



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**“MODELAMIENTO Y SIMULACIÓN DE UNA CALDERA”**

**INFORME DE MATERIA DE GRADUACIÓN**

Previo a la obtención del título de:

**INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN  
ELECTRÓNICA Y AUTOMATIZACIÓN INDUSTRIAL**

Presentada por:

Janeth Katherine Silva Aspiazu

**GUAYAQUIL – ECUADOR**

**AÑO: 2015**

## AGRADECIMIENTO

Agradezco a Dios por enseñarme que cuando lo ponemos en primer lugar, el resto de cosas viene por añadidura. A mi madre por motivarme a ser politécnica. A mi hermana por enseñarme que con esfuerzo y Dios podemos lograr muchas cosas. A esos compañeros y profesores que me motivaron siempre a perseverar y a conquistar mis sueños. A mis amigos que me han acompañado en todo este largo caminar.

*Janeth Katherine Silva Aspiazu*

## DEDICATORIA

A esas personas que tienen un sueño y siguen en el camino para cumplirlo. Muchas cosas quedarán en el camino pero nada como la satisfacción de un sueño cumplido.

*Janeth Katherine Silva Aspiazu*

## TRIBUNAL DE SUSTENTACIÓN



---

Ph.D. Douglas Plaza G.

PROFESOR DE LA MATERIA DE GRADUACIÓN



---

M.Sc. Victor Asanza A.

PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

## DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este informe, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral”

(Reglamento de Graduación de la ESPOL)

  
\_\_\_\_\_  
Janeth Katherine Silva Aspiazu

## RESUMEN

El objetivo de este trabajo es modelar y simular un generador de vapor o caldera evaluando el desempeño de cada uno de los programas seleccionados. Los programas seleccionados para implementar el modelo matemático son Matlab/Simulink y Scilab/Xcos.

En la primera etapa se seleccionará un generador de vapor específico y luego se describirán los análisis realizados para obtener la descripción matemática del mismo. Para poder realizar esto es necesario conocer los principios de funcionamiento del sistema y las variables de interés del mismo.

En la segunda etapa se implementa la ecuación obtenida que representa el sistema en los diferentes simuladores, en la simulación se especifican las limitaciones y consideraciones previas para poder realizar la simulación.

Después de la ejecución del sistema en los programas se procede a analizar el desempeño de cada uno de los simuladores para poder establecer las ventajas y desventajas respectivas de cada uno.

Como resultado, Matlab/Simulink se diferencia de Scilab/Xcos debido a que los tiempos de simulación del mismo son diez veces menor que Xcos, a la vez los valores obtenidos en Simulink son más precisos que Xcos.

## ÍNDICE GENERAL

	Pág.
AGRADECIMIENTO .....	II
DEDICATORIA .....	III
TRIBUNAL DE SUSTENTACIÓN .....	IV
DECLARACIÓN EXPRESA .....	V
RESUMEN .....	VI
ÍNDICE GENERAL.....	VII
ABREVIATURAS .....	X
SIMBOLOGÍA .....	XI
ÍNDICE DE FIGURAS.....	XII
ÍNDICE DE TABLAS .....	XIV
INTRODUCCIÓN .....	XV
Capítulo 1 .....	1
1. ANTECEDENTES.....	1
1.1 Justificación .....	1
1.2 Objetivo general .....	3

1.3 Objetivos específicos.....	3
1.4 Metodología.....	3
1.5 Alcance.....	5
1.6 Resultados esperados.....	5
Capítulo 2 .....	6
2. MARCO TEÓRICO .....	6
2.1 Modelos Matemáticos.....	6
2.2 Calderas .....	8
2.3 Simuladores .....	13
Capítulo 3 .....	15
3. MODELAMIENTO MATEMÁTICO DEL SISTEMA .....	15
3.1 Balance de ecuaciones .....	17
Capítulo 4 .....	23
4. IMPLEMENTACIÓN .....	23
4.1 Matlab/Simulink .....	25
4.2 Scilab/Xcos.....	30
Capítulo 5 .....	37
5. RESULTADOS .....	37
5.1 Respuesta del sistema .....	37



5.2 Cálculo de errores .....	39
CONCLUSIONES .....	42
RECOMENDACIONES.....	44
ANEXOS	
BIBLIOGRAFÍA	

## ABREVIATURAS

Matlab	Matrix Laboratory.
Pa	Pascales. Unidad de presión en el sistema internacional.
s	Segundos. Unidad de tiempo en el sistema internacional.

## SIMBOLOGÍA

$Q$	Flujo de calor.
$q_s$	Flujo másico de vapor
$q_w$	Flujo másico de agua
$H_s$	Entalpía de vapor saturado
$H_w$	Entalpía de agua saturada
$H_f$	Entalpía de agua de alimentación saturada
$p$	Presión
$T$	Temperatura
$V_w$	Volumen total de agua
$V_s$	Volumen total de vapor
$v$	

## ÍNDICE DE FIGURAS

Figura 1.1: Metodología .....	4
Figura 2.1: Estructura general de una caldera [5].....	9
Figura 2.2: Caldera pirotubular [6] .....	10
Figura 2.3: Caldera Acuotubular [6] .....	11
Figura 2.4: Caldera con lecho fluidizado [7].....	12
Figura 2.5: Caldera de Flujo Agua – Vapor [8].....	13
Figura 3.1: Grafico esquemático de la caldera [1].....	18
Figura 4.1 Parámetros tabla de vapor saturado T vs H .....	26
Figura 4.2 Parámetros tabla de vapor saturado T vs Hs.....	27
Figura 4.3 Parámetros tabla P vs T .....	27
Figura 4.4 Parámetros de configuración del flujo másico del vapor.....	28
Figura 4.5 Simulación del flujo másico del vapor perturbado.....	29
Figura 4.6 Flujo másico del vapor perturbado.....	29
Figura 4.7 Implementación de una caldera en Simulink .....	30
Figura 4.8 Parámetros T vs Hw Scilab.....	32
Figura 4.9 Parámetros T vs Hs Scilab .....	32
Figura 4.10 Parámetros T vs P Scilab .....	33
Figura 4.11 Configuración Scope en Scilab .....	34
Figura 4.12 Configuración de reloj en Scilab .....	34

Figura 4.13 Simulación del flujo másico del vapor Scilab .....	35
Figura 4.14 Parámetros de configuración CM Scope .....	36
Figura 4.15 Implementación del modelo en Scilab .....	36
Figura 5.1 Presión del tambor [1].....	38
Figura 5.2 Presión del tambor Simulink .....	38
Figura 5.3 Reporte de rendimiento de Simulink.....	39
Figura 5.4 Reporte de rendimiento de Scilab.....	40

## ÍNDICE DE TABLAS

Tabla 1: Valores numéricos de los coeficientes de e1 a una presión normal de operación .....	21
Tabla 2 Tiempos de Ejecución.....	40
Tabla 3 Desviación de la Presión en Simulink/Xcos .....	41

## INTRODUCCIÓN

En la actualidad en el área industrial los procesos se han ido automatizando para garantizar los parámetros de calidad del producto a elaborar y disponer información de manera inmediata, reduciendo los errores y optimizando los recursos materiales y humanos que intervienen en el proceso.

Para asegurar la calidad del producto o sistema es necesario someterlo al mismo a diferentes situaciones físicas, las cuales pueden tener limitaciones debido al factor económico o factor tiempo. Debido a esto se realizan modelos matemáticos que consisten en ecuaciones que describen el comportamiento del sistema con variables de entrada que pueden ser perturbadas para conocer el funcionamiento del sistema o producto bajo estas condiciones especiales.

La posibilidad de implementar procesos automatizados con desarrollo tecnológico en nuestro país, se basa en la capacidad de desarrollar modelos matemáticos de diferentes sistemas con un control específico para tener una precisión exacta en la elaboración de los productos, creando la posibilidad de competir con estos en los mercados internacionales.

En el capítulo dos trataremos el funcionamiento de la caldera a modelar y las variables características de la misma, en el tercero describiremos el proceso de obtención del modelo matemático y las características previas a considerar para la realización del mismo. El capítulo cuatro describe la implementación del modelo de la caldera en los diferentes simuladores y el último capítulo comprende los resultados de la simulación en cada uno de los programas y sus varianzas respectivas al momento de la toma de datos.



# Capítulo 1

## 1. ANTECEDENTES

La capacidad para mejorar o diseñar nuevos equipos en cualquier campo de aplicación se basa en las limitaciones del mismo y sus rangos de trabajo. Para poder determinar estas características es necesario someter cada equipo a una serie de pruebas físicas o químicas con para asegurar los estándares de calidad y un buen funcionamiento del mismo, en esta evaluación intervienen los modelos matemáticos y simuladores que permiten realizar un análisis detallado del comportamiento del equipo o producto evitando las limitaciones físicas o económicas.

### 1.1 Justificación

El desarrollo tecnológico en la industria se basa en la rápida comunicación que existe para la obtención de datos de campo en

tiempo real desde un computador. La actualización tecnológica de los nuevos equipos para las diferentes áreas industriales se basa en el desarrollo de nuevos modelos matemáticos que predicen la respuestas de los sistemas con mejores precisiones y mayores rangos de operación del mismo ampliando el rango de su utilización. El avance tecnológico ha creado una mayor demanda de energía debido al incremento en la implementación de equipos en la industria. La demanda de energía en una industria grande se disminuye con la creación de generadores de energía, propios de cada planta, si su demanda es alta. Debido a esto se emplean generadores de vapor y turbinas a vapor, que son una forma de obtener una gran cantidad de energía a un bajo costo de operación, por esto el modelamiento matemático de las calderas o generadores de vapor permite desarrollar sistemas de generación de energía con más eficiencia.

En el proceso de la obtención de la descripción matemática del generador de vapor, se utilizan programas con interfaces gráficas que permiten visualizar la respuesta del generador en tiempo real y se utiliza el modelo que se asemeje más al comportamiento requerido del sistema. Se realizan ajustes al modelo o las variables del mismo de acuerdo a la aplicación del generador. Este trabajo busca analizar el

rendimiento y desempeño de los diferentes programas a ser utilizados basados en la precisión y rapidez de los mismos.

## **1.2 Objetivo general**

Modelar y simular una caldera, evaluando el desempeño de los programas de simulación tales como: Matlab/Simulink y Scilab/Xcos .

## **1.3 Objetivos específicos**

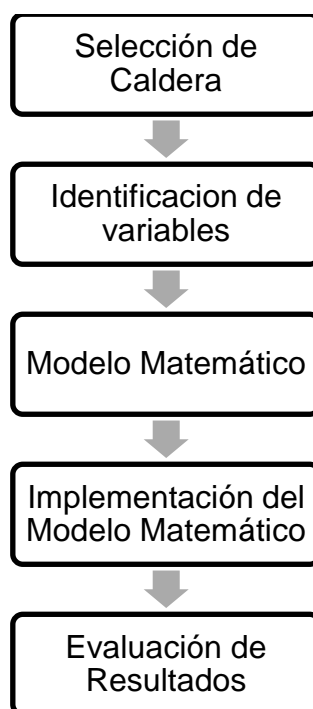
Para cumplir con el objetivo general se deben cumplir los siguientes objetivos específicos:

- Seleccionar el tipo de caldera a modelar.
- Identificar las variables y principios físicos del sistema.
- Diseñar un modelo matemático de la caldera escogida.
- Implementar el modelo matemático en cada uno de los software de simulación.
- Simular el modelo de la caldera en cada uno de los softwares.
- Comparar los resultados de cada uno de los softwares utilizados.
- Evaluar la eficiencia de las herramientas de simulación utilizadas.

## **1.4 Metodología**

Para la obtención del modelo matemático previamente se escogerá un modelo específico de caldera, donde se puedan conocer las constantes de

las mismas para poder realizar la simulación real de la misma, posteriormente se procederán a identificar las variables significativas que intervienen en el funcionamiento de la misma y luego se procederá a aplicar las leyes fundamentales que rigen el comportamiento de la misma para poder obtener su modelo matemático. Con la obtención del modelo y las constantes previamente definidas se procede a la implementación del mismo en cada uno de los simuladores y a la toma de los valores obtenidos de los tiempos de ejecución y de la propia simulación para poder compararlos entre sí.



**Figura 1.1:** Metodología

### **1.5 Alcance**

El alcance de este informe es llegar a simular el funcionamiento del tambor de una caldera de vapor utilizada en la generación de energía eléctrica P16 marca Annex de 160MW [1] en los programas antes mencionados, para poder comparar los diferentes simuladores con su respectivo error e identificar el mejor para la simulación de sistemas dinámicos.

### **1.6 Resultados esperados**

Manteniendo la misma estructura de los modelos matemáticos esperamos desarrollar un análisis comparativo del desempeño del software propietario Matlab/Simulink y el programa libre Scilab/Xcos sobre la caldera seleccionada. Además comprobar si el modelo matemático establecido se ajusta a las características de los modelos presentados en otros trabajos, donde se utilizan calderas reales.

# Capítulo 2

## 2. MARCO TEÓRICO

Para poder comprender el funcionamiento del sistema y su modelo matemático es necesario aclarar conceptos básicos teóricos sobre los diferentes tipos de calderas, su funcionamiento, y modelamiento de sistemas. En este capítulo se explicará sobre los modelos matemáticos, los diferentes tipos de calderas y los diferentes softwares de simulación con sus limitaciones.

### 2.1 Modelos Matemáticos

Los modelos matemáticos son un conjunto de ecuaciones que describen el comportamiento de un sistema para poder comprender su funcionamiento y de ser el caso controlarlo. El sistema al ser una

combinación de procesos que actúan y realizan un objetivo determinado, puede ser: físico, biológico, económico y similares. [2]

Los modelos matemáticos se dividen principalmente en estáticos y dinámicos. En los modelos estáticos las variables son constantes en todo momento y su respuesta no cambia en función del tiempo. Sin embargo en los modelos dinámicos, algunas de las variables que intervienen en el sistema cambian con respecto al tiempo, describiendo trayectorias temporales. El objetivo del análisis de este modelo es estudiar el comportamiento de algunos de sus elementos [2].

Los sistemas pueden ser controlados. Cuando existe un control en lazo cerrado el sistema tiene una relación específica entre la salida y la entrada de referencia (error), donde el resultado es utilizado como entrada en el controlador para disminuir el mismo en el nuevo ciclo del sistema y llevar la salida a un valor deseado. Cuando no existe un controlador la salida no tiene influencia alguna en el control, tampoco existe retroalimentación por lo que no se mide la salida y no se la compara con la entrada. En la práctica este sistema funciona solo bajo las calibraciones de los equipos utilizados en el sistema,

por lo que no hay control sobre las posibles perturbaciones del medio en el que se encuentra el sistema.

## **2.2 Calderas**

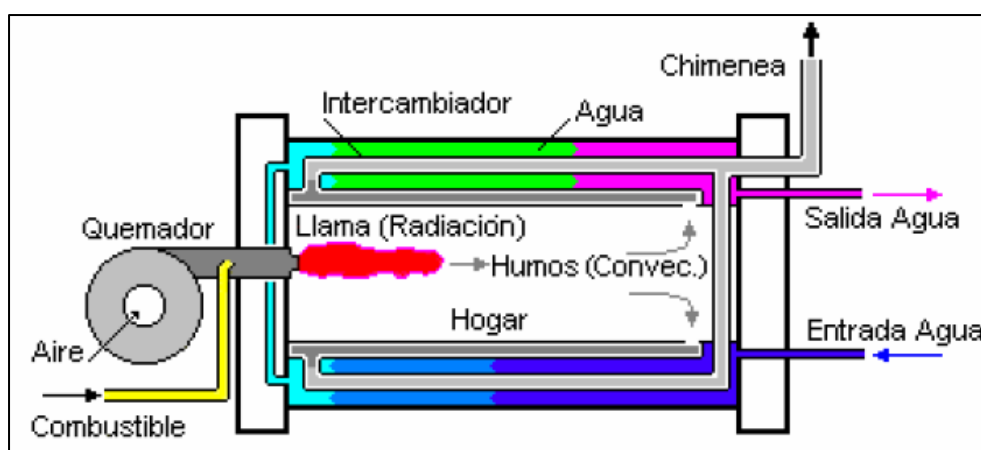
La caldera es un intercambiador de calor donde la energía es generada por un proceso de combustión o por calor contenido en el gas que circula a través de la misma. Este calor es suministrado a un fluido que se calienta o vaporiza que luego es transportado a un consumidor en el cual cederá su energía. Físicamente una caldera convierte la energía química del combustible en energía térmica o la energía de gases calientes en energía térmica, cuando no intervienen combustibles líquidos.

Como se aprecia en la Figura 2.1 tenemos la estructura general de una caldera con la señalización de cada una de sus partes. Sus componentes principales son el tambor y la cámara de vapor. El tambor es el espacio donde se encuentra el agua al interior de la caldera, por lo que de acuerdo al tamaño del tambor se distingue la capacidad de la caldera. De acuerdo a la relación que hay entre el tambor y la superficie de calefacción se distinguen las calderas de gran volumen, mediano y pequeño volumen de agua [3]. La cámara



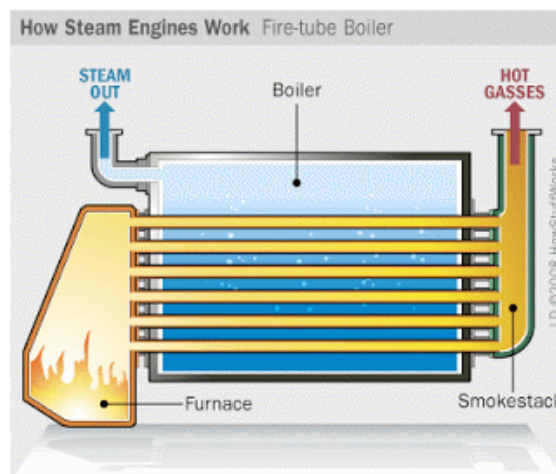
de vapor es el espacio ocupado por el vapor en la caldera, donde se separa el vapor del agua. Mientras más variable sea la producción de vapor, mayor debe ser su volumen de manera que aumente la distancia entre el nivel del tambor y la cámara de vapor para su separación. Se la llama también separador líquido-vapor [4].

Además dentro de la caldera ubicamos el quemador el cual enciende el combustible y el hogar donde se realiza la combustión. El intercambiador de calor es donde ocurre el flujo de calor desde los gases hasta el agua a través de la superficie de los tubos de intercambio. En ellas también se generan las burbujas de vapor [4]. La chimenea es la vía de escape de los humos y gases de la combustión después de haber realizado el intercambio de calor con el agua.



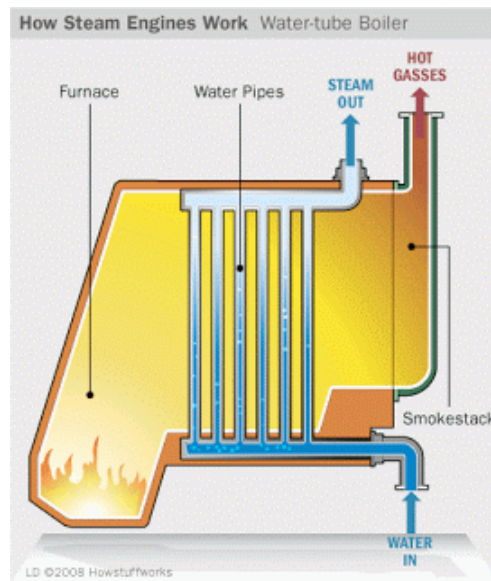
**Figura 2.1:** Estructura general de una caldera [5]

Una vez explicada la estructura general de las calderas especificamos la clasificación de las calderas denotando la que será simulada. Las calderas pueden ser eléctricas o térmicas y pueden dividirse en las cuales se dividen en pirotubulares y acuotubulares. Las primeras o también llamadas “Calderas de tubos de Humo” (Figura 2.2), los gases de combustión pasan por dentro de los tubos y el agua circula en la parte externa, evaporándose al contacto con los mismos. Son generalmente usadas para pequeñas y medianas potencias.



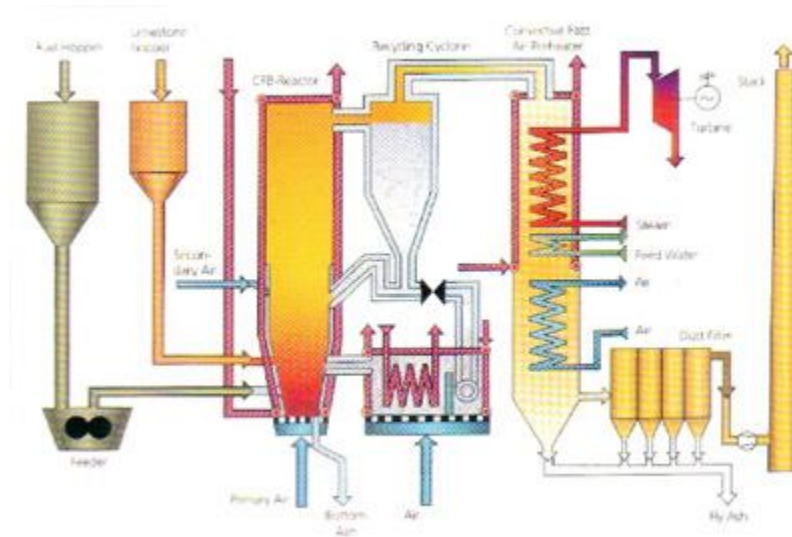
**Figura 2.2:** Caldera pirotubular [6]

Las calderas acuotubulares (Figura 2.3) se dan cuando el flujo de trabajo circula por los tubos durante su calentamiento. Las calderas acuotubulares se las emplea cuando se trabajan a altas presiones, por lo que generan mayor potencia de salida. Estas últimas son las seleccionadas para la posterior simulación.



**Figura 2.3:** Caldera Acuotubular [6]

Dentro de las calderas tenemos la forma de combustión por lecho fluidizado (Figura 2.4). Esta forma de combustión consiste en colocar el combustible: biomasa, carbón, entre otros sobre una cama o lecho fluido de cenizas, piedra caliza, etc. Donde por transferencia de calor se calienta el combustible sólido mientras se bombea aire hacia arriba para la formación de remolinos favoreciendo la mezcla del gas y combustible. El lecho no se puede fundir por lo que la temperatura esta entre los 850-900°C.



**Figura 2.4:** Caldera con lecho fluidizado [7]

La producción de calor se puede dar con un flujo agua-vapor (Figura 2.5). El agua suministrada a la caldera es calentada con un calor constante hasta que llega a su punto de ebullición y se transforma en vapor, este vapor se lo transporta a través de tubos hacia la cámara de vapor donde se separa el agua del vapor para posteriormente ser recirculada a la caldera. Los tubos por donde circula el vapor pueden ser llamados ascendentes o descendentes según sea el caso de la dirección del flujo. Esta condición será tomada en cuenta para la simulación.

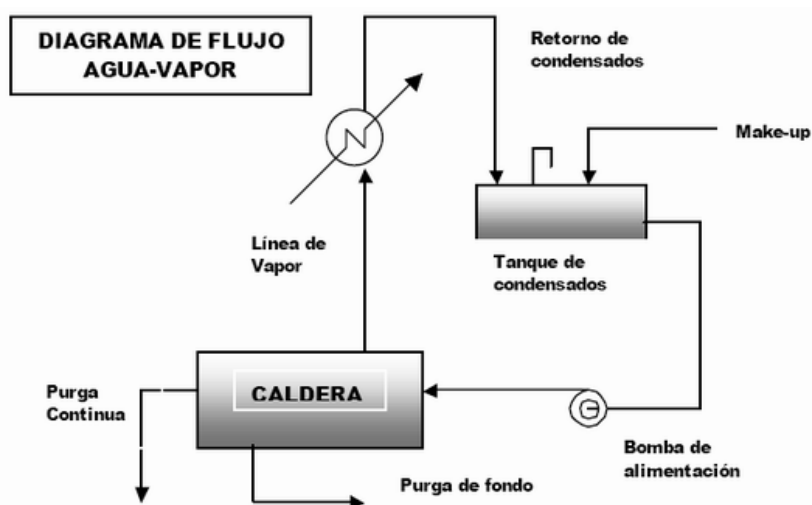


Figura 2.5: Caldera de Flujo Agua – Vapor [8]

### 2.3 Simuladores

Matlab es un programa que utiliza un lenguaje de programación de alto nivel, que permite la creación de interfaces de usuario con programas escritos en otros idiomas como son C, C++, Java, Fortran y Python, dando libertad al usuario de programar en cualquiera de estos programas. A su vez tiene un paquete Simulink, el cual permite la simulación grafica multidominio de diseños de modelos dinámicos y sistemas embebidos. La desventaja de este programa es que es un producto exclusivo de MathWorks por lo que para poder usarlo es necesario adquirir una licencia, en el caso de Simulink las diferentes librerías deben ser compradas por separado de acuerdo al requerimiento del usuario.

Scilab es un programa con código abierto y un alto nivel de programación. Se utiliza para el procesamiento de imágenes y modelado y simulación de sistemas dinámicos a través de su entorno Xcos, similar a Simulink, que posee paquetes que incluyen herramientas para simular los diferentes tipos de sistemas eléctricos, hidráulicos, de control, etc.

# Capítulo 3

## 3. MODELAMIENTO MATEMÁTICO DEL SISTEMA

Para desarrollar el modelo matemático del sistema es necesario seleccionar una caldera con sus especificaciones físicas, materiales utilizados en su construcción y consideraciones necesarias para realizar el modelamiento. En este capítulo se estudiara a fondo las características físicas de la caldera seleccionada y las suposiciones realizadas para la obtención del modelo. La caldera a ser modelada es acuotubular, este tipo de caldera es utilizada para la generación de altas potencias de energía eléctrica. El modelo especificado por el fabricante corresponde a la caldera P16 marca Annex con una potencia de salida de 160MW [10]. Como se estableció previamente la selección de este equipo se debe a las pruebas previamente realizadas en el mismo para la obtención de los valores de las constantes y variables de dominio del sistema. Las

calderas están fabricadas de acero inoxidable o cobre, se utiliza cobre para minimizar costos en su fabricación.

Previo a la obtención del modelo se debe considerar la excelente eficiencia del intercambiador de calor que poseen las calderas para la ebullición y condensación del agua, esto se debe a su diseño de tubo carcaza formado por tubos en forma de U que tienen una gran área de transferencia de calor [2], asegurando un equilibrio térmico con la mezcla de agua y vapor saturados en cualquier parte del sistema. La parte clave en este sistema es la rápida liberación o absorción de energía almacenada con el cambio de presión, la misma que se debe al equilibrio térmico existente en cada parte del sistema producido por el intercambiador de calor antes mencionado. El cambio de presión en el sistema y el equilibrio térmico del mismo permite un cambio de temperatura uniforme en cada una de las partes del sistema.

Este sistema es un proceso isotérmico debido a que se trabaja a una temperatura constante durante todo su funcionamiento, en el caso de las calderas se trabaja con una temperatura de saturación del vapor debido a que mejora la eficiencia del intercambiador de calor y disminuye costos operativos [9]. Como se estableció previamente el sistema tiene un

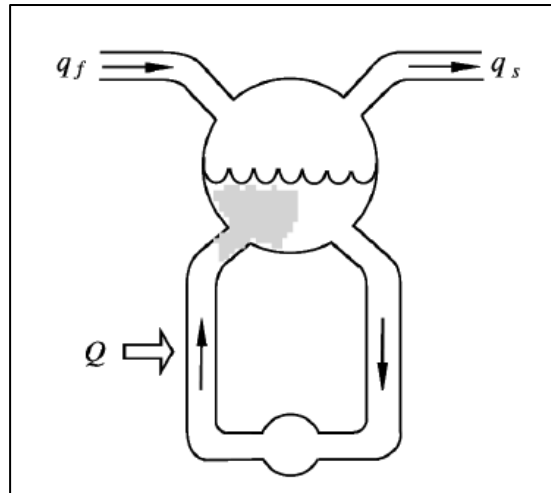


equilibrio térmico por lo que la temperatura de los tubos y el tambor es casi igual a la temperatura de saturación del vapor, de acuerdo a Amströn y Bell [1] las pruebas físicas hechas en la caldera reflejan que la temperatura de los tubos es cercana a la temperatura de saturación del vapor. Se considera constante el nivel del tambor en la caldera debido a que no se toma en cuenta la distribución de agua y vapor en el sistema [1] despreciando el fenómeno físico que se produce dentro del tambor que muestra el nivel como crestas de agua, este fenómeno se da cuando se abre la válvula de vapor provocando que el nivel del agua en el tambor aumente debido a la caída de presión en el tambor haciendo que las burbujas de vapor debajo del nivel del agua se hinchen produciendo las crestas de agua en el nivel [1], estas causan un comportamiento de fase no mínima en el nivel del tambor [1].

### **3.1 Balance de ecuaciones**

El comportamiento de la presión dentro del tambor puede ser descrito en base a las ecuaciones de balance de masas y energías dentro del sistema. Se considerará un sistema básico de la caldera como es el que se muestra en la Figura 3.1. Donde no se considera el número de tubos elevadores y bajantes, tomando en cuenta sólo el

calor administrado por el intercambiador de calor  $Q$ , el flujo másico de alimentación de agua  $q_f$  y el flujo másico del vapor  $q_s$ .



**Figura 3.1:** Grafico esquemático de la caldera [1]

Previo a la obtención de las ecuaciones según los balances de masa y energía, se deben establecer las diferentes variables que intervienen en el sistema como son el volumen  $V$ , la densidad relativa  $\rho$ , la energía interna almacenada en la masa  $u$ , la entalpia  $h$ , la temperatura  $t$ , el flujo másico  $q$  y el calor específico del cobre o metal  $C_p$ . Para una correcta representación de las variables se especificarán los subíndices de cada una los cuales se referirán al vapor  $s$ , al agua  $w$ , al metal  $m$  y al sistema total  $t$ . La variable  $mt$  es la masa de cobre o acero inoxidable total del sistema, que implica la masa del tambor y la de los tubos elevadores y bajantes. Con las variables definidas previamente realizaremos el balance de masas global del sistema, que representa la cantidad cuantitativa y

cualitativa de los requerimientos de materiales involucrados en el sistema [4] y el balance de energía global que simboliza la cantidad de energía que interviene en el sistema. El balance de masas global es:

$$\frac{d}{dt}[\rho_s V_{st} + \rho_w V_{wt}] = q_f - q_s \quad (3.1)$$

El balance de energía global es:

$$\frac{d}{dt}[\rho_s u_s V_{st} + \rho_w u_w V_{wt} + m_t C_p t_m] = Q + q_f h_f - q_s h_s \quad (3.2)$$

La energía interna del sistema puede ser representada de acuerdo a la entalpia, presión y densidad de acuerdo a la siguiente ecuación:

$$u = h - p/\rho \quad (3.3)$$

Reemplazando la ecuación (3.3) en (3.2) obtenemos:

$$\frac{d}{dt}[\rho_s h_s V_{st} + \rho_w h_w V_{wt} - p V_t + m_t C_p t_m] = Q + q_f h_f - q_s h_s \quad (3.4)$$

Esta ecuación describe el comportamiento de la caldera en función de las entalpias y densidades del agua y vapor respectivamente. Dado que la suma de los volúmenes de vapor y agua total del sistema representan el volumen total de las variables que intervienen el proceso tenemos que:

$$V_{wt} + V_{st} = V_t \quad (3.5)$$

Se puede representar la temperatura  $t_m$  en función de la presión debido al equilibrio térmico y las condiciones del sistema previamente especificadas. Combinando las ecuaciones 3.1, 3.3 y 3.4 obtenemos

un modelo diferencial de la caldera en función de la presión, entalpías y temperatura cuyos valores pueden ser determinados con las tablas de vapor saturado, las cuales se anexan al final de este documento. Simplificando el sistema al multiplicar la ecuación 3.3 por la variable  $h_w$  menos la ecuación 3.4 obtenemos:

$$\begin{aligned} h_c \frac{d}{dt}(\rho_s V_{st}) + \rho_s V_{st} \frac{dh_s}{dt} + V_{wt} \rho_w \frac{dh_w}{dt} - V_t \frac{dp}{dt} + m_t C_p \frac{dt_s}{dt} \\ = Q - q_f(h_w - h_f) - q_s h_c \end{aligned} \quad (3.6)$$

Donde:

$$h_c = h_s - h_w \quad (3.7)$$

Debido a que en las condiciones del sistema se estableció que el nivel del agua en el tambor era constante, se desprecian las variaciones de volumen de vapor en la ecuación (3.6) obteniendo:

$$\frac{dp}{dt} e_1 = Q - q_f(h_w - h_f) - q_s h_c \quad (3.8)$$

$$e_1 = h_c V_{st} \frac{d\rho_s}{dt} + \rho_s V_{st} \frac{dh_s}{dt} + V_{wt} \rho_w \frac{dh_w}{dt} + m_t C_p \frac{dt_s}{dt} - V_t \quad (3.9)$$

El término  $V_t$  representa la relación entre la energía interna y la entalpia del sistema, según Denn [11] este término se lo puede despreciar en la dinámica del sistema. De acuerdo a los experimentos realizados por Amström y Bell [9] en la caldera P16 marca Annex de 160MW los valores correspondientes a los términos previamente especificados corresponden a la tabla 1 donde podemos

notar que los coeficientes predominantes del sistema son  $V_{wt}\rho_w \frac{dh_w}{dt}$ ,  $m_t C_p \frac{dt_s}{dt}$  lo que quiere decir que el cambio de las energías almacenadas en el agua y el cobre son los fenómenos físicos que predominan en la dinámica de la presión en el tambor.

**Tabla 1:** Valores numéricos de los coeficientes de e1 a una presión normal de operación

Caldera	$h_c V_{st} \frac{d\rho_s}{dt}$	$\rho_s V_{st} \frac{dh_s}{dt}$	$V_{wt}\rho_w \frac{dh_w}{dt}$	$m_t C_p \frac{dt_s}{dt}$	$V_t$
P16-G16 80MW	360	-40	2080	1410	85
P16-G16 160MW	420	-40	1970	1410	85

Reemplazando los valores de la tabla 1 en la ecuación 3.9 obtenemos el valor de la constante  $e_1 = 3725$  cuando la caldera trabaja a media potencia. El modelo para la caldera P16 marca Annex de 80MW queda definido por la siguiente ecuación:

$$\frac{dp}{dt} = \frac{1}{3725} [Q - q_f(h_w - h_f) - q_s h_c] \quad (3.10)$$

La ecuación 3.10 describe en gran parte los cambios de presión en el tambor de la caldera en relación al flujo de calor suministrado por el intercambiador de calor, flujo másico de la alimentación de agua, flujo másico del vapor y temperatura del agua de alimentación. La ventaja de este modelo se debe a la obtención de los valores de las

diferentes entalpías, de las tablas de vapor saturado por los motivos previamente mencionados. De acuerdo a la tabla de vapor respectiva los valores de entalpia se obtienen para una temperatura y presión específica, se procede a utilizar la temperatura de saturación del agua como fija y variar la presión para poder describir el modelo matemático del sistema.

Según Amströn y Bell, 2000 [9] el calor generado  $Q$  se lo puede escribir como la diferencia de la energía almacenada del vapor y agua de alimentación, como se establece en la ecuación:

$$Q = q_s h_s - q_f h_f \quad (3.11)$$

Después de haber establecido la dinámica de la presión del tambor en relación a las entalpías y calor del sistema, podemos implementar estas ecuaciones en un software de simulación obteniendo las entalpías de los diferentes estados del agua a través de una ecuación que represente el comportamiento de la misma o con la ayuda de las tablas de vapor.

# Capítulo 4

## 4. IMPLEMENTACIÓN

En el capítulo anterior se diseñó el modelo matemático del comportamiento dinámico de la presión del tambor en una caldera en función de las diferentes entalpías del sistema y el calor suministrado por el intercambiador de calor  $Q$ . En este capítulo se procederá a explicar la forma de implementar el respectivo modelo del sistema en los diferentes software de simulación como son Matlab/Simulink y Scilab/Xcos, detallando las diferentes configuraciones realizadas para poder obtener una respuesta en el tiempo de la presión del tambor.

El modelo matemático de la caldera a realizar se fundamenta en las propiedades termodinámicas de las sustancias que intervienen en el proceso como son el agua y el vapor. Las propiedades de las sustancias son la entalpia y la presión, pero debido a que no existen tablas que las relacionen

directamente previamente encontraremos la temperatura de las mismas con una tabla que relaciona la presión con la variable antes mencionada.

Los flujos que intervienen en el modelo corresponden a los de vapor, agua y calor, donde el calor puede ser representado en función de las entalpías y flujos másicos del agua y vapor, como podemos notar en las ecuaciones (3.10) y (3.11) previamente explicadas en el capítulo anterior.

En el caso del flujo másico de agua este será representado por una constante asumiendo el caso de que en la alimentación de agua de la caldera no existe perturbación alguna para alterar su flujo, el flujo másico de vapor será representado por una señal cuadrada que simule el comportamiento de la válvula de salida, haciendo que varíe su flujo en medianas oscilaciones. Los parámetros para la respectiva simulación se los tomaron de las pruebas realizadas por Armströn y Bell [1] debido a que las constantes previamente escogidas se las tomaron del mismo estudio.

Dado que el sistema puede ser expresado como una ecuación de primer orden la complejidad de simulación se basa en la elaboración de las tablas de vapor y en el diseño de la señal de perturbación en el flujo másico del vapor, por lo que analizaremos el proceso de implementación en las dos secciones antes mencionadas y en la implementación final del modelo matemático.



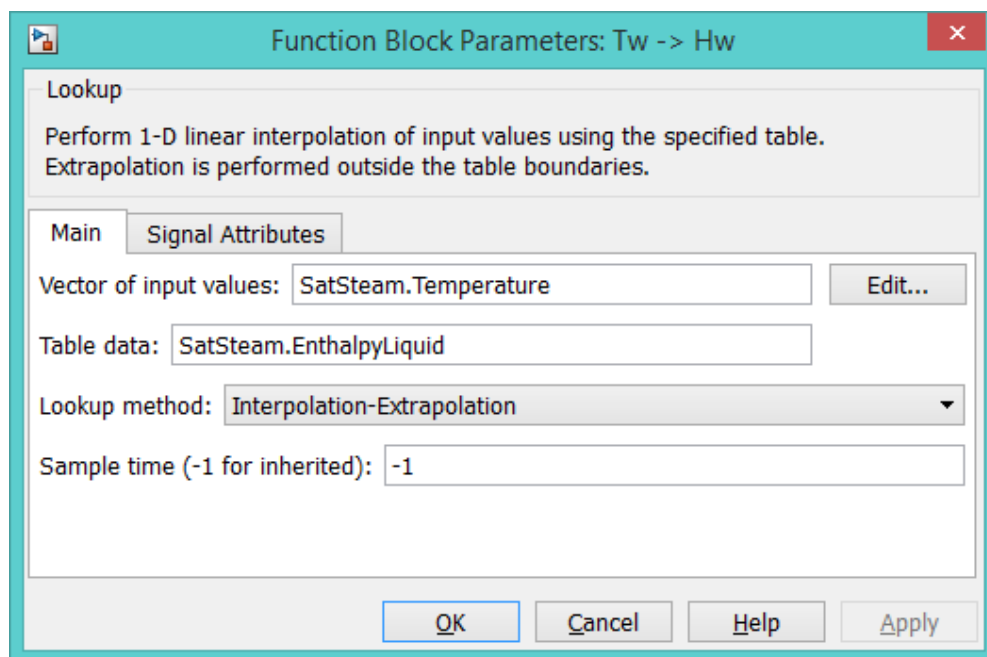
#### 4.1 Matlab/Simulink

Simulink es un entorno de programación visual que funciona sobre el entorno de programación de Matlab, debido a su entorno la programación se realiza mediante la conexión de diferentes bloques y configuración de sus respectivos parámetros de acuerdo al sistema.

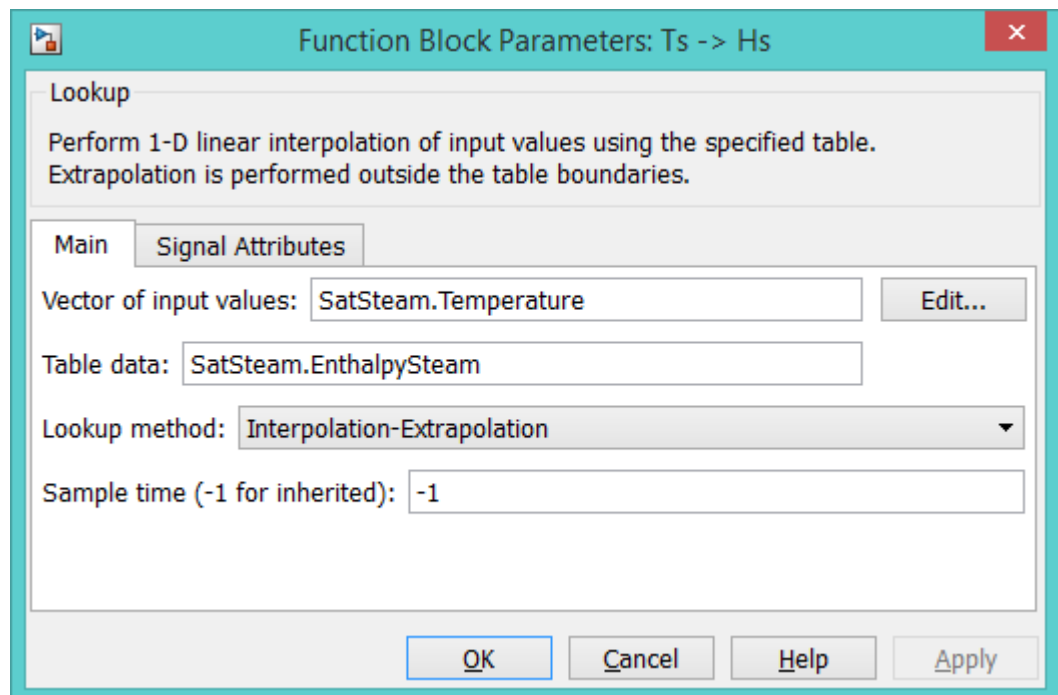
Para poder representar las tablas se procedió a utilizar el bloque *1-D Lookup table* de la librería *Lookup table* de *Simulink* [10]. Es necesario tener los respectivos datos de cada tabla en variables almacenadas en el *workspace* para posteriormente utilizarlos en la configuración del bloque antes mencionado.

En la configuración del bloque *1-D Lookup Table* se hace doble clic en el mismo y aparece una ventana de configuración como la que se muestra en la figura 4.2, donde se colocarán los datos de las tablas previamente almacenadas en el *workspace* como los parámetros de la tabla. En el parámetro *Vector of input values* se coloca la variable de entrada en el bloque que en este caso será la temperatura, a partir de esta variable se obtendrá la entalpía de acuerdo a la variable

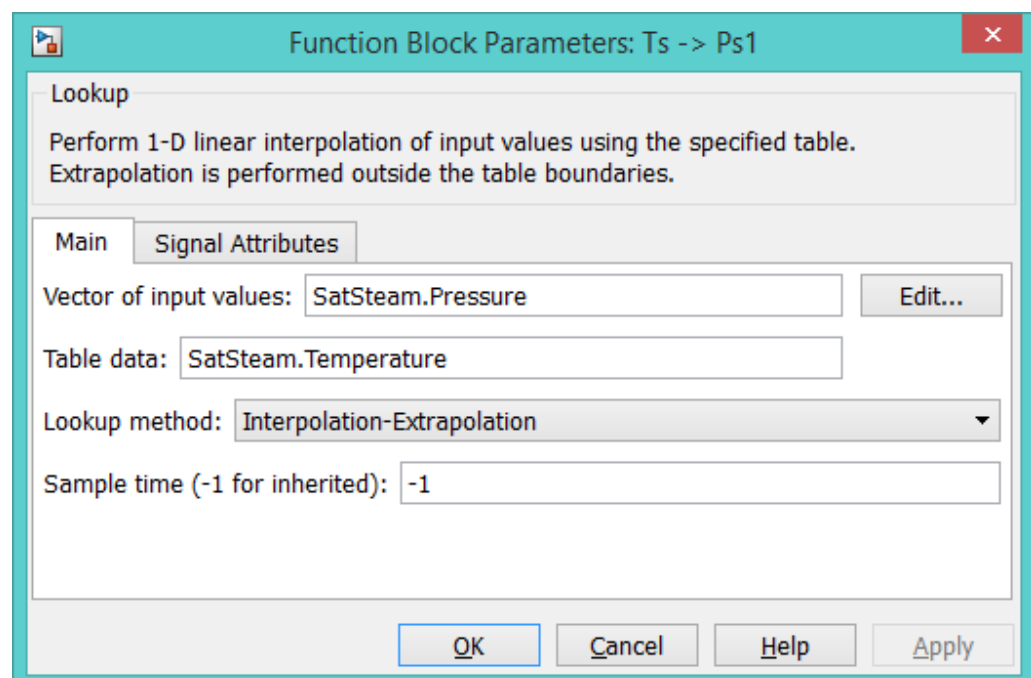
especificada en el parámetro *Table data*. Es necesario seleccionar el método de búsqueda "*Lookup method*" como *Interpolation-Extrapolation* que representa la manera de obtener los valores no especificados en la tabla. En las tablas de vapor se utiliza el método de interpolación para establecer un resultado cercano al real de acuerdo a los valores experimentales obtenidos. Las ventanas de configuración para cada una de las diferentes tablas se muestran a continuación:



**Figura 4.1** Parámetros tabla de vapor saturado T vs H



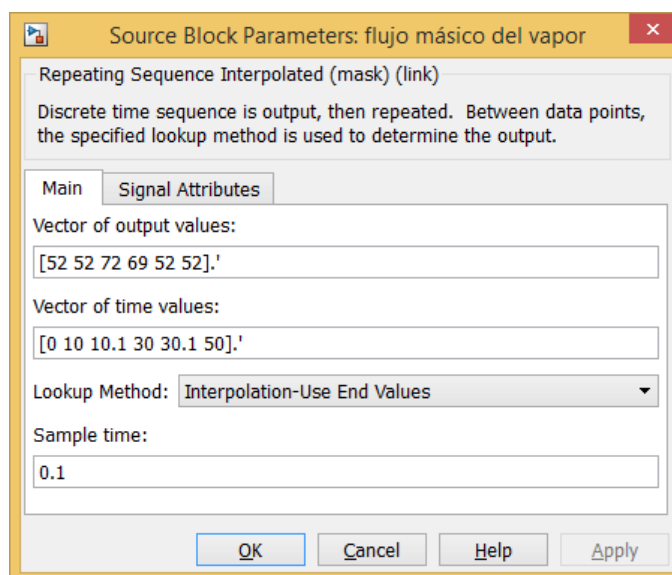
**Figura 4.2** Parámetros tabla de vapor saturado T vs Hs



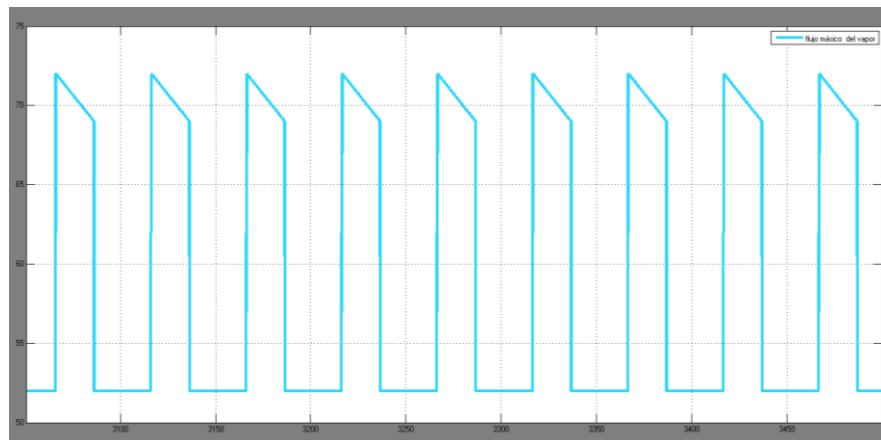
**Figura 4.3** Parámetros tabla P vs T

Obtenidas las tablas de vapor y la gráfica Temperatura vs Presión, podemos obtener las respectivas entalpías en base a la temperatura del vapor que sale del sistema la cual obtenemos con la presión de salida interpolada según la gráfica antes mencionada, obteniendo así todas las variables de entalpia que intervienen en el sistema.

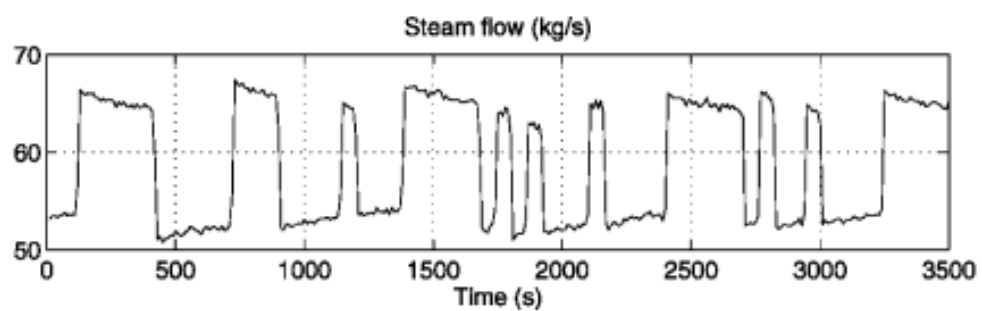
Para simular el flujo oscilante de vapor a la salida del tambor se utilizó el bloque *Repeated Sequence Interpolated* que repite una gráfica, previamente definida por puntos, en el tiempo de simulación. La especificación de cada parámetro se la realiza en la ventana de configuración del bloque, la misma que aparece al realiza doble clic. Los parámetros configurados en el bloque se muestran en la figura 4.4 abajo detallada.



**Figura 4.4** Parámetros de configuración del flujo másico del vapor

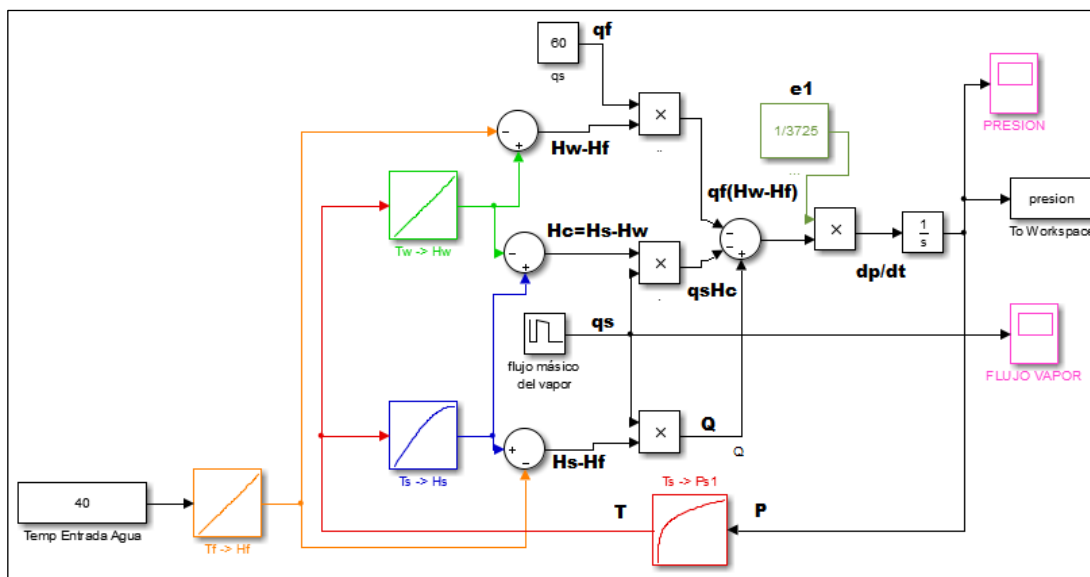


**Figura 4.5** Simulación del flujo másico del vapor perturbado



**Figura 4.6** Flujo másico del vapor perturbado

Después de haber configurado los parámetros de las tablas y variables a utilizar, se procede a la colocación y conexión de cada uno de los bloques a intervenir en el sistema de acuerdo a la ecuación y a la simulación del mismo con un tiempo de simulación de 3500 s. Este tiempo se debe a que el sistema empieza a estabilizarse en alrededor de los 1500-2000 s según las pruebas realizadas por Armströn y Bel [1].



**Figura 4.7** Implementación de una caldera en Simulink

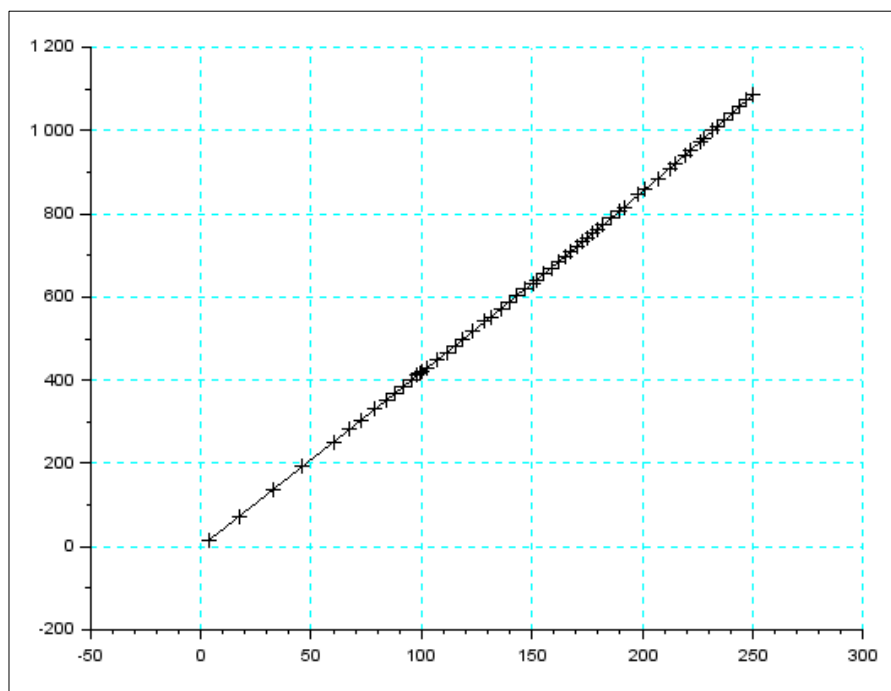
Para obtener la respuesta en el tiempo de cada una de las señales de salida se procedió a colocar visores a cada una de ellas, los mismos que permiten visualizar las variables en función del tiempo de simulación del programa.

## 4.2 Scilab/Xcos

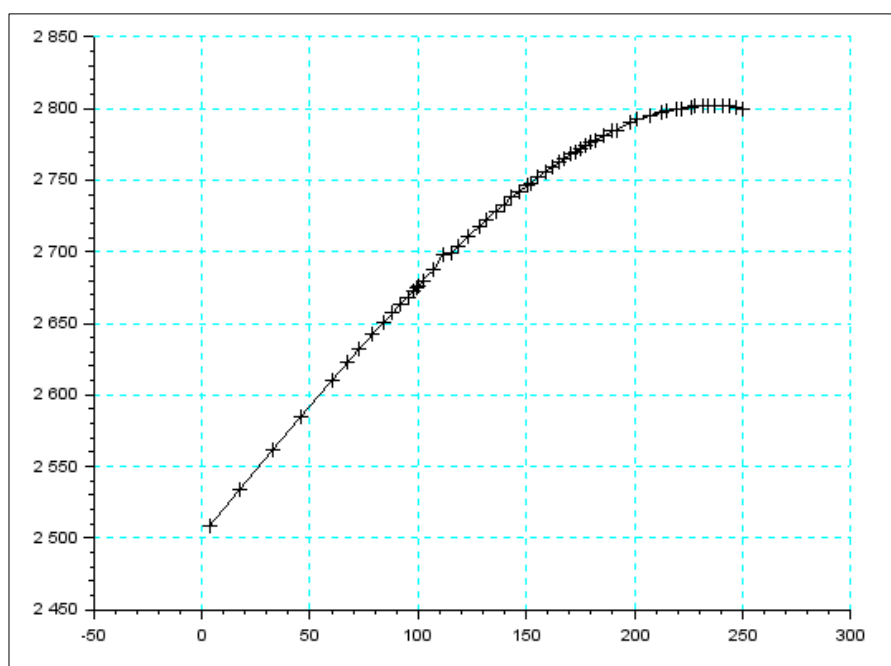
Xcos al igual que *Simulink* es un editor gráfico que permite la simulación de modelos de sistemas dinámicos e híbridos, por lo que la forma de programación se basa bloques correspondientes a las diferentes librerías de acuerdo a los sistemas dinámicos.

Para la implementación de las respectivas tablas utilizaremos el bloque *LOOKUP\_f* de la paleta *Lookup tables palette*, de las paletas de Xcos, este bloque realiza la interpolación respectiva del dato de entrada en relación a una gráfica previamente definida como parámetro en el mismo. La grafica se la programa en la consola “*Console*” donde se crean las variables con los datos de cada tabla y posteriormente se grafica cada tabla con la ayuda del comando *plot(x,y)* para cada uno de los respectivos casos, luego en la ventana de la gráfica se procede a guardar el correspondiente archivo *.scg* en una ruta conocida. [11]

Posteriormente se procede a abrir el bloque *LOOKUP\_f* donde aparece un gráfico por default el cual lo eliminamos con la opción borrar en el comando editar y procedemos a subir con la ayuda de la opción *load* en el comando File, la respectiva gráfica previamente guardada, luego la guardamos con la opción *save* y así terminamos de implementar todas las tablas. A continuación se presentan las respectivas configuraciones en las figuras 4.7, 4.8 y 4.9.

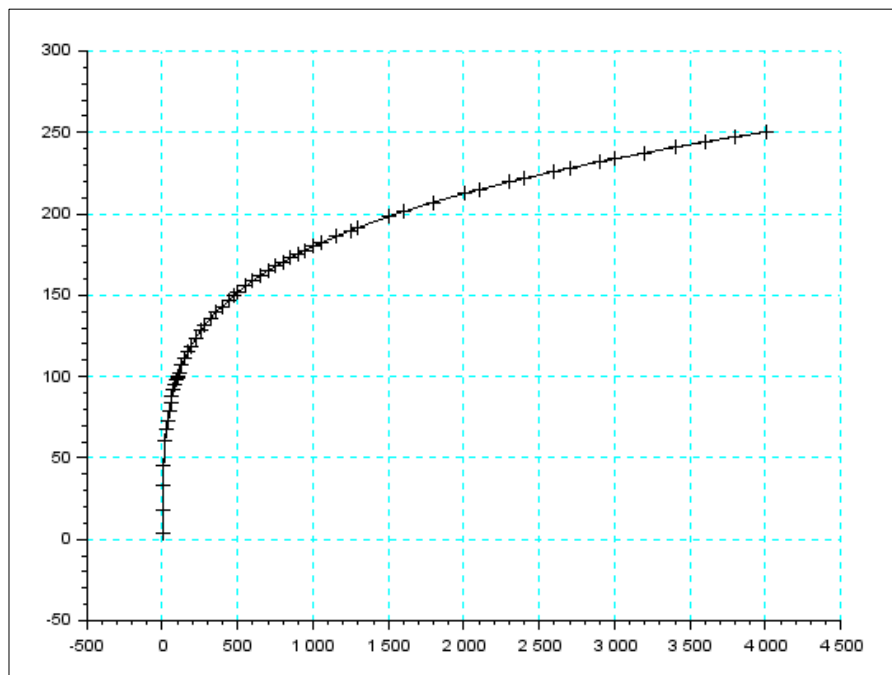


**Figura 4.8 Parámetros T vs Hw Scilab**



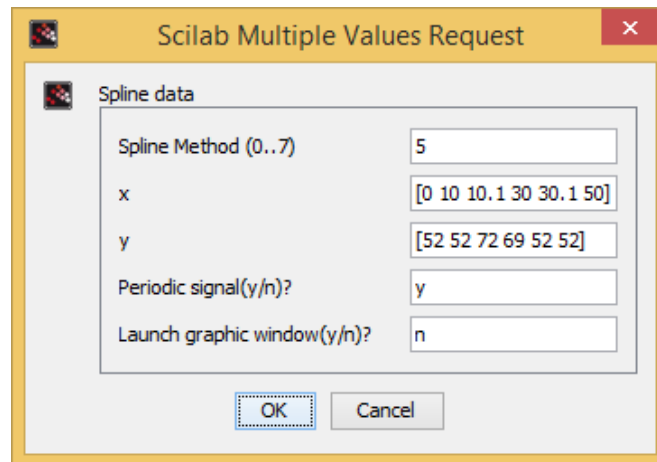
**Figura 4.9 Parámetros T vs Hs Scilab**





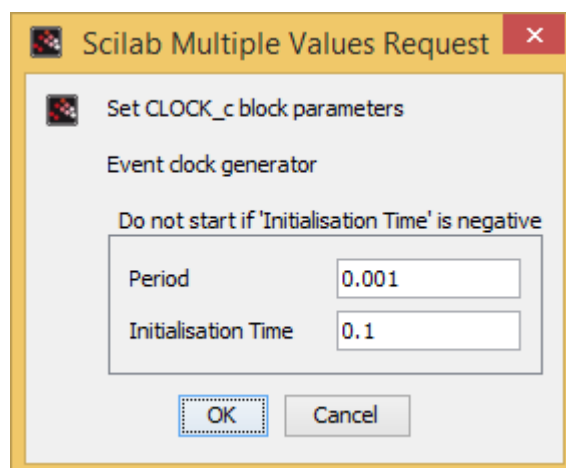
**Figura 4.10 Parámetros T vs P Scilab**

Para graficar el flujo de vapor a la salida del tambor se utilizó el bloque *Signal Builder* de la paleta *Sources palette* que genera eventos sucesivos retroalimentados, habiendo la posibilidad de causar eventos en un punto discontinuo de la señal. La especificación de los parámetros del bloque se lo realiza punto a punto en una ventana de configuración como se muestra en la Figura 4.11 Configuración Scope en Scilab

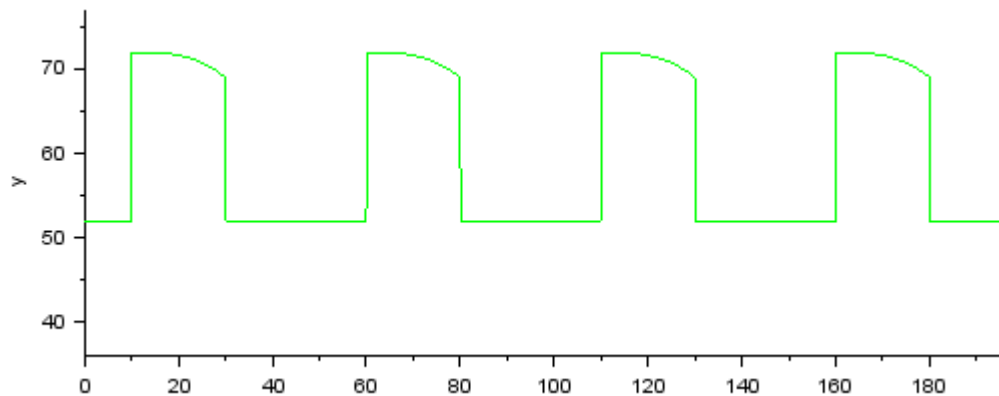


**Figura 4.11 Configuración Scope en Scilab**

En el proceso de simulación a diferencia de otros simuladores es necesario utilizar un bloque de reloj *CLOCK\_cblock* de la paleta *Palette Sources*, en el cual se establece el periodo y el tiempo de inicialización de la simulación.



**Figura 4.12 Configuración de reloj en Scilab**



**Figura 4.13 Simulación del flujo másico del vapor Scilab**

Para poder visualizar el comportamiento de la variable en el tiempo se selecciona el bloque *CM Scope* de la paleta *Sinks palette* donde se configuran los valores máximos y mínimos de los ejes de las diferentes variables a graficar, el mismo también recibe una señal de entrada que es el bloque antes mencionado *CLOCK\_cblock*.

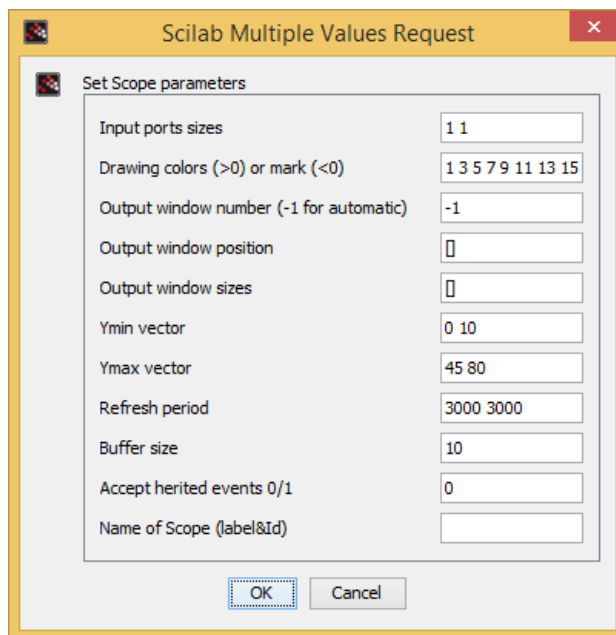


Figura 4.14 Parámetros de configuración CM Scope

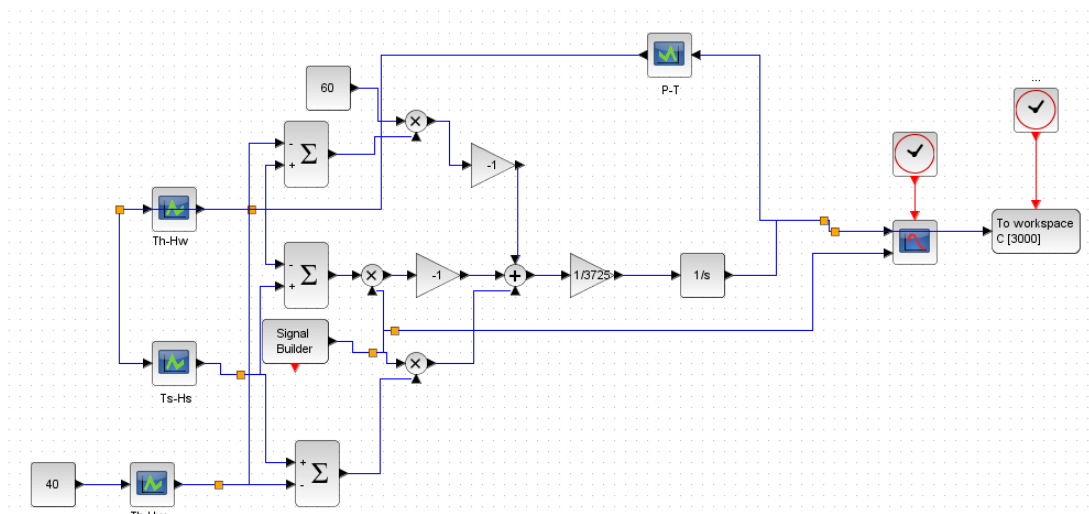


Figura 4.15 Implementación del modelo en Scilab

# Capítulo 5

## 5. RESULTADOS

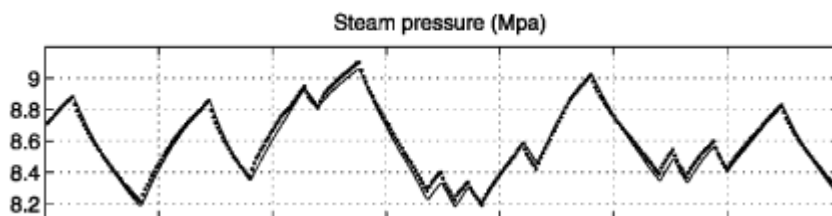
En los anteriores capítulos describimos el procedimiento para la obtención del modelo y la implementación del mismo en cada software de simulación, el presente capítulo está orientado a contrastar los resultados obtenidos de las simulaciones entre sí y con datos reales.

Se procede a comparar el comportamiento del sistema para cierta condición y verificar si efectivamente corresponde al modelo propuesto.

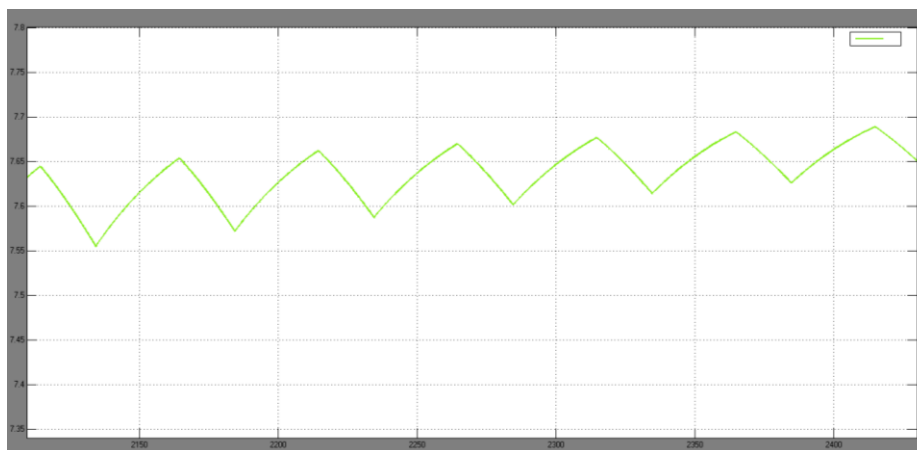
### 5.1 Respuesta del sistema

Debido a que en la simulación del sistema se procedió a simular el flujo de salida de la válvula de vapor, se procede a comparar los datos obtenidos con

los resultados de las pruebas experimentales obtenidas por Amströn y Bell [1] que se muestran a continuación.



**Figura 5.1 Presión del tambor [1]**



**Figura 5.2 Presión del tambor Simulink**

En el capítulo 4 se estableció en la Figura 4.5 **Simulación del flujo másico del vapor perturbado** Figura 4.5 el flujo de vapor simulado por lo que según el modelo establecido se puede decir que es válido, ya que los errores se deben a que el flujo suministrado en el sistema es un flujo real y en nuestro caso es una simulación.

## 5.2 Cálculo de errores

Para poder establecer los criterios de diferenciación entre cada simulador realizaremos un análisis del tiempo de ejecución de cada uno y el error absoluto y relativo de cada uno de los valores obtenidos en las respectivas implementaciones.

Una medida para obtener la eficiencia del simulador es el tiempo de ejecución el cual se lo obtiene de diferentes maneras según sea el caso del simulador. En el programa de Matlab se creó el programa `eficiencia.m` del Anexo que manda a simular el modelo en *Simulink* al ser ejecutado con el botón *run and Time*. Como resultado obtuvimos el tiempo de ejecución del modelo en *Simulink* que fue de 153.86s como se muestra en la figura a continuación.

<b>Simulink Profile Report: Summary</b>	
<i>Report generated 13-May-2015 21:08:34</i>	
Total recorded time:	153.86 s
Number of Block Methods:	103
Number of Internal Methods:	9
Number of Model Methods:	11
Number of Nonvirtual Subsystem Methods:	2
Clock precision:	0.00000004 s
Clock Speed:	2401 MHz
To write this data as <code>modelocalderaProfileData</code> in the base workspace <a href="#">click here</a>	

**Figura 5.3 Reporte de rendimiento de Simulink**

En el caso de Scilab se creó una función `calderav1.sce` donde al igual que en el anterior programa se manda a simular el archivo de Xcos y

se obtiene el reporte mostrado en la Figura 5.4 con un tiempo total de 1971.92 segundos de ejecución.

```
-->exec('C:\Users\Usuario\Desktop\Caldera\Caldera_Simulados\SCILAB_cALDERAS\calderav1.sce', -1)
Warning : redefining function: calderav1 . Use funcprot(0) to avoid this message

|1|0      |0| 1: function []=fun()
|1|0      |1| 2: //EJECUCIÓN DEL MODELO DESDE SCILAB
|1|0.11   |3| 3: xcoss('caldera_Scilba.xcos')
|1|0.44   |4| 4: importXcosDiagram('C:\Users\Usuario\Desktop\Caldera\Caldera_Simulados\SCILAB_
|1|0      |4| 5: typeof(scs_m)
|1|0      |6| 6: scs m.props.context
|1|1971.92|5| 7: xcoss_simulate(scs_m, 4);
|1|0      |0| 8: endfunction
```

**Figura 5.4 Reporte de rendimiento de Scilab**

**Tabla 2 Tiempos de Ejecución**

TIEMPO EJECUCIÓN	SCILAB	MATLAB
	1971,92	153,86

Otro de los criterios importantes para poder establecer el software más confiable es la desviación estándar en las mediciones de cada uno. Para poder definir las desviaciones de cada uno de ellos lo ideal sería compararlos con datos reales obtenidos del sistema pero debido a que no podemos tener un generador de vapor físicamente con las características descritas, se calculan las desviaciones con los datos de los simuladores para una muestra de 3500 segundos con intervalos de 1 segundo, se considera este tiempo debido a que en este intervalo la variable se estabiliza.



En la tabla 3 se muestran los valores de las desviaciones para cada uno de los valores obtenidos de los simuladores. La muestra fue de los 100 últimos datos tomados a cada segundo de la simulación.

**Tabla 3 Desviación de la Presión en Simulink/Xcos**

<b>RESULTADOS EN ESTADO ESTABLE</b>	<b>PROMEDIO [Mpa]</b>	<b>DESVIACIÓN ESTÁNDAR</b>	<b>PORCENTAJE DE DESVIACIÓN</b>
Simulink	8,67	0,539	6,19%
Scilab	6,546	2,028	30,98%

# CONCLUSIONES

En los capítulos anteriores hemos definido las ecuaciones del modelo matemático, los parámetros de simulación y los resultados del mismo. Datos que nos permiten emitir un criterio sobre nuestro trabajo. Fundamentados en lo apreciado en el. Podemos concluir lo siguiente:

1. La respuesta del modelo matemático a la señal de flujo de vapor es similar al comportamiento de la obtenida en las pruebas experimentales de los textos citados.
2. Los software Matlab/Simulink y Scilab/Xcos tienen la capacidad de simular cualquier modelo matemático y tienen gran similitud en su entorno de programación gráfico.
3. El software Matlab/Simulink al tener una licencia se diferencia de los otros programas por su facilidad de manejo de los bloques y un rápido tiempo de ejecución; sin embargo este programa consume demasiada

memoria RAM al momento de utilizarlo. En el caso de Scilab/Xcos a pesar de tener una interfaz parecida al Simulink su interfaz es difícil de manejar ya que cada función específica tiene que ser programada previamente y su tiempo de ejecución supera en un mil por ciento al de Matlab/Simulink según la tabla 2.

4. Mediante el desarrollo de modelos matemáticos podemos diseñar equipos para que funcionen bajo las condiciones necesarias según sea el caso y simular su comportamiento en el sistema. El modelo propuesto posee variables y constantes que cambian de acuerdo al tamaño, tipo y uso necesario.
5. De acuerdo a los tiempos de ejecución Matlab es el más rápido en comparación con Scilab por lo que a pesar de ser este un software libre, no posee las competencias necesarias para poder reemplazar un simulador que utiliza algoritmos de procesamiento más rápido como Matlab.

# RECOMENDACIONES

1. En el proceso de obtención del modelo matemático se facilita el mismo con la forma física de la caldera, debido a que este hecho permite obtener las constantes físicas del sistema. En este caso se tuvieron que asumir ciertos casos y justificar los mismos
2. Tener una bibliografía que comprenda todo sobre el modelo a desarrollar. Esta puede tener artículos de revistas científicas (papers), normas de calidad que detallen sus condiciones físicas y el proceso de obtención de las constantes del sistema los que serán utilizados como guías para obtener el modelo.
3. Leer tutoriales sobre cada uno de los software y familiarizarse con los bloques a ser utilizados durante la implementación ya que pueden existir funciones o bloques prediseñados que pueden ser cambiados de acuerdo a la necesidad del modelo a ser implementado.

# **ANEXOS**

## ANEXO 1

### TABLAS DE VAPOR DE SATURACIÓN DE MATLAB

<b>PRESIÓN KPa</b>	<b>TEMPERATURA C</b>	<b>ENTALPÍA AGUA</b>	<b>ENTALPÍA VAPOR</b>
0,8	3,8	15,8	2509
2	17,5	73,5	2534
5	32,9	137,8	2562
10	45,8	191,8	2585
20	60,1	251,5	2610
28	67,5	282,7	2623
35	72,7	304,3	2632
45	78,7	329,6	2642
55	83,7	350,6	2650
65	88	368,6	2657
75	91,8	384,5	2663
85	95,2	398,6	2668
95	98,2	411,5	2673
100	99,6	417,5	2675
101,33	100	419,1	2676
110	102,3	428,8	2680
130	107,1	449,2	2687
150	111,4	467,1	2698
170	115,2	483,2	2699
190	118,6	497,8	2704
220	123,3	517,6	2711
260	128,7	540,9	2718
280	131,2	551,4	2722
320	135,8	570,9	2728
360	139,9	588,5	2733
400	143,1	604,7	2738
440	147,1	619,6	2742
480	150,3	633,5	2746
500	151,8	640,1	2748

<b>PRESIÓN Kpa</b>	<b>TEMPERATURA C</b>	<b>ENTALPÍA AGUA</b>	<b>ENTALPÍA VAPOR</b>
550	155,5	655,8	2752
600	158,8	670,4	2756
650	162	684,1	2759
700	165	697,1	2762
750	167,8	709,3	2765
800	170,4	720,9	2768
850	172,9	732	2770
900	175,4	742,6	2772
950	177,7	752,8	2774
1000	179,9	762,6	2776
1050	182	772	2778
1150	186	790	2781
1250	189,8	807	2784
1300	191,6	815	2785
1500	198,3	845	2790
1600	201,4	859	2792
1800	207,1	885	2795
2000	212,4	909	2797
2100	214,9	920	2798
2300	219,6	942	2800
2400	221,8	952	2800
2600	226	972	2801
2700	228,1	981	2802
2900	232	1000	2802
3000	233,8	1008	2802
3200	237,4	1025	2802
3400	240,9	1042	2802
3600	244,2	1058	2802
3800	247,3	1073	2801
4000	250,3	1087	2800

## ANEXO 1

### CÓDIGO PARA OBTENER EL RENDIMIENTO EN MATLAB

```
%CARGA Y SIMULACIÓN DEL MODELO  
open_system('modelocaldera.slx')  
sim ('modelocaldera.slx')
```



## ANEXO 2

### CÓDIGO PARA OBTENER EL RENDIMIENTO EN SCILAB

```
//DECLARACIÓN DE FUNCIÓN PARA EVALUAR RENDIMIENTO DEL  
SIMULADOR  
  
function []=calderav1()  
    //EJECUCIÓN DEL MODOELO DESDE SCILAB  
    xcos caldera_Scilba.zcos  
  
    importXcosDiagram("C:\Users\Usuario\Desktop\Caldera\Caldera_Simulados\SCILA  
B_cALDERAS\caldera_Scilba.zcos")  
    typeof(scs_m)  
    scs_m.props.context  
    xcos_simulate(scs_m,4);  
endfunction  
  
//EVALUACIÓN DEL SIMULADOR  
add_profiling("calderav1");  
calderav1();  
profile(calderav1)  
showprofile(calderav1)
```

## ANEXO 3

### CÓDIGO PARA EL FLUJO DE VAPOR MODELICA

```
block flujo_Vapor
//loadModel(Modelica);
import Modelica.SIunits;
import Modelica.RealOutput;
import Modelica.Blocks.Interfaces;
//Declaracion de constantes
//extends Modelica.Blocks;
Modelica.Blocks.Interfaces.RealOutput u;
parameter Integer nout = 1 "Number of outputs";
OutPort outPort(n = nout) "Connector of Real output signals";
parameter Real amplitude = 19 "Amplitude of trapezoid";
parameter SIunits.Time rising(final min = 0) = 0.01 "Rising
duration of trapezoid";
parameter SIunits.Time width(final min = 0) = 20 "Width
duration of trapezoid";
parameter SIunits.Time falling(final min = 0) = 0.01 "Falling
duration of trapezoid";
parameter SIunits.Time period(final min =
Modelica.Constants.small, start = 50) "Time for one period";
parameter Integer nperiod = -1 "Number of periods (< 0 means
infinite number of periods)";
parameter Real offset = 52 "Offset of output signal";
parameter SIunits.Time startTime = 20 "Output = offset for
time < startTime";
extends Interfaces.SO;
protected
Real n[nout] = outPort.signal;
parameter SIunits.Time T_rising = rising "End time of rising
phase within one period";
parameter SIunits.Time T_width = T_rising + width "End time of
width phase within one period";
parameter SIunits.Time T_falling = T_width + falling "End time
of falling phase within one period";
SIunits.Time T_start "Start time of current period";
Integer count "Period count";
initial algorithm
count := integer((time - startTime) / period);
T_start := startTime + count * period;
equation
when integer((time - startTime) / period) > pre(count) then
count = pre(count) + 1;
T_start = time;
end when;
```

```

y = offset + (if time < startTime or nperiod == 0 or nperiod >
0 and count >= nperiod then 0 else if time < T_start +
T_rising then amplitude * (time - T_start) / rising else if
time < T_start + T_width then amplitude else if time < T_start
+ T_falling then amplitude * (T_start + T_falling - time) /
falling else 0);
u = y;
annotation(Icon(coordinateSystem(preserveAspectRatio = true,
extent = {{-100, -100}, {100, 100}}), graphics = {Line(points
= {{-80, 68}, {-80, -80}}, color = {192, 192, 192}),
Polygon(points = {{-80, 90}, {-88, 68}, {-72, 68}, {-80, 90}},
lineColor = {192, 192, 192}, fillColor = {192, 192, 192},
fillPattern = FillPattern.Solid), Line(points = {{-90, -70},
{82, -70}}, color = {192, 192, 192}), Polygon(points = {{90, -
70}, {68, -62}, {68, -78}, {90, -70}}, lineColor = {192, 192,
192}, fillColor = {192, 192, 192}, fillPattern =
FillPattern.Solid), Text(extent = {{-147, -152}, {153, -112}},
lineColor = {0, 0, 0}, textString = "period=%period"),
Line(points = {{-81, -70}, {-60, -70}, {-30, 40}, {9, 40},
{39, -70}, {61, -70}, {90, 40}}, color = {0, 0, 0})),
Diagram(coordinateSystem(preserveAspectRatio = true, extent =
{{-100, -100}, {100, 100}}), graphics = {Polygon(points = {{-
81, 90}, {-87, 68}, {-75, 68}, {-81, 90}}, lineColor = {95,
95, 95}, fillColor = {95, 95, 95}, fillPattern =
FillPattern.Solid), Line(points = {{-81, 68}, {-81, -80}},
color = {95, 95, 95}), Line(points = {{-91, -70}, {81, -70}},
color = {95, 95, 95}), Polygon(points = {{89, -70}, {67, -65},
{67, -76}, {89, -70}}, lineColor = {95, 95, 95}, fillColor =
{95, 95, 95}, fillPattern = FillPattern.Solid), Polygon(points
= {{-46, -30}, {-48, -41}, {-44, -41}, {-46, -30}}, lineColor
= {95, 95, 95}, fillColor = {95, 95, 95}, fillPattern =
FillPattern.Solid), Line(points = {{-46, -30}, {-46, -70}},
color = {95, 95, 95}, pattern = LinePattern.Solid, arrow =
{Arrow.None, Arrow.None}), Polygon(points = {{-46, -70}, {-48,
-60}, {-44, -60}, {-46, -70}, {-46, -70}}, lineColor = {95,
95, 95}, fillColor = {95, 95, 95}, fillPattern =
FillPattern.Solid), Text(extent = {{-80, -46}, {-42, -55}},
lineColor = {0, 0, 0}, textString = "offset"), Text(extent =
{{-49, -71}, {-6, -81}}, lineColor = {0, 0, 0}, textString =
"startTime"), Text(extent = {{-80, 95}, {-47, 80}}, lineColor
= {0, 0, 0}, textString = "y"), Text(extent = {{66, -78}, {89,
-89}}, lineColor = {0, 0, 0}, textString = "time"),
Line(points = {{-31, 82}, {-31, -70}}, color = {95, 95, 95},
pattern = LinePattern.Dash), Line(points = {{-11, 59}, {-11,
40}}, color = {95, 95, 95}, pattern = LinePattern.Dash),
Line(points = {{19, 59}, {19, 40}}, color = {95, 95, 95},
pattern = LinePattern.Dash), Line(points = {{39, 59}, {39, -
30}}, color = {95, 95, 95}, pattern = LinePattern.Dash),
Line(points = {{-31, 76}, {59, 76}}, color = {95, 95, 95}),

```

```

Line(points = {{-31, 56}, {39, 56}}, color = {95, 95, 95}),
Text(extent = {{-3, 86}, {24, 77}}, lineColor = {0, 0, 0},
textString = "period"), Text(extent = {{-11, 68}, {18, 59}},
lineColor = {0, 0, 0}, textString = "width"), Line(points =
{{-43, 40}, {-11, 40}}, color = {95, 95, 95}, pattern =
LinePattern.Dash), Line(points = {{-40, 40}, {-40, -30}},
color = {95, 95, 95}, pattern = LinePattern.Solid, arrow =
{Arrow.None, Arrow.None}), Text(extent = {{-77, 11}, {-44,
1}}, lineColor = {0, 0, 0}, textString = "amplitude"),
Polygon(points = {{-31, 56}, {-24, 58}, {-24, 54}, {-31, 56}},
lineColor = {95, 95, 95}, fillColor = {95, 95, 95},
fillPattern = FillPattern.Solid), Polygon(points = {{-11, 56},
{-18, 58}, {-18, 54}, {-11, 56}}, lineColor = {95, 95, 95},
fillColor = {95, 95, 95}, fillPattern = FillPattern.Solid),
Polygon(points = {{-31, 76}, {-22, 78}, {-22, 74}, {-31, 76}},
lineColor = {95, 95, 95}, fillColor = {95, 95, 95},
fillPattern = FillPattern.Solid), Polygon(points = {{59, 76},
{51, 78}, {51, 74}, {59, 76}}, lineColor = {95, 95, 95},
fillColor = {95, 95, 95}, fillPattern = FillPattern.Solid),
Line(points = {{-81, -30}, {-31, -30}, {-11, 40}, {19, 40},
{39, -30}, {59, -30}, {79, 40}, {99, 40}}, color = {0, 0,
255}, thickness = 0.5), Polygon(points = {{-40, 40}, {-42,
29}, {-38, 29}, {-40, 40}}, lineColor = {95, 95, 95},
fillColor = {95, 95, 95}, fillPattern = FillPattern.Solid),
Polygon(points = {{-40, -30}, {-42, -20}, {-38, -20}, {-40, -
30}, {-40, -30}}, lineColor = {95, 95, 95}, fillColor = {95,
95, 95}, fillPattern = FillPattern.Solid), Line(points = {{59,
84}, {59, -30}}, color = {95, 95, 95}, pattern =
LinePattern.Dash), Polygon(points = {{39, 56}, {32, 58}, {32,
54}, {39, 56}}, lineColor = {95, 95, 95}, fillColor = {95, 95,
95}, fillPattern = FillPattern.Solid), Polygon(points = {{19,
56}, {26, 58}, {26, 54}, {19, 56}}, lineColor = {95, 95, 95},
fillColor = {95, 95, 95}, fillPattern = FillPattern.Solid),
Polygon(points = {{19, 56}, {12, 58}, {12, 54}, {19, 56}},
lineColor = {95, 95, 95}, fillColor = {95, 95, 95},
fillPattern = FillPattern.Solid), Polygon(points = {{-11, 56},
{-4, 58}, {-4, 54}, {-11, 56}}, lineColor = {95, 95, 95},
fillColor = {95, 95, 95}, fillPattern = FillPattern.Solid),
Text(extent = {{-35, 68}, {-6, 60}}, lineColor = {0, 0, 0},
textString = "rising"), Text(extent = {{16, 68}, {44, 60}},
lineColor = {0, 0, 0}, textString = "falling")),
Documentation(info = "<html>

```

<p>

The Real output  $y$  is a trapezoid signal:

</p>

<p>

```
<img  
src=\"modelica://Modelica/Resources/Images/Blocks/Sources/Trap  
ezoid.png\"  
alt=\"Trapezoid\">  
</p>  
</html>"));  
end flujo_Vapor;
```

## ANEXO 4

### CÓDIGO DEL MODELO DE LA CALDERA EN MODELICA

```
within Modelica;
model CALDERA_3
  //Declaracion de constantes
  parameter Real e1 = 2.685e-4;
  parameter Real tempo = 40;
  parameter Real qf = 60;
  //Declaracion de variables
  Real pprima(start = 0);
  Real qs(start = 0);
  Real hs(start = 0);
  Real hw(start = 0);
  Real hf(start = 0);
  Real hc(start = 0);
  Real Q(start = 0);
  Real temp(start = 0);
  Real Presion(start = 0);
equation
  Q = qs * hs - qf * hf;
  hc = hs - hw;
  pprima = e1 * (Q - qf * (hw - hf) - qs * hc);
  Presion = der(pprima);
  temp= ((31.763)*log(Presion))-35.97;
  hw=(temp*4.3455)-14.203;
  hs=-((1e-
5)*temp^3)+(0.001*(temp^2))+(1.7687*temp)+2502.8;
end CALDERA3;
```

## BIBLIOGRAFÍA

- [1] K.J. Aström and R. D. Bell, "Drum-boiler dynamics," *Automatica*, pp. 363-378, 2000.
- [2] Katsuhiko Ogata, *Ingeniería de Control Moderna*.: Pearson, 2010.
- [3] [Online]. <http://www.uv.es/olmos/Ecuaciones%20diferenciales.pdf>
- [4] Equipos y Laboratorio de Colombia. [Online]. [http://www.equiposylaboratorio.com/sitio/contenidos\\_mo.php?it=4875](http://www.equiposylaboratorio.com/sitio/contenidos_mo.php?it=4875)
- [5] Gas Natural Fenosa. [Online]. <http://www.empresaeficiente.com/es/catalogo-de-tecnologias/calderas#ancla>
- [6] Carlos J. Renedo. [Online]. <http://personales.unican.es/reneDOC/Trasp%20WEB/Trasp%20AA/008%20Calderas%20OK.pdf>
- [7] (2008) <http://science.howstuffworks.com/transport/engines-equipment/steam2.htm>.
- [8] (2015, mayo) VijSun. [Online]. <http://www.vijsunengineers.net/circulating-fluidized-bed-boilers.html>
- [9] Kurt C. Rolle, *Termodinámica*, 1st ed.: Prentice Hall, 2006.
- [10] K. J. Amströn and K. Eklund, "A simplified non-linear model of a drum boiler-turbine unit," *International Journal of Control*, no. 16, pp. 145-169, 1972.
- [11] M. Denn, "Process modeling," *Wiley*, 1987.
- [12] Matlab. Mathworks. [Online]. [www.mathworks.com](http://www.mathworks.com)
- [13] Scilab Enterprises. Scilab. [Online]. [www.Scilab.org](http://www.Scilab.org)
- [14] Ogata Katsuhiko, *Dinámica de sistemas*. México: Prentice-Hall Hispanoamericana, 1987.