



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO.**

---

---

**FACULTAD DE INGENIERÍA.**

**SIMULADOR DEL CONDENSADOR, SISTEMA DE AGUA  
DE ALIMENTACIÓN Y CONTROLADOR DE INVENTARIO  
DE REFRIGERANTE DEL REACTOR, DE UNA  
NUCLEOLÉCTRICA TIPO BWR E INTERFAZ GRÁFICA  
DE USUARIO PARA EL SUN-RAH.**

**T E S I S**

**QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO EN COMPUTACIÓN  
P R E S E N T A:**

**ALITET SÁNCHEZ BELLO.**



**Asesor: Dr. Jaime B. Morales Sandoval.**

**México D.F.  
2004**

## ÍNDICE.

<b>1. Introducción.....</b>	<b>1</b>
1.1. Antecedentes energía nuclear (Marco histórico, entrenamiento) .....	1
1.2. Conceptos generales.....	3
1.3. Plantas nucleares.....	4
1.3.1.Reactores de agua ligera a presión (PWR).....	5
1.3.2.Reactores de agua ligera hirviendo (BWR).....	5
1.3.3.Reactores de agua pesada a presión (PGR ó CANDU).....	5
1.3.4.Reactores refrigerados por bióxido de carbono y moderados por grafito (GCR).....	6
1.3.5.Reactores de cría enfriados por sodio (LMFBR).....	6
1.3.6.Reactores de propulsión.....	7
1.4. Relación con plantas termoeléctricas.....	8
1.5. Estado actual.....	9
1.5.1.Panorama actual de la energía eléctrica.....	9
1.5.2.Panorama actual en simuladores de plantas nucleoelectricas.....	11
1.5.2.1. Tipos de simuladores.....	11
1.5.2.2. Simuladores existentes.....	12
1.6. Planeamiento general del problema a resolver.....	14
1.7. Solución esperada.....	16
Referencias Bibliográficas.....	17
<b>2. Análisis del sistema.....</b>	<b>19</b>
2.1. Identificación de las necesidades.....	19
2.2. Evaluación del concepto de sistema para establecer la viabilidad.....	20
2.3. Jerarquización.....	21
2.4. Ingeniería de la Información.....	22
2.5. Ingeniería del proyecto.....	23
Referencias Bibliográficas.....	24
<b>3. Diseño del simulador del sistema de agua de alimentación.....</b>	<b>25</b>
3.1. Introducción.....	25
3.2. Justificación termodinámica de los componentes.....	25
3.3. Métodos de solución.....	27
3.4. Diagrama del proceso.....	28

3.5. Especificaciones.....	28
3.6. Estructura del simulador de condensado, agua de alimentación y controlador asociado...	32
3.7. Desarrollo del modelo y controladores.....	33
3.7.1.El Condensador.....	33
3.7.2.Calentadores de baja presión.....	34
3.7.3.Calentadores de alta presión.....	34
3.7.4.Turbo-bombas de agua de alimentación.....	34
3.7.5.Controlador del sistema de agua de alimentación.....	34
3.8. Especificaciones de la integración del modelado con la interfase grafica para el condensador, sistema de agua de alimentación y controlador.....	36
Referencias Bibliográficas.....	37

#### **4. Modelado matemático del sistema de agua de alimentación**

4.1. Principios termodinámicos transferencia de calor.....	38
4.1.1.Intercambiadores de calor de agua de alimentación cerrados.....	38
4.2. Ecuaciones y Justificaciones.....	39
4.2.1.Calentadores de Agua de Alimentación.....	41
4.2.2.Condensador.....	47
4.2.3.Flujo de Agua de Alimentación.....	48
4.3. Preparación de las ecuaciones para el modelado.....	49
4.3.1.Calentador 1.....	49
4.3.2.Calentador 2.....	51
4.3.3.Calentador 3.....	52
4.3.4.Calentador 4.....	54
4.3.5.Calentador 5.....	56
4.3.6.Calentador 6.....	58
4.3.7.Condensador.....	59
4.3.8.Bomba de Agua de Alimentación.....	60
4.4. Condiciones de frontera, variables de estado, cantidades medibles.....	61
4.4.1.VARIABLES DE ESTADO.....	61
4.4.2.Estados Iniciales.....	62
4.5. Entradas.....	62
4.6. Parámetros.....	63
4.7. Conclusiones.....	65
Referencias Bibliográficas.....	65

<b>5. Interfaz Gráfica</b> .....	66
5.1. Introducción.....	66
5.2. Interfaz de programación: DirectX.....	66
5.2.1.Requerimientos.....	67
5.2.2.Bases de DirectX.....	68
5.3. Integración entre Matlab y Visual C++.....	69
5.3.1.Funciones de integración con Matlab.....	70
5.4. Paradigma de programación.....	73
5.5. Especificaciones.....	75
5.5.1.Ciclo de ejecución principal.....	76
5.6. Alcance.....	77
5.7. Desarrollo.....	77
5.7.1.Características proporcionadas por librerías DirectX.....	78
5.7.2.Características proporcionadas por terceros.....	79
5.7.3.Características propias desarrolladas específicamente para el simulador.....	80
5.7.4.Proceso específico de simulación de un modelo en la interfaz gráfica.....	82
5.7.5.Implementación específica: Sistema de Condensador Agua de Alimentación y controlador asociado.....	84
5.8. Despliegues.....	85
5.8.1.Galería de elementos.....	85
5.9. Interacción entre despliegues.....	87
5.10. Menús.....	88
5.11. Conclusiones.....	90
Referencias bibliográficas.....	91
<b>6. Pruebas de los modelos y la interfaz</b> .....	92
6.1. Diseño de pruebas a modelos.....	92
6.2. Resultados de pruebas a modelos individuales.....	93
6.2.1.Condensador.....	93
6.2.2.Calentador de baja y alta presión.....	94
6.2.3.Controlador de inventario de refrigerante.....	96
6.2.4.Bomba de agua de alimentación.....	97
6.3. Pruebas a los modelos integrados.....	98

6.3.1. Calentadores de agua de alimentación dentro del sistema completo de agua de alimentación.....	98
6.3.2. Controlador de agua de alimentación dentro del sistema completo de agua de alimentación.....	103
6.4. Resultados de pruebas de integración entre los modelos y la interfaz.....	105
6.5. Conclusiones.....	108
<b>7. Conclusiones.....</b>	<b>109</b>

## Apéndices

- A. Tablas de Vapor.....	111
- B. Implementación en Matlab.....	122
- C. Pruebas integradas del Sistema de Condensado, Agua de alimentación, Controlador de Inventario de Refrigerante y del Sistema de Turbina, Línea de Vapor, Recalentador y Generador.....	126
- D. Factores de Conversión.....	130
- E. Tablas de Parámetros e Implementación de los dispositivos modelados.....	131

## **Lista de tablas.**

<b>Tabla 1.1:</b> Diferentes tipos de reactores térmicos.....	7
<b>Tabla 1.2:</b> Panorama del empleo de la energía nuclear en el mundo.....	10
<b>Tabla 3.1:</b> Configuración de la computadora en la que se realizó el desarrollo de la interfaz grafica .....	30
<b>Tabla 6.1:</b> Valores de entrada para esquema de prueba.....	99
<b>Tabla 6.2:</b> Resultado comparativo de los valores de entalpía obtenidos a las pruebas de los modelos.....	100
<b>Tabla 6.3:</b> Resultado comparativo de los valores de flujo (lado carcaza) obtenidos a las pruebas de los modelos.....	100

## Lista de figuras.

<b>Figura 1.1:</b> Esquema de una termoeléctrica.....	8
<b>Figura 1.2:</b> Pantalla del simulador de entrenamiento Isis™.....	13
<b>Figura 1.3:</b> Pantalla del simulador de cinética del reactor Comet™.....	13
<b>Figura 1.4:</b> Simulador para entrenamiento de operadores [OTS] de la empresa Invensys™.....	14
<b>Figura 1.5:</b> Simulador de principios básicos de la Central Nuclear Embalse.....	15
<b>Figura 1.6:</b> Flash Nuclear Simulator, simulador escrito en flash con código en Basic.....	15
<b>Figura 1.7:</b> “Control the Nuclear Power Plant”, escrito en Java, con estados predeterminados.....	15
<b>Figura 2.1:</b> Esquema jerárquico de las partes del simulador.....	21
<b>Figura 2.2:</b> Plantilla de arquitectura para el proyecto.....	24
<b>Figura 3.1.</b> Ciclo Ranking ideal y ciclo regenerativo con calentador de agua de alimentación abierto.....	26
<b>Figura 3.2.</b> Comparación entre los dos ciclos.....	27
<b>Figura 3.3:</b> Diagrama para el proceso de modelado y pruebas.....	29
<b>Figura 3.4:</b> Capas componentes del simulador y situación de los modelos dentro de la estructura de simulación.....	32
<b>Figura 4.1:</b> Esquema de un calentador cerrado.....	39
<b>Figura 4.2:</b> Diagrama de localización de las variables en el simulador. Las variables de estado se identifican con un asterisco.....	40
<b>Figura 4.3:</b> Esquema en Simulink de las entradas definidas para la simulación independiente del sistema de agua de alimentación.....	63
<b>Figura 5.1:</b> La arquitectura de DirectX y su relación con Win32.....	68
<b>Figura 5.2:</b> Ejemplificación de los comandos empleados para establecer la comunicación entre Matlab y el programa en “C”.....	74
<b>Figura 5.3:</b> Clases empleadas en el simulador del sistema de agua de alimentación.....	75
<b>Figura 5.4:</b> La mecánica del ciclo de procesamiento de mensajes de Windows, en la ejecución de la interfaz gráfica.....	76
<b>Figura 5.5:</b> Lista doblemente ligada.....	81
<b>Figura 5.6:</b> Graficas tipo X-Y para visualización de variables en el simulador.....	81
<b>Figura 5.7:</b> Tiempo de simulación inicial (la interfaz gráfica lo modifica cada ciclo) y método de integración adecuado para el modelo.....	83
<b>Figura 5.8:</b> a) Salidas en el esquema de Simulink b) Representación de las salidas en el Workspace de Matlab c) Configuración en los parámetros de la simulación.....	84
<b>Figura 5.9:</b> Galería de elementos.....	85
<b>Figura 5.10:</b> Diagrama jerárquico de la organización de las pantallas de interfaz en el simulador SUN-RAH.....	87

<b>Figura 5.11:</b> Organigrama de las pantallas del simulador de agua de alimentación y sus ramificaciones dependiendo de la selección del usuario.....	88
<b>Figura 5.12:</b> Pantallas de ejemplo en el uso del menú gráfico.....	89
<b>Figura 6.1:</b> Esquema de prueba para el condensador.....	93
<b>Figura 6.2:</b> Señales significativas de entrada al sistema de condensado y su respectiva salida.....	93
<b>Figura 6.3:</b> Esquema de prueba para un calentador de agua de alimentación.....	94
<b>Figura 6.4:</b> Entradas para la prueba de un calentador cerrado de agua de alimentación.....	95
<b>Figura 6.5:</b> Salidas para la prueba de un calentador cerrado.....	95
<b>Figura 6.6:</b> Correlación entre la entrada y la salidas en la prueba de el calentador cerrado de agua de alimentación.....	96
<b>Figura 6.8:</b> Señales de entrada y salida del controlador de inventario de refrigerante. También se observa la grafica del nivel en el reactor.....	97
<b>Figura 6.9:</b> Modelo de la bomba de agua de alimentación.....	97
<b>Figura 6.10:</b> Comparación entre la entrada y la salida en el modelo de la bomba de agua de alimentación.....	98
<b>Figura 6.11:</b> Esquema de pruebas de los calentadores, el controlador, la bomba, y nivel de agua en el reactor.....	98
<b>Figura 6.12:</b> Parámetros de la simulación de prueba de los modelos de agua de alimentación.....	99
<b>Figura 6.13:</b> Toma de valores en los modelos de Matlab, en el proceso de pruebas a los modelos integrados.....	99
<b>Figura 6.14:</b> Grafica de la salida de la variable entalpía en los calentadores de baja presión.....	101
<b>Figura 6.15:</b> Gráficas de las salidas de los calentadores de baja presión.....	101
<b>Figura 6.16:</b> Parámetros para la simulación.....	102
<b>Figura 6.17:</b> Esquema de la integración entre los calentadores de baja y de alta presión.....	102
<b>Figura 6.18:</b> Entalpía y flujo comparativos del calentador de alta presión.....	102
<b>Figura 6.19:</b> Configuración de prueba completa del sistema de agua de alimentación.....	103
<b>Figura 6.20:</b> entradas para el modelo de pruebas.....	104
<b>Figura 6.21:</b> a) $H_{cM}$ : Nivel de agua en el reactor      b) <i>ajuste</i> : Variable del controlador.....	104
<b>Figura 6.22:</b> a) $m_{fw}$ : Flujo de agua de alimentación a alterar por la bomba de agua de alimentación. b) $W_{fw}$ : Flujo de agua de alimentación que se envía a los calentadores.....	104
<b>Figura 6.23:</b> $h_{fw6}$ : Entalpía del flujo de agua de alimentación al final de el tren de calentadores..	105
<b>Figura 6.24:</b> Entrada escalón para la prueba de correlación entre Simulink y la interfaz gráfica...	106
<b>Figura 6.25:</b> a) Gráfica de la entalpía del flujo de agua $h_{fw6}$ . b) Gráfica de la entalpía en la interfaz gráfica.....	106
<b>Figura 6.26:</b> Prueba de correlación entre la simulación en Simulink y la Interfaz, que muestra la alteración de las entalpías en los calentadores. ....	106

**Figura 6.27:** Uso de las perillas en el panel de control para alterar los parámetros de la simulación en tiempo de ejecución.....107

**Figura 6.28:** Panel de control de la simulación del sistema de agua de alimentación.....107

# Capítulo 1. Introducción.

## 1.1. Antecedentes energía nuclear.

A continuación se presenta un panorama referente a la generación de energía y a los simuladores de reactores nucleares que se han desarrollado en los últimos años.

El marco histórico de la energía nuclear, contiene varias alusiones a temas relativos al entrenamiento y desarrollo de simuladores, con el afán de capacitar a los operadores y proveer una mayor seguridad en caso de alguna eventualidad, o de previsión y control de eventos indeseables.

En la historia nuclear de varios países del mundo, se observa el interés por fomentar cada vez en mayor grado el entrenamiento de excelencia a operadores de instalaciones nucleares. Como ejemplo podemos mencionar el desarrollo del entrenamiento de operadores y de los simuladores de centrales nucleoelectricas.

En 1947 En la universidad de Detroit, el Dr. Enrico Fermi controló por primera vez la reacción nuclear en cadena.

En 1954 comenzó la operación de la primera planta de uso comercial en Obninsk, Rusia, aunque este evento se difundió en forma masiva.

En 1957 comenzó la operación de la primera planta comercial de Estados Unidos en Shippingport Pennsylvania.

En 1979 Ocurrió el mayor accidente de una planta nuclear en Estados Unidos, en la planta de Three Mile Island, debido a una conjunción de fallas de los equipos, equipo mal diseñado y ineficiencia y desconocimiento de los operadores del estado del reactor.

Ese mismo año la comisión investigadora Kennedy, instó la creación del “Institute of Nuclear Power Operations (INPO)”, este instituto generó reglamentos, objetivos y criterios para la operación de plantas nucleares y conduce evaluaciones a las plantas nucleares desde 1981.

En 1985 se creó en E.U. la “National Academy for Nuclear Training”, una Academia encargada de proveer y supervisar los programas educativos en materia nuclear que se proveían en los centros educativos.

En 1994 la “Nuclear Regulatory Commission (NRC)”, especificó que los operadores de plantas nucleares deberían ser recapitados cada seis años, por las empresas que los contratan.

A su vez, los programas con los que se entrena al personal, es revisado cada cuatro años por el “National Nuclear Accrediting Board”.

Las compañías nucleares, invierten miles de horas de trabajo y cientos de millones de dólares en entrenamiento. En 1979 las compañías en los Estados Unidos tenían únicamente 12 simuladores del centro de control de una planta. Actualmente hay 74 en uso. Desde 1979 el número de personal entrenado profesionalmente ha crecido de 440 a casi 5000 operadores, así como el espacio destinado a entrenamiento que se ha octuplicado.

Existen ahora nuevos niveles de seguridad y eficiencia y el número de paradas de emergencia ha decrecido a casi cero desde 1993.

El empleo de el entrenamiento para mejorar la eficiencia de un planta nuclear se evidenció en el año 2000 cuando la Comisión de Regulación Nuclear norteamericana renovó el permiso de operación de la planta "Cliffs Nuclear Power Plant" por otros veinte años, extendiendo sus beneficios económicos. [1]

En México los simuladores de este tipo fueron desarrollados en su mayor parte por el Instituto de Investigaciones Eléctricas.

A finales de los setentas cuando la CFE buscaba cómo mejorar la capacitación en centrales térmicas, ya que se iniciaba la construcción de las de 350 MW de capacidad y se construía la nucleoelectrica de Laguna Verde, se presentaron muchas ofertas de simuladores extranjeros. En el instituto se fabricó un simulador que reproduce la operación de la unidad 1 de la central termoeléctrica de Tula.

El doctor José Miguel González Santaló, fue el jefe del proyecto que logró fabricar un simulador que permitía el entrenamiento de los operadores, contar con tecnología nacional en el campo de los simuladores y promover la investigación en el país.

El departamento de simulación del instituto (IIE) [2] actualmente ha generado siete simuladores, que se encuentran actualmente en operación: el Termos I, el simulador de la nucleoelectrica de Laguna Verde; el simulador de los operadores de línea del Sistema de Transporte Colectivo Metro; el simulador de rodado de turbina; el emulador de máquinas síncronas; el simulador de operadores de centrales geotermoeléctricas y el simulador de operadores de centrales de ciclo combinado (concluido en mayo del 2000), y actualmente se encuentra en construcción el simulador Termos II.

Solamente en el área termoeléctrica se estima que los simuladores han proveído entrenamiento aproximadamente a de 7580 operadores. Asimismo, se calcula que el beneficio económico del simulador Termos I corresponde en promedio a 4500 dólares por año por MW de capacidad. Esta cifra equivale aproximadamente a cerca de 850 millones de dólares en los 18 años de operación de este simulador. Actualmente la lista de espera de este centro de adiestramiento abarca hasta el año 2002 y en ella esperan operadores de turno de México y Latinoamérica. [2]

## 1.2. Conceptos Generales.

La definición de simulación es difícil debido a los diferentes conceptos que las personas dedicadas a este campo consideran como su área de trabajo, las definiciones pueden ir desde: “Es el modelado y análisis de un sistema” hasta “Es el estudio de sistemas dependientes del tiempo que involucran un cambio de estado o algún evento”. [3]

Frecuentemente es necesario simular, es decir, reproducir artificialmente un fenómeno o las relaciones entrada-salida de un sistema. Esto ocurre siempre que la operación de un sistema o la experimentación en él son imposibles, costosas, peligrosas o poco prácticas, como en el entrenamiento de personal de operación, pilotos de aviones, etc. Si esta reproducción está basada en la ejecución de un programa en una computadora digital, entonces la simulación se llama digital y usualmente se conoce como simulación por computadora.

Rosenbluth y Wiener señalaron en 1945 que los modelos constituyen una necesidad central del procedimiento científico. La aparición de las computadoras abrió nuevas posibilidades para el análisis de modelos que podían ser procesados con más rapidez y exactitud. Ziegler en su libro dedicado a los aspectos teóricos de la simulación [4], dice: “La frase modelado y simulación designa el complejo de actividades relacionadas con la construcción de modelos de los sistemas del mundo real y su simulación en computadora. En particular, ‘modelado’ trata en principio, sobre las relaciones entre sistemas reales y modelos; ‘simulación’ se refiere principalmente a las relaciones entre las computadoras y los modelos.”

Gordon [5] dice que “podemos entonces definir la simulación de sistemas como una técnica de solución de problemas a través de la observación del comportamiento de un modelo dinámico del sistema en el tiempo”, Pritsker [6], involucra el concepto de estado del sistema y trata su dinámica como “el movimiento de un estado a otro según las reglas de operación definidas”.

La implementación de las simulaciones ha sufrido varias alteraciones y cambios de paradigma, conforme las capacidades de los equipos de cómputo se van incrementando y su costo se ha ido reduciendo.

En los años sesenta aparecieron nuevos conceptos provocados por el rápido incremento de la capacidad de los sistemas de cómputo. La idea fundamental de los nuevos lenguajes y entornos de simulación consiste en la creación de objetos en la memoria operativa; cada objeto contiene sus datos (atributos, parámetros formales, constantes y variables locales) y su lista de instrucciones, además puede funcionar en forma independiente, ejecutando sus instrucciones. Otra idea importante de este tipo de programación se refiere al concepto de herencia de propiedades y a las clases de objetos. Esto permite construir una nueva clase de objetos que obtienen todas las propiedades de otra clase definida anteriormente.

La simulación por computadora no es un campo cerrado y hay que interpretarla como una herramienta interdisciplinaria, por lo que el conocimiento de las áreas de la ciencia e ingeniería ayudan a crear simuladores más sólidos y eficientes.[7]

### **1.3. Plantas Nucleares**

Los reactores se pueden clasificar de diferentes maneras. Si el material fisionable se encuentra en barras sumergidas en el moderador, el reactor se llama heterogéneo.

El reactor es homogéneo si el moderador y el material fisionable son mezclados, por ejemplo, el uranio o alguna sal del elemento fisionable se encuentra disuelta en agua corriente, agua pesada, o algún otro moderador.

Los reactores se clasifican también de acuerdo con la energía de los neutrones en reactores rápidos, intermedios o térmicos. A su vez, se clasifican de acuerdo con sus objetivos como: reactores de investigación, reactores de potencia (que producen calor para generar electricidad), reactores productores de plutonio, y en reactores de tipo mixto que sirven para varios fines a la vez.

En el reactor de potencia, la energía de la fisión nuclear la absorbe un medio circulante de enfriamiento a través del reactor. El refrigerante caliente, se ocupa para calentar agua que luego genera vapor. Este último hace girar una turbina que a su vez, impulsa al generador que produce electricidad. El vapor se condensa y enfría con un sistema de agua de enfriamiento. En algunos reactores de potencia el refrigerante del reactor es agua que se convierte directamente en el vapor que se emplea dentro de la turbina.

En la actualidad hay varios tipos de reactores de potencia que ya han alcanzado su etapa comercial, por ser sistemas “probados”. Los tipos principales son:

- Reactores de agua a presión (RAPr o PWR).
- Reactores de agua hirviendo (RAH o BWR).
- Reactores de Agua pesada (RAPe o HWR).
- Reactores enfriados por gas (REG o GCR).

El papel principal de todos estos generadores es generar calor, en función del combustible nuclear y con el aprovechamiento de este calor, producir energía eléctrica. [8]

### **1.3.1.Reactores de agua ligera a presión (PWR).**

En este tipo de reactor, los ensambles de combustible se encuentran dentro de una vasija a presión, llana de agua ligera, que desempeña el papel de moderador y refrigerante y a pesar de la alta temperatura que reina en su interior, no entra en ebullición debido a la presión interna de la vasija.

El agua caliente se extrae del reactor y se envía al generador de vapor, que no es más que un intercambiador de calor, donde el agua cede gran parte de su energía calorífica a otro volumen del mismo líquido, para después regresar al reactor. Por su parte, el agua que fue calentada en el generador de vapor, entra en ebullición, produciéndose así el vapor que sirve para mover al grupo turbogenerador, para que posteriormente sea condensado por un tercer circuito de agua procedente de un lago, río, o una torre de refrigeración.

La vasija presurizada de un reactor típico tiene unos 15 m de altura y 5 m de diámetro, con paredes de 25 cm de espesor, el núcleo alberga unas 80 toneladas de dióxido de uranio, contenidas en tubos delgados resistentes a la corrosión y agrupados en haces de combustible. Este tipo de reactor utiliza como combustible uranio enriquecido.

### **1.3.2.Reactores de agua ligera hirviendo (BWR)**

Se asemeja mucho a los PWR, ya que también utiliza agua ligera como moderador y refrigerante, y como combustible uranio enriquecido. La diferencia estriba en que en los reactores BWR el agua entra en ebullición en el interior de la vasija, produciéndose así directamente el vapor que se utilizará para mover el turbogenerador; el vapor posteriormente es condensado por el agua de enfriamiento procedente de una fuente independiente como un lago, río o el mar, regresando como agua caliente al reactor para repetir el ciclo.

La sencillez de este reactor lo hace menos costoso, y la ausencia de los generadores de vapor determina que su eficiencia sea más elevada que la del PWR. Ambos reactores integran la familia de los reactores de agua ligera, que dominan ampliamente el mercado de la industria nucleoelectrónica. Este tipo de reactor es el utilizado en Laguna Verde.

### **1.3.3.Reactores de agua pesada a presión (PGR ó CANDU).**

La principal característica de este reactor, desarrollado en Canadá y conocido también como CANDU (Canadian Deuterium Uranium), consiste en que utiliza uranio natural como combustible y agua pesada como moderador y refrigerante.

El núcleo de éste reactor, se encuentra contenido dentro de un cilindro lleno de agua pesada denominada “calandria” , el cual está atravesado axialmente por tubos de paredes relativamente gruesas llamados “tubos de presión”, en cuyo interior se alojan los elementos combustibles. Por dentro de los tubos, bañando los elementos combustibles para refrigerarlos, circula agua pesada cuya temperatura como es natural, se eleva considerablemente sin llegar a la ebullición, debido a la elevada presión que prevalece en el interior de los tubos.

El agua pesada caliente pasa a continuación a los generadores de vapor, en los que se transfiere gran parte de su energía térmica a otro circuito que contiene agua ligera, regresando posteriormente al reactor para continuar con el proceso de refrigeración del mismo.

Por su parte el agua ligera al ser calentada entra en ebullición, produciéndose el vapor destinado a mover el turbogenerador; este vapor, como en los casos anteriores, es condensado y regresado como agua caliente al generador de vapor para repetir el ciclo.

#### **1.3.4.Reactores refrigerados por bióxido de carbono y moderados por grafito (GCR).**

Aún cuando en las primeras etapas del desarrollo de la industria nucleoelectrica estos reactores ocuparon un lugar importante, su popularidad ha disminuido sensiblemente con el tiempo debido principalmente a razones económicas.

A diferencia de los anteriores, este sistema no utiliza agua como refrigerante, sino bióxido de carbono; emplea grafito como moderador y uranio natural en forma metálica como combustible.

En estos reactores la vasija está sustituida por un contenedor de concreto de gruesas paredes. El núcleo está formado por una gran cantidad de ensambles combustibles, localizados en el interior de una pila de bloques de grafito, por los que atraviesan los ductos en los que circula el gas refrigerante que es el  $\text{CO}_2$  .

Este gas arrastra el calor generado por la reacción nuclear y al circular posteriormente por el serpentín de un intercambiador de calor lleno de agua eleva la temperatura de ésta haciéndola hervir.

El vapor producido es conducido al turbogenerador para generar la energía eléctrica, posteriormente es condensado y regresado al intercambiador de calor para repetir el ciclo.

El  $\text{CO}_2$  después de ceder gran parte de su calor al agua, es recirculado de nuevo a través del núcleo para mantenerlo refrigerado.

#### **1.3.5.Reactores de cría enfriados por sodio (LMFBR)**

Este reactor funciona principalmente con neutrones rápidos, es decir, de elevada energía, teniendo la particularidad de producir más combustible que el que consume en su operación. Esto es posible gracias a la propiedad que tiene el uranio 238 de convertirse en plutonio 239 mediante la captura de un neutrón

libre, dando lugar a que de un núcleo no fisionable se obtenga otro que sí es y por lo tanto puede ser utilizado en la manufactura de combustible. Este reactor utiliza combustible enriquecido en más del 20% ya sea con uranio 235 o plutonio 239 y sodio líquido como refrigerante. Su peculiaridad es que el núcleo se rodea con un manto de uranio natural o empobrecido, que al absorber neutrones poco moderados, se transforma en plutonio y de esta manera se crea nuevo combustible.

Todo el conjunto del núcleo y capa fértil mide unos 3 m de alto por unos 5 m de diámetro, y está montado en una gran vasija que contiene sólido líquido que sale del reactor a unos 500° C. El vapor se genera en un circuito secundario de sodio separado del circuito de refrigeración del reactor, por los intercambiadores de calor intermedios para posteriormente mover el turbogenerador, condensarse y repetir el ciclo. Todo el sistema del reactor nuclear está situado dentro de un gran edificio de contención de acero y hormigón.

El plutonio criado puede utilizarse como carga inicial de nuevos reactores de cría o como recargas de reactores CANDU, PWR, BWR ó GCR.

### 1.3.6. Reactores de propulsión.

Para la propulsión de grandes buques de superficie, como el portaaviones *Nimitz* se emplean reactores nucleares similares al PWR. La tecnología básica del sistema PWR fue desarrollada por primera vez en el programa estadounidense de reactores navales. [9]

TABLA 2 - DIFERENTES TIPOS DE REACTORES TÉRMICOS <sup>1</sup>						
DESIGNACIÓN	SIGLA (EN INGLÉS)	COMBUSTIBLE	REFRIGERANTE PRIMARIO	MODERADOR	BARRERA DE PRESIÓN	CICLO DE GENERACION DE VAPOR
REACTOR DE AGUA A PRESIÓN	PWR (Pressurized Water Reactor)	dióxido de uranio enriquecido al 3,2% aproximadamente en el isótopo U <sup>235</sup>	agua común	agua común	recipiente de presión	indirecto
REACTOR DE AGUA EN EBULLICIÓN	BWR (Boiling Water Reactor)	dióxido de uranio enriquecido al 2,6% aproximadamente en el isótopo U <sup>235</sup>	agua común	agua común	recipiente de presión	directo
REACTOR DE AGUA PESADA	PHWR (Pressurized Heavy Water Reactor)	dióxido de uranio natural	agua pesada	agua pesada	recipiente / tubos de presión	indirecto
REACTOR REFRIGERADO CON GAS	AGR – MAGNOX (Advanced Gas-cooled Reactor)	uranio natural y dióxido de uranio enriquecido al 2,3% aproximadamente en el isótopo U <sup>235</sup>	dióxido de carbono	grafito	recipiente de contención <sup>2</sup>	indirecto
REACTOR DE TUBOS DE PRESIÓN MODERADO CON GRAFITO	RBMK <sup>3</sup>	dióxido de uranio enriquecido entre el 2,0 y el 2,4% en el isótopo U <sup>235</sup>	agua común	grafito	tubos de presión	directo

<sup>1</sup> Reactores nucleares en los que la reacción de fisión es sostenida por neutrones térmicos.  
<sup>2</sup> Estos reactores tienen núcleos de grandes dimensiones y baja densidad de potencia; el gas refrigerante circula a presión moderada, razón por la cual el recipiente de presión actúa también como sistema de contención.  
<sup>3</sup> Siglas rusas; la central de Chemobyl tenía cuatro unidades RBMK; la número 4 fue destruida por un accidente en abril de 1986.

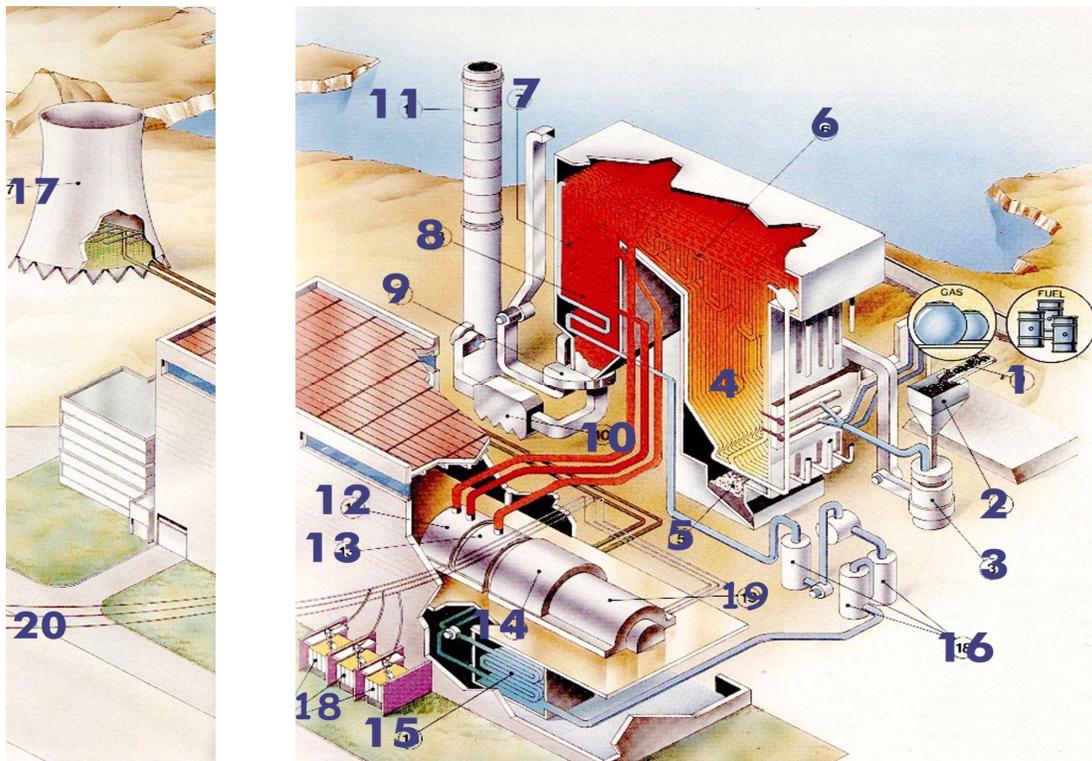
Tabla 1.1: Diferentes tipos de reactores térmicos.

#### 1.4. Relación con plantas termoeléctricas.

Varios sistemas en las plantas nucleoelectricas, también son empleados en otros tipos de plantas de potencia, como las plantas termoeléctricas.

Los sistemas de agua de alimentación se encuentran en todas la plantas térmicas, como las termoeléctricas ya sea de gas o de combustoleo, e incluso las de ciclo combinado, de hecho, el empleo de calentadores de baja y alta presión así como los sistemas de condensado, son herencia de este tipo de plantas, con el fin de aumentar la eficiencia del ciclo de potencia y reducir posibles daños térmicos en las componentes del reactor.

El sistema de agua de alimentación funciona de la siguiente manera: Después de accionar las turbinas, el vapor pasa a la fase líquida en el condensador (15). El agua obtenida por la condensación del vapor se somete a diversas etapas de calentamiento (16) y se inyecta de nuevo en la caldera en las condiciones de presión y temperatura más adecuadas para obtener el máximo rendimiento del ciclo. El sistema de agua de circulación que refrigera el condensador puede operarse en circuito cerrado, trasladando el calor extraído del condensador a la atmósfera mediante torres de refrigeración (17), o descargando dicho calor directamente al mar o al río.



**Figura 1.1:** Esquema de una termoeléctrica. El sistema de agua de alimentación es similar al de una planta nucleoelectrica.

## **1.5. Estado Actual.**

### **1.5.1. Panorama actual de la energía eléctrica.**

La energía eléctrica empleada en la industria, servicios domésticos y alumbrado público se genera principalmente en plantas hidroeléctricas y centrales termoeléctricas. En las centrales hidroeléctricas se aprovechan caudales permanentes de agua que generalmente se almacenan en presas. El agua se hace caer por medio de una tubería de presión, transformando la energía potencial gravitatoria en energía cinética, la cual es transformada a su vez en energía eléctrica al impulsar el agua al rodete de una turbina acoplada a un generador de corriente eléctrica. En las centrales termoeléctricas tradicionales se produce calor a costa de quemar combustibles fósiles.

Durante casi todo el siglo pasado, los combustibles fósiles (derivados del petróleo y gas natural), han abastecido la mayor parte de los requerimientos energéticos del mundo. Sin embargo, en un futuro no muy lejano estas fuentes de energía se encontrarán agotadas. A final de los años 70's el consumo total de energía de todos los países era alrededor de los  $1 \times 10^{17}$  BTU por año. Para el año 2000, fue de casi cinco veces esta cantidad.

La selección de centrales generadoras depende de las posibilidades de cada lugar. Donde hay recursos hidráulicos y caídas convenientes se le da preferencia al sistema hidráulico; cuando las condiciones lo permiten el sistema de generación puede ser hidráulico y térmico; Solo cuando el agua es escasa y las condiciones del terreno no son favorables, se recurre a la generación termoeléctrica exclusivamente.

La generación eléctrica obtenida en centrales hidráulicas no es tan barata como suele pensarse. Las obras civiles y eléctricas que deben realizarse incluyen presas, canales, compuertas, tuberías de presión, casa de máquina, líneas de transmisión, etc. Todo esto hace que la inversión en este tipo de centrales hidroeléctricas sea alta aunque su combustible "sea barato". Además, no siempre es posible privar el agua a una región agrícola con el fin de construir una central hidráulica.

Los caudales de agua quedan limitados a las condiciones meteorológicas y algunas centrales hidráulicas han sido desmanteladas por la escasez de agua, después de algún tiempo de haber sido instaladas.

La mayor parte de los combustibles fósiles, tales como la hulla, turba, esquistos bituminosos, petróleo y gas natural se queman en los hornos de las centrales térmicas. La química ha demostrado que no es conveniente usar estos recursos naturales no renovables como combustibles, ya que, por ejemplo, de la hulla se puede obtener alcohol, caucho, lubricantes, fibras artificiales, plásticos, medicinas, explosivos, etc. También son muchos los productos que se derivan del petróleo.

Para suministrar el combustible a las centrales termoeléctricas, se necesitan vías terrestres, ferroviarias, a veces el empleo de barcos tanques u oleoductos para hacer llegar el combustible a su destino, donde se almacena en grandes recipientes. Esto encarece la generación de la corriente eléctrica.

Es por esto, que se hace necesaria la utilización de fuentes de energía alternas al petróleo. Dos de estas fuentes que han sido consideradas son la Energía Solar y la Energía Nuclear.

El descubrimiento de la fisión nuclear en 1939 fue un evento de gran importancia, dado que abrió el prospecto de una nueva fuente de poder, utilizando la energía interna de los núcleos atómicos.

Los materiales básicos con los que se pueden ser utilizados para producir energía nuclear por fisión son los elementos Uranio y Torio. Los minerales que contienen a estos elementos están distribuidos ampliamente en la corteza terrestre a una profundidad de entre 4 y 5 kilómetros, aproximadamente  $10^{12}$  tons, lo que representa un potencial considerable de energía.

Para satisfacer la demanda de energía eléctrica, diversos países han empezado a utilizar reactores que generan corriente eléctrica, ya que estos pueden competir con las grandes centrales térmicas e hidráulicas con respecto al costo de generación de energía siempre y cuando la capacidad de generación del reactor sea suficientemente grande para las condiciones particulares de estos países.

Los reactores de potencia tienen como ventajas sobre las centrales térmicas, que no requieren oleoductos, trenes con tanques de combustible, carreteras para conducir carros tanques, ni barcos tanque. Tampoco se consumen combustibles fósiles no renovables que producen una gran cantidad de productos de combustión y que al ser expulsados por grandes chimeneas contaminan el ambiente.

Por último, la central nuclear se construye en un área menor a la necesitada por las centrales térmicas e hidráulicas. Sin embargo, ésta se presenta como una de las mejores opciones para sustituir una gran porción de la energía fósil que actualmente impera en el ámbito mundial para la generación de energía eléctrica. Actualmente, cerca del 20% de la energía eléctrica generada en el mundo es por medios nucleares. Lo anterior genera una necesidad de despertar el interés de los sectores científico y tecnológico del país por el tema de la Energía Nuclear.

Países con mayor numero de reactores nucleares (2000)		Países con mayor porcentaje de energía proveída por reactores (1999)	
Estados Unidos	103	Francia	75 %
Francia	59	Lituania	73 %
Japón	53	Bélgica	58 %
Inglaterra	35	Bulgaria	47 %
Rusia	29	Eslovaquia	47 %
Alemania	19	Suiza	47 %
Corea del Sur	16	Ucrania	44 %
Canadá	14	Corea del Sur	43 %
Ucrania	13	Hungría	38 %
Suiza/India	11	Armenia	36 %

**Tabla 1.2:** Panorama del empleo de la energía nuclear en el mundo [10]

Para nuestro caso particular, las áreas relacionadas al campo de la energía nuclear son poco conocidas para la mayoría de los egresados de la facultad lo que no genera suficientes conocimientos de los beneficios que pueden obtenerse de las tecnologías utilizadas en ese campo.

### **1.5.2. Panorama Actual en Simuladores de Plantas Nucleoeléctricas**

Existen varios esquemas de simulación que se emplean actualmente en diversos sistemas, ya que estos persiguen diversos objetivos. Los simuladores que emplean códigos de análisis específico de algún componente, los simuladores que pretenden proveer la apariencia de dispositivos y paneles en la planta real para la experimentación. Los simuladores de escritorio que pretenden obtener las tendencias para un uso didáctico o de diseño, etc. algunos tipos se listan a continuación.

#### **1.5.2.1. Tipos de Simuladores.**

Simuladores tipo panorámico.

Los simuladores de este tipo tienen una pantalla grande, que despliega una imagen de la planta, Los operadores en entrenamiento pueden confirmar el comportamiento de la planta, como resultado de su manipulación, y observar los cambios efectuados en la pantalla.

Simuladores tipo escritorio.

Este tipo de simuladores son muy compactos, Todas las operaciones y procesos son visualizados en un monitor CRT. La capacidad de tener un entrenamiento completo es posible debido a que la lógica de control y el modelo de la planta son básicamente los mismos que en los otros simuladores. El simulador puede instalarse con un costo inicial bajo y con un requerimiento de espacio menor al de otros simuladores.

Simulador tipo réplica total.

Este tipo de simulador es una fiel copia de un panel de control, con el mismo diseño y equipo de una planta real. El comportamiento de este simulador es acorde al real.

Simulador tipo compacto.

Este simulador es una versión reducida del de réplica total, sin pérdida de las principales capacidades operativas ni funcionales.

### 1.5.2.2. Simuladores existentes.

Entre los simuladores de plantas nucleares existen los simuladores de gran alcance, de tipo comercial que son desarrollados por empresas dedicadas a la simulación y construcción de plantas eléctricas, líneas de transmisión, transporte eléctrico, etc. Son empresas que tienen varias decenas de años proporcionando los simuladores y cada año presentan productos nuevos o actualizaciones a los ya existentes. Estos productos requieren hardware específico muchas veces proporcionado por la misma empresa que vende el simulador, así como de módulos de entrada y salida y de medición específico que también proporciona la empresa a manera de particularizar el simulador a una planta existente.

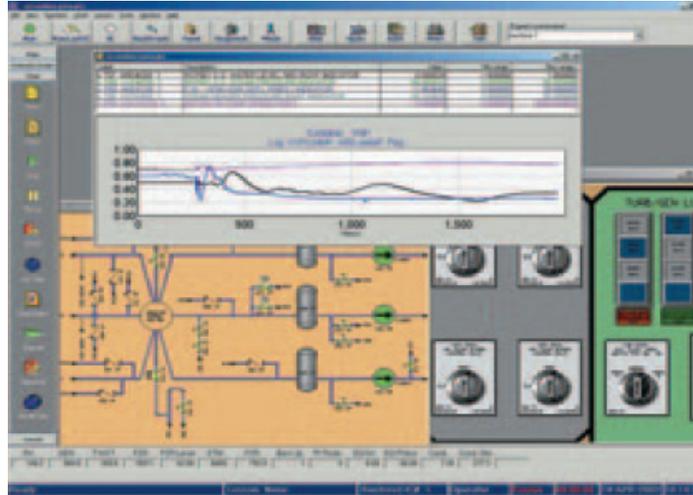
Por otra parte existen los simuladores dedicados al entrenamiento y/o docencia, que no están enfocados en el ámbito comercial y que son proyectos de universidades o de algunas empresas. Estos simuladores son proyectos que tienen como finalidad mostrar las potencialidades de la energía nuclear a los estudiantes y mostrar el funcionamiento de una planta nuclear a futuros operadores o técnicos que necesiten observar comportamientos a ciertas condiciones de emergencia, operación normal, incremento de potencia, etc.

Estos simuladores no cuentan con hardware específico proporcionado por los desarrolladores, por lo que dependen de la interfaz gráfica para mostrar las variables y los datos que requieren, así como para manipular los parámetros de operación. Es por esto que la interfaz debe poder imitar la apariencia del panel real. Por lo general estos simuladores están basados en nuevas tecnologías como Java, Flash o DirectX, pero también los hay en entornos HTML o de otro tipo. De entre los simuladores comerciales podemos mencionar los de empresas como CAE o Invensys.

De los simuladores de tipo académico, existen varios esfuerzos desarrollados en Flash o en Java, los cuales se mencionaran más adelante cuando se justifique la elección de el desarrollo en Matlab y C, contra los desarrollos en Flash, Java ,Basic, etc.

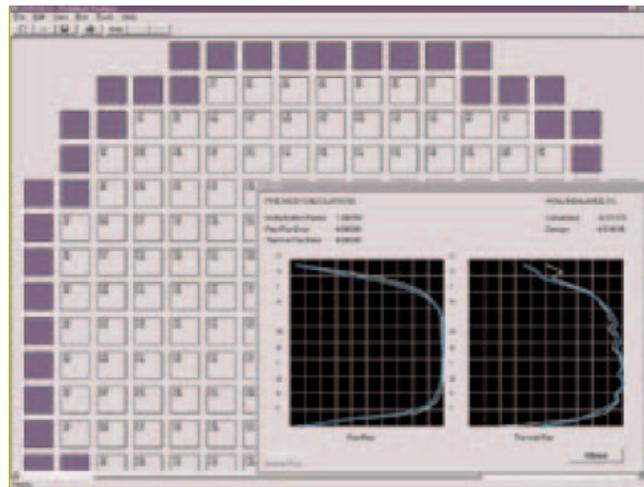
La empresa canadiense CAE provee simuladores de replica total, y de escritorio desde 1970, en diversos campos de la simulación de plantas de potencia. Han desarrollado simuladores de plantas nucleoelectricas específicas en varios países como Estados Unidos, Suecia, China, Suiza, Sudáfrica, Rumania, etc. Cuenta con varios proyectos de simulación general como el proyecto Isis™, que es una estación de instrucción diseñada para asistir al instructor para controlar el entorno de entrenamiento, y monitorear su progreso. Isis es un simulador basado en tecnología Web, implementado en un servidor Java, que provee interconexión entre servidores de varias plataformas ( Linux, Windows, PDAs, etc) a través de una capa de comunicación XML.

Este tipo de simulador provee lecciones virtuales que debe seguir el operador en entrenamiento y que pueden ser seguidas y corregidas por el instructor desde otro lugar por medio de reportes.



**Figura 1.2:** Pantalla del simulador de entrenamiento Isis™.

El simulador Comet, es otro simulador de la empresa CAE es un simulador de modelo de cinética de reactor, basado en principios físicos y diversas técnicas de simulación, ya sea de PWR y BWR, para su uso en entrenamiento o para mejorar el rendimiento general del reactor.



**Figura 1.3:** Pantalla del simulador de cinética del reactor Comet™.

La empresa Invensys™, cuenta con el simulador OTS (Operator Training Simulator), el cual es un entorno para entrenar operadores y para efectuar análisis que sirvan para el diseño o la optimización de los procesos de una planta. Este es un simulador tipo escritorio que puede ser expandido por módulos de la misma empresa para agregar funcionalidades, como el monitoreo de accidentes o situaciones de emergencia, manejo de combustible, etc.

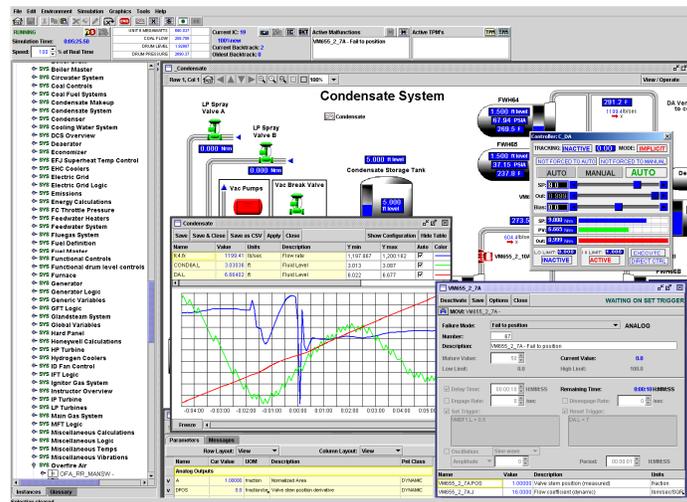


Figura 1.4: Simulador para entrenamiento de operadores [OTS] de la empresa InvenSys™.

### 1.6. Planeamiento General del Problema a Resolver.

En la UNAM actualmente no se cuenta con una herramienta computacional o simulador de una central nucleoelectrica que pueda ser utilizado para el entrenamiento de estudiantes o ingenieros interesados en el funcionamiento de ellas. Existen algunas propuestas para desarrollar un simulador de aula que permita modelar en detalle los sistemas principales de una central nuclear tipo BWR con mímicos de los paneles de una central como Laguna Verde, el cual utilizará el código RELAP-V como herramienta de modelado matemático de los sistemas físicos. Sin embargo, aún contando con el no se tendrá un acceso múltiple desde Internet ya utilizará las computadoras dedicadas del Laboratorio de Ingeniería de Reactores Nucleares de la Facultad de Ingeniería. Para un uso accesible múltiple desde cualquier PC conectada a Internet se piensa que un simulador con modelos de orden reducido puede instalarse en un servidor y dar el comportamiento de tendencias que tiene una central tipo BWR sin utilizar muchos recursos de computo. A continuación se presentan algunos simuladores disponibles en Internet, desarrollados para entrenamiento y fines académicos no comerciales.

Uno de ellos es el Simulador de Principios Básicos de la Central Nuclear Embalse, desarrollado por el Centro Atómico Bariloche en Argentina por Celso Alberto Flury y Félix Maciel Palacio, este simulador esta basado en modelos matemáticos de algunos componentes, principalmente el reactor, para analizar su desempeño. [11]

Otro simulador es el Flash Nuclear Power Simulator de Geoffrey Noles escrito en Basic y con interfaz en Flash. [12]

El simulador “Control The Nuclear Power Plant (Demonstration)” de Henrik Ericsson [13] es un simulador escrito en Java que simula condiciones de emergencia en secuencias predeterminadas,

Presentando animaciones y permitiendo interactuar con el sistema accionando ciertas válvulas y bombas en el ciclo de potencia.

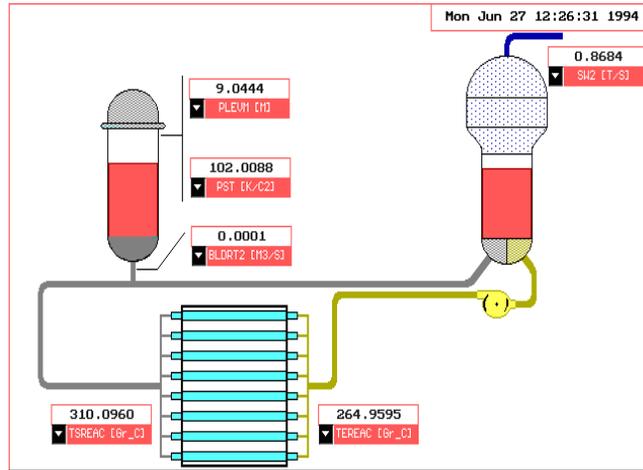


Figura 1.5: Simulador de principios básicos de la Central Nuclear Embalse, modelado físico de algunos componentes.

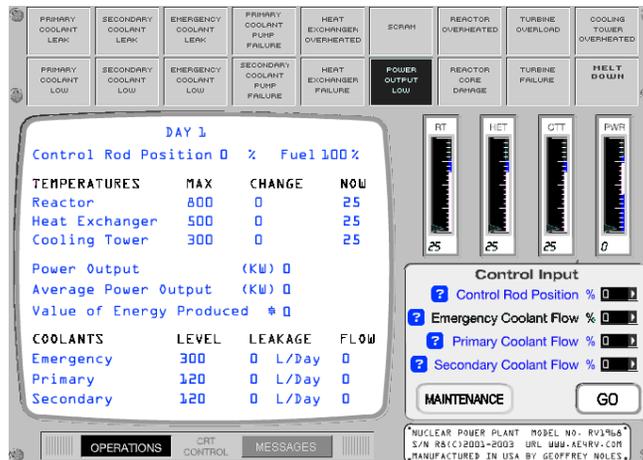


Figura 1.6: Flash Nuclear Simulator, simulador escrito en Flash con código en Basic.

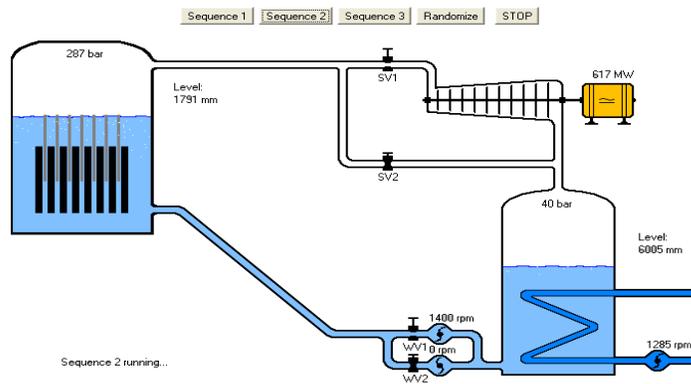


Figura 1.7: “Control the Nuclear Power Plant”, escrito en Java, con estados predeterminados.

### **1.7. Solución esperada.**

Como resultado del método de simulación que se eligió para el desarrollo del simulador SUN-RAH, se espera tener una simulación en tiempo continuo, con tiempos de respuesta similares a los que se tendría en la planta real o incluso más rápido, considerando que algunos procesos reales tardan hasta varios minutos en mostrar cambios perceptibles, y que en el simulador pueden observarse estos en pocos segundos.

Del simulador se espera que cumpla con los modelos de los cuales esta formado, de manera integrada y confiable, de forma tal que de acuerdo a los parámetros y rangos de las señales determinados, efectué las transformaciones indicadas, y presente las señales dentro de los esquemas lógicos de operación de los diversos componentes de la planta.

El simulador deberá presentar una interfaz de usuario atractiva, de fácil acceso y uso, intuitiva en su manejo y con posibilidad de visualizar los valores e información varia de los procesos de la planta. Esta interfaz deberá dar la posibilidad de acceder a los diversos componentes de la planta en forma inmediata o por medio de capas de acercamiento (*scope*). La interfaz contará también con la posibilidad de alterar valores de válvulas o controles propios de la operación de la planta y de observar en tiempo real los resultados de la alteración realizada.

Los sistemas deberán residir en un entorno de simulación matemática como Simulink o Matlab y la interfaz se ejecutará como una aplicación Windows basada en las librerías DirectX. La simulación podrá ejecutarse sin la presencia de la interfaz grafica de manera que los modelos puedan depurarse de manera fácil e inmediata. Por su parte la interfaz también podrá ejecutarse independientemente con algunas modificaciones mínimas, de manera que pueda alterarse o agregarse funcionalidad, sin depender de valores de la simulación, con lo que la interfaz puede ser mejorada continuamente en futuras revisiones.

La interfaz y la implementación de la situación en Matlab deben cumplir con el propósito de ser atractivas para los inexpertos y los nuevos estudiantes recién allegados a las disciplinas nucleares, Los modelos deben ser lo suficientemente formales y exactos para proveer señales que se comporten como los sistemas reales.

En esta tesis, el capítulo uno es una introducción, donde se presentan los temas a tratar, y los marcos en los que la tesis se desarrolla. Se mencionan los tipos de reactores nucleares existentes y las características de los sistemas a simular, así como un panorama de los simuladores existentes.

En el capítulo dos, se hace un análisis del sistema mencionando la planeación, que se consideró en el desarrollo de los componentes del sistema.

En el tercer capítulo se da una explicación general del sistema que se va a modelar matemáticamente, identificando los dispositivos componentes y explicando su funcionamiento general.

En el capítulo cuarto, se detallan los modelos empleados y se describe la implementación elegida para su análisis y posterior introducción en un entorno de simulación.

En el capítulo cinco, se justifica el empleo de una interfaz gráfica como protocolo de comunicación con el usuario y describe el funcionamiento de los componentes y estructuras que la integran.

El capítulo seis muestra las pruebas que se le efectuaron a los modelos, la interfaz y su integración. Se hace una comprobación de estabilidad y de confiabilidad, a fin de validar los modelos.

El último capítulo presenta las conclusiones obtenidas del trabajo de tesis y las perspectivas a futuro de este tipo de proyectos.

### **Referencias bibliográficas.**

[1] NukeWorker.com, *Nuclear Plant Personnel Training Facts*

[http://www.nukeworker.com/study/radiation\\_faqs/Training\\_Facts.shtml](http://www.nukeworker.com/study/radiation_faqs/Training_Facts.shtml)

[2] Instituto de Investigaciones Eléctricas: *Una historia del Instituto de Investigaciones Eléctricas: aceptar el reto / Efervescencia*. México.

<http://www.iie.org.mx/historia2000/eferve.pdf>

[3] Paul Fishwick: *Identity Crisis* pag 5

Feature Column: The art of modeling en *Modeling & Simulation* Volume 2, Number 2 April-June 2003

[4] Ziegler B.P., *Theory of Modelling and Simulation*, John Wiley & Sons, New York 1976.

[5] Gordon G., *The Application of GPSS to Discrete System Simulation*, Prentice Hall, Englewood Cliffs, New Jersey, 1975.

[6] Pritsket A.A.B., *Compilation of definitions of simulation*, SIMULATION, August 1979.

[7] Stanislaw Raczynski, *Simulación por Computadora*, México, Editorial Limusa 1993.

[8] Eduardo Felizia, *Artículo: Centrales Nucleares: La Evaluación Probabilística de su Seguridad*,

En Pagina Web de la Revista de Divulgación Científica y Tecnológica de la Asociación Ciencia Hoy, Volumen 5 - N°35 – 1996

<http://www.ciencia-hoy.retina.ar/hoy35/centr02.htm>

[9] The European Nuclear Society, *Glossary of nuclear terms by Winfried Koelzer*,

<http://www.euronuclear.org/info/encyclopedia.htm>

[10] Nationmaster.com, *Map & Graph: Energy: Top 100 Nuclear electricity generation*,  
[http://www.nationmaster.com/graph-T/ene\\_nuc\\_ele\\_gen](http://www.nationmaster.com/graph-T/ene_nuc_ele_gen)

[11] Celso Alberto Fluir, *Simulador de Principios Básicos de la Central Nuclear de Embalse*,  
<http://www.cab.cnea.gov.ar/cab/ingnuclear/control/simulador.htm>

[12] Geoffrey Noles , *Flash Nuclear Power Simulator*,  
<http://www.ae4rv.com/games/nuke.htm>

[13] Henrik Ericsson, Linköping University, *Nuclear Power Plant Applet Description*.  
<http://www.ida.liu.se/~her/npp/description.html>

## Capítulo 2. Análisis del sistema.

### 2.1. Identificación de las necesidades.

El escenario actual de la energía nuclear, revela una necesidad por abandonar la dependencia de la generación basada en hidrocarburos. Las alternativas reclaman atención, y hay escenarios en donde en algunos años incluso la actual aproximación a la generación con gas, luce insuficiente.

La energía nuclear, vuelve a rondar la escena de la generación en varios países del mundo, en donde se había olvidado o se ha ocultado su franca presencia.

Los países industrializados del primer mundo, tales como Francia, Japón, E.U. etc., gastan millones de dólares en campañas informativas que ayuden a incrementar la confianza de la gente en este tipo de energía. La mayor aceptación o tolerancia hacia las plantas nucleares, permitirá abrir más de ellas y proveer de una energía más rentable para los países y las compañías generadoras.

Gran parte de la confianza que un pueblo o gobierno pudieran tener en la industria nuclear, depende de la capacitación de los operadores, ya a que los accidentes de este tipo de plantas se han debido, entre otras cosas, a inexperiencia o conocimiento deficiente de la operación de la planta. Por otra parte, varios accidentes o situaciones de emergencia se han superado debido al conocimiento profundo de los operadores y su capacidad para mantener la planta segura e incluso regresarla a un estado óptimo de generación, como lo muestran los niveles actuales de seguridad, eficiencia y disponibilidad.

Las aplicaciones de la tecnología en el entrenamiento y la docencia de los temas de energía, requiere que existan simuladores en los cuales experimentar, representar, analizar, diseñar, educar, etc., los efectos de la manipulación de parámetros en un modelado específico. Dependiendo del alcance y enfoque de los simuladores estos consideran diferentes niveles de refinamiento y de complejidad.

En el caso del simulador SUN-RAH, y en particular del simulador de agua de alimentación, se requiere que el modelado cuente con el suficiente detalle para simular los eventos comunes que le ocurrirían a los modelos en operación en estado estable en una planta nucleoelectrónica, a su vez debe ser capaz de responder a las variaciones en las señales de entrada, y generara una salida en un tiempo razonable, similar al tiempo real de respuesta de el dispositivo que se esta simulando.

También se necesita implementar el modelo del controlador que típicamente se encuentra en una planta real y que en la simulación proporcione una señal de control que depende de varias condiciones ponderadas adecuadamente.

Las señales de control deberán poder visualizarse en algún panel de control similar al de la planta real, en donde se consignan los valores y las gráficas a medir y se puedan alterar con controles que imiten los que se encuentren en la planta.

Finalmente se necesita que la interfaz grafica permita visualizar esquemas de los componentes y mostrar animaciones e información acerca del estado actual de la planta de una manera atractiva, con

el fin de disminuir las diferencias entre las interfaces académicas y las más desarrolladas interfaces de entretenimiento.

Las necesidades técnicas del proyecto dependen de los recursos disponibles para el desarrollo del simulador. Se necesita equipo de cómputo básico con capacidad para manejar gráficos avanzados para la interfaz y con capacidad de procesamiento y almacenamiento para soportar un entorno matemático avanzado.

La lista de requerimientos para el programa y la interfaz se menciona más a detalle en los capítulos dedicados al análisis matemático y de la interfaz gráfica respectivamente.

El tiempo de desarrollo debe ser el suficiente para cumplir con dos factores básicos, en primer lugar, para aprender los fundamentos de las tecnologías nuevas que se van a utilizar, ya que muchas de ellas eran desconocidas mas allá de la implementación de las aplicaciones que hacían uso de ellas. El otro factor de tiempo es el refinamiento y la adecuación de la interfaz de acuerdo al grado de respuesta que se necesita en los modelos, lo cual conlleva pruebas y situaciones hipotéticas congruentes con los valores reales esperados.

## **2.2. Evaluación del concepto de sistema para establecer la viabilidad.**

El desarrollo de un sistema de este tipo es posible debido al advenimiento de sistemas de cómputo casero más poderosos y a un menor costo. El aumento del poder de cómputo ha sido impulsado en gran medida por las necesidades de mejorar las capacidades multimedia de las computadoras en los hogares o escuelas.

Por lo tanto es factible que una computadora casera tenga la velocidad de procesamiento suficiente como para soportar un desarrollo en un programa de análisis matemático como Matlab, y también de soportar efectos avanzados de visualización para la interfaz gráfica tales como la translucencia, y la transparencia, los cuales se explican más adelante en el capítulo referente a la interfaz gráfica.

El desarrollo se realizará con programas disponibles en versiones evaluatorias o introductorias proporcionadas por libros o disponibles en Internet, o en software con licencia de la UNAM.

Por lo tanto en el aspecto de equipo de hardware y software se cuenta con los elementos básicos a bajo costo. Sin embargo, si se deseara ampliar las capacidades del modelado o llevar el sistema a nivel de un simulador con mayor detalle o mayores despliegues se requeriría un mayor número de desarrolladores, editores gráficos, programadores, etc.

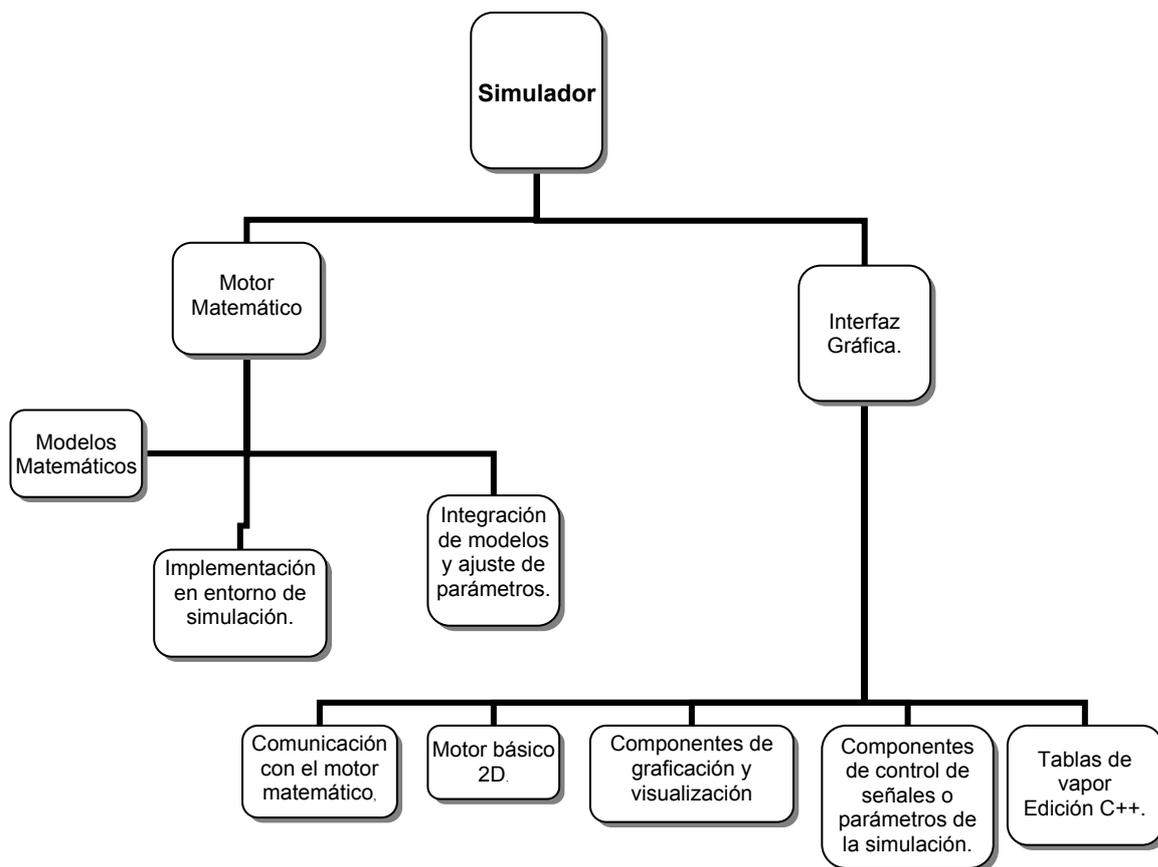
Se cuenta con el suficiente tiempo de desarrollo para comprender los mecanismos de las tecnologías a utilizar, adaptarlas a nuestras necesidades y refinar los modelos hasta que se encuentren integrados en una estructura modelable que entregue datos representativos de los procesos de la planta. Sin embargo, un desarrollo mayor con la búsqueda y adecuación de los parámetros y modelos a mayor detalle a fin de lograr un símil con alguna planta real es por el momento un proceso de larga duración y de varios

años de depuración y refinamiento. Esto último puede plantearse en desarrollos posteriores por compañeros estudiantes o profesionales interesados en estos retos.

La disponibilidad de el lenguaje de programación propuesto: Lenguaje C++, de el entorno DirectX y de el hardware esta fuertemente garantizada por el apoyo de empresas de desarrollo de software como Microsoft®, Nvidia®, etc., las cuales encuentran provechoso que los desarrolladores empleen sus tecnologías y proveen facilidades y ejemplos para que se usen constantemente, suministrando versiones actualizadas aproximadamente cada seis meses de los componentes de software y hardware, por lo que si en el futuro se deseara actualizar el código matemático o de la interfaz, existirá manera de hacerlo fácilmente.

### 2.3. Jerarquización.

La jerarquización de los módulos que componen el simulador se presenta en el siguiente esquema:



**Figura 2.1:** Esquema jerárquico de las partes del simulador.

Como se puede observar el simulador tiene dos partes componentes principales:

El “motor matemático” y la interfaz gráfica. El componente dedicado al modelado, simulación y depuración de los modelos matemáticos es llamado “motor matemático” ya que ese encarga de realizar

las integraciones, derivaciones, y operaciones algebraicas necesarias para representar los sistemas con los que cuenta la simulación, ya se en la fase de diseño como en la de ejecución.

La parte de interfaz es la encargada de obtener los valores provenientes de la simulación que realizó el motor matemático y mostrar los resultados en diferentes gráficas o visualizaciones. Es también responsable de mostrar los controles que permitirán modificar valores en la simulación y también de comunicarle estas decisiones a el motor matemático. El desarrollo se planeó de esta manera por la comodidad de poder probar y depurar los modelos matemáticos y resultados de la simulación y por otra parte poder efficientar y agregar funcionalidades a la interfaz, sin que la dependencia de uno u otro retrasaran el proceso de desarrollo y aprendizaje en la programación de ambos entornos.

#### **2.4. Ingeniería de la Información.**

La ingeniería de la información define las siguientes arquitecturas, que ayudan a definir los procesos de manejo de la información. Para esto propone una esquematización en varios niveles en donde se distinguen las áreas de interés específico en las diversas capas concentradoras de los procesos sobre la información y datos. Los niveles se escriben como:

- Vista global.
- Vista de dominio.
- Vista de elemento.
- Vista detallada.

La vista global se refiere a los objetivos de los interesados en el desarrollo, y sus principales necesidades. La vista de dominio se refiere al área donde se consideran las características que tendrán los flujos de datos y su relevancia para los intereses globales, así como las instancias pertinentes para manejar estos flujos. La vista de elemento define los procesos específicos y como los datos serán modificados por los sistemas y funciones dentro de la aplicación de las instancias determinadas en la vista de dominio.

En la vista detallada se presenta la implementación y construcción de las funciones que se han descrito anteriormente. En el simulador de agua de alimentación y controlados asociado.

Estas capas se manejan de la siguiente manera:

- Vista global: El interés de los desarrolladores por crear un simulador de entrenamiento y/o docencia, robusto y lo suficientemente detallado para que presente las respuestas adecuadas en las condiciones de operación de una planta nucleoelectrica en estado estable. El flujo de información es la lista de requerimientos, viabilidad, recursos, etc.
- Vista de dominio: De acuerdo a los recursos y las necesidades previstas, definimos una separación de los desarrollos, planteando un entorno matemático y otro de interfaz grafica con

la suficiente independencia para agilizar la generación de resultados. El flujo de datos se divide en la parte matemática de las variables físicas y sistemas que deberán ser modelados, las condiciones de las respuestas esperadas, y por otra parte, los atributos gráficos de la interfaz, los efectos y modalidades de acceso, los gráficos adecuados, etc.

- Vista de elemento: En la parte matemática se consideran los métodos y ecuaciones específicos que serán empleados para modelar los dispositivos como el condensador, los calentadores de agua de alimentación, la bomba de agua de alimentación, el controlador de inventario de refrigerante, etc. En la parte grafica se considera la necesidad de tener medidores de nivel, tipo aguja de despliegues gráficos, graficadores de señal, botones animados, menús rotatorios, etc.
- Vista detallada: Aquí se consigna todo el código o la construcción real de los módulos descritos anteriormente. La implementación matemática se detalla en un capítulo posterior. La implementación de la interfaz se explica en otro capítulo, aunque por su excesiva extensión no se registra todo el código escrito. [1]

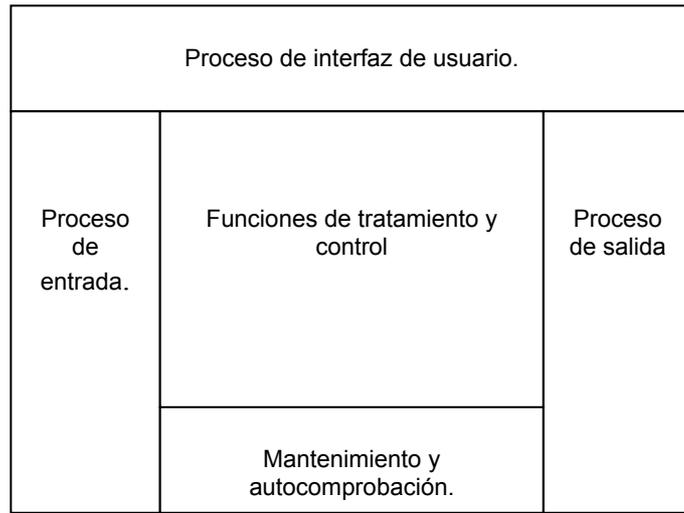
## **2.5. Ingeniería del proyecto.**

Todos los sistemas basados en computadora pueden modelarse como la transformación de la información empleando una arquitectura del tipo entrada-proceso-salida. Esta visión puede extenderse para incluir dos características adicionales del sistema: proceso de la interfaz de usuario y proceso de mantenimiento y autocomprobación. Mediante la representación de entrada, proceso, salida, proceso de la interfaz de usuario y de autocomprobación, se puede crear un modelo de componentes de sistema que establezca el fundamento para el análisis de requisitos posteriores y etapas de diseño en cada una de las disciplinas de ingeniería.

Para desarrollar el modelo de sistema, se emplea un esquema de arquitectura y se asignan elementos a cada una de las cinco regiones de tratamiento del esquema: (1) Interfaz de usuario, (2) entrada, (3) tratamiento y control del sistema, (4) salida y (5) mantenimiento y autocomprobación.[2]

Es la separación de la interfaz y la implementación, uno de los principales esquemas modernos de desarrollo, es recomendado por la independencia que permite actualizar la interfaz o los modelos de manera separada, por lo que las actualizaciones se hacen por módulos y no por reemplazo de aplicaciones enteras. [3]

Varios documentos de programación hacen la recomendación categórica de que las clases de interfaz no deben tener acceso directo a los bits de datos en la aplicación. En su lugar, las clases de interfaz harán peticiones para cambiar los datos de la aplicación llamando métodos definidos por las clases de datos de la aplicación. Este esquema de separar “iglesia” y “estado” es actualmente un excelente paradigma de diseño.



**Figura 2.2:** Plantilla de arquitectura para el proyecto.

### Referencias Bibliográficas.

[1] Roger S. Pressman. *Ingeniería del Software, Un enfoque práctico Cuarta Edición*. McGrawHill, Madrid 1997. Capítulos 8 y 9.

[2] Roger S. Pressman. *Ingeniería del Software, Un enfoque práctico Cuarta Edición*. McGrawHill, Madrid 1997. pp 174-175.

[3] Mike MacShaffry, *Game Coding Complete*, Paraglyph Press, Arizona E.U.A 2003 pag 51.

## **Capítulo 3. Diseño del simulador del sistema de agua de alimentación.**

### **3.1. Introducción.**

En esta parte se hace un diseño conceptual del simulador del sistema de agua de alimentación para el SUN-RAH. Se establecen formalmente los requerimientos y fundamentos para los modelos matemáticos así como para la interfase con el usuario. Esta última incluye un buen número de características y despliegues gráficos par el SUN-RAH que surgen de este diseño. El simulador del condensador, agua de alimentación y controlador de inventario de refrigerante del núcleo del reactor, es un desarrollo que propone un esquema de construcción de aplicaciones enfocadas en el entrenamiento y la docencia, que permitan la interacción de modelos matemáticos complejos e interfaces gráficas atractivas. Este esquema de desarrollo permitirá crear diagramas e imágenes que representen el comportamiento de los sistemas reales y que con mayor o menor realismo, desplieguen gráficos o animaciones que permitan realizar el entrenamiento de operadores, o adaptarse a diversos niveles de complejidad en la enseñanza de la tecnología. A su vez este tipo de proyectos pretende ser un atractor de desarrollos tanto de estudiantes como de diversos profesionales, con el interés de mejorar y detallar los sistemas propuestos y generar nuevos.

El enfoque computacional de este simulador, permite el desarrollo subsiguiente más detallado de los sistemas físicos a simular, así como también de personal dedicado a la computación, beneficiando a ambos grupos, por las posibilidades de expandir el campo de influencia de un área técnica, así como de difundir los beneficios de las tecnologías desarrolladas a otros grupos de especialistas. El objetivo principal del simulador de condensador y agua de alimentación es proveer los modelos que representan el proceso de regeneración en el ciclo termodinámico perteneciente a la operación de la planta.

### **3.2. Justificación termodinámica de los componentes.**

El proceso de regeneración es empleado como una modificación de los ciclos termodinámicos ideales, para proveer mayor eficiencia térmica general y evitar esfuerzos de los componentes que harían disminuir su vida útil. Como ejemplo de esto se puede ver el siguiente ciclo Ranking modificado, donde se muestra la eficiencia del ciclo con regeneración. En el ciclo Ranking el vapor es sobrecalentado en la caldera y se condensa por completo en el condensador. El ciclo ideal esta compuesto por los siguientes procesos:

- Compresión isentrópica en una bomba
- Adición de calor a presión constante en una caldera
- Expansión isentrópica en una turbina
- Rechazo de calor a presión en un condensador.

La eficiencia térmica del ciclo se determina como:

$$\eta_t = \frac{W_{neta}}{q_{entrada}} = 1 - \frac{q_{salida}}{q_{entrada}}$$

La eficiencia térmica también puede verse como la relación entra el área encerrada por el ciclo en un diagrama  $T-s$  y el área bajo el proceso de adicción de calor.

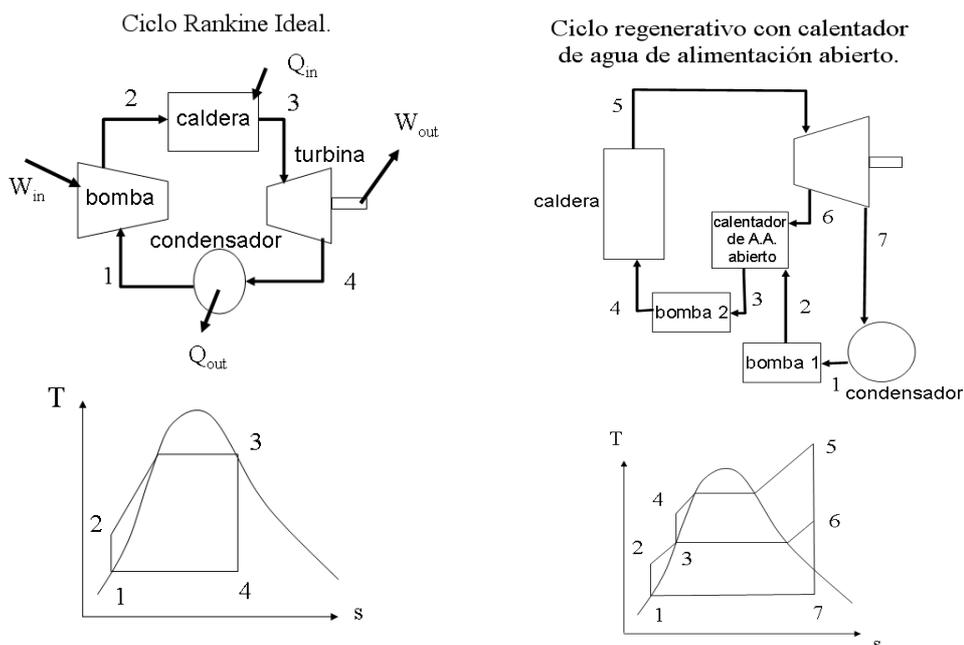
Bajo el ciclo Rankine ideal el calor que se añade al fluido de trabajo durante el proceso, se hace a una temperatura baja, por lo que se reduce la temperatura promedio a la que se añade calor, y por consiguiente, la eficiencia del ciclo es menor [1]

Para remediar esta deficiencia se eleva el calor del flujo que sale de la bomba de condensado (que recibe el nombre de agua de alimentación) antes de que entre a la caldera, una manera de hacer esto es transferir calor al agua de alimentación del vapor en expansión en una intercambiador de calor a contraflujo, es decir utilizar regeneración.

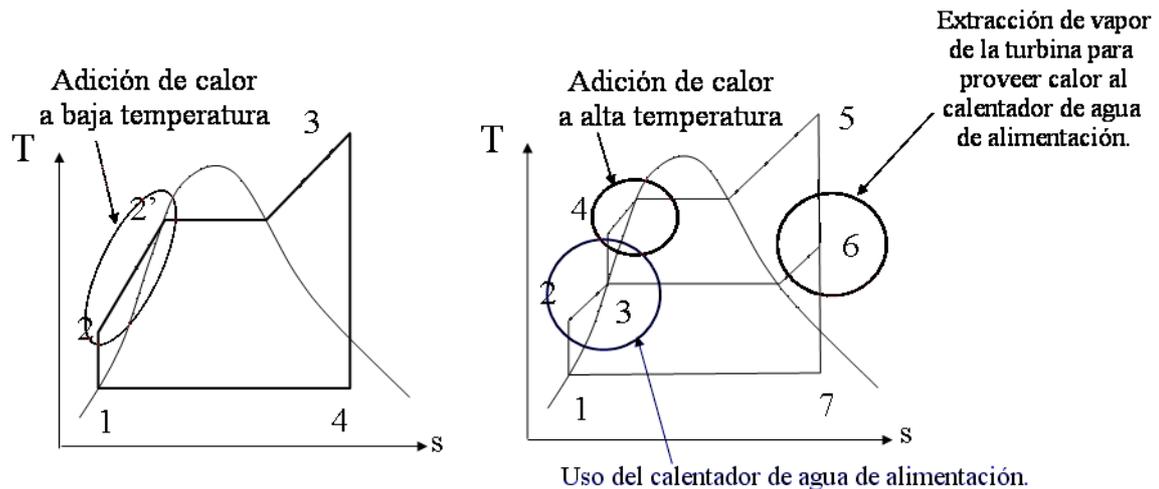
Este “drenado” del vapor se hace en diversos puntos a lo largo del flujo que pasa por la turbina, y se utiliza para calentar el agua de alimentación. El dispositivo donde el agua se calienta mediante regeneración se llama regenerador o calentador de agua de alimentación.

La regeneración no sólo mejora la eficiencia del ciclo sino que también ayuda a controlar la gran relación de flujo de volumen del vapor en las etapas finales de la turbina (debido a los grandes volúmenes específicos a grandes presiones). La regeneración se emplea en todas las plantas modernas de potencia de vapor desde su introducción a principio de la década de los veinte.[1]

Las siguientes figuras muestran las ventajas de emplear calentadores de agua de alimentación y justifican su uso en plantas de energía. Los calentadores de agua de alimentación no son exclusivos de plantas nucleoelectricas, sino que son empleados en plantas termoeléctricas, de ciclo combinado, etc.



**Figura 3.1.** Ciclo Rankine ideal y ciclo regenerativo con calentador de agua de alimentación abierto, el ciclo regenerativo mejora la eficiencia incrementando la temperatura del flujo de agua antes de entrar a la caldera.



**Figura 3.2.** Comparación entre los dos ciclos. El uso del regenerador para calentar el agua de alimentación que se dirige a la caldera, incrementa la temperatura promedio en la adición de calor en la caldera, lo que incrementa la eficiencia.

### 3.3. Métodos de solución.

Existen varios tipos de simuladores comerciales, académicos, experimentales, etc. Para realizar un simulador con los alcances y especificaciones del SUN-RAH, existen varias opciones viables dependiendo del tipo de efectividad y alcances que se pretenden lograr. Hay varias opciones de cómo representar un simulador, considerando las ventajas y desventajas de cada método. Por una parte se tiene la opción de agrupar varios posibles escenarios de operación y representar su ejecución a petición del usuario o a algún evento producido por la interacción del usuario con algún elemento gráfico.

Otro esquema es el de representar una serie de animaciones predeterminadas de uno o varios elementos gráficos, que se irían activando de acuerdo a las características consignadas en una base de datos conforme el usuario alterara los valores de algún control. Por lo general este tipo de simulaciones espera a que los valores alterados por el usuario lleguen a cierto nivel, lo compara con la base de datos, y si esta dentro de cierto rango de una lista de ellos, emprende una serie de animaciones y alteración de valores que formaran nuevos estados dentro del ambiente gráfico.

Estos esquemas de representación son útiles para la docencia y para aplicaciones de demostración de funcionamiento inmediato en donde el usuario es inexperto o es su primera aproximación a un sistema, ya que permite explorar escenarios reducidos y observar primordialmente las reacciones de los sistemas a ciertas condiciones de emergencia, accidente y operación óptima.

El método para desarrollar este simulador es el de modelar matemáticamente todos los dispositivos presentes en una planta real, para que estos reaccionaran a diversas señales, y presentar respuestas que su vez puedan ser las entradas de otros modelos, y de esta manera presentar las respuestas de cada modelo de acuerdo a las condiciones de la planta en cada paso continuo de simulación, como si se

hiciera en un dispositivo real. Los modelos de los sistemas deben representarse en ecuaciones diferenciales que describan su comportamiento. De ser necesario analizar los modelos desde el punto de vista de los sistemas lineales invariantes en el tiempo, y realizar transformaciones de Laplace para llevar los modelos a funciones de transferencia.

Las funciones de transferencia puede llevarse a entornos de análisis matemático como Matlab® y Simulink® y presentarse en una interfase gráfica basada en DirectX. Este método de simulación proporciona ventajas respecto a los otros debido a que las respuestas pueden observarse en tiempo real de manera continua. Los parámetros pueden alterarse en diferentes cantidades y el sistema reaccionará dinámicamente consecuentemente, por lo que no existen respuestas predeterminadas y se podrá ajustar posteriormente con diferentes conjuntos de parámetros para diferentes rangos de operación de la planta, permitiendo respuestas congruentes con la realidad. Este método otorga también el control que un operador necesita para llevar a cabo un entrenamiento fiable, ya que las situaciones que se presenta en una planta real, distan mucho de ser situaciones predeterminadas, por lo que el operador debe reaccionar ante cualquier situación.

Como desventaja de este método, el tiempo de modelado puede llegar a tomar más tiempo del requerido en otros esquemas de simulación debido a que se debe procurar que el modelo responda fehacientemente a la realidad por lo que el refinamiento de los modelos puede llegar a ser un proceso de análisis matemático intenso, así como la inmersión de estos modelos en los programas como Matlab y Simulink en sus lenguajes particulares.

### **3.4. Diagrama del proceso.**

El diagrama para realizar el modelado se muestra en la figura 2.3.

El diagrama del proceso de modelado del sistema de agua de alimentación puede ser similar para modelar los dispositivos del sistema y del controlador. Cabe mencionar que los parámetros no se ajustan para reproducir las respuestas de alguna planta específica.

La integración puede realizarse de lo particular a lo general, programando los modelos independientemente, para después integrarlo en módulos completos (calentadores, condensador, controlador, etc.) Las pruebas deben realizarse por módulo y posteriormente por conjunto a los modelos integrados. Las pruebas y sus resultados para los módulos individuales y de conjunto se muestran en un capítulo posterior.

### **3.5. Especificaciones.**

Las ecuaciones de los modelos de los calentadores de agua de alimentación, del condensador, y de la bomba de agua de alimentación, f analizadas desde el punto de vista de los Sistemas Lineales Invariantes en el Tiempo, por lo que se realizaron transformaciones de Laplace para llevar los modelos en ecuaciones diferenciales, a funciones de transferencia, de manera que el enfoque de

sistemas que programas de computo como Matlab, y Simulink, pudieran incorporar, integrar y simular con facilidad. Esto no es una limitación a la representación del sistema real, ya que posteriormente al tener datos de una planta específica los parámetros del modelo lineal pueden ajustarse a diferentes rangos o dominios de las variables o permitir variaciones funcionales continuas numéricas o algebraicas.

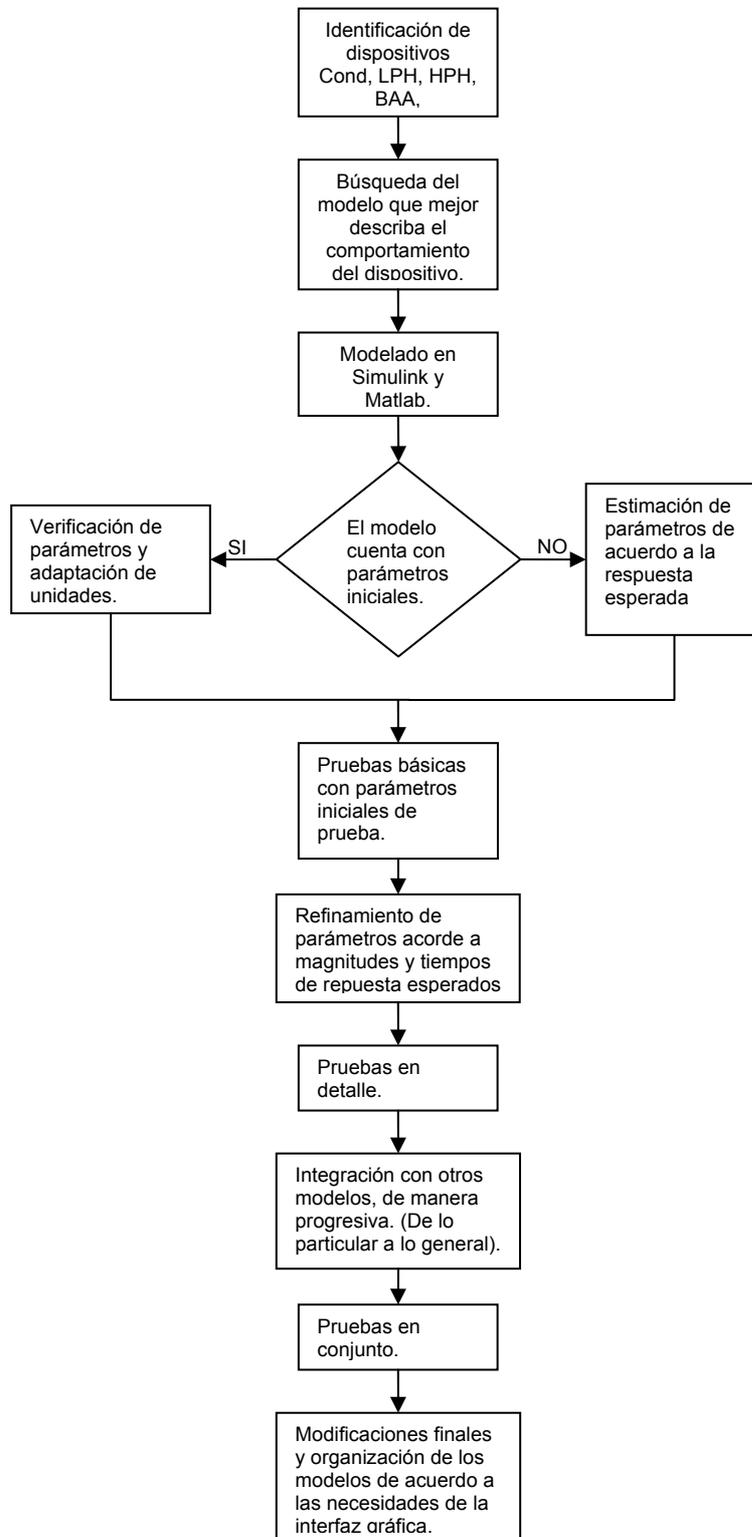


Figura 3.3: Diagrama para el proceso de modelado y pruebas.

Las especificaciones para los modelos son:

- Basado en Modelos de Orden Reducido.  
El empleo de este tipo de modelos nos permite, concentrarnos en los comportamientos, que tiene la planta alrededor de valores de estado estable. La respuesta final que alcanzan estos modelos puede ser tan buena como la de modelos más complicados.
- Plataforma de ejecución y desarrollo basado en computadoras al alcance de estudiantes.  
El simulador debe ejecutarse en computadoras compatibles IBM PC de gama media, que puedan encontrarse fácilmente en las instituciones universitarias, a niveles licenciatura y postgrado o incluso de uso domestico, de igual forma el desarrollo de este simulador puede continuarse o refinarse empleando un poder de computo básico.

Procesador: Pentium 4 2.8 GHz
Memoria RAM: 512 MB
Tarjeta de Video: ATI Radeon 9800 XT 256MB.
Disco duro. Hitachi 120 GB.

**Tabla 3.1:** Configuración de la computadora en la que se realizo el desarrollo de la interfaz grafica, de mayor demanda de poder de cómputo.

- Instalación y mantenimiento con software libre.  
El sistema operativo para el desarrollo de esta simulación debe incluir:  
Microsoft Windows XP version 5.1 Build 2600  
con los siguientes paquetes de software:  
MathWorks Matlab 6.5®.  
Microsoft Visual C++ 6.0® versión de evaluación.(Obtenida del CD en el Libro Tricks of the Windows Programming Gurus de André LaMothe ) [2]  
Y con las librerías gratuitas:  
Microsoft DirectX 9.0 SDK. [3]

El desarrollo puede realizarse con apoyo de programas propietarios bajo licencia en la UNAM, pero las versiones liberadas solo deberán usar software libre.

- Modelado de una nucleoeléctrica tipo BWR genérica sin referencia específica a central alguna.  
Los modelos deben cumplir con documentos de análisis de los dispositivos, integrarse y validarse, de acuerdo a las leyes de la física y la ingeniería, encontrados en diversos textos. Los dispositivos modelados representarán una configuración propuesta básica, presente en la mayoría de las plantas, y acorde a los sistemas específicos a simular, por lo que la adición, o

adecuación de lo que se propone, con el fin de adecuarlo a una planta específica posteriormente.

- Inicialización con estados estacionarios previamente verificados en forma cualitativa.  
Los parámetros elegidos para la inicialización de los modelos y para el banco de pruebas inicial pueden ser similares a los del documento del Instituto de Investigaciones Eléctricas.[4]
- Control y Monitoreo de simulaciones con gráficos predeterminados: diagramas simplificados de proceso, paneles simplificados, gráficas de tendencias, textos explicativos, fotografías típicas de sistemas y componentes.  
Los gráficos empleados no corresponden a planta alguna en particular. Los gráficos que se presentan, provienen de las fuentes bibliográficas o de Internet, Ponerlas como pie o al final y su organización responde a las necesidades de visualización y claridad.
- Actualizable a diferentes tipos de centrales tipo BWR.  
Todos los sistemas y elementos gráficos del simulador deben producir módulos para que puedan ser modificados y actualizados de manera independiente y otros elementos puedan integrarse sin afectar al resto del simulador. Debe establecerse una estructura jerárquica en que permita hacer prototipos rápidos y cambios independientes, de manera que inclusive fallando uno de los sistemas componentes el resto no se vea afectado, e incluso ayudar a detectar los errores.  
El usuario interactuar con tres bloques llamados capas, la capa gráfica es la que se visualicen los datos que el usuario requiere. La capa de integración para acceder a la capa de ejecución, que es el motor matemático del sistema, el cual realiza operaciones y las devuelve a la parte gráfica por medio de la capa de integración nuevamente. La capa de ejecución, a través de la capa de integración para obtener algunos de los parámetros iniciales y las alteraciones que el usuario haga en tiempo real en la capa de visualización, para actualizar las condiciones del simulador y representar por la misma vía los resultados obtenidos.
- Código fuente único y no liberable para control de versiones.  
En el caso de actualización de los modelos o parámetros para hacerlos representar alguna planta en particular, el control de código y versiones asegura la propiedad intelectual y los procedimientos clasificados que estas plantas pudieran tener, en un marco de competencia o seguridad.

### 3.6. Estructura del simulador de condensado, agua de alimentación y controlador asociado.

Este simulador debe dividirse en módulos que funcionen de manera independiente y estos módulos estructurarse en capas. Cada una de estas capas para proveer una función específica y complementaria. La separación de la lógica de los sistemas y de la interfase, debe permitir realizar mejoras a cada uno de los módulos, sin tener que descomponer y recomponer el sistema completo y permitir una gran portabilidad hacia otros sistemas operativos. En este caso la separación de la ejecución matemática y la visualización gráfica, permitir reformas o mejoras a los modelos matemáticos o actualizaciones a las interfases gráficas, o bien simulaciones en diferentes lenguajes o programas, etc.

- La capa de visualización,

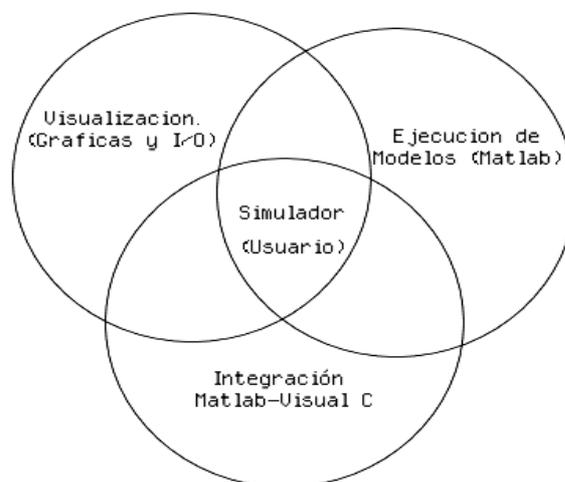
Esta capa provee la interactividad con el usuario final y es su referencia para las diferentes opciones de operación del simulador. En esta capa el usuario puede alterar parámetros, observar alteraciones o tendencias de las variables del simulador, etc. Esta capa esconde el funcionamiento de los demás módulos del simulador y traduce los resultados matemáticos obtenidos en gráficos y esquemas.

- La capa de ejecución.

Esta capa debe ser el motor matemático del simulador donde todos los cálculos sean efectuados de acuerdo con parámetros calculados por el mismo motor, o los proporcionados por el usuario a través de la capa de visualización.

- La capa de integración

Se encarga de acceder las diversas librerías y procesos que proporcionan los programas de la simulación, con el fin de que estos se puedan comunicar de manera confiable, en ambos sentidos, es decir, entregando parámetros o recibiendo instrucciones.



**Figura 3.4:** Capas componentes del simulador y situación de los modelos dentro de la estructura de simulación.

### **3.7. Desarrollo del modelo y controladores**

#### **3.7.1.El Condensador.**

Este dispositivo recibe el fluido de la turbina de baja presión, después de ser usado para la generación de energía, así como las descargas de recirculación del vapor directamente del reactor, a manera de regular la generación, al no mandar ese vapor a las turbinas, sino de retirar vapor del reactor y mandarlo directamente a la condensación. En un modelo sencillo de la planta se considera principalmente como entrada del condensador, el flujo de vapor de bypass, que sirve para evitar el paso por las turbinas y el flujo de las propias turbinas de baja presión que al haber empleado el flujo, lo devuelven al condensador para comenzar el ciclo.

La salida del condensador es la que necesita regresar a la vasija del reactor, para evitar los choques térmicos, y debido a las altas temperaturas que alcanzara el fluido en el reactor, el condensador debe entregar el fluido al siguiente elemento dentro del sistema, los calentadores de baja presión.

El condensador tiene dos funciones principales:

- Condensar y recuperar, el vapor que pasa a través de la turbina (Los condensadores son usados en todas las plantas que usan el vapor como fuerza motora)
- Mantener un vacío para optimizar la eficiencia de la turbina.

Existe generalmente un condensador por cada turbina de baja presión existente en la planta. El condensador esta compuesto por miles de pequeños tubos que están fabricados de metales refinados, cobre, acero "*stainless*", o titanio. El condensador es simplemente un gran intercambiador de calor entre los tubos que están regularmente montados en disposición horizontal, y contiene una estructura que los soporta. Periódicamente, los tubos del condensador deben ser limpiados manualmente, los tubos deben ser verificados de manera no destructiva, para determinar si están siendo usados, en la operación correctamente. Las filtraciones no pueden ser toleradas, debido a que los químicos como el sodio, y cloruros, pueden concentrarse en el reactor en una planta de tipo BWR.

El agua del sistema de recirculación de agua, entra al condensador a través de un contenedor de agua, el nivel del contenedor debe ser mantenido sobre el nivel de los tubos más altos, de otro modo el condensador no operará eficientemente. Generalmente, entre mejor es enfriada el agua circulante, es más eficiente la planta. Las plantas se vuelven menos eficientes cuando los tubos del condensador están dañados, o sucios por acumulación de otras sustancias que les impidan la facultad de transferir el calor del vapor al agua. El vacío en el condensador es mantenido, por una bomba mecánica de vacío, o por un eyector de aire, que succiona los gases, pero no el vapor del condensador. El nivel del contenedor de agua a la entrada del condensador es mantenido por un eyector de aire u otro sistema

que mantenga el nivel de la columna de agua por encima de los tubos, donde se necesita. Del lado del vapor del condensador, el agua condensada es recolectada en un pozo caliente, de donde proveerá el agua para la succión de las bombas de condensado.

### **3.7.2. Calentadores de baja presión.**

Los calentadores de baja presión comúnmente ordenados en forma de cascada, reciben el flujo del condensador y gradualmente lo van calentando de manera tal que conforme va pasando el fluido pueda ser controlado conforme a su temperatura, ya que a la salida de diversos sistemas de la planta como las turbinas o los recalentadores se requerirá que el fluido reingrese en la vasija pero sin que el choque térmico sea crítico, por lo que se le coloca en uno de los calentadores de temperatura afín.

### **3.7.3. Calentadores de alta presión.**

Estos calentadores, normalmente uno por cada bomba de agua de alimentación, reciben el flujo de la turbo-bomba de agua de alimentación, y le proporcionan la temperatura de operación de la vasija. Proporcionan a la vasija la cantidad de calor que se requerirá para la operación del reactor. Son el último paso importante antes de que el flujo de agua de alimentación sea dividido e ingrese en la vasija del reactor.

### **3.7.4. Turbo-bombas de agua de alimentación.**

Las bombas de agua de alimentación reciben el fluido del último de los calentadores de baja presión. Estas bombas se encargan de proporcionar el nivel de presión requerido en la vasija del reactor, y de controlar la cantidad de agua que va a entrar a la vasija, para ajustar el desempeño del reactor. Esta bomba es activada por una pequeña turbina, la cual es accionada por el flujo de vapor del calentador, después de la turbina de alta presión.

### **3.7.5. Controlador del sistema de agua de alimentación.**

Para regular el flujo de agua de alimentación que el sistema manejará en condiciones de estado estable, se desarrolló un controlador basado en las especificaciones de un documento referente al desarrollo de modelos de simulación en plantas nucleares [4], y conforme a las estructuras y lineamientos generales que debe tener este tipo de controladores. Las características que este controlador demuestra responden a las necesidades de estabilidad y convergencia, en los tiempos adecuados a los diferentes tipos de demandas, que las situaciones regulares y de emergencia de la planta pudieran demandar.

El controlador considera en su organización, los flujos que se presentan en diversas partes de la planta como el reactor y las turbinas, este tipo de variables, aunadas a las que el propio sistema de agua de alimentación pudiera proporcionar, conforman las diferentes etapas de control.

El controlador emplea controladores tipo PI, para la adecuación y estabilización de de las señales que recibe, así mismo emplea diversas implementaciones de segundo orden, para controlar los tiempos de estabilidad y amortiguamiento, adecuados para los tiempos propios de los sistemas particulares que pretende alterar el controlador.

Estas consideraciones son muy importantes ya que si no se hiciera de esta manera, los cambios súbitos en los dispositivos mecánicos o térmicos que componen los sistemas de agua de alimentación e incluso otros sistemas, podrían verse afectados notoriamente llegando incluso a la inutilización de los equipos.

El funcionamiento del controlador de agua de alimentación se describe a continuación:

El controlador de agua de alimentación, compara el flujo del agua de alimentación actual, con el que esta circulando por la línea de vapor, esta diferencia o error es ponderada y estabilizada con un controlador PI. El nivel de agua dentro del reactor también es otra variable que entra en el controlador, esta es nivelada y con otro controlador PI es adecuadamente proporcionada con el fin de que represente una variación similar a la que presenta el control PI que se obtuvo de la comparación de los flujos, como se mencionó en el párrafo anterior.

La unión de estas dos señales es estabilizada nuevamente por un controlador PI, que adapta la señal para que responda a los niveles que se necesitan para simular la dinámica de las válvulas de agua de alimentación. La respuesta determinada por la dinámica de las válvulas a la señal generada por los errores, tanto de flujos como de nivel, es la que determina la diferencia que se agregara o sustraerá a el nivel que el usuario de el controlador desea para el agua de alimentación, de manera que los niveles, tanto de flujos, como de nivel, se estabilicen en los valores adecuados, de acuerdo con las necesidades del usuario.

En la búsqueda de un controlador de agua de alimentación se consideraron varias opciones, se decidió por este controlador ya que es fácilmente configurable debido a que muchos de los bloques representan sensores, y compensadores, los modelos pueden afinarse con suma precisión a lo que se encuentra en una planta real.

Los parámetros asignados a los diversos sistemas no corresponden a ninguna planta específica, por lo que responden de manera general, a los diversos transitorios simulados. El posterior refinamiento y asignación de nuevos parámetros puede hacerse de manera sencilla, si se pretende adaptar su funcionamiento a los comportamientos del controlador de alguna planta específica. Una vez más los modelos fueron referenciados al documento del IIE, y sus respectivas salidas comparadas con las respuestas de los controladores encontrados en diversos documentos, los valores encontrados no representan una planta en específico, sino un conjunto de tendencias generales que deben tener este tipo de controladores. La adecuación de estos controladores a la emulación de respuestas reales de

alguna planta, es un proceso que puede realizarse posteriormente si se obtienen los tiempos y parámetros requeridos y si así se requiere.

### **3.8. Especificaciones de la integración del modelado con la Interfase grafica para el condensador, sistema de agua de alimentación y controlador.**

El simulador de agua de alimentación se divide en dos bloques confortantes básicos: La simulación matemática de los modelos implementados como sistemas lineales invariantes en el tiempo, y el desarrollo de una interfaz en lenguaje C, basado en librerías de acceso a capacidades gráficas llamadas DirectX.

La integración de los modelos con la interfaz gráfica, se realizó en un contexto en el cual se da prioridad a la velocidad y a la confiabilidad en los valores de las variables calculadas, contra las observadas en los despliegues de usuario en la interfaz. La traducción del formato de datos numéricos de Matlab hacia el formato de tipos de datos en el lenguaje C, es lo suficientemente preciso, para su representación gráfica y cálculos numéricos requeridos.

La velocidad es determinante pues debe ser tal, que permita la apariencia de continuidad en la ejecución, aun cuando el simulador tarde en entregar los valores hacia la interfaz.

La interfaz recoge los valores del Workspace de Matlab, por los que estos pueden ser almacenados, o reportados en formatos de tabla, arreglo, etc. Matlab también recibe datos de la interfaz desde el Workspace, por lo que el cambio de estado en la simulación se realiza considerando los parámetros puntuales en cada intervalo de tiempo. Esto se explica con más detalle en el capítulo de la interfaz gráfica.

La integración entre los entornos matriciales matemáticos como Matlab y su anexo de simulaciones Simulink, con la programación para Windows en el lenguaje C , es un proceso poco documentado y escaso en el desarrollo de aplicaciones, lo cual se hace presente en las búsquedas de Internet o en los documentos de los diferentes programas ( Matlab y Visual C++ ), La integración empleando tecnologías de gráficos avanzadas DirectX, es un proceso incluso más escaso, por lo que se considera como una especificación contar con un sistema de simulación estable, confiable y numéricamente preciso, que pueda servir como base de futuros desarrollos en ámbitos varios.

### **Referencias Bibliográficas.**

[1] Çengel, Yanus A. y Boles, Michael A., *Termodinámica Segunda Edición*. McGrawHill México, 1994 pp. 524 – 538.

[2] LaMothe, André, *Tricks of the Windows Game Programming Gurus*, Editorial SAMS 2002, Segunda Edición, Indiana E.U.

[3] Microsoft Download Center, *DirectX SDK*  
<http://www.microsoft.com/downloads/>

[4] Schwarzblat Morris, Villalobos Antonio, Morales Sandoval Jaime B., “*Desarrollo de un modelo de simulación de plantas nucleoelectricas tipo BWR*”, Instituto de Investigaciones Eléctricas, Departamento de Energía Nuclear, Cuernavaca, Morelos, Junio 1983.

## Capítulo 4. Modelado matemático del sistema de agua de alimentación.

### 4.1. Principios termodinámicos transferencia de calor.

#### 4.1.1. Intercambiadores de calor de agua de alimentación cerrados.

El tipo de calentador utilizado en las centrales de potencia de vapor que se simula en este proyecto es el calentador de agua de alimentación cerrado. En este calentador el calor se transfiere del vapor extraído al agua de alimentación sin que suceda ninguna mezcla, en este caso, la corriente de vapor extraído de la turbina y la corriente de agua de alimentación, no se mezclan, así que pueden estar a presiones diferentes.

En un calentador de agua de alimentación cerrado ideal, el agua de alimentación se calienta hasta la temperatura de salida del vapor extraído, el cual idealmente abandona el calentador como un líquido saturado a la presión de extracción. En las centrales reales, el agua de alimentación sale del calentador a una temperatura menor a la de salida del vapor extraído debido a que se requiere una diferencia de temperatura de al menos unos cuantos grados para que se logre un transferencia de calor efectiva.

Después, el vapor condensado se bombea a la línea del agua de alimentación o se envía a otro calentador o al condensador mediante un dispositivo denominado trampa. Una trampa permite que el líquido sea estrangulado hasta una región de presión inferior, pero atrapa el vapor. La entalpía del vapor permanece constante durante este proceso de estrangulamiento.

Los calentadores de agua de alimentación cerrados son complejos debido a la red de tuberías internas, por lo que resultan más caros. Los calentadores de agua de alimentación cerrados no requieren una bomba independiente para cada calentador, pues el vapor extraído y el agua de alimentación pueden estar a presiones diferentes.

Las ecuaciones que describen el comportamiento básico de un calentador de agua de agua de alimentación pueden ser explicadas si consideramos los siguientes conceptos termodinámicos.

$$e\dot{m}_{sistema} \pm \sum e\dot{m}_{flujos} = \dot{Q} - \dot{W} \quad (1)$$

que nos dice que de acuerdo a la primera ley de la termodinámica.

Las energías propias de un sistema por unidad de masa por segundo ( $\dot{m}$ ), adicionadas a las sumas de las energías de los flujos es igual a la energía calorífica menos el trabajo generado.

De acuerdo a las siguientes ecuaciones relacionamos la entalpía con el calor del sistema.

$$dU = dQ - dW \quad (2)$$

$$dH = dU + d(PV) \quad (3)$$

Así, si consideramos el trabajo como:

$$dW = d(PV) \quad (4)$$

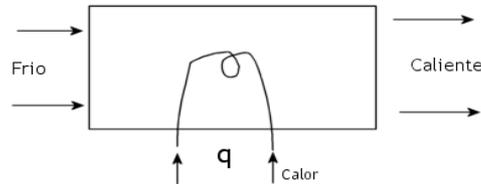
Entonces:

$$dH = dQ - d(PV) + d(PV) \quad (5)$$

reduciendo:

$$dH = dQ \quad (6)$$

Si consideramos un proceso que se de en estas circunstancias, como el siguiente:



**Figura 4.1:** Esquema de un calentador cerrado.

Donde:

$$\dot{m}\Delta h = \dot{q} \quad (7)$$

o también:

$$w\Delta h = \dot{q}, \text{ donde } w \text{ es el flujo.} \quad (8)$$

Así:

$$\frac{\Delta h}{T} = \frac{\dot{q}}{wT} \quad (9)$$

Por lo tanto:

$$\frac{dh}{dt} = \frac{\dot{q}}{wT} \quad (10)$$

Esta ecuación es la que se empleará en el modelado de los calentadores tanto de baja como de alta presión en un modelado posterior.

El condensador observa las mismas características de transferencia de calor, desarrolladas para los calentadores de agua de alimentación. En el simulador el modelo del condensador esta más enfocado a la generación de el flujo de agua de alimentación, que es el detonante de los procesos en el ciclo termodinámico de potencia. [1]

## 4.2. Ecuaciones y Justificaciones.

El diagrama a continuación, muestra la localización del sistema de agua de alimentación dentro de la configuración de la planta. El diagrama nombra, las variables que se calculan en el sistema de agua de alimentación y marca las variables de estado con un asterisco.

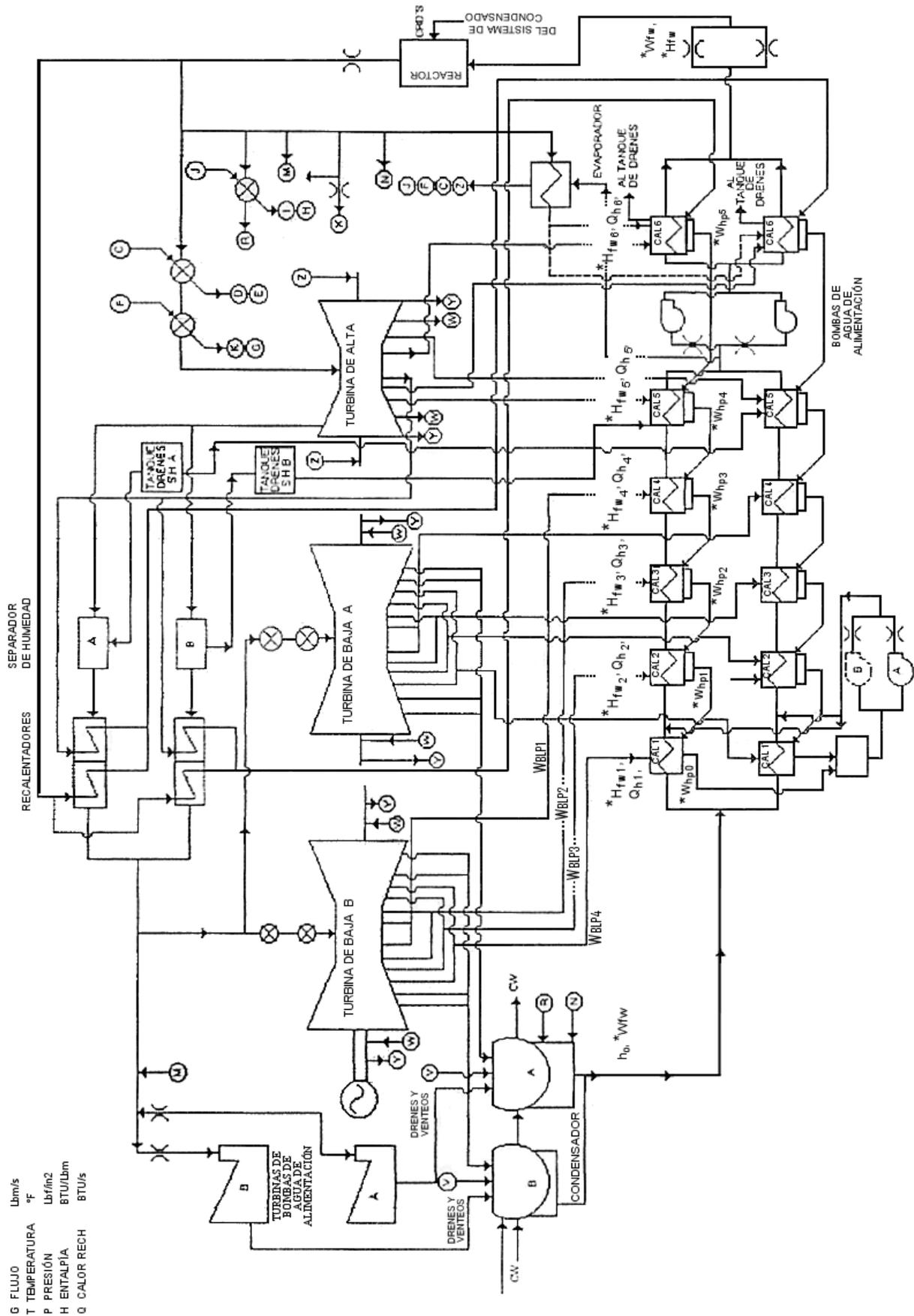


Figura 4.2: Diagrama de localización de las variables en el simulador. Las variables de estado se identifican con un asterisco.

#### 4.2.1. Calentadores de Agua de Alimentación.

Todos los calentadores de agua de alimentación comparten ciertos parámetros, ya que están en una configuración de cascada.

El flujo de agua de alimentación, es el mismo y está representado por  $W_{fw}$ , ya que este flujo es el que entra al reactor, y que proviene de el condensador, por lo que su paso por los calentadores solo le provoca ciertos retrasos, propios de su estancia en el interior de los calentadores, sin embargo a este flujo no se le adiciona ni se le sustrae ningún otro flujo.

Calentador de Agua de Alimentación #1. Del balance de calor en el calentador 1 obtenemos el modelo de la dinámica de:

$$\frac{dh_{fw1}}{dt} = \frac{Q_{H1}}{T_{H1}W_{fw}} + \frac{h_0 - h_{fw1}}{T_{H1}} \quad (11)$$

En donde  $h_{fw1}$  es la entalpía del flujo que sale del primer calentador y va hacia el segundo calentador.

-  $T_{H1}$ , es la constante de tiempo, que representa la estancia del flujo que pasa por el interior del primer calentador (lado tubos), en su camino hacia el reactor.

-  $h_0$ , es la entalpía del condensador, en este caso se considera una constante, debido a que el condensador mantiene una configuración abierta a la transferencia de calor del exterior, por lo que su entalpía dependerá principalmente del ambiente circundante, y de la capacidad de extracción de calor del propio condensador, debido a su área y las torres de enfriamiento del condensado. Es tarea del calentador elevar esta entalpía.

-  $Q_{H1}$  Es el calor transferido por los flujos en el calentador número uno, y está dado por:

$$Q_{H1} = H_{HFW1}(W_{HP1} + W_{BLP4}) \quad (12)$$

En donde  $H_{HFW1}$  es el coeficiente de transferencia de calor en el calentador 1, lo cual nos indica la cantidad de calor que el primer calentador generará de acuerdo a los flujos  $W_{HP1}$  y  $W_{BLP4}$  donde:

- $W_{HP1}$ , es un flujo que proviene de la extracción de una de las turbinas y otros flujos, y que después de ser utilizado por el segundo calentador, se le transfiere al primer calentador, a fin de que emplee el calor que aun posee en el primer calentador.
- $W_{BLP4}$ , es un flujo extraído de la turbina de baja y media presión, y que es la principal fuente de calor para elevar la temperatura del flujo de agua de alimentación. Al ser la cuarta extracción de este tipo en la turbina, el vapor posee menos energía calorífica que en las tres primeras extracciones.

Del balance de masa obtenemos que:

$$\frac{dW_{HP0}}{dt} = \frac{W_{BLP4} + W_{HP1} - W_{HP0}}{T_{H1P}} \quad (13)$$

Donde:

- $W_{HP0}$ , es el flujo que ha pasado por el primer calentador, proveniente de las turbinas de baja y media presión, y del flujo de regreso del segundo calentador y que ahora que ha sido despojado de cierto calor, de acuerdo con el coeficiente de transferencia de calor, mencionado anteriormente, es dirigido hacia el condensador.
- $T_{H1P}$ , es una constante de tiempo de la estancia de estos flujos concurrentes y proveedores de calor, en la periferia del conducto donde fluye el agua de alimentación (Lado Carcaza), estos flujos no se mezclan con el flujo principal de agua de alimentación, la única transferencia es la del calor desprendido por la estancia de estos flujos en el calentador, durante el tiempo referido, y su salida posterior.

Calentador de Agua de Alimentación #2. Análogamente al calentador número uno el calentador dos será presentado por:

$$\frac{dh_{fw2}}{dt} = \frac{Q_{H2}}{T_{H2}W_{fw}} + \frac{h_{fw1} - h_{fw2}}{T_{H2}} \quad (14)$$

En donde  $h_{fw2}$  es la entalpía del flujo que sale del segundo calentador y va hacia el tercero.

- $T_{H2}$ , es la constante de tiempo, que representa el tiempo en el que el flujo pasa por el interior del segundo calentador (lado tubos), en su camino hacia el reactor.
- $h_{fw1}$ , es la entalpía del primer calentador, calculada en el calentador anterior. Es tarea de el segundo calentador elevar esta entalpía, y se realización dependerá de el calor obtenido de las extracciones y otros flujos varios.

-  $Q_{H2}$  es el calor transferido por los flujos en el calentador número dos y esta dado por:

$$Q_{H2} = H_{HFW2}(W_{BLP3} + W_{HP2}) \quad (15)$$

En donde  $H_{HFW2}$  es el coeficiente de transferencia de calor en el calentador 2, lo cual nos indica la cantidad de calor que el segundo calentador generará de acuerdo a los flujos  $W_{HP2}$  y  $W_{BLP3}$  donde:

- $W_{HP2}$ , es un flujo que proviene de la extracción de una de las turbinas y otros flujos, y que después de ser utilizado por el tercer calentador, se le transfiere al segundo calentador, a fin de que emplee el calor remanente.
- $W_{BLP3}$ , es un flujo extraído de la turbina de baja y media presión, y que es la principal fuente de calor para elevar la temperatura del flujo de agua de alimentación. Al ser la tercera extracción de este tipo en la turbina, el vapor posee menos energía calorífica que en las dos primeras extracciones.

El flujo que transfiere calor del calentador número dos  $W_{HP1}$  esta dado por:

$$\frac{dW_{HP1}}{dt} = \frac{W_{BLP3} + W_{HP2} - W_{HP1}}{T_{H2P}} \quad (16)$$

Donde:

- $W_{HP1}$ , es el flujo que ha pasado por el segundo calentador, proveniente de las turbinas de baja y media presión, y del flujo de regreso del tercer calentador y que ahora que ha sido despojado de cierto calor, de acuerdo con el coeficiente de transferencia de calor, mencionado anteriormente, es dirigido hacia el primer calentador.
- $T_{H2P}$ , es una constante de tiempo de la estancia de estos flujos concurrentes y proveedores de calor, en la periferia del conducto donde fluye el agua de alimentación (Lado Carcaza), en el segundo calentador.

Calentador de Agua de Alimentación #3.

Haciendo los balances de masa y de calor en el

calentador número tres obtenemos que:

$$\frac{dh_{fw3}}{dt} = \frac{Q_{H3}}{T_{H3}W_{fw}} + \frac{h_{fw2} - h_{fw3}}{T_{H3}} \quad (17)$$

En donde  $h_{fw3}$  es la entalpía del flujo que sale del tercer calentador y va hacia el cuarto.

- $T_{H3}$ , es la constante de tiempo, que representa el tiempo en el que el flujo pasa por el interior del tercer calentador (lado tubos), en su camino hacia el reactor.
- $h_{fw2}$ , es la entalpía del segundo calentador, calculada anteriormente. Es tarea de el tercer calentador elevar esta entalpía, y se realización dependerá de el calor obtenido de las extracciones y otros flujos varios.

-  $Q_{H3}$  es el calor transferido por los flujos concurrentes en el calentador número tres y esta dado por:

$$Q_{H3} = H_{HFW3} (W_{HP3} + W_{BLP2}) \quad (18)$$

En donde  $H_{HFW3}$  es el coeficiente de transferencia de calor en el calentador 3, lo cual nos indica la cantidad de calor que el tercer calentador generará de acuerdo a los flujos  $W_{HP3}$  y  $W_{BLP2}$  donde:

- $W_{HP3}$ , es un flujo que proviene de la extracción de una de las turbinas y otros flujos, y que después de ser utilizado por el cuarto calentador, se le transfiere al tercer calentador, a fin de que emplee el calor remanente.
- $W_{BLP2}$ , es un flujo extraído de la turbina de baja y media presión, y que es la principal fuente de calor para elevar la temperatura del flujo de agua de alimentación. Al ser la segunda extracción de este tipo en la turbina, el vapor posee menos energía calorífica que en la primera extracción.

El flujo que transfiere calor del calentador número tres  $W_{HP2}$  esta dado por:

$$\frac{dW_{HP2}}{dt} = \frac{W_{BLP2} + W_{HP3} - W_{HP2}}{T_{H3P}} \quad (19)$$

Donde:

- $W_{HP2}$ , es el flujo que ha pasado por el tercer calentador, proveniente de las turbinas de baja y media presión, y del flujo de regreso del cuarto calentador y que ahora que ha sido despojado de cierto calor, de acuerdo con el coeficiente de transferencia de calor, mencionado anteriormente, es dirigido hacia el segundo calentador.
- $T_{H3P}$ , es una constante de tiempo de la estancia de estos flujos concurrentes y proveedores de calor, en la periferia del conducto donde fluye el agua de alimentación (Lado Carcaza), en el tercer calentador.

Calentador de Agua de Alimentación #4. Para el calentador número cuatro obtenemos las siguientes ecuaciones al realizar los balances de calor y masa.

$$\frac{dh_{fw4}}{dt} = \frac{Q_{H4}}{T_{H4}W_{fw}} + \frac{h_{fw3} - h_{fw4}}{T_{H4}} \quad (20)$$

En donde  $h_{fw4}$  es la entalpía del flujo que sale del cuarto calentador y va hacia el quinto.

-  $T_{H4}$ , es la constante de tiempo, que representa el tiempo en el que el flujo pasa por el interior del cuarto calentador (lado tubos).

-  $h_{fw3}$ , es la entalpía del tercer calentador, calculada anteriormente.

-  $Q_{H4}$  es el calor transferido por los flujos concurrentes en el calentador número cuatro y esta dado por:

$$Q_{H4} = H_{HFW4} (W_{HP4} + W_{BLP1}) \quad (21)$$

En donde  $H_{HFW4}$  es el coeficiente de transferencia de calor en el calentador 4 en función de  $W_{HP4}$  y  $W_{BLP1}$  donde:

- $W_{HP4}$ , es un flujo que proviene de la extracción de una de las turbinas y otros flujos, y que después de ser utilizado por el quinto calentador, se le transfiere al cuarto calentador.
- $W_{BLP1}$ , es un flujo extraído de la turbina de baja y media presión. Al ser la primera extracción de este tipo en la turbina, posee la mayor energía calorífica de las extracciones de este tipo de turbinas.

El flujo que transfiere calor del calentador número cuatro  $W_{HP3}$  esta dado por:

$$\frac{dW_{HP3}}{dt} = \frac{W_{BLP1} + W_{HP4} - W_{HP3}}{T_{H4P}} \quad (22)$$

Donde:

- $W_{HP3}$ , es el flujo que ha pasado por el cuarto calentador, proveniente de las turbinas de baja y media presión, y del flujo de regreso del quinto calentador y que ahora es dirigido hacia el tercer calentador.
- $T_{H4P}$ , es una constante de tiempo de la estancia de estos flujos en la carcaza del cuarto calentador.

Calentador de Agua de Alimentación #5.

Haciendo los balances de calor y de masa para el

calentador cinco obtenemos:

$$\frac{dh_{fw5}}{dt} = \frac{Q_{H5}}{T_{H5}W_{fw}} + \frac{h_{fw4} - h_{fw5}}{T_{H5}} \quad (23)$$

En donde  $h_{fw5}$  es la entalpía del flujo que sale del quinto calentador y va hacia el sexto.

- $T_{H5}$ , es la constante de tiempo, que representa el tiempo en el que el flujo pasa por el interior del quinto calentador (lado tubos).
- $h_{fw4}$ , es la entalpía del cuarto calentador, calculada anteriormente.
- $Q_{H5}$  es el calor transferido por los flujos concurrentes en el calentador número cinco y esta dado por:

$$Q_{H5} = H_{HFW5} (W_{HP5} + W_{BHP2} + W_{MS}) \quad (24)$$

En donde  $H_{HFW5}$  es el coeficiente de transferencia de calor en el calentador 5 en función de  $W_{HP5}$ ,  $W_{BHP2}$  y  $W_{MS}$  donde:

- $W_{HP5}$ , es un flujo que proviene de la extracción de una de las turbinas y otros flujos, y que después de ser utilizado por el sexto calentador, se le transfiere al quinto calentador.
- $W_{BHP2}$ , es un flujo extraído de la turbina de alta presión. Al ser la segunda extracción de este tipo en la turbina de alta presión, posee menor energía que la primera extracción en la turbina de alta presión, pero mayor que las extracciones de las turbinas de baja presión.
- $W_{MS}$ , es el flujo extraído del separador de vapor, como producto de la condensación en este dispositivo, y que por su energía aunque poca cantidad, es usado en el calentador.

El flujo que transfiere calor del calentador número cinco  $W_{HP4}$  esta dado por:

$$\frac{dW_{HP4}}{dt} = \frac{W_{HP5} + W_{BHP2} + W_{MS} - W_{HP4}}{T_{H5P}} \quad (25)$$

Donde:

- $W_{HP4}$ , es el flujo que ha pasado por el quinto calentador, proveniente de las turbinas de alta presión, el flujo del separador de vapor y del flujo de regreso del sexto calentador y que ahora es dirigido hacia el cuarto calentador.
- $T_{H5P}$ , es una constante de tiempo de la estancia de estos flujos en la carcasa del quinto calentador.

#### Calentador de Agua de Alimentación #6.

De igual forma que para los demás calentadores, para el calentador número seis obtenemos:

$$\frac{dh_{fw6}}{dt} = \frac{Q_{H6}}{T_{H6} W_{fw}} + \frac{h_{fw5} - h_{fw}}{T_{H6}} \quad (26)$$

En donde  $h_{fw6}$  es la entalpía del flujo que sale del sexto calentador y va hacia el reactor.

-  $T_{H6}$ , es la constante de tiempo, que representa el tiempo en el que el flujo pasa por el interior del sexto calentador (lado tubos).

-  $h_{fw5}$ , es la entalpía del quinto calentador, calculada anteriormente.

-  $Q_{H6}$  es el calor transferido por los flujos concurrentes en el calentador número seis y esta dado por:

$$Q_{H6} = H_{HFW6} (W_{PR1} + W_{BHP1}) \quad (27)$$

En donde  $H_{HFW6}$  es el coeficiente de transferencia de calor en el calentador 6 en función de  $W_{PR1}$  y  $W_{BHP1}$  donde:

- $W_{PR1}$ , es un flujo que proviene de una extracción en el recalentador, que se encuentra entre las turbinas de alta y de baja/media presión.
- $W_{BHP1}$ , es un flujo extraído de la turbina de alta presión. Al ser la primera extracción de este tipo en la turbina de alta presión, posee la mayor energía, proporcionando un flujo de calor importante, para que el flujo de Agua de Alimentación adquiriera la entalpía necesaria a la entrada del Reactor.

El flujo que transfiere calor del calentador número cinco  $W_{HP5}$  esta dado por:

$$\frac{dW_{HP5}}{dt} = \frac{W_{PR1} + W_{BHP1} - W_{HP5}}{T_{H6P}} \quad (28)$$

Donde:

- $W_{HP5}$ , es el flujo que ha pasado por el sexto calentador, proveniente de las turbinas de alta presión y del flujo del recalentador y que ahora es dirigido hacia el quinto calentador.
- $T_{H6P}$ , es una constante de tiempo de la estancia de estos flujos en la carcaza del sexto calentador.

La entalpía  $h_{fw6}$ , también es conocida como  $h_{Fw}$  debido, a que la entalpía del flujo que sale de este calentador es la que entra directamente al reactor. Por lo que se considera la entalpía del Agua de Alimentación.

#### 4.2.2. Condensador.

A partir del balance de masa en el condensador podemos modelarlo a través de la siguiente ecuación:

Modelado matemático del sistema de agua de alimentación.

$$\frac{dW_{con}}{dt} = \frac{W'_3 + W_{BT} + W_{BOMB} + W_{HP0} + W_{ALIV} - W_{fw}}{T_{CON}} \quad (29)$$

donde:

- $W_{con}$ , es el flujo interno del condensador
- $W'_3$ , es un flujo proveniente de la salida de la turbina de baja presión.
- $W_{BT}$ , es el flujo de vapor de la válvula de bypass, en la línea de vapor.
- $W_{BOMB}$ , es un flujo de vapor de la bomba de condensado.
- $W_{HP0}$ , es el flujo de vapor secundario del primer calentador.
- $W_{ALIV}$ , es un flujo de vapor para accesorios, primordialmente en la línea de vapor.
- $W_{fw}$ , es el flujo de agua de alimentación requerido por el reactor y cuya entalpía es modificada por el tren de calentadores de baja y alta presión.
- $T_{CON}$ , es el tiempo en que estos flujos tardan en fluir en el condensador.

#### 4.2.3. Flujo de Agua de Alimentación.

El modelo del flujo de agua de alimentación esta determinado por la operación de la bomba de agua de alimentación.

$$\frac{dW_{fw}}{dt} = \frac{1}{\tau_{FW}} (W_{FSP} - W_{fw}) \quad (30)$$

Con la Restricción.

$$0.0 \leq W_{fw} \leq 1.15W_{FR}$$

Donde:

- $W_{fw}$ , es el flujo de agua de alimentación.
- $\tau_{FW}$ , es la constante de tiempo de estabilidad de la bomba.
- $W_{FSP}$ , es el valor de Set Point del flujo, es decir el valor que actualiza el flujo de agua de alimentación.

Las ecuaciones de los modelos de los calentadores de agua de alimentación, del condensador, y de la bomba de agua de alimentación, fueron analizadas desde el punto de vista de los Sistemas Lineales Invariantes en el Tiempo, por lo que se realizaron transformaciones de Laplace para llevar los modelos en ecuaciones diferenciales, a funciones de transferencia, de manera que el enfoque de

sistemas que programas de computo como Matlab, y Simulink, pudieran incorporar, integrar y simular con facilidad.

Las señales de entrada de los modelos, se integraron junto con la ecuación de transferencia del modelo en una representación gráfica de Matlab, con el fin de integrar todos los elementos en una simulación coherente y estructurada conforme a la descripción de los modelos, esto con el fin de claridad y organización, y también para hacer factible una depuración eficaz.

Los modelos empleados y sus representaciones se listan a continuación.

### 4.3. Preparación de las ecuaciones para el modelado.

#### 4.3.1. Calentador 1:

Ecuación del Modelo. (En el dominio del tiempo)

Ecuación de entalpía en el primer calentador:

$$\frac{dh_{fw1}}{dt} = \frac{Q_{H1}}{T_{H1} w_{fw}} + \frac{h_0 - h_{fw1}}{T_{H1}} \quad (31)$$

Función de Transferencia (En el dominio de “s” (Laplace) )

$$H(s) = \frac{1}{T_{H1}s + 1}$$

$$Y(s) = h_{fw1}(s) \quad (32)$$

$$X(s) = \frac{Q_{H1}(s)}{W_{fw}(s)} + h_0$$

Parámetros. El número que sigue al parámetro, es la referencia en el índice posterior.

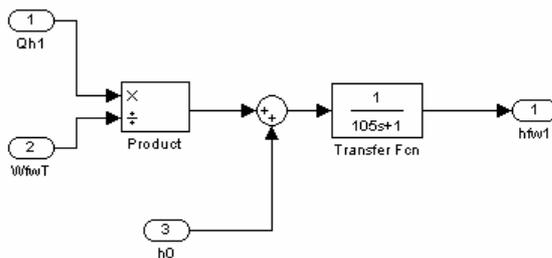
$$T_{H1} = 105$$

PARÁMETRO: {1}

$$h_0 = 201.199$$

PARÁMETRO: {2}

Implementación en MATLAB®.



Modelado matemático del sistema de agua de alimentación.

Ecuación del Modelo. (En el dominio del tiempo):

Ecuación del calor en el primer calentador:

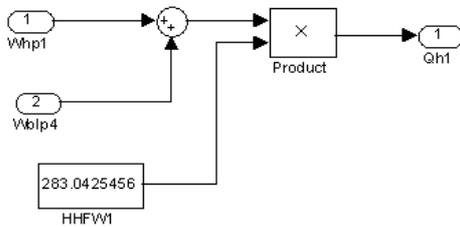
$$Q_{H1} = H_{HFW1} (W_{HP1} + W_{BLP4}) \quad (33)$$

Parámetros

$$H_{HFW1} = 283042.5456$$

PARÁMETRO: {3}

Implementación en MATLAB®.



Ecuación del Modelo. (En el dominio del tiempo)

Ecuación del flujo lado carcasa en el primer calentador:

$$\frac{dW_{HP0}}{dt} = \frac{W_{BLP4} + W_{HP1} - W_{HP0}}{T_{H1P}} \quad (34)$$

Función de Transferencia (En el dominio de “s” (Laplace) )

$$H(s) = \frac{1}{T_{H1P}s + 1}$$

$$Y(s) = W_{HP0}(s) \quad (35)$$

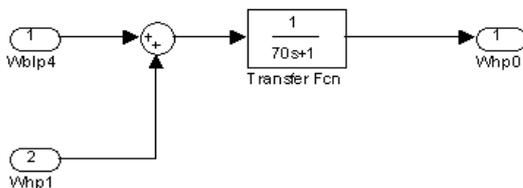
$$X(s) = W_{BLP4}(s) + W_{HP1}(s)$$

Parámetros.

$$T_{H1P} = 70$$

PARÁMETRO: {4}

Implementación en MATLAB®.



### 4.3.2. Calentador 2:

Ecuación del Modelo. (En el dominio del tiempo)

Ecuación de entalpía en el segundo calentador:

$$\frac{dh_{fw2}}{dt} = \frac{Q_{H2}}{T_{H2}W_{fw}} + \frac{h_{fw1} - h_{fw2}}{T_{H2}} \quad (36)$$

Función de Transferencia (En el dominio de “s” (Laplace) )

$$H(s) = \frac{1}{T_{H2}s + 1} \quad (37)$$

$$Y(s) = h_{fw2}(s)$$

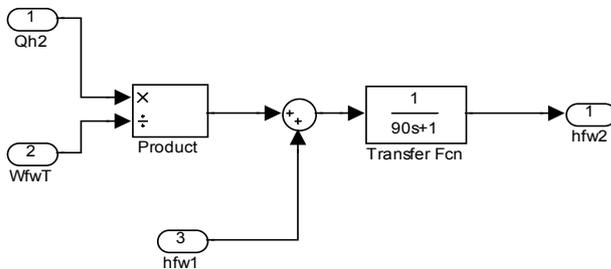
$$X(s) = \frac{Q_{H2}(s)}{W_{fw}(s)} + h_{fw1}(s)$$

Parámetros.

$$T_{H2} = 90$$

PARÁMETRO: {5}

Implementación en MATLAB®.



Ecuación del Modelo. (En el dominio del tiempo)

Ecuación del calor en el segundo calentador:

$$Q_{H2} = H_{HFW2}(W_{BLP3} + W_{HP2}) \quad (38)$$

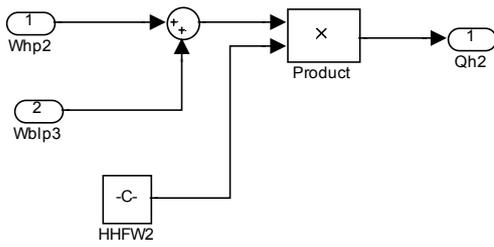
Parámetros.

$$H_{HFW2} = 205671.2273$$

PARÁMETRO: {6}

Modelado matemático del sistema de agua de alimentación.

Implementación en MATLAB®.



Ecuación del Modelo. (En el dominio del tiempo):

Ecuación del flujo lado carcasa en el segundo calentador:

$$\frac{dW_{HP1}}{dt} = \frac{W_{BLP3} + W_{HP2} - W_{HP1}}{T_{H2P}} \quad (39)$$

Función de Transferencia (En el dominio de “s” (Laplace) )

$$H(s) = \frac{1}{T_{H2P}s + 1} \quad (40)$$

$$Y(s) = W_{HP1}(s)$$

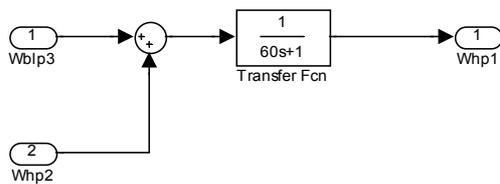
$$X(s) = W_{BLP3}(s) + W_{HP1}(s)$$

Parámetros.

$$T_{H2P} = 60$$

PARÁMETRO: {7}

Implementación en MATLAB®.



### 4.3.3. Calentador 3:

Ecuación del Modelo. (En el dominio del tiempo)

Ecuación de entalpía en el tercer calentador:

$$\frac{dh_{fw3}}{dt} = \frac{Q_{H3}}{T_{H3}W_{fw}} + \frac{h_{fw2} - h_{fw3}}{T_{H3}} \quad (41)$$

Función de Transferencia (En el dominio de “s” (Laplace) )

$$H(s) = \frac{1}{T_{H3}s + 1}$$

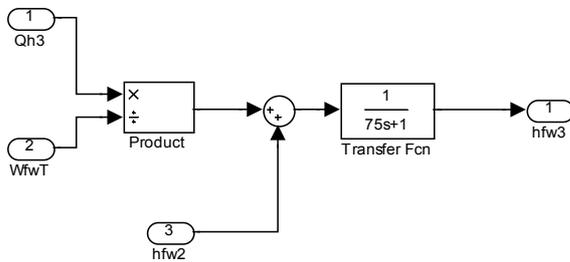
$$Y(s) = h_{fw3}(s) \tag{41}$$

$$X(s) = \frac{Q_{H3}(s)}{W_{fw}(s)} + h_{fw2}(s)$$

Parámetros.

$$T_{H3} = 75 \tag{PARÁMETRO: \{8\}}$$

Implementación en MATLAB®.



Ecuación del Modelo. (En el dominio del tiempo)

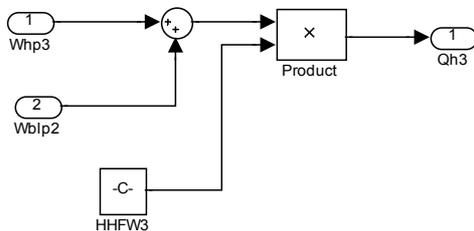
Ecuación del calor en el tercer calentador:

$$Q_{H3} = H_{HFW3}(W_{HP3} + W_{BLP2}) \tag{42}$$

Parámetros.

$$H_{HFW3} = 302412.4739 \tag{PARÁMETRO: \{9\}}$$

Implementación en MATLAB®.



Ecuación del Modelo. (En el dominio del tiempo)

Ecuación del flujo lado carcasa en el tercer calentador:

$$\frac{dW_{HP2}}{dt} = \frac{W_{BLP2} + W_{HP3} - W_{HP2}}{T_{H3P}} \tag{43}$$

Modelado matemático del sistema de agua de alimentación.

Función de Transferencia (En el dominio de “s” (Laplace) )

$$H(s) = \frac{1}{T_{H3P}s + 1}$$

$$Y(s) = W_{HP2}(s)$$

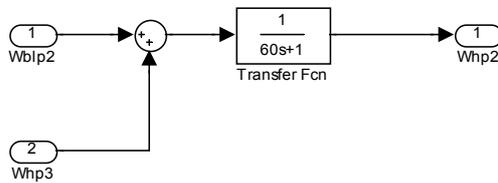
$$X(s) = W_{BLP2}(s) + W_{HP3}(s)$$
(44)

Parámetros.

$$T_{H3P} = 60$$

PARÁMETRO: {10}

Implementación en MATLAB®.



#### 4.3.4. Calentador 4:

Ecuación del Modelo. (En el dominio del tiempo)

Ecuación de entalpía en el cuarto calentador:

$$\frac{dh_{fw4}}{dt} = \frac{Q_{H4}}{T_{H4} w_{fw}} + \frac{h_{fw3} - h_{fw4}}{T_{H4}}$$
(45)

Función de Transferencia (En el dominio de “s” (Laplace) )

$$H(s) = \frac{1}{T_{H4}s + 1}$$

$$Y(s) = h_{fw4}(s)$$

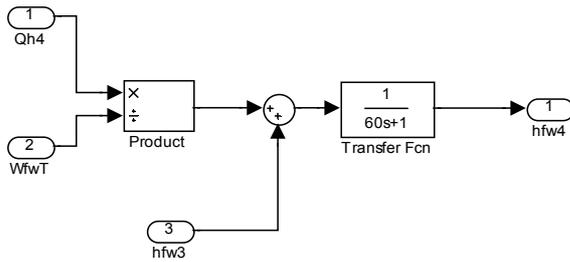
$$X(s) = \frac{Q_{H4}(s)}{W_{fw}(s)} + h_{fw3}(s)$$
(46)

Parámetros.

$$T_{H4} = 60$$

PARÁMETRO: {11}

Implementación en MATLAB®.



Ecuación del Modelo. (En el dominio del tiempo)

Ecuación del calor en el cuarto calentador:

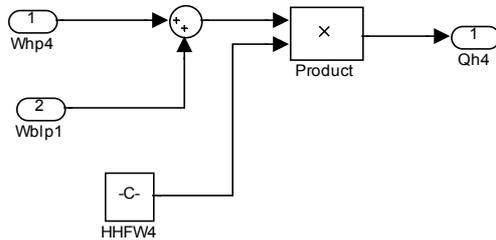
$$Q_{H4} = H_{HFW4} (W_{HP4} + W_{BLP1}) \tag{47}$$

Parámetros.

$$H_{HFW4} = 367169.1568$$

PARÁMETRO: {12}

Implementación en MATLAB®.



Ecuación del Modelo. (En el dominio del tiempo)

Ecuación del flujo lado carcasa en el cuarto calentador:

$$\frac{dW_{HP3}}{dt} = \frac{W_{BLP1} + W_{HP4} - W_{HP3}}{T_{H4P}} \tag{48}$$

Función de Transferencia (En el dominio de “s” (Laplace) )

$$H(s) = \frac{1}{T_{H4P}s + 1}$$

$$Y(s) = W_{HP3}(s) \tag{49}$$

$$X(s) = W_{BLP1}(s) + W_{HP4}(s)$$

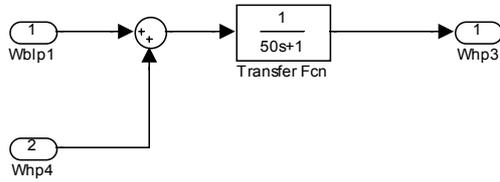
Modelado matemático del sistema de agua de alimentación.

Parámetros.

$$T_{H4P} = 50$$

PARÁMETRO: {13}

Implementación en MATLAB®.



#### 4.3.5. Calentador 5:

Ecuación del Modelo. (En el dominio del tiempo)

Ecuación de entalpía en el quinto calentador:

$$\frac{dh_{fw5}}{dt} = \frac{Q_{H5}}{T_{H5}W_{fw}} + \frac{h_{fw4} - h_{fw5}}{T_{H5}} \quad (50)$$

Función de Transferencia (En el dominio de “s” (Laplace))

$$H(s) = \frac{1}{T_{H5}s + 1}$$

$$Y(s) = h_{fw5}(s)$$

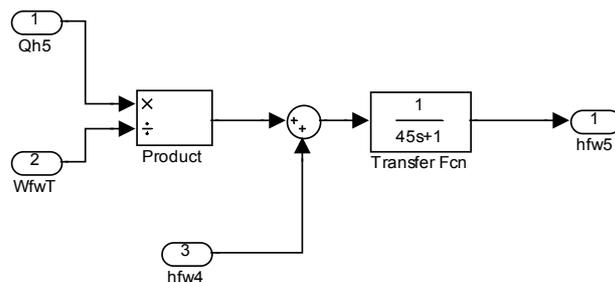
$$X(s) = \frac{Q_{H5}(s)}{W_{fw}(s)} + h_{fw4}(s) \quad (51)$$

Parámetros.

$$T_{H5} = 45$$

PARÁMETRO: {14}

Implementación en MATLAB®.



Ecuación del Modelo. (En el dominio del tiempo)

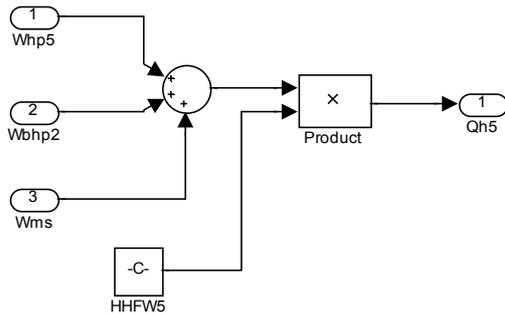
Ecuación del calor en el quinto calentador:

$$Q_{H5} = H_{HFW5}(W_{HP5} + W_{BHP2} + W_{MS}) \quad (52)$$

Parámetros.

$$H_{HFW5} = 663106.9585 \quad \text{PARÁMETRO: \{15\}}$$

Implementación en MATLAB®.



Ecuación del Modelo. (En el dominio del tiempo)

Ecuación del flujo lado carcasa en el quinto calentador:

$$\frac{dW_{HP4}}{dt} = \frac{W_{HP5} + W_{BHP2} + W_{MS} - W_{HP4}}{T_{H5P}} \quad (53)$$

Función de Transferencia (En el dominio de “s” (Laplace) )

$$H(s) = \frac{1}{T_{H5P}s + 1} \quad (54)$$

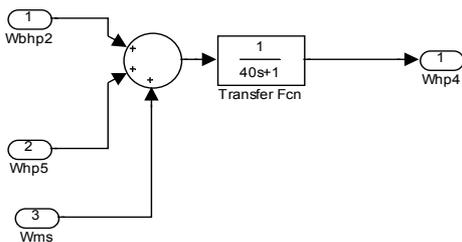
$$Y(s) = W_{HP4}(s)$$

$$X(s) = W_{BHP2}(s) + W_{HP5}(s) + W_{MS}(s)$$

Parámetros.

$$T_{H5P} = 40 \quad \text{PARÁMETRO: \{16\}}$$

Implementación en MATLAB®.



#### 4.3.6. Calentador 6:

Ecuación del Modelo. (En el dominio del tiempo)

Ecuación de entalpía en el sexto calentador:

$$\frac{dh_{fw6}}{dt} = \frac{Q_{H6}}{T_{H6}W_{fw}} + \frac{h_{fw5} - h_{fw}}{T_{H6}} \quad (55)$$

Función de Transferencia (En el dominio de “s” (Laplace))

$$H(s) = \frac{1}{T_{H6}s + 1}$$

$$Y(s) = h_{fw6}(s)$$

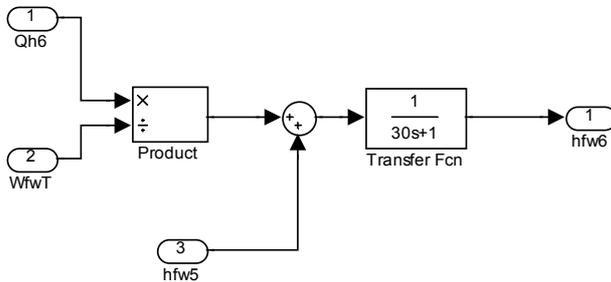
$$X(s) = \frac{Q_{H6}(s)}{W_{fw}(s)} + h_{fw5}(s) \quad (56)$$

Parámetros.

$$T_{H6} = 30$$

PARÁMETRO: {17}

Implementación en MATLAB®.



Ecuación del Modelo. (En el dominio del tiempo)

Ecuación del calor en el sexto calentador:

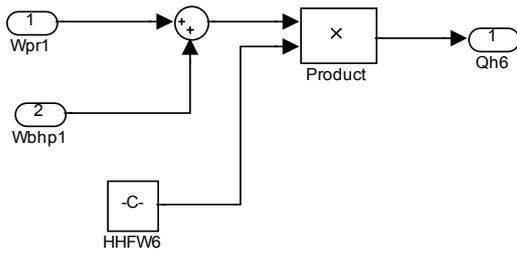
$$Q_{H6} = H_{HFW6}(W_{PR1} + W_{BHP1}) \quad (57)$$

Parámetros.

$$H_{HFW6} = 403509.6205$$

PARÁMETRO: {18}

Implementación en MATLAB®.



Ecuación del Modelo. (En el dominio del tiempo)

Ecuación del flujo lado carcasa en el sexto calentador:

$$\frac{dW_{HP5}}{dt} = \frac{W_{PR1} + W_{BHP1} - W_{HP5}}{T_{H6P}} \tag{58}$$

Función de Transferencia (En el dominio de “s” (Laplace) )

$$H(s) = \frac{1}{T_{H6P}s + 1} \tag{59}$$

$$Y(s) = W_{HP5}(s)$$

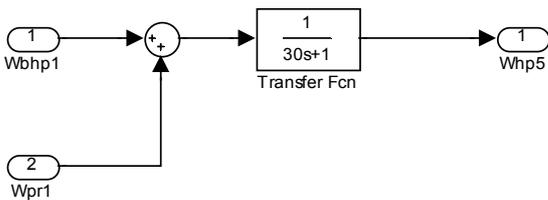
$$X(s) = W_{BHP1}(s) + W_{PR1}(s)$$

Parámetros.

$$T_{H6P} = 30$$

PARÁMETRO: {19}

Implementación en MATLAB®.



**4.3.7. Condensador.**

Ecuación del Modelo. (En el dominio del tiempo)

Ecuación del flujo dentro del condensador:

$$\frac{dW_{con}}{dt} = \frac{W'_3 + W_{BT} + W_{BOMB} + W_{HP0} + W_{ALIV} - W_{fw}}{T_{CON}} \tag{60}$$

Modelado matemático del sistema de agua de alimentación.

Función de Transferencia (En el dominio de “s” (Laplace )

$$H(s) = \frac{1}{T_{CON}s} \tag{61}$$

$$Y(s) = W_{con}(s)$$

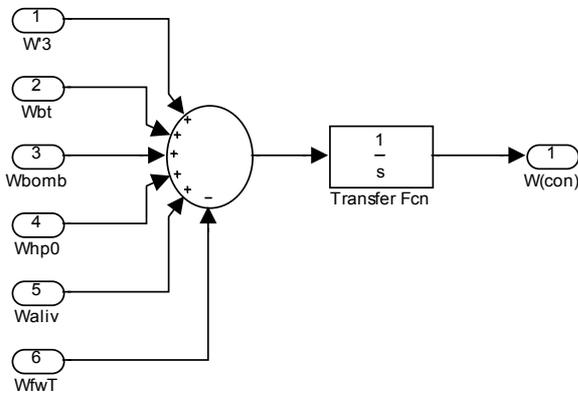
$$X(s) = W'_3(s) + W_{BT}(s) + W_{BOMB}(s) + W_{HP0}(s) + W_{ALIV}(s) - W_{fw}(s)$$

Parámetros.

$$T_{CON} = 70.$$

PARÁMETRO: {20}

Implementación en MATLAB®.



#### 4.3.8. Bomba de Agua de Alimentación.

Ecuación del Modelo. (En el dominio del tiempo)

Ecuación del flujo de agua de alimentación entregado por la bomba:

$$\frac{dW_{fw}}{dt} = \frac{1}{\tau_{FW}} (W_{FSP} - W_{fw}) \tag{62}$$

Función de Transferencia (En el dominio de “s” (Laplace))

$$H(s) = \frac{1}{\tau_{FW}s + 1} \tag{63}$$

$$Y(s) = W_{fw}(s)$$

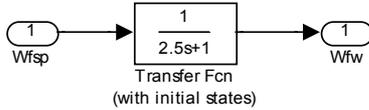
$$X(s) = W_{FSP}(s)$$

Parámetros.

$$\tau_{FW} = 2.5$$

PARÁMETRO: {21}

Implementación en MATLAB®.



#### 4.4. Condiciones de frontera, variables de estado, cantidades medibles

##### 4.4.1. Variables de estado:

$\frac{dh_{fw1}}{dt}$  : En donde  $h_{fw1}$  es la entalpía del flujo que sale del primer calentador y va hacia el segundo calentador.

$\frac{dh_{fw2}}{dt}$  : En donde  $h_{fw2}$  es la entalpía del flujo que sale del segundo calentador y va hacia el tercero.

$\frac{dh_{fw3}}{dt}$  : En donde  $h_{fw3}$  es la entalpía del flujo que sale del tercer calentador y va hacia el cuarto.

$\frac{dh_{fw4}}{dt}$  : En donde  $h_{fw4}$  es la entalpía del flujo que sale del cuarto calentador y va hacia el quinto.

$\frac{dh_{fw5}}{dt}$  : En donde  $h_{fw5}$  es la entalpía del flujo que sale del quinto calentador y va hacia el sexto.

$\frac{dh_{fw6}}{dt}$  : En donde  $h_{fw6}$  es la entalpía del flujo que sale del sexto calentador y va hacia rector.

$\frac{dW_{HP0}}{dt}$  :  $W_{HP0}$ , es el flujo que ha pasado por el primer calentador, proveniente de las turbinas y va al condensador

$\frac{dW_{HP1}}{dt}$  :  $W_{HP1}$ , es el flujo que ha pasado por el segundo calentador, proveniente de las turbinas y va al primer calentador.

$\frac{dW_{HP2}}{dt}$  :  $W_{HP2}$ , es el flujo que ha pasado por el tercer calentador, proveniente de las turbinas y va al segundo calentador.

Modelado matemático del sistema de agua de alimentación.

$\frac{dW_{HP3}}{dt}$  :  $W_{HP3}$ , es el flujo que ha pasado por el cuarto calentador, proveniente de las turbinas y va al tercer calentador.

$\frac{dW_{HP4}}{dt}$  :  $W_{HP4}$ , es el flujo que ha pasado por el quinto calentador, proveniente de las turbinas y va al cuarto calentador.

$\frac{dW_{HP5}}{dt}$  :  $W_{HP5}$ , es el flujo que ha pasado por el sexto calentador, proveniente de las turbinas y va al quinto calentador.

#### 4.4.2.Estados Iniciales.

Estado Inicial: Bomba de Agua de Alimentación = 1046.160407 kg/s

#### 4.5. Entradas.

Para hacer las pruebas del sistema de agua de alimentación y controlador se inventario de refrigerante, se emplearon valores de entrada a ciertas variables que en el caso de la planta acoplada, estos serían calculados por el sistema de turbinas y recalentador. Posteriormente se espera la integración con este modelo, y con el reactor para obtener el sistema completo de la planta nuclear.

Los valores que se eligieron se obtuvieron de una planta nucleoelectrica típica corresponden a los flujos de extracción de las turbinas de alta y baja presión.

Los valores son:

$W_{blp1}$  (primera extracción de la turbina de baja presión) = 48.5344 kg/s

$W_{blp2}$  (segunda extracción de la turbina de baja presión) = 42.1841 kg/s

$W_{blp3}$  (tercera extracción de la turbina de baja presión) = 30.2546 kg/s

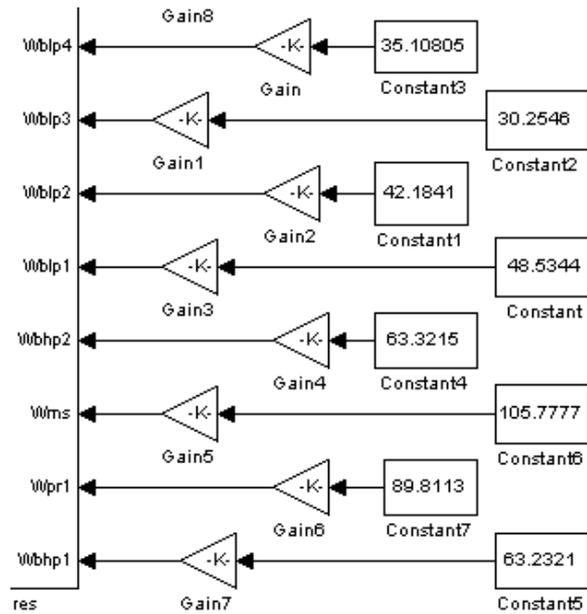
$W_{blp4}$  (cuarta extracción de la turbina de baja presión) = 35.10805 kg/s

$W_{bhp1}$  (primera extracción de la turbina de alta presión) = 63.2321 kg/s

$W_{bhp2}$  (segunda extracción de la turbina de alta presión) = 63.3215 kg/s

$W_{ms}$  (extracción del separador de vapor) = 105.7777 kg/s

$W_{pr1}$  (extracción del recalentador) = 89.8113 kg/s



**Figura 4.3:** Esquema en Simulink de las entradas definidas para la simulación independiente del sistema de agua de alimentación.

La interfaz grafica es la encargada de inicializar las otras entradas al sistema estas son variables por el usuario pero el sistema considera valores iniciales similares a los que existirían en una planta real.

Estos valores son inicializados a:

*Level Setpoint* (Nivel deseado de agua de alimentación en la vasija del reactor, valor del controlador) = 0.97 m

*mIso Steamflow* (Flujo de agua de alimentación deseado, valor del controlador) = 1058 kg/s

#### 4.6. Parámetros.

Lista de parámetros y los valores considerados, así como la fuente de la cual se obtuvo el parámetro para su localización.

IIE DMSPNB: Documento “Desarrollo de un modelo de simulación de plantas nucleoelectricas tipo BWR” [2]

{1}  $T_{H1}$ : Constante de tiempo asociada al calentamiento del vapor en el calentador 1.  
 $T_{H1} = 105$  s. (IIE DMSPNB, Apéndice B pag 154.)

{2} Entalpía a la salida del condensador.  
 $h_0 = 201199.4$  J/kg (IIE DMSPNB, Apéndice B pag 153.)

{3}  $H_{HFW1}$ : Coeficiente de transferencia de calor para el calentador 1.  
 $H_{HFW1} = 283042.5456$  J/(kg·K) (IIE DMSPNB, Apéndice B pag 153.)

- {4}  $T_{H1P}$ : Constante de tiempo asociada al flujo de vapor en el calentador 1.  
 $T_{H1P} = 70$  s. (IIE DMSPNB, Apéndice B pag 154.)
- {5}  $T_{H2}$ : Constante de tiempo asociada al calentamiento del vapor en el calentador 2.  
 $T_{H2} = 90$  s. (IIE DMSPNB, Apéndice B pag 154.)
- {6}  $H_{HFW2}$ : Coeficiente de transferencia de calor para el calentador 2.  
 $H_{HFW2} = 205671.2273$  J/(kg·K) (IIE DMSPNB, Apéndice B pag 153.)
- {7}  $T_{H2P}$ : Constante de tiempo asociada al flujo de vapor en el calentador 2.  
 $T_{H2P} = 60$  s. (IIE DMSPNB, Apéndice B pag 154.)
- {8}  $T_{H3}$ : Constante de tiempo asociada al calentamiento del vapor en el calentador 3.  
 $T_{H3} = 75$  s. (IIE DMSPNB, Apéndice B pag 154.)
- {9}  $H_{HFW3}$ : Coeficiente de transferencia de calor para el calentador 3.  
 $H_{HFW3} = 302412.4739$  J/(kg·K) (IIE DMSPNB, Apéndice B pag 153.)
- {10}  $T_{H3P}$ : Constante de tiempo asociada al flujo de vapor en el calentador 3.  
 $T_{H3P} = 60$  s. (IIE DMSPNB, Apéndice B pag 154.)
- {11}  $T_{H4}$ : Constante de tiempo asociada al calentamiento del vapor en el calentador 4.  
 $T_{H4} = 60$  s. (IIE DMSPNB, Apéndice B pag 154.)
- {12}  $H_{HFW4}$ : Coeficiente de transferencia de calor para el calentador 4.  
 $H_{HFW4} = 367169.1568$  J/(kg·K) (IIE DMSPNB, Apéndice B pag 153.)
- {13}  $T_{H4P}$ : Constante de tiempo asociada al flujo de vapor en el calentador 4.  
 $T_{H4P} = 50$  s. (IIE DMSPNB, Apéndice B pag 154.)
- {14}  $T_{H5}$ : Constante de tiempo asociada al calentamiento del vapor en el calentador 5.  
 $T_{H5} = 45$  s. (IIE DMSPNB, Apéndice B pag 154.)
- {15}  $H_{HFW5}$ : Coeficiente de transferencia de calor para el calentador 5.  
 $H_{HFW5} = 663106.9585$  J/(kg·K) (IIE DMSPNB, Apéndice B pag 153.)
- {16}  $T_{H5P}$ : Constante de tiempo asociada al flujo de vapor en el calentador 5.  
 $T_{H5P} = 40$  s. (IIE DMSPNB, Apéndice B pag 154.)
- {17}  $T_{H6}$ : Constante de tiempo asociada al calentamiento del vapor en el calentador 6.  
 $T_{H6} = 30$  s. (IIE DMSPNB, Apéndice B pag 154.)
- {18}  $H_{HFW6}$ : Coeficiente de transferencia de calor para el calentador 6.  
 $H_{HFW6} = 403509.6205$  J/(kg·K) (IIE DMSPNB, Apéndice B pag 153.)
- {19}  $T_{H6P}$ : Constante de tiempo asociada al flujo de vapor en el calentador 6.  
 $T_{H6P} = 30$  s. (IIE DMSPNB, Apéndice B pag 154.)
- {20}  $T_{CON}$ : Constante de tiempo, en el condensador.  
 $T_{CON} = 70$  s (IIE DMSPNB, Apéndice B pag 141.)
- {21}  $\tau_{FW}$ : Constante de tiempo en la bomba de agua de alimentación.  
 $\tau_{FW} = 2.5$  s (IIE DMSPNB, Apéndice B pag 150.)

#### **4.7. Conclusiones.**

Dentro de la estructura del simulador el modelado es el alma que provee los valores necesarios para la representación gráfica y en su conjunto trata de borrar la línea entre la simulación y la realidad. Los parámetros de ciertos modelos de los dispositivos y del controlador de refrigerante, fueron un tanto difíciles de obtener, debido a que estos parámetros son propiedad de las diferentes plantas y empresas de equipo nuclear, por lo que se estimaron varios de estos parámetros de acuerdo a los balances de planta y las respuestas esperadas de los documentos empleados en el modelado. Estos modelos pueden cambiarse o refinarse con más detalle si así se quisiera, de hecho si se mantienen las señales que componen la comunicación con la interfaz gráfica, la manera en que estas señales son obtenidas no afecta el comportamiento del simulador, pudiendo hacer cambios y ver sus resultados inmediatamente.

#### **Referencias Bibliográficas.**

[1] Çengel, Yanus A. y Boles, Michael A., *Termodinámica Segunda Edición*. McGrawHill México, 1994  
Tomos I y II

[2] Schwarzblat Morris, Villalobos Antonio, Morales Sandoval Jaime B., “*Desarrollo de un modelo de simulación de plantas nucleoelectricas tipo BWR*”, Instituto de Investigaciones Eléctricas, Departamento de Energía Nuclear, Cuernavaca, Morelos, Junio 1983.

## Capítulo 5. Interfaz Gráfica.

### 5.1. Introducción:

Como parte de un trabajo continuo en el área nuclear para mejorar los procesos de entrenamiento y creación de elementos humanos en el área nuclear, nuevos requerimientos para los sistemas de información y de computación han surgido. Esto ha contribuido a incrementar el interés y la aceptación de sistemas de software en las aplicaciones nucleares, y en particular en los cuartos de control de las plantas nucleoelectricas. En los años recientes ha habido un cambio del hardware y el software propietarios, al empleo de sistemas en el mercado no desarrollados específicamente para una sola aplicación. Como consecuencia de esto existe una demanda de sistemas que soporten el diseño y la implementación de Interfaces Gráficas de Usuario (GUIs) [1]

Los requerimientos de una interfaz grafica de usuario para aplicaciones nucleares, no son muy diferentes de las interfaces para aplicaciones de control. Sin embargo en las aplicaciones nucleares, es común que no existan procesos iniciados desde la interfaz gráfica, todas las aplicaciones de control son aplicadas por elementos de hardware, y la interfaz son empleadas únicamente para presentar los datos del proceso.

En la interfaz que se desarrollo para el simulador SUN-RAH, se emplean equipos de cómputo sencillos con la finalidad de tener una interfaz grafica que cumpla con sus objetivos de facilidad de uso y capacidad de despliegue de valores. La interfaz también permite modificar los parámetros de la simulación conforme esta se esta ejecutando, y observar los efectos de la modificación.

En la búsqueda de el mayor impacto visual y de una integración coherente con las simulaciones en Matlab®, se opto por hacer la programación empleando las librerías Microsoft DirectX® y Visual C++®, ya que este entorno de programación puede comunicarse con Matlab, empleado las rutinas y librerías proporcionadas por el programa matemático.

### 5.2. Interfaz de programación DirectX.

Microsoft DirectX es un conjunto avanzado de interfases de programación multimedia (APIs: Application Programming Interface), incorporadas dentro de los Sistemas operativos Windows.

DirectX provee una plataforma de desarrollo estándar, para computadoras con el sistema operativo Windows, de manera que los desarrolladores de software, puedan tener acceso a características especializadas del hardware, sin tener que escribir código específico del hardware donde se ejecutan los programas.

Esta tecnología fue introducida en 1995, y es reconocida por Microsoft, como el estándar para la creación de contenido multimedia, en los sistemas Windows.

DirectX es una tecnología de Windows, que permite el mejor desempeño en gráficos y sonido, cuando se ejecutan aplicaciones multimedia como juegos, videos, etc., en una computadora personal.

En el núcleo del funcionamiento de DirectX, esta el concepto de APIs, una API actúa como un puente entre el hardware y el software, de manera que estos puedan comunicarse entre ellos. Las APIs DirectX, permiten características avanzadas como el acceso a las capacidades que tiene el hardware de emplear gráficos tridimensionales. También se encarga de controlar y acelerar gráficos bidimensionales, diversos dispositivos como el ratón, teclado y el joystick, etc.

La versión mas reciente de DirectX es la versión 9.0b y es soportada por los sistemas operativos: Windows 98, Windows 98 SE, Windows Millennium Edition, Windows 2000, Windows Server 2003 y Windows XP. Windows 95 soporta hasta la version 8.0a.

### **5.2.1.Requerimientos:**

Sistema Operativo:

Microsoft Windows® 98, Windows Millennium Edition (Windows Me), ó Windows 2000, Windows Server 2003, Windows XP.

Disco Duro:

Aproximadamente 65 megabytes de espacio disponible para su instalación, una vez instalado pueden ser eliminados los archivos de instalación. El promedio de espacio ocupado es de 18 MB, Las versiones más recientes sobrescriben las más antiguas.

La versión de DirectX que se emplea en el desarrollo del simulador es la 8.1. Esta versión era la más reciente cuando se empezó el desarrollo del simulador, y también es la que se maneja en los textos usados como referencia y aprendizaje de esta interfaz:

Microsoft DirectX, provee dos tipos básicos de distribuciones, la versión de usuario y la versión de programador, En la versión de usuario, solo se instalan las librerías dinámicas DLL, y demás componentes, necesarios para la ejecución de un programa que emplea la interfaz.

En la versión de programador, se ofrecen diversas librerías, para varios lenguajes de programación, junto con archivos de definición de funciones (archivos .h), diversos ejemplos y utilerías, así como varios archivos de ayuda.

La versión de programador, también incluye la versión de usuario y ambas son gratuitas.

DirectX para C++

Los componentes de la versión 8.1 de DirectX (Pueden cambiar de versión en versión) son:

- DirectX Graphics: Combina los módulos de versiones anteriores DirectDraw y Direct3D en una sola API, esta interfaz es usada para la programación de todo lo relacionado con gráficas, incluye varias librerías que simplifican muchas tareas en la programación de gráficos.
- DirectX Audio: Combina los módulos de versiones anteriores DirectSound y DirectMusic, Contiene librerías que proveen una API para la programación de lo relacionado con el audio.
- DirectXInput: Provee el soporte para varios dispositivos de entrada incluyendo modernos controles sensibles a la fuerza aplicada.
- DirectXPlay: Provee soporte para sesiones multiusuario, ya sea un juego o en una aplicación compatible.
- DirectXShow: Provee un sistema de captura y reproducción de flujos multimedia, tales como animaciones y videos.
- DirectXSetup: Es una API sencilla que provee una manera fácil de instalar las aplicaciones y componentes DirectX.

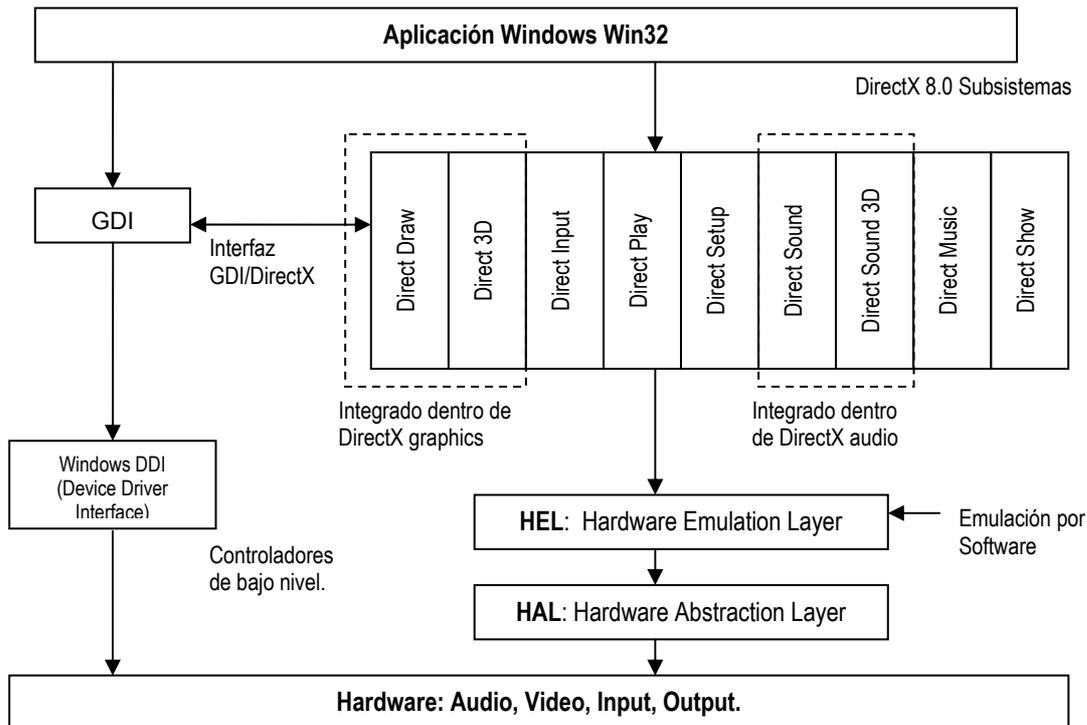


Figura 5.1: La arquitectura de DirectX y su relación con Win32.

### 5.2.2. Bases de DirectX

DirectX está basado en objetos COM. COM son las siglas de *Component Object Model*, es un modelo de programación orientada a objetos, usado en muchas aplicaciones. El uso de componentes COM permite modularizar la programación de manera que cada sistema y subsistema que realicen una operación independiente, puede encapsularse en un objeto COM, estos componentes siguen reglas muy estrictas de definición basadas en la programación orientada a objetos. La gran ventaja de los

componentes COM reside en que permite alterar los módulos o reemplazarlos, sin tener que recompilar la aplicación, de manera que se pueden hacer mejoras a los módulos componentes, sin alterar el resto de la aplicación y sin tener que recompilar todo el desarrollo.

### **5.3. Integración entre Matlab y Visual C++.**

Los modelos matemáticos que representan la planta nuclear, ya sea parcial o totalmente, y que de diversas formas (ya sea en ecuaciones en forma de texto, de bloques de Simulink, o de mezclas de ambos), forman parte de una representación simulable de Matlab. Pueden ser representados en otros entornos computacionales de acuerdo a lo que el propio programa Matlab dispone. Aprovechando esta característica que en versiones recientes de Matlab es cada vez más poderosa, se puede diseñar un motor de simulación que se comunique fiablemente con un entorno de programación, como Visual C++, de manera que un programa externo a Matlab pueda comunicarse con este entorno y pueda ejecutar comandos de línea de texto en una ventana de comando de Matlab, pudiendo así acceder a la mayor parte de la funcionalidad de Matlab, así como a variables del workspace.

Con el fin de que la simulación tuviera un entorno diferente al que provee Matlab y donde los parámetros pudieran alterarse y visualizarse de manera más parecida a las condiciones de una planta nuclear real, así como de ver información en diferentes presentaciones, de manera más cómoda y puntual que en el árido entorno de Matlab, se diseñó un programa que servirá de Fachada (Front-End), para la simulación, este programa fue realizado en Visual C++ 6 y está conformado de los siguientes módulos principales.

- Funciones de acceso al entorno Matlab, así como a los datos provenientes de estas simulaciones y actualización de variables o parámetros.
- Funciones encargadas de hacer cálculos de cada uno de los elementos que conforman la simulación de la planta nuclear, ya sea de cálculo de parámetros o de variables necesarias para la representación de los datos en el programa.
- Funciones que clasifican la información de las diferentes funciones de los electos mencionadas anteriormente y las relacionan y preparan para su graficación, entrada como parámetro a Matlab, cálculos posteriores, etc.
- Funciones de entorno gráfico, encargadas de llevar todos los datos y sus actualizaciones a la pantalla.

El motor de la simulación está programado en las extensiones de Interfaz Externa/API de Matlab estas son las encargadas de tender un puente de comunicación, entre el Matlab y el compilador Visual C++, de manera que cuando el programa se está ejecutando, una ventana de comandos de Matlab ( command window ), se ejecute en segundo plano, permitiéndole a una función cualquiera en una función de VC++, acceder a esta ventana de comandos y ejecutar la simulación de alguno de los modelos necesarios, pudiendo además, esta función obtener e introducir datos hacia y desde Matlab,

de acuerdo a una acción de usuario, o como producto de algún cálculo realizado. De esta manera la comunicación entre el simulador y el programa Front-End o de fachada del simulador están relacionados fuertemente, sin embargo el simulador es independiente, por lo que pueden hacerse pruebas y simulaciones desde Matlab, sin la intervención directa del programa Front-End, permitiendo así una depuración fiable y rápida.

La manera en que el programa ejecuta la simulación en Matlab, difiere de la manera tradicional en la que se realiza una simulación en Simulink, ya que en este tipo de simulaciones, el tiempo en el que se realiza la simulación es fijo, y los parámetros que se fijan, se mantienen durante todo el comportamiento de la simulación, proporcionando diferentes salidas a lo largo del tiempo de simulación, conforme a un intervalo de tiempo ya sea fijo o variable, dependiendo del método de integración. Así en una simulación de 10 segundos, con un intervalo de 1 segundo, tendremos diez salidas y además de diez valores de las variables de estado.

Para dotar a la simulación de un aspecto continuo de ejecución, se realiza un ciclo de simulaciones con un tiempo fijo de corta duración, y con intervalos de tiempo fijos para obtener las salidas, guardando en cada caso los valores de las variables de estado de los componentes de salida, proporcionados por Matlab por cada intervalo de tiempo determinado como paso de simulación, de manera que cada vez que se realiza la simulación se hace en un tiempo que avanza a voluntad, definido en el programa Front-End. Con el tiempo de simulación y las variables de estado del tiempo anterior, es posible calcular el siguiente par de valores de salida y variable de estado. Al aumentar el tiempo continuamente y emplear los valores de estado anteriores inmediatos, es posible hacer que la simulación dure el máximo valor para una variable contador en VC++, de 64 bits que es de: 18446744073709551616, suponiendo que el contador de simulación avanza una vez cada 1/1000000 de segundo, tardaría 584942 años en desbordarse el contador, lo cual es una buena probabilidad de que no se desborde.

### **5.3.1. Funciones de integración con Matlab.**

La integración con Matlab se realizó empleando las siguientes capacidades de este entorno matemático.

Uso del Matlab Engine®, proporcionado por matlab para la integración de C y Matlab.

El ambiente Matlab provee una interfaz para programas externos escritos en lenguaje C o Fortran.

La documentación de Matlab [2]. Recomienda el empleo de interfaces de usuario programadas en C, y el empleo de programas de Matlab corriendo en el fondo, como parte de una estrategia para acortar el tiempo de desarrollo de aplicaciones.

Para que el compilador, En este caso Visual C++ 6, acepte las librerías y los comandos dirigidos a una ventana de comandos externa de Matlab, deben hacerse varias ligas y especificar varias direcciones de

localización de archivos, etc. Esto puede hacerse de manera semiautomática invocando el comando: `mex -setup` en cualquier línea de comandos de Matlab.

#### Rutinas MX.

Existe una librería de funciones encargada de proveer el acceso de programas externos a matrices con el formato empleado internamente por Matlab. Estas funciones siempre cuentan con el prefijo “mx”. La mayor parte de estas rutinas permiten manejar las matrices creadas en programas escritos en C o Fortran, como si fueran matrices de Matlab.

#### Rutinas MEX.

Las rutinas que cuentan con el prefijo “mex”, piden la ejecución de alguna función o programa dentro de el entorno Matlab, es decir que las funciones no son realizadas dentro del programa de C, sino que se pide que un intérprete de comandos de Matlab las realice. Varias de estas funciones colocan variables dentro del Workspace de Matlab. El acceso a estas variables, desde el programa de C o Fortran debe especificarse, para obtenerlas ya sea en una formato compatible con los tipos definidos de C( int, flota, double , etc ), o de un formato de matriz compatible con el de Matlab, definido anteriormente con alguna rutina MX, como de específico anteriormente.

#### Funciones C Engine®

Existen varias funciones del *Engine* de Matlab. Las que se emplearon para hacer la interfaz de este simulador fueron:

engClose	Cierra la sesión del motor de Matlab.
engEvalString	Evalúa en Matlab una expresión en forma de texto.
engGetVariable	Obtiene una variable del <i>workspace</i> .
engGetVisible	Activa la visibilidad de la sesión.
engOpen	Inicia una sesión del motor Matlab.
engOutputBuffer	Manda la salida de Matlab a un buffer determinado.
engPutVariable	Coloca variables dentro del <i>workspace</i> de Matlab.
engSetVisible	Muestra u Oculta una sesión del motor Matlab.

Estas funciones requieren en el caso del lenguaje C que en el programa incluya la cabecera `include.h`, de la siguiente manera: `#include “engine.h”`

La capa de integración en el caso de este simulador fue implementada usando los módulos de Matlab conocidos como: Matlab compiler y Matlab runtime Server, estos módulos proveen dos tipos de librerías.

Librerías de compilación: Son librerías que agregan funciones al tiempo de compilación de un programa.

Librerías dinámicas: que agregan funcionalidad a un programa en el momento de ejecución.

Las librerías empleadas fueron las siguientes:

libeng.lib      Para acceder a el Matlab engine.

libmx.lib      Para acceder en “C” a los tipos de datos compatibles con los usados en Matlab.

Cabe mencionar que aparte de lenguaje C y Fortran, Matlab también puede comunicarse con Java, sin embargo se eligió el lenguaje C, por la familiaridad obtenida a lo largo de la carrera, y por que es la interfaz más documentada, además que existen ejemplos de terceros disponibles por Internet, que son de ayuda en la verificación de sintaxis y en la colocación de las funciones dentro del programa.

La integración con Matlab a través de las funciones del engine, puede hacerse de varias maneras. Existen funciones que crean estructuras de datos, compatibles con los tipos de datos de Matlab, y obtiene los datos directamente del Workspace de Matlab, los datos obtenidos, son colocados en una estructura compatible en el lenguaje C que puede desglosarse en tipos de datos comunes como enteros o de punto flotante. Sin embargo estas funciones revisten varios problemas, ya que el formato interno de Matlab es muy complejo, ya que permite datos en números complejos, reales de diferente precisión, enteros, variables lógicas, texto, etc. Encontrar la función o conjunto secuencial de funciones que permitan contextualizar el dato que queremos obtener y colocarlo en una variable tipo “int” o “float”, es difícil e inexacto y en algunos casos el mismo “engine” no posee las suficientes funciones para rescatar los datos con seguridad. Este problema no es exclusivo de este simulador y esta documentado en la documentación del API de Matlab [2]. Aunque Matlab asegura que en futuras versiones estos errores serán corregidos, la premura del desarrollo, hizo necesario buscar una alternativa estable, considerando que los valores a obtener serían reales.

Para mandar los datos a Matlab se empleo la función del “engine”: engEvalString, esta función permite introducir un texto a la ventana de Matlab, tal y como si este fuera tecleado directamente por el usuario, con la ventaja de que como el desarrollo se hizo en “C”, este texto puede ser alterado automáticamente por ciclos o por diversas trayectorias del programa.

Este método también es empleado en el simulador para iniciar la ejecución de los modelos desarrollados en Simulink, con los parámetros de tiempo de simulación y método de simulación requeridos.

Para obtener los datos de Matlab, se emplea una característica proporcionada por el “engine” de generar un “buffer” de texto en donde se almacena la ultima información que el programa muestra después de introducir un comando, esto es, obtener el texto cuando en una sesión normal de Matlab, el usuario presiona “Enter”. Con este texto en un arreglo estable y único de “C”, se procede a obtener los

valores desplegados, con una sencilla función de eliminación de texto alfabético, y con la conversión pertinente de texto a número.

La estabilidad obtenida con este método de extracción e introducción de valores, es superior en todos los casos a la obtenida con las funciones de extracción directa de variables del “engine” de Matlab, las cuales no presentan una documentación que explore todos los casos de obtención de variables, por lo que su funcionamiento esta limitado y es incierto. Lo cual pudimos comprobar tras una depuración exhaustiva de varios casos en el simulador.

Otra información obtenida de capturar el texto que proporciona Matlab, es el estado de la simulación, ya que esta puede estarse ejecutado, estar pausada, haberse detenido, tener un error, etc.

Para ejemplificar la integración en la figura 4.2 se presentan algunos comandos empleados y una breve descripción de sus parámetros. Cabe mencionar que este no es el archivo empleado en simulador sino únicamente un ejemplo de las funciones representativas.

#### **5.4. Paradigma de programación.**

El enfoque orientado a objetos para el desarrollo de software se propuso por primera vez a finales de los años 60. Sin embargo, las tecnologías de objetos han necesitado casi veinte años para llegar a ser ampliamente usadas. Durante la primera mitad de los años 90, la programación orientada a objetos se convirtió en el paradigma de elección para muchos productores de software y un creciente número de sistemas de información y profesionales de la ingeniería.

Las tecnologías de objeto llevan a reutilizar componentes de software, y esto conlleva a un desarrollo de software más rápido y programas de mejor calidad. El software orientado a objetos es más fácil de mantener debido a que su estructura es inherentemente autónoma.

El concepto de orientación a objetos determina la siguiente definición:

*Orientación a objetos = objetos + atributos de clase + herencia + comunicación.*

Una clase es un conjunto de variables y métodos con características pertenecientes a un conjunto de elementos. Un objeto es un elemento perteneciente a un clase y que hereda los atributos de la clase (variables y métodos) pudiendo definir o refinar los suyos propios.

La comunicación entre los objetos se hace entre las variables y métodos de la clase definidas para tal efecto.

La interfaz gráfica fue programada empleando el paradigma de programación orientada a objetos, Se definieron diversas clases que representaran los distintos elementos a visualizar, o que realizan algún calculo específico.

```

#include <windows.h>
#include "engine.h"
#define BUFSIZE 64
.
.
.
if (!(ep = engOpen(NULL)))
{
    exit(-1);
}

engOutputBuffer(ep, buffer, BUFSIZE);

engEvalString( ep, "load_system('extraheatmascontrol');" );

float EscanValor( void )
{
    for(int i=0;i<BUFSIZE;i++)
    {
        if( isdigit( buffer[i] ) || buffer[i] == 46 )
            cvalor[i] = buffer[i];
        else
            cvalor[i] = ' ';
    }
    return atof( cvalor );
}

engEvalString( ep,
    "set_param('extraheatmascontrol','Solver','ode15s','StartTime','0','StopTime','1');" );

engEvalString( ep, "set_param('extraheatmascontrol','SimulationCommand','start');" );

engEvalString( ep, "close_system('extraheatmascontrol',0);" );

```

Archivo de cabecera con las funciones del engine Matlab.  
 Tamaño del "buffer" de texto a utilizar.

Inicialización de la ventana de comandos de Matlab

Establecimiento del "buffer".

Carga la simulación.

Uso de la función engEvalString para escribir un texto directamente en Matlab.

Función que obtiene un valor numérico dentro del buffer eliminando el texto y regresando un valor tipo "float".

Función que introduce los parámetros de la simulación.

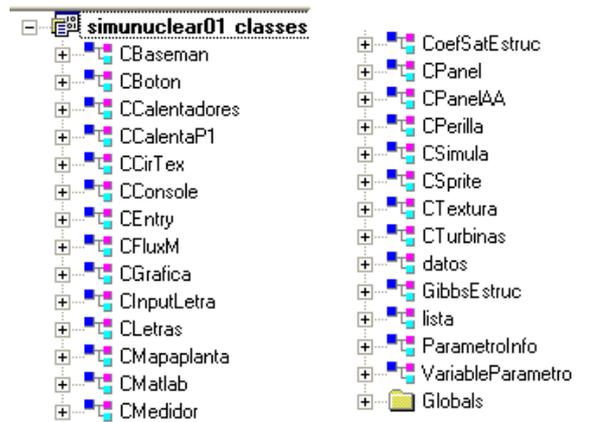
Función que inicia la simulación.

Función que cierra la simulación en Simulink.

**Figura 5.2:** Ejemplificación de los comandos empleados para establecer la comunicación entre Matlab y el programa en "C".

Las funciones que se forman parte del motor gráfico básico y que se encargan de las funciones de representación que todas las demás clases emplearan, se colocan de manera global. Esto con el doble fin de reutilizar las rutinas que proporciona el libro de el Zen de Direct3D [3]. Así se puede evitar hacer las clases en demasiados niveles de herencia que dificulten depurar el código, tratando de encontrar los errores en varias capas de programación.

Las clases principales se observan en el siguiente diagrama:



**Figura 5.3:** Clases empleadas en el simulador del sistema de agua de alimentación

### 5.5. Especificaciones.

La interfaz gráfica esta diseñada para facilitar la visualización de los procesos que se realizan dentro de una central nucleoelectrica de tipo BWR. La interfaz también tiene el propósito de ocultar la interacción común con una computadora, de manera que el usuario tenga la sensación de que esta operando un dispositivo computarizado, ajeno a un programa clásico de PC de plataforma Windows, con lo que se pretende una mayor confianza con el programa y un mayor deseo de exploración y manipulación por parte del usuario.

Las especificaciones de la interfaz, incluyen una visualización no realista animada y esquematizada de los procesos de la planta, sin embargo se tratará que los módulos que representan equipo o manipulación dentro del simulador de manera análoga al equipo que hay en la planta, sean representados con la mayor fidelidad posible a lo real. De esta manera se pretende cumplir con la capacidad de entrenamiento para operadores que se mencionó anteriormente.

En el afán de hacer más atractiva la interfaz, se consideraron varios tipos de esquemas de visualización proporcionados por Windows. Entre los que destacan las GUI de Windows y los sistemas de desarrollo DirectX.

Windows proporciona un programa de aplicación con acceso a cientos de llamadas de funciones directa o indirectamente a través del conjunto de clases MFC. Estas llamadas son controladas por varios módulos principales, incluyendo los módulos GDI (Graphic Device Interface). El módulo GDI contiene todas las rutinas para crear y mostrar gráficos [4].

Desde un principio se observo que el emplear este tipo de Interfaces tendría ciertas desventajas de acuerdo a las especificaciones de disociación con el entorno Windows, y que la programación en este formato crearía un simulador de apariencia poco atractiva hacia el usuario inexperto, lo cual no cumple con los objetivos predeterminados. La tecnología DirectX es varias veces más rápida y más robusta que la tecnología GDI y/o MCI (Media Control Interface), de la cual es nativo Windows [5].

La programación empleado GDI, reviste ciertas ventajas de velocidad y facilidad de depuración, así como de una gran cantidad de código escrito automáticamente por el entorno de desarrollo.

La interfaz DirectX cumplía con la mayor parte de nuestras especificaciones, y forma parte de la propuesta de hacer atractivo este tipo de simuladores y aplicaciones, no solo para estudiantes y operadores de tecnología nuclear, sino también para estudiantes en áreas de la computación, sistemas, diseño gráfico, etc.

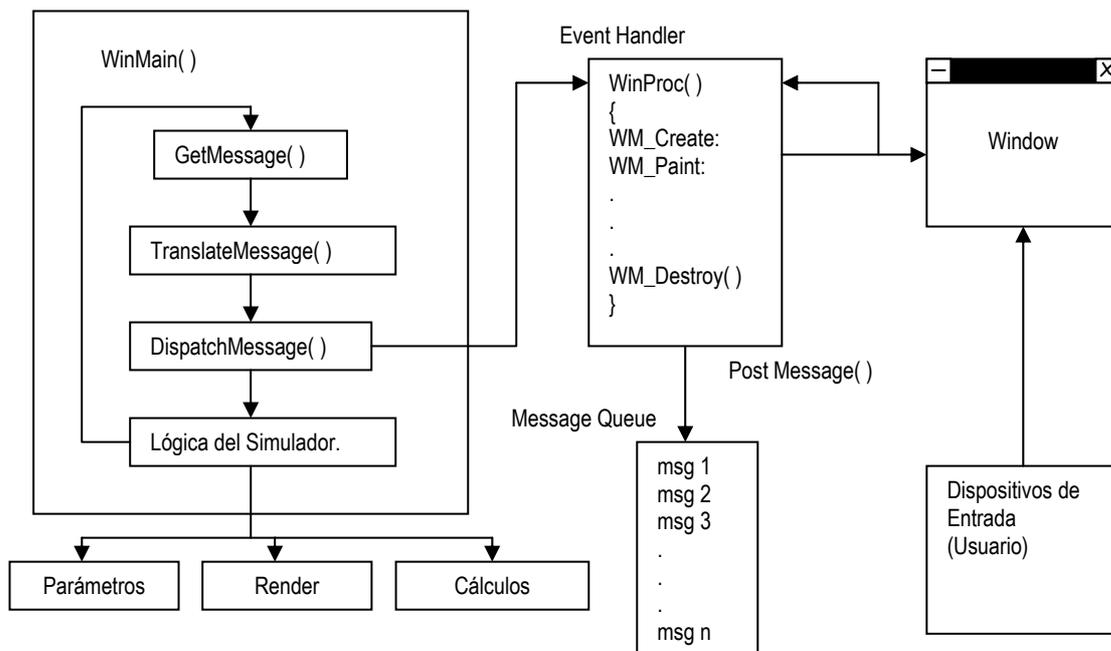
La interfaz DirectX requiere de una cierta capacidad gráfica inusual en los sistemas computacionales de grado universitario, como es una placa de aceleración gráfica básica. Este requerimiento es necesario para lograr el despliegue de efectos avanzados como la translucencia y la transparencia, acelerar las operaciones aritméticas matriciales relativas al despliegue de graficas, coordinar los traslapes y colisiones de los elementos gráficos en pantalla, etc.

### 5.5.1.Ciclo de ejecución principal.

Las aplicaciones Windows son ejecutadas en un ciclo que se encarga de recibir eventos ocurridos en las diferentes ventanas, de administrar esos eventos, efectuar operaciones con ellos, y de mostrar los efectos resultantes en la ventana correspondiente.

Para la ejecución de la interfaz gráfica, el programa deberá realizar el mismo ciclo, con lo que obtiene diversos parámetros y genera respuestas a los eventos de los dispositivos de entrada.

El ciclo de ejecución ocurre constantemente durante la vida de la aplicación y la secuencia de sus fases determina el comportamiento del simulador, como se muestra en la siguiente figura.



**Figura 5.4:** La mecánica del ciclo de procesamiento de mensajes de Windows, en la ejecución de la interfaz gráfica.

Dentro del ciclo la parte denominada como “Render”, se encarga de la visualización de los elementos gráficos en la pantalla, así como la parte de parámetros y de cálculos se enfocan a modificar los parámetros de la simulación y enviarlos a Matlab, o de hacer alguna calculo necesario en la representación de algún elemento.

### **5.6. Alcance.**

La interfaz gráfica pretende ser un vínculo entre el motor matemático y el usuario. Como tal debe representar facilidad de operación y suficientes capacidades para poder manipular, visualizar comparar y verificar datos y parámetros desde y hacia el motor matemático que esta ejecutando la simulación.

En su diseño el simulador se consideró para ser operado por estudiantes de nivel medio superior, superior o incluso operadores de plantas reales en entrenamiento, por lo que se considero que interfaces llamativas, en 2D, en una interfaz que permitiera animaciones, gráficos bitmap, efectos visuales como translucencia , transparencia, ajuste de texturas a superficies, etc., cumpliría con los requerimientos del publico al que esta dirigido, en el afán de mantener sus atención y de brindar una experiencia didáctica cómoda.

El programa no considera modelos tridimensionales, vistas de distintos ángulos, etc. El impacto que estos tendrían sobre una simulación no fue estudiado a fondo, pudiendo favorecer el efecto educativo o crear distractores en modelos complejos.

El motor grafico creado en DirectX es fácilmente ampliable a espacios tridimensionales, ya que el entorno DirectX considera el espacio 2D como un caso particular de el espacio 3D.

Debido al impacto que los videojuegos en tercera dimensión han tenido en los últimos tiempos, sobre todo en gente mayor de 15 años, el alcance de este tipo de desarrollos puede ser incrementado haciendo interfaces atractivas.

El desarrollo en DirectX, sufre constantes revisiones y amplia las capacidades de acuerdo con la tecnología en hardware disponible, por lo que representa una posibilidad de realizar interfaces de alto desempeño y avanzadas prestaciones con recursos moderados.

### **5.7. Desarrollo.**

Una interfaz de usuario brinda dos facultades primordiales a nuestro desarrollo, en primer lugar convierte y esconde el trabajo de desarrollo de los modelos, permitiendo únicamente visualizar los resultados y modificar las entradas, y en segundo lugar, permite que el usuario tenga una mayor comodidad para encontrar, manipular y visualizar los componentes con los que debe forjar cierta familiaridad debido a su carácter de entrenamiento y de educación.

El proceso de diseño consideró necesario hacer un esfuerzo por escapar de las interfaces comunes de tipo Windows, que aunque en su tiempo fueron una revolución, ahora se sienten estrictas y

restringidas, sobre todo en aplicaciones en donde se pretende llamar la atención por parte de usuarios inexpertos.

El uso de interfaces de tipo menú y tablas, se ha reemplazado en los desarrollos actuales por aplicaciones con interfaces más atractivas. Así, las paginas Web en HTML, han cedido terreno a las que emplean componentes dinámicos programados en lenguajes como Flash®, Director®, Shockwave®, etc.

### **5.7.1. Características proporcionadas por librerías DirectX.**

Las posibilidades para hacer interfaces mas poderosas en el aspecto grafico y de interactividad, son varias, y ante la encrucijada de emplear una opción u otra, se decidió que para que el proyecto tuviera el mayor numero de posibilidades y capacidades, se emplearía la tecnología grafica más robusta y que permitía mayor realce de los conceptos a simular, por lo que se decidió hacer la interfaz en el entorno DirectX®.

DirectX permite una gran cantidad de efectos gráficos y capacidades de control total de lo expuesto en pantalla, entre los efectos que provee se encuentran:

- Superposición: Se refiere a que los objetos puedan ordenarse de acuerdo a su “distancia”, de manera que en el caso de que dos objetos que comparten las mismas dimensiones y posición, se observará solo el que se encuentre más cercano a la pantalla.
- Transparencia: Se refiere a observar completamente un objeto lejano a uno cercano que lo cubre completamente cuando el objeto posterior este presente.
- Translucencia: Se refiere a observar un objeto lejano y un objeto cercano que lo cubre completamente, ambos son visibles debido a un efecto de “cristal”, en donde aunque se nota la presencia del objeto cercano, el objeto lejano, también se percibe.

Cabe mencionar que aunque otros efectos son posibles, estos no son empleados en la versión actual del simulador.

Una característica que se emplea en al simulador es el uso de texturas.

Una textura es un gráfico del tipo mapa de bits cuadrado de dimensiones fijas, ya sea de 32x32, 64x64, 256x256, etc., que puede ser incorporado a una superficie virtual, para después ser visualizado, de manera que el grafico adopta la forma de la superficie virtual, según se le indique, ajustándose al tamaño de la superficie. DirectX permite generar superficies de cualquier forma, ya sea triangular, rectangular, circular, o formas irregulares, también permite colocar una textura a esas superficies de diferentes maneras, ya se expandiéndola, contrayéndola, duplicándola, recortándola, etc. Aunque el proceso de adaptación y conformación de una textura es un proceso matemático de operación de

matrices en diversas iteraciones, el entorno DirectX permite hacer la asignación de texturas de una manera automática, o de influir en el proceso matemático a voluntad.

### 5.7.2. Características proporcionadas por terceros.

De entre la bibliografía consultada el libro que otorgó la mayor cantidad de código y ayuda en la comprensión de los conceptos involucrados en el desarrollo de aplicaciones gráficas con DirectX fue el libro de programación de Meter Walsh [3]. Este libro proporciona el código fuente en CD-ROM y la explicación a cada línea de los métodos proporcionados, la licencia de uso menciona que las funciones pueden emplearse en desarrollos personales siempre que se conserve el comentario que indica el nombre del autor de las funciones y el año en el que se desarrollaron.

Varias de las funciones descritas en el libro emplean el concepto de *sprite*.

Un *sprite* es un objeto gráfico de determinadas dimensiones, colocado en una posición particular, dentro de una superficie capaz de ser visualizada en la pantalla.

Los *sprites* son estructuras autónomas que pueden contener animaciones o imágenes fijas y cuyo comportamiento y posición es determinado por el programa que los invoca.

En la implementación de *sprite* empleada en el simulador se emplean *sprites* con una posición fija y con animaciones, así, un *sprite* contiene una colección de imágenes alternándose en ciclos o de acuerdo a decisiones del usuario.

Otros métodos proveen las funciones, para dibujar un *pixel* en pantalla, y basándose en esa función básica, generan métodos para dibujar líneas, rectángulos, cargar imágenes de archivos, etc.

Algunas funciones son simples temporizadores o inicializadores, útiles para configurar un entorno estable de programación.

A continuación se muestra una lista de los métodos proporcionados por el libro y su función específica.

-Manejo de errores:

SetError: Despliega un código de error en la ventana de depuración.

-Código de Inicialización de DirectX3D:

InitDirect3DDevice: Inicializa un dispositivo DirectX3D

-Codigos de graficos en 2D:

SetPixel32: Dibuja un píxel de un color específico.

SetLine32: Dibuja una línea de un color específico.

Rectangle32: Dibuja un rectángulo.

LoadBitmapToSurface: Carga una imagen de mapa de bits a una superficie.

CopySurfaceToSurface: Carga una superficie hacia otra superficie compatible con la primera.

- Códigos de validación de dispositivo.

ValidateDevice: Verifica cada ciclo si el dispositivo es valido, en caso de lo contrario, intenta readquirirlo, de ser posible.

RestoreGraphics: Reinicializa las superficies faltantes del dispositivo, cuando este se ha reiniciado.

- Código de escritura de letras (Fuentes).

LoadAlphabet: Carga un mapa de bits de contenido alfabético, dentro de la memoria.

UnloadAlphabet: Descarga el alfabeto de la memoria.

PrintChar: Imprime un carácter en una superficie, empleando un alfabeto cargado en memoria.

PrintString: Imprime una cadena de caracteres en una superficie.

- Códigos de Tiempo.

InitTiming: Inicializa un contador.

EsTiempo: Verifica el cumplimiento de un lapso de tiempo.

Pause: Espera un tiempo determinado.

- Código para manejo de *Sprites*.

Initialize: Inicializa un conjunto de *sprites* con una imagen proveniente de un archivo o en memoria.

Shutdown: Libera en memoria el espacio ocupado por los *sprites*.

SelectActiveSprite: Selecciona del conjunto, el *sprite* activo a ser visualizado.

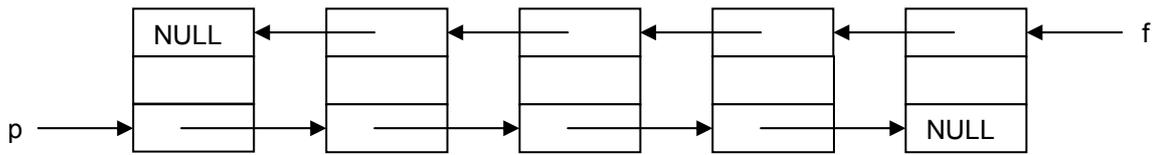
Todas estas funciones emplean el “Framework” de DirectX, que es un grupo de funciones de DirectX que realizan operaciones entre archivos o en superficies compatibles. Estas funciones están explicadas a detalle en la documentación de DirectX o en el libro de Wolfgang F. Engel [6].

### **5.7.3. Características propias desarrolladas específicamente para el simulador.**

- Cálculo de gradientes de color: Desarrollo de métodos para el incremento en el tono de un color como resultado de un parámetro de nivel en la conformación de líneas o de áreas.

Puede encontrarse en el cuerpo del archivo FluxMeter.h

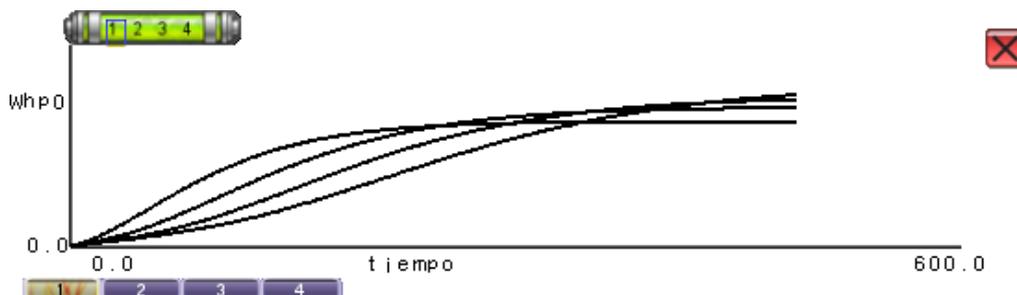
- Implementación de un sistema de conexión de puntos contiguos para su uso en graficas de tendencia tipo X,Y: Este sistema emplea listas doblemente ligadas para llevar un inventario de los puntos visibles, y su relación con puntos anteriores, de manera que se genera una línea contigua basada en los valores a representar en la gráfica. Este sistema también permite el recortar partes de la grafica, actualizando los puntos en la lista de manera que la visualización responda al recorte.



**Figura 5.5:** Lista doblemente ligada.

- Desarrollo del módulo “Gráficas”: Empleando las funciones basadas en las listas doblemente ligadas y los conceptos de graficación de puntos y líneas, se generaron diversos objetos para permitir la visualización de las graficas bidimensionales de línea tipo X-Y (plot) como las proporcionadas por programas matemáticos como Matlab y Simulink, de acuerdo a las siguientes especificaciones de diseño:

- Las gráficas son visualizadas en tiempo continuo, es decir que conforme se ejecuta la simulación se van añadiendo valores a las graficas, de manera que si un parámetro altera al comportamiento de un modelo, esto se pueda visualizar inmediatamente suceda, sin ejecutar rutinas de calculación de graficas actualizadas.
- Las gráficas son auto ajustables tanto en el eje de las abscisas como en el de las ordenadas, es decir que, se definirá una ventana en donde se observa la gráfica, pero si el tiempo o los valores rebasan los parámetros iniciales, esta se ajustara, dejando atrás una porción de los valores anteriores para abrir espacio para la visualización de nuevos valores.
- Las gráficas aceptan cuatro variables simultaneas y los grafican en la misma área de visualización en diferentes colores, por lo tanto el ajuste automático se hace por cada uno de los rangos pertenecientes a las variables, esto implica que cada rango puede ser consultado independientemente.



**Figura 5.6:** Graficas tipo X-Y para visualización de variables en el simulador.

- Diseño de un esquema de comunicación entre Matlab y Visual C++.

El desarrollo de los protocolos de comunicación e intercambio entre Matlab y C, se hizo empleando las rutinas propias de Matlab y de el lenguaje C, explicadas en el capítulo dos. La implementación sufrió de variadas alteraciones durante el proceso de desarrollo, debido a la exigencia de

controlabilidad y ejecución en tiempos razonables de un simulador. La escasa documentación se reviso en varias ocasiones, y se adaptaron diversas ideas provenientes de diversas fuentes.

- Otros elementos se desarrollaron como parte de las necesidades de visualización e interacción, requeridas en la simulación, estos elementos emplean las características descritas anteriormente combinándolas en diversas formas, para conseguir los fines que persiguen.

#### **5.7.4. Proceso específico de simulación de un modelo en la interfaz gráfica.**

Para hacer que los modelos corran en el simulador con la apariencia de estarse ejecutando en tiempo real, se emplea la técnica de ejecución de la simulación completa a intervalos de tiempo pequeños (por ejemplo de 0 a 1, de 1 a 2, etc. ), de manera que al finalizar cada simulación sin importar el número de pasos intermedios que se tuvieron en la simulación, el estado final del sistema es colocado como el estado inicial de la siguiente simulación corta, el tiempo es incrementado, y las entradas reevaluadas otra vez, de esta manera con los valores de estado, de entradas y el tiempo es calcular los nuevos valores de el siguiente tiempo.

Por ejemplo:

Al inicio de la simulación se colocan los estados en los valores iniciales de la simulación (típicamente estos valores son ceros, debido a que le sistema no tiene memoria), las entradas son colocadas en sus valores iniciales, y el tiempo en su valor inicial (típicamente cero, también)

Se ejecuta una simulación de poca duración, por ejemplo un segundo, aunque esto puede variar, dependiendo del tipo de sistema.

La simulación ejecutará los pasos de inicio, ejecución, etc., y entregara de acuerdo al método de integración (*solver*) elegido varios valores de los estados y salidas de pasos intermedios entre el tiempo de simulación, en este caso de cero a uno, (por ejemplo: 0 segundos, 0.1 segundos., 0.5 segundos y 1 segundo), cabe mencionar que el número de estos pasos depende en gran medida del método de integración elegido, sin embargo, siempre entregará los estados y salidas del tiempo final.

Las salidas y los estados son enviadas a Matlab, por los métodos ya explicados y se procesan de acuerdo a lo necesitado por el usuario ó en cálculos internos. Los valores de estado también son realimentados a Matlab.

Para las subsecuentes simulaciones de Simulink, los valores de estado no serán valores iniciales o ceros, sino que estos valores serán los obtenidos de la simulación anterior, las entradas serán reevaluadas nuevamente, para verificar alteraciones o cambios de parte del usuario o de algún otro proceso. El tiempo de simulación no partirá ahora del inicio (cero), sino que iniciará en donde la simulación anterior se detuvo y finalizará un intervalo después.

Es posible que los estados internos de los diversos pasos de integración aumenten o disminuyan, sin embargo, solo se considerarán los estados y salidas al tiempo final, por lo que esta técnica es muy flexible respecto al número de salidas de Matlab.

Debido a que no todos los modelos dinámicos admiten métodos de solución con pasos de integración fijos, y que en los métodos con pasos de integración variables es difícil determinar exactamente el tiempo de los pasos y si estos ocurrirán un valor esperado.

Este método permite definir el intervalo temporal de representación de datos, lo cual es extremadamente útil al momento de presentar gráficas y de estandarizar los tiempos de otros componentes y procesos en el simulador.

Otra fundamental ventaja de este método, es la apariencia de ejecución continua de la simulación.

Los intervalos de simulación se hacen de un tiempo inicial a uno final, estos tiempos van siendo incrementados conjuntamente de acuerdo a un contador en una variable ya sea de 32 o 64 bits, por lo que incluso con intervalos de ejecución diminutos (de un millonésimo de unidad de tiempo), se asegura que el contador no se desbordara en una gran cantidad de años.

Si ocurren cambios en las entradas, el sistema calculará estados nuevos que correspondan a esta alteración al siguiente paso, por lo que en un solo intervalo, el sistema reflejará los efectos de estas alteraciones, sin tener que esperar a que la simulación termine intervalos de varias unidades de tiempo, para poder cambiar los parámetros y volver a correr la simulación.

Aunque el programa Matlab provee las instrucciones para alterar los métodos y los tiempos de simulación. La configuración que permite que los estados finales sean los estados iniciales de la siguiente simulación, debe ser indicado en la ventana gráfica de configuración “Simulation Parameters” en la edición del modelo Simulink.



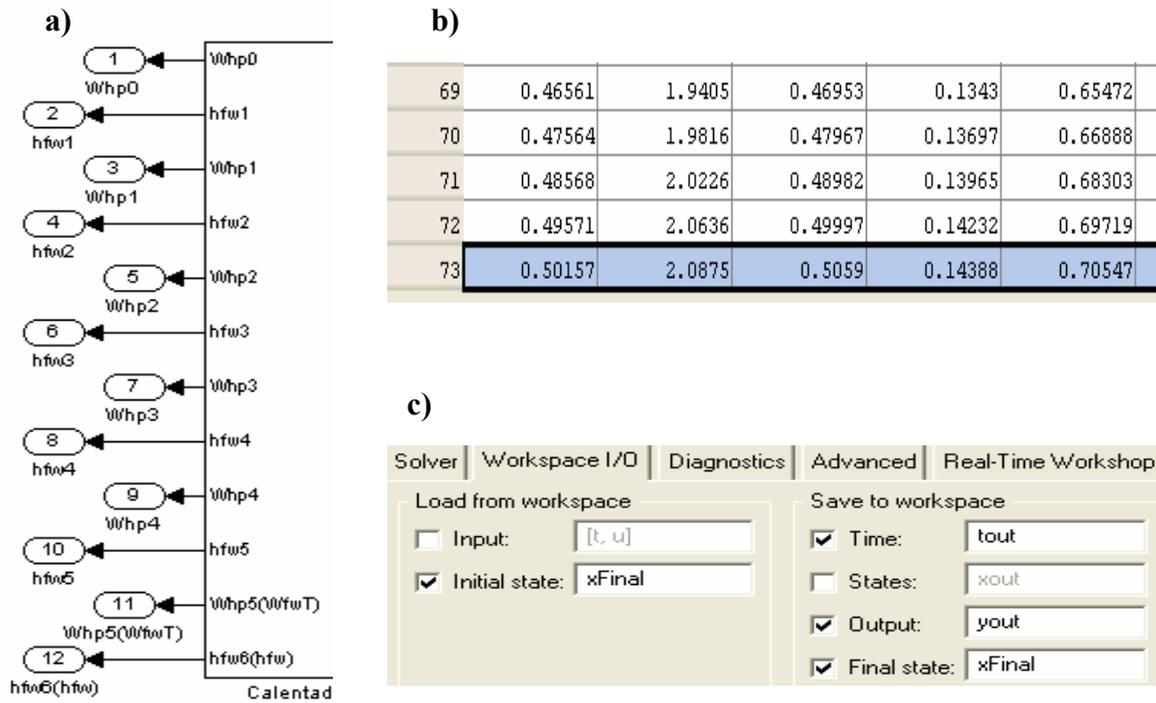
**Figura 5.7:** Tiempo de simulación inicial (la interfaz gráfica lo modifica cada ciclo) y método de integración adecuado para el modelo.

Los estados y las salidas de cada intervalo de simulación, son escritos en el Workspace de Matlab, por lo que pueden ser obtenidas por la interfaz empleando el buffer de texto descrito anteriormente en el documento de interfaces externas de Matlab [2].

La salidas son escritas en Matlab como una columna de datos, que registra todos los valores calculados dentro del intervalo de simulación especificado, por lo que existe una función dentro de la

interfaz grafica que se encarga de filtrar estos valores y obtener únicamente el valor final, que corresponde al final del intervalo.

Por cada salida en un modelo de Simulink, se obtendrá una columna del arreglo de salidas en el Workspace, por lo que la función también debe ser capaz de acceder a los valores finales de cada columna.



**Figura 5.8:**

a) Salidas en el esquema de Simulink

b) Representación de las salidas en el Workspace de Matlab: Cada columna representa los valores de una salida. Los valores corresponden a la simulación del tiempo 0 al tiempo 1. Los valores del último paso de integración corresponden al tiempo final y son los que son considerados.

c) Configuración en los parámetros de la simulación para que cada intervalo inicie con el estado final de la simulación anterior. (Debe ser indicado desde esta ventana)

### 5.7.5. Implementación específica: Sistema de Condensador Agua de Alimentación y controlador asociado.

El modelo de agua de alimentación que se empleo en las pruebas de integración de la interfaz gráfica es el modelo: “extraheatmascontrol.mdl”, este es un archivo de Simulink, en el que se encuadran tanto los modelos del condensador, los calentadores de baja y alta presión, el controlador de inventario de

refrigerante, la bomba de agua de alimentación, y un pequeño modelo que simula el nivel de agua dentro del reactor.

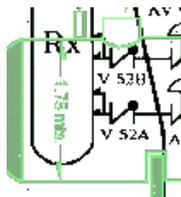
Para hacer la simulación, es necesario configurar los parámetros de simulación, como se indicó anteriormente de manera que el estado final de una simulación sea el estado inicial de la siguiente.

También debe especificarse que se desea que las salidas se coloquen en el Workspace. Algunas salidas pueden forzarse a escribirse en el Workspace empleando el bloque “to Workspace” de las librerías de Simulink.

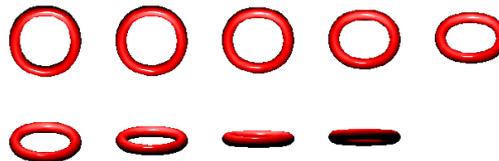
## 5.8. Despliegues.

### 5.8.1. Galería de elementos.

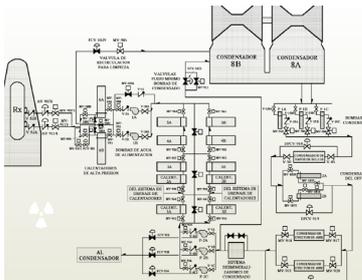
Los elementos que componen los despliegues gráficos se basan en las funciones descritas anteriormente. Los elementos consignan varias animaciones de acuerdo a los eventos y valores proporcionados por el motor matemático en el simulador. Las técnicas principales se muestran a continuación



Translucencia



Sprites Animados



Mapa de bits

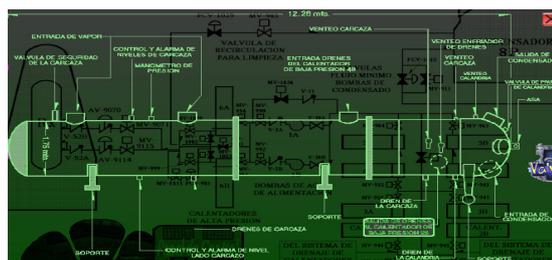


Textura.

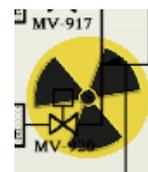
Los elementos grafico realizados con estas técnicas son:



Medidor de aguja.



Esquemas de dispositivo con translucencia.



Menús Rotatorios

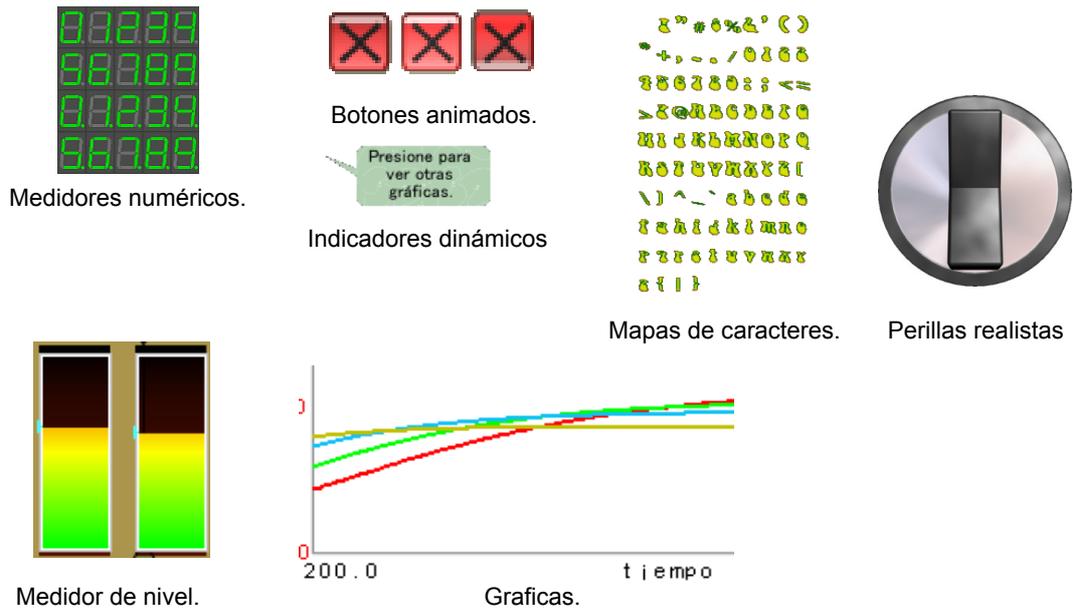


Figura 5.9: Galería de elementos.

Los medidores de aguja muestran el valor obtenido de Matlab, como un despliegue numérico y una posición de la aguja, esta posición depende del rango asignado al medidor y al valor.

Los esquemas con translucencia permiten observar los esquemas de dispositivos individuales, sin perder de vista el diagrama principal. Los menús rotatorios, permiten acceder a cualquier parte del simulador, son explicados mas adelante en este capítulo. Los medidores numéricos componen un numero con caracteres tipo "led", dependiendo del valor obtenido de Matlab.

Los botones animados, presentan una animación dependiendo de se el usuario ha presionado sobre ellos o no. Los indicadores dinámicos, siguen el movimiento del ratón presentando un mensaje que amplía una información o propone otra.

Los mapas de caracteres permiten generar un juego de caracteres completamente personalizable, o agregar símbolos definibles a las pulsaciones del teclado.

Las perillas, son texturas aplicadas a una superficie circular que rota dependiendo de los movimientos del ratón, sirven para ajustar parámetros de los dispositivos y sistemas.

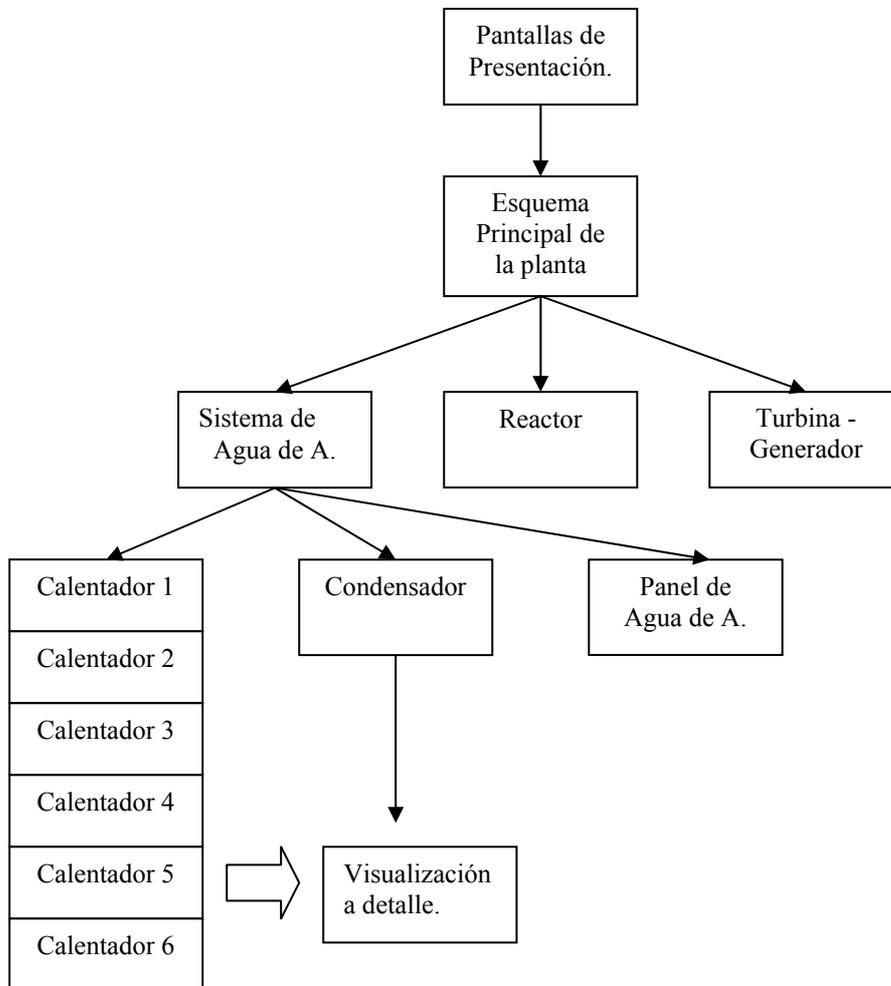
Los medidores de nivel, muestran un gradiente de color de acuerdo al valor dado y el rango determinado.

Las graficas muestran hasta cuatro graficas de una variable contra el tiempo, estas graficas son autoactualizables como se explico anteriormente.

Estos elementos conforman los despliegues los cuales se presentan de acuerdo a los datos que el usuario decida ver ya sea pulsando sobre los botones o través del menú.

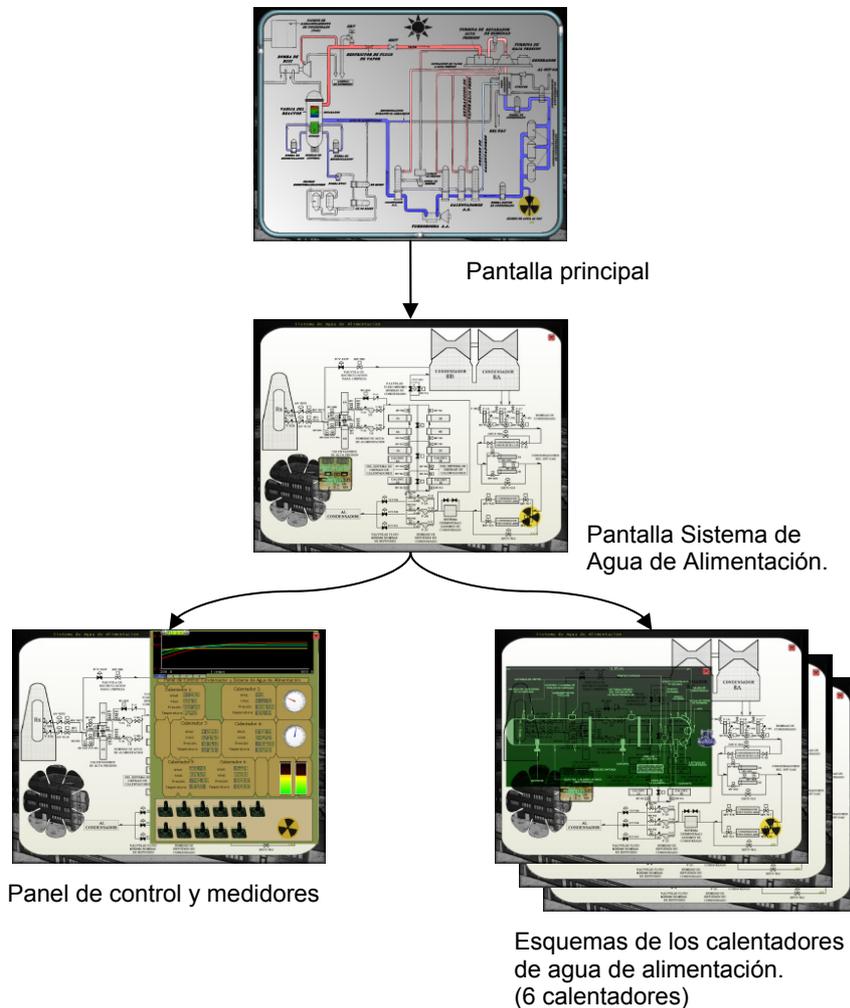
### 5.9. Interacción entre despliegues

Los despliegues de la interfaz gráfica del sistema de condensado y agua de alimentación, responden a una organización jerarquizada de acuerdo al siguiente esquema.



**Figura 5.10:** Diagrama jerárquico de la organización de las pantallas de interfaz en el simulador SUN-RAH.

La interfaz del sistema de agua de alimentación: Sistema de Agua de A. presenta la siguiente organización, donde cada rama es una posibilidad que tiene el usuario, en esta jerarquía se excluye el menú que permite acceder a cualquier pantalla.

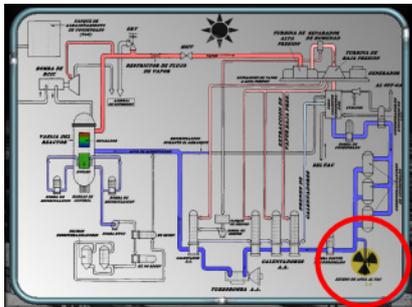


**Figura 5.11:** Organigrama de las pantallas del simulador de Agua de Alimentación y sus ramificaciones dependiendo de la selección del usuario.

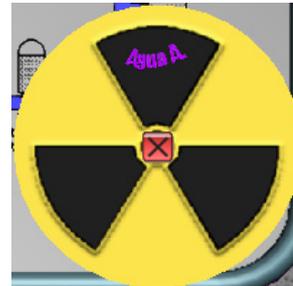
### 5.10. Menús.

Las Interfaces gráficas de usuario (GUI's) han empleado tradicionalmente los menús, para acceder rápidamente a las secciones donde el usuario desea modificar algún parámetro o visualizar algún dato. Los menús de Windows son estructuras ampliamente conocidas, los entornos de programación para Windows como Visual Studio de Microsoft, proveen una manera de crear estos menús casi automáticamente, por lo que es difícil tratar de romper la inercia y diseñar un menú diferente. El menú que propone el simulador SUN-RAH consiste en un círculo rotatorio con la forma del símbolo de la energía nuclear y que gira para proporcionar acceso a las diferentes pantallas, por medio de botones animados. Emplea texturas y sprites para lograr los efectos deseados y propone una manera atractiva

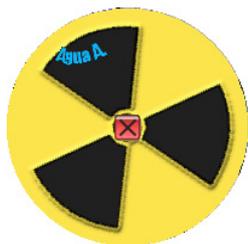
de desarrollar menús, para aplicaciones en general. La evolución de los menús, libera las restricciones propias de la estructura tradicional y permite diseñar menús acorde al tema de la aplicación, aunque la propuesta de un menú circular podría parecer inapropiada en otro tipo de aplicaciones o simulaciones, la libertad otorgada por DirectX, permite generar menús interactivos que permitan un mayor impacto con el público al que van dirigidos.



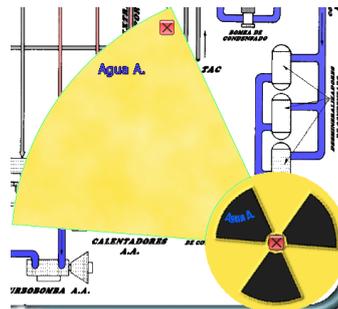
El menú permanece en la parte derecha inferior minimizado



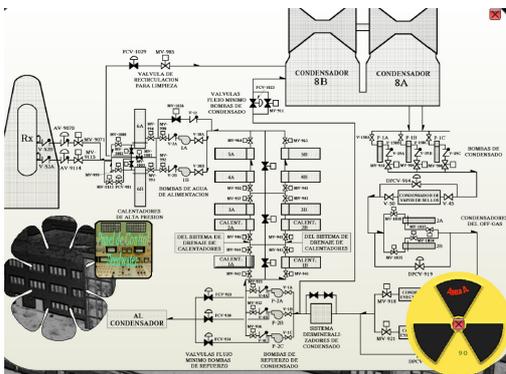
Al presionar sobre el menú minimizado se agranda el menú, mostrando los sistemas de los que se compone el simulador. (en este caso: agua de alimentación)



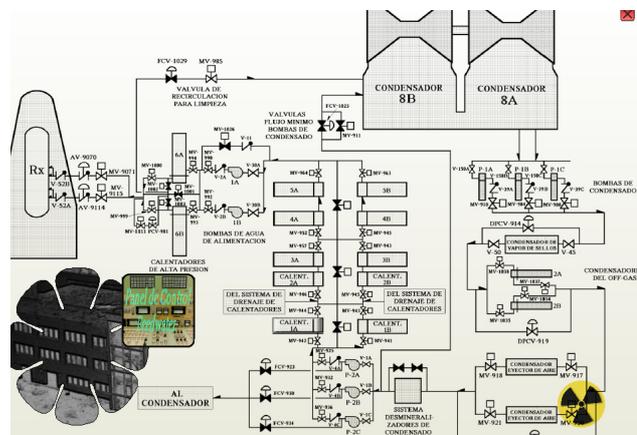
Al presionar sobre el sector deseado este gira hasta una posición donde pueda desplegar las opciones del menú.



El menú se despliega mostrando las opciones para la sección elegida.



Una vez elegida la opción del menú este regresa a su posición maximizada, esperando que el usuario se desplace a otra sección ó minimice el menú.



Si se decide minimizar el menú este regresa a su posición original esperando un nuevo comando.

Figura 5.12: Pantallas de ejemplo en el uso del menú gráfico.

### **5.11. Conclusiones.**

Las interfaces gráficas son un vínculo entre la teoría y la práctica. Son el reino de las sombras que nos sugieren objetos que se encuentran ocultos. Deben por lo tanto ser precisas, fieles y cuantitativamente amigables. La interfaz del sistema de agua de alimentación está diseñada para enfrentar al usuario a un tópico que le puede ser desconocido o que pudiera no haber visto en una representación gráfica interactiva anteriormente, por lo tanto, con la elección de un entorno de programación atractivo, se pretende atraer a un mayor número de estudiantes hacia la comprensión de temas nucleares, el modelado de sistemas, el diseño de controladores, la programación de interfaces, etc.

Evitar la aridez en las interfaces gráficas, permite un trabajo más cómodo con los dispositivos representados y alienta el deseo de comprender los principios a simular o incluso de mejorar las interfaces para materializar una visión propia de cómo algún tema debería explicarse o representarse.

El haber elegido un entorno de programación principalmente empleado en la creación de videojuegos, no es únicamente una decisión objetiva, también se basa en el deseo personal de conjuntar las técnicas desarrolladas para el entretenimiento con los modelos desarrollados por la ingeniería. Tanto el entretenimiento se ha visto enormemente beneficiado de los conceptos y teorías provenientes de la ciencia y la técnica, como también la tecnología puede beneficiarse de las capacidades demostrativas del entretenimiento.

Esto es claro en la industria de la computación, donde el mayor desarrollo de procesadores, compiladores, sistemas operativos, etc., se ha planeado para impulsar la industria multimedia. Así el empleo de interfaces gráficas en simulaciones puede verse beneficiado con el empleo de estas tecnologías, ya que una de sus finalidades es la de representar.

**Referencias bibliográficas.**

[1] Kjell Arne Barmsnes, Terje Johnsen, Carl-Victor Sundling, *Implementation of Graphical User Interfaces in Nuclear Applications*, Control Room Systems Development Division  
Presented at ENS Topical Meeting on I&C of VVER, Prague, April 21-24, 1997.

[2] Matlab documentación capítulo 6, *MATLAB External Interfaces*.

© COPYRIGHT 1984 - 2003 by The MathWorks, Inc. September 2003 Online only Revised for  
MATLAB 6.5.1

[3] Peter Walsh, *The Zen of Direct3D Game Programming*, California, Premier Press, 2002.

[4] Chris H. Pappas, William H. Murray III, *Visual C++.NET Manual de Referencia*, Madrid, Mc  
Graw Hill 1ª edición 2002.

[5] André LaMothe, *Tricks of the Windows Game Programming Gurus, Second Edition*, Indiana,  
Sams Publishing, 2002.

[6] Wolfgang F. Engel, *Beginning Direct3D Game Programming*, California, Premier Press, 2002.

## Capítulo 6. Prueba de los modelos y la interfaz.

### 6.1. Diseño de pruebas a modelos.

Para hacer la prueba de los modelos se plantean pruebas que indiquen su comportamiento general. Aunque las señales que se introducen de entrada a los sistemas no son siempre del tipo que manejarían en operaciones en estado estable en la simulación, estas señales de pruebas nos indicarán que tanto se altera la salida a las perturbaciones a una u otra entrada. De esta manera podemos determinar la importancia de la modificación de algún parámetro dentro de la configuración de los sistemas.

Para probar el sistema del condensador, empleamos una señal senoidal en el flujo que viene de la turbina  $W'_3$ , debido a las variaciones pequeñas que se esperan en el flujo proveniente de la turbina.

Se coloca un escalón en el flujo que viene de los calentadores  $W_{hp0}$ , para demostrar que el flujo depende fuertemente de este valor. Y una señal aleatoria en el flujo de agua de alimentación  $W_{fwT}$  extraído del condensador, debido a que también se esperan pequeñas variaciones en esta demanda. En el resto de las señales se colocan valores constantes debido a que sus valores en la planta son pequeños y no influyen demasiado en el flujo del condensador.

Para hacer la prueba de uno de los calentadores de agua de alimentación, se emplea una señal de escalón como entrada en el flujo que regresa al calentador posterior al de prueba, en esta caso  $W_{hp1}$ . Una señal senoidal que representa la extracción de la turbina de baja presión  $W_{bip4}$ , que oscila entre los valores típicos de esta extracción. El flujo de agua de extracción que será calentado  $W_{fwT}$  es representado por una señal senoidal para así observar el efecto de la extracción en la entalpía del agua de alimentación.

Para el resto de los calentadores el comportamiento es similar por lo que este banco de pruebas se extiende hacia el resto de los seis calentadores.

Para probar el controlador se fija el nivel de *Setpoint* de agua en el reactor a un valor constante típico. El valor de la demanda de la línea de vapor  $mISO_{Steamflow}$  se aumenta de valor en un rango lógico dentro de la demanda de una planta típica, para conocer la respuesta del controlador a un cambio abrupto en la demanda de vapor hacia el reactor. Las pruebas del controlador esperan mostrar como el controlador aumenta el flujo de agua de alimentación para mantener el nivel de agua en el reactor.

El modelado de la bomba de agua de alimentación es un sistema de primer orden, por lo que se probará con una señal senoidal, esperando obtener el retraso propio del arranque de la bomba, para esto se le coloca una señal senoidal en la entrada y se observa como se estabiliza el sistema.

Para hacer las pruebas del sistema en conjunto se emplean valores de una planta típica y se comparan con los obtenidos en la simulación.

También se hace una prueba de respuesta a alteraciones en las condiciones para mostrara la estabilidad del sistema y su respuesta a los eventos comunes.

## 6.2. Resultados de pruebas de los modelos individuales.

### 6.2.1. Condensador.

De acuerdo al diseño de la prueba para el condensador se colocan las señales en las entradas del modulo de flujo de condensado y se observa la salida.

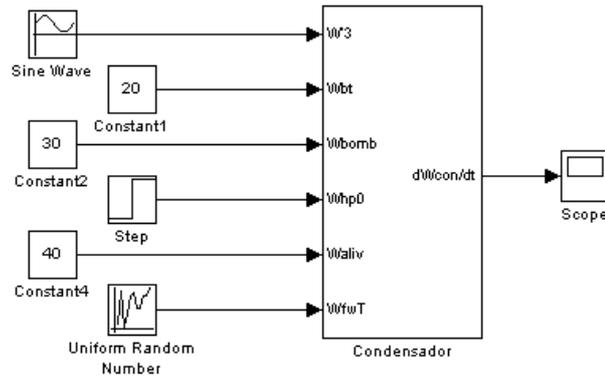


Figura 6.1: Esquema de prueba para el condensador.

De manera gráfica, mostrando las entradas y la salida:

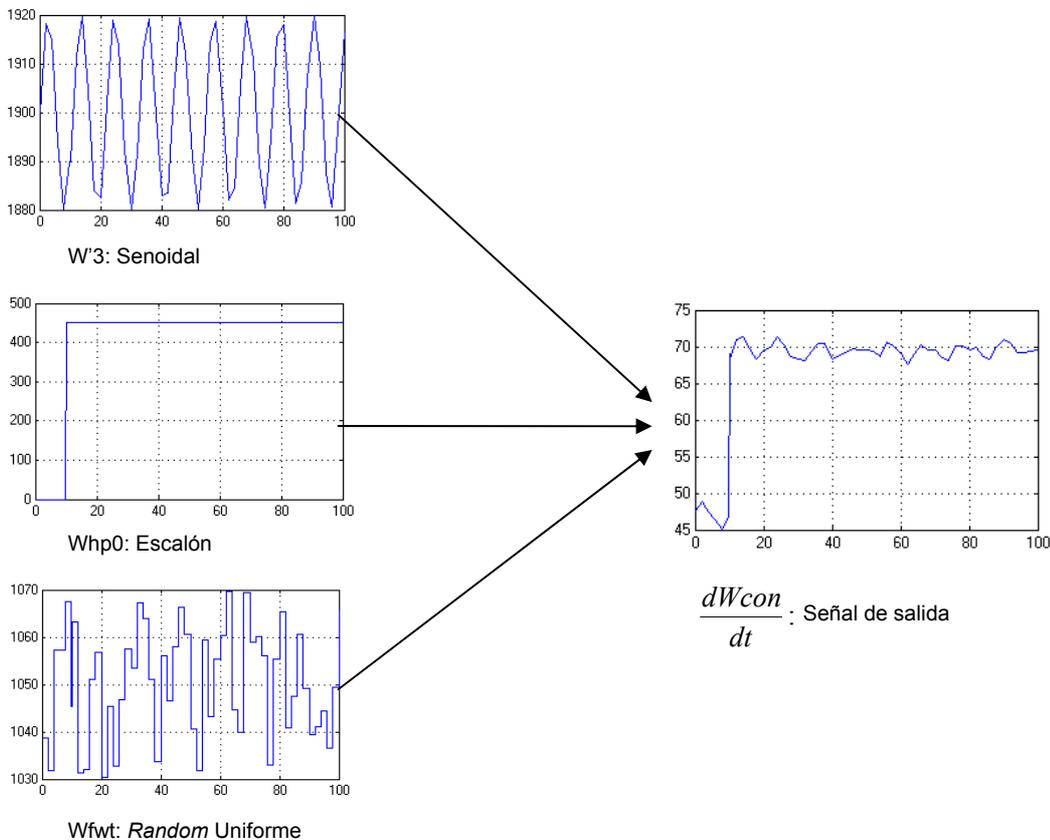


Figura 6.2: Señales significativas de entrada al sistema de condensado y su respectiva salida.

En esta gráfica se observa como la salida es producto de las señales de entrada, la componente senoidal del agua de las turbinas, junto con la señal aleatoria uniforme colocada como señal de flujo de agua de alimentación que se le extrae al condensador, crean la cresta irregular que se observa en la salida. Por su parte, el flujo de agua proveniente de los calentadores representado por un escalón tardío, provoca un mayor flujo interno a su llegada. Con lo que se confirma su importancia en el flujo interno del condensador.

### 6.2.2. Calentador de baja y alta presión.

Los calentadores de agua de alimentación de baja presión, emplean el agua de extracción de las turbinas y el flujo de agua de otros calentadores, para elevar la entalpía del flujo de agua de alimentación.

La siguiente prueba nos muestra que tan importante es el agua que se suministra desde los otros calentadores, y como influye en el agua que el propio calentador provee, así como en la entalpía del flujo de Agua de Alimentación.

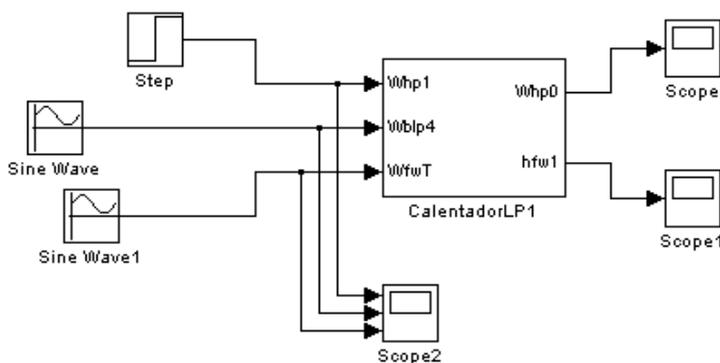
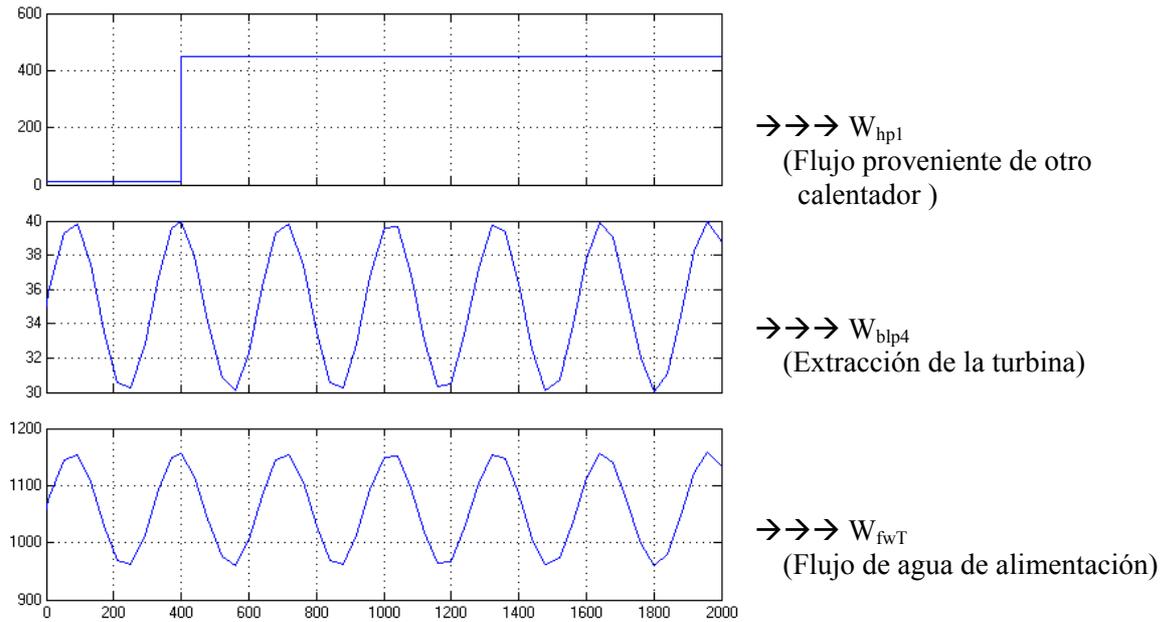


Figura 6.3: Esquema de prueba para un calentador de agua de alimentación.

Las señales de las extracciones de la turbina, así como del flujo de agua de alimentación, fueron puestas en valores predeterminados de la simulación, como señales senoidales, con una cierta amplitud que permitiera alterar notoriamente los valores fijos especificados. La señal de el flujo de otros calentadores fue aumenta drásticamente a través de un escalón a un tiempo determinado.

Las señales de entrada se muestran en la figura 6.4.

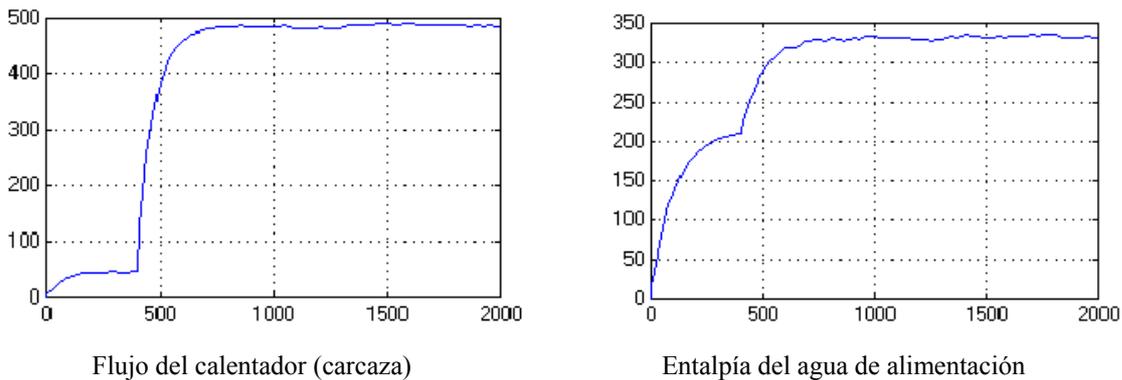


**Figura 6.4:** Entradas para la prueba de un calentador cerrado de agua de alimentación.

Las salidas del sistema son la entalpía que lleva el flujo de agua de alimentación hacia el reactor y el flujo de agua que este calentador proporcionara al calentador anterior, después de haberlo utilizado en la transferencia calorífica con el flujo de agua principal.

De acuerdo al comportamiento teórico de los modelos, la entalpía se verá afectada considerablemente al aumentar el flujo del otro calentador, ya que no solo depende del agua de la turbina para subir su entalpía. El flujo de que saldrá de este calentador, se verá notoriamente afectado ya que depende estrechamente del flujo que le proporcione el otro calentador.

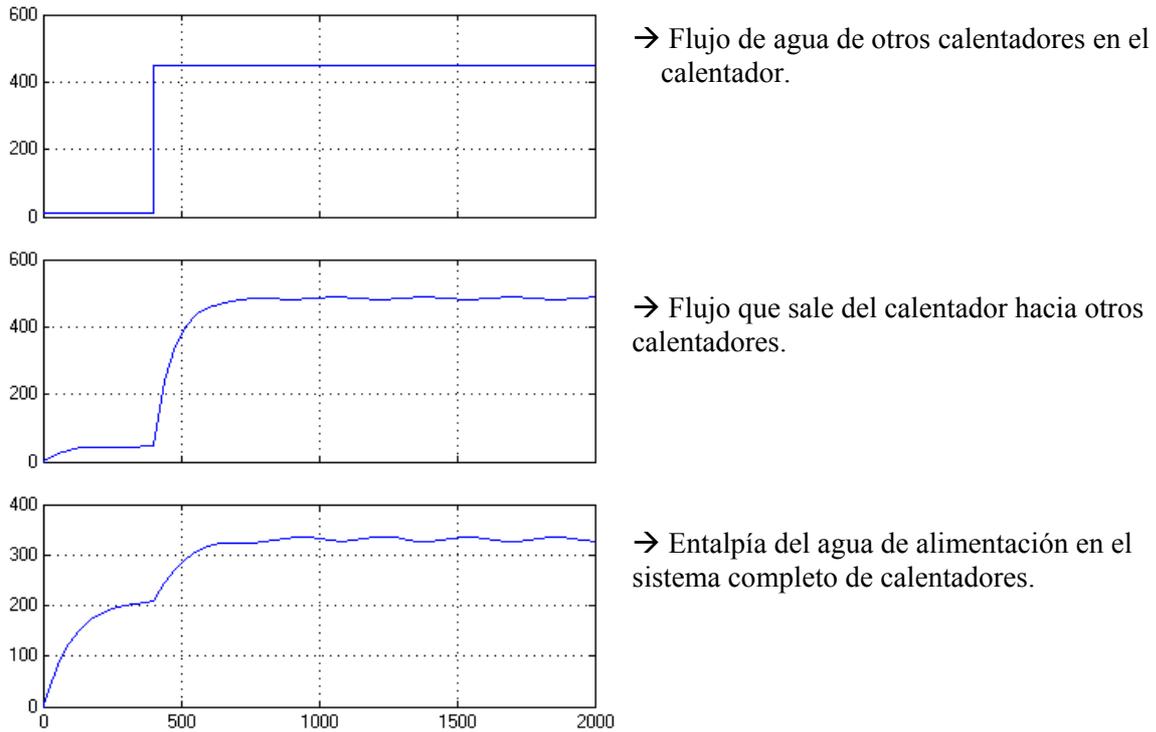
Al hacer la simulación, de acuerdo a los parámetros establecidos se obtuvieron las siguientes respuestas:



**Figura 6.5:** Salidas para la prueba de un calentador cerrado.

Podemos observar que al tiempo en que el nivel del flujo de otros calentadores aumenta, también lo hacen las señales de flujo de calentadores y de la entalpía de agua de alimentación. Por lo que

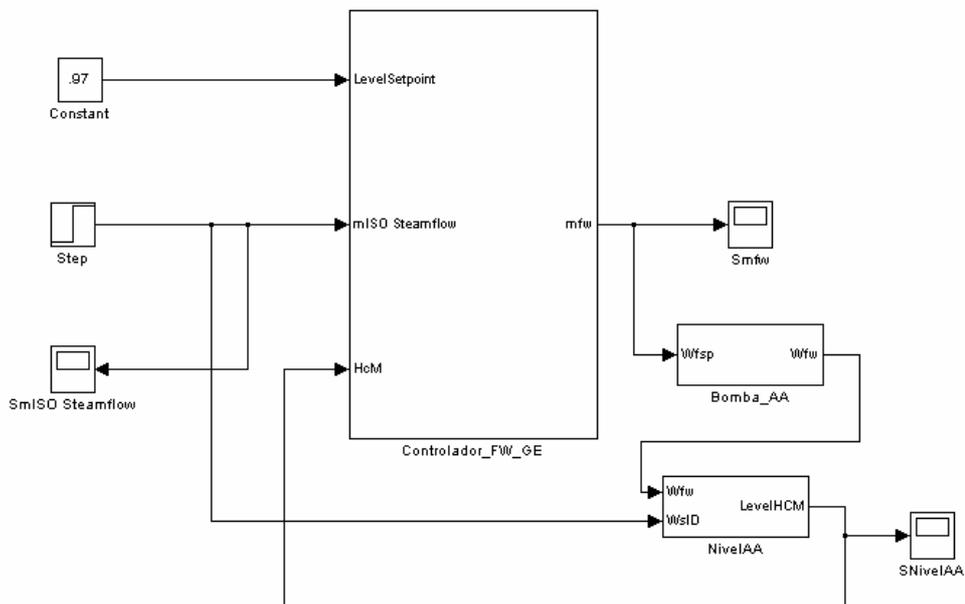
concluimos que el modelo se comporta para esta señal de manera adecuada de acuerdo con las suposiciones previamente establecidas al tiempo indicado.



**Figura 6.6:** Correlación entre la entrada y las salidas en la prueba de el calentador cerrado de agua de alimentación.

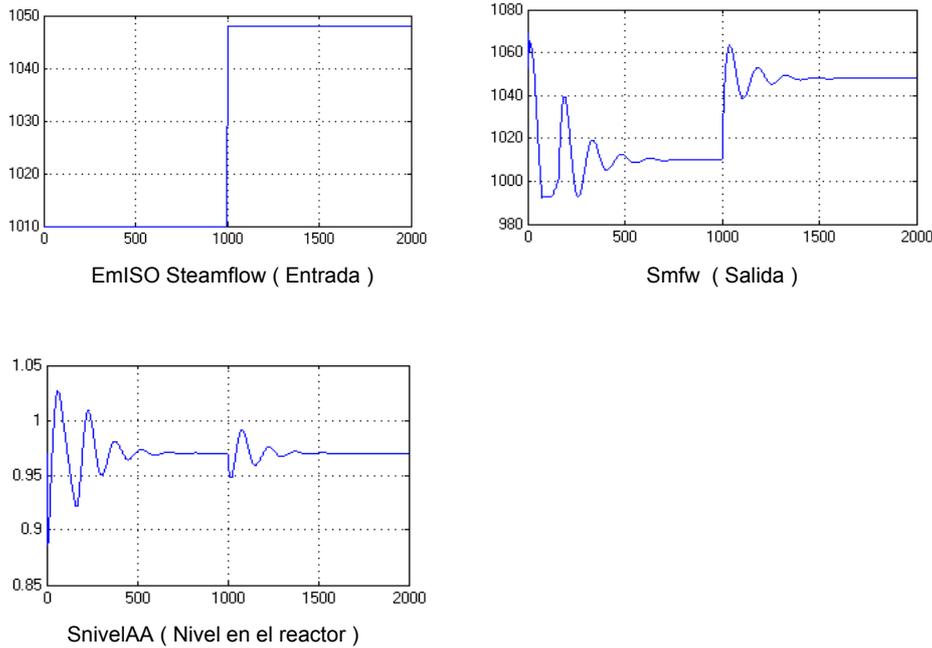
### 6.2.3. Controlador de inventario de refrigerante.

Como se mencionó en el diseño, las pruebas del controlador se hacen empleando los modelos de la bomba de agua de alimentación y un modelo sencillo del nivel en el reactor para obtener la respuesta aproximada al ajuste del controlador y sus efectos en los demás sistemas.



**Figura 6.7:** Esquema de prueba controlador de inventario de refrigerante.

Las señales que obtenemos son:

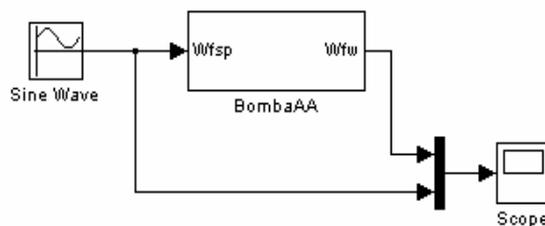


**Figura 6.8:** Señales de entrada y salida del controlador de inventario de refrigerante. También se observa la grafica del nivel en el reactor.

Se observa como se demanda un cantidad mayor de vapor, lo cual se simula como una entrada tipo escalón en donde de un requerimiento inicial se incrementa a un valor mayor. Debido al incremento en el vapor, el controlador ajusta el nivel de agua de alimentación, incrementándolo lo necesario para que el nivel de agua en el reactor se mantenga en el valor de *Setpoint* predeterminado. En la grafica del nivel del reactor se puede observar que el nivel del reactor se ajusta a los cambios y se estabiliza en el valor especificado por el *Setpoint*.

#### 6.2.4. Bomba de agua de alimentación.

El modelo de la bomba de agua de alimentación es un sistema de primer orden. Las pruebas consignan una sola señal de entrada y su salida correspondiente.



**Figura 6.9:** Modelo de la bomba de agua de alimentación.

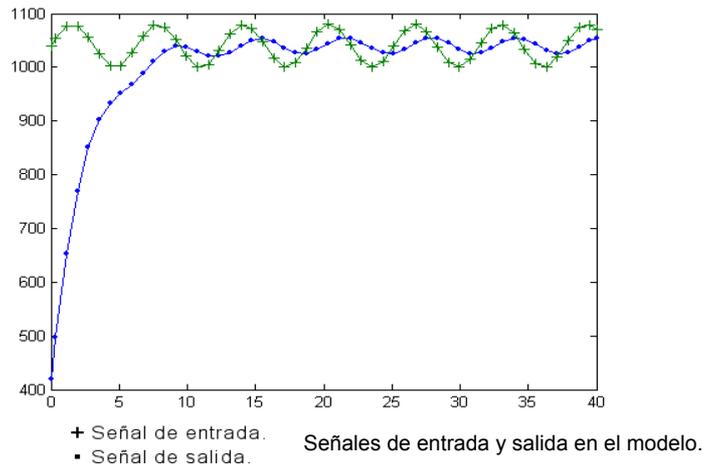


Figura 6.10: Comparación entre la entrada y la salida en el modelo de la bomba de agua de alimentación.

Como se describió en la parte de modelado el sistema es de primer orden. El modelo de la bomba de agua de alimentación agrega únicamente un atraso y redimensiona la señal del flujo que se le proporciona como entrada.

### 6.3. Pruebas a los modelos integrados.

#### 6.3.1. Calentadores de agua de alimentación dentro del sistema completo de agua de alimentación.

Para realizar las pruebas al conjunto dispositivos de la planta modelados, se ensambló una simulación de los calentadores tanto de baja y alta presión, la bomba de agua de alimentación, el controlador y un modelo básico del nivel de agua en el reactor. Los valores obtenidos de la simulación de este sistema serán comparados con los valores en un balance de una planta nuclear típica.

Las pruebas se hicieron en el siguiente modelo integrado:

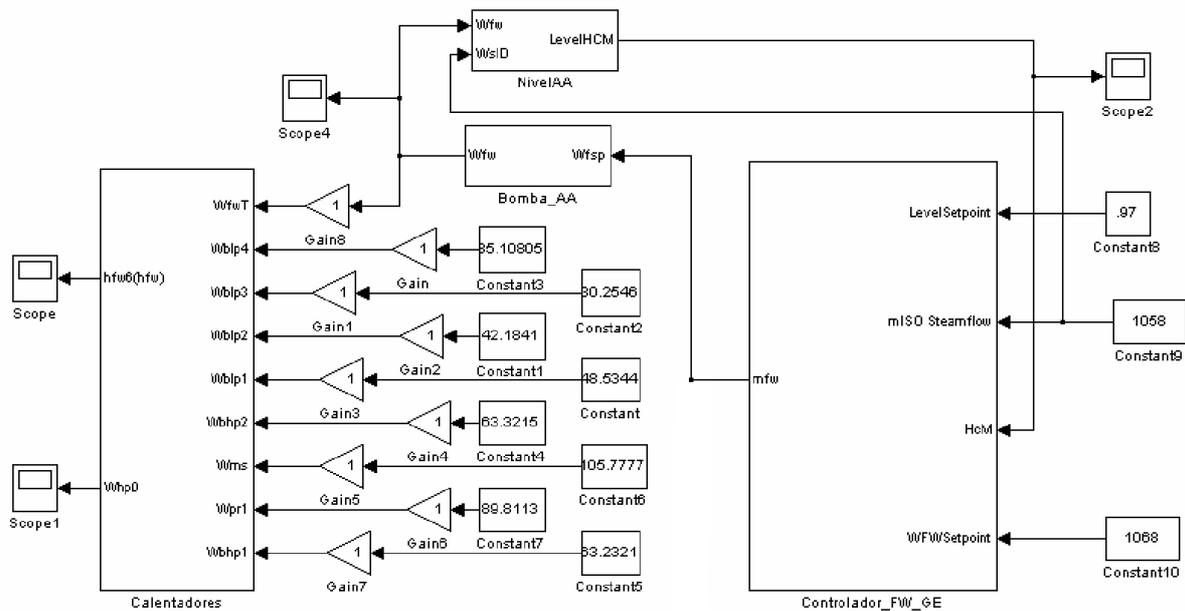
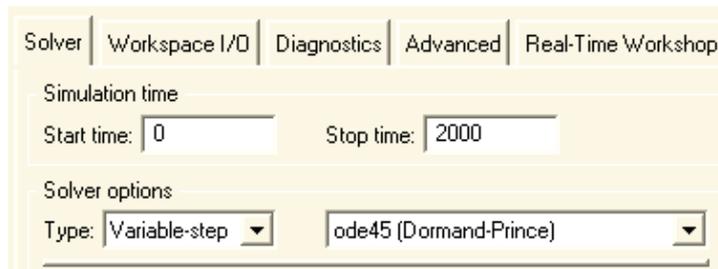


Figura 6.11: Esquema de pruebas de los calentadores, el controlador, la bomba, y nivel de agua en el reactor.

Los parámetros de la simulación elegidos para la prueba fueron los de un método de paso variable y un tiempo de 4000 unidades de tiempo.



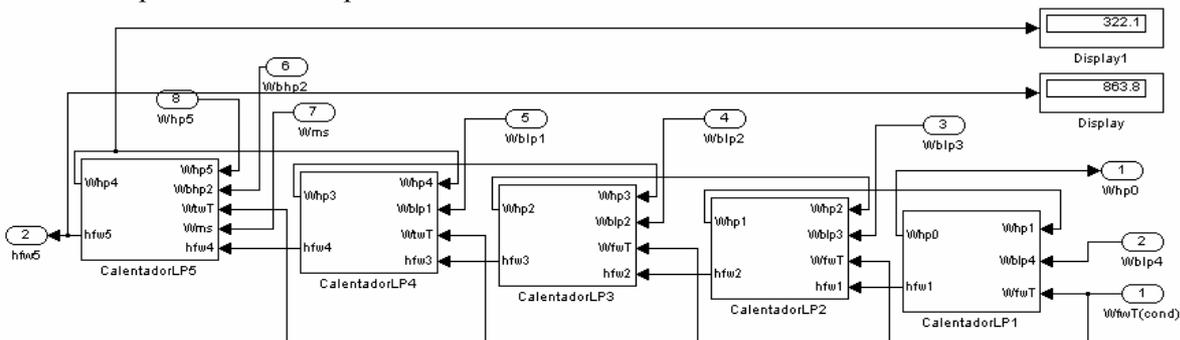
**Figura 6.12.** Parámetros de la simulación de prueba de los modelos de agua de alimentación.

Las entradas al sistema son las extracciones de las turbinas y son fijadas en valores típicos de la planta nuclear.

Variable de entrada (valor predeterminado)	Valor de la variable ( todos son flujos [kg/s])
WfwT: Flujo de agua de alimentación	1058
Wblp4: Extracción turbina baja presión	35.1058
Wblp3: Extracción turbina baja presión	30.2548
Wblp2: Extracción turbina baja presión	42.1841
Wblp1: Extracción turbina baja presión	48.5344
Wbhp1: Extracción turbina alta presión	63.2331
Wbhp2: Extracción turbina alta presión	63.3215
Wms: Extracción separador de vapor	105.777
Wpr1: Extracción recalentador.	89.8113
<i>Level Setpoint</i> : Nivel del de agua deseado en el reactor	0.97 [m]
<i>mISO Steamflow</i> : Flujo de extracción de vapor al reactor.	1058
<i>WFW Setpoint</i> : Flujo obtenido del sistema de agua de alimentación.	1069

**Tabla 6.1:** Valores de entrada para esquema de prueba.

Las salidas están representadas por graficadotes de Matlab llamados “scopes”. Alternamente se recogieron los valores en cada uno de los calentadores, para obtener los valores de entalpía y flujo necesarios para realizar comparaciones cuantitativas.



**Figura 6.13:** Toma de valores en los modelos de Matlab, en el proceso de pruebas a los modelos integrados.

Los valores comparables son la entalpía del flujo a la salida de los calentadores y la cantidad de flujo que ha sido utilizado por el calentador para calentar el agua de alimentación y que es entregada a un calentador anterior o al condensador en caso del primer calentador.

La comparación se hace entre los resultados de Matlab y los valores del balance de planta (BOP) de una planta nucleoelectrónica típica.

Entalpía a la salida del calentador. [kJ/kg]	Balance de Planta (BOP)	MATLAB. (Variable calculada)
1: $h_{fw1}$	312.3818	329.1
2: $h_{fw2}$	390.5354	415.3
3: $h_{fw3}$	514.046	533.3
4: $h_{fw4}$	656.39	661.9
5: $h_{fw5}$	801.307	863.8
6: $h_{fw6}$	928.3066	922.2

**Tabla 6.2:** Resultado comparativo de los valores de entalpía obtenidos a las pruebas de los modelos.

Flujo lado carcasa a la salida del calentador. [kg/s]	Balance de Planta (BOP)	MATLAB (Variable calculada)
1: $W_{hp0}$	480.2636	478.2
2: $W_{hp1}$	444.1576	443.1
3: $W_{hp2}$	412.7690	412.9
4: $W_{hp3}$	368.77	370.7
5: $W_{hp4}$	322.050	322.1
6: $W_{hp5}$	153.0421	153

**Tabla 6.3:** Resultado comparativo de los valores de flujo (lado carcasa) obtenidos a las pruebas de los modelos.

Como se puede observar los resultados varían ligeramente, esto es debido a que los parámetros de los modelos no corresponden a ninguna planta en específico. Sin embargo podemos afirmar al ver la similitud que los modelos se comportan adecuadamente de acuerdo a los valores de flujo de agua de alimentación y extracciones consignados.

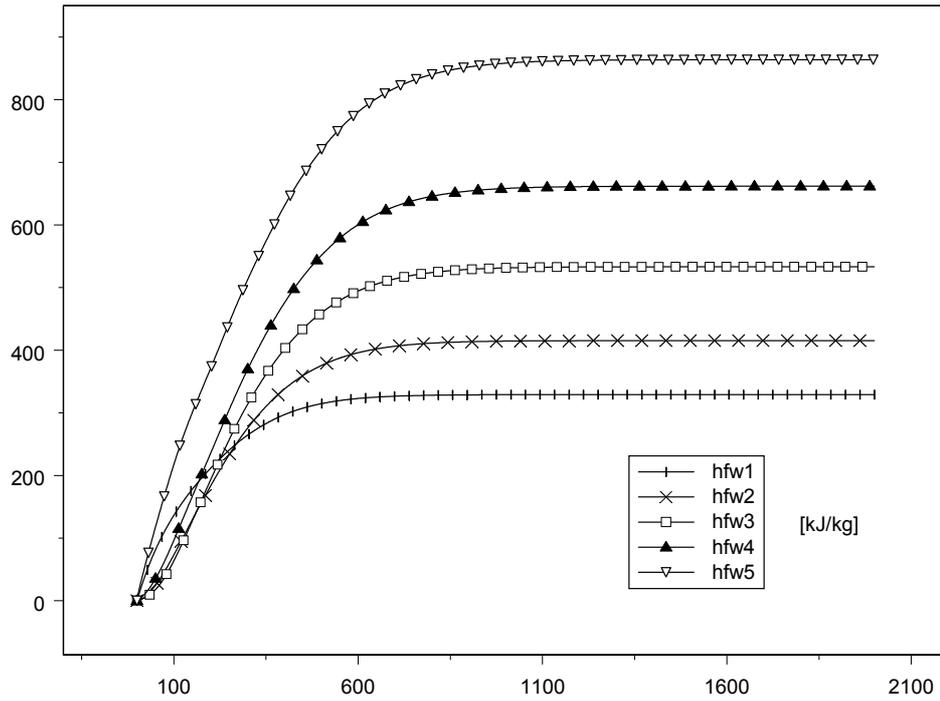


Figura 6.14: Grafica de la salida de la variable entalpía en los calentadores de baja presión.

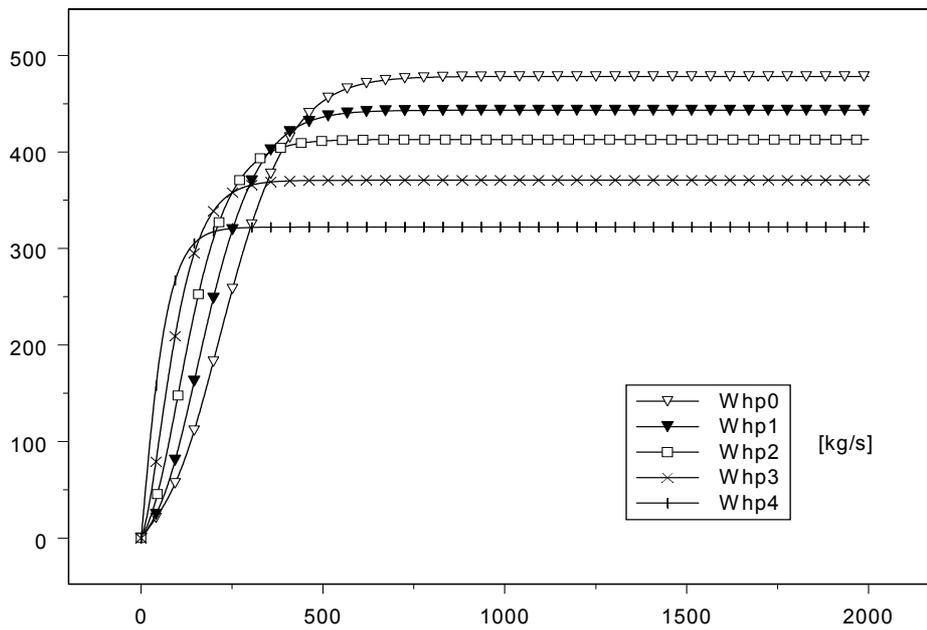
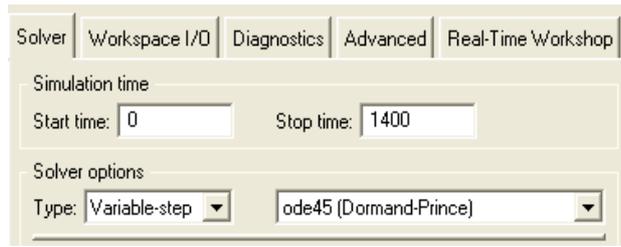


Figura 6.15: Gráficas de las salidas de los calentadores de baja presión.

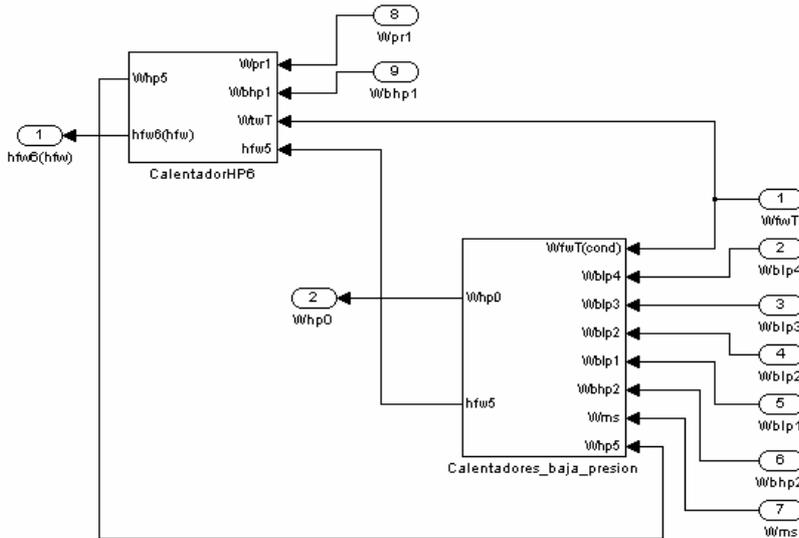
Para las pruebas al calentado de alta presión, se modificó el tiempo de simulación con el fin de observar mejor los parámetros comparables. Para la prueba los calentadores están integrados junto a los calentadores de baja presión, en la configuración descrita anteriormente.

Prueba de los modelos y la interfaz.



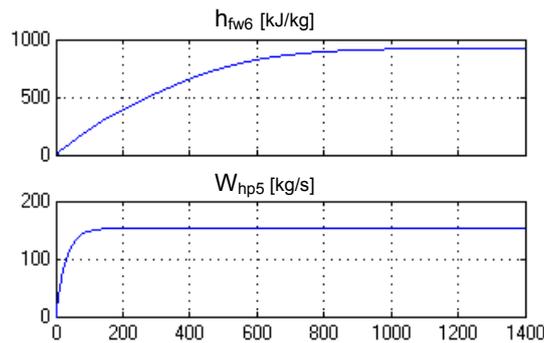
**Figura 6.16:** Parámetros para la simulación

La integración con el resto de los calentadores se realizó en la siguiente configuración.



**Figura 6.17:** Esquema de la integración entre los calentadores de baja y de alta presión

Los resultados de la entalpía y el flujo son:



**Figura 6.18:** Entalpía y flujo comparativos del calentador de alta presión.

La entalpía obtenida en el último calentador es la entalpía del flujo de agua de alimentación total y la conservará a la entrada del reactor.

### 6.3.2. Controlador de agua de alimentación dentro del sistema completo de agua de alimentación.

Al hacer las pruebas con los modelos del controlador se observa la estabilidad del sistema en una situación de estado estable típica de una planta nucleoelectrónica.

El esquema del sistema fue configurado de la siguiente manera:

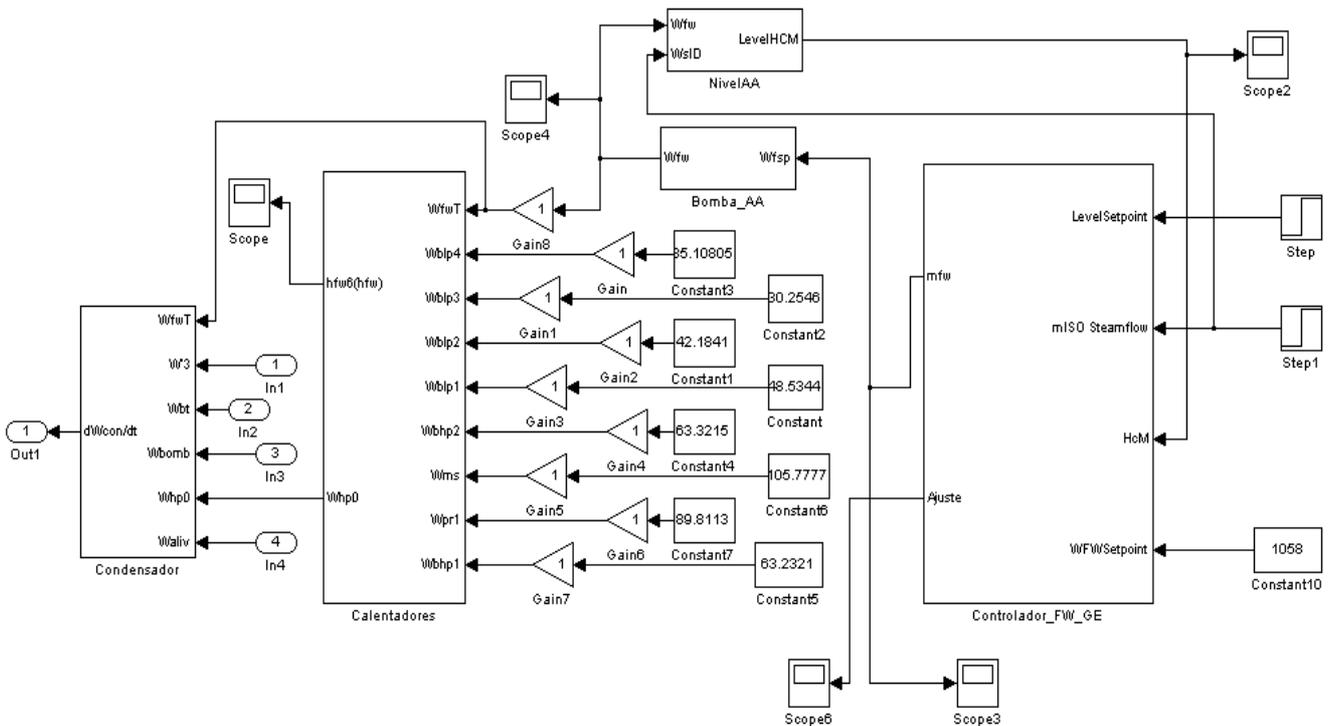
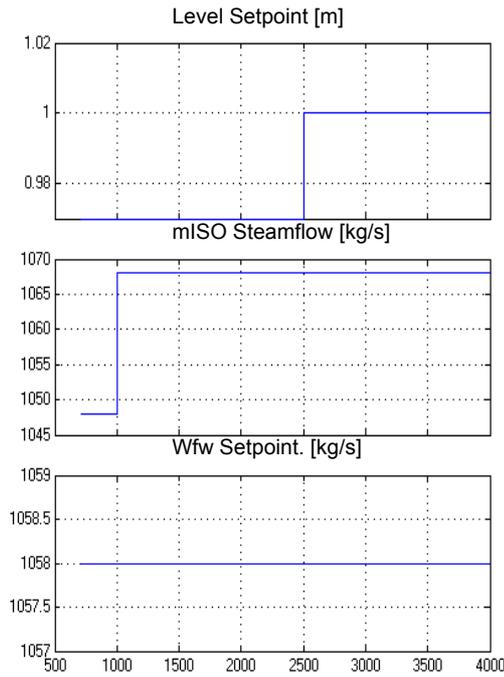


Figura 6.19: Configuración de prueba completa del sistema de agua de alimentación

Para hacer una prueba de los modelos del sistema de agua de alimentación en conjunto con el controlador se realizó la siguiente prueba, en donde se colocan los siguientes escalones a las entradas del *Setpoint* de nivel y de la demanda de vapor en el reactor. El nivel de agua de alimentación se mantiene constante a manera de que se observe como el controlador realiza el ajuste para incrementar o reducir la demanda de acuerdo a la demanda de la línea y el nivel en el reactor.

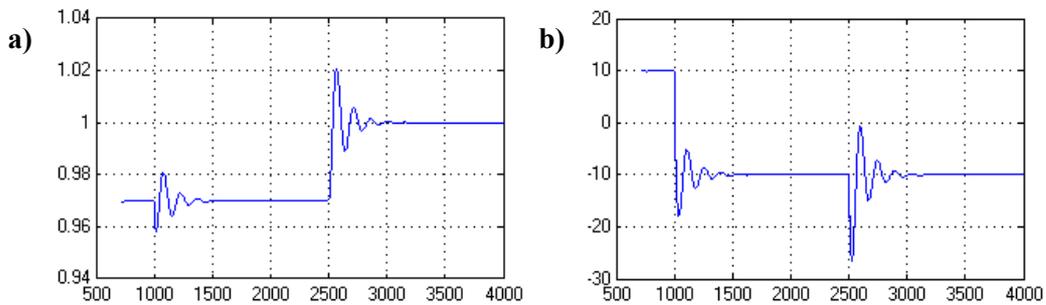
Las variables que se miden son las siguientes:

- $H_{cm}$ : Es el nivel de agua dentro del reactor calculado con el modelo de agua del reactor simple.
- $A_{ajuste}$ : Es una variable interna del controlador que no tiene relación con el exterior, pero que muestra el ajuste que hace el controlador al valor de  $W_{FWSetpoint}$ .
- $m_{fw}$ : Es el valor de flujo de agua de alimentación que ha ajustado el controlador y que manda a la bomba de agua de alimentación para que regule al flujo a ese nivel.
- $W_{fw}$ : Es el valor del flujo de agua que la bomba de agua de alimentación entrega a los calentadores.
- $h_{fw6}$ : Es la entalpía final que logra el sistema de agua de alimentación.

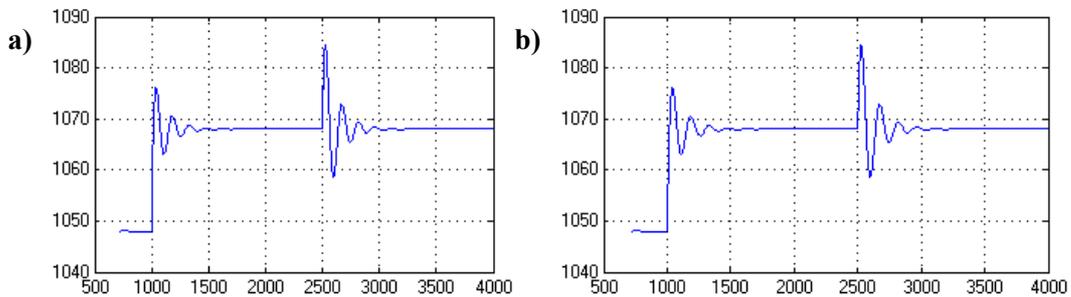


**Figura 6.20:** entradas para el modelo de pruebas

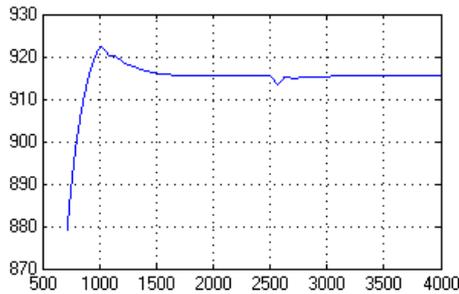
Las gráficas de las salidas muestran la estabilidad del sistema y las dimensiones de alteración o ajuste obtenidas:



**Figura 6.21:** a)  $H_{cM}$ : Nivel de agua en el reactor      b) *ajuste*: Variable del controlador



**Figura 6.22:** a)  $m_{fw}$ : Flujo de agua de alimentación que solicita se altere por la bomba de agua de alimentación.  
 b)  $W_{fw}$ : Flujo de agua de alimentación regulado que se envía a los calentadores.



**Figura 6.23:**  $h_{fw6}$ : Entalpía del flujo de agua de alimentación al final de el tren de calentadores.

Como se puede observar en las gráficas, el primer cambio ocurre en el tiempo 1000 y es el de una mayor demanda de vapor en la variable  $mISO_{Steamline}$ , lo cual repercute en el controlador ajustando el agua de alimentación disponible, para cumplir con la demanda del nuevo flujo. En las gráficas de ajuste se puede ver como el controlador ajuste el  $W_{fw}Setpoint$ , y en las graficas de  $m_{fw}$  y  $W_{fw}$  se puede ver como las variables se ajustan al nuevo valor al tiempo 1000 de la simulación. También se ve que el nivel de agua del reactor  $H_{cM}$  sufre una alteración debido a que este valor se considera dentro de los cálculos del controlador. Estos ajustes se ven como respuestas de un sistema de segundo orden debido a que el controlador emplea controladores PI, como se mencionó en el capítulo de descripción de modelos.

Al tiempo 2500 ocurre un cambio en el  $LevelSetpoint$  del nivel de agua en el reactor, requiriendo ahora un nivel superior al inicial, esto, provoca un incremento en el nivel  $H_{cM}$  que el controlador mostrara en el reactor y que incrementara ligeramente el flujo de agua de alimentación, por lo que los valores de  $m_{fw}$  y  $W_{fw}$  sufrirán alteraciones representadas como la respuesta a un sistema de segundo orden. Para después volver al estado estable determinado por la demanda de vapor en la turbina.

#### 6.4. Resultados de pruebas de integración entre los modelos y la interfaz.

Para probar al integración entre los modelos y la interfaz se presentan situaciones comparativas entre los resultados de la interfaz y los obtenidos directamente de los modelos en el entorno e simulación Simulink.

Una de las limitaciones de las simulaciones en Simulink es que la única modalidad de simulación que permite el cambio de parámetros en tiempo real es el modo "EXTERNAL", este modo permite conectar controles electrónicos a algún puerto de la computadora para alterar algún parámetro. En el caso de las simulaciones normales se requiere especificar entradas y tiempos de ejecución fijos, por lo que las alteraciones deben seguir patrones predeterminados.

En la interfaz estas alteraciones son el tiempo de ejecución, por lo que visualizar los efectos de alguna manipulación es inmediato.

Para mostrar la correlación entre los resultados de la simulación en Simulink y la de la interfaz se realizó una prueba en donde se colocó una señal tipo escalón en la entrada del flujo de Agua de alimentación en los calentadores. Esta señal cambia de un valor de 1058 a uno de 500 en el tiempo 300.

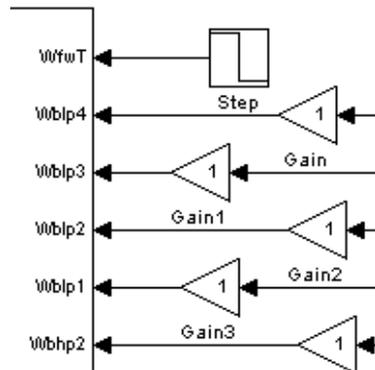


Figura 6.24: Entrada escalón para la prueba de correlación entre Simulink y la Interfaz Gráfica.

Las graficas comparativas de la entalpía final del flujo de agua de alimentación son:

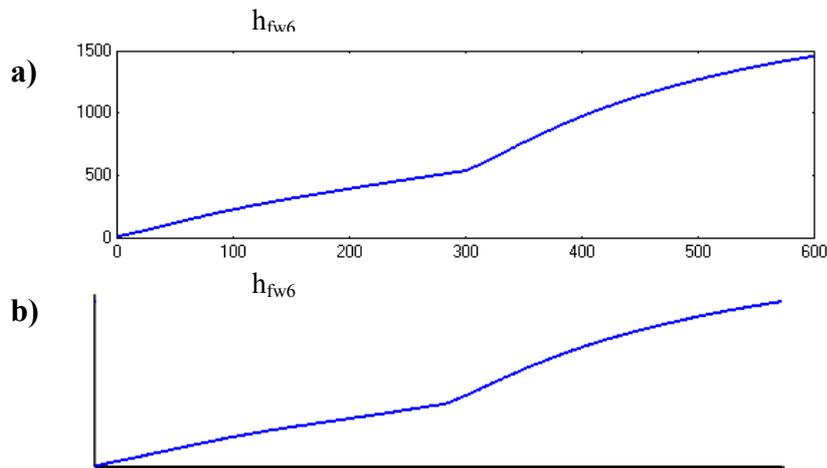


Figura 6.25: a) Gráfica de la entalpía del flujo de agua  $h_{fw6}$ . b) Gráfica de la entalpía en la interfaz gráfica.

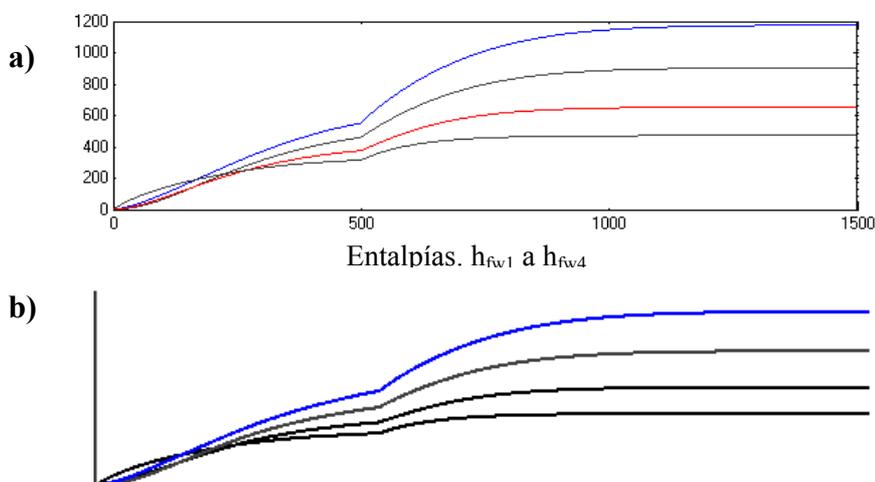


Figura 6.26: Prueba de correlación entre la simulación en Simulink y la Interfaz, que muestra la alteración de las entalpías en los calentadores.

Cabe mencionar que las alteraciones hechas a las variables en la interfaz se hicieron empleando elementos tipo perilla como se explica en el capítulo referente a la interfaz gráfica.



La perilla altera el valor en el panel de control, lo cual se refleja en la gráfica, en tiempo de ejecución.

Figura 6.27: Uso de las perillas en el panel de control para alterar los parámetros de la simulación en tiempo de ejecución.

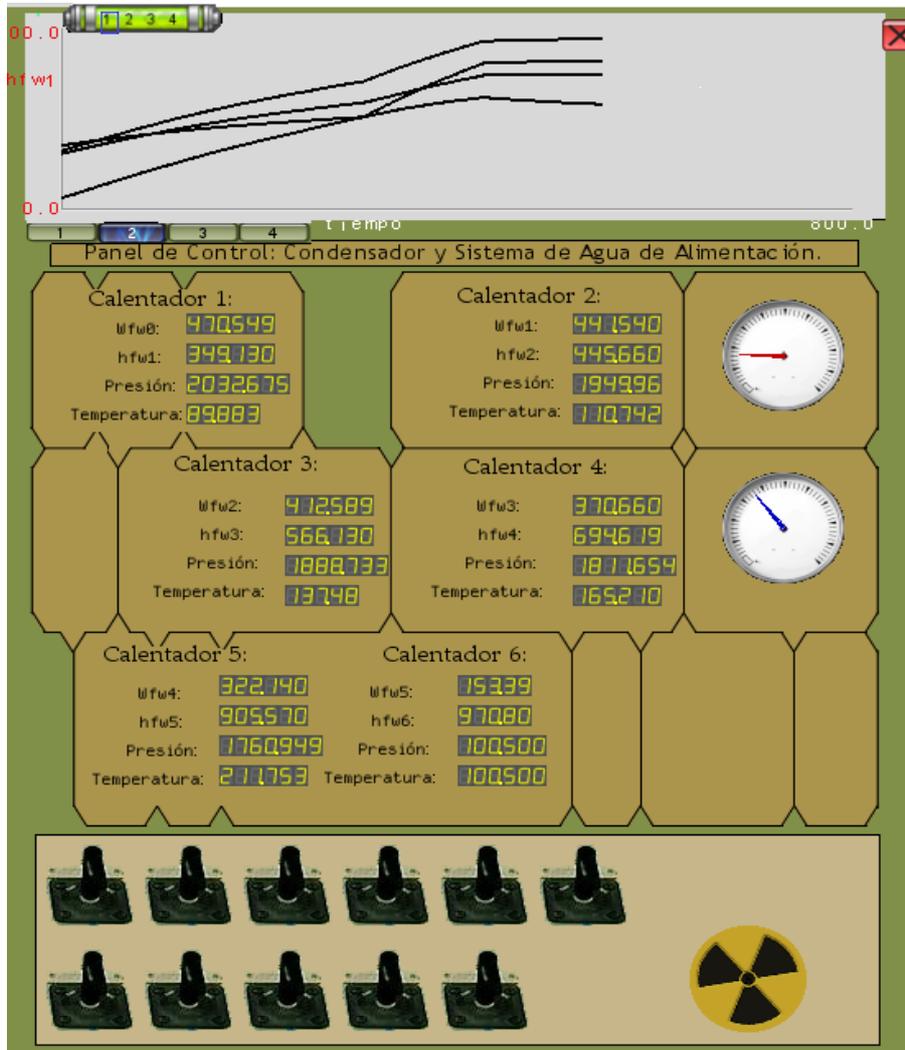


Figura 6.28: Panel de control de la simulación del sistema de agua de alimentación en el SUN-RAH. Las perillas se encuentran abajo y modifican parámetros e la simulación, las gráficas se ven en la parte de arriba. En el medio se pueden ver diversos datos de la simulación en forma de displays o de medidores de aguja o de nivel.

## **6.5. Conclusiones.**

Las pruebas a los modelos se hicieron una vez trasladados a Simulink. Así que las pruebas fueron programadas y efectuadas con las señales de entrada y tiempos predeterminados, dando resultados correctos. Las pruebas a la interfaz sufrieron un proceso diferente de pruebas, ya que cada que se agregaba un elemento o se cambiaba algo, se corría una prueba que verificara el funcionamiento del elemento gráfico y cuando este estaba correcto, se ejecutaba otra prueba para verificar si el elemento no afectaba a otras unidades o a la simulación.

El proceso de desarrollo de la interfaz fue un proceso de incertidumbre en gran medida, debido a que el empleo de esta tecnología era nuevo para nosotros. El aprendizaje en cada ciclo de prueba-corrección, genero la confianza para continuar el desarrollo y para establecer un motor gráfico sólido en el cual, cualquier futuro desarrollo puede basarse o complementarse.

Aunque aun falta mejorar la velocidad y la fiabilidad de algunos procesos de cambio de parámetros.

La interfaz cumple en cambiar los parámetros de una simulación en Matlab. En reanudar una simulación detenida con estos nuevos parámetros, de extraer los valores de varias de las salidas del modelo y de presentarlas en una ventana de gráficos autoactualizable. Permite también modificar parámetros interactuando con elementos gráficos, con cajas de texto de desarrollo propio, de mostrar jerarquías, translucencias, etc. que proveen una mejor experiencia al usuario en el conocimiento y entrenamiento de la operación de una planta de potencia.

## Capítulo 7. Conclusiones.

El simulador de condensador, sistema de agua de alimentación y controlador de inventario de refrigerante, forma parte de un simulador más completo: el Simulador Universitario Nuclear de reactores de agua hirviendo SUN-RAH. En un principio se pretendía hacer ese simulador de manera integral, para ser presentado como un trabajo de tesis único, aunque desarrollado por diversos estudiantes. Sin embargo, diversos cambios surgieron durante el proceso de diseño y desarrollo que propiciaron la separación en tres sistemas integrantes de la planta nucleoelectrica, y que serían simulados por separado: el sistema de condensado y agua de alimentación, el sistema de turbina y generador, y el reactor.

Las razones de la separación fueron variadas, destacan las relativas a la ausencia de parámetros confiables de los dispositivos nucleares reales, debido a que los organismos o empresas que manejan estas plantas restringen el acceso a este tipo de datos, por motivos de seguridad. Este hecho retrasó los procesos de prueba y validación de los modelos, pues al no tener respuestas intermedias en los componentes de un sistema, los modelos tenían que refinarse basándose en las respuestas de un sistema mayormente integrado.

En este trabajo además del modelado del condensador principal, el sistema de agua de alimentación y su controlador principal, se desarrolló la interfaz gráfica del SUN-RAH, tablas de vapor, estructuras de datos, métodos de programación, etc., lo cual propició una especialización en otras áreas, código extra de computación, metodologías nuevas, etc. Esta información y nuevas dimensiones en el proyecto, propiciaron la separación de los trabajos de tesis, con el fin de que cada documento pudiera abarcar diferentes espectros de desarrollo, sin atarse a la referencia de un tema único. Aunque el simulador no se presenta en configuración conjunta, en uno de los apéndices de este trabajo, se menciona la operación funcional con valores cercanos a los de una planta nucleoelectrica típica del sistema de Condensado, Agua de alimentación y controlador propio, con el sistema e Turbinas, Recalentador y generador.

El SUN-RAH es un proyecto que debe incluir modelos de otros sistemas de una planta típica y quedan por desarrollarse modelos de sistemas de protección y de seguridad de una nucleoelectrica así como pruebas integradas con el reactor, y la afinación de detalles en las pruebas al sistema conjunto. El simulador no está aún completo, existen varios tópicos en los cuales los sistemas podrían mejorarse, refinarse o remplazarse para buscar un mejor funcionamiento, facilidad de depuración etc. Los modelos matemáticos del sistema de agua de alimentación pueden refinarse de manera que devuelvan las salidas considerando los tiempos adecuados para los transitorios diversos de la operación de la planta en estados de emergencia, paro, arranque, etc. La interfaz gráfica es un campo abierto para: la exploración, la implementación de métodos de

programación, las teorías de graficación, la representación de fenómenos físicos en forma gráfica tales como la dinámica de fluidos, materiales, etc.

La interfaz gráfica esta basada en uno de los componentes más modernos, por lo que se beneficia de los desarrollos en otras áreas, los programas de graficación y animación como 3dMax, Maya, Softimage XSI, que son empleados en la animación en películas, videojuegos comerciales etc. Estos programas aparte de su funcionalidad evidente, proporcionan funciones complejas como física de partículas, cuerpo rígido y elástico, mecánica de fluidos con diversos materiales y sustancias, entre otros, y pueden ser empleados en el desarrollo de animaciones fotorealistas interactivas que representen los fenómenos en el reactor, el condensador, la turbina, etc. La interfaz por su parte puede ser rápidamente adaptada a nuevas teorías de la programación como son las tecnologías orientadas a objetos, STL, COM+, herramientas CASE. Estos posibles desarrollos permitirían componentes reutilizables en otros contextos, de manera que la creación de simuladores en otras áreas podría encapsularse en sistemas de bases de datos, iteradores, adaptadores, contenedores, etc.

Las tablas de vapor desarrolladas para este trabajo son adaptaciones de códigos de tablas anteriores y de documentos de tablas recientes cuya codificación no está disponible gratuitamente o únicamente en versiones *shareware* de uso limitado. Otro posible campo de desarrollo es emplear el desarrollo iniciado y completarlo para el resto de las condiciones del agua y otros materiales de manera que se tengan tablas de vapor a la medida de las varias necesidades termodinámicas de la universidad. De manera personal, el diseño de este simulador representó para mí un reto inusitado. El generar una estructura organizada de sistemas cuyas reacciones representaran dispositivos con los que no estoy familiarizado, fue una labor extensa y harto intimidatoria, he de confesar. Para equilibrar mis fuerzas internas recurrí a una de mis pasiones confesables y desarrollé una interfaz gráfica, con los elementos que se emplean en los videojuegos. Durante el estudio en la carrera, extrapolaba los conocimientos que recibía hacia alguna aplicación interesante para mí. Considero que en muchos casos los estudiantes hacen lo mismo, y creo que mezclar los mundos de lo académico y el entretenimiento, es una de las mejores maneras de conservar el ímpetu creativo que permite sobrellevar los retos, en pos de un mejor diseño y desarrollo

## Apéndice A.

### Tablas de vapor.

#### Tablas de Vapor versión 1.

*Basadas en: Steam / water properties. The American Society of Mechanical Engineers (ASME), 1967.*

En la fase de diseño y especificación de los modelos que serían parte del simulador SUN-RAH, se observó la necesidad de contar con una implementación en forma de programa o módulo, que pudiera ser utilizada para obtener las propiedades del agua, ya fuera en estado de saturación o de vapor sobrecalentado.

La solución que encontramos fue la de incorporar unas tablas de vapor, en forma de un programa que en respuesta a ciertos parámetros, regresara los valores de las propiedades requeridas.

Como parte de los requisitos estas tablas deberían poder ser ejecutadas dentro del programa Matlab, debido a que la implementación de los modelos y su ejecución se realizarían también en este entorno matemático.

Existen librerías y módulos de Matlab comerciales, que contienen las características necesarias para ser utilizadas por los programas, sin embargo su costo es excesivo y las versiones de evaluación o de estudiante, ofrecen un tiempo de ejecución reducido o una interfase mínima que hace imposible su aplicación práctica.

Debido a que varios de los modelos requerían de estas tablas de manera intensiva, así como también eran necesarias para verificar el funcionamiento mínimo de los modelos, se buscó una alternativa viable, tanto por su rápida implementación, como por su costo accesible.

Se encontraron, los programas desarrollados por Michael Lynn McGuire, [1] en donde se codifican las funciones que dan las diferentes propiedades del agua en diversos estados. Estos programas fueron codificados en lenguaje C y eran compatibles con el sistema operativo Windows. El autor de estos programas especificaba que al ser compilados, proporcionaría una serie de librerías dinámicas que podrían ser usadas en diferentes programas para Windows, programados en algún lenguaje compatible con las especificaciones de librerías dinámicas DLL. Sin embargo debido a la modularidad, y a la adecuada documentación dentro del cuerpo del código de los programas, fue posible separarlas en programas individuales, que calcularan diferentes propiedades de manera independiente.

Las propiedades que se necesitaban dentro de los modelos del simulador, fueron seleccionadas de los programas que se tenían y se preparo una lista de propiedades que se requerían.

- Temperatura de Saturación a una presión especificada.
- Presión de saturación a una temperatura especificada.
- Densidad del líquido.
- Entalpía del líquido.
- Entropía del líquido.
- Densidad del vapor.
- Entalpía del vapor.
- Entropía del vapor.
- Derivada de la densidad del líquido respecto a la presión.
- Derivada de la densidad del vapor respecto a la presión.
- Derivada de la entalpía del líquido respecto a la presión.
- Derivada de la entalpía del vapor respecto a la presión.

Los programas originales están basados en las especificaciones de 1967 de la “The American Society of Mechanical Engineers (ASME)”, y las unidades que manejan son las del sistema métrico ingles, pro lo que se hicieron las conversiones pertinentes al sistema internacional (SI) dentro del cuerpo de las funciones, por lo que estas se alteraron de su formato original.

### **Integración con Matlab.**

La integración con Matlab se hizo empleando las facilidades proporcionadas por el modulo de Matlab conocido como Matlab Compiler. Se emplearon las interfaces MEX [2], siguiendo los modelos e indicaciones que se proporcionan en los ejemplos, de este tipo de interfaces en la documentación de Matlab.

Este tipo de interfaces crean una “función puente”, entre Matlab y el compilador de “C”.

Esta función genera una serie de estructuras de datos compatibles con el Workspace de Matlab, y llama a funciones de lenguaje C. Los valores que las funciones regresan son convertidos por la función y de acuerdo a las instrucciones del programador, colocados dentro de una estructura válida de Matlab.

La “función puente”, y todas las demás funciones que necesita, en este caso los métodos que calculan las propiedades del agua, generan un programa autónomo. Este programa no genera código relacionado con el interprete Windows shell, que es el que ejecuta los programas en el sistema operativo, sino que es un programa de Matlab creado en el lenguaje C.

La compilación de este tipo del programa debe hacerse desde el entorno Matlab, el cual invocará al compilador, y le proporcionará los comandos y características propias necesarias para hacer una librería dinámica del tipo DLL (Dynamic Link Library).

Los parámetros y estructuras definidos en la “función puente”, son los que se requerirán para llamar a las funciones dentro de Matlab, y para recibir los resultados que se obtengan de ellas.

Las funciones pueden ser llamadas en cualquier ventana de comandos de Matlab, donde pueden ser empleadas en cualquier proceso o función que ejecute el usuario, incluso en aplicaciones como Simulink o proporcionadas por terceros.

### **Consideraciones.**

Las tablas construidas de esta manera, tienen una desventaja, respecto a las nativas de matlab, o las que están escritas en C y ejecutadas por el sistema operativo. Matlab, agrega muchas verificaciones y estructuras innecesarias, al menos hasta la versión 3.0 del compilador de Matlab, la optimización de las funciones es mínima, esto repercute en la velocidad de ejecución de las funciones de las librerías creadas de esta forma, lo cual puede ser un problema en los programas que hacen uso intensivo de estas funciones, como en el entorno Simulink. La reducción de tiempo es considerable si existen retroalimentaciones, o si se usan los valores que regresan están funciones para actualizar parámetros de algún método que a su vez use la misma función. De hecho varios de los modelos del simulador SUN-RAH, fueron afectados en velocidad por el uso intensivo de estas funciones, siendo este factor el de mayor importancia en la rapidez de su ejecución.

Cabe mencionar aquí que existe otra manera de hacer implementaciones externas de las tablas de vapor. Los módulos COM (Component Object Model), del sistema operativo Windows, proporcionan una manera de encapsular métodos dentro de componentes autónomos, con interfaces estandarizadas por el propio sistema operativo. Estas interfases son poco flexibles, pero proveen una compatibilidad total con e sistema operativo, con lo que inclusive programas escritos en oros lenguajes tales como Java o Visual Basic, pueden emplearlos de manera directa y sin conocer su código interno.

El programa Matlab, provee la facilidad de generar por medio de un compilador externo, componentes COM, y utilizarlos dentro de sus propios programas y aplicaciones.

Las tablas de vapor no emplearon esta tecnología debido a su elevado tiempo de desarrollo, que no solo incluye el tiempo de razonamiento y diseño de las interfases, sino también considera el tiempo de aprendizaje de esta tecnología, que sufre constantes revisiones de acuerdo e los nuevos compiladores en el mercado. Sin embargo no descartamos la posibilidad de robustecer el desarrollo de tablas de propiedades, del proyecto del simulador u otros proyectos, implementado las tablas como componentes COM, lo cual seguramente proporcionara una mayor velocidad de ejecución, una mejor integración con el entorno Windows y sólida estructura de aprendizaje en el desarrollo de simuladores.



```
A = vapt(700,500)          Rango temperatura °K { 273.16, 1144.26}
|                          Rango Presion Pa { 611.4, 41368540 }
|                          |
|                          |_____ (Presión, Temperatura) <---- *Atención*
|
|_____ Matriz de salida (necesaria)
```

Esta función da los valores v, h, s dados la presión y la temperatura la hice así, por que así está en el libro donde verifique si estaban bien los valores.

Y da los valores en una matriz de 4 renglones, una columna, así:

```
1      Error (Si = 1 / NO = 0) (default = 0 )
2      ro (densidad del liquido o vapor)
3      h
4      s
```

(las derivadas se pueden poner fácilmente)

```
A = vaphp(9000,3000)      Rango Entalpía J/kg { 1022.8(variable), 4291600}
|                          Rango Presion Pa { 0611.4, 41368540 }
|                          |
|                          |_____ (Presión, Entalpía) <---- *Atención*
|
|_____ Matriz de salida (necesaria)
```

Esta función da los valores t, ro, s dados la presión y la entalpía

Y da los valores en una matriz de 4 renglones, una columna, así:

```
1      Error (Si = 1 / NO = 0) (default = 0 )
2      Temperatura
3      ro (densidad del liquido o vapor)
4      s
```

(las derivadas se pueden poner fácilmente)

Para Compilarlas deben estar en el directorio.

C:\MATLAB6p5\work u el directorio work de otra versión de Matlab

Y se debe poner

```
mex -v -g vaphp.cpp      u otro archivo.cpp
```

con lo que se creará un archivo .dll en el directorio work.

Con lo que se pueden llamar las funciones como se menciona arriba del texto.

## Tablas de Vapor, versión 2.

**Basadas en: *The International Association for the Properties of Water and Steam (IAPWS), 1997.***

Las tablas que se desarrollaron para su uso en Matlab, tenían ciertas limitaciones, como se mencionó anteriormente. El uso de tablas de vapor en algunos métodos internos del simulador requería una respuesta más rápida. Debido a que los cálculos y procedimientos realizados con estas tablas no se involucraban directamente con procesos de simulación realizados por Matlab se generaron nuevas tablas que se ejecutarían directamente en el código del simulador escrito en C. Las tablas desarrolladas realizan operaciones de cálculo de presiones y temperaturas de acuerdo a parámetros conocidos de entalpía y flujo que se obtiene de los cálculos realizados en Matlab, así que sería redundante enviar estos datos a Matlab para su proceso y después regresarlos a el simulador escrito en C y ejecutado por Windows.

Para el desarrollo de las tablas se optó por no utilizar las mismas funciones empleadas anteriormente, aunque estas estaban escritas en C, sino que se decidió generar nuevas funciones, basándose en un documento más actual ya que el documento anterior era de 1967. Las especificaciones para las propiedades del agua fueron obtenidas del documento de 1997: de la Asociación Internacional de Propiedades del Agua y Vapor (The International Association for the Properties of Water and Steam (IAPWS)) [3], en donde se especifican algoritmos iterativos para obtener las propiedades del agua en diferentes condiciones físicas.

Las propiedades básicas que se obtuvieron fueron:

- Capacidad calorífica isobárica.
- Energía interna.
- Entalpía.
- Temperatura de saturación.
- Volumen específico.

Así como composiciones de estas de acuerdo a las necesidades propias del simulador como se muestra:

```
..... ◆ CapacidadCalorificalsobarica(float Presion, float Temperatura)
..... ◆ CpDadoPs(float PresionS)
..... ◆ EnergiaInterna(float Presion, float Temperatura)
..... ◆ HDadoPs(float PresionS)
..... ◆ PressFW1(float PresionRef, float FeedW)
..... ◆ SatCoeffnit()
..... ◆ TemperaturaSaturacion(float PresionS)
..... ◆ TempFeedWaterCent(float PresionSat, float Entalpia)
..... ◆ VolumenEspecifico(float Presion, float Temperatura)
```

Tablas de vapor.

### **Consideraciones.**

Las tablas de vapor basadas en el código de 1967, proporciona una buena respuesta, sin embargo el código no estaba preparado para Matlab por lo que se hizo la conversión pertinente.

El documento de la IAPWS es muy específico de acuerdo a las características de presión y temperatura en la que operan las tablas, y ofrece diferentes métodos para cada una de estas regiones.

Las tablas desarrolladas no ofrecen el comportamiento en todas las regiones, debido a que las necesidades específicas de simulación que no sobrepasaban ciertos límites de operación.

Sin embargo consideramos que estas tablas contienen un gran potencial de ser ampliadas completamente, representando una alternativa a las tablas comerciales de costo elevado.

### **Pruebas.**

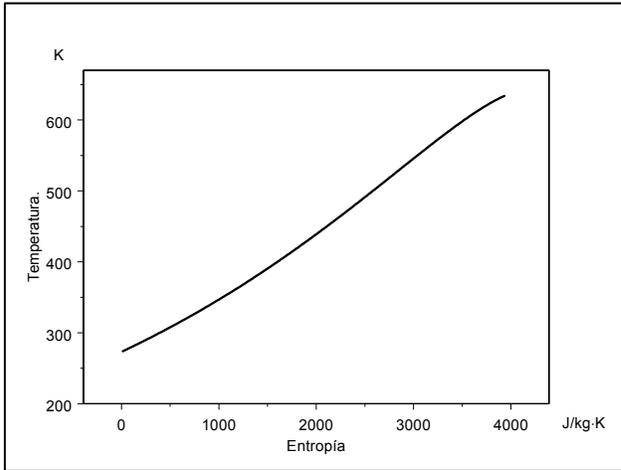
Tablas ASME67.

Para verificar el comportamiento de las tablas se empleó el programa Steam versión 6.2 de Archon Engineering. [4], este programa ofrece un periodo de evaluación de 15 días y provee gráficas de las propiedades, calculadas por el documento de la ASME de 1967, así como por el documento de la IAPWS de 1997.

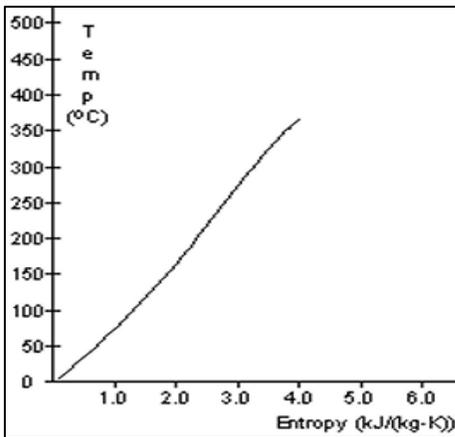
Se obtuvieron las gráficas de este programa y se compararon con las gráficas que se obtuvieron de las tablas desarrolladas para el simulador. Se hicieron graficas comparando dos propiedades, conforme a las graficas predeterminadas del programa “Steam”.

Para la graficación de las propiedades se empleo tomaron los datos de el Workspace de Matlab y se graficaron empleando las facilidades del programa.

Temperatura – Entropía.

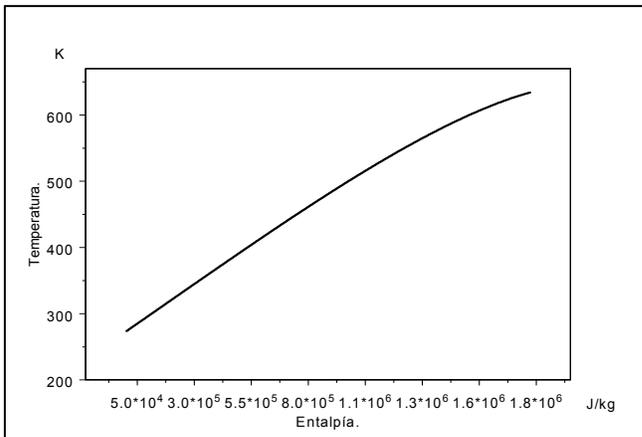


Gráfica T-S. Obtenida de tablas propias, desarrolladas para el simulador, Código 1967 ASME.



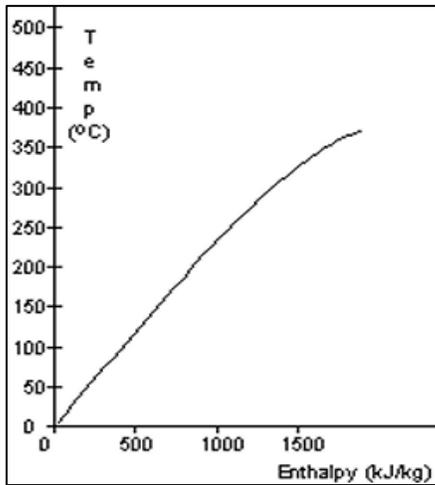
Gráfica T-S. obtenida del programa “Steam” de Archon Engineering código 1967 ASME.

Temperatura – Entalpía.



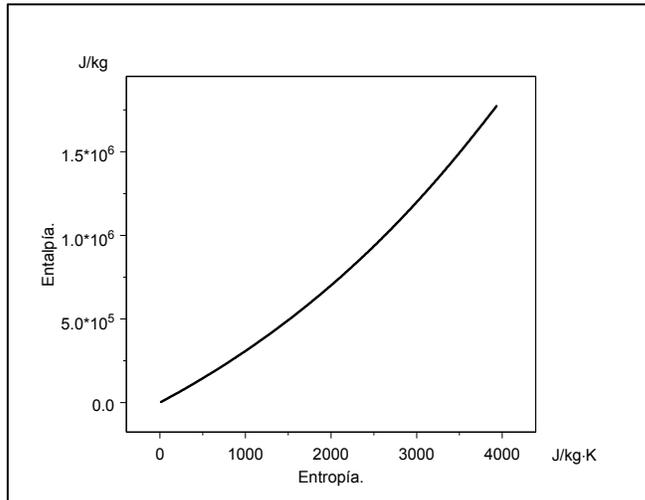
Gráfica T-H. Obtenida de tablas propias, desarrolladas para el simulador, Código 1967 ASME.

Tablas de vapor.

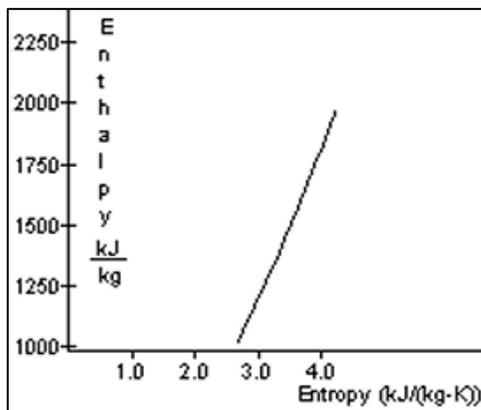


Grafica T-H. obtenida del programa “Steam” de Archon Engineering código 1967 ASME.

Entalpía – Entropía.

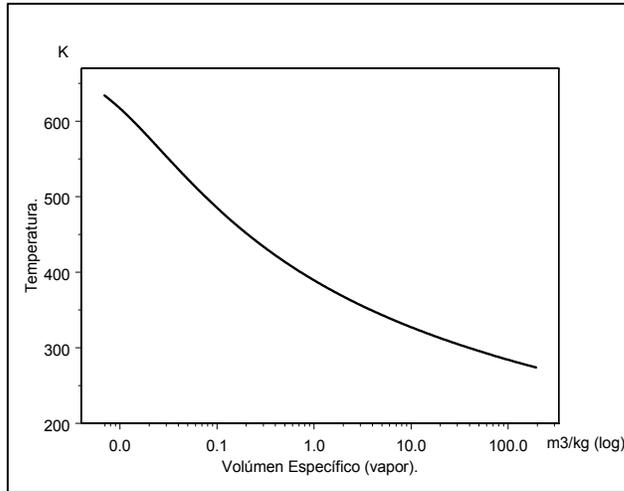


Gráfica H-S. Obtenida de tablas propias, desarrolladas para el simulador, Código 1967 ASME.

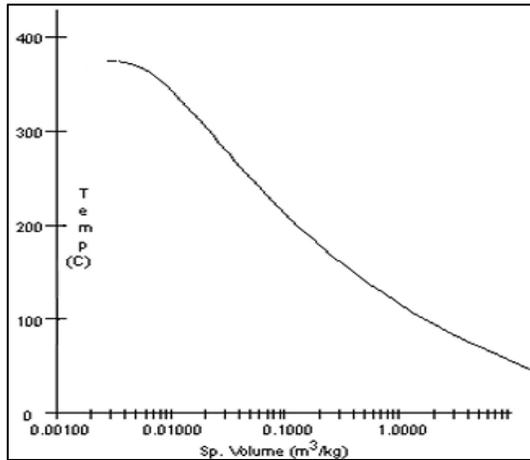


Grafica H-S. obtenida del programa “Steam” de Archon Engineering código 1967 ASME.

Temperatura – Volumen Específico (vapor).

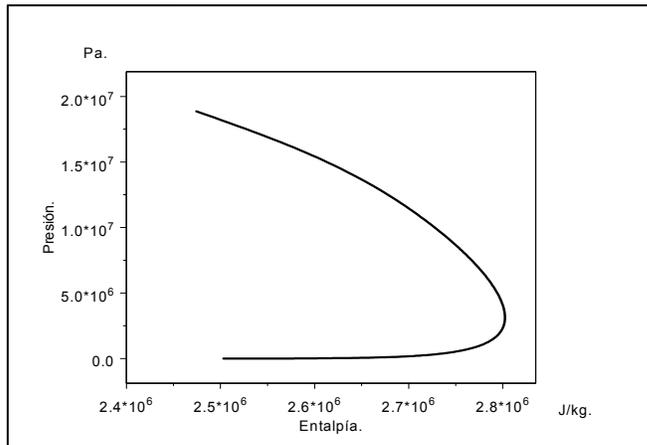


Gráfica T-V. Obtenida de tablas propias, desarrolladas para el simulador,Codigo 1997 IAPWS.



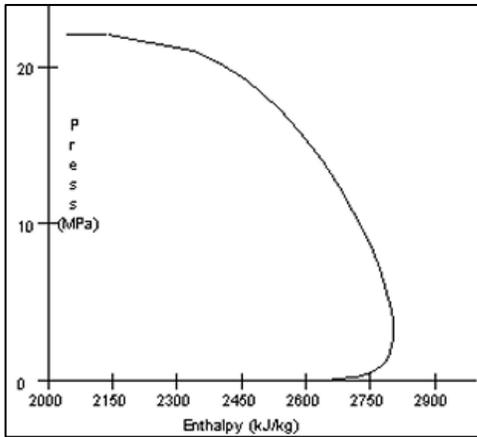
Gráfica T-V. obtenida del programa “Steam” de Archon Engineering codigo 1997 IAPWS.

Presión – Entalpía.



Gráfica P-H. Obtenida de tablas propias, desarrolladas para el simulador,Codigo 1997 IAPWS.

Tablas de vapor.



Grafica P-H. obtenida del programa “Steam” de Archon Engineering codigo 1997 IAPWS.

Como se puede observar los valores obtenidos por las funciones de desarrollo propio, presentan un comportamiento aceptable, de manera que pueden ser consideradas en los cálculos que requieran las propiedades del vapor de agua en ciertas condiciones.

### Referencias Bibliográficas.

[1] McGuire Michael Lynn, “*The 1967 ASME Steam Tables*”, WinSim, 1985

<http://www.winsim.com/steam/steam.html>

[2] Matlab documentación capítulo 6, *MATLAB External Interfaces*.

© COPYRIGHT 1984 - 2003 by The MathWorks, Inc. September 2003 Online only Revised for MATLAB 6.5.1

[3] The International Association for the Properties of Water and Steam, *Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam*, Germany, 1997.

<http://www.iapws.org/>

[4] Archon Engineering, *Archon Engineering Desktop Software Demo Version / Steam*, Columbia, 2004.

<http://www.archoneng.com/>

## Apéndice B.

### Implementación en Matlab.

En este apéndice se muestra la estructura jerárquica de implementación en el entorno Simulink de Matlab. Los números en contorno muestran los subsistemas, Los números en negro representan el desglose de esos subsistemas, para esto se mencionan el número del sistema original un punto y el número de su ramificación.

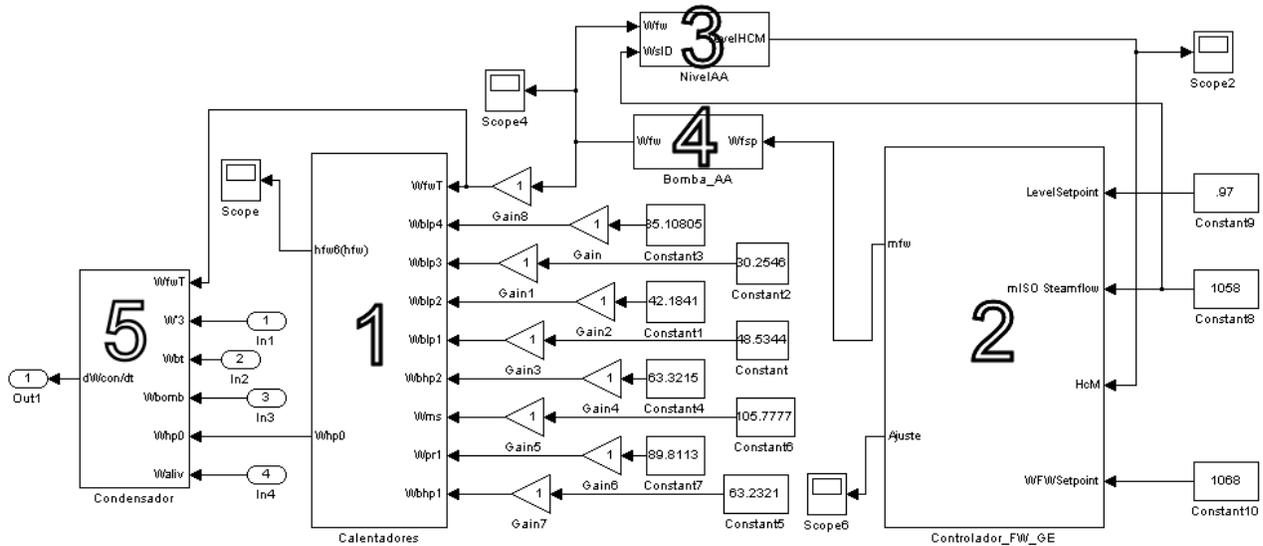


Figura 1. Modelo principal, con parámetros de prueba.

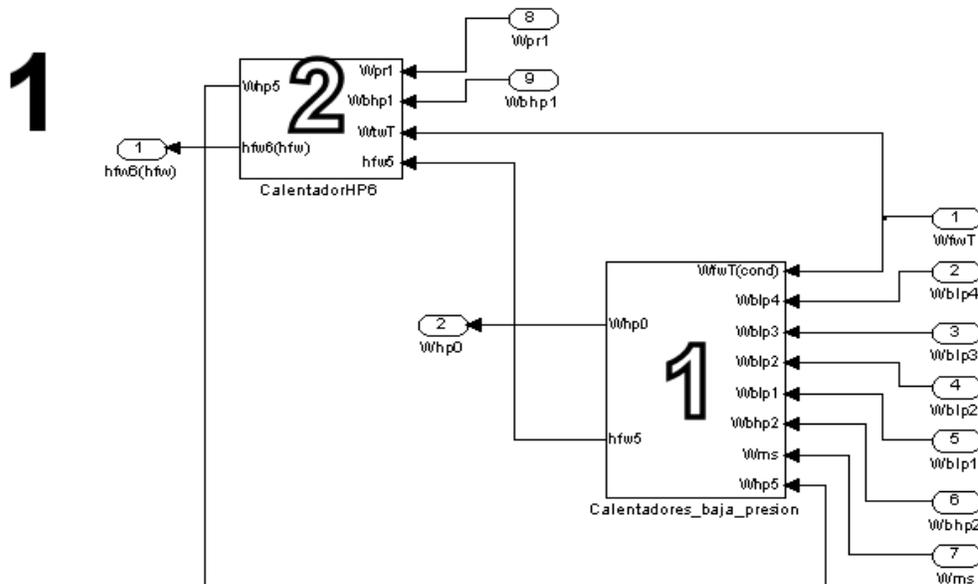


Figura 2: Modelo "Calentadores".

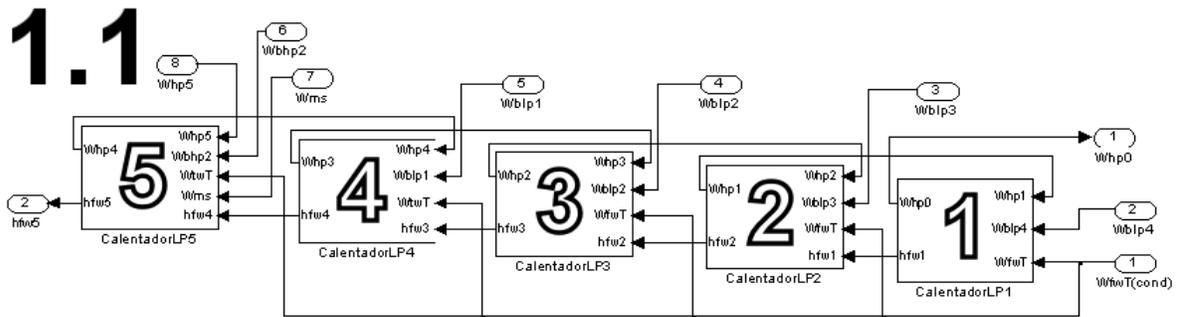


Figura 3: Calentadores de baja presión.

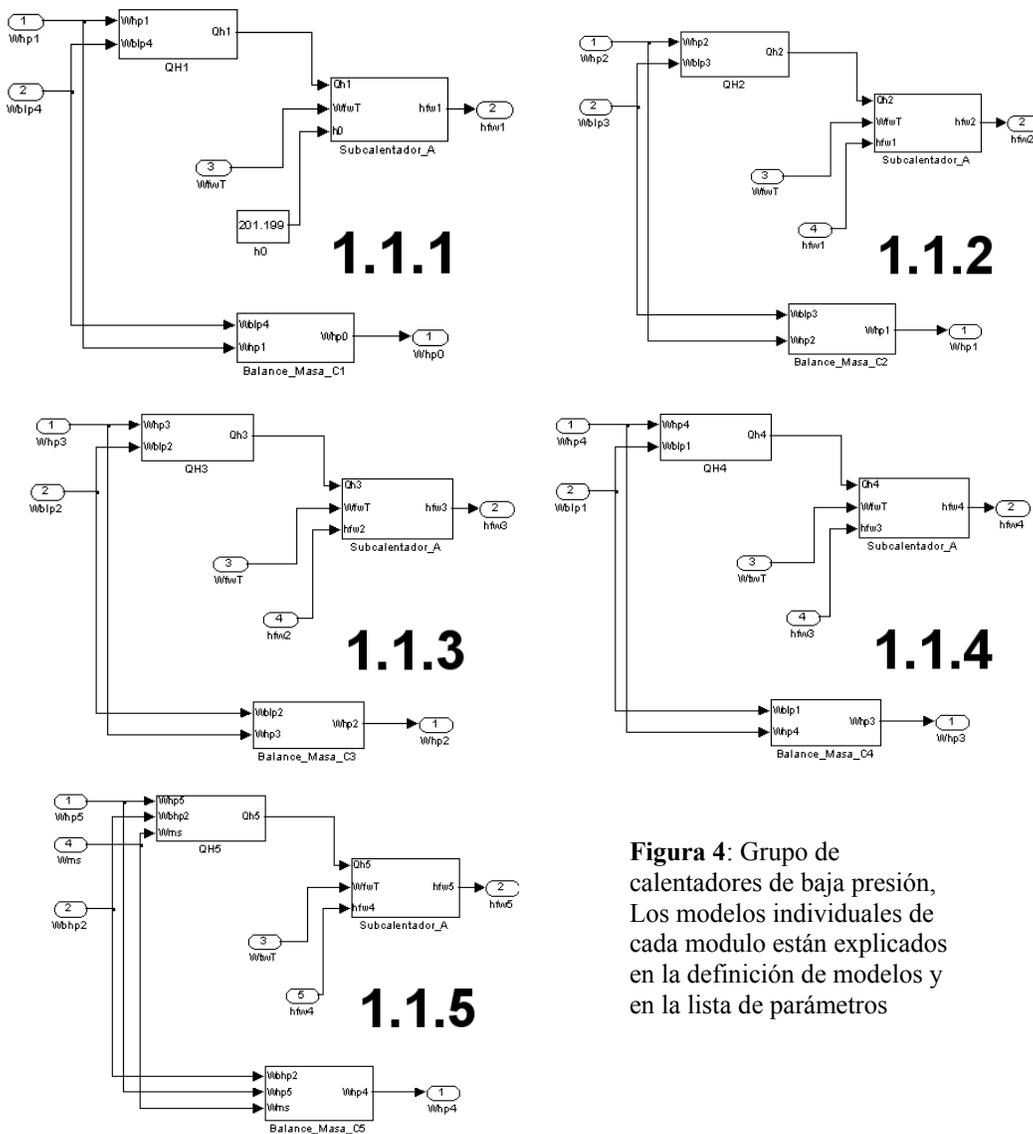
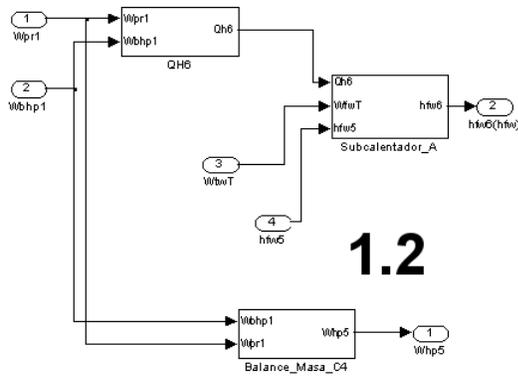


Figura 4: Grupo de calentadores de baja presión, Los modelos individuales de cada modulo están explicados en la definición de modelos y en la lista de parámetros

Implementación en Matlab.



1.2

Figura 5: Calentador de alta presión.

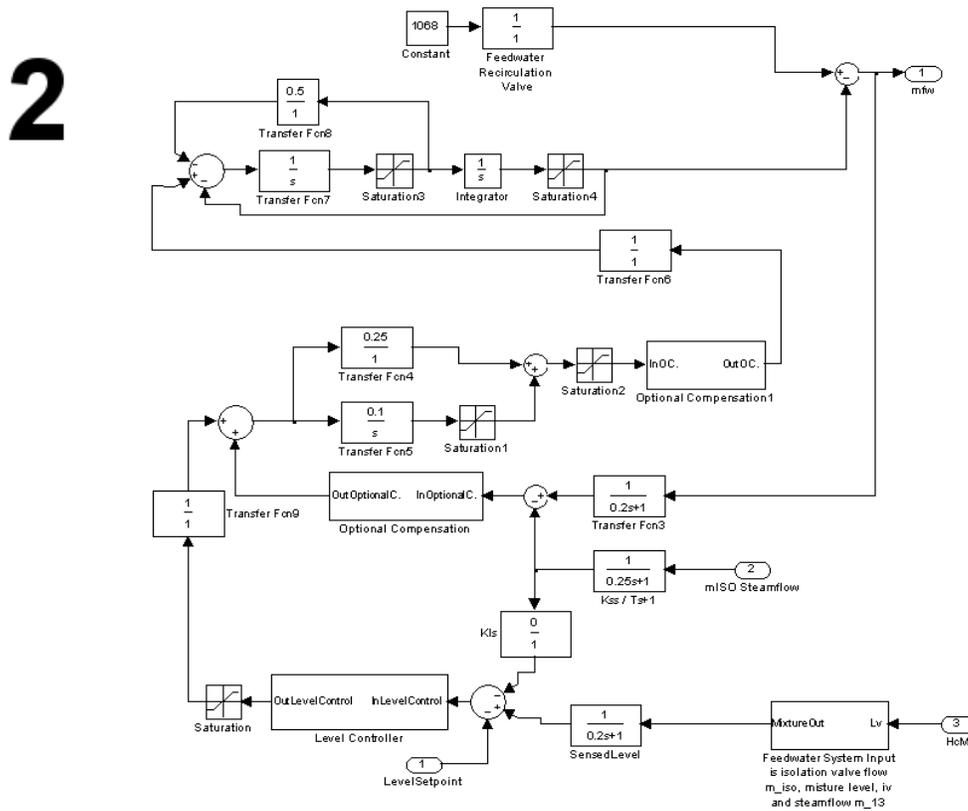
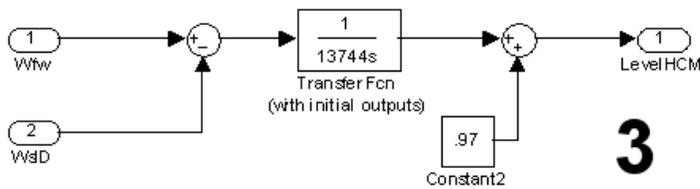


Figura 6: Diagrama del controlador, los módulos del controlador están explicados en la definición del mismo.



3

Figura 7: Módulo de nivel creado exclusivamente para pruebas ya que no sustituye el comportamiento del reactor.

Apéndice B.

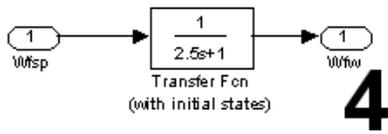


Figura 8: Bomba de agua de alimentación.

5

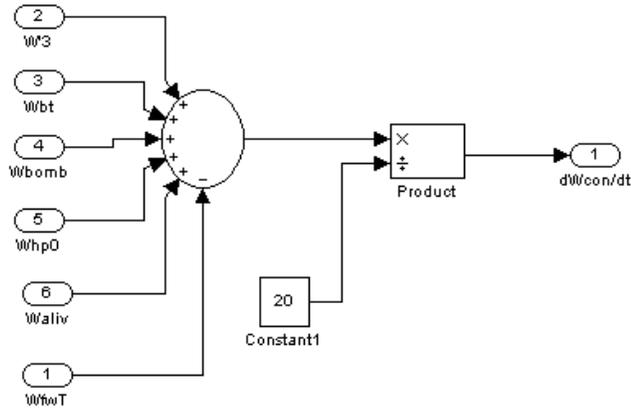
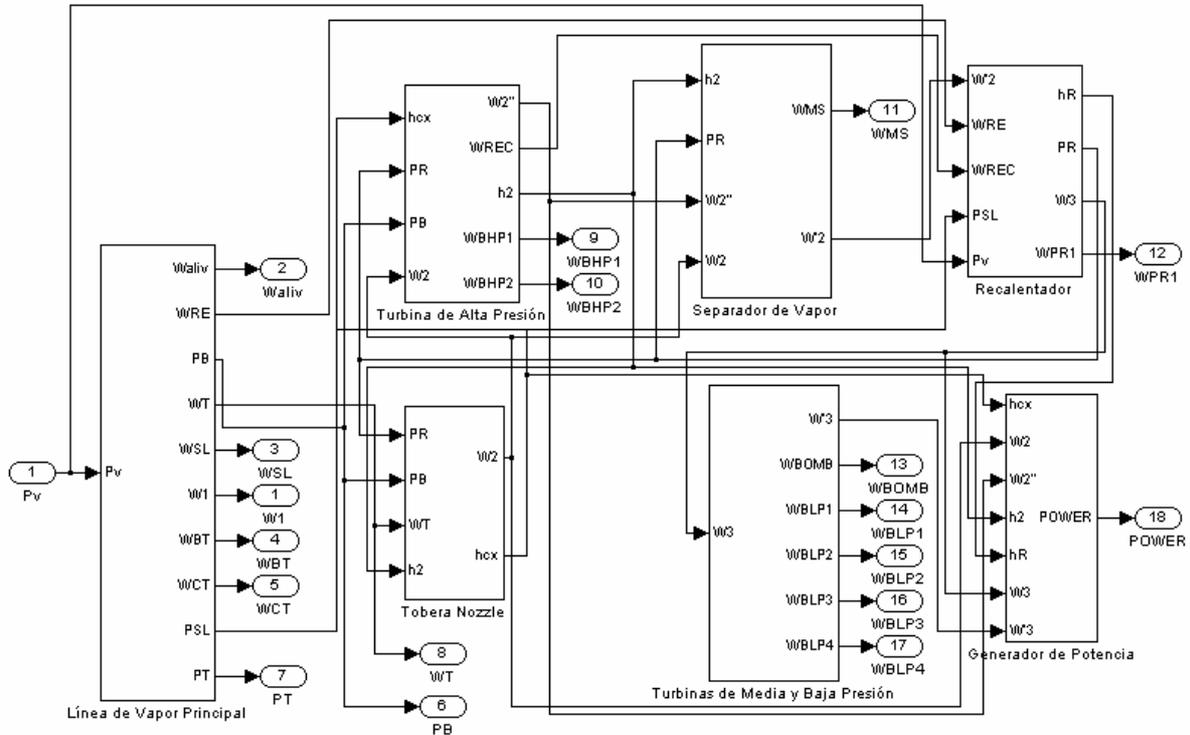


Figura 9: Condensador.



El sistema de Turbina, Línea de Vapor, Recalentador, y Generador no ha sido explicado pues corresponde a otro documento de tesis, sin embargo se presenta un esquema, del mismo a fin de que se observe su complejidad y diseño.



**Figura 2:** Esquema de el sistema de Turbinas, Separadores de Vapor, Línea de vapor, Recalentador, Generador de Potencia, etc.

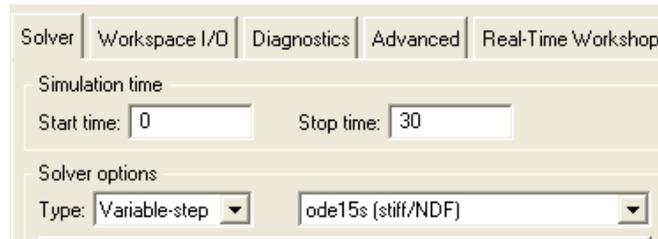
Como ejemplo de la estabilidad y de la congruencia de valores se presenta la siguiente tabla en donde se consignan los valores reemplazados por la nueva configuración y los valores obtenidos del nuevo sistema.

Variable de entrada (valor predeterminado anterior)	Valor de la variable ( todos son flujos [kg/s])
Wbpl4: Extracción turbina baja presión	35.1058
Wbpl3: Extracción turbina baja presión	30.2548
Wbpl2: Extracción turbina baja presión	42.1841
Wbpl1: Extracción turbina baja presión	48.5344
Wbhp1: Extracción turbina alta presión	63.2331
Wbhp2: Extracción turbina alta presión	63.3215
Wms: Extracción separador de vapor	105.777
Wpr1: Extracción recalentador.	89.8113

**Tabla 1:** Valores predeterminados con los que se inicio la simulación anteriormente.

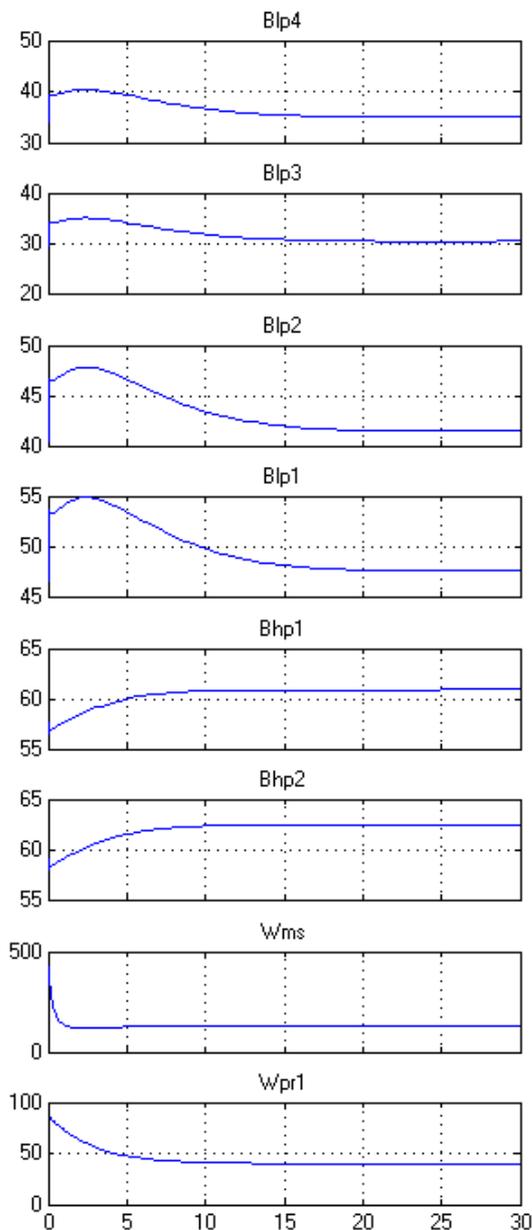
Pruebas integradas del Sistema de Condensado, Agua de alimentación y Controlador de Inventario de Refrigerante y del Sistema de Turbina, Línea de Vapor, Recalentador y Generador.

Para esta simulación de comprobación de reemplazo de entradas se usó la siguiente configuración:



**Figura 3:** Parámetros para la simulación de comparación.

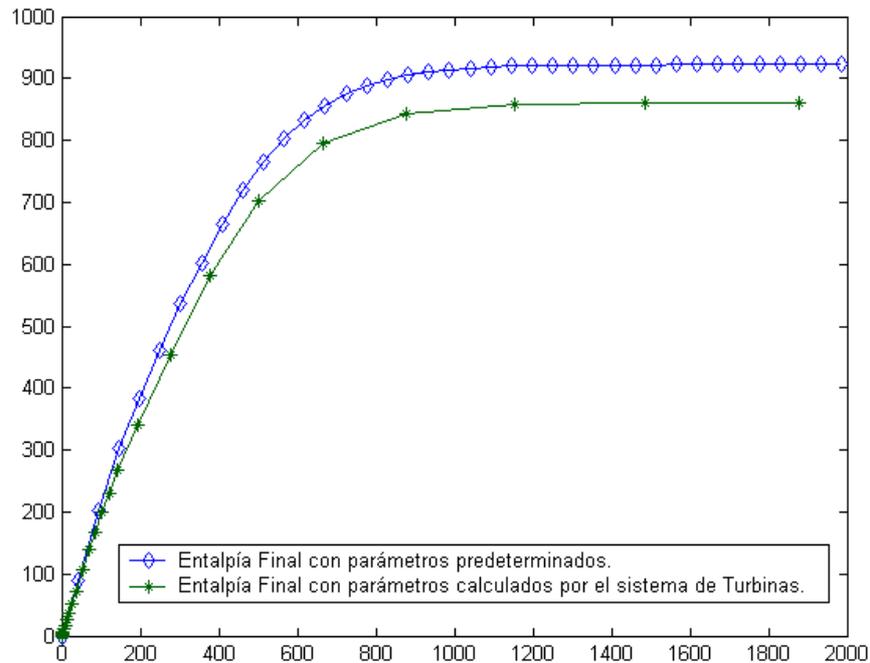
Los valores obtenidos del nuevo sistema son:



Variable de entrada (valor calculado) [kg/s]	
B <sub>1p4</sub>	:35.03
B <sub>1p3</sub>	:30.37
B <sub>1p2</sub>	:41.52
B <sub>1p1</sub>	:47.6
B <sub>hp1</sub>	:60.91
B <sub>hp2</sub>	:62.44
W <sub>ms</sub>	:133.9
W <sub>pr1</sub>	:40.31

**Figura 4:** Graficas y valores obtenidos por los modelos, al tiempo de estabilización de los sistemas y que se usaron como entradas para la simulación integrada.

Se puede ver que los valores obtenidos son similares, por lo que la integración se lleva a cabo congruentemente. Los valores de salida en cada caso son similares entre si, lo cual se puede observar en esta gráfica.



**Figura 5.** Comprobación de la entalpía final del flujo de Agua de Alimentación en ambas configuraciones.

Como se puede ver la entalpía no es exactamente la misma, sin embargo los valores conservan la tendencia y la diferencia es realmente mínima considerando la complejidad en el calculo de estos valores por el sistema de Turbinas y Líneas de vapor.

**Conclusiones:**

La integración entre el sistema de Agua de Alimentación y el de Turbinas, resulto un proceso realmente sencillo y transparente debido a que los parámetros que se colocaron fueron revisados constantemente durante el desarrollo de ambos sistemas. La congruencia de los parámetros permitió que las respuestas fueran similares en las salidas y brinda seguridad, a la expectativa de que el sistema del Reactor opere en condiciones similares.

## Apéndice D.

### Factores de Conversión. Sistema Inglés / Sistema Métrico.

Algunas de los parámetros que se usarían en los modelos, se encontraban en el sistema inglés, por lo que se usaron los factores de conversión al sistema internacional, provenientes en la primera página del libro Termodinámica de Yanus A. Çengel y Michael A. Boles. [1] .Los factores más empleados son:

Densidad.

$$1 \text{ kg/m}^3 = 0.062428 \text{ lbm/pe}^3$$

Energía, calor, trabajo, energía específica.

$$1 \text{ kJ} = 0.94782 \text{ Btu.}$$

$$1 \text{ Btu} = 5.40395 \text{ psia} \cdot \text{pe}^3$$

$$1 \text{ Btu/lbm} = 2.326 \text{ kJ/kg}$$

$$1 \text{ kJ/kg} = 0.430 \text{ Btu/lbm.}$$

Longitud

$$1 \text{ m} = 39.370 \text{ pulg} = 3.2808 \text{ pe.}$$

$$1 \text{ pe} = 12 \text{ pulg.}$$

$$1 \text{ pulg} = 2.54 \text{ cm}$$

Masa

$$1 \text{ kg} = 2.2046226 \text{ lbm}$$

$$1 \text{ lbm} = 0.45359237 \text{ kg.}$$

Potencia

$$1 \text{ kW} = 3412.14 \text{ Btu/h}$$

Presión

$$1 \text{ Pa} = 1.4504 \times 10^{-4} \text{ psia} = 0.020886 \text{ lbf/pe}^2$$

$$1 \text{ psia} = 144 \text{ lbf/pe}^2 = 6.894757 \text{ kPa.}$$

Calor Específico

$$1 \text{ Btu/(lbm} \cdot ^\circ\text{F)} = 4.1868 \text{ kJ/(kg} \cdot ^\circ\text{C)}$$

$$1 \text{ kJ/(kg} \cdot ^\circ\text{C)} = 0.23885 \text{ Btu/(lbm} \cdot ^\circ\text{F)}$$

Volumen Específico

$$1 \text{ m}^3/\text{kg} = 16.02 \text{ pe}^3/\text{lbm.}$$

$$1 \text{ pe}^3/\text{lbm} = 0.062428 \text{ m}^3/\text{kg}$$

Temperatura

$$T(^{\circ}\text{F}) = 1.8 T(^{\circ}\text{C}) + 32$$

$$T(\text{K}) = T(^{\circ}\text{C}) + 273.15$$

Volumen

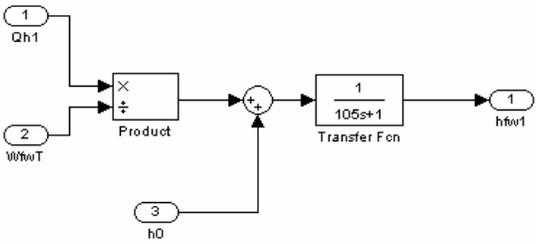
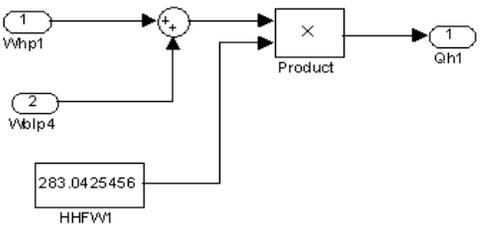
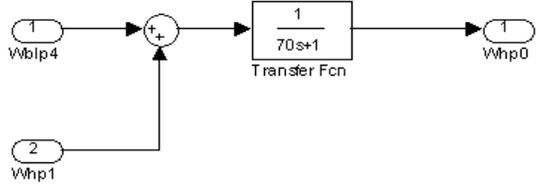
$$1 \text{ m}^3 = 6.1024 \times 10^4 \text{ pulg}^3 = 35.315 \text{ pe}^3$$

[1] Çengel, Yanus A. y Boles, Michael A., *Termodinámica Segunda Edición*. McGrawHill México, 1994

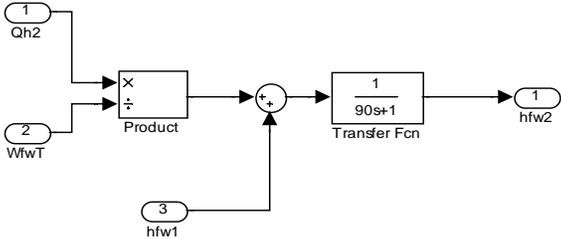
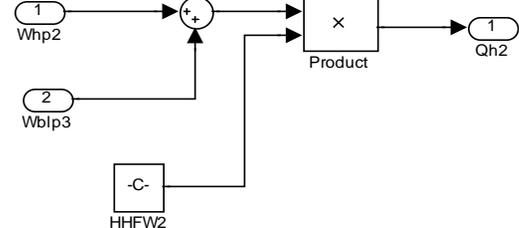
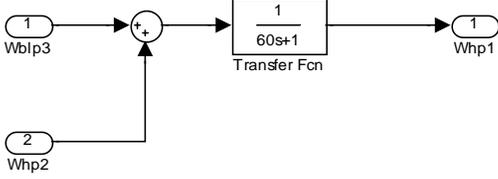
## Apéndice E.

### Tablas de Parámetros e Implementación de los dispositivos modelados

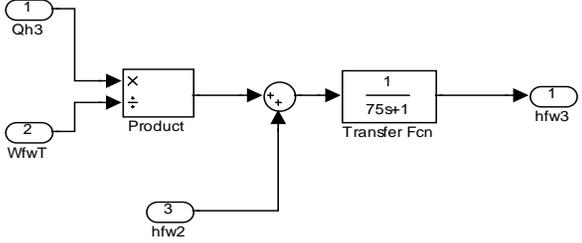
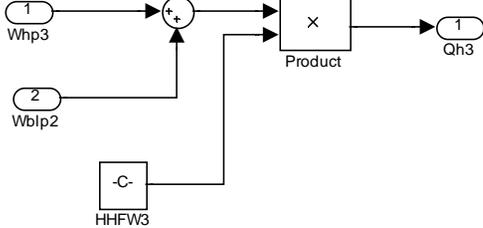
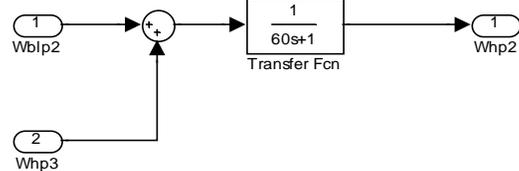
#### Primer Calentador.

Ecuación del Modelo. (En el dominio del tiempo)	Función de Transferencia (En el dominio de "s" (Laplace) )	Parámetros. El número que sigue al parámetro, es la referencia en el índice posterior.	Implementación en MATLAB®.
Ecuación de entalpía en el primer calentador:  $\frac{dh_{fw1}}{dt} = \frac{Q_{H1}}{T_{H1} W_{fw}} + \frac{h_0 - h_{fw1}}{T_{H1}}$	$H(s) = \frac{1}{T_{H1}s + 1}$ $Y(s) = H_{fw1}(s)$ $X(s) = \frac{Q_{H1}(s)}{W_{fw}(s)} + h_0$	$T_{H1} = 105$ [1] $h_0 = 201.199$ [2]	
Ecuación del calor en el primer calentador:  $Q_{H1} = H_{HFW1}(W_{HP1} + W_{BLP4})$		$H_{HFW1} = 283042.5456$ [3]	
Ecuación del flujo lado carcasa en el primer calentador:  $\frac{dW_{HP0}}{dt} = \frac{W_{BLP4} + W_{HP1} - W_{HP0}}{T_{H1P}}$	$H(s) = \frac{1}{T_{H1P}s + 1}$ $Y(s) = W_{HP0}(s)$ $X(s) = W_{BLP4}(s) + W_{HP1}(s)$	$T_{H1P} = 70$ [4]	

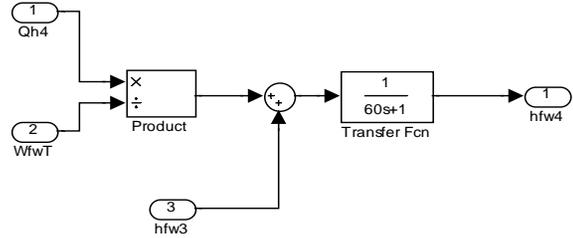
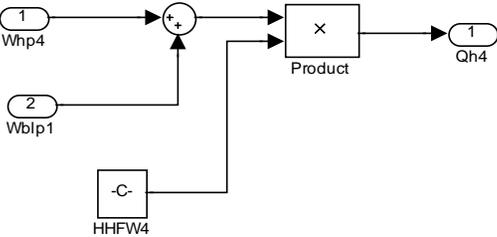
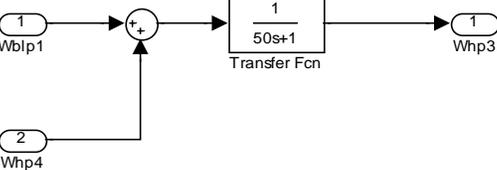
## Segundo Calentador.

Ecuación del Modelo. (En el dominio del tiempo)	Función de Transferencia (En el dominio de "s" (Laplace) )	Parámetros. El número que sigue al parámetro, es la referencia en el índice posterior.	Implementación en MATLAB®.
<p>Ecuación de entalpía en el segundo calentador:</p> $\frac{dh_{fw2}}{dt} = \frac{Q_{H2}}{T_{H2}W_{fw}} + \frac{h_{fw1} - h_{fw2}}{T_{H2}}$	$H(s) = \frac{1}{T_{H2}s + 1}$ $Y(s) = h_{fw2}(s)$ $X(s) = \frac{Q_{H2}(s)}{W_{fw}(s)} + h_{fw1}(s)$	$T_{H2} = 90 \quad [5]$	
<p>Ecuación del calor en el segundo calentador:</p> $Q_{H2} = H_{HFW2}(W_{BLP3} + W_{HP2})$		$H_{HFW2} = 205671.227 \quad 3 \quad [6]$	
<p>Ecuación del flujo lado carcasa en el segundo calentador:</p> $\frac{dW_{HP1}}{dt} = \frac{W_{BLP3} + W_{HP2} - W_{HP1}}{T_{H2P}}$	$H(s) = \frac{1}{T_{H2P}s + 1}$ $Y(s) = W_{HP1}(s)$ $X(s) = W_{BLP3}(s) + W_{HP2}(s)$	$T_{H2P} = 60 \quad [7]$	

**Tercer Calentador.**

Ecuación del Modelo. (En el dominio del tiempo)	Función de Transferencia (En el dominio de “s” (Laplace) )	Parámetros. El número que sigue al parámetro, es la referencia en el índice posterior.	Implementación en MATLAB®.
<p>Ecuación de entalpía en el segundo calentador:</p> $\frac{dh_{fw3}}{dt} = \frac{Q_{H3}}{T_{H3}W_{fw}} + \frac{h_{fw2} - h_{fw3}}{T_{H3}}$	$H(s) = \frac{1}{T_{H3}s + 1}$ $Y(s) = h_{fw3}(s)$ $X(s) = \frac{Q_{H3}(s)}{W_{fw}(s)} + h_{fw2}(s)$	$T_{H3} = 75 \quad [8]$	
<p>Ecuación del calor en el tercer calentador:</p> $Q_{H3} = H_{HFW3}(W_{HP3} + W_{BLP2})$		$H_{HFW3} = 302412.4739 \quad [9]$	
<p>Ecuación del flujo lado carcasa en el tercer calentador:</p> $\frac{dW_{HP2}}{dt} = \frac{W_{BLP2} + W_{HP3} - W_{HP2}}{T_{H3P}}$	$H(s) = \frac{1}{T_{H3P}s + 1}$ $Y(s) = W_{HP2}(s)$ $X(s) = W_{BLP2}(s) + W_{HP3}(s)$	$T_{H3P} = 60 \quad [10]$	

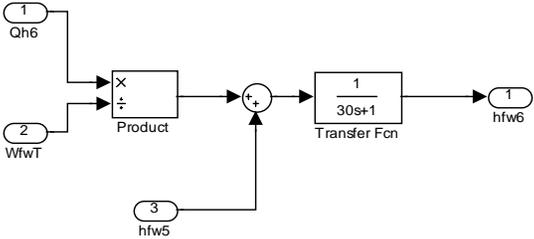
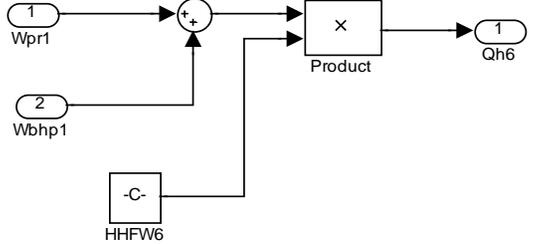
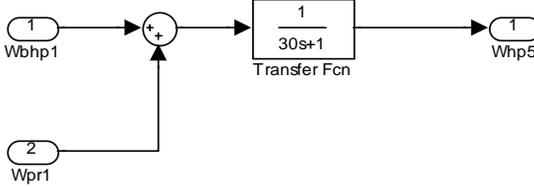
## Cuarto Calentador.

Ecuación del Modelo. (En el dominio del tiempo)	Función de Transferencia (En el dominio de "s" (Laplace) )	Parámetros. El número que sigue al parámetro, es la referencia en el índice posterior.	Implementación en MATLAB®.
<p>Ecuación de entalpía en el segundo calentador:</p> $\frac{dh_{fw4}}{dt} = \frac{Q_{H4}}{T_{H4}W_{fw}} + \frac{h_{fw3} - h_{fw4}}{T_{H4}}$	$H(s) = \frac{1}{T_{H4}s + 1}$ $Y(s) = h_{fw4}(s)$ $X(s) = \frac{Q_{H4}(s)}{W_{fw}(s)} + h_{fw3}(s)$	$T_{H4} = 60 \quad [11]$	
<p>Ecuación del calor en el cuarto calentador:</p> $Q_{H4} = H_{HFW4}(W_{HP4} + W_{BLP1})$		$H_{HFW4} = 367169.156 \quad 8 \quad [12]$	
<p>Ecuación del flujo lado carcasa en el cuarto calentador:</p> $\frac{dW_{HP3}}{dt} = \frac{W_{BLP1} + W_{HP4} - W_{HP3}}{T_{H4P}}$	$H(s) = \frac{1}{T_{H4P}s + 1}$ $Y(s) = W_{HP3}(s)$ $X(s) = W_{BLP1}(s) + W_{HP4}(s)$	$T_{H4P} = 50 \quad [13]$	

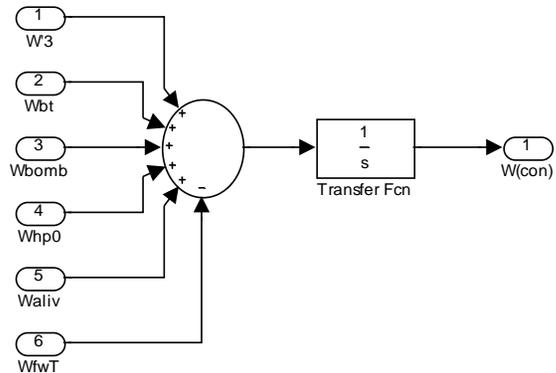
**Quinto Calentador.**

Ecuación del Modelo. (En el dominio del tiempo)	Función de Transferencia (En el dominio de “s” (Laplace) )	Parámetros. El número que sigue al parámetro, es la referencia en el índice posterior.	Implementación en MATLAB®.
<p>Ecuación de entalpía en el segundo calentador:</p> $\frac{dh_{fw5}}{dt} = \frac{Q_{H5}}{T_{H5}W_{fw}} + \frac{h_{fw4} - h_{fw5}}{T_{H5}}$	$H(s) = \frac{1}{T_{H5}s + 1}$ $Y(s) = h_{fw5}(s)$ $X(s) = \frac{Q_{H5}(s)}{W_{fw}(s)} + h_{fw4}(s)$	$T_{H5} = 45 \quad [14]$	
<p>Ecuación del calor en el quinto calentador:</p> $Q_{H5} = H_{HFW5}(W_{HP5} + W_{BHP2} + W_{MS})$		$H_{HFW5} = 663106.958 \quad 5 \quad [15]$	
<p>Ecuación del flujo lado carcasa en el quinto calentador:</p> $\frac{dW_{HP4}}{dt} = \frac{W_{HF5} + W_{BHP2} + W_{MS} - W_{HP4}}{T_{H5P}}$	$H(s) = \frac{1}{T_{H5P}s + 1}$ $Y(s) = W_{HP4}(s)$ $X(s) = W_{BHP2}(s) + W_{HF5}(s) + W_{MS}(s)$	$T_{H5P} = 40 \quad [16]$	

Sexto Calentador.

Ecuación del Modelo. (En el dominio del tiempo)	Función de Transferencia (En el dominio de “s” (Laplace) )	Parámetros. El número que sigue al parámetro, es la referencia en el índice posterior.	Implementación en MATLAB®.
<p>Ecuación de entalpía en el segundo calentador:</p> $\frac{dh_{fw6}}{dt} = \frac{Q_{H6}}{T_{H6}W_{fw}} + \frac{h_{fw5} - h_{fw}}{T_{H6}}$	$H(s) = \frac{1}{T_{H6}s + 1}$ $Y(s) = h_{fw6}(s)$ $X(s) = \frac{Q_{H6}(s)}{W_{fw}(s)} + h_{fw5}(s)$	$T_{H6} = 30$ [17]	
<p>Ecuación del calor en el sexto calentador:</p> $Q_{H6} = H_{HFW6}(W_{PR1} + W_{BHP1})$		$H_{HFW6} = 403509.6205$ [18]	
<p>Ecuación del flujo lado carcasa en el sexto calentador:</p> $\frac{dW_{HP5}}{dt} = \frac{W_{PR1} + W_{BHP1} - W_{HP5}}{T_{H6P}}$	$H(s) = \frac{1}{T_{H6P}s + 1}$ $Y(s) = W_{HP5}(s)$ $X(s) = W_{BHP1}(s) + W_{PR1}(s)$	$T_{H6P} = 30$ [19]	

**Condensador.**

<p>Ecuación del Modelo. (En el dominio del tiempo)</p>	<p>Función de Transferencia (En el dominio de “s” (Laplace) )</p>
<p>Ecuación del flujo dentro del condensador:</p> $\frac{dW_{con}}{dt} = \frac{W'_3 + W_{BT} + W_{BOMB} + W_{HP0} + W_{ALIV} - W_{fw}}{T_{CON}}$	$H(s) = \frac{1}{T_{CON}s}$ $Y(s) = W_{con}(s)$ $X(s) = W'_3(s) + W_{BT}(s) + W_{BOMB}(s) + W_{HP0}(s) + W_{ALIV}(s) - W_{fw}(s)$
<p>Parámetros. El número que sigue al parámetro, es la referencia en el índice posterior.</p>	<p>Implementación en MATLAB®.</p>
<p><math>T_{CON} = 70.</math> [20]</p>	

### Bomba de Agua de Alimentación.

<p>Ecuación del Modelo. (En el dominio del tiempo)</p>	<p>Función de Transferencia (En el dominio de “s” (Laplace) )</p>
<p>Ecuación del flujo de agua de alimentación entregado por la bomba:</p> $\frac{dW_{fw}}{dt} = \frac{1}{\tau_{FW}} (W_{FSP} - W_{fw})$	$H(s) = \frac{1}{\tau_{FW}s + 1}$ $Y(s) = W_{fw}(s)$ $X(s) = W_{FSP}(s)$
<p>Parámetros. El número que sigue al parámetro, es la referencia en el índice posterior.</p>	<p>Implementación en MATLAB®.</p>
<p><math>\tau_{FW} = 2.5</math> [21]</p>	