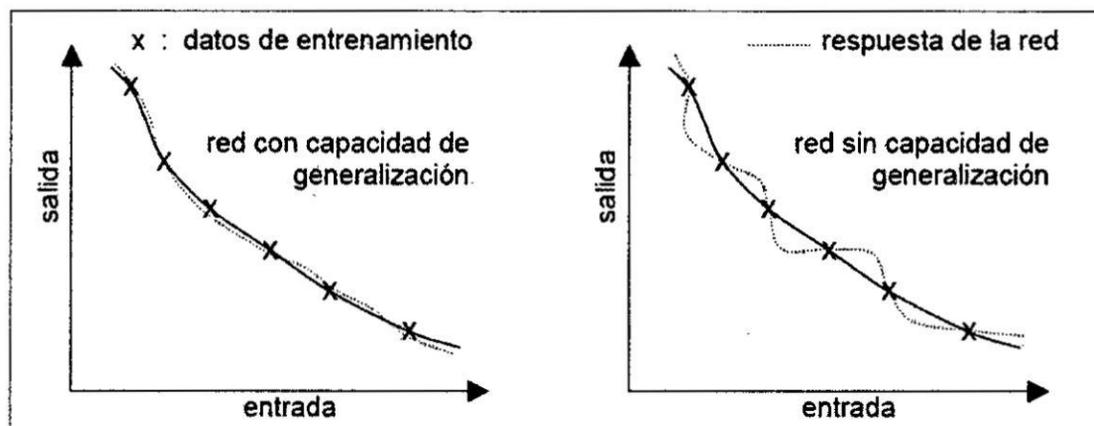


Figura N° 09 Capacidad de generalización de las redes neuronales

Esta propiedad de la red es denominada “capacidad de generalización” y debe ser comprobada en todos los casos para asegurar que la red brindará la respuesta esperada.

Por lo tanto, cuando se realiza el proceso de aprendizaje, es muy importante comprobar paralelamente su capacidad de generalización. Para ello es necesario dividir al conjunto de pares de entrenamiento en dos grupos, uno de ellos para ajustar los pesos de la red y el otro de validación para comprobar su generalización. Más aún, es recomendable repetir el proceso de entrenamiento intercambiando el rol de ambos grupos de datos o inclusive formar nuevos grupos fraccionando los anteriores. En la Figura 14 se representan las dos posibilidades que presenta la evolución de los errores en el proceso de aprendizaje.

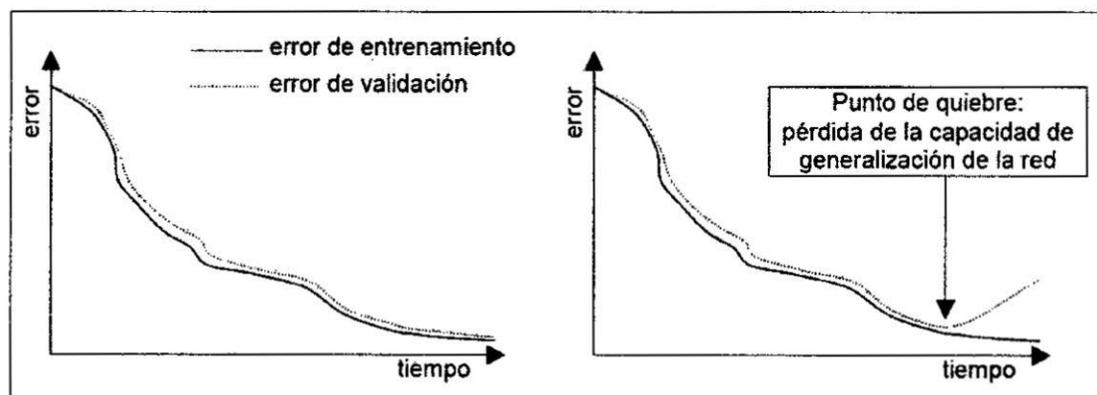


Figura N° 10 Evolución del error en el proceso de entrenamiento

Una grave equivocación, frecuentemente cometida, es la de evaluar solamente el error de entrenamiento, con lo cual puede ocurrir que no se advierta que la red ha perdido generalidad y se continúe el proceso de ajuste de pesos.

Aproximación de funciones

Tal como fue reseñado en el punto 2.1.4, el proceso y la unidad de control pertenecen a diferentes dominios. Por este motivo, los valores del vector de estado y del proceso sólo pueden ser conocidos desde el sistema de control a través de interfases, que se materializan mediante sensores y conversores A/D. Si bien la conversión de señales será desarrollada más adelante en el punto 4.5, es fácil advertir que difícilmente habrá simultaneidad en las mediciones de las diferentes variables, ya que sus frecuencias de muestreo serán distintas, o como mínimo, estarán desfasadas en el tiempo.

Sin embargo, en la unidad de control se deben conocer los valores de todos los componentes del vector de estado para un mismo instante de tiempo; a su vez esta unidad operará con su propia frecuencia, por lo que es necesario prever un procedimiento de extrapolación de estas funciones. Es necesario conocerlas en cierto instante y además, para algunos de los criterios incorporados en la unidad de control, es necesario conocer también las derivadas de estas variables.

Todo esto justifica la necesidad de disponer de un método eficiente para extrapolar y derivar funciones a partir de un muestreo discreto, tal como se presenta en la Figura 15; y para ello se analizaron diferentes posibilidades.

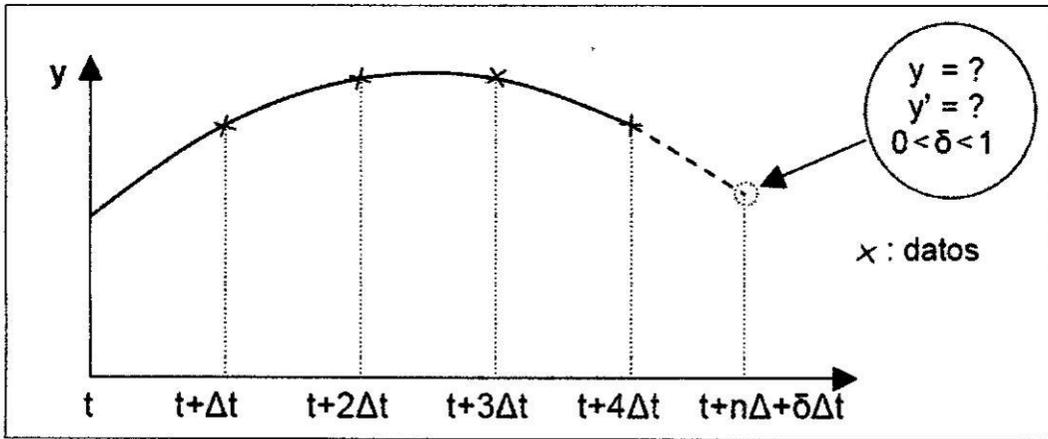


Figura N° 11 Aproximación de funciones

Tras evaluar las opciones disponibles, se seleccionó una aproximación por mínimos cuadrados con un polinomio de tercer grado como el método más conveniente. Aquí se propone un polinomio de grado bajo por ser compatible con la evolución que se espera de la función, y para el caso de la derivada, también se espera acotar el riesgo de error como consecuencia de alguna oscilación en el extremo extrapolado.

En realidad, la regresión por mínimos cuadrados es utilizada cuando los datos incluyen errores importantes e inclusive se dispone de numerosas lecturas para un cierto instante de tiempo. Aquí se presenta la primera de estas condiciones cuando el sistema de control trabaja en una instalación real, pero no es el caso del simulador, en que los elementos del vector de estado son generados por otro proceso. Sin embargo, aún así ésta es una buena elección para extrapolar y obtener la derivada de las funciones.

Asumiendo entonces que la función será aproximada por una ecuación de tercer grado:

$$Y = a_0 + a_1 \cdot t + a_2 \cdot t^2 + a_3 \cdot t^3$$

puede determinarse el error que se comete cuando se la utiliza para representar un conjunto de n pares de datos (y, t) :

$$E = \sum_1^n (y_k - a_0 - a_1.t_k - a_2.t_k^2 - a_3.t_k^3)^2$$

Los coeficientes pueden ser evaluados haciendo mínima la función que representa el cuadrado del error (McCraquen,1964 y Chapra,2004), con lo que se obtienen las ecuaciones:

$$\partial E / \partial a_i = 0; i = 0, 1, 2, 3$$

que conducen a la expresión matricial:

$$B. a = c$$

donde los elementos de la matriz B y del vector c quedan definidas como:

$$B_{ij} = \sum_1^n t_k^{i+j-2}$$

$$c_i = \sum_1^n y_k.t_k^{i-1}$$

y en forma extendida el sistema queda expresado como:

$$\begin{pmatrix} n & \sum_1^n t_k & \sum_1^n t_k^2 & \sum_1^n t_k^3 \\ \sum_1^n t_k & \sum_1^n t_k^2 & \sum_1^n t_k^3 & \sum_1^n t_k^4 \\ \sum_1^n t_k^2 & \sum_1^n t_k^3 & \sum_1^n t_k^4 & \sum_1^n t_k^5 \\ \sum_1^n t_k^3 & \sum_1^n t_k^4 & \sum_1^n t_k^5 & \sum_1^n t_k^6 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} \sum_1^n y_k \\ \sum_1^n y_k.t_k \\ \sum_1^n y_k.t_k^2 \\ \sum_1^n y_k.t_k^3 \end{pmatrix}$$

de donde se obtienen los coeficientes de la parábola cúbica resolviendo el sistema de ecuaciones. Es decir:

$$a = B^{-1}. c$$

Una vez definidos estos coeficientes, la ecuación cúbica (64) permite extrapolar la función y obtener también la expresión de su derivada:

$$y' = a_1 + 2 \cdot a_2 \cdot t + 3 \cdot a_3 \cdot t^2$$

Al trabajarse con cuatro pares de puntos, como es éste el caso, pudo haberse determinado en forma directa una parábola cúbica que pase por todos ellos. Sin embargo, la función extrapolada no es parabólica y al forzarla con esta aproximación pueden generarse oscilaciones que ocasionarían un gran error al extrapolar y un error aún mayor en la derivada.

La regresión por mínimos cuadrados es apropiada para evitar este tipo de fenómenos y además ofrece mayor libertad, ya que permite que en cada caso se utilicen de las cantidades de pares de puntos que parezcan más convenientes.

4.3 Marco Metodológico.

Se retorna, en este apartado, al objeto del presente trabajo, que es la construcción de un modelo computacional destinado a simular la interacción de una unidad de control y un conjunto mecánico, y para ello se seguirán las pautas ya establecidas para el desarrollo de todo sistema de computación. Para comenzar, es conveniente presentar el contexto del problema y para ello se hace referencia al modelo en cascada del ciclo de vida del desarrollo de software, que es representado en la Figura 16. Este modelo prevé una fase inicial de ingeniería de requerimientos, que es seguida por las de diseño, programación y testeo. A esta fase inicial, que está destinada a la identificación, análisis y documentación de los requerimientos, se le asigna en la actualidad particular importancia, por ser la encargada de definir el primer modelo conceptual del sistema a ser desarrollado.

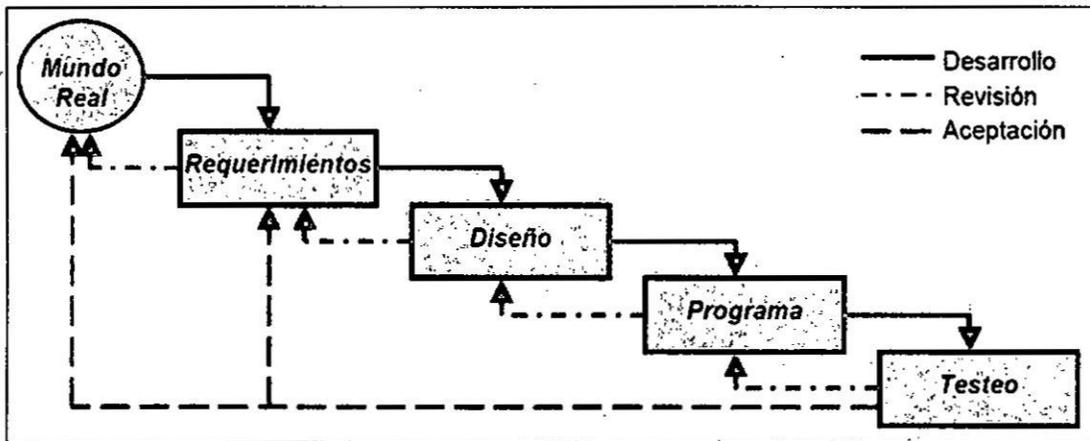


Figura N° 12 Ciclo de vida del desarrollo de software

Si bien el paradigma de objetos ha cambiado la forma en que actualmente se desarrolla software, no son pocos los autores (Bustos Reinoso, 1999) que consideran que es cuestionable su utilización en la fase inicial del modelado conceptual de un sistema; y para tratar este problema se han presentado numerosas y diversas propuestas.

Hacia fines de los años '80, era común encontrar metodologías de análisis y diseño orientado a objetos (Constantine, 1989) que utilizaban técnicas ajenas a los objetos en el modelado conceptual. En estas propuestas, y con algunas dificultades, se identificaban a partir de los DFD y DER los objetos que formarían parte del modelo que luego sería utilizado en la posterior fase de diseño.

Poco tiempo después, al comenzar los '90, varias de las nuevas metodologías sugerían que había que desechar todo modelado que no estuviese orientado a objetos y recomendaban completar íntegramente el ciclo de desarrollo bajo este paradigma. Como ejemplos de esta tendencia, pueden citarse los trabajos de Rebecca Wirfs-Brock (Wirfs-Brock, 1990) y de Kenneth Rubin (Rubin, 1992).

Sin embargo, estas metodologías entraron en conflicto con el principio de la ingeniería de software que establece que el análisis debe ser declarativo al definir "qué" debe hacerse, en contraposición con el diseño que, por ser una

actividad creativa, debe definir “cómo” ha de hacerse. En la práctica, los modelos orientados a objetos surgidos de la fase de análisis sólo son menos detallados que los del diseño, con lo que la distinción conceptual de ambas fases aquí se desvanece. Más aún, hasta podría decirse que los conceptos de análisis (descomposición y examen crítico) y de orientación a objetos (encapsulamiento de atributos y capacidades) en realidad se contraponen (Bustos Reinoso, 1999).

Una forma de resolver este conflicto es postergar el encapsulamiento tanto como sea posible, lo que implica sacar el modelado conceptual fuera del paradigma de objetos, adoptando para ello métodos que dispongan de la necesaria flexibilidad y poder expresivo.

Bajo este razonamiento, se retornó a la utilización de modelos no orientados a objetos en la especificación de requerimientos, para luego derivar de éstos los modelos de objetos. Resulta indudable que para ello contribuyó enormemente la amplia aceptación que experimentó el modelo de Casos de Uso propuesto originalmente por Ivar Jacobson (Jacobson, 1992) y su posterior difusión acompañando a UML (Jacobson, Booch y Rumbaugh, 1999).

No obstante, es necesario recordar que los Casos de Uso recibieron en sus comienzos algunos comentarios bastante desfavorables, que con el tiempo fueron quedando en el olvido. Si bien muchas de estas críticas fueron en su momento relativizadas por provenir de defensores del Análisis Estructurado o del modelado de “puro objetos”, al releerlas se encuentran en ellas mensajes muy interesantes y algunas enseñanzas muy oportunas que mantienen hoy plena vigencia.

Una de estas opiniones desfavorables, que era compartida por numerosos autores, destacaba que los Casos de Uso inducían a perpetuar viejas soluciones al reproducir en detalle sistemas existentes, ya sean

computarizados o no, en lugar de contribuir a encontrar la mejor propuesta para resolver un problema.

Recurriendo a críticas más específicas, puede citarse la opinión de Bertrand Meyer (Meyer, 1997:739), quién puntualizó que los Casos de Uso podían conducir con facilidad a una mala definición de las clases, por lo que solamente los recomendaría para el análisis orientado a objetos en caso de disponerse de un equipo de desarrollo muy experimentado.

Otra crítica se centraba en que los Casos de Uso quitaban flexibilidad a los modelos al enfatizar en los flujos de control una prematura secuencialidad, cuando hay todavía insuficiente conocimiento de un problema. Al respecto, Don Mills (Mills, 2002) destacó que las secuencias de operaciones que se obtenían normalmente representaban restricciones del modelo y no verdaderos requerimientos y que, por el contrario, las verdaderas secuencias deberían ser definidas con mayor nivel de abstracción a partir de precondiciones.

Por su parte, Larry Constantine enfatizó (Constantine, 1995) la necesidad de conocer con profundidad los problemas, dejando al margen elaboraciones innecesarias o restricciones artificiales. Para ello propuso sus "Casos de Uso Esenciales", incorporando a los Casos de Uso de Jacobson los conceptos del Análisis Esencial desarrollado por Steve McMenamin y John Palmer (McMenamin, 1984).

A la luz de todas estas críticas, parece oportuno destacar un enfoque que representó una muy interesante alternativa al UML. Éste fue presentado por Leandro Antonelli (Antonelli, 1999) y condujo a definir las mismas Fichas CRC (Clases, Responsabilidades y Colaboraciones) propuestas por Rebecca Wirfs-Brock (Wirfs-Brock, 1990), solo que formando parte de una secuencia en la que se estableció previamente el Léxico Extendido del Lenguaje (LEL) y la descripción del problema a través de escenarios. Para lo primero se utilizó una metodología propuesta por J.C. Leite (Leite, 1995) a la

que se le incorporaron los escenarios y las heurísticas necesarias para derivar estas Fichas CRC.

Todo lo expuesto permite comprobar que en las últimas dos décadas se ha venido desarrollando un importante cuerpo de teorías, métodos y técnicas que, con mayor o menor éxito, han procurado demostrar que son apropiadas para el análisis, diseño e implementación de sistemas de computación de variada naturaleza.

Sin embargo, y tal como ya fue anticipado en el prólogo de este trabajo, esta afirmación no es completamente válida para los sistemas de computación en general y lo es aún menos al estar referida a sistemas de tiempo real, en cuyo tratamiento todavía se aprecia un importante nivel de desconcierto y es frecuente el uso de metodologías ad-hoc carentes de toda generalidad (Huang, 2004).

Al respecto, ya fue señalado que los sistemas de tiempo real poseen requerimientos críticos que están referidos a su comportamiento temporal, precisión y seguridad, algunos de los cuales pueden ser identificados como requerimientos no funcionales. A esto es necesario agregar la también comentada creciente difusión que estos sistemas vienen teniendo, con su presencia en los más diversos e inclusive insospechados ámbitos del quehacer humano.

Una respuesta a este problema ha arribado por el lado de los métodos formales, que están siendo objeto de un renovado interés que se manifiesta con la presentación de una amplia variedad de propuestas. En efecto, su natural rigurosidad los convierte en una opción muy atractiva para estos casos en que la ambigüedad e inconsistencia son inaceptables.

Sin embargo, a pesar de que muchos de estos métodos demostraron ser apropiados para soportar complejos desarrollos de software, debe reconocerse que su difusión es normalmente muy limitada y aún más escasa

su utilización en la industria del software. Es decir, estas propuestas difícilmente trascienden los ámbitos universitarios y los centros de investigación. Uno de los problemas es la disponibilidad de desarrolladores capacitados en estos métodos, que no crece con la rapidez con que lo hace la demanda de este tipo de software. Además, las necesidades de una formación más sólida y mayor tiempo de entrenamiento los hace demasiado costosos para el desarrollo "ordinario" de sistemas de tiempo real.

Por el contrario, el ya mencionado UML continúa consolidando su gran difusión en todo tipo de desarrollos e inclusive viene ganando popularidad en el campo de los sistemas de tiempo real. En efecto, UML para Tiempo Real (alias UML-RT) ya despierta mucho interés y sus especificaciones están incluidas en la definición de UML 2.0. A pesar de ello, la aplicación de UML-RT en el dominio de los sistemas de tiempo real manifiesta todavía algunos problemas importantes, tales como:

- a. La carencia de una adecuada definición formal dificulta los procesos de verificación o simulación.
- b. A pesar de tratarse de una notación efectiva para el diseño e implementación de sistemas, muestra su mayor debilidad en la definición y representación de sus especificaciones. En particular, no son soportadas algunas conductas tales como la no determinista y las que son consecuencia de la simultaneidad de eventos.
- c. La representación del tiempo y de sus restricciones no son facultades nativas y las mejoras que son introducidas presentan todavía algunas restricciones.

Aquí cabe recordar que UML es un lenguaje de modelado cuya notación permite capturar y comunicar tanto la estructura de objetos como su comportamiento, pero que no fue originalmente concebido para modelar sistemas de tiempo real. Por ello, las limitaciones señaladas vienen motivado diversas líneas de investigación con la finalidad de superarlas. Entre otras muchas, pueden mencionarse las metodologías y herramientas

para el desarrollo de sistemas de tiempo real orientados a objeto basadas en UML propuestas por Becker (Becker, 1999) y Drake (Drake, 2001).

Todo lo expuesto permitió comprobar que no existe un enfoque consolidado para el tratamiento de los sistemas de tiempo real, de donde surgió la idea de explorar un enfoque alternativo que combine diferentes metodologías y que es presentado a continuación

Ingeniería de requerimientos

La primera fase del ciclo de vida del desarrollo de software debe conducir a la elaboración de las especificaciones de sus requerimientos (SRS), como producto final de un proceso iterativo que incluye actividades de elicitación, análisis, documentación, verificación y validación. Las dos últimas actividades están destinadas a la revisión de los requerimientos y cabe aquí citar el difundido ejemplo de B. Boehm (Boehm, 1984) que contribuyó a establecer claramente la diferencia entre ambos conceptos, formulando las siguientes dos preguntas: a) validación: ¿estoy construyendo el producto correcto? y b) verificación: ¿estoy construyendo correctamente el producto? Estas especificaciones (SRS) estarán destinadas a establecer los requerimientos del sistema de manera completa, correcta, precisa, no ambigua y consistente. Para ello, la consideración de estos atributos es esencial realizarla en el marco de las cualidades exigidas para la solución al problema planteado, que por tratarse de un sistema de tiempo real incluye de manera destacada la confiabilidad y la eficiencia en su desempeño. Más aún, es sabido que las especificaciones de los requerimientos de estos sistemas de tiempo real deben contemplar tanto un enfoque estructural como de comportamiento de sus componentes, y para esto último es esencial incluir un detallado estudio de sus interacciones con un riguroso tratamiento temporal.

Se comienza aquí por aceptar que para tratar correctamente un problema es fundamental un preciso conocimiento de la semántica de su vocabulario. Su correcta definición y la de su impacto en el problema contribuirán

indudablemente a la obtención de especificaciones de requerimientos consistentes y carentes de ambigüedades. La veracidad de esta afirmación es aún más evidente en casos como el tratado, en el que se involucran áreas tan diversas como Mecánica, Electrónica, Teoría de Control, Metrología y Ciencia de la Computación.

Léxico extendido del Lenguaje

Por ello parecieron apropiadas las recomendaciones de JC Leite (Leite,1995) y Leonardi (Leonardi,1998) de construir un "Léxico Extendido del Lenguaje" (LEL) como primer paso en la fase de elicitación de requerimientos, para lo cual se deja de lado, por el momento, el entendimiento del propio problema.

El LEL es un meta-modelo diseñado para ayudar a capturar el vocabulario de la aplicación y su entorno, utilizando el lenguaje natural para la representación de sus símbolos. Para ello se describen las nociones e impactos de cada palabra o frase, donde la noción describe su significado y el impacto determina los efectos del uso u ocurrencia del elemento en la aplicación. Es así que, según cada caso, la semántica de cada símbolo se representa con una o más nociones y uno o más impactos. Estos conjuntos de símbolos forman una red que permite representar al LEL en un hipertexto y navegar en él para conocer todo el vocabulario del dominio.

En la descripción de las nociones e impactos existen dos reglas básicas que se deben cumplir simultáneamente y son las siguientes:

- a. **Principio de circularidad:** maximizar el uso de símbolos al definir el significado de otros símbolos.
- b. **Principio del vocabulario mínimo:** minimizar el uso de símbolos externos al lenguaje de la aplicación.

Además, para poder lograr una descripción apropiada de los términos del LEL se debe establecer si éstos son utilizados como sujetos, verbos u

objetos, o si describen situaciones en las frases que los actores utilizan para enunciarlos.

En el presente trabajo, se ha previsto la construcción de un LEL en forma manual a través de un editor de textos. Sin embargo, se advierte la conveniencia de disponer de una herramienta CASE que mediante la tecnología de hipertexto pueda facilitar la administración de la información y su validación. En las Figuras 18 y 19 se representan esquemas globales de la metodología propuesta y el lugar que allí ocupa el LEL.

Verificación y validación

La fase final de desarrollo de un proyecto exige someter la primera versión del producto a una serie planificada de pruebas con el objeto de comprobar su correcto funcionamiento bajo todas las condiciones posibles de operación.

Sin embargo, las pruebas de verificación y validación deben comenzar antes, con el fin de comprobar que los métodos de cada objeto cumplan individualmente con sus responsabilidades. Se trata de un control botton-up que debe comenzar con los métodos más elementales y ascender en la estructura del sistema hasta llegar a la evaluación de todo el sistema en conjunto, tal como es representado en el esquema de la Figura 20. Este control progresivo, que es recomendable en todo sistema, es esencial en sistemas técnicos donde sus métodos son implementados a través de complejos algoritmos.

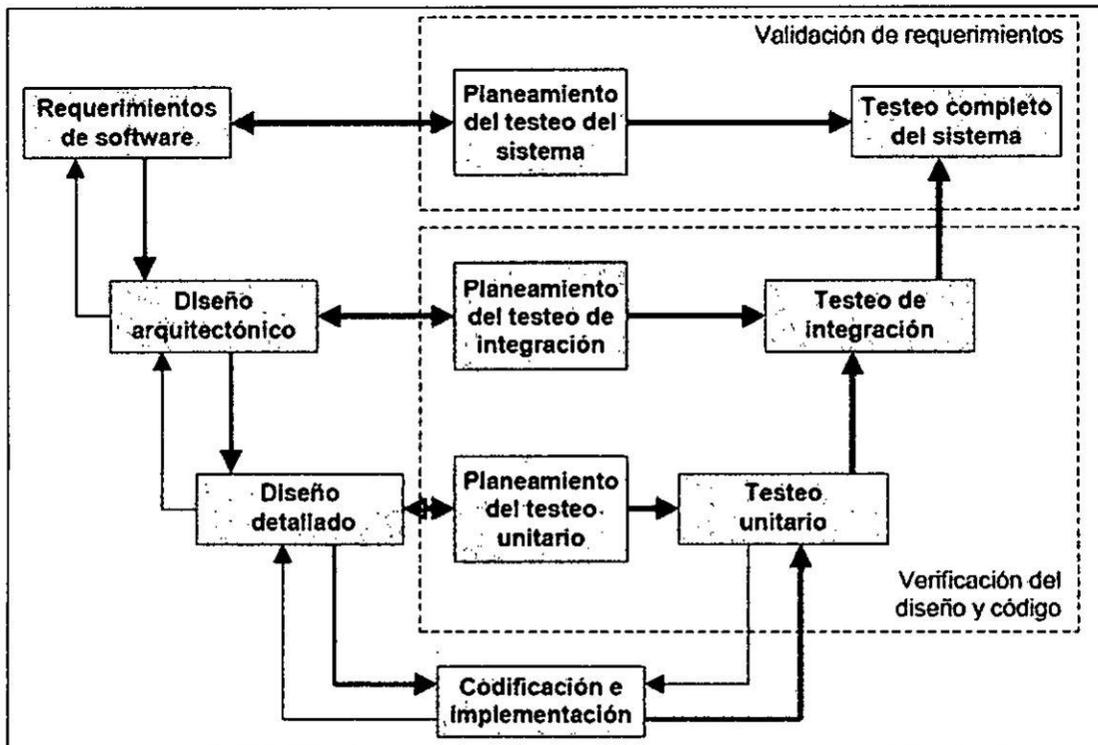


Figura N° 13 El modelo en "V" de desarrollo de software

En este esquema, que representa el modelo en "V" de desarrollo de software (Oliveros, 2001), se muestran con claridad los lugares que ocupan los procesos de verificación y validación de un sistema.

A propósito, parece aquí conveniente volver a las definiciones de los conceptos de verificación y validación, recurriendo para ello nuevamente a los clásicos interrogantes propuestos por Boehm (Boehm, 1984):

Verificación: ¿Estoy construyendo correctamente el producto?

Validación: ¿Estoy construyendo el producto correcto?

De manera natural, se asocia la validación y verificación de un programa con un proceso, adecuadamente planificado, que está destinado a comprobar su correcto funcionamiento. Sin embargo, no puede el testeo dinámico ser usado como único elemento para certificar la corrección de un programa, ya que éste sólo podrá mostrar la presencia de errores, pero no garantizar la ausencia de ellos. En efecto, en la mayoría de los casos es imposible asegurar que han sido verificadas todas las posibles combinaciones de

condiciones de operación, por lo que el testeo dinámico es una condición necesaria pero no suficiente de corrección.

Para superar esta dificultad, las propuestas se vienen orientando a obtener suficiente conocimiento sobre el comportamiento de los programas a partir del examen anticipado de sus diseños y códigos, en lugar de hacerlo a partir del resultado de sus ejecuciones. Se recurre a la inspección y verificación estática para las comprobaciones de corrección.

Se llega así al enfoque moderno de la revisión de un programa, que se refiere al proceso de demostrar anticipadamente que los resultados que proporcionará serán los deseados, o bien detectar la presencia de eventuales errores e identificarlos completamente. Este proceso está orientado a comprobar esta propiedad esencial de todo programa que es su corrección y que puede definirse como "la capacidad de los productos de software de ejecutar sus tareas en concordancia con las definiciones de sus requerimientos y especificaciones".

En este contexto, la verificación formal de programas está destinada a demostrar, usando argumentos matemáticos, que un programa satisface su especificación. Una especificación (E1, E2) de un programa P, establece que, si P comienza a funcionar en un estado que satisface la precondición E1, entonces termina, en un tiempo finito, en un estado que satisface la condición final E2. La corrección se establece a través de la correspondencia entre el comportamiento deseado del programa y su comportamiento real, donde el comportamiento deseado es definido mediante especificaciones y el comportamiento real debe deducirse mediante el análisis del texto. Este análisis se hace en términos de las componentes elementales que conforman el programa. En resumen, todo proceso de verificación formal debe estar dirigido a comprobar algunos aspectos esenciales tales como que la ejecución de un programa siempre termina y que su código es consistente con su especificación.

Es aquí importante establecer dos condiciones que son necesarias para poder realizar una verificación formal basada en un enfoque matemático: a) todas las características del lenguaje de programación, sintácticas y semánticas, tiene que estar definidas formalmente, y b) el programa debe especificarse con una notación que sea consistente con la técnica de verificación matemática usada.

Dadas las exigencias de robustez, la verificación y validación de los sistemas de tiempo real deben ser cuidadosamente planificadas y minuciosamente realizadas. No tienen, sin embargo, especificaciones particulares y su proceso de testeo debe ser conducido en concordancia con las normas establecidas, entre las que se destacan:

- IEEE 829 : Standard para la documentación de los test de software
- IEEE 1008 : Standard para el test de unidades
- IEEE 1028 : Standard para las revisiones de software
- IEEE 1044 : Standard para la clasificación de anomalías de software
- BSS 7925-2 : Standard para el test de componentes de software

Definición de requerimientos

A continuación, se comienza por describir las connotaciones propias del objeto de estudio, con el fin de presentar y conocer el problema desde sus diferentes puntos de vista, para finalmente especificar sus requerimientos. Al hacerlo, se evitan los tecnicismos propios de la mecánica que sean innecesarios para el objetivo de este trabajo.

Ensayos de motores

Una particularidad de los motores de combustión interna es su permanente vigencia. Su historia de más de cien años muestra una progresiva e ininterrumpida mejora en sus prestaciones y rendimientos, a pesar de mantenerse inalterada la concepción mecánica básica que fue anticipada por Beau de Rochas en 1862 y materializada por August Otto en su exitoso primer motor de cuatro tiempos en 1876. Esto no fue obviamente fruto de la casualidad. Mas bien fue el resultado de un persistente y esmerado trabajo de ingeniería, que, con un claro enfoque multidisciplinario, perfeccionó todos los detalles del motor desde los puntos de vista de la cinemática, dinámica, transferencia de calor, termodinámica, metalurgia, resistencia de materiales y fluidodinámica, entre otros. Y acompañando esta evolución fue necesario desarrollar y perfeccionar equipos y técnicas de ensayo, ya que no había otra forma de comprobar los resultados de las sucesivas mejoras que no fuese a través de los ensayos experimentales.

En efecto, si bien la secuencia de operaciones de un motor en funcionamiento está perfectamente definida y tiene un sólido respaldo conceptual, los fenómenos físicos que se presentan son lo suficientemente complejos como para que el desarrollo de los motores haya debido estar siempre apoyado en los test de prototipos.

Es así que los motores de combustión interna fueron y son ensayados experimentalmente. A las mediciones tradicionales de torque, potencia y consumo, entre otras, se suma el análisis de la composición de los gases de escape, que por sus características contaminantes viene adquiriendo

creciente importancia. En la actualidad, estas pruebas se realizan en forma automática y para ello son requisitos esenciales tanto la operación del motor en condiciones preestablecidas de velocidad y carga, con un margen de error muy estrecho, como la posibilidad de pasar con gran agilidad de una condición de operación a otra.

Las instalaciones para cumplir con tales exigencias, denominadas "bancos de ensayos", están dotadas de una bancada para fijar el motor a ser ensayado, un freno dinamométrico, una transmisión cardánica para vincular el motor y el freno, numerosos sensores destinados a medir parámetros de variada naturaleza, un actuador electromecánico para operar el acelerador y un sistema computarizado de supervisión y control. A estos elementos, deben agregarse los circuitos de combustible, refrigeración, ventilación y sistemas auxiliares.

Este conjunto de un motor acoplado a un freno encontrará su equilibrio a diferentes velocidades cuando en un régimen estacionario el torque entregado por el motor iguale la carga absorbida por el freno. Puede establecerse para ello que el motor desarrollará su torque según una familia de curvas, cada una de las cuales corresponde a una condición de alimentación dada por la posición de la bomba inyectora. A su vez, la capacidad de frenado del dinamómetro responderá a otra familia de curvas, donde cada una está asociada a la tensión aplicada en su bobinado en el caso del freno eléctrico o del caudal de circulación de agua en el caso del freno hidráulico.

Para alcanzar rápidamente, en todas las condiciones de marcha, la requerida estabilidad del conjunto motor-freno es entonces necesario un apropiado sistema de control y un cuidadoso ajuste de sus parámetros. Este sistema operará para ello sobre las dos variables básicas de control, que son la alimentación de combustible al motor y la capacidad de frenado del dinamómetro. Aquí, debe acotarse que en un mismo banco se ensayan habitualmente motores de diversa potencia, que un mismo motor cambia su

respuesta según la disponibilidad de algunos accesorios, tales como el turbocompresor o el Intercooler, y que aún con los mismos accesorios, debido a su regulación electrónica, un motor suele mostrar comportamientos muy diferentes a distintos regímenes de velocidad. Por todo ello, el requerido ajuste del sistema de control suele en algunos casos ser una tarea muy laboriosa.

No son, sin embargo, los ensayos la solución a todos los problemas. Muy por el contrario, en muchos casos debe reconocerse que son más bien una nueva e inevitable fuente de dificultades. En una breve enumeración, pueden mencionarse las dificultades inherentes a la medición de parámetros, el tiempo demandado por la construcción de los prototipos e instalaciones de ensayo, el tiempo requerido por los propios ensayos, la necesaria y cuidadosa interpretación de los resultados y los elevados costos involucrados. También es necesario advertir que en estos ensayos se conjugan algunos factores claramente contradictorios. En efecto, se pretende utilizar equipos electrónicos sofisticados para medir señales físicas con elevado nivel de precisión y alcanzar condiciones de marcha estacionaria con márgenes de error muy estrechos, todo ello en un ambiente con muy altas temperaturas, presencia de gases de combustión y elevados niveles de contaminación sonora y vibratoria.

Por otra parte, la complejidad alcanzada en algunos casos por los motores y sus sistemas de control es tan grande que las metodologías tradicionales de diseño basadas en uso intensivo de ensayos han quedado prácticamente obsoletas. Así, al considerarse el alto valor práctico que indudablemente tiene aquí la validación dinámica, es decir las pruebas de buen funcionamiento, debe tenerse presente que sólo es posible asegurar la corrección de un diseño a partir de haber evaluado experimentalmente todas las condiciones de operación posibles. Esto lleva a reiterar que la validación dinámica es una excelente herramienta para comprobar la presencia de errores, pero en la mayoría de los casos no es suficiente para asegurar la ausencia de ellos.

Simulación de ensayos

Por todo lo expuesto, no debe sorprender que siempre haya habido un marcado interés por desarrollar modelos matemáticos capaces de representar los fenómenos físicos involucrados en el funcionamiento de motores. Tampoco debe sorprender que la aparición y perfeccionamiento de las computadoras hayan servido de estímulo a la mejora de tales modelos y a integrar estos modelos en procesos de simulación. En efecto, se comprendió desde un principio el incalculable valor que tendría la posibilidad de evaluar el funcionamiento de motores a través de modelos numéricos correctos y confiables.

Es así que la posibilidad de conducir estos procesos de simulación llevó a replantear el objeto de los ensayos, ya que en muchos casos su finalidad pasó a ser la de suministrar datos para ajustar y validar el desempeño de los modelos, convirtiéndose estos últimos en las herramientas centrales del proceso de diseño.

Además, el desarrollo de modelos y los procesos de simulación trajeron aparejados algunos beneficios colaterales muy convenientes (Boehm, 1973), tales como:

- a) La necesidad de definirlos llevó a una mejor comprensión de los fenómenos físicos involucrados y la interacción entre ellos.
- b) Se sistematizó el conocimiento sobre el objeto estudiado y se lo almacenó en forma accesible.
- c) Se identificaron áreas sobre las que había insuficiente conocimiento y se estimularon nuevas líneas de investigación.
- d) Se aprovecharon mejor los ensayos, reduciéndose la cantidad requerida y favoreciéndose la optimización de los diseños.

La necesidad de asegurar las condiciones para que un modelo y sus resultados reproduzcan acabadamente la esencia y conductas del objeto

real, que es estudiado, se explica si se entiende que la simulación de un sistema es la manipulación experimental por computadora de un modelo lógico-matemático. Más aún, en caso de tratarse de sistemas de tiempo real estas conductas deben ser reproducidas estrictamente, sin distorsiones en el tiempo, por lo que esta última exigencia se suma a las demás condiciones de diseño, propias del paradigma de objetos.

Sin embargo, también es necesario destacar que el mejor modelo no es el más preciso ni el más completo, sino más bien el que represente las variables esenciales del sistema con la precisión suficiente. Es así que las diversidades de modelos empleados para simular motores pueden ser, en primera instancia, agrupados como “cajas blancas”, “grises” y “negras”:

Los modelos de “cajas blancas” son usados cuando se conoce profundamente el fenómeno físico, se dispone de herramientas de análisis numérico basadas en el método de los elementos finitos –FEM- y se cuenta con un gran poder computacional. Normalmente, estos modelos tridimensionales presentan gran complejidad geométrica y están destinados a reproducir fenómenos de interacción entre medios continuos, tales como fluido-dinámicos, termodinámicos y de sólidos elásticos. Su objetivo final es predecir la performance del motor a partir de su geometría y condiciones de funcionamiento.

Hay casos en que estos modelos se convierten progresivamente en “cajas grises” o “negras”, a medida que ciertos datos o la falta de conocimiento suficiente sobre los fenómenos físicos involucrados deben ser reemplazados por información experimental. En la selección del tipo de representación más apropiada intervienen también, como factores condicionantes, las exigencias funcionales del sistema representado, como son el tiempo de respuesta o la predictibilidad de sus resultados.

Una de las líneas de investigación más importantes en el campo de la simulación de motores se viene desarrollando en el Departamento de Ingeniería Mecánica de la Universidad de Salerno, Italia. Los modelos allí

empleados se apoyan en redes neuronales multicapa de perceptrones (Arsie, 2001), que a partir de algunos parámetros de entrada permiten predecir el desempeño del motor. Se trata de modelos de "caja negra", que son entrenados mediante el procedimiento de backpropagation (Arsie, 1998) con conjuntos de datos provenientes de los resultados de ensayos. Estos modelos han demostrado sus ventajas en la representación de fenómenos no-lineales y en aquellos casos en que los datos disponibles presentan un nivel de incerteza importante. Éste y otros modelos similares permiten hacer un máximo aprovechamiento de la información obtenida de los ensayos, brindando resultados relativos a la potencia, composición de las emisiones del motor, consumo u otros parámetros y permitiendo simular estrategias de control.

En otros casos, se representa el funcionamiento de estos conjuntos a partir de la combinación de modelos más detallados de los diferentes componentes del motor. Por ejemplo, se evalúa el torque generado por cada pistón en cada momento a partir de: 1) la fase en el ciclo del motor (admisión, compresión, explosión y escape), 2) la posición del pistón, 3) la masa de aire cargada en el cilindro, 4) la cantidad de combustible inyectado y 5) el tiempo de encendido. La secuencia de condiciones en cada cilindro es determinada por un autómata finito y su contribución es incorporada al conjunto de acuerdo a su orden de encendido y la velocidad del motor (Balluchi, 2000). En estos modelos de "cajas grises", los valores representativos del comportamiento del motor en el tiempo son obtenidos a partir de la combinación de las contribuciones de sus distintos componentes.

Señales de Torque

Ya fue anticipado que el único torque que puede ser leído es el T_c , que es evaluado a través de las deformaciones δ de una celda de carga. En el sistema real, esta señal del conjunto scc es muestreada regularmente con un período que es impuesto para la rapidez del conversor análogo / digital disponible t_a y es convertida en una señal sincrónica del conjunto scd, luego:

$$fc': scc \in scd; Tc = fc'(\delta, \tau_a) \quad (91)$$

Sin embargo, en el modelo sólo se cambia la representación en el tiempo de la señal dentro del mismo conjunto scd ya que:

$$In: scc \in scd; \delta = \ln(y, \Delta t) \quad (92)$$

$$fc: scd \in scd; Tc = fc(\delta, \tau_a) \quad (93)$$

Señales de Velocidad

La medición de la velocidad del motor Ω merece un comentario especial. En efecto, se trata de la variable ω_r que ya fue convertida al conjunto scd por la integración numérica. Luego, su medición se realiza a través de un dispositivo apropiado que la convierte en un tren periódico de pulsos ρ (sdd) y para su interpretación física es vuelta a convertir en una señal perteneciente al conjunto scd identificada como Ω :

$$In: scc \in scd; \omega_r = \ln(y, \Delta t) \quad (94)$$

$$fp: scd \in sdd; \rho = fp(\omega_r) \quad (95)$$

$$f\Omega: sdd \in scd; \Omega = f\Omega(\rho) \quad (96)$$

donde fp representa al dispositivo de medición y $f\Omega$ representa al taquímetro del sistema, los que serán tratados en detalle por separado.

Eventos

La señal de seguros λ , la señal de habilitación del freno η y la de encendido del motor μ , pertenecen al conjunto sdd de señales asincrónicas y están relacionadas a eventos que producirán cambios en las condiciones de operación del sistema. Se originan y son aplicadas sin sufrir transformaciones en su representación.

Otros eventos que son reconocidos por la unidad de control primario están relacionados con ciertas condiciones de las variables y en el caso del sistema real o de las variables u del modelo y actúan sobre la misma unidad

pautas que están establecidas en el programa de ensayos y las acciones del operador. El contexto del sistema está por lo tanto conformado por todos los demás elementos ya descritos en el punto 4.4 y representados en el esquema de la Figura 24.

Vocabulario del Léxico Extendido del Lenguaje

Una vez presentado el problema en forma general, debe continuarse con el reconocimiento y definición precisa de su terminología. Se recurre para ello al LEL (Léxico Extendido del Lenguaje) según las previsiones de la metodología adoptada y para implementar su vocabulario se emplea una tabla en la que se describen las nociones e impactos de cada palabra o frase. Aquí, debe recordarse que la noción representa el significado de cada término y el impacto determina los efectos de su uso u su ocurrencia en la aplicación.

Según las recomendaciones de Leandro Antonelli (Antonelli, 1999 y 2001), este vocabulario es progresivamente ajustado y perfeccionado a medida que se progresa en el conocimiento del problema real y sus resultados son presentados en la Tabla 2.

Nº	Símbolo	Noción	Impacto
1	<u>Motor</u>	Máquina rotativa-alternativa de combustión interna.	<u>Conjunto Motor-Freno</u> <u>Variable de Proceso</u>
2	<u>Freno dinamométrico</u>	Máquina rotativa destinada a absorber la potencia entregada por un <u>motor</u> . Es una evolución del "freno de Prony".	<u>Conjunto Motor-Freno</u> <u>Variable de Proceso</u>
3	<u>Acoplamiento o Conjunto Motor-Freno</u>	Conjunto formado por un <u>Motor</u> y un <u>Freno</u> vinculados entre sí por una transmisión cardánica.	<u>Variables de proceso</u>
4	<u>Ensayo</u>	Evaluación experimental de un equipo en general y de un <u>motor</u> en particular.	<u>Conjunto motor-freno</u>
5	<u>Taquímetro</u>	Instrumento destinado a determinar la velocidad de un <u>motor</u> .	<u>Vector de estado</u>
6	<u>Sensor de Torque</u>	Instrumento destinado a determinar el torque de un <u>motor</u> o <u>freno</u> .	<u>Vector de estado</u>

7	<u>Seguros</u>	Unidad destinada a delatar si las condiciones de operación del <u>Motor</u> o <u>Freno</u> pueden poner en peligro su integridad.	<u>Control secundario</u>
8	<u>Medición</u>	Operación experimental destinada a asignar a cierta magnitud de un dominio físico una cantidad de unidades de una escala apropiada (Maiztegui, 2000).	<u>Determinación</u>
9	<u>Determinación</u>	Asignación de un valor a una variable física, ya sea directamente a través de una <u>medición</u> o en forma indirecta a partir de otras <u>mediciones</u> .	<u>Taquímetro</u> <u>Freno dinamométrico</u>
10	<u>Evento</u>	Identificador atemporal de un cambio en la condición externa de un sistema..	<u>Unidad de control</u>
11	<u>Estado</u>	Condición distintiva en que se encuentra un sistema.	<u>Variable de estado</u>
12	<u>Variable de estado</u>	El menor conjunto de variables que definen completamente el <u>estado</u> de un sistema.	<u>Vector de estado</u>
13	<u>Vector de estado</u>	Arreglo unidimensional que agrupa a todas las <u>variables de estado</u> . Determina unívocamente el <u>estado</u> de un sistema.	<u>Espacio de estados</u> <u>Interfase con operador</u>
14	<u>Espacio de estados</u>	Espacio n-dimensional cuyos ejes de coordenadas están formados por las <u>variables de estado</u> . Todo <u>estado</u> de un sistema queda representado como un punto en su <u>espacio de estados</u>	<u>Control primario</u>
15	<u>Variable de proceso (VP)</u>	Una de las <u>variables de estado</u> que se desea <u>controlar</u> , como es el caso de la velocidad del <u>motor</u> o su <u>torque</u> .	<u>Taquímetro</u> <u>Freno dinamométrico</u>
16	<u>Set point (SP)</u>	Es el valor deseado de la <u>variable de proceso</u> , es decir, el valor que debe ser alcanzado y mantenido estable por la <u>unidad de control</u> .	<u>Control primario</u>
17	<u>Error (e)</u>	Diferencia entre el <u>set point (SP)</u> y la <u>variable de proceso (PV)</u> .	<u>Control primario</u>
18	<u>Control</u>	Acción destinada a procurar que en todo momento las <u>variables de proceso</u> alcancen los valores prefijados por los <u>set points</u> , es decir, que el <u>error</u> sea nulo.	<u>Unidad de control</u>
19	<u>Unidad de Control</u>	Dispositivo destinado a cumplir acciones de <u>control</u> .	<u>Control primario</u> <u>Control secundario</u>
20	<u>Control primario</u>	Función de la <u>Unidad de Control</u> , que tiene la finalidad de operar directamente sobre el objeto controlado.	<u>Variable de control</u> <u>Señales de Control</u> <u>Lógica activa</u>

21	<u>Control secundario</u>	Función de la <u>Unidad de Control</u> , que tiene la finalidad de interpretar y ejecutar los <u>Programas de Ensayo</u> .	<u>Control primario</u> <u>Interfase con operador</u> <u>Paso de Ensayo Activo</u>
22	<u>Variable de control (VC)</u>	Variable que está asociada a la acción utilizada para <u>controlar</u> un sistema. En el caso de un <u>motor</u> se trata de la posición del acelerador.	<u>Vector de control</u>
23	<u>Vector de control</u>	Vector en el que se reúnen las <u>variables de control (VC)</u> .	<u>Motor</u> <u>Freno dinamométrico</u>
24	<u>Señal de control</u>	Señales digitales de encendido del <u>motor</u> y <u>habilitación del freno</u> .	<u>Motor</u> <u>Freno dinamométrico</u>
25	<u>Modo automático</u>	Modo de operación de un <u>motor</u> en el que se da cumplimiento a un <u>Programa de Ensayo</u> .	<u>Control secundario</u>
26	<u>Controlabilidad</u>	Posibilidad de cambiar el <u>estado</u> de un sistema entre dos puntos cualesquiera del <u>espacio de estados</u> en un intervalo de tiempo finito. Obsérvese que la condición de <u>controlabilidad</u> determina la existencia de una solución en el problema del diseño de un sistema de <u>control</u> .	<u>Control primario</u>
27	<u>Obsevabilidad</u>	Posibilidad de acceder a todas las <u>variables de estado</u> a través de <u>mediciones</u> . Si un sistema no es completamente observable, algunas de sus variables deben ser <u>determinadas</u> en forma indirecta.	<u>Variables de estado</u>
28	<u>Realimentación</u>	Aplicación de una cierta <u>lógica activa</u> . Este proceso suele denominarse de lazo cerrado.	<u>Control primario</u>
29	<u>Lógica activa</u>	Proceso a través del cual se calculan valores convenientes para la <u>variable de control (VC)</u> a partir de los valores alcanzados por la <u>variable de proceso (VP)</u> y su <u>error</u> .	<u>Realimentación</u>
30	<u>Paso de Ensayo</u>	Condiciones deseadas para la operación de un <u>motor</u> durante un cierto intervalo de tiempo.	<u>Programa de ensayo</u>
31	<u>Paso de Ensayo Activo</u>	<u>Paso de Ensayo</u> empleado en cierto instante para el <u>control del motor</u> .	<u>Control primario</u>
32	<u>Programa de Ensayo</u>	Secuencia teórica de <u>Pasos de Ensayo</u> acordes a los objetivos de un <u>ensayo</u> .	<u>Control secundario</u>
33	<u>Secuencia de ensayo</u>	Secuencia real de <u>Pasos de Ensayo</u> a ser cumplidos.	<u>Interfase con operador</u>
34	<u>Parada de emergencia</u>	Orden de detención inmediata del motor.	<u>Interfase con operador</u>
35	<u>Final del ensayo</u>	Orden de finalizar la <u>secuencia de ensayo</u> abandonando el <u>modo automático</u> y detener el motor	<u>Interfase con operador</u>

36	<u>Función de transferencia (G)</u>	Cociente entre las expresiones matemáticas de las <u>variables de proceso</u> (salida) y las <u>variables de control</u> del sistema (entrada) en función del tiempo. Suele también denominarse transmitancia.	<u>Conjunto motor-freno</u>
37	<u>Interfase con operador</u>	Pantalla de representación, teclado y otros medios de ingreso de datos.	Control secundario

Tabla N° 01 Vocabulario del LEL

Identificación de eventos

Del vocabulario del LEL se seleccionan los símbolos que se refieren a elementos externos al sistema y que impactan sobre el propio sistema o que pertenecen al sistema y su comportamiento tiene alguna connotación temporal. Luego, se trabaja sobre esta selección hasta reconocer los eventos externos y temporales que estimulan el sistema, teniendo aquí en cuenta las ya mencionadas condiciones que debe cumplir un evento y que fueron enumeradas por Page-Jones (Page-Jones, 1992).

Entradas al LEL	Evento	Contexto / Sistema	Sistema	Contexto
		Origen/intermediarios del evento	Receptor de mensaje y elementos intermedios	Destinos de respuesta
5 / 13	1	<u>Taquímetro</u>	<u>Vector de estado</u>	<u>Interfase con operador</u>
6 / 13	2	<u>Sensor de Torque</u>	<u>Vector de estado</u>	<u>Interfase con operador</u>
34/37/21/ 20/24	3	<u>Parada de emergencia</u> <u>Interfase con operador</u>	<u>Control secundario</u> <u>Control primario</u> <u>Señales de control</u>	<u>Motor</u> <u>Freno dinamométrico</u> <u>Interfase con operador</u>
35/37/21/ 20/24	4	<u>Final del ensayo</u> <u>Interfase con operador</u>	<u>Control secundario</u> <u>Control primario</u> <u>Señales de control</u>	<u>Motor</u> <u>Freno dinamométrico</u> <u>Interfase con operador</u>
33/37/21/ 31	5	<u>Alterar secuencia de ensayo</u> <u>Interfase con operador</u>	<u>Control secundario</u> <u>Paso de ensayo activo</u>	<u>Interfase con operador</u>
7/21/20/ 24	6	<u>Seguros</u>	<u>Control secundario</u> <u>Control primario</u> <u>Señales de control</u>	<u>Motor</u> <u>Freno dinamométrico</u> <u>Interfase con operador</u>
32/21	7	<u>Programa de Ensayo</u>	<u>Control secundario</u> <u>Paso de ensayo activo</u>	<u>Interfase con operador</u>
28/20/22/ 23	8	<u>Realimentación</u>	<u>Control primario</u> <u>Variable de control</u> <u>Vector de control</u>	<u>Motor</u> <u>Freno dinamométrico</u>

13/14/20/ 29	9	<u>Vector de estado</u>	<u>Espacio de estados</u> <u>Control primario</u> <u>Lógica activa</u> <u>Realimentación</u>	-----
-----------------	---	-------------------------	---	-------

Tabla N° 02 Entradas al LEL y eventos

En el caso de tratarse de elementos internos al sistema, estos encadenamientos o secuencias que vinculan las entradas al LEL contribuyen a reconocer los flujos de control, luego van a facilitar la definición de las actividades esenciales y serán muy importantes para permitir la trazabilidad en el modelo.

Diagrama de contexto

Considerando exclusivamente la interacción mutua entre los elementos externos y el propio sistema, se representa el diagrama de contexto, que es presentado en la Figura 41:

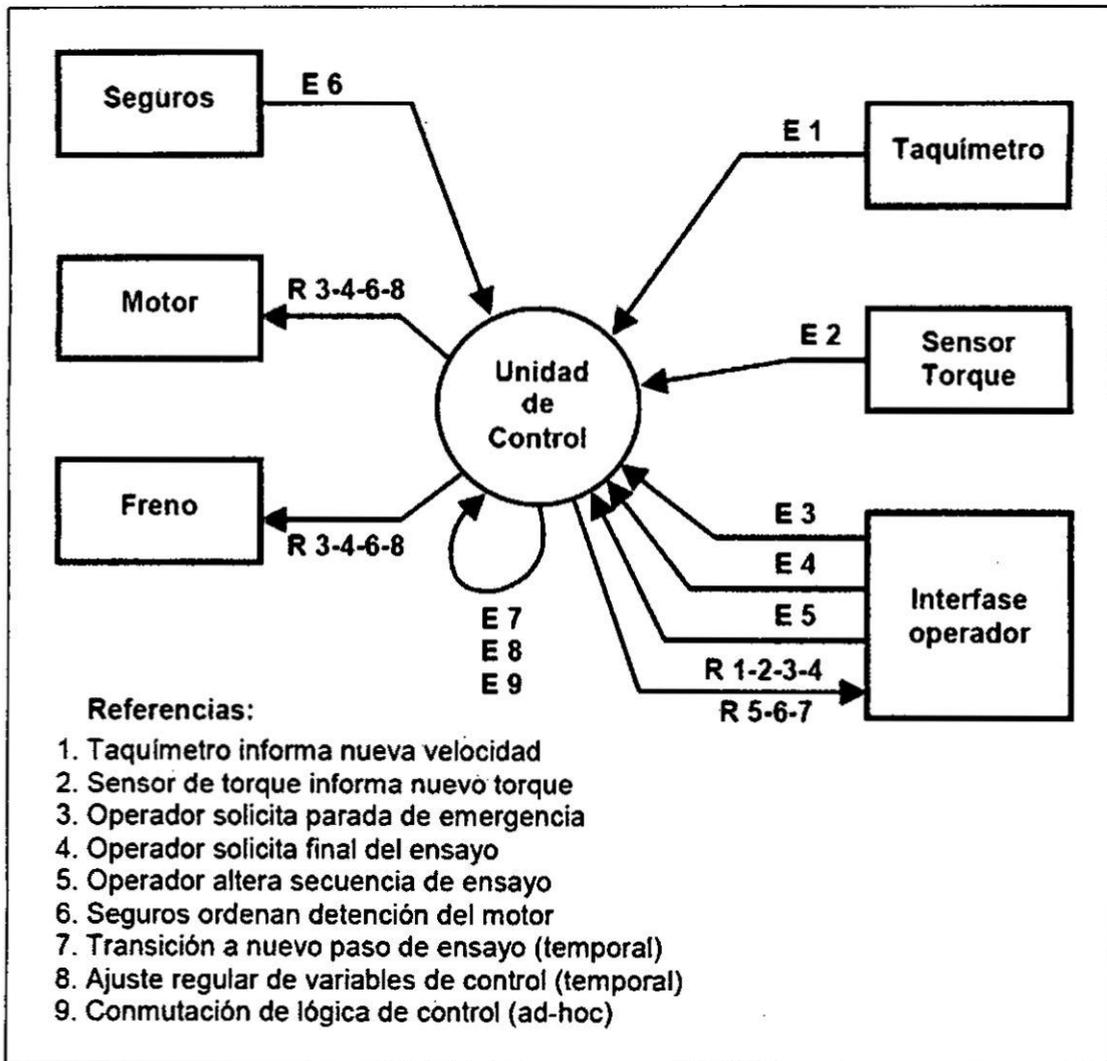


Tabla de eventos

Los eventos identificados en primera instancia en la Tabla 3 y representados en el diagrama de contexto, son presentados en la nueva Tabla 4 en la que se identifican sus actividades esenciales.

Nº	EVENTO				ACTIVIDAD ESENCIAL	RESPUESTA	DESTINO
	Denominación	Tipo	Fuente	Estímulo			
1	Taquímetro informa nueva velocidad	ext.	Taquímetro	Mensaje	Recibir y registrar nueva velocidad del motor	Mensaje con valor de velocidad	Interfase operador

2	<u>Sensor de torque</u> informa nuevo torque	ext.	<u>Sensor de Torque</u>	Mensaje	Recibir y registrar nuevo torque en el <u>freno</u>	Mensaje con valor de Torque	<u>Interfase operador</u>
3	Operador solicita <u>parada de emergencia</u>	ext.	<u>Interfase operador</u>	Mensaje	Detener el <u>motor</u> en forma inmediata	Mensaje de detención del <u>motor</u>	<u>Motor y Freno</u> <u>Interfase operador</u>
4	Operador solicita <u>final del ensayo</u>	ext.	<u>Interfase operador</u>	Mensaje	Terminar <u>secuencia de ensayo</u> y detener el motor	Mensajes de <u>fin de ensayo</u> y detención del <u>motor</u>	<u>Motor y Freno</u> <u>Interfase operador</u>
5	Operador altera <u>secuencia de ensayo</u>	ext.	<u>Interfase operador</u>	Mensaje	Alterar la <u>secuencia de ensayo</u>	Mensaje con nuevo <u>paso de ensayo activo</u>	<u>Interfase operador</u>
6	<u>Seguros</u> ordenan detención del <u>motor</u>	ext.	<u>Seguros</u>	Mensaje	Detener el <u>motor</u> en forma inmediata	Mensaje de detención del <u>motor</u>	<u>Motor y Freno</u> <u>Interfase operador</u>
7	Transición a nuevo <u>paso de ensayo</u>	Tpo	<u>Unidad de Control</u>	Intervalo de Tiempo	Cumplir las previsiones del <u>programa de ensayo</u>	Mensaje con nuevo <u>paso de ensayo activo</u>	<u>Interfase operador</u>
8	Ajuste regular de <u>variables de control</u>	Tpo	<u>Unidad de Control</u>	Intervalo de Tiempo	Asegurar una condición estable de operación	Señales de control de <u>motor y freno</u>	<u>Motor y Freno</u>
9	Conmutación de <u>lógica activa de control</u>	cond. esp.	<u>Unidad de Control</u>	Condición de <u>vector de estado</u>	Definir la <u>lógica activa</u> de control más apropiada	---	---

Tabla N° 03 Nómima de eventos

Fichas de eventos

N° Evento	1
Denominación	<u>Taquímetro</u> informa nueva velocidad
Tipo	Externo
Descripción	El <u>taquímetro</u> ha completado la <u>determinación</u> del último valor de velocidad del <u>motor</u> y la transmite a la <u>unidad de control</u> , que debe actualizar su <u>vector de estado</u> .
Fuente	<u>Taquímetro</u>

Estímulo	Mensaje conteniendo el valor de la velocidad del <u>motor</u> expresada en rpm.
Actividad esencial	<p>Recibir y registrar nueva velocidad del <u>motor</u>:</p> <ul style="list-style-type: none"> • Completa recepción de mensaje • Controla consistencia del mensaje recibido • Decodifica valor de velocidad del <u>motor</u> • Actualiza <u>vector de estado del sistema</u> • Transmite valor de velocidad del motor a la <u>interfase con operador</u>
Respuesta	Mensaje con valor de la velocidad del <u>motor</u>
Destino	<u>Interfase con operador</u>
Efecto	Actualiza condición de operación del <u>motor</u>
DFD	<pre> graph LR Taquimetro[Taquimetro] -- E 1 --> Registrar((Registrar nueva velocidad del motor)) Registrar -- R 1 --> Interfase[Interfase operador] Registrar --> Estado[Vector de estado] </pre>

Cuadro N° 01 Ficha de evento

Fichas de eventos (cont.)

N°Evento:	2
Denominación	<u>Sensor de torque</u> informa nuevo torque
Tipo	Externo
Descripción	El <u>sensor de torque</u> ha completado la <u>determinación</u> del último valor de torque del <u>freno</u> y lo transmite a la <u>unidad de control</u> , que debe actualizar su <u>vector de estado</u> .
Fuente	<u>Sensor de torque</u>
Estímulo	Mensaje conteniendo el valor del torque del <u>freno</u> expresada en Kg.m.

Actividad esencial	Terminar <u>secuencia de ensayo</u> y detener el <u>motor</u> : <ul style="list-style-type: none"> • Completa recepción del mensaje • Controla consistencia del mensaje recibido • Interpreta mensaje • Finaliza operación en <u>Modo Automático</u> • Ordena detención del <u>motor</u> (señal de control) • <u>Deshabilita freno dinamométrico</u> (señal de control) • Actualiza registro de operación • Actualiza interfase con operador
Respuesta	Mensajes de parada del <u>motor</u> , <u>deshabilita freno</u> y condición de operación
Destino	Motor, Freno Dinamométrico e Interfase con operador
Efecto	Fin del <u>Ciclo de ensayo</u> y detención del <u>motor</u>
DFD	

Cuadro N° 04 Ficha de evento

Fichas de eventos (cont.)

N° Evento	5
Denominación	Operador altera <u>secuencia de ensayo</u>
Tipo	Externo
Descripción	El operador ordena alterar la secuencia prevista en el <u>programa de ensayo</u> , con avance a próximo <u>paso</u> o retroceso a paso anterior.
Fuente	<u>Interfase con operador</u>
Estímulo	Mensaje con el comando que corresponda
Actividad esencial	<p>Alterar la <u>secuencia de ensayo</u> prevista en el programa:</p> <ul style="list-style-type: none"> • Completa recepción de mensaje • Controla consistencia del mensaje recibido • Interpreta mensaje • Almacena registro de datos en paso actual • Actualiza registro de <u>paso de ensayo activo</u> • Actualiza registro de operación • Actualiza <u>interfase con operador</u>
Respuesta	Mensaje informativo
Destino	<u>Interfase con operador</u>
Efecto	Cambiar la condición de operación del <u>conjunto motor-freno</u>
DFD	

Cuadro N° 05 Ficha de evento

Fichas de eventos (cont.)

N° Evento	6
Denominación	<u>Seguros ordenan detención del motor</u>
Tipo	Externo
Descripción	Debe detenerse el <u>motor</u> en forma inmediata. Se presentó una condición de <u>anomalía</u> en el <u>motor</u> , <u>freno dinamométrico</u> o instalaciones auxiliares
Fuente	<u>Seguros</u>
Estímulo	<u>Mensaje</u> indicando <u>anomalía</u> en <u>seguros</u> .
Actividad esencial	<p>Detener el <u>motor</u> en forma inmediata:</p> <ul style="list-style-type: none"> • Completa recepción de mensaje • Controla consistencia del mensaje recibido • Decodifica e identifica <u>seguro</u> • Ordena detención del <u>motor</u> (señal de control) • Deshabilita <u>freno dinamométrico</u> (señal de control) • Actualiza registro de operación • Actualiza <u>interfase con operador</u>
Respuesta	Mensajes de parada del <u>motor</u> , deshabilita <u>freno dinamométrico</u> y condición de operación
Destinos	<u>Motor</u> , <u>freno dinamométrico</u> e <u>Interfase con el operador</u>
Efecto	Detener el <u>motor</u> e identificar la causa de la falla
DFD	<pre> graph LR Seguros[Seguros] -- E 6 --> Detener((Detener el motor en forma inmediata)) Detener -- R 6 --> Interfase[Interfase operador] Detener -- R 6 --> Motor[Motor] Detener -- R 6 --> Freno[Freno] Detener --> Registro[registro de operación] </pre>

Cuadro N° 06 Ficha de evento

Fichas de eventos (cont.)

N° Evento	7
------------------	---

Denominación	Transición a nuevo <u>paso de ensayo</u>
Tipo	Temporal
Descripción	Activa próximo <u>paso de ensayo</u> según previsión de <u>programa de ensayo</u>
Fuente	<u>Sistema de Control</u>
Estímulo	
Actividad esencial	<p>Cumplir las previsiones del <u>programa de ensayo</u>:</p> <ul style="list-style-type: none"> • Lee <u>programa de ensayo</u> • Lee <u>vector de estado</u> • Almacena registro de datos en paso actual • Determina próximo paso • Actualiza registro de <u>paso de ensayo activo</u> • Actualiza registro de operación • Actualiza <u>interfase con operador</u>
Respuesta	Mensaje de nueva condición de operación
Destino	<u>Interfase con operador</u>
Efecto	Cambiar <u>la condición</u> de operación del conjunto motor-freno
DFD	<pre> graph LR PE[programa de ensayos] --> CPE((Cumplir programa de ensayo)) VE[vector de estado] --> CPE CPE --> RD[registro de datos] CPE --> PA[paso de ensayo activo] CPE --> RO[registro de operación] CPE -- R7 --> IO[Interfase operador] </pre>

Cuadro N° 07 Ficha de evento

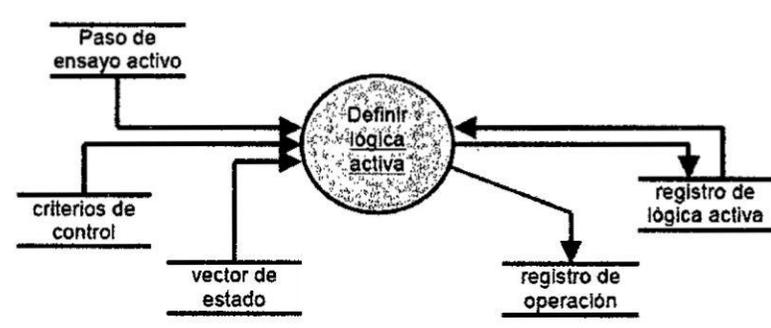
Fichas de eventos (cont.)

N° Evento	8
Denominación	Ajuste regular de <u>variables de control</u>
Tipo	Temporal
Descripción	Ajuste regular de las <u>variables de control</u> para asegurar una operación estable acorde a las exigencias del programa de ensayo

Fuente	<u>Sistema de Control</u>
Estímulo	
Actividad esencial	<p>Asegurar una condición estable de operación:</p> <ul style="list-style-type: none"> • Lee velocidad y torque de <u>vector de estado</u> • Lee <u>programa de ensayo</u> • Lee <u>paso de ensayo activo</u> • Calcula error de <u>variables controladas</u> • Lee criterios de control • Lee <u>lógica activa</u> • Calcula ajuste necesario en <u>variables de control</u> • Determina nuevos valores de las <u>variables de control</u> del <u>Motor y freno dinamométrico</u> • Actualiza registro de operación
Respuesta	Mensajes de control
Destino	<u>Motor y freno dinamométrico</u>
Efecto	Hacer mínima las desviaciones respecto de los set-points
DFD	<pre> graph LR PE[programa de ensayos] --> P((Asegurar condición estable de operación)) VE[vector de estado] --> P CC[criterios de control] --> P PEA[paso de ensayo activo] --> P RL[registro de lógica activa] --> P P --> RO[registro de operación] P --> M[Motor] P --> F[Freno] M --- R7_1[R 7] F --- R7_2[R 7] </pre>

Fichas de eventos (cont.)

N° Evento	9
Denominación	Conmutación de <u>lógica activa</u> de control
Tipo	Condición especial
Descripción	Conmutación de la lógica de control para asegurar la operación dentro de los límites establecidos
Fuente	<u>Sistema de Control</u>
Estímulo	

Actividad	Definir la <u>lógica activa</u> de control mas apropiada: <ul style="list-style-type: none"> • Lee <u>vector de estado</u> • Lee <u>paso de ensayo activo</u> • Lee el registro de <u>lógica activa</u> • Lee criterios de control • Identifica condiciones de conmutación • Selecciona la <u>lógica activa</u> más apropiada • Actualiza registro de la <u>lógica activa</u> a ser usada • Actualiza registro de operación
Respuesta	
Destino	
Efecto	Seleccionar la <u>lógica de control</u> más apropiada para operar el <u>conjunto motor-freno</u> dentro de los límites establecidos
DFD	

Cuadro N° 09 Ficha de evento

Modelo de eventos del sistema (diagrama 0)

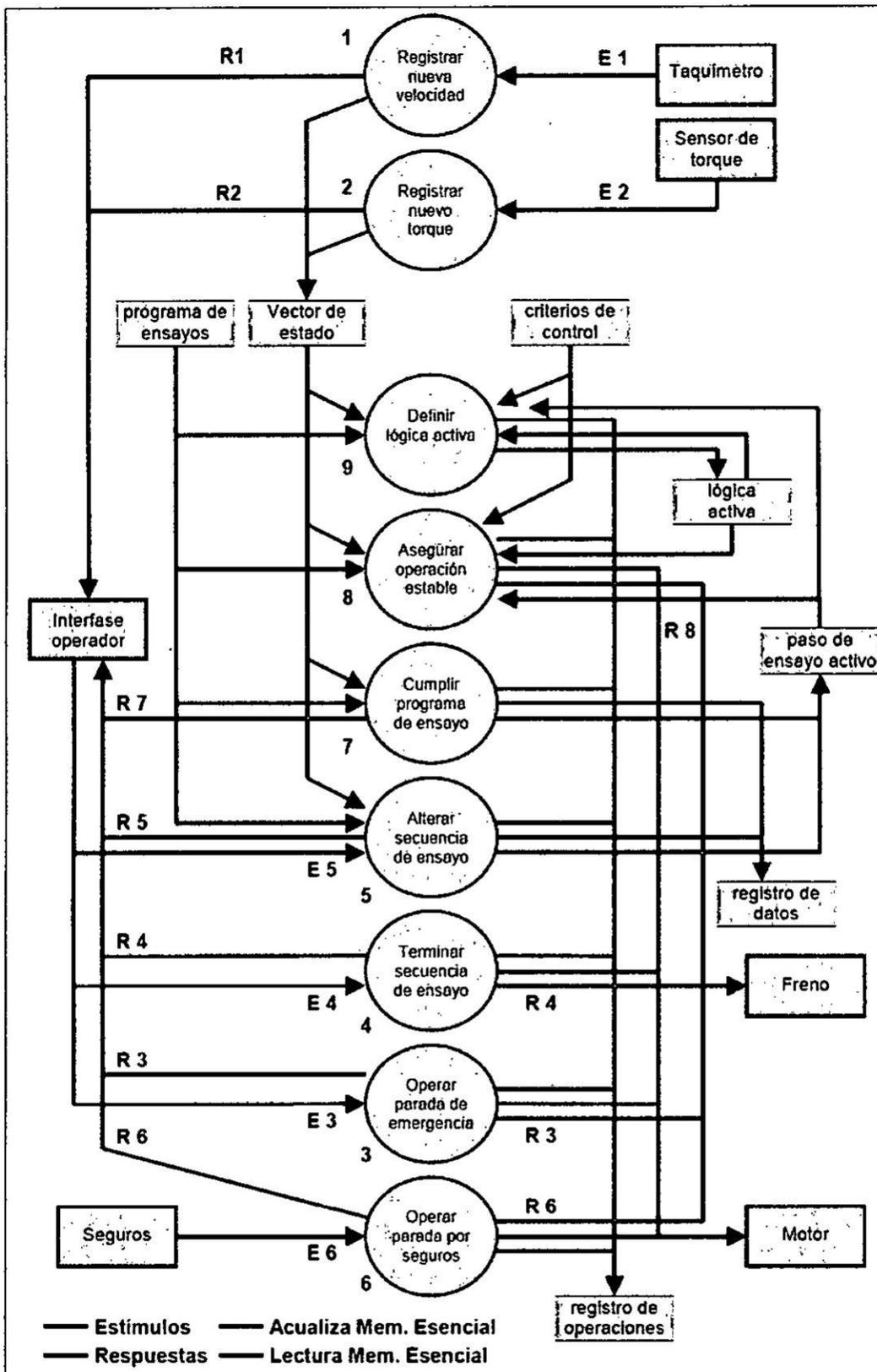


Figura N° 15 Modelo de eventos (diagrama "0")

Matriz de eventos - datos

A partir de los eventos del sistema, puede armarse una matriz que resume la forma en que cada uno opera sobre los datos, representada en la Tabla 5 tal como lo sugiere David Ruble (Ruble, 1997):

Evento	Programa de ensayo	Vector de Estado	Criterios de Control	Lógica activa	Paso de ensayo activo	Registro de datos	Registro de operación
1		U					
2		U					
3							C
4	R						C
5	R	R			U	C	C
6							C
7	R	R			U	C	C
8	R	R	R	R	R		C
9	R	R	R	R,U			C

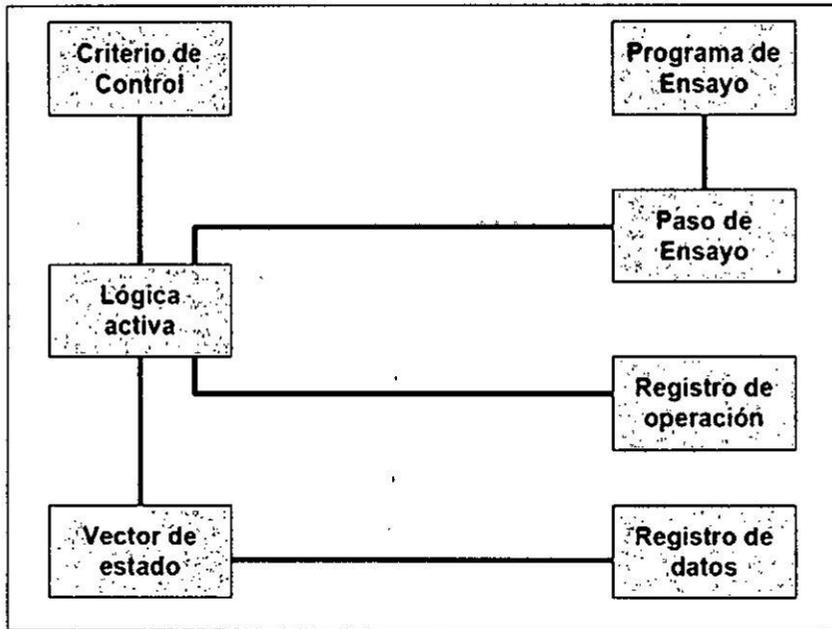
Tabla N° 04 Matriz de eventos

Referencias:

- C: Carga de nuevo registro (Create)
- R: Lectura de registro (Read)
- U: Actualización (Update)
- D: Borrado (Delete)

Diagrama de Entidad-Relación (DER)

Una primera versión de un diagrama de Entidad-Relación es obtenida a partir del Diagrama 0 y de las descripciones contenidas en las Fichas de Eventos. La consistencia de este diagrama puede confirmarse en la Matriz de Eventos-Datos, donde cada relación debe estar respaldada por algún evento que contenga ambas entidades.



Estructuras de almacenes de datos

Nómbre almacén.	Estructura del registro
Paso de ensayo	Tiempo de permanencia
	Velocidad
	Torque
	Identificador de próximo paso
Vector de estado	Velocidad
	Torque en el freno
	Velocidad extrapolada a tiempo de referencia
	Torque de freno extrapolado a tiempo de referencia
	Torque de motor calculado en tiempo de referencia
	Derivada de velocidad
	Derivada de torque del freno
Derivada de torque del motor	
Criterio de control	Antecedente (regla de producción)
	Consecuente (regla de producción)
Lógica activa	Identificador de opción de control
	Identificador de realimentación PID
Registro de datos	Tiempo de ensayo
	Identificador de paso de ensayo activo
	Identificador de lógica activa
	Velocidad
	Torque en freno
	Torque en motor
Registro de operación	Tiempo de ensayo
	Código de operación
	Argumento de operación

Tabla N° 05 Estructuras de almacenes de datos

Actividades esenciales y objetos

La tarea de establecer vínculos entre las actividades esenciales y los objetos puede facilitarse enormemente con un ordenamiento como el de la Tabla 8,

donde se enumeran las actividades esenciales, se identifican las responsabilidades asociadas y a cada una de ellas se le asigna el objeto que lo contendrá. Este proceso se apoya en los siguientes dos postulados básicos:

- a) La funcionalidad global de un sistema queda completamente definida por su Modelo Esencial.
- b) El conjunto de las responsabilidades de todas las clases representa lo que el sistema en su conjunto es capaz de realizar.

Al plantearse esta relación, se están equilibrando los modelos Esenciales y de Objetos, ya que cada una de las N capacidades del sistema debe estar presente en ambos modelos, es decir que:

$$fk(E) = f'k(O); k = 1, \dots, N \quad (102)$$

Actividades esenciales		Responsabilidades	Objetos	
1	Recibir y registrar nueva velocidad del motor	Recibir un mensaje del taquímetro	Operador De Mensajes	2
		Controlar consistencia y decodificarlo	Decodificador De Mensajes	3
		Operar mensajes del taquímetro	Interfase Taquímetro	1
		Actualizar vector de estado	Vector De Estado	4
		Informar nueva velocidad a la Interfase con el operador	Interfase Con Operador	5
2	Recibir y registrar nuevo torque en el freno	Recibir un mensaje del Sensor de Torque	Operador De Mensajes	2
		Controlar consistencia y decodificarlo	Decodificador De Mensajes	3
		Operar mensajes del sensor de torque	Interfase SensorTorque	6
		Actualizar vector de estado	Vector De Estado	4
		Informar nuevo torque a la Interfase con el operador	Interfase Con Operador	5
3	Detener el motor en forma inmediata por parada de	Recibir un mensaje de la Interfase con el Operador	Operador De Mensajes	2
		Operar orden de detención por parada de emergencia	Interfase Con Operador	5
		Controlar consistencia y decodificarlo	Decodificador De Mensajes	3
		Ordenar detención del motor	Controla Motor Freno	7

	emergencia	Informar parada de emergencia a la Interfase con el operador	Interfase Con Operador	5
		Actualiza registro de operaciones	Registro De Operaciones	8
4	Terminar secuencia de ensayo y detener el motor	Recibir un mensaje de la Interfase con el Operador	Operador De Mensajes	2
		Operar orden de finalizar el ensayo y detener el motor	Interfase Con Operador	5
		Controlar consistencia y decodificarlo	Decodificador De Mensajes	3
		Finalizar la operación en Modo Automático	Intérprete De Ensayos	9
		Ordenar detención del motor	Controla Motor Freno	7
		Informar fin de ensayo a la Interfase con el operador	InterfaseConOperador	5
		Actualiza registro de operaciones	Registro De Operaciones	8
5	Alterar secuencia de ensayo según orden del operador	Recibir un mensaje de la Interfase con el Operador	Operador De Mensajes	2
		Operar orden de cambiar la secuencia de ensayo	Interfase Con Operador	5
		Controlar consistencia y decodificar el mensaje	Decodificador De Mensajes	3
		Almacenar condición del ensayo	Registro De Datos	10
		Leer el Programa de Ensayos	Programa De Ensayos	12
		Leer el Vector de Estado	Vector De Estado	4
		Determinar próximo paso de ensayo	Intérprete De Ensayos	9
		Actualizar registro de Paso de Ensayo Activo	Registro De Paso Activo	11
		Actualiza registro de operaciones	Registro De Operaciones	8
		Informar de nuevo paso activo a interfase con operador	Interfase Con Operador	5
6	Detener el motor en forma inmediata por condición de los seguros	Operar orden de seguros de detener el motor	Operador De Mensajes	2
		Recibir un mensaje de los seguros	Interfase Seguros	13
		Controlar consistencia y decodificar el mensaje	Decodificador De Mensajes	3
		Ordenar detención del motor	Controla Motor Freno	7
		Informar detención del motor a la Interfase con el operador	Interfase Con Operador	5
		Actualiza registro de operaciones	Registro De Operaciones	8

7	Transitar a un nuevo paso de ensayo según previsión del programa	Controlar el tiempo de ensayo	Control Secundario	14
		Leer el Programa de Ensayos	Programa De Ensayos	12
		Leer el Vector de Estado	Vector De Estado	4
		Almacenar condición del ensayo	Registro De Datos	10
		Determinar próximo paso de ensayo	Interprete De Ensayos	9
		Actualizar registro de Paso de Ensayo Activo	Registro De Paso Activo	11
		Actualiza registro de operaciones	RegistroDeOperaciones	8
		Informar nuevo paso activo a la Interfase con el operador	Interfase Con Operador	5
8	Asegurar una condición estable de operación en el conjunto motor-freno	Determinar los ajustes necesarios en las variables de control	Control Primario	15
		Lee Vector de Estado	Vector De Estado	4
		Leer el Programa de Ensayos	Programa De Ensayos	12
		Lee Registro de Paso de Ensayo Activo	Registro De Paso Activo	11
		Lee Criterios de Control	Tabla Criterios De Control	16
		Lee registro de Lógica Activa	Registro de Lógica Activa	17
		Ajusta variables de control	Ajusta Variables Control PID	18
		Almacenar condición del ensayo	Registro De Datos	10
		Cambiar condiciones del Motor y Freno	Opera Motor Freno	7
		Actualiza registro de operaciones	RegistroDeOperaciones	8
9	Seleccionar la Lógica de Control más conveniente	Seleccionar la Lógica de Control	ControlPrimario	15
		Lee Registro de Paso de Ensayo Activo	Registro De Paso Activo	11
		Lee Vector de Estado	Vector De Estado	4
		Lee Criterios de Control	Tabla Criterios De Control	16
		Determina Lógica Activa	Selecciona Lógica Activa	19
		Almacena nueva lógica de control	Registro De Logica Activa	17
		Actualiza registro de operaciones	Registro De Operaciones	8

Tabla N° 06 Relación de actividades y objetos

Clases, responsabilidades y colaboraciones

El ordenamiento anterior ha permitido identificar los objetos directamente relacionados con las actividades esenciales. Luego, para resumir las

responsabilidades de cada uno y para establecer las relaciones entre ellos se recurre a las fichas CRC.

Este proceso que ha permitido la identificación de las principales clases será así completado siguiendo muchas de las recomendaciones habituales para el manejo de las fichas CRC, de manera de facilitar la asignación de responsabilidades, definir la jerarquía entre clases e identificar clases auxiliares o de servicios. Para ello se ha tomado como referencia el trabajo de David Bellin y Susan Suchman Simonie (Bellin, 1997).

Para la identificación de las clases, Bellin y Suchman sugieren en su trabajo dos técnicas grupales: "brainstorming" y "role playing". Si bien la eficacia de estas técnicas puede ser discutible para los casos en que se debe partir desde cero, es indudable que representan una alternativa válida para perfeccionar modelos que están muy avanzado y donde deben ajustarse sus detalles e interrelaciones.

Se presentan a continuación las fichas que definen las clases, sus responsabilidades y colaboradores (CRC) que resultan de este proceso:

Clase: 1. InterfaseTaquímetro	
Superclases:	
Subclases:	
Descripción: Al recibirse un mensaje del taquímetro, es necesario leerlo, comprobar su consistencia, decodificarlo, actualizar el vector de estado e informar a la interfase con el operador sobre la nueva velocidad.	
Responsabilidades:	Colaboradores:
- Operar Mensajes de taquímetros	- Vector De Estado (4) - Interfase Con Operador (5)

Clase: 2 OperadorDeMensajes	
Superclases:	
Subclases:	

<p>Descripción: El sistema recibe mensajes que contienen información proveniente del taquímetro, sensor de torque, seguro e interfase con el operador. Estos mensajes deben ser leídos para cada caso.</p>	
<p>Responsabilidades:</p> <ul style="list-style-type: none"> - Leer mensaje de taquímetro - Leer mensaje de sensor de torque - Leer mensaje de seguros - Leer mensaje de interfase c/operador - Transmitir mensajes a interfase c/op. - Transmitir mensajes a motor/freno 	<p>Colaboradores:</p> <ul style="list-style-type: none"> - AdministradorPuertaDeEntrada (20) - InterfaseTaquímetro (1) - Interfase Con Operador (5) - Interfas SensorTorque (6) - Interfase Seguros (13) - Decodificador de Mensajes (3)

<p>Clase: 3 DecodificadorDeMensajes</p>	
<p>Superclases:</p>	
<p>Subclases:</p>	
<p>Descripción: Según su origen, los mensajes contienen números enteros, reales, comandos o textos. Estos contenidos deben ser reconocidos y validados.</p>	
<p>Responsabilidades:</p> <ul style="list-style-type: none"> - Decodificar mensaje de taquímetro - Decodificar mensaje de torque - Decodificar mensaje de seguros - Decodificar mensaje de interfase con operador 	<p>Colaboradores:</p>

<p>Clase: 4 VectorDeEstado</p>	
<p>Superclases:</p>	
<p>Subclases:</p>	
<p>Descripción: Actualizar y lee datos de variables del vector de estado. Debe también extrapolar nuevos valores y derivarlos.</p>	
<p>Responsabilidades:</p> <ul style="list-style-type: none"> - Actualiza velocidad en vector de estado - Actualiza torque en vector de estado - Lee velocidad en vector de estado - Lee torque en vector de estado - Extrapola velocidad - Extrapola torque - Deriva velocidad - Deriva torque 	<p>Colaboradores:</p> <ul style="list-style-type: none"> - Extrapolador Derivador (21)

<p>Clase: 5 InterfaseConOperador</p>	
<p>Superclases:</p>	

Subclases:	
Descripción: Se reciben y transmiten mensajes a la Interfase con el Operador. Estos mensajes deben ser preparados o interpretados según su origen y destino.	
Responsabilidades: <ul style="list-style-type: none"> - Informar nueva velocidad - Informar nuevo torque - Informar parada de emergencia - Informar fin de ensayo - Informar cambio en secuencia ensayo - Informar detención del motor - Informar nuevo paso activo - Recibir mensajes del operador - Operar parada de emergencia - Operar fin del ensayo - Operar cambio en secuencia ensayo 	Colaboradores: <ul style="list-style-type: none"> - Controla Motor Freno (7) - Interprete De Ensayo (9)

Clase: 6 InterfaseSensorTorque	
Superclases:	
Subclases:	
Descripción: Al recibirse un mensaje del sensor de torque, es necesario leerlo, comprobar su consistencia, decodificarlo, actualizar el vector de estado e informar a la interfase con el operador sobre el nuevo torque.	
Responsabilidades: <ul style="list-style-type: none"> - Operar Mensajes de sensor de torque 	Colaboradores: <ul style="list-style-type: none"> - Vector De Estado (4) - Interfase Con Operador (5)

Clase: 7 ControlaMotorFreno	
Superclases:	
Subclases:	
Descripción: Operar sobre el motor y freno dinamométrico a través de señales digitales de habilitación o de las variables de control.	
Responsabilidades: <ul style="list-style-type: none"> - Habilitar el motor - Habilitar el freno dinamométrico - Deshabilitar el motor - Deshabilitar el freno dinamométrico - Definir posición del acelerador - Definir capacidad de frenado 	Colaboradores: <ul style="list-style-type: none"> - Operador De Mensajes (2) - Registro De Operaciones (8)

Clase: 8 RegistroDeOperaciones	
Superclases:	

Subclases:	
Descripción: Registrar la actividad cumplida por el sistema en un medio de almacenamiento permanente.	
Responsabilidades:	Colaboradores:
<ul style="list-style-type: none"> - Registrar paradas de emergencia - Registrar fin de ensayos - Registrar cambios en secuencias de ensayo - Registrar paradas por seguros - Registrar transiciones a nuevo paso de ensayo - Registrar regulación de las variables de control - Registrar conmutación de lógica de control 	

Clase: 9 IntérpreteDeEnsayos	
Superclases:	
Subclases:	
Descripción: Reconocer los comandos del programa de ensayo.	
Responsabilidades:	Colaboradores:
<ul style="list-style-type: none"> - Habilitar el modo automático de ensayo - Volver a modo manual de ensayo - Determinar próximo paso de ensayo - Determinar paso anterior 	<ul style="list-style-type: none"> - Vector De Estado (4) - Interfase Con Operador (9) - Registro De Datos (10) - Registro De Paso Activo (11) - Programa De Ensayos (12)

Clase: 10 RegistroDeDatos	
Superclases:	
Subclases:	
Descripción: Almacenar registros históricos del vector de estado en un medio permanente.	
Responsabilidades:	Colaboradores:
<ul style="list-style-type: none"> - Almacenar registros de datos de ensayo 	

Clase: 11 RegistroDePasoActivo	
Superclases:	
Subclases:	
Descripción: Registrar y leer el paso de ensayo que está activo	
Responsabilidades: <ul style="list-style-type: none"> - Registrar paso de ensayo activo - Leer paso de ensayo activo 	Colaboradores:

Clase: 12 ProgramaDeEnsayos	
Superclases:	
Subclases:	
Descripción: Leer registros de pasos de ensayo	
Responsabilidades: <ul style="list-style-type: none"> - Leer paso de ensayos 	Colaboradores:

Clase: 13 InterfaseSeguros	
Superclases:	
Subclases:	
Descripción: Al recibirse un mensaje de la interfase de seguros, es necesario leerlo, comprobar su consistencia, decodificarlo y actuar en consecuencia.	
Responsabilidades: <ul style="list-style-type: none"> - Operar Mensajes de seguros 	Colaboradores: <ul style="list-style-type: none"> - ControlaMotorFreno (7) - InterfaseConOperador (5)

Clase: 14 ControlSecundario	
Superclases:	

Subclases:	
Descripción: Verificar regularmente la recepción de mensajes y el cumplimiento del programa de ensayos.	
Responsabilidades:	Colaboradores:
<ul style="list-style-type: none"> - Interpretar el programa de ensayos - Verificar la presencia de mensajes 	<ul style="list-style-type: none"> - IntérpreteDeEnsayos(9) - OperadorDeMensajes (2)

Clase: 15 ControlPrimario	
Superclases:	
Subclases:	
Descripción: Seleccionar la lógica de control más apropiada y determinar los ajustes necesarios en las variables de control.	
Responsabilidades:	Colaboradores:
<ul style="list-style-type: none"> - Seleccionar la lógica de control - Ajustar las variables de control 	<ul style="list-style-type: none"> - AjustaVariablesControlPID(18) - SeleccionaLogicaActiva(19)

Clase: 16 TablaCriteriosDeControl	
Superclases:	
Subclases:	
Descripción: Acceder a la tabla de criterios de control, leerlos y decodificarlos. Los criterios de control están expresados como reglas de producción.	
Responsabilidades:	Colaboradores:
<ul style="list-style-type: none"> - Leer criterios de control 	

Clase: 17 RegistroDeLogicaActiva	
Superclases:	
Subclases:	
Descripción: Registrar la lógica se se considera más conveniente en cada caso.	
Responsabilidades:	Colaboradores:
<ul style="list-style-type: none"> - Leer el registro de lógica activa - Actualizar el registro de lógica activa 	

Clase: 18 AjustaVariableDeControlPID	
Superclases:	
Subclases:	
Descripción: Determinar los valores de los ajustes necesarios en las variables de control a partir de los errores en las variables controladas y del criterio de control.	
Responsabilidades:	Colaboradores:
<ul style="list-style-type: none"> - Determinar los ajustes en las variables de control 	<ul style="list-style-type: none"> - Vector De Estado (4) - Registro De Paso Activo (11) - Programa De Ensayos (12) - Tabla Criterios De Control (16) - Registro de Lógica Activa (17) - ControlaMotorFreno (7)

Clase: 19 SeleccionaLogicaActiva	
Superclases:	
Subclases:	
Descripción: Seleccionar la lógica más conveniente, según el vector de estado, los criterios de control y el programa de ensayos.	
Responsabilidades	Colaboradores:
<ul style="list-style-type: none"> - Determina la lógica más conveniente 	<ul style="list-style-type: none"> - Vector De Estado (4) - Registro De Operaciones (8) - Registro De Paso Activo (11) - Programa De Ensayos (12) - Tabla Criterios De Control (16) - Registro de Lógica Activa (17)

Clase: 20 AdministradorPuertaDeEntrada	
Superclases:	
Subclases:	
Descripción: Recibir y transmitir mensajes a través de la puerta de comunicaciones	
Responsabilidades: - Opera puerta de comunicaciones	Colaboradores:

Clase: 21 ExtrapoladorDerivador	
Superclases:	
Subclases:	
Descripción: Extrapolar y derivar valores sobre una secuencia de datos conocidos.	
Responsabilidades: - Extrapolar una cierta función - Derivarla en el punto extrapolado	Colaboradores:

Diagrama de clases

Tras lo señalado, y una vez que con la ayuda de las fichas CRC se ha revisado y eventualmente ajustado la distribución de las responsabilidades del sistema entre sus clases, debe proseguirse con la representación del diagrama de clases. Al igual que lo que ocurre con el Diagrama de Entidad Relación, el diagrama de clases es una representación estática de la estructura de un sistema. Por ello es conveniente revisar las similitudes y diferencias entre ambos diagramas.

Debe recordarse que el DER fue aquí desarrollado en el marco del análisis esencial, mientras que el diagrama de clases responde al paradigma de objetos, lo que establece una base para realizar esta comparación.

En ambos diagramas, deberían estar representados los mismos depósitos de datos y sus atributos, sólo que en el diagrama de clases se incorporan todos los objetos que están asociados con los métodos que pueden transformar esos atributos.

Por razones de espacio y claridad de representación, el diagrama de clases es desdoblado representándose por separado el contacto de los objetos que son externos al sistema. Se obtiene así un diagrama de contexto orientado a objetos. Además, para facilitar su identificación se usan diferentes colores.

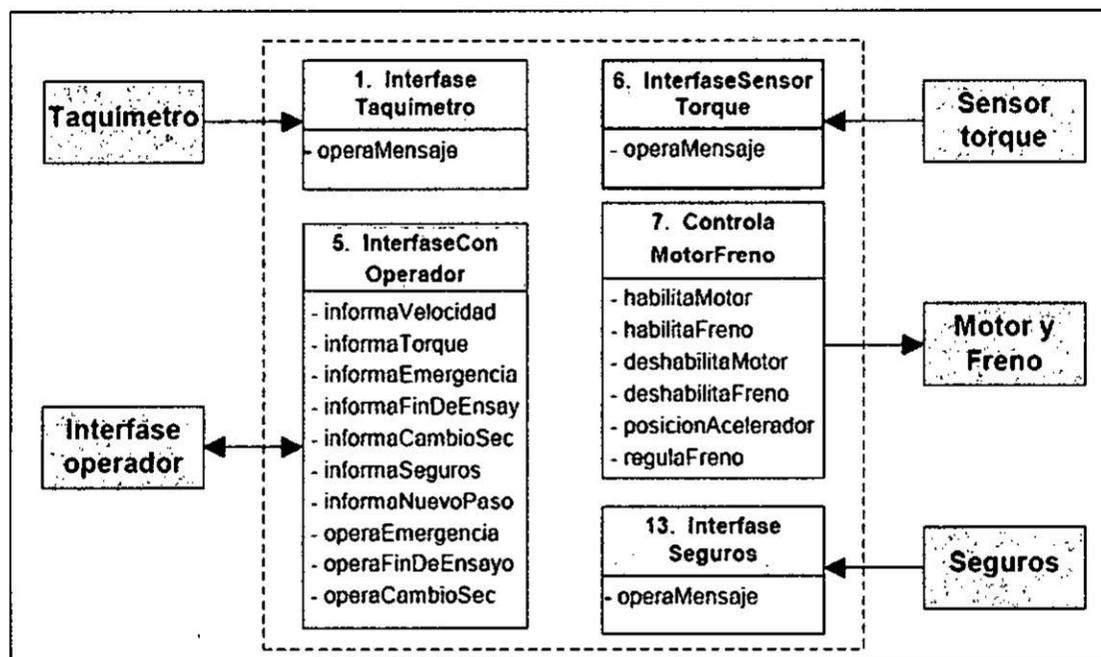


Figura N° 17 Diagrama de Clases / Contexto

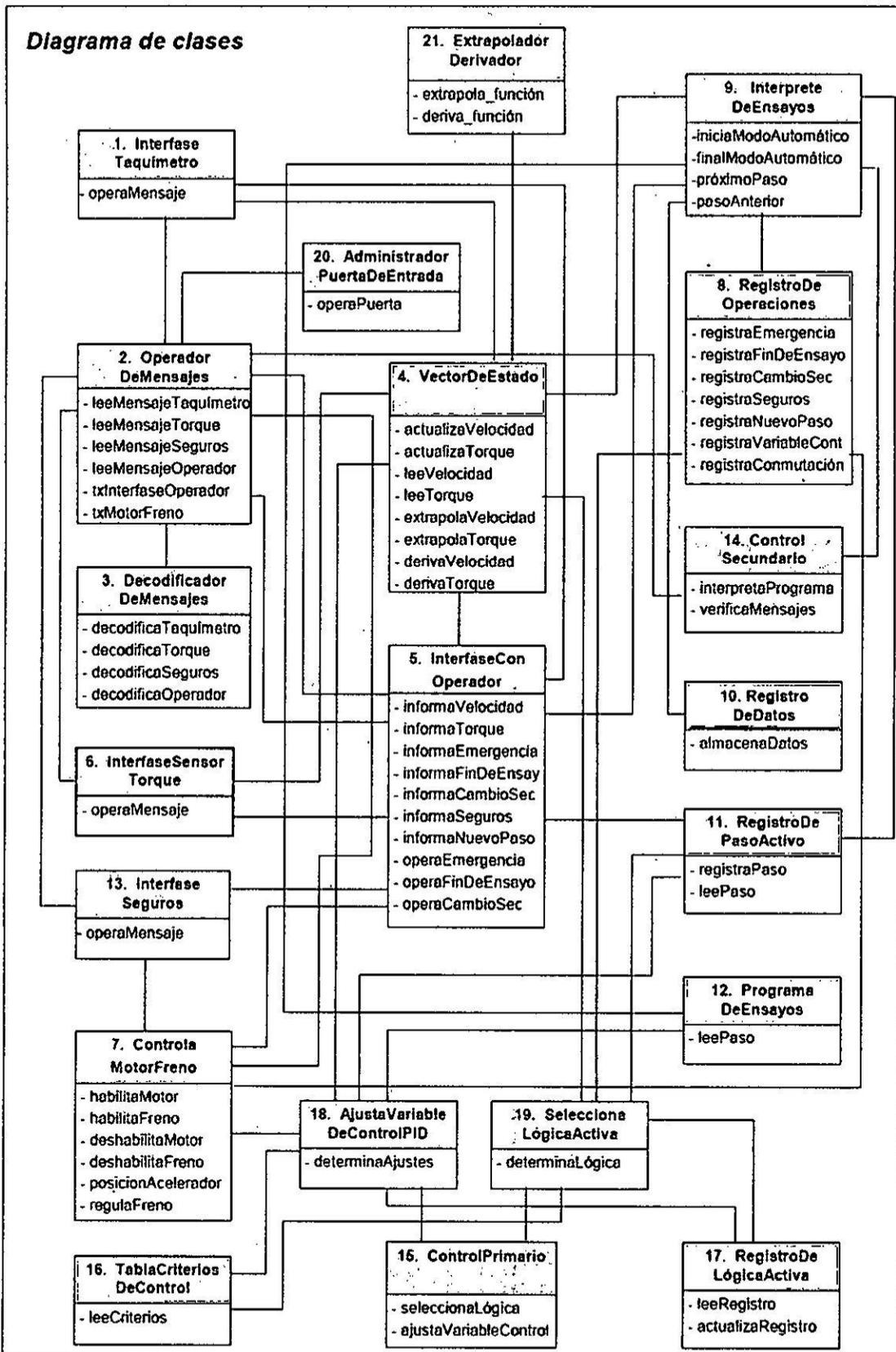


Figura N° 18 Diagrama de Clases

En la Figura 45, se presenta el diagrama de clases completo y en ambos diagramas se utiliza la siguiente convención en la asignación de colores:

	Objetos externos al sistema
	Objetos en contacto con elementos externos: entradas
	Objetos en contacto con elementos externos: salidas
	Objetos activados regularmente (temporales)
	Depósitos de datos
	Transformación de datos: procesos

CAPITULO V

V. VARIABLES E HIPOTESIS

5.1 Variables de la Investigación

Relacionando las variables relevantes que intervienen en el presente problema objeto de estudio, que conllevan a la explicación, demostración y comprobación de la formulada hipótesis, se han identificado las siguientes variables.

5.1.1 Variable independiente

La variable independiente, también conocida como variable manipulada, se encuentra en el centro de cualquier diseño experimental cuantitativo. Para nuestro presente trabajo de investigación la variable independiente es: **El modelamiento**.

5.1.2 Variable dependiente

En cualquier experimento real, un investigador manipula una variable independiente para influir en una o varias variables dependientes. Para nuestro presente trabajo de investigación tenemos las siguientes variables dependientes:

- **Variable U** = Taquimetro informa nueva velocidad.
- **Variable V** = Sensor de torque informa nuevo torque.
- **Variable W** = Seguros orden detención del motor.
- **Variable X** = Transición a nuevo paso de ensayo.

- **Variable Y** = Ajuste regular de variables de control.
- **Variable Z** = Conmutación de lógica de control.

5.2 Operacionalización de variables

Nº	EVENTO				ACTIVIDAD ESENCIAL	RESPUESTA	DESTINO
	Denominación	Tipo	Fuente	Estímulo			
1	<u>Taquímetro</u> informa nueva velocidad	ext.	<u>Taquímetro</u>	Mensaje	Recibir y registrar nueva velocidad del <u>motor</u>	Mensaje con valor de velocidad	<u>Interfase operador</u>
2	<u>Sensor de torque</u> informa nuevo torque	ext.	<u>Sensor de Torque</u>	Mensaje	Recibir y registrar nuevo torque en el <u>freno</u>	Mensaje con valor de Torque	<u>Interfase operador</u>
3	Operador solicita <u>parada de emergencia</u>	ext.	<u>Interfase operador</u>	Mensaje	Detener el <u>motor</u> en forma inmediata	Mensaje de detención del <u>motor</u>	<u>Motor y Freno</u> <u>Interfase operador</u>
4	Operador solicita <u>final del ensayo</u>	ext.	<u>Interfase operador</u>	Mensaje	Terminar <u>secuencia de ensayo</u> y detener el motor	Mensajes de <u>fin de ensayo</u> y detención del <u>motor</u>	<u>Motor y Freno</u> <u>Interfase operador</u>
5	Operador altera <u>secuencia de ensayo</u>	ext.	<u>Interfase operador</u>	Mensaje	Alterar la <u>secuencia de ensayo</u>	Mensaje con nuevo <u>paso de ensayo activo</u>	<u>Interfase operador</u>
6	<u>Seguros</u> ordenan detención del <u>motor</u>	ext.	<u>Seguros</u>	Mensaje	Detener el <u>motor</u> en forma inmediata	Mensaje de detención del <u>motor</u>	<u>Motor y Freno</u> <u>Interfase operador</u>
7	Transición a nuevo <u>paso de ensayo</u>	Tpo	<u>Unidad de Control</u>	Intervalo de Tiempo	Cumplir las previsiones del <u>programa de ensayo</u>	Mensaje con nuevo <u>paso de ensayo activo</u>	<u>Interfase operador</u>
8	Ajuste regular de <u>variables de control</u>	Tpo	<u>Unidad de Control</u>	Intervalo de Tiempo	Asegurar una condición estable de operación	Señales de control de <u>motor y freno</u>	<u>Motor y Freno</u>
9	Conmutación de <u>lógica activa de control</u>	cond. esp.	<u>Unidad de Control</u>	Condición de <u>vector de estado</u>	Definir la <u>lógica activa</u> de control más apropiada	---	---

5.3 Hipótesis

5.3.1 Hipotesis general

Un sistema modelado de una unidad de control para ensayos en motores de combustión está destinados a la realización de pruebas en forma completamente automática, incluyendo la adquisición de datos, monitoreo local y remoto, presentación de valores en pantalla, y migración de resultados de ensayos en bases de datos.

5.3.2 Hipotesis especifica

Un sistema de modelado de una unidad de control para ensayos en motores de combustión está destinado respecto a los analógicos, a la introducción de variantes que posibiliten una calibración mas amigable y la optencion de aun mejores desempeños. Para ello, disponer de un modelo que permita evaluar alternativas de control resultaría de un valor incalculable.

CAPITULO VI

VI. METODOLOGIA

6.1 Tipo de Investigación

Ya que se ha planteado la hipótesis a probar y se ha identificado las variables, que actúan sobre las unidades de control, el proceso de la investigación es de tipo: Aplicado-. matematico, además es Experimental -Tecnológica, cuyo inicio corresponde al mes de abril del año 2016 y terminó en noviembre del año 2016.

6.2 Diseño de la Investigacion

6.2.1 Diseño arquitectónico

El proceso de diseño arquitectónico comprende el establecimiento de un marco de trabajo estructural básico para el sistema, lo que implica identificar sus

componentes principales y la comunicación entre ellos (Sommerville, 2002). Así, uno de los aspectos principales que deben ser considerados es la forma en que se vincularán entre sí los componentes que ya fueron identificados en el esquema de la Figura 24, considerando para ello los límites que fueron establecidos en el punto 4.6.1.

Todas las funciones requeridas para resolver el problema planteado pueden ser presentadas globalmente en un esquema como el de la Figura 46, a partir del cual se identifican fácilmente los principales elementos involucrados. Obsérvese que, si bien todos ellos forman parte del sistema, la especificación de requerimientos desarrollada en el punto 4.6 se refiere exclusivamente a la unidad de control y, por lo tanto, sólo se incluyen las funciones que fueron identificadas como básicas para esta actividad. Todas las demás componentes fueron reconocidas como parte de su contexto.

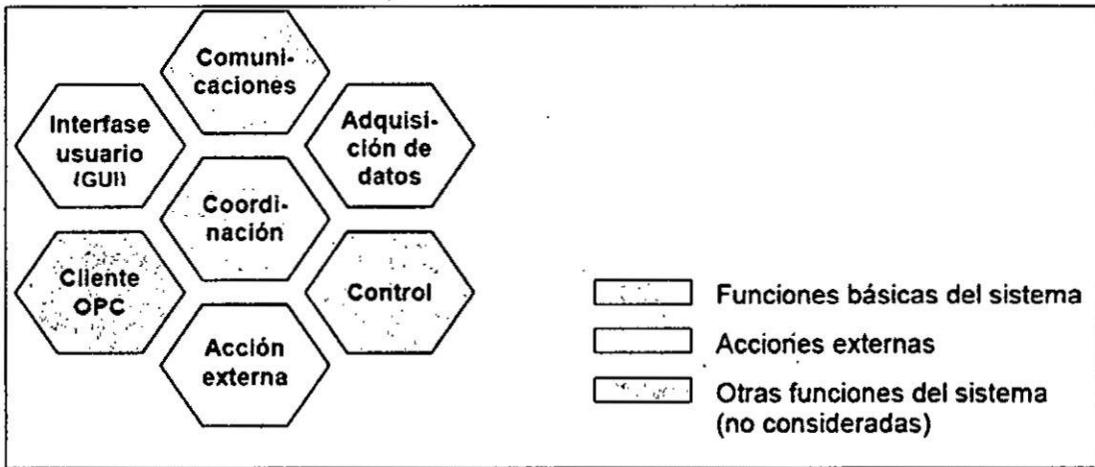


Figura N° 19 Funcionalidades y componentes

Para resolver este tipo de problemas se presentan dos opciones, los sistemas de control centralizados o los sistemas distribuidos.

En el primer caso, la totalidad de los elementos del sistema y sus componentes externos forman parte del hardware de la unidad de control. Aquí, las entradas y salidas analógicas y digitales están a cargo de placas de circuitos, que son coordinados por la CPU del sistema central. Estas placas contienen los

convertidores A/D, convertidores D/A, multiplexores, contadores y otros elementos.

En el caso de los sistemas distribuidos, las capacidades de percepción, acción y supervisión están descentralizadas y conectadas en red al sistema central a través de uno o más vínculos. Para esta comunicación, se utilizan protocolos propietarios que son habitualmente implementados en base a la norma RS-485, que presenta la ventaja de su relativa inmunidad a ambientes ruidosos (ruidos de radiofrecuencia).

Para este trabajo, se adoptó la segunda opción, es decir la de un sistema distribuido. Si bien un análisis comparativo de las dos opciones mencionadas excede el alcance de este trabajo, puede anticiparse que ambas son válidas y representan arquitecturas que son adoptadas con frecuencia en la actualidad.

A fines de implementar la arquitectura seleccionada se reconocieron los siguientes elementos principales: a) subsistema de adquisición de datos, b) subsistema de accionamiento exterior, c) interfase con el operador y d) unidad de control. El subsistema de adquisición de datos reúne al taquímetro, sensor de torque y seguros. Cabe aquí destacar que los controladores lógicos programables (PLC) representan, en la actualidad, una opción muy difundida para implementar los periféricos de percepción y acción (adquisición de datos y accionamiento exterior). En estos casos, se recurre a computadoras personales (PC) para implementar la interfase con el operador y la unidad de control. En el esquema de la figura 47 se representa la arquitectura adoptada usando, para mayor claridad, el mismo código de colores que en la Figura 46.

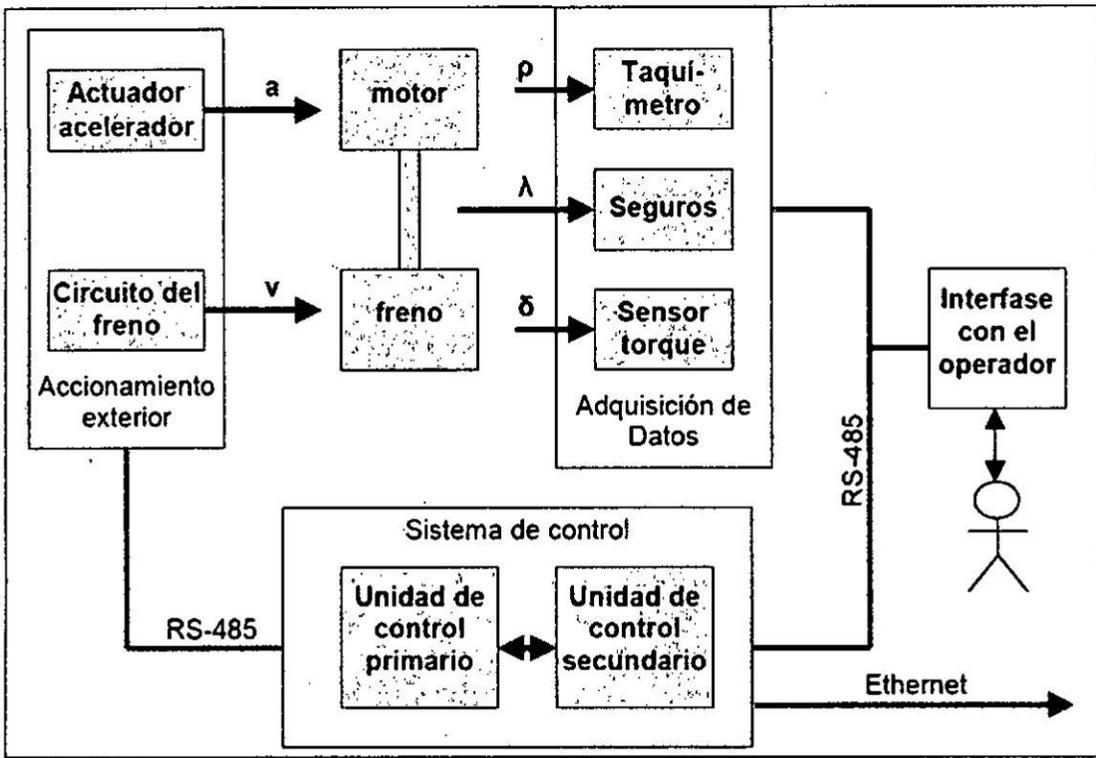


Figura N° 20 Arquitectura del sistema

En este tipo de arquitecturas, la unidad central coordina las comunicaciones, habilitando y consultando en forma sucesiva a los diferentes elementos de manera de evitar colisiones (polling). El vínculo ethernet es a su vez implementado como un cliente OPC de manera de asegurar completa compatibilidad con los standards del mercado y facilitar su supervisión por parte de sistemas SCADA (Supervisory Control And Data Adquisition).

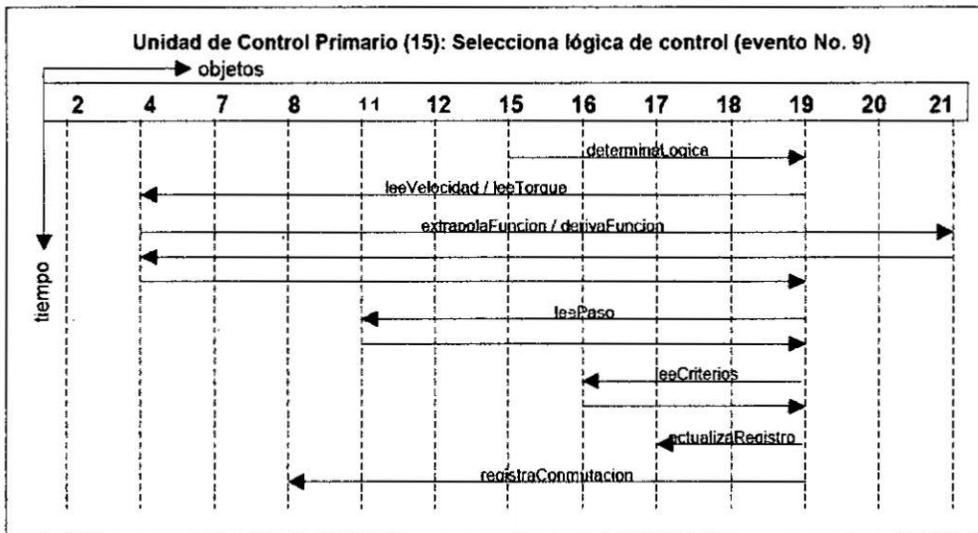
6.2.2 Diseño intermedio

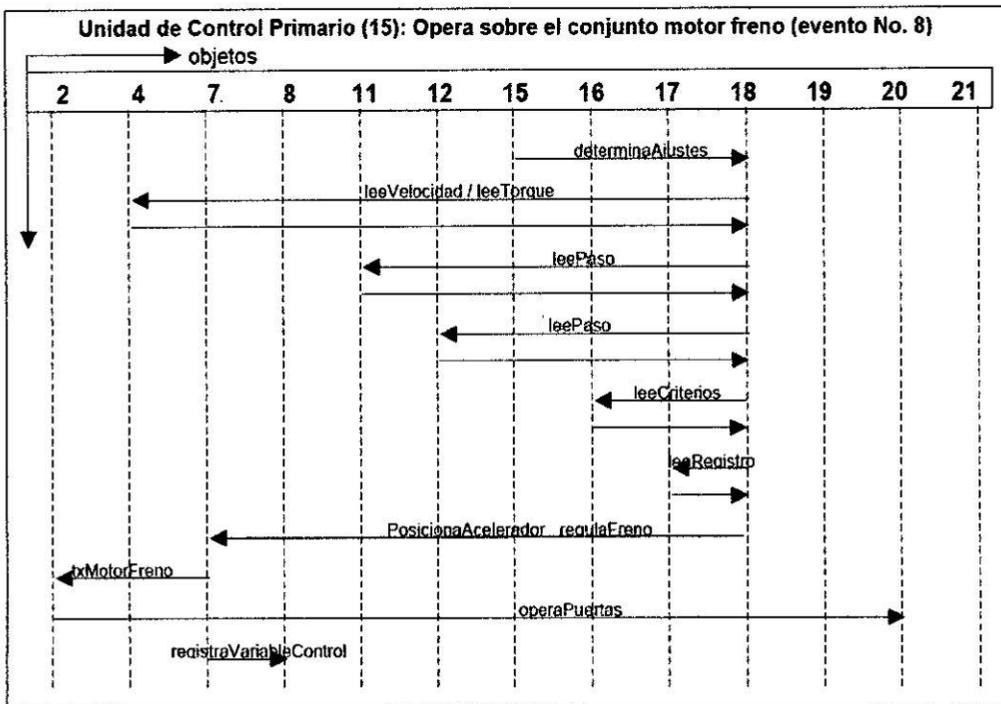
El diseño intermedio involucra un refinamiento de la solución presentada en la etapa anterior, y para ello se revisará la sucesión de operaciones que han sido previstas al identificarse las actividades principales en el modelo conceptual.

Obsérvese que, en la forma que está concebida la arquitectura del sistema, los objetos ControlPrimario (15) y ControlSecundario (14) deben operar en forma independiente, con frecuencias compatibles con la rapidez de respuesta requerida en cada caso.

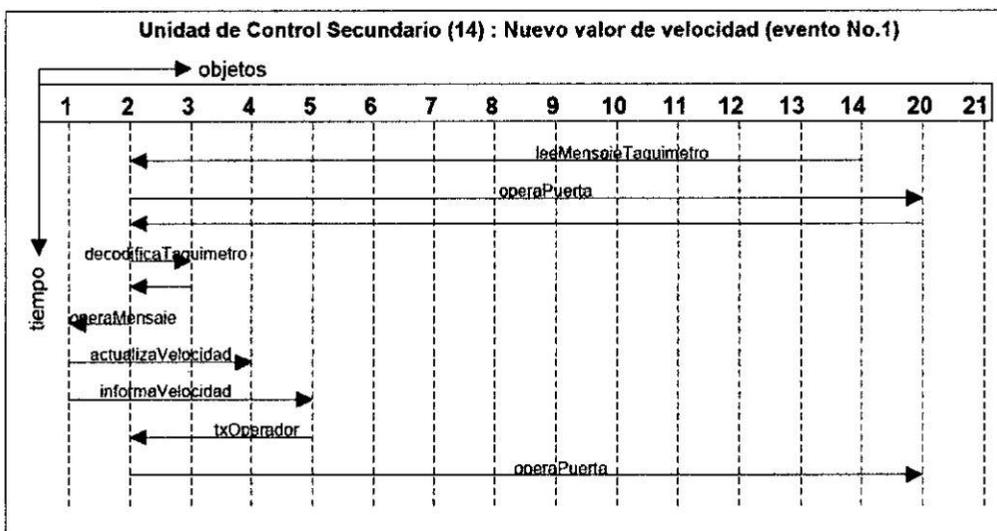
La frecuencia de operación del ControlPrimario encuentra un límite en la capacidad de muestreo del taquímetro y sensor de torque, ya que sólo se justifica una acción de control si en el vector de estado se dispone de datos actualizados para las variables controladas. Esto significa que la respuesta del sistema estará supeditada a la frecuencia de operación de estos instrumentos, y siempre existirán valores límites a partir de los cuales la estabilidad del objeto controlado no podrá ser garantizada. Es también importante observar que las constantes de las opciones de realimentación PID presentadas en las figuras 7, 8 y 9 (K_p , T_r y T_d) dependen de la frecuencia de operación de la unidad de control, por lo que una vez que sus valores han sido definidos, esta frecuencia no puede ser alterada. En resumen, ControlPrimario es un objeto activo que debe siempre operar a una frecuencia preestablecida.

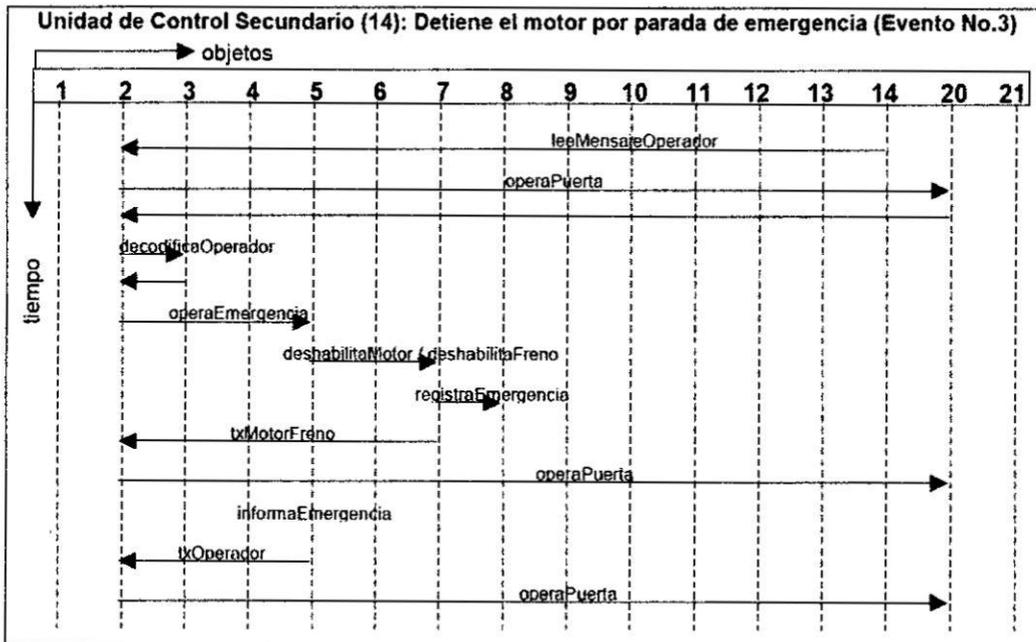
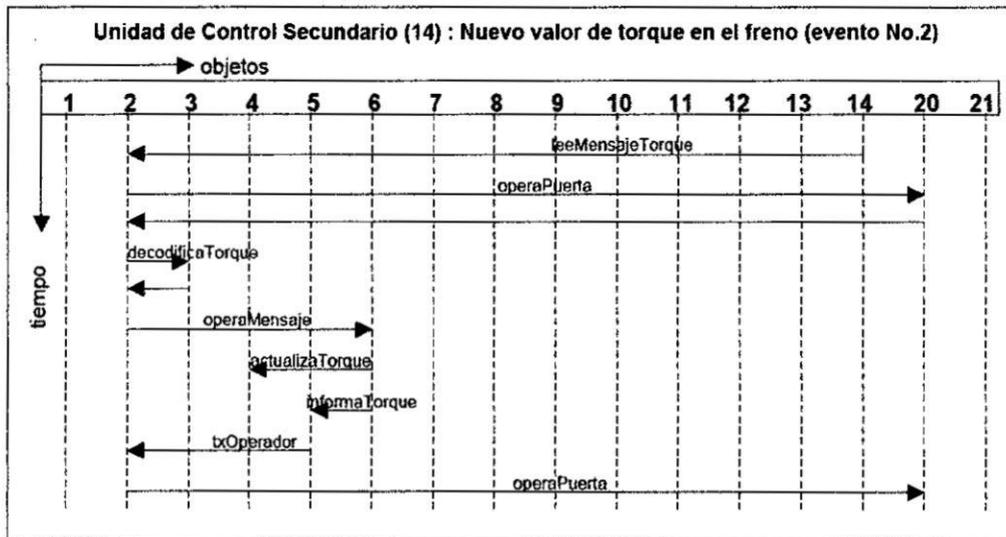
Se comienza por presentar los diagramas que corresponden a la Unidad de Control primario, identificando en cada caso las actividades esenciales cumplidas, que están a cargo de los objetos SeleccionaLógicaActiva (19) y AjustaVariable ControlPID (18),





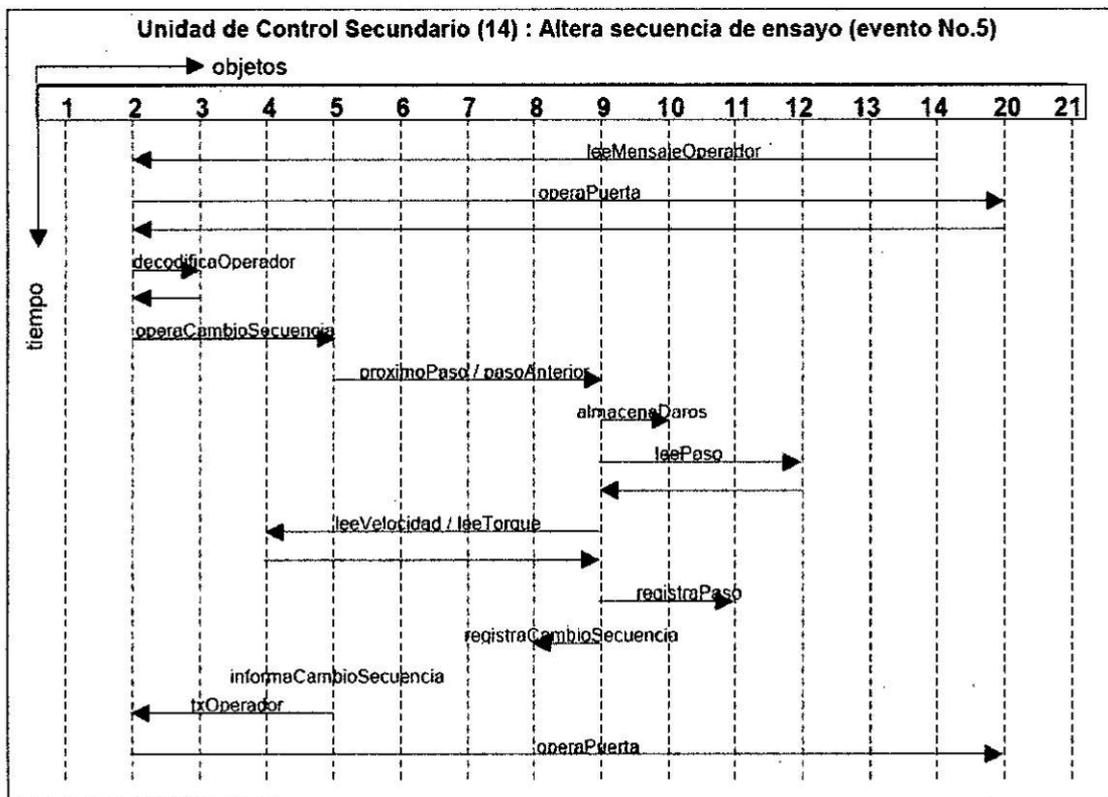
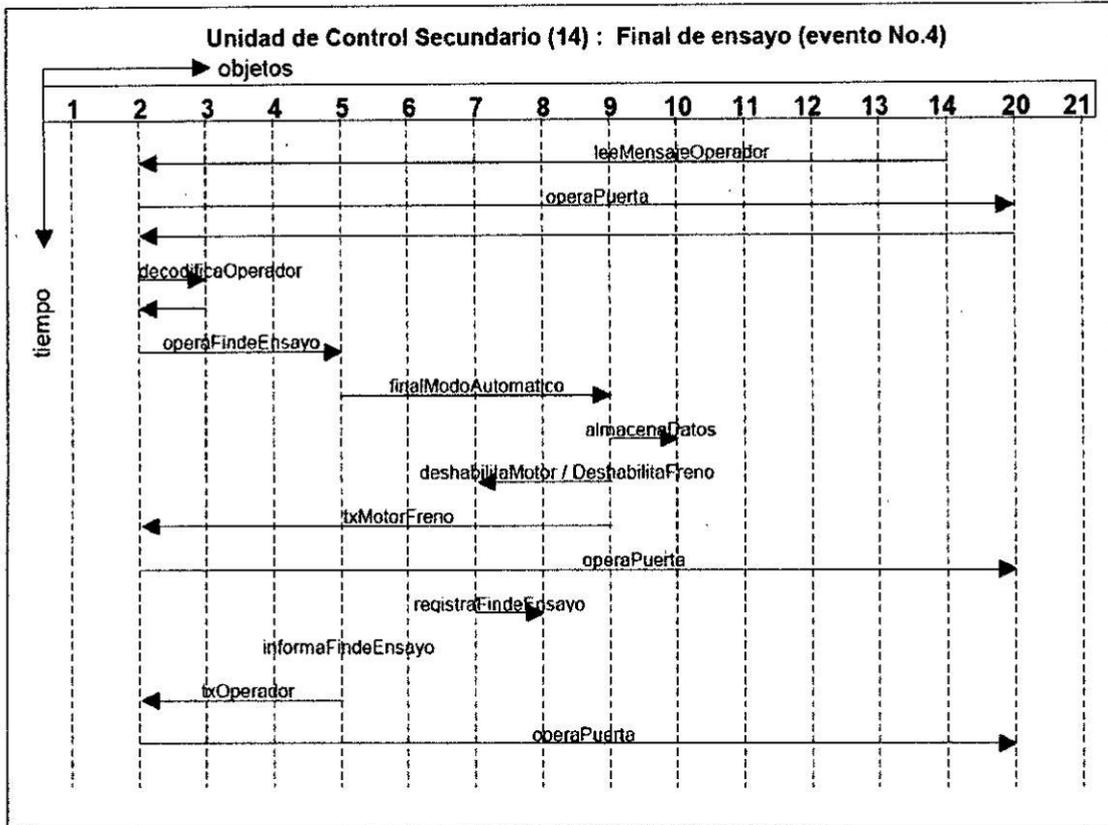
Muy diferente es el caso del objeto ControlSecundario, que tiene una responsabilidad activa y otra reactiva. La primera corresponde a la interpretación de los programas de ensayos y operará a muy baja frecuencia, ya que la unidad de tiempo de un paso de ensayo es el segundo. La segunda responsabilidad se refiere a la recepción de mensajes del exterior (interpretaMensajes), lo que conduce a una operación reactiva y asíncrona que estará estimulada por las interrupciones originadas en la puerta de comunicaciones. A continuación, se presentan los diagramas de secuencia que corresponden a la unidad de control secundario.

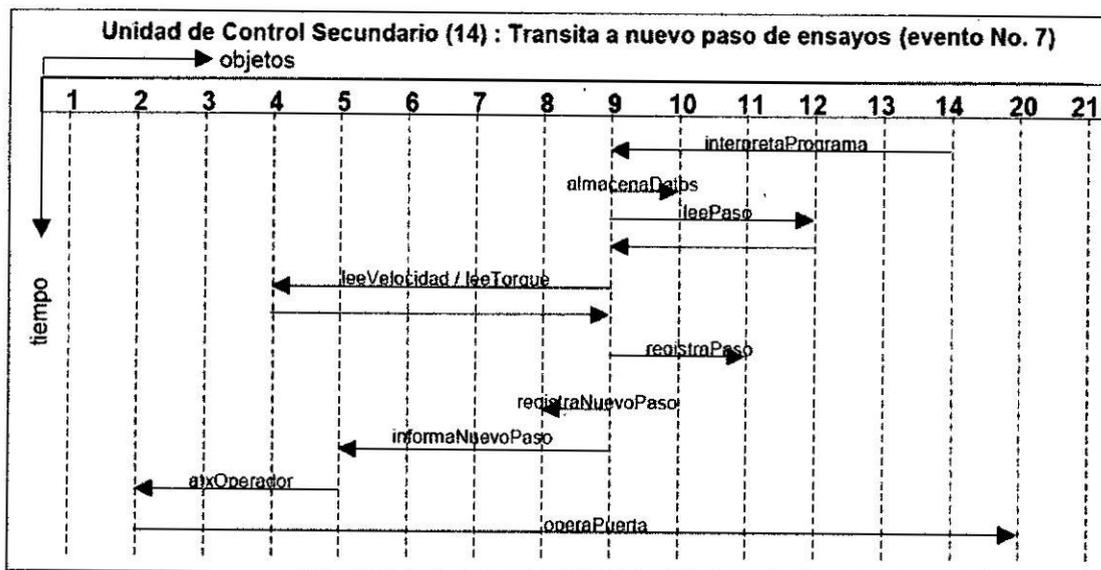
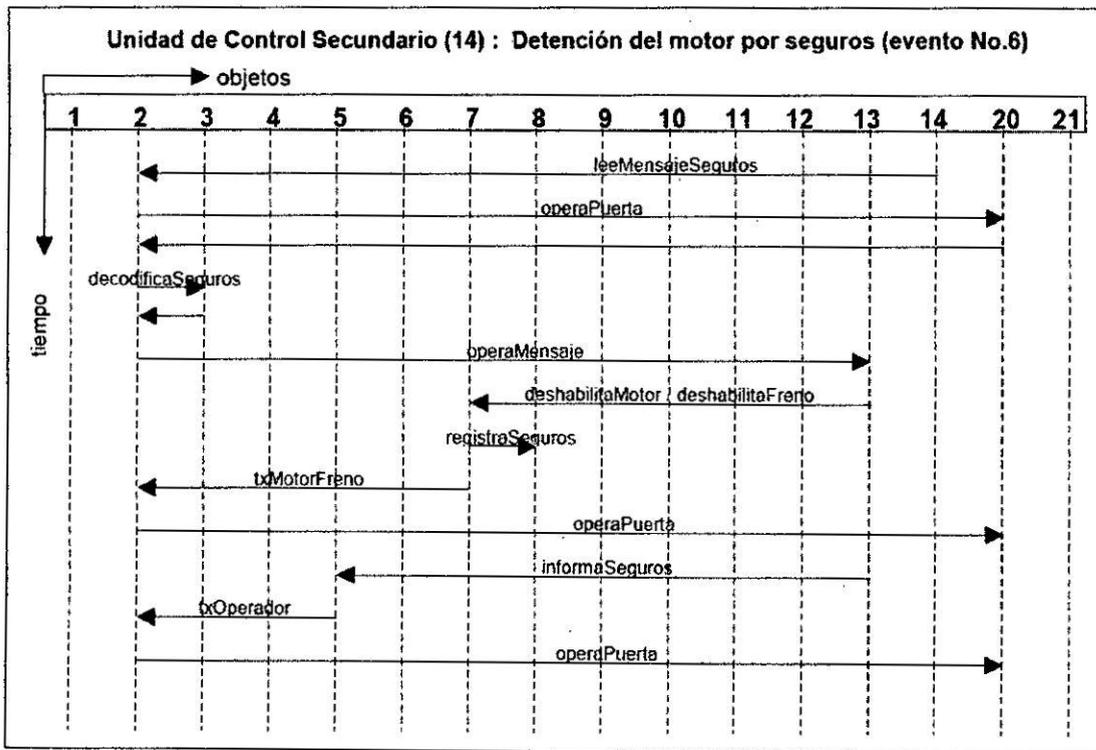




Estos primeros tres diagramas de secuencias corresponden a la recepción de mensajes del taquímetro, sensor de torque y detención del motor por parada de emergencia. Este último mensaje proviene de la interfase con el operador, al igual que los mensajes de detención del motor y cambio de la secuencia de ensayos, que se presentan a continuación. Por último, se representan los diagramas de secuencia correspondientes a la detención del motor por seguros e interpretación del programa de ensayos. Tal como ya fue anticipado, este último objeto cumple una actividad temporizada, verificando regularmente el cumplimiento de la

secuencia de ensayo prevista en el programa.





En este punto, es conveniente destacar que los patrones de diseño (design patterns), que vienen teniendo una gran difusión en todos los campos de aplicación de la ingeniería de software, también ocupan un lugar cada vez más importante en el diseño intermedio de sistemas de tiempo real. Si bien los sistemas de tiempo real operan sobre una gran diversidad de plataformas, y con finalidades muy

diferentes, hay soluciones que son consideradas "clásicas" y con muy pocas variantes son universalmente aceptadas para resolver problemas específicos. En particular, pueden mencionarse los subsistemas de comunicaciones, implementación de lógicas de control e interfases con el usuario como las áreas en las que con mayor facilidad se adoptan patrones de diseños normalizados. Sin embargo, estas soluciones normalizadas son en su mayoría conceptuales y no hay todavía catálogos de patrones de diseño de uso general (Zalewski, 2002), por lo que al reconocerlas como patrones se está más bien haciendo referencia a enfoques que han demostrado ser efectivos y son considerados como standards "de-facto". Además, un sistema de tiempo real involucra también numerosas operaciones convencionales, que encuentran una fuente de posibles soluciones en los catálogos de patrones de diseño de aplicación general (Gamma et Al, 1995).

6.2.3 Diseño detallado

En concordancia con los mismos límites ya comentados, se centra el diseño detallado en las unidades de control primario y secundario. Para este tipo de aplicación y este nivel de definición, los autómatas finitos representan, en muchos casos, el mejor medio para documentar sus capacidades y particularidades operativas.

6.2.4 Unidad de control secundario

La clase que representa la unidad de control secundario dispone de dos métodos. Éstos, que están destinados a interpretar los programas de ensayos y a recibir los mensajes provenientes del exterior, se describen a continuación.

a. Intérprete de ensayos

El intérprete de ensayos es un módulo activo que interpreta regularmente las sentencias que definen un programa de ensayos, tal como fue presentado en el punto 4.4.8. La actividad cumplida por este módulo queda convenientemente representada con el grafo de la Figura 48 y el significado de cada transición es presentado en la Tabla 9.

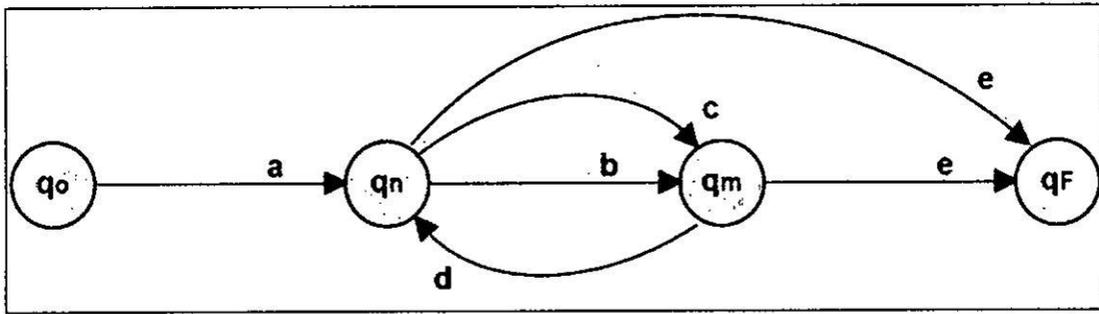


Figura N° 21 Intérprete de comandos

entrada	comando	evento	Descripción de la función de transición
a	Inicia ensayo	4	$f(q_0, a) = q_n ; n = 1$
b	Transición normal a próximo paso	7	$f(q_n, b) = q_m ; m = n + 1$ (próximo paso)
c	Avanza a próximo paso	5	$f(q_n, c) = q_m ; m = n + 1$ (próximo paso)
d	Retrocede a paso anterior	5	$f(q_m, d) = q_n ; n = m - 1$ (paso anterior)
e	Final del ensayo	4	$f(q_n, e) = q_F ; f(q_m, e) = q_F$

Tabla N° 07 Transiciones en el intérprete de comandos

En las instalaciones para ensayos de motores que están destinadas a pruebas de larga duración se incluyen, en el programa de ensayos, otros tipos de instrucciones, que están destinadas a permitir ciclos cerrados anidados y otras facilidades. En estos casos, los intérpretes de ensayos quedan representados por autómatas a pila, y los programas son definidos por lenguajes Tipo 2 (Chomsky).

b. Recepción de mensajes del exterior

Tal como fue propuesto en el diseño arquitectónico del sistema, la unidad de control secundario está conectada a un bus de comunicación serial RS-485 por el que recibe los mensajes provenientes del taquímetro, sensor de torque, seguros e interfase del operador. Estas unidades son presentadas en el esquema de la Figura 49:



Figura N° 22 interconexión de bus serial

A los fines de esta comunicación, se adopta un protocolo común, que prevé múltiples comandos para el caso de la interfase con el operador, tal como el presentado en la Tabla 10. Para representar la recepción de estos mensajes se recurre también a un autómata finito, que es representado en la Figura 50. La principal función de este autómata es la validación de la integridad y consistencia de los mensajes recibidos. Obsérvese que, con el fin de facilitar la interpretación de las transiciones previstas, los símbolos que pertenecen al alfabeto de entrada del autómata son identificados en el encabezamiento de las columnas de la Tabla 10.

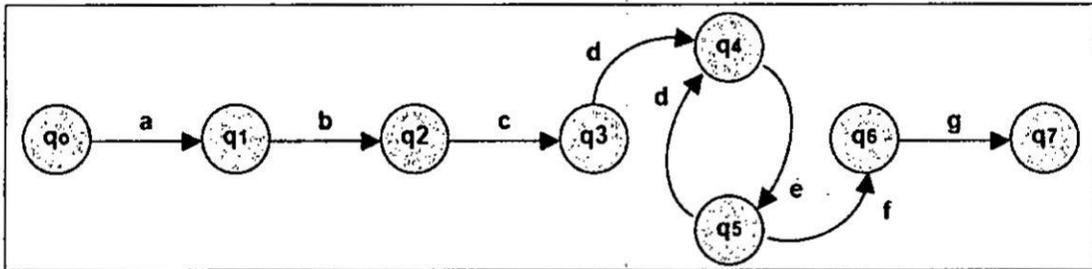


Figura N° 23 Validación de mensaje

a	b	c	d	e	d	...	F	g
Inicio de mensaje	Origen del mensaje	Cantidad de caracteres	Cdo	Argu-mento	Cdo	CRC	Fin de mensaje

Tabla N° 08 Protocolo de comunicaciones

6.2.5 Unidad de control primario

La unidad de control primario fue definida en la Figura 4 como una unidad conmutada de control. En efecto, se trata de una unidad activa que tiene por finalidad seleccionar la lógica de control más conveniente en cada caso y ajustar regularmente las variables de control.

a. Selección de la lógica activa

La selección de la lógica activa es convenientemente representada por un autómata finito, cuyo conjunto de estados queda definido por las opciones de realimentación que están disponibles en cada caso, resumidas en la Tabla 11.

Estas opciones de realimentación ya fueron definidas en los esquemas de las Figuras 33, 34, 36 y 37. Obsérvese que, para implementar estas opciones, se dispone a su vez de los esquemas de realimentación definidos como PID, PI-D y PID-D de las Figuras 7, 8 y 9; por lo que, en caso de considerarlas, la cantidad de combinaciones posibles sería sustancialmente mayor.

Sin embargo, el verdadero problema no es la definición del conjunto de estados de este autómeta, sino su función de transición, es decir las condiciones bajo las cuales transitará de un estado a otro, y a las que ya se hizo referencia en las expresiones 24 y 27 al definirse el control conmutado en el punto 2.3.1. Algunas de estas transiciones son obvias, y pueden ser anticipadas. Otras, por el contrario, sólo serán descubiertas a partir de un detallado proceso de evaluación del comportamiento de la unidad de control y del objeto controlado. Esto significa que, en la práctica, la función de transición no puede ser completamente definida con anticipación; y por ello es conveniente posibilitar su progresiva actualización. Una alternativa, quizás la más ventajosa, es asociar estas transiciones a los resultados de reglas de producción que son almacenadas en un archivo de configuración, que será leído al iniciarse la operación del sistema. Así, en este archivo, que es denominado tabla de criterios de control, bajo la forma de reglas de producción se incorporarán nuevas transiciones a medida que la experiencia lo haga aconsejable.

Estado	Descripción	Variable de Control (VC)	Variable de Proceso (VP)	Condición especial
Q0	Motor descargado a velocidad mínima	Ninguna	Ninguna	Arranque y detención
Q1	Motor libre o con frenado constante	Posición del Acelerador	Velocidad	Tensión de frenado constante
Q2		Posición del acelerador	Torque	
Q3	Motor a carga constante	Tensión en el Freno	Velocidad	Acelerador en posición constante
Q4		Tensión en el Freno	Torque	
Q5	Operación General (Caso 1)	Posición del Acelerador	Torque	Ajuste simultáneo de la tensión de frenado y posición del acelerador
		Tensión en el Freno	Velocidad	
	Operación General	Posición del Acelerador	Velocidad	

q ₆	(Caso 2)	Tensión en el Freno	Torque	
----------------	----------	---------------------	--------	--

Tabla N° 09 Condiciones de Control

En resumen, la selección de la lógica de control está a cargo de un autómata finito (Ec. 21) que es representado en la Figura 51, donde $S = \{ q_0, q_1, q_2, q_3, q_4, q_5, q_6 \}$, $S_0 = q_0$, $S_F = q_0$ y $\Sigma = \{ a, b, c, d, e, f, g, h, i, j, k, l \}$.

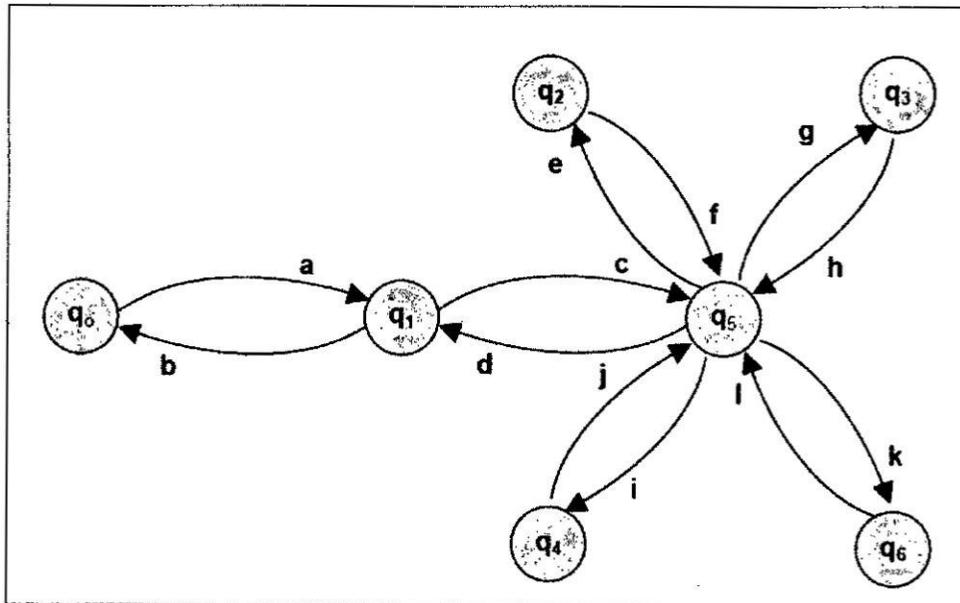


Figura N° 24 Selección de la lógica de control

Este autómata se comporta como una máquina de Moore, en donde, sus salidas están directamente relacionadas con los estados adoptados en cada instante de tiempo. No es por lo tanto necesario representar estas salidas. Como ya fue expuesto, la función de transición de este autómata queda vinculada a las reglas de producción que son leídas de la tabla de criterios de control y fue anticipada conceptualmente por las expresiones 26 y 27 del punto 2.1.4.

b. Ajuste de las variables de control

Este método debe determinar los errores en cada una de las variables de proceso y luego, a partir de los esquemas de realimentación que correspondan a cada caso (lógica activa), calcular las magnitudes de los ajustes que deben introducirse en las variables de control. Luego, a través de la interfase RS-485, estos ajustes son comunicados a los elementos de accionamiento exterior, que son el actuador de

acelerador y el circuito de excitación del freno.

Este método debe disponer de todas las opciones de control y realimentación, para ser aplicadas en cada caso, pero su implementación no ofrece dificultades especiales.

6.2.6 Programación

Un sistema es el producto final que materializa el objetivo de un proceso de desarrollo de software y su finalidad es cumplir con las especificaciones previstas en sus requerimientos. Para ello, el sistema debe ser escrito en cierto lenguaje, y, por lo tanto, es natural que el punto de partida de la fase de programación sea la selección de dicho lenguaje.

Ya fue anticipado que en esta selección inevitablemente influirá la preferencia y experiencia del o de los programadores. Sin embargo, se trata de una decisión técnica que debe apoyarse esencialmente en argumentos objetivos. Para ello, en el punto 3.3.1 de este trabajo se identificaron los atributos que son relevantes en los lenguajes destinados a sistemas de tiempo real.

En este caso, el lenguaje seleccionado fue Delphi, que es una implementación orientada a objetos de Pascal. Las respuestas que ofrece Delphi a los atributos requeridos son las siguientes:

a. Control de tiempo

Se dispone de una resolución de 1 ms en la gestión del tiempo, pudiendo generarse secuencias de eventos que tengan como mínimo este período. El valor indicado es suficiente para los objetivos de este sistema.

b. Concurrencia

Soporta la definición de procesos "multihilo" y la activación periódica de objetos a partir de eventos temporales, brindando el soporte necesario para procesos concurrentes.

c. Seguridad

Se dispone de herramientas de “debugging” y directivas en el compilador que permiten un control minucioso del código. En caso de producirse un error, el programa ejecutable brinda referencias precisas sobre su causa, por lo que se facilita la obtención de productos finales sólidos y confiables.

d. Portabilidad

Delphi genera programas ejecutables completamente autocontenidos, lo que facilita la portabilidad del sistema dentro del entorno de plataformas operativas Windows, sin necesidad de librerías externas para su posterior operación. Además, la portabilidad a otras plataformas está asegurada ya que, para los sistemas UNIX y LINUX, existe una versión de Delphi denominada Kylix. Por último, hay también una versión multiplataforma y completamente libre de Delphi llamado Lazarus que tiene su origen en la comunidad del software libre. El nombre Lazarus proviene del Oráculo de Delfos.

e. Eficiencia

Delphi, al igual que Pascal, es un lenguaje fuertemente tipeado, y esto contribuye a su rapidez en tiempo de ejecución. Además, se trata de un compilador y link-editor que disponen de diversas opciones de optimización, lo que permite generar programas ejecutables muy eficientes.

Adicionalmente, pueden mencionarse otros atributos que no están asociados a la operatividad en tiempo real de los sistemas, pero que sí están relacionados con la metodología de diseño propuesta. Éstos son:

f. Orientación a objetos

Delphi es un lenguaje claramente orientado a objetos, con un buen manejo de clases, herencia y polimorfismos. Cubre todas las necesidades de un desarrollo complejo, bajo una rigurosa implementación del paradigma de objetos.

g. Vigencia y compatibilidad

Las recientes versiones de Delphi demuestran su vigencia, y en todos los casos, puede comprobarse una excelente compatibilidad con versiones anteriores.

6.2.7 Sistemas operativos

Para merecer la calificación de RTOS (Real-Time Operating System) un sistema operativo debe cumplir numerosas condiciones, destacándose las siguientes tres características que son consideradas básicas:

- a) **Determinismo**: se dice que un sistema operativo es determinista cuando se puede calcular el máximo tiempo que puede demandar una invocación a cualquier recurso del sistema.
- b) **Latencia de interrupciones**: se denomina "latencia de interrupciones" al máximo tiempo que transcurre desde que una señal de interrupción llega al procesador hasta que es ejecutada su ISR (Interrupt Service Routine) asociada. Las interrupciones son elementos clave de los RTOS. Cuando una interrupción tiene lugar, el procesador debe realizar varias tareas antes de ejecutar su ISR asociada, tales como: 1) identificar la interrupción, 2) comprobar su prioridad según el esquema de prioridades vigente y 3) preservar la actividad actual en caso que deba ser interrumpida.
- c) **Cambio de contexto**: se refiere al intervalo de tiempo que es necesario para realizar un cambio de contexto (context switch o thread switch), denominándose así al proceso de cambiar de una tarea a otra.

Las demás condiciones están referidas a su capacidad para acceder a los recursos del sistema, cantidad de memoria requerida, sistema de protección de memoria, arquitectura micro kernel y entorno de desarrollo apropiado. Además, los sistemas operativos de tiempo real modernos deben cumplir las recomendaciones POSIX (Portable Operating System Interface X) del IEEE que procura ofrecer un entorno normalizado de recursos básicos a los programadores.

A partir de las características enunciadas, se deduce que se trata de sistemas operativos especiales, altamente especializados, de escasa difusión y habitualmente de elevado costo. En este último caso, se trata de productos comerciales, aunque debe destacarse que hay también numerosos sistemas operativos de tiempo real que son desarrollados por universidades y centros de investigación. Uno de los clásicos que corresponde al primer grupo es el QNX-

Neutrino, que además de cumplir con las normas POSIX, exhibe características de multitarea, multiusuario y multiplataforma.

Las circunstancias señaladas y la creciente demanda de este tipo de sistemas han contribuido a que numerosas aplicaciones de tiempo real sean implementadas sobre sistemas operativos convencionales, especialmente Windows. Esta tendencia se ve favorecida por la gran difusión y aceptación que tienen las diversas versiones de Windows en los ámbitos industriales. Estos sistemas pueden ser denominados de "cuasi Tiempo Real" y son empleados en aplicaciones tolerantes a cierta flexibilidad en el manejo de los tiempos que ya fueron calificadas como básicos. Aun aceptando que no se trata aquí de verdaderos sistemas de tiempo real, el masivo incremento de estas aplicaciones sobre entornos Windows es preocupante, tanto por la falta de fiabilidad y robustez que son características de estos sistemas operativos, como también por los desmedidos recursos de hardware que demandan.

Para la implementación de estos sistemas de "cuasi Tiempo Real" el clásico MS-DOS era una excelente opción, que lamentablemente ha perdido vigencia por su falta de compatibilidad con los recursos disponibles en la actualidad. Por otro lado, es posible reconocer a Windows como una opción aceptable para llevar adelante procesos de simulación y en razón de su amplia disponibilidad es adoptado para este trabajo.

6.2.8 Implementación

Para poder operar el simulador del sistema de control de ensayos de motores, es indispensable la implementación de todos los módulos que ya fueron representados en la Figura 24 y descriptos en detalle en el punto 4.4. La implementación de estos módulos presenta los siguientes aspectos particulares:

a. Unidad de Control

Tanto en la especificación de requerimientos como en la fase de diseño, se puso el foco en los principales componentes de esta Unidad, que son los de control primario y secundario. Su implementación contempla los aspectos particulares ya

definidos en estas fases de especificación y diseño.

b. Conjunto motor-freno

El comportamiento del motor y freno es representado por redes neuronales, y para acoplar ambos elementos, se opta en esta oportunidad por un eje rígido. Por lo tanto, la respuesta del conjunto se obtiene integrando la ecuación 83 del punto 4.4.3.b.

c. Taquímetro

El taquímetro se activa periódicamente, según lo enunciado en el punto 4.4.4, y tiene a su cargo la conversión de la señal de velocidad, tal como fue detallado en el punto 4.5.2.

d. Sensor de Torque

El sensor de torque se activa también periódicamente, según ya fue detallado en el punto 4.4.5. A los fines de concretar las lecturas del torque del freno se debe convertir su señal según se describe en el punto 4.5.1.

Por otra parte, ya fue establecida la imposibilidad de medir el torque en el motor y por esta razón su valor es determinado a partir de la ecuación 89. Para ello, se calcula la aceleración del motor mediante la ecuación 72.

e. Seguros

La función de la unidad de seguros fue descrita en el punto 4.4.6 y sus señales son convertidas según lo previsto en 4.5.3.

f. Interfase con el operador

Sus características fueron enunciadas en el punto 4.4.9. Por tratarse de una interfase de simulación, se ha previsto una representación gráfica central sobre un plano cartesiano ortogonal y diversas opciones que están destinadas a desplegar ventanas que detallan el estado de los diferentes procesos numéricos (Anexo A).

6.3 Poblacion y muestra

Poblacion

Es el conjunto de elementos que son objeto de estudio estadístico. Para nuestro estudio de investigación son todas las empresas industriales de Lima metropolitana.

Muestra

Es una parte de población con la que realmente se realiza el estudio. Para nuestro estudio de investigación es las empresas industriales de Cercado de Lima.

6.4 Tecnicas e instrumentos de recolección de datos

La recolección de datos se realizo via encuesta fisica a los operadores calificados de los motores de combustión interna de las diversas empresas de Cercado de Lima.

6.5 Procedimientos de recolección de datos

Previo a la encuesta realizada a los operadores calificados de los diversos motores de combustión interna presente en las empresas de Cercado de Lima, se les brindo una breve charla sobre el alcance de nuestro trabajo de investigación asi hacer incapie que la información que ellos nos brinde nos es de suma utilidad.

6.6 Procesamiento de recolección de datos

El procesamiento y análisis de la recolección de datos fue estadístico – analítico de cada una de las encuestas realizadas al personal calificado en operar los motores de combustión interna.

CAPITULO VII

VII. RESULTADOS

7.1 Resultados parciales

Evaluación del simulador

Tal como ya fue señalado en la consideración del marco metodológico, el

desarrollo de un sistema debe contemplar una serie planificada de pruebas. Su objeto es comprobar que, bajo todas las condiciones posibles de operación, cada módulo cumplirá adecuadamente con sus responsabilidades y que, el producto final, será capaz de exhibir un correcto funcionamiento (punto 3.4 y Figura 20). Si bien este control progresivo es siempre recomendable, reviste particular importancia en aquellos sistemas que incluyen la implementación de algoritmos complejos.

Debe observarse que, a pesar de que este documento ha centrado su atención en el diseño de la unidad de control, han debido también diseñarse e implementarse los demás módulos de software, que son necesarios para simular el comportamiento del conjunto motor-freno y los elementos de adquisición de datos. Todos estos componentes principales ya fueron identificados en el punto 4.4 de la fase de definición de requerimientos, su interconexión se presentó en el esquema de arquitectura del sistema en la Figura 47 y el alcance de la implementación de cada uno fue resumido en el punto 5.

La secuencia ordenada de pruebas previstas, para la verificación del sistema en su conjunto y de sus componentes, son las siguientes:

- a) Testeo del algoritmo de integración de las ecuaciones del movimiento
- b) Testeo de los algoritmos de extrapolación y derivación
- c) Testeo de las funciones de aproximación del comportamiento del motor y freno
- d) Evaluación de la simulación de la respuesta dinámica del conjunto motor-freno en ausencia de acciones de control.
- e) Evaluación de la estimación del torque del motor a partir del torque medido en el freno.
- f) Evaluación de la interacción del conjunto motor-freno y la unidad de control.

Con el fin de disponer de valores de referencia para el cálculo de errores, en las evaluaciones previstas en los puntos "a", "b" y "c" se utilizarán expresiones matemáticas conocidas y, que tengan un comportamiento similar al esperado en los elementos reales. Debido a la disponibilidad de valores de comparación, se

otorga gran importancia a la exhaustiva evaluación de estos procesos que pueden ser contrastados con soluciones matemáticas exactas. Por el contrario, la posterior evaluación del desempeño del sistema en su conjunto encerrará muchas dificultades por la poca disponibilidad de resultados. En efecto, sólo para algunas condiciones de operación se dispondrá de valores experimentales.

Para cada una de las pruebas previstas en la secuencia de verificación, se ha hecho una breve selección de resultados, que son presentados a continuación junto con una descripción de las expresiones o valores tomados como referencia.

En este gráfico están superpuestas las curvas de la función original (Ec.103) y las que corresponden a los resultados de integrar su derivada (Ec.105) por el método de Runge-Kutta de 4° orden (Ec. 36 y 37). La escala del grafico no permite apreciar el error, que en este caso estuvo siempre por debajo del 0,5%. Además, al evaluarse la función en períodos prolongados de más de 10 minutos, se pudo comprobar el buen desempeño del método de integración, ya que el error siempre se mantuvo acotado por debajo del valor indicado.

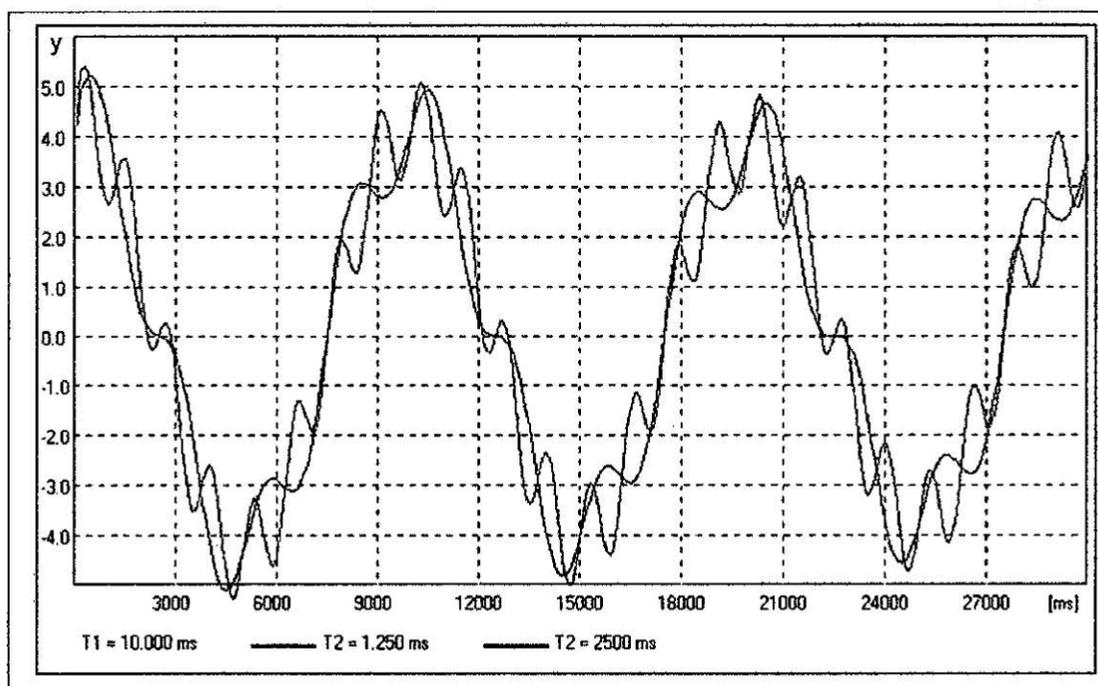


Figura N° 25 Función de evaluación de la integración numérica

También se evaluó el comportamiento de estos métodos de integración numérica

con diversas relaciones entre los períodos de las dos armónicas. En la Figura 53, se resumen los resultados obtenidos con diferentes intervalos de integración ($T_3 = \Delta t$) y distintas relaciones entre los períodos T_1 y T_2 .

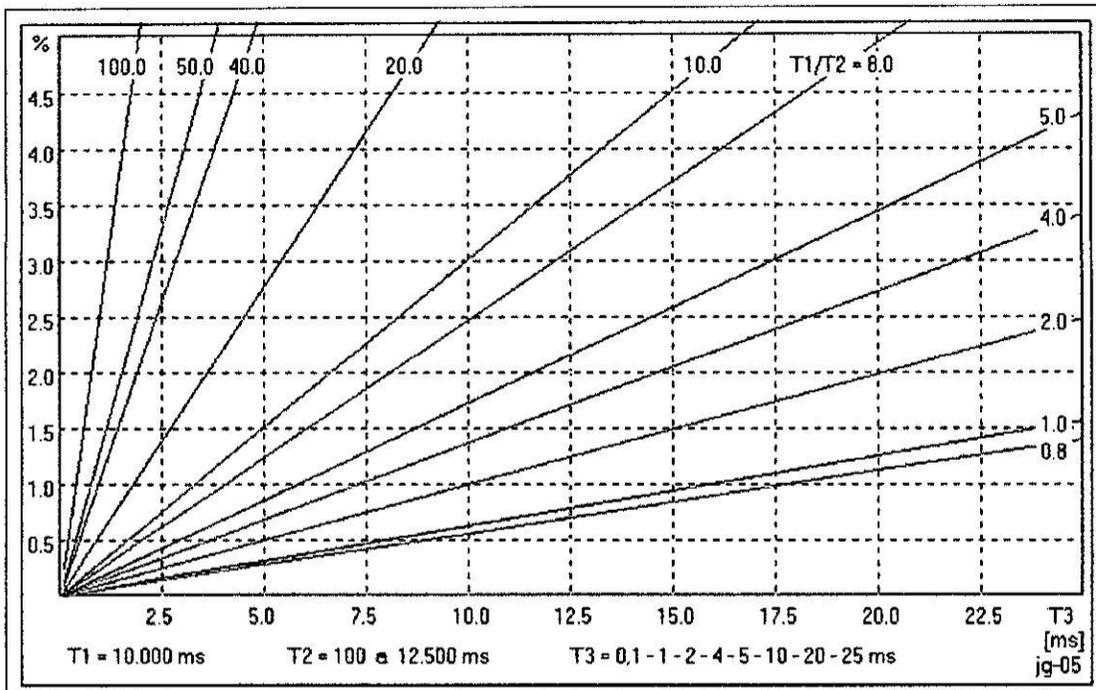


Figura N° 26 Evolución del error con el intervalo de integración T_3

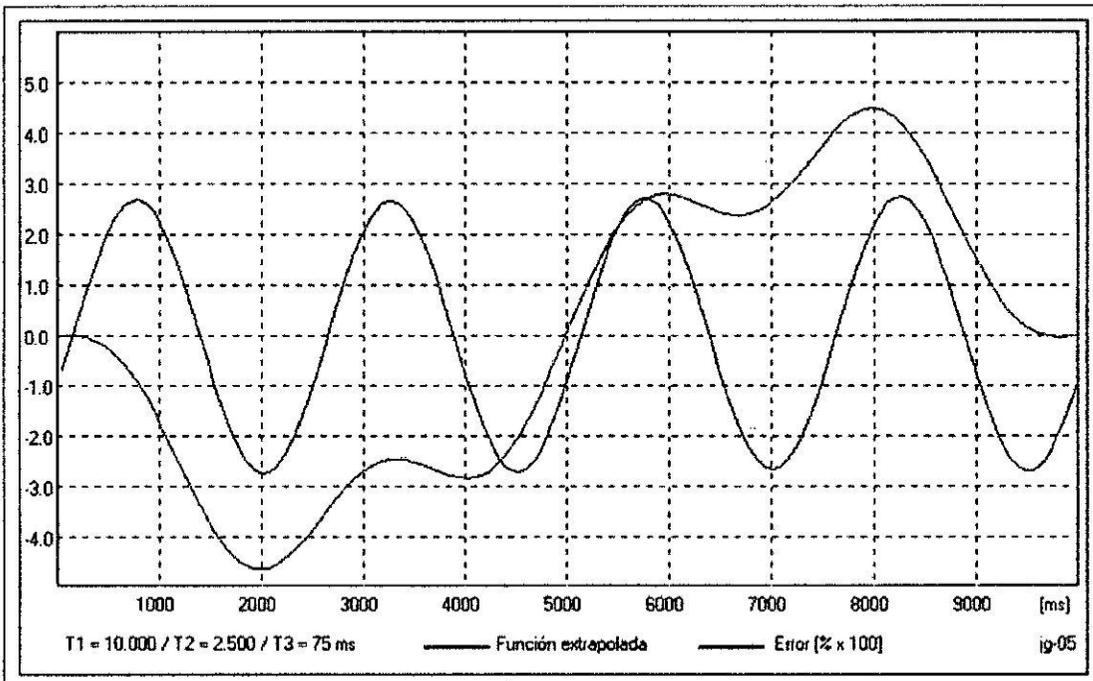
En este gráfico puede observarse que, cuando la relación entre los períodos T_1/T_2 aumenta, el error depende significativamente del intervalo de integración. No obstante, los intervalos que son propios de los procesos de tiempo real están en el orden de 1 o 2 milisegundos; por lo que los errores obtenidos con estos intervalos son satisfactorios en todos los casos. Obsérvese el comportamiento claramente lineal que manifiestan estas curvas de error para todas las relaciones T_1/T_2 .

En todas las evaluaciones realizadas se compararon los resultados obtenidos con los métodos de Runge-Kutta de 4° orden (Ec. 36 y 37), Runge-Kutta-Butcher de 5° orden (Ec.38) y el método de pasos múltiples de Milne, obteniéndose siempre resultados prácticamente idénticos. Esta circunstancia permitió prescindir del método de paso variable de Runge-Kutta_Fehlberg (Ec. 41), lo que fue muy ventajoso por las dificultades que encierra la implementación de los métodos de paso variable en los entornos de tiempo real. Finalmente, para esta

implementación se seleccionó el método de Runge-Kutta_Butcher.

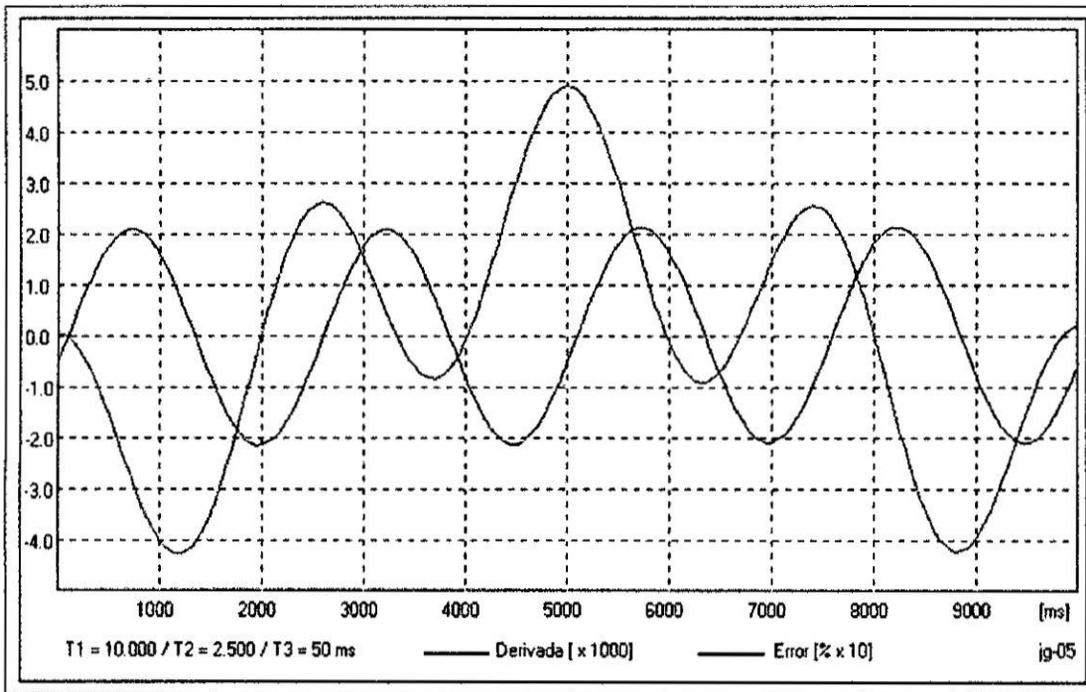
Extrapolación y derivación en puntos extrapolados

Debe recordarse que, la necesidad de conocer en un mismo instante de tiempo los valores provenientes de diferentes fuentes, como son el taquímetro y el sensor de torque, hace necesaria la extrapolación de estas funciones en el entorno del último punto conocido. Este entorno no será nunca mayor que el intervalo con que estos valores son muestreados, por lo que para la evaluación del algoritmo se toma como referencia el caso representado en la Figura 15. Aquí se utilizan cuatro puntos conocidos y el valor es extrapolado a una distancia Δt del último punto que es igual al intervalo de muestreo, lo que significa que $\delta = 1$.



Figuras N° 27 Extrapolación de un punto de la función

Esta extrapolación fue implementada por el método de los mínimos cuadrados con un polinomio de tercer grado (Ec.64); y para evaluar los resultados se utilizó la misma función que fue empleada con la integración numérica (Ec.103).



Figuras N° 28 Derivación numérica de la función extrapolada

La derivación de la función en el punto extrapolado fue también realizada por mínimos cuadrados (Ec.72) y algunos de los resultados obtenidos en intervalos de 10 segundos son representados en las Figuras 54 y 55.

La Figura 54 presenta superpuestas la función de referencia y los valores extrapolados en un intervalo de 10 segundos. Como ya ocurrió con anterioridad, la magnitud de los errores son lo suficientemente pequeños como para que no sea posible distinguir entre ambas curvas. En el mismo gráfico se presenta la curva del error relativo que fue multiplicado por cien para posibilitar su visualización. Puede observarse que los mayores errores están en la proximidad de los valores extremos de la función.

Por su parte, la Figura 55 representa las curvas de la derivada de la función y la de su error relativo, que aquí fueron respectivamente multiplicadas por mil y por diez para posibilitar su visualización. En este caso, los mayores valores de la curva del error relativo están en proximidad de los puntos de inflexión de la curva que representa la derivada.

Un aspecto muy importante que debe ser considerado es el comportamiento del error cuando aumenta el intervalo de tiempo entre los valores conocidos Δt y la distancia al valor extrapolado. Para ello, se hicieron varios estudios que dieron lugar a gráficos como el representado en la Figura 56, que corresponde al caso en que la relación entre los períodos de ambas funciones es $T_1/T_2 = 4$. Obsérvese que el mayor error en la derivada es aquí menor a 0,1% para un Δt menor a 40 ms, y que el mayor error en la evaluación de la función es aproximadamente unas cincuenta veces más chico. En un caso extremo en que el período del muestreo fuese de 100 ms (diez valores por segundo), el error relativo en la derivada sería del orden de 1,7% y de 0,085 % en la función extrapolada.

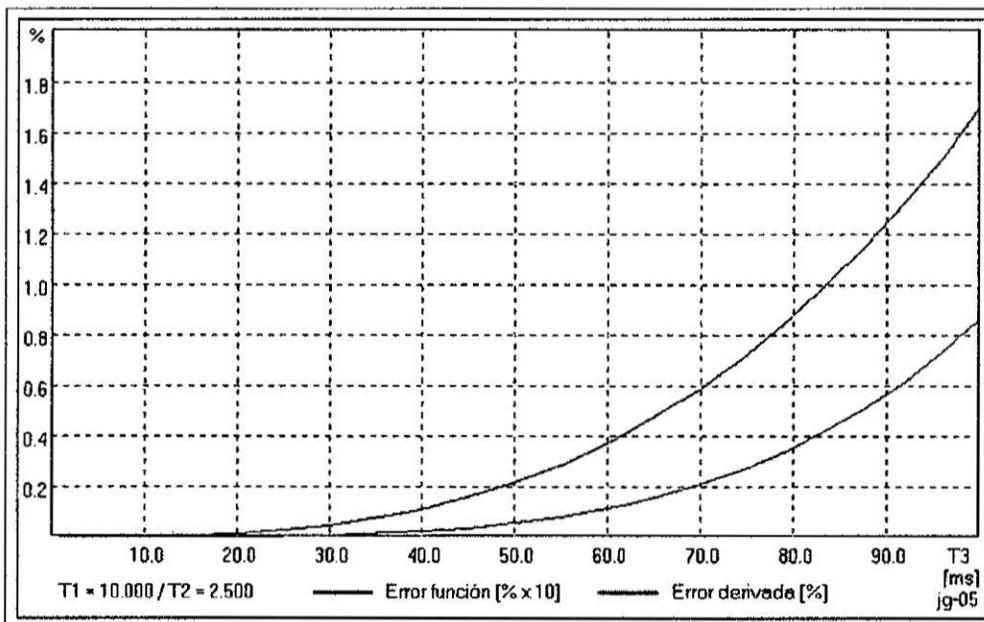


Figura N° 29 Comportamiento de los errores con diferentes intervalos entre los puntos conocidos

A partir de los estudios realizados, se concluye que la extrapolación y la derivación por el método de los mínimos cuadrados permite obtener muy buenas aproximaciones. Por ser los errores suficientemente bajos, se considera que el método es adecuado para ser implementado en este trabajo.

Aproximación de las funciones del motor y freno

Las redes neuronales artificiales serán aquí empleadas para aproximar el

comportamiento del motor y del freno dinamométrico. Para ello, en cada caso debe seleccionarse una arquitectura conveniente. A su vez, para determinar los valores de sus parámetros, debe someterse luego cada red a un proceso de entrenamiento.

Como ocurrió con los casos anteriores, se utilizan expresiones matemáticas conocidas para evaluar el comportamiento de las funciones de aproximación a ser adoptadas. En este caso, en que la arquitectura más apropiada para cada función debe obtenerse a partir de sucesivas pruebas, la utilización de funciones conocidas para el proceso de entrenamiento es altamente recomendable. Con este fin, se adoptan las funciones siguientes para representar al motor y al freno:

$$T_m(\omega, a) = (A_0 + A_1 \cdot \omega + A_2 \cdot \omega^2 + A_3 \cdot \omega^3 + A_4 \cdot \omega^4) \cdot (1 - A_m a) \quad (106)$$

$$T_r(\omega, v) = (B_0 + B_1 \cdot \omega + B_2 \cdot \omega^2 + B_3 \cdot \omega^3 + B_4 \cdot \omega^4) \cdot (1 - B_r v) \quad (107)$$

donde $A_0 = 0.4, A_1 = 1.1, A_2 = -0.8, A_3 = 0.0, A_4 = 0.0, A_m = 0.8$
 y $B_0 = 0.1, B_1 = 0.1, B_2 = 0.2, B_3 = 0.0, B_4 = 0.6, B_r = 0.5$

Por tener dos argumentos, las funciones anteriores quedan representadas por superficies sobre ejes cartesianos. Para facilitarse su visualización, en la Figura 57 son presentadas en perspectiva.

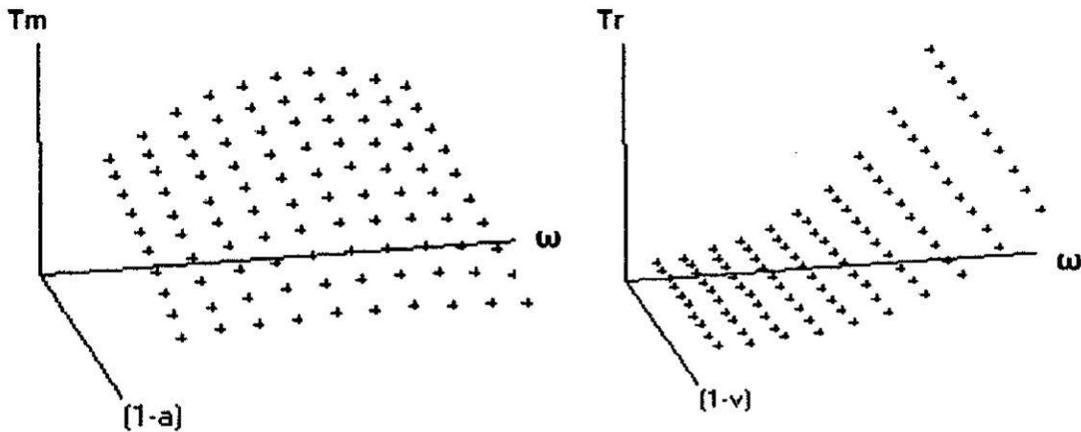


Figura N° 30 Curvas teóricas del motor y freno

El proceso de selección de la arquitectura más conveniente para cada red es laborioso, ya que pueden utilizarse varias capas ocultas y cada capa puede a su

vez contener un número de células que debe ser determinado. Para estos casos se adoptaron redes con dos capas ocultas, tales como la representada en la Figura 10, realizándose los procesos de entrenamiento en las siguientes condiciones:

- a) En razón de las características de las funciones a ser aproximadas, las redes tenían dos unidades en la capa de entrada y una en la de salida. Se trabajó con valores de entrenamiento normalizados en el rango 0.0 a 1.0.
- b) Se exploraron diversas combinaciones, con tres a quince unidades en cada capa oculta. Cada caso fue identificado por un código que incluye las unidades de cada capa separadas por guiones, tal como 2-N-M-1.
- c) El entrenamiento fue realizado con el método de backpropagation, que fue definido en el punto 2.3.3.
- d) Se adoptó la técnica denominada Adaptive Back Propagation (Ec. 63) definida en el punto 2.3.5, que ajusta el factor de aprendizaje η durante el proceso de entrenamiento. Se evitó así la necesidad de hacer pruebas con diferentes valores para encontrar el factor más conveniente.
- e) Las pruebas realizadas demostraron la conveniencia de utilizar un factor de momentum β , con valores superiores a 0,80.
- f) Todos los procesos de entrenamiento fueron realizados a partir del mismo conjunto de pesos iniciales, que fueron generados en forma aleatoria.
- g) Con las ecuaciones 106 y 107 se definieron conjuntos de pares de valores de entrenamiento y conjuntos de evaluación. Estos últimos están destinados a comprobar la capacidad de generalización de las diferentes configuraciones.
- h) Los conjuntos de entrenamiento y evaluación tuvieron entre 80 y 120 puntos, tanto en el caso del motor como del freno.
- i) Durante los procesos de entrenamiento, se evaluaron tanto los errores obtenidos con el conjunto de entrenamiento como con el conjunto de evaluación.

En la Figura 58, se representa la evolución del error durante el proceso de entrenamiento de redes con diferentes configuraciones. Los datos de entrenamiento son de la función del motor y corresponden a 100 puntos distribuidos regularmente sobre el plano X-Y. Se utilizó un coeficiente de

momentum $\beta = 0,95$ y el coeficiente de aprendizaje fue variable a partir de un valor mínimo $\eta = 0.08$.

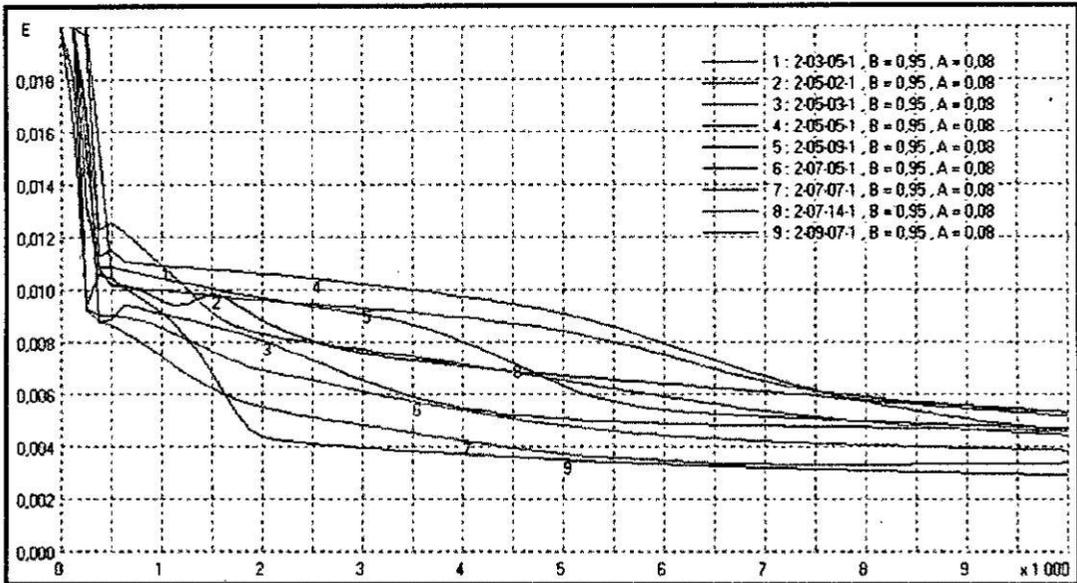


Figura N° 31 Evolución del error en el proceso de entrenamiento de redes con diferentes arquitecturas

En la Figura 59, se presenta un proceso similar, sólo que en este caso destinado a explorar el factor de momentum β más conveniente para una red en particular. Por ello, se trabaja siempre con la misma configuración y un valor de $\eta = 0.10$. Los datos de entrenamiento corresponden también a la función del motor.

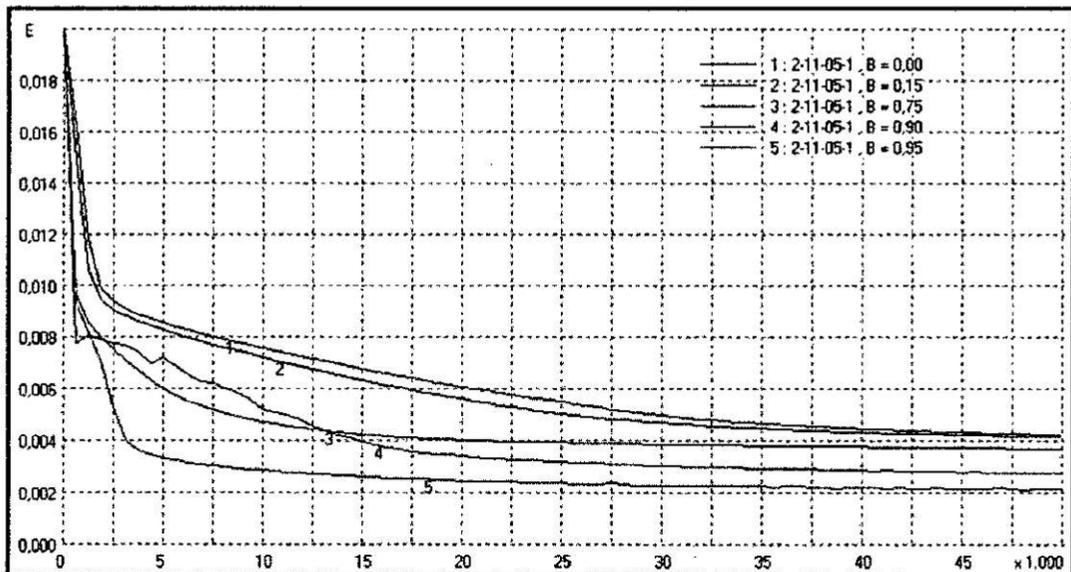


Figura N° 32 Evolución del error en el proceso de entrenamiento con diferentes valores de β

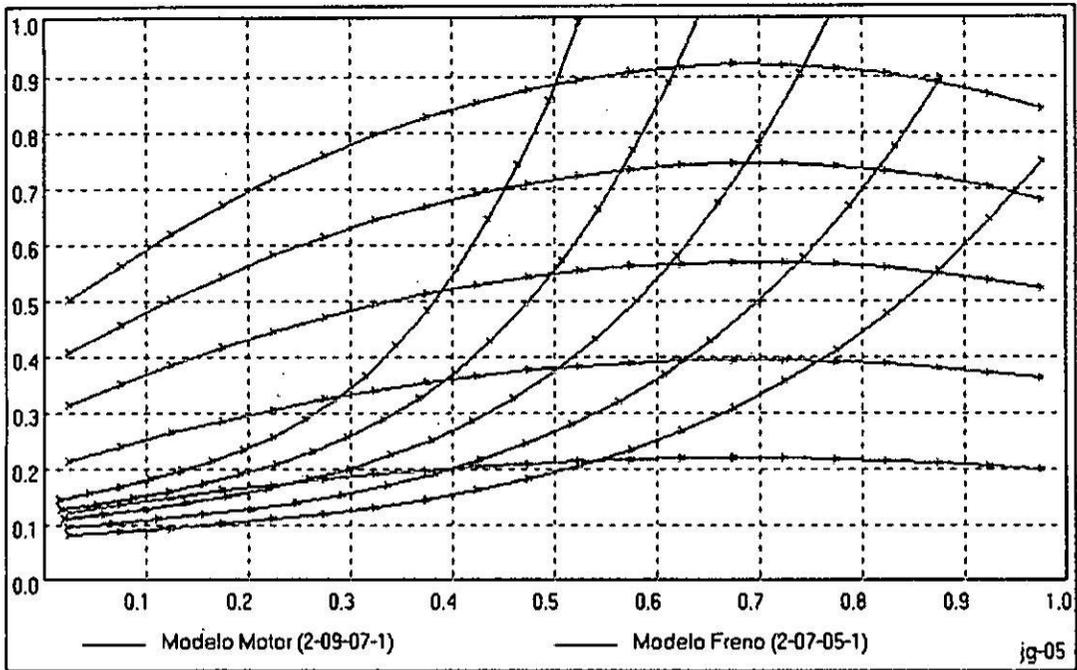


Figura N° 33 Reproducción por redes neuronales de las curvas del motor y freno

Como resultado de todas las pruebas realizadas, se concluyó que las redes más convenientes tienen las configuraciones presentadas en la siguiente tabla:

Representación	Unidades en cada capa			
	Entrada	Oculto 1	Oculto	Salida
Motor	2	9	7	1
Freno	2	7	5	1

En la Figura 60, se representan las curvas del motor y freno correspondientes a cinco condiciones de acelerador y tensión de frenado, obtenidas con estas redes. Al igual que en otros casos anteriores, la escala del gráfico no permite distinguir los puntos obtenidos con las expresiones matemáticas de aquellos calculados con los modelos neuronales.

Evaluación del comportamiento del conjunto motor-freno

Para estas primeras evaluaciones del proceso de simulación de la respuesta dinámica del conjunto Motor-Freno, se opta por un eje rígido. Por lo tanto, el modelo dinámico queda representado por la ecuación 83 del punto 4.4.3.b, que debe ser integrada en el tiempo.

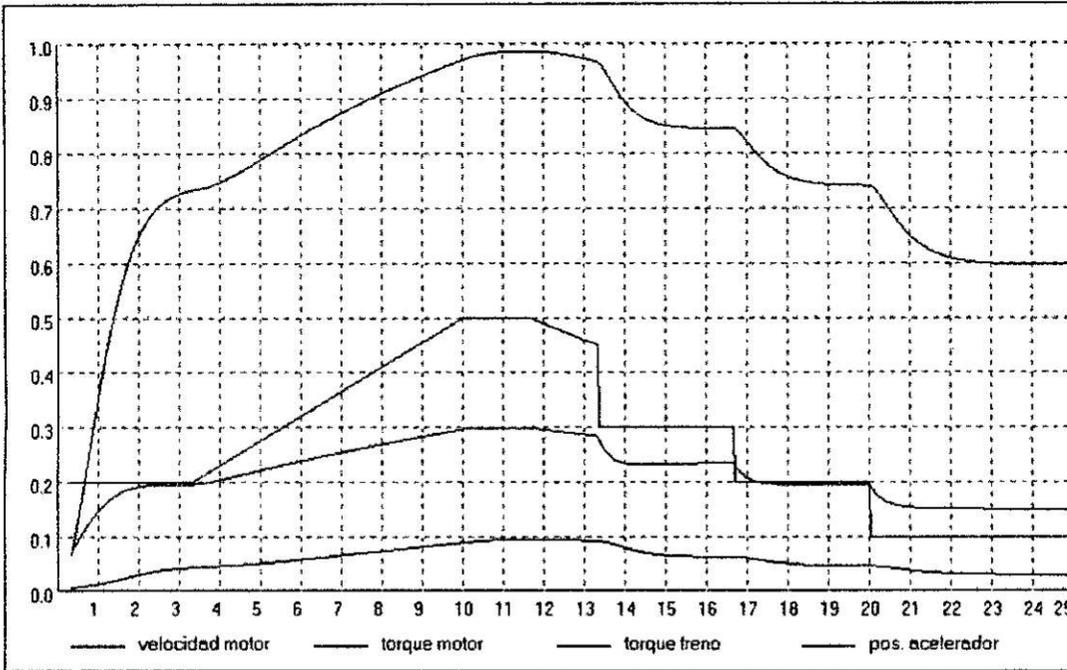


Figura N° 34 Respuesta del conjunto motor-freno a un cambio de posición del acelerador

Para estudiar el comportamiento de este proceso numérico, se hicieron numerosas pruebas, resultando todas ellas satisfactorias. En la Figura 61, se reproducen los resultados de una de estas evaluaciones, que corresponde en este caso a una aceleración que combina un escalón y una rampa para llevar el motor a su máxima velocidad. Luego, el acelerador desciende una rampa y tres escalones sucesivos, dejando el motor al 60% del valor anterior. En estos casos se trabajó con la constante $T1 = 0$ (Figura 26). En este gráfico se representan: la posición del acelerador, el torque entregado por el motor, la velocidad angular y el torque del freno, todas ellas durante un intervalo de 25 seg. Aquí son presentados todos estos valores como magnitudes adimensionales, es decir, referidos a un valor máximo unitario. De esta forma, el gráfico admite una única escala en las ordenadas que facilita su interpretación.

Evaluación del torque del motor a partir del torque del freno

Una vez que fue comprobado que el comportamiento dinámico del motor y freno es correctamente simulado, se procede a evaluar los resultados de la determinación del torque del motor a partir de los valores del torque del freno. En todos los casos

se mantiene un acoplamiento rígido vinculando ambos elementos principales.

Como se recordará, la imposibilidad de medir directamente el torque del motor hace necesaria su evaluación indirecta a partir de otros parámetros. Para ello, debe calcularse la aceleración del motor (Ec. 72), para posteriormente determinar el valor de su torque con la ecuación 89 del punto 4.4.5. Éste es un punto crítico que tendrá mucha importancia en el desempeño de la unidad de control, razón por la cual se hacen numerosas pruebas tendientes a confirmar que el error obtenido está en todos los casos dentro de un rango aceptable.

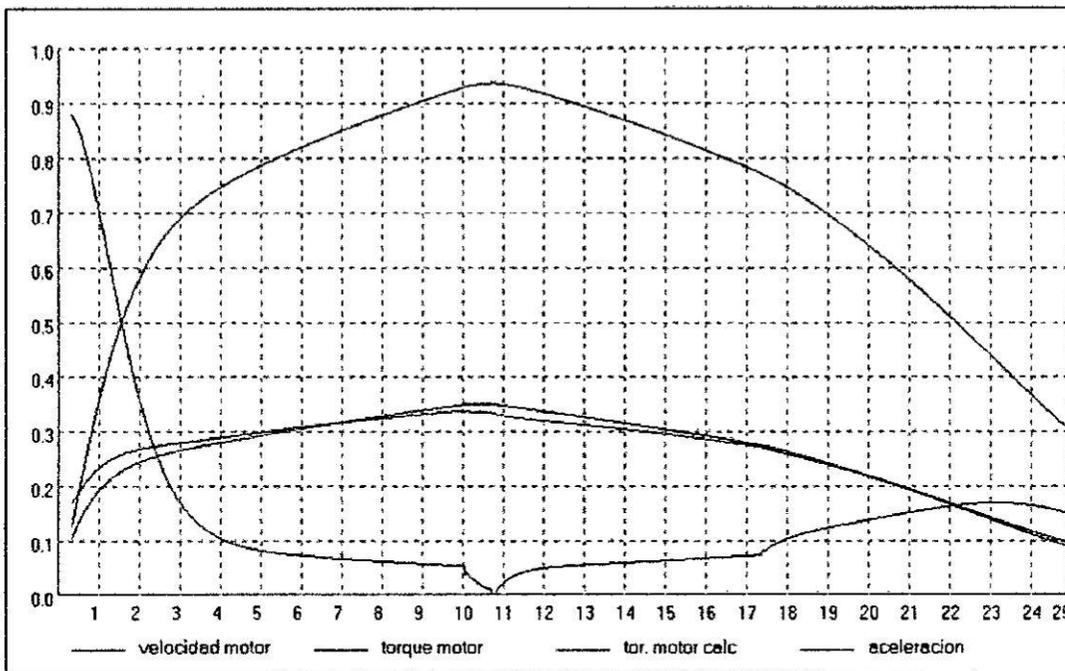


Figura N° 35 Evaluación del torque del motor a partir del torque del freno

En la Figura 62, se presentan resultados correspondientes a una curva suave de aceleración y desaceleración del motor, que se desarrolla en un intervalo de 25 seg. En el mismo gráfico se presentan las curvas de aceleración, torque real del motor y torque del motor determinado en forma indirecta. La aceleración fue calculada a partir de la velocidad, y es representada en valor absoluto, para no tener que ampliar la escala del gráfico al campo de los números negativos. Como se observa, el error obtenido en la determinación del torque es suficientemente pequeño en la zona en que se operará con la unidad de control. Aquí también, como en todos los casos, se representan magnitudes adimensionales.

Evaluación de la interacción del conjunto motor-freno y la unidad de control

El último paso en este proceso de simulación es la evaluación de la unidad de control operando sobre el conjunto compuesto por el motor y freno. Algunos de los resultados obtenidos son presentados en las Figuras 63 y 64, en las que se representan la velocidad, el torque del motor y el torque del freno. Los resultados obtenidos estuvieron, en todos los casos, dentro de los valores esperados.

En el primer ejemplo se lleva un motor al 80% de velocidad y se lo estabiliza en ese valor con una carga de aproximadamente el 60%, representado en el gráfico por el punto A. Para ello se trabaja en forma simultánea sobre el acelerador y freno, de acuerdo a lo previsto por el estado q_5 de la Tabla 11. Tal como fue establecido, la operación bajo el estado q_5 prevé una acción simultánea sobre ambas variables de control. Luego, operando en el estado q_1 se trabaja sólo con el acelerador, para reducir la velocidad y estabilizarlo al 45% de la velocidad máxima, representado por el punto B.

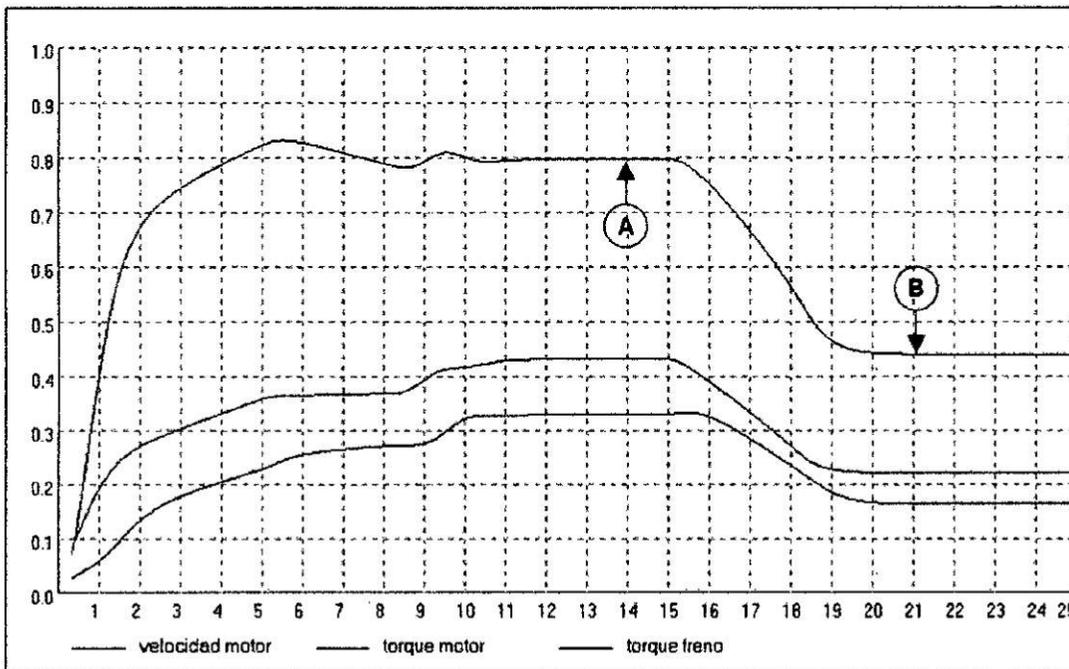


Figura N° 36 Respuesta del conjunto motor-freno ante cambios en las condiciones requeridas

En el segundo caso se lleva suavemente el motor al 85 % de la velocidad con un torque del 60%, como se representa en el punto C de la Figura 64. Aquí la unidad

de control operó alternativamente entre los estados q_2 y q_3 , es decir sobre una sola variable de control por vez. Nótese que son las mismas acciones de control que las previstas en el estado "q5", sólo que en este primer caso las acciones son simultáneas y se cumplen en paralelo. Como puede observarse, y ya fue anticipado, para pasar de una condición a otra sobre el plano de estados se dispone de numerosas variantes. Finalmente, el motor es desacelerado hasta alcanzar la condición definida por el punto D.

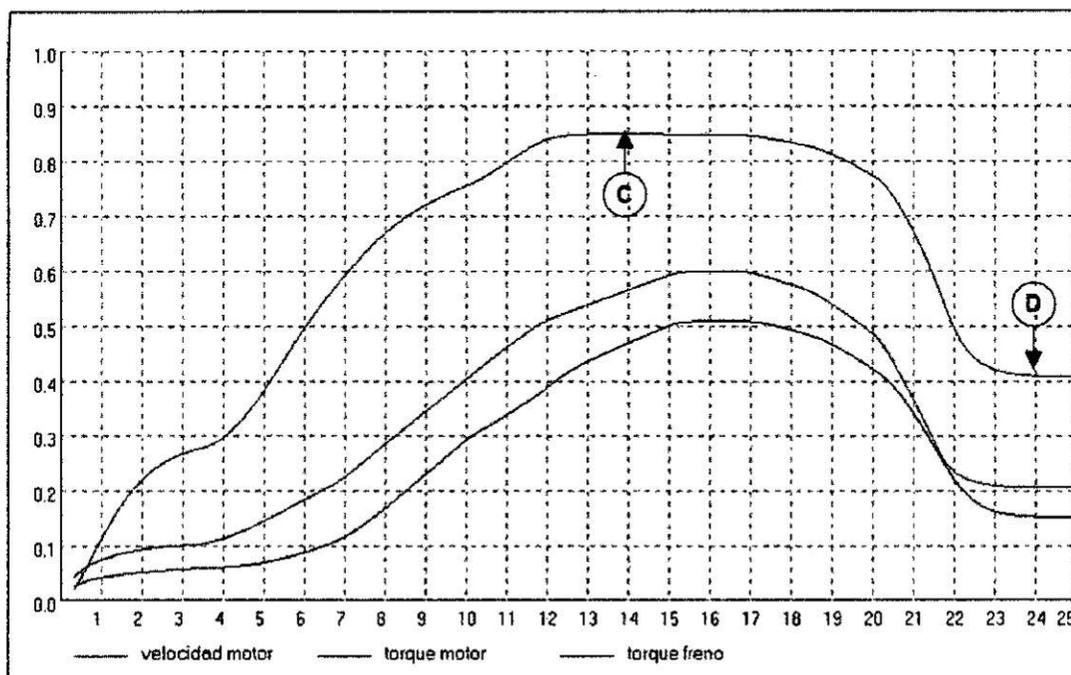


Figura N° 37 Respuesta del conjunto motor-freno ante cambios en las condiciones requeridas

En la obtención de estos resultados se consideró que tanto el motor como el freno comienzan a responder en forma inmediata, una vez alteradas sus variables de control. Esto equivale a decir que el parámetro $T1=0$ en ambos casos (Figuras 26 y 28).

7.2 Resultados finales

Las pruebas realizadas permitieron comprobar que el simulador desarrollado, reproduce correctamente el comportamiento de un conjunto motor-freno, su unidad de control y los dispositivos auxiliares de adquisición de datos y accionamiento. Para llegar a esta conclusión fueron evaluados, por separado, cada uno de los algoritmos utilizados y finalmente se hizo lo propio con el simulador en su conjunto.

En todos los casos, los resultados fueron los esperados.

Con este desarrollo se dispone de lo que podría denominarse una plataforma de simulación de ensayos de motores, capaz de reproducir los componentes principales de un sistema de ensayos y, en particular, su unidad de control.

Sin embargo, debe observarse que esta plataforma tendrá que ser configurada para simular adecuadamente el ensayo de cada motor en particular. Para ello, deben definirse los parámetros que son propios de la instalación de ensayos y los que son específicos de cada tipo de motor.

La configuración de la instalación incluye los siguientes pasos:

- a) Entrenar la red neuronal con datos característicos que serán obtenidos experimentalmente del freno dinamométrico.
- b) Conocer el momento de inercia del rotor del freno.
- c) Conocer el torque necesario para arrastrar el freno a velocidad constante, a fin de determinar las fuerzas de rozamiento estática y dinámica.
- d) Conocer el tiempo de respuesta del freno (constante T_1 , Figura 28).
- e) Conocer las frecuencias de muestreo de la velocidad y torque.

Para el caso del motor, este proceso de configuración incluirá:

- a) Entrenar la red neuronal, con datos característicos a ser obtenidos del motor a ser simulado.
- b) Conocer el momento de inercia rotatorio equivalente del motor.
- c) Conocer el torque necesario para arrastrar el motor a velocidad constante, a fin de determinar las fuerzas de rozamiento estática y dinámica.
- d) Conocer el tiempo de respuesta del acelerador y del motor respecto del acelerador (constantes T_a y T_1 , Figura 26).

Una vez ajustados los parámetros del sistema, éste podrá reproducir correctamente el ensayo de cierto motor en todas sus condiciones de operación. Luego, el esfuerzo se debe orientar a evaluar y completar la tabla de criterios de control, a través de la definición o ajuste de las reglas de producción. Se podrán

así obtener transiciones más rápidas y respuestas menos oscilantes.

Tal como puede observarse, a partir de su configuración, el simulador quedará disponible para reproducir ensayos de motores y su interacción con la unidad de control. Nótese que, para representar fenómenos vibratorios, el modelo dinámico debe ser representado por el sistema de ecuaciones diferenciales (Ec. 78 a 80), en reemplazo de la ecuación diferencial que representa al sistema rígido (Ec. 83).

CAPITULO VIII

VIII. DISCUSION DE RESULTADOS

8.1 Contrastacion de hipótesis con los resultados

Un sistema de modelado de una unidad de control para ensayos en motores de combustión está destinado respecto a los analógicos, a la introducción de variantes que posibiliten una calibración mas amigable y la optencion de aun mejores desempeños. Para ello, disponer de un modelo que permita evaluar alternativas de control resultaría de un valor incalculable

8.2 Contrastacion de hipótesis con otros estudios similares

Consideraremos los antecedentes de estudio indicados en el capitulo IV para la contrastación de los resultados obtenidos en la presente Tesis.

En el 2010, el trabajo realizado por Julio Massa y Jose Stuardi en la Universidad Nacional de Cordoba – Argentina, en el articulo denominado "Implementacion de un sistema de control adaptativo a través de una red neuronal multicapa de perceptores", en este describen el diseño que reemplazaba una unidad de control conmutado por una unidad de control adaptativo, que fue implementada sobre una red neuronal multicapa de perceptores, incorporando inteligencia computacional al proceso de selección de la estructura de control mas apropiada en cada caso.

based modeling of hybrid systems”, Computers in Industry, 36(1–2), pp. 139–146.

Teoría de control

- Voda A.; Landau I. (1995); “A Method for the Auto-calibration of PID Controllers”, Automatica, 31(1) 41-53.
- Minorsky (1922); “Directional stability of automatically steered bodies”, Journal of the American Society of Naval Engineering, Vol. 34, p. 284.
- Norman S. Nise (2010); Sistemas de Control para Ingeniería, 3ra Edición.

Métodos numéricos

- Chapra S. (2004); “Applied Numerical Methods”, McGraw-Hill.
- Rogers, D.; Adams, A. (1990); “Mathematical Elements for Computer Graphics”, McGraw Hill.
- Prentice Hall. Nakamura S (1997), Análisis Numérico y visualización gráfica con Matlab”.

CAPITULO XII

XII. ANEXOS

Anexo A: Simbología.

<p style="text-align: center;">Sistema controlado</p> <p>w : Vector de estado híbrido x : Vector de setpoints $y(t)$: Vector de estado de planta $z(t)$: Vector de señales de control A : Matriz Hurwitz V : Función de Lyapunov Q : Opciones de conmutación ξ : Función de transición e : Vector error g : Acción del sistema de control h : Acción del propio sistema $p_i(y,t)$: Superficie de deslizamiento</p> <p style="text-align: center;">Sistema conmutado de control</p> <p>S : Conjunto finito de estados Σ : Alfabeto de símbolos de entrada R : Alfabeto de símbolos de salida δ : Función de transición ϕ : Función de salida So : Estado inicial ($So \in S$) SF : Condiciones de detención ($SF \subset S$) P : Subconjuntos de S</p>	<p style="text-align: center;">Interfases</p> <p>Z^* : Conjunto de eventos reconocidos r_i : Regiones en el espacio de estados ψ : Relación eventos-alfabeto entrada u : Vector de estado modelo discreto</p> <p style="text-align: center;">Lógica de control</p> <p>G : Función de transferencia Kp : Constante proporcional Kp / Tr : Constante integral Kp / Td : Constante derivativa</p> <p style="text-align: center;">Integración de ecuaciones diferenciales</p> <p>ki : Constante método Runge-Kutta Δt : Intervalo de tiempo c : Error</p> <p style="text-align: center;">Aproximación de funciones</p> <p>a : Constantes polinomio aproximación c : Términos independientes B : Matriz de mínimos cuadrados</p>
---	--

Anexo B: Nomina de eventos.

EVENTO					ACTIVIDAD ESENCIAL	RESPUESTA	DESTINO
Nº	Denominación	Tipo	Fuente	Estímulo			
1	Taquímetro informa nueva velocidad	ext.	Taquímetro	Mensaje	Recibir y registrar nueva velocidad del motor	Mensaje con valor de velocidad	Interfase operador
2	Sensor de torque informa nuevo torque	ext.	Sensor de Torque	Mensaje	Recibir y registrar nuevo torque en el freno	Mensaje con valor de Torque	Interfase operador
3	Operador solicita parada de emergencia	ext.	Interfase operador	Mensaje	Detener el motor en forma inmediata	Mensaje de detención del motor	Motor y Freno Interfase operador
4	Operador solicita final del ensayo	ext.	Interfase operador	Mensaje	Terminar secuencia de ensayo y detener el motor	Mensajes de fin de ensayo y detención del motor	Motor y Freno Interfase operador
5	Operador altera secuencia de ensayo	ext.	Interfase operador	Mensaje	Alterar la secuencia de ensayo	Mensaje con nuevo paso de ensayo activo	Interfase operador

6	<u>Seguros</u> ordenan detención del <u>motor</u>	ext.	<u>Seguros</u>	Mensaje	Detener el <u>motor</u> en forma inmediata	Mensaje de detención del <u>motor</u>	<u>Motor y</u> <u>Freno</u> <u>Interfase</u> <u>operador</u>
7	Transición a nuevo <u>paso de</u> <u>ensayo</u>	Tpo	<u>Unidad</u> <u>de</u> <u>Control</u>	Intervalo de Tiempo	Cumplir las previsiones del <u>programa de</u> <u>ensayo</u>	Mensaje con nuevo <u>paso</u> <u>de ensayo</u> <u>activo</u>	<u>Interfase</u> <u>operador</u>
8	Ajuste regular de <u>variables de</u> <u>control</u>	Tpo	<u>Unidad</u> <u>de</u> <u>Control</u>	Intervalo de Tiempo	Asegurar una condición estable de operación	Señales de control de <u>motor y freno</u>	<u>Motor y</u> <u>Freno</u>
9	Conmutación de <u>lógica activa</u> de control	cond. esp.	<u>Unidad</u> <u>de</u> <u>Control</u>	Condición de <u>vector</u> <u>de estado</u>	Definir la <u>lógica activa</u> de control más apropiada	---	---

GUIA DE ENCUESTA

La técnica utilizada que se realizó a técnicos de diferentes empresas tiene por finalidad recolectar información importante para el estudio de **"MODELAMIENTO DE UNA UNIDAD DE CONTROL PARA ENSAYOS DE MOTORES DE COMBUSTION EN TIEMPO REAL"**.

Se le solicita a Ud. que, con relación a las preguntas formuladas que a continuación se le presentara, se sirva a responder en términos claros, para la investigación que se viene llevando a cabo. Esta encuesta es anónima así que se le agradecerá por la participación

1. ¿Conoce Ud. lo que es una unidad controlada para ensayos de motores de combustión?

Si ()

No ()

No sabe ()

2. ¿Ud. Cree que sería más productivo si todos los ensayos de motores de combustión estuvieran automatizados?

Si ()

No ()

No sabe ()

3. ¿Ha visto ensayos para motores controlados de forma analógica (son manejados manualmente en todo el procedimiento)?

Si ()

No ()

No sabe ()

4. ¿Qué tan importante cree que es la automatización del desarrollo de ensayos de motores de combustión interna en las industrias?

Muy importante ()

Importante ()

Poco importante ()

5. ¿Tiene Ud. conocimientos técnicos sobre el manejo de ensayos para motores de combustión?

Si ()

No ()

No sabe ()

6. ¿Dispone de software o sistemas que faciliten el uso del ensayo para motores de combustión?

Si ()

No ()

No sabe ()

7. ¿Cree usted que la gestión administrativa es importante para el desarrollo de un modelamiento de control para ensayos de motores en tiempo real?

Si ()

No ()

No sabe ()

8. ¿Considera necesario la capacitación constante para el manejo para ensayos de motores de combustión interna?

Si ()

No ()

No sabe ()

9. ¿Quisiera que se implemente una unidad controlada para ensayo de motores de combustión interna en tiempo real en su entidad laboral?

Si ()

No ()

No sabe ()

10. ¿Cree usted que la implementación de una unidad controlada de ensayo para motores de combustión interna ayudaría en el manejo de diversos problemas?

Si ()

No ()

No sabe ()

11. ¿Cree usted que es rentable la implementación de una unidad controlada para ensayos de motores de combustión interna en tiempo real?

Alta rentabilidad ()

Mediana rentabilidad ()

Baja rentabilidad ()

12. ¿Ha recibido usted cursos de capacitación sobre las unidades controladas para ensayos de motores de combustión interna?

Si ()

No ()

No sabe ()

13. ¿Qué tan importante es la mano de obra calificada en el manejo de ensayos para motores de combustión interna?

Muy importante ()

Importante ()

Poco importante ()

14. ¿Cree que el modelamiento de una unidad controlada para motores de ensayos de combustión interna en tiempo ha despertado interés en usted?

Si ()

No ()

No sabe ()

15. ¿Cree usted que la entidad para quien labora estaría interesada en una unidad controlado para ensayos de motores?

Si ()

No ()

No sabe ()

16. ¿Cree usted que el desarrollo de una unidad controlada para ensayos de motores, realizaría cambios en el área de su trabajo?

Si ()

No ()

No sabe ()

Anexo D: MATRIZ DE CONSISTENCIA.

"MODELAMIENTO DE UNA UNIDAD DE CONTROL PARA ENSAYOS DE MOTORES DE COMBUSTION EN TIEMPO REAL".

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES	TIPO DE INVESTIGACIÓN
<p>Problema principal</p> <p>¿Qué consideraciones se deben tomar para el desarrollo de un modelamiento de una unidad de control para ensayos de motores de combustión en tiempo real?</p> <p>Problemas secundarios</p> <p>a.- ¿De qué manera un modelamiento de una unidad de control para ensayos de motor de combustión en tiempo real</p>	<p>Objetivo General</p> <p>Desarrollar, implementar y evaluar un simulador de tiempo real de una unidad de control destinada a conducir ensayos de motores de combustión interna en forma completamente automática.</p> <p>Objetivos secundarios</p> <p>a.- Comprobar con un</p>	<p>Hipótesis principal</p> <p>Un sistema modelado de una unidad de control para ensayos en motores de combustión está destinados a la realización de pruebas en forma completamente automática, incluyendo la adquisición de datos, monitoreo local y remoto, presentación de valores en pantalla, y migración de resultados de ensayos</p>	<p>Variable Independiente:</p> <p>X: Modelamiento de una unidad de control.</p> <p>Indicadores:</p> <p>x₁ = Taquímetro x₂ = Sensor de torque x₃ = Seguros x₄=Variables de control</p> <p>Variable Dependiente:</p> <p>Y: Ensayos en motores</p>	<p>1. Tipo de Investigación</p> <p>Aplicada</p> <p>2. Nivel de investigación</p> <ul style="list-style-type: none"> ○ Descriptivo ○ Explicativo <p>3. Diseño</p> <p>Por objetivos</p> <p>4. Población</p>

	real.	b. La importancia del desarrollo del desarrollo de un modelamiento de una unidad de control para ensayos en motores en el área tanto de la mecánica como el área de control estimula a los estudiantes a buscar mejores desarrollos y aplicaciones para dicho modelo.		o Ficha de encuesta.
--	-------	---	--	----------------------

