

“Análisis de factibilidad y primera implementación de cálculo de límites de disparo deslizantes para el PSPR del reactor CAREM 25 utilizando tecnología FPGA”

***CARRERA: ESPECIALIZACIÓN EN REACTORES NUCLEARES
Y SU CICLO DE COMBUSTIBLE***

Alumno: Francisco Gabriel Rodríguez

Director: Claudio Verrastro

Noviembre 2015



UNSAM
UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

Índice

Resumen	3
1. Introducción	4
1.1 Objetivo	5
1.2 Antecedentes.....	5
1.3 Primer Sistema de Protección de Reactor (PSPR)	6
1.4 Sistemas de Parada de CAREM 25	6
1.5 Criterios de diseño.....	7
1.6 Arquitectura	9
1.7 Diversidad	10
2. Abreviaturas.....	12
3. Medición de variables	13
3.1 Calibración de sensores	13
3.2 Implementación de digitalización	16
4. Valores límites de disparo deslizantes	18
4.1 Recuento de señales y parámetros para el cálculo de valores límite deslizantes.....	19
4.2 Valor límite deslizante a partir de presión del primario	20
4.3 Valor límite deslizante de presión del primario a flujo neutrónico	29
4.4 Valor límite deslizante VLDD-FNP	34
4.5 Digitalización de flujo neutrónico y obtención de VLDD-FNP	34
4.6 Limitación de rango y tasa para VLDD-FNP	35
4.7 Calculo de caudal de generador de vapor a partir del flujo (VLDD-BCGV) ...	37
4.8 Calculo de Temperatura de salida de los GV's (TEN).....	41
5. Conclusiones	53
6. Referencias bibliográficas	54
7. Anexos.....	55
7.1 Instrucciones para el uso de funciones en MATLAB	55
7.2 Obtención de ATEN y BTEN.....	56
7.3 Esquema general de límites deslizantes.....	59
7.4 Código fuente	60

Resumen

El primer sistema de protección del reactor CAREM 25, requiere que los valores de comparación, para la demanda de los sistemas de parada, sean modificados automáticamente a medida que la planta cambia sus condiciones de operación.

Una característica importante del PSPR es que contará con tecnología digital basada en dispositivos de lógica configurable FPGA/CPLD. Estos dispositivos son actualmente utilizados en aplicaciones críticas de la industria nuclear y aeroespacial, ya que brindan flexibilidad de implementación, conservando máximos requerimientos de confiabilidad, siguiendo los lineamientos de las normas para su aplicación, en sistemas de seguridad para centrales nucleares.

El objetivo de este trabajo, es realizar un análisis de factibilidad de la implementación del procesamiento, para la generación de valores límite de disparo deslizantes, utilizando un módulo basado en tecnología FPGA, en el primer sistema de protección del reactor CAREM 25.

A partir de un documento de requerimientos funcionales y un modelo MATLAB Simulink realizado por el grupo de dinámica de planta del proyecto CAREM25, se realizó una modelización de la implementación del cálculo, de los valores límite de disparo deslizantes.

Se dividió el sistema en bloques funcionales, que pueden ser implementados por componentes de hardware simple configurables en un FPGA. Luego se armaron las cadenas de procesamiento para cada valores límite de disparo deslizante.

Una vez obtenido el modelo de la implementación real del sistema, se procedió a verificar el comportamiento del sistema, utilizando curvas de señales de entrada típicas del funcionamiento de la planta. Estas señales son:

1. Presión del circuito primario (RPV) y Flujo neutrónico en Calentamiento normal
2. Presión del circuito primario (RPV) y Flujo neutrónico en Enfriamiento normal
3. Presión del circuito primario (RPV) y Flujo neutrónico en Enfriamiento abrupto

De esta manera se verificó que el error, de la implementación del sistema usando los módulos basados en FPGA, es casi un orden de magnitud inferior al error introducido por otras etapas, como adquisición de la variable física por los sensores y acondicionamiento de la señal.

Además se detectaron distintas características del procesamiento, especificado en los requerimientos, que pueden llevar al sistema a actuar con mayor error de medición del esperado, disminuyendo la disponibilidad de planta. Se advierten estos problemas y se sugieren distintas soluciones a los grupos de instrumentación y procesos.

1. Introducción

A principios del año 2008 el sector de IyC del proyecto CAREM 25 contrata al sector de IyC de GAEN, para el desarrollo del primer sistema de protección del Reactor (PSPR), encargado de accionar los sistemas de parada del reactor [1].

Para la implementación del sistema, se decidió utilizar tecnología FPGA ya que provee una confiabilidad muy superior a los sistemas modernos basados en software. Además, se introdujo una nueva arquitectura, que utiliza el recurso de diversidad a nivel de módulos, capaz de reducir la probabilidad de falla en modo común, aumentando la confiabilidad del sistema.

En los sistemas de protección de reactor (SPR) tradicionales, los pedidos de accionamiento a los sistemas de seguridad, se producen cuando las variables físicas medidas sobrepasan ciertos límites configurados previamente.

En el caso del reactor CAREM25, estos límites dependen del estado operativo del reactor. El sistema de protección debe recalcular estos límites de, acuerdo a la evolución de las variables de seguridad tanto en el arranque, durante la operación como en el apagado del reactor, para modificar el valor de los límites de seguridad. Estos parámetros de disparo automáticamente modificados, se denominan "Valores límite de disparo deslizantes" (VLDD) [2].

Para especificar de forma completa los requerimientos, el grupo de dinámica de planta del proyecto CAREM25, realizó un modelo con las funciones del PSPR en su totalidad utilizando el paquete de software Simulink de MATLAB. Además acompaña este modelo, un documento con la explicación del funcionamiento del sistema.

Si bien, la especificación del sistema es completa con estos datos de entrada, el modelo en Simulink representa el comportamiento ideal del sistema, donde no se tienen en cuenta errores de adquisición, digitalización y procesamiento de las señales.

En este trabajo se analiza la factibilidad de realizar el procesamiento necesario, para generar los VLDD, en los módulos basados en FPGA compatibles con la arquitectura actual planteada para el PSPR.

En este trabajo se estudian los requerimientos funcionales del sistema hechos en simulink y se realizará una primera implementación del sistema simulando los componentes de hardware, como ser ADC, limitadores de tasa y rango, tablas de conversión de unidades. Se simula esa implementación en MATLAB y se calcula el error de los resultados obtenidos, comparando las salidas del modelo en simulink y el implementado.

Debido a la complejidad del sistema se presenta un diagrama general del procesamiento necesario para generar los VLDD en el punto 7.3.

1.1 Objetivo

El objetivo de este trabajo es realizar un análisis de factibilidad de la implementación, del módulo de procesamiento para la generación de VLDD utilizando un módulo, basado en tecnología FPGA.

Se pueden puntualizar las siguientes tareas necesarias para cumplir con este objetivo:

1. Examen detallado del documento de requerimientos funcionales del primer sistema de protección.
2. Elaboración de un modelo, que permita tal funcionalidad, dividiendo el sistema en componentes reutilizables simples de implementar.
3. Modelizar el proceso de digitalización de las señales analógicas que se utilizan para el cálculo de VLDD.
4. Modelizar los distintos componentes que se utilizan para el procesamiento de las señales en la generación de los VLDD.
5. Modelizar las cadenas completas de procesamiento utilizando los componentes desarrollados en el punto anterior.
6. Generar los VLDD correspondientes a distintas variables de entrada. Estas señales son:
 - a. Presión del primario y Flujo neutrónico en Calentamiento normal
 - b. Presión del primario y Flujo neutrónico en Enfriamiento normal
 - c. Presión del primario y Flujo neutrónico en Enfriamiento abrupto
7. Medición y caracterización del error entre los resultados del modelo generado y la simulación hecha en simulink del funcionamiento esperado.
8. Discusión y análisis de la implementación generada.

1.2 Antecedentes

Existen amplios antecedentes a nivel mundial del empleo de FPGA en sistemas de seguridad de reactores de potencia[3][4]. Se han elaborado normas para el desarrollo de equipos electrónicos basados en estos dispositivos [5][6][7].

Por otra parte el grupo de lyC de CNEA instaló desde el año 2003 sistemas de protección en reactores experimentales con dispositivos configurables. Entre estos sistemas están:

- Lógica de SCRAM del reactor RA1[8].
- Impulsímetro digital para canal de arranque[9].

También se está instalando, en CNAII y diseñando para RA10, sistemas similares al propuesto para CAREM25 [10][11].

En cuanto a antecedentes externos al Departamento de lyC se puede mencionar a la empresa ucraniana Radiy, que desarrolla SPR con tecnología FPGA y actualizó la mayor parte de los SPRs de las centrales nucleares ucranianas[12].

1.3 Primer Sistema de Protección de Reactor (PSPR)

El PSPR es el sistema que tiene como objetivo principal asegurar en todo momento el cumplimiento de las funciones de seguridad, en una central nuclear[1]. Las funciones de seguridad principales son:

1. Control de la reactividad
2. Control de inventario del circuito primario y el adecuado transporte de calor hacia la fuente fría final
3. Adecuado confinamiento para evitar la liberación de material radioactivo

Para ello, el PSPR dispone de un conjunto de componentes, que inicia de manera automática los sistemas frontales de seguridad, de modo de garantizar que no se excedan los límites especificados, de diseño de la central. El PSPR siempre tiene prioridad sobre las acciones del sistema de control de la planta.

En el marco del criterio de defensa en profundidad, el reactor CAREM 25 posee un segundo sistema de protección, cuya fabricación está a cargo de la empresa INVAP. Las lógicas de actuación final del PSPR, poseen una entrada para las señales de disparo del SSPR, de manera que este pueda actuar de forma independiente el primer sistema de parada del reactor (caída de barras). Además, este sistema posee un segundo sistema de parada, compuesto por dos tanques redundantes para la inyección de ácido bórico en el recipiente del presión del reactor.

Entre los sub-sistemas del PSPR, se encuentran los Sistemas de Instrumentación, encargados de monitorear un conjunto de variables, denominadas variables de seguridad y de iniciar acciones de protección cuando dichas variables excedan los límites preestablecidos. Dichas variables corresponden a magnitudes termohidráulicas, nucleares y radiológicas.

1.4 Sistemas de Parada de CAREM 25

Para cumplir con la función de seguridad de control de reactividad, el PSPR puede disparar cinco acciones de protección[1]:

1. Primer sistema de extinción (PSE)
2. Sistema de inyección de seguridad (SIS)
3. Sistema de Seguridad de extracción de calor residual (SSECR)
4. Aislamiento de la contención (AC)
5. Aislación del generador de vapor (AGV)

Además de la función de los disparos automáticos de los mencionados sistemas, el PSPR cumple las siguientes funciones:

1. Vigila la evolución de las variables y detecta su apartamiento más allá de los valores límite establecido.
2. Implementa la lógica de disparo, que inicia las acciones de protección necesarias, para llevar a la planta a un estado seguro.

3. Indica los valores de las variables de seguridad medidas y su estado interno en las salas de control.
4. Provee al operador de los disparos manuales, de las acciones de protección que acciona.
5. Supervisa el propio funcionamiento seguro del PSPR.

1.5 Criterios de diseño

Los criterios tomados en cuenta en el diseño del PSPR son los siguientes[1]:

1.5.1 Simplicidad en el diseño

En cada etapa del diseño se elige una solución simple, para cumplir con los requerimientos. Se minimizará la cantidad de interconexiones entre módulos y la cantidad de componentes. En el caso de la implementación con FPGA, se utilizarán componentes y máquinas de estado simples, evitando el uso de dispositivos con software.

1.5.2 Tecnología probada

Los componentes principales están diseñados en base a dispositivos configurables tipo FPGA que son utilizados en instrumentación y control de reactores alrededor del mundo.

1.5.3 Redundancia

Para mantener una alta disponibilidad y cumplir con la el criterio de falla simple, el PSPR tiene todos sus componentes redundados en 4 trenes.

1.5.4 Falla simple

El sistema está diseñado para que la falla de un componente, no afecte el funcionamiento del sistema.

1.5.5 Independencia

Cada redundancia es capaz de cumplir su función, sin ser afectada por fallas en las otras redundancias. Por ejemplo, la conexión entre redundancia es a través de fibras ópticas, que protegen el sistema de interferencias galvánicas de forma intrínseca.

1.5.6 Diversidad

El diseño propuesto para la implementación del PSPR incorpora, más allá de los conceptos tradicionales de redundancia, falla sin riesgo, auto verificación, etc; el concepto de diversidad, consistente en la ejecución de las funciones de seguridad por dos unidades distintas en paralelo (diversidad #1 y diversidad #2), de distinto fabricante y configuradas por distintos equipos de trabajo.

Esta característica, se incorpora para cumplir con los requerimientos de auto verificación y mitigar las fallas en modo común, en la configuración de los dispositivos electrónicos complejos.

Cada par de módulos diversos homólogos funcionan en paralelo en el mismo tren y realizan exactamente la misma función, a pesar de que cada uno de ellos, tenga un funcionamiento interno diferente.

Los elementos diversificados en los trenes de instrumentación, son aquellos que tienen dispositivos configurables y cumplen funciones de seguridad, por ejemplo:

- Módulos de adquisición y digitalización
- Módulos de procesamiento para valores límite de disparo deslizantes
- Módulos de comparación
- Módulos de votación

Una característica importante del concepto de diversidad, es que no existe ningún tipo de prioridad entre módulos diversos homólogos. Ambos componentes pueden realizar la función de seguridad. Esto difiere de la arquitectura tradicional encontrada en otros sistemas de seguridad, en que un módulo de alta confiabilidad, se encuentra realizando una función determinada y, por otro lado, otro módulo de menor jerarquía realiza tareas de supervisión. Allí, el módulo de supervisión no suplente la función de seguridad y se limita a producir señales de alarma, cuando detecta algún tipo de error.

1.5.7 Auto verificación

Cada uno de los trenes ejecuta las funciones de seguridad, por dos unidades distintas en paralelo (diversidad #1 y diversidad #2), que realizan una verificación cruzada.

La Figura 2 y Figura 3 muestra que los módulos diversos homólogos están conectados entre sí. La función de esta conexión es poder realizar una verificación de sus resultados, generando una señal de falla, cuando se detectan discrepancias.

1.5.8 Falla sin riesgo

Para cumplir con el requerimiento de falla sin riesgo, cualquier estado de falla previsto, lleva al módulo o componente en que ésta se produce a un estado seguro, por ejemplo el disparo de la señal de protección.

1.6 Arquitectura

En la Figura 1 se presenta el esquema general del PSPR. Está compuesto por cuatro *trenes de medición, votación y disparo*, cada uno con sensores, comparadores, votadores y drivers independientes que miden las mismas *variables de seguridad*, conformando así un sistema cuádruple redundado.

Las *señales de seguridad* se digitalizan y se comparan en las *lógicas de comparación*, contra valores de disparo, que definen si la señal de seguridad está en el rango especificado. El resultado de esta comparación, se denomina *señal de resultado de comparación*.

Las *señales de resultado de comparación* se intercambian entre todos los trenes, de forma que cada tren, recibe las señales de resultado de comparación propias y las de los otros trenes.

Cada módulo de *lógica de votación y disparo*, recibe información sobre las *señales de resultado de comparación*, generadas por la lógica de comparación de cada tren (del propio y de los tres trenes restantes). Las *señales de resultado de comparación*, se someten a una votación 2oo4. Las señales de protección están redundadas por cada tren, se someten a una votación en la última instancia del PSPR, que es la *lógica de votación final*.

Las lógicas de votación final activan el actuador de la acción de protección, llevando al reactor a un estado seguro.

El PSPR provee un sistema de visualización y comando disponible en sala de control principal y de emergencia. Además informa al sistema de control del estado del sistema y las variables medidas. Estas interfaces pueden observarse en la parte inferior de la Figura 1.

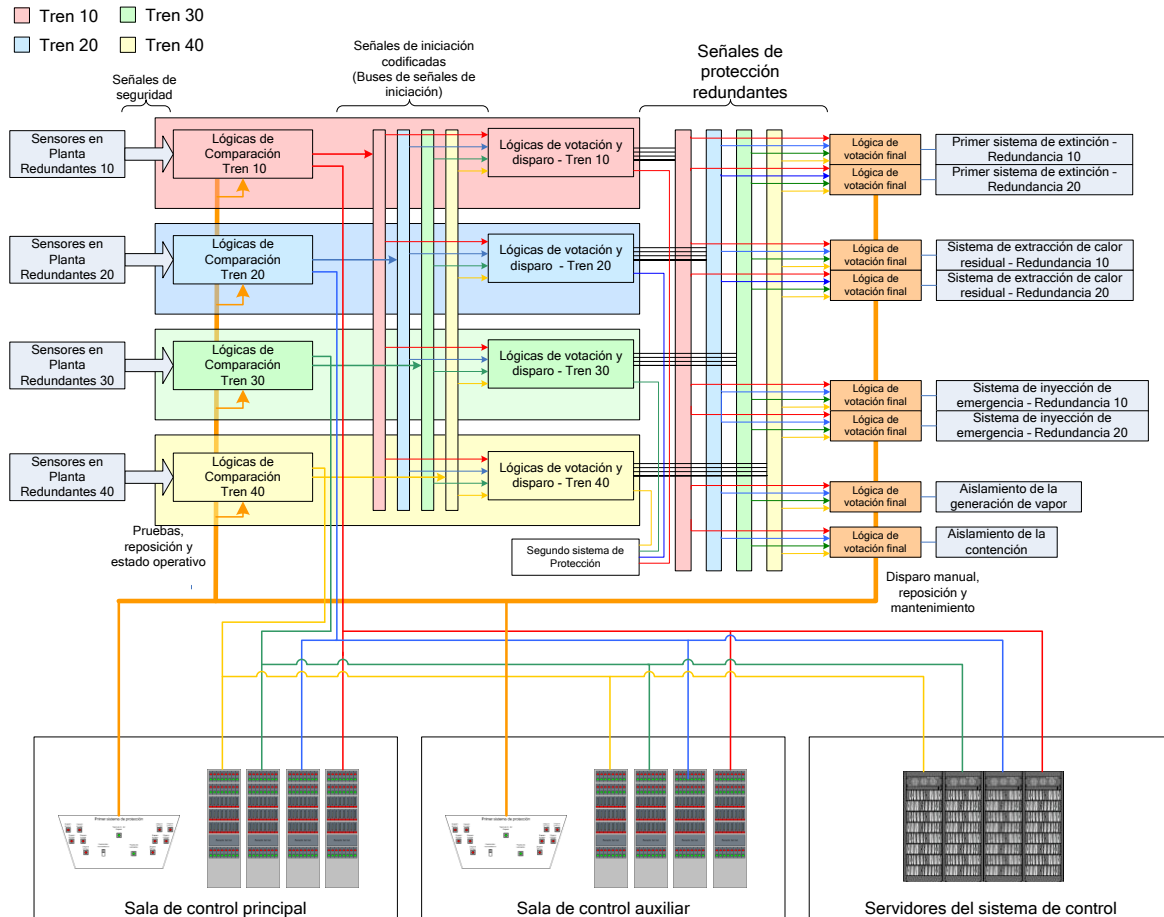


Figura 1 - Arquitectura del PSPR, esquema general.

1.7 Diversidad

Una característica de este diseño es la introducción del concepto de diversidad en cada uno de los trenes de instrumentación. Esto implica que varios de los módulos electrónicos se encuentran implementados dos veces, utilizando diferentes tecnologías y distintos equipos de desarrollo. De esta forma, cada par de módulos diversos homólogos funcionan en paralelo en el mismo tren y realizan exactamente la misma función, a pesar de que cada uno de ellos tenga una implementación interna diferente.

En la Figura 2 y Figura 3 se pueden distinguir los elementos que se encuentran diversificados en los trenes de instrumentación, éstos son:

- Módulo de Comparación y Disparo
- Módulo de Votación
- Módulos de procesamiento para valores límite de disparo deslizantes

Los módulos diversos homólogos, están conectados entre sí, con el objetivo de poder realizar un chequeo cruzado de sus resultados, generando una señal de falla cuando se detectan discrepancias

La ausencia de prioridad entre los módulos diversos, hace que en algún punto dicha diversidad se necesite "resolver", de modo de generar una única señal de disparo de tren. El encargado de resolver la diversidad es el módulo de drivers, que genera un único disparo mediante una lógica 1 de 2 entre los disparos de las dos unidades de votación.

En el caso del módulo de procesamiento de valores de disparo deslizantes, se realiza el cálculo de forma paralela por dos módulos diversos, pero la verificación se realiza a nivel del comparador. Comprobando que la variable dispare, con una mínima diferencia de tiempo, entre diversidades.

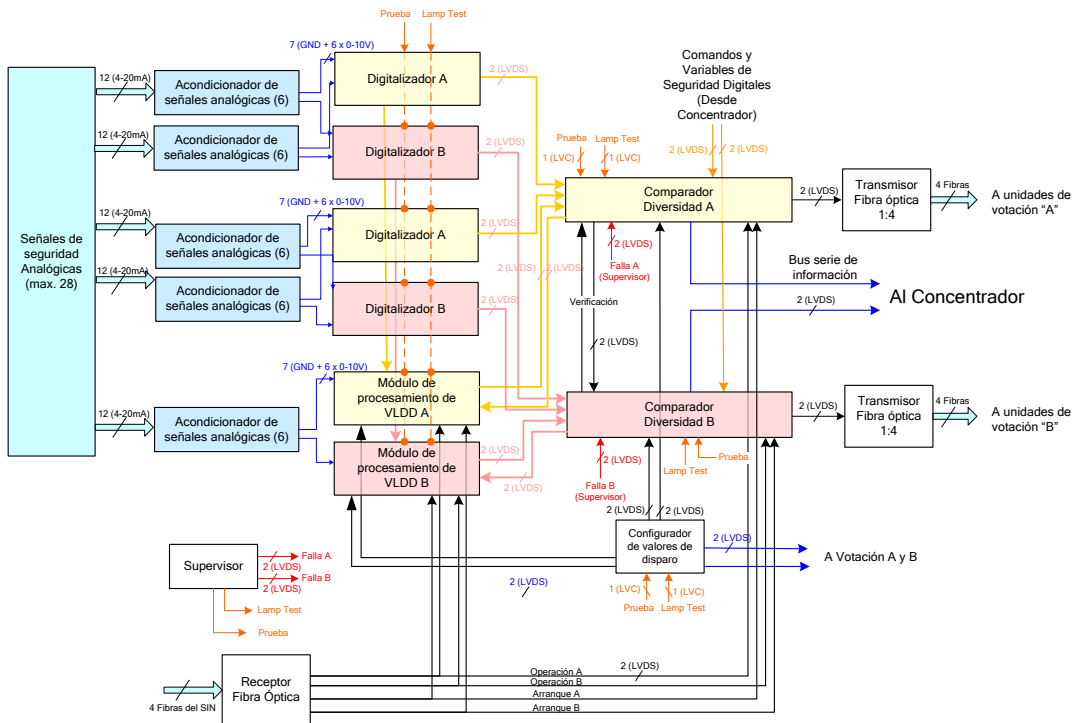


Figura 2 - Arquitectura de la etapa de adquisición y comparación del PSPR

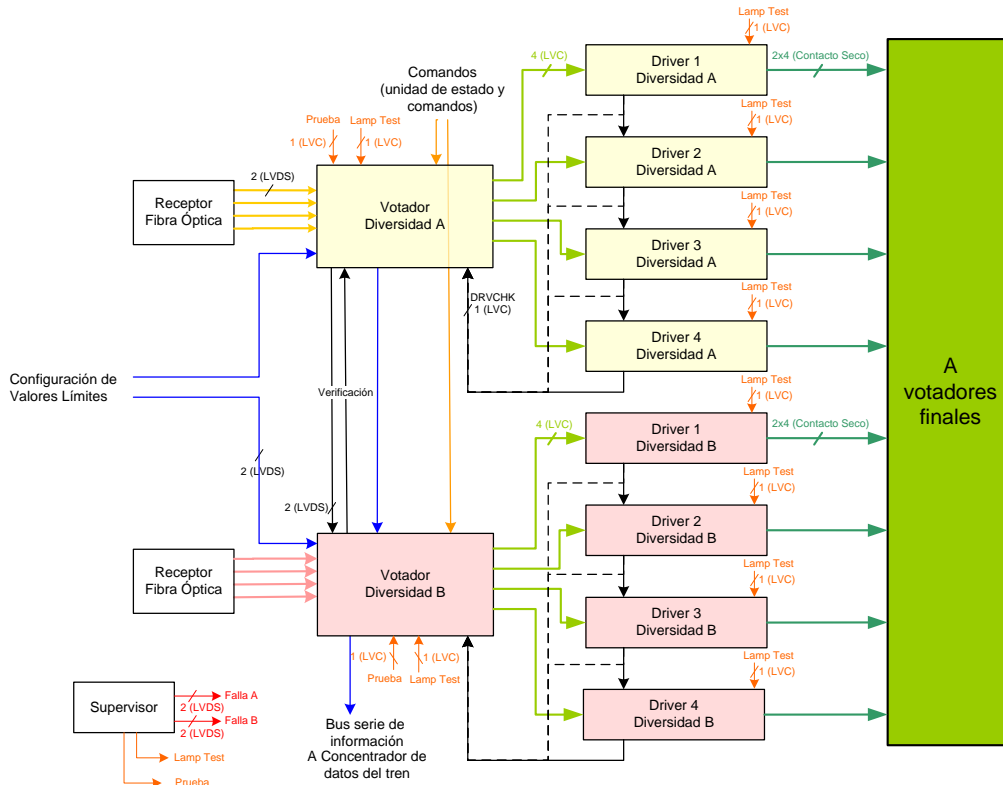


Figura 3 – Arquitectura de la etapa de lógica de disparo y votación del PSPR

2. Abreviaturas

ACGV1: señal digital, alto CGV.

ADC: Conversor analógico – digital

AFNP1: señal digital, alto FNP.

APP0: señal digital, muy alta PP.

APP1: señal digital, alta PP.

ATEN1: señal digital, alta TEN.

BCGV1: señal digital, bajo CGV.

BPP0: señal digital, muy baja PP.

BPP1: señal digital, baja PP.

BTEN1: señal digital, baja TEN.

CGV: señal analógica, caudal de alimentación a los GV's.

DEL: Delta, diferencia positiva o negativa de la función de referencia.

DTGV: señal analógica, salto térmico en el GV del lado primario: TSAT-TEN.

FNPF: señal analógica, flujo neutrónico, rango de potencia (Campbell).

FPGA: Field programmable gate array. Circuito integrado digital con lógica configurable.

GAEN: Gerencia de área de energía nuclear de la comisión nacional de energía atómica.

GV: Generador de vapor

IyC: Subgerencia de instrumentación y control. Proveedor del servicio.

PP: señal analógica, presión en el recipiente de presión / circuito primario.

PSPR: Primer sistema de protección del reactor.

RPV: Recipiente de presión del reactor

TEN: señal analógica, temperatura a la salida de los GV's, lado primario (dentro del RPR).

TSAT: señal analógica, temperatura de saturación a la presión PP.

VLDD: Valor Límite de disparo deslizante

3. Medición de variables

3.1 Calibración de sensores

En el documento de requerimientos funcionales del PSPR, los cálculos de límites deslizantes, se realizan en unidades de ingeniería. Es decir, las señales que interconectan cada entrada, salida y módulos intermedios, son variables y funciones de punto flotante que representan el valor de una magnitud física, por ejemplo, MPa, °C, Porcentaje del flujo a plena potencia.

Para la implementación se utilizan sensores, que convierten la variable física en un valor de corriente, típicamente 4-20mA. Esta conversión puede darse de las siguientes formas:

3.1.1 Función lineal con valores calibrados

En un sensor, la función de conversión de una magnitud física a una señal eléctrica, sigue una regla predeterminada, un ejemplo típicos son los sensores, cuya salida es una corriente en el rango de 4 a 20mA. En el caso de un sensor de temperatura la corriente sigue la siguiente función:

$$I = 4mA + 0,1 \frac{mA}{K} (Temperatura K - 273K)$$

La función de conversión a temperatura es fija durante la operación del sensor. Esta calibración, debe ser comprobada durante los mantenimientos periódicos. Esto simplifica su digitalización ya que la cantidad de cuentas equivalente a grados es fija.

Si se reemplaza el sensor, el nuevo sensor esta calibrado de la misma forma, por lo que no necesita configurarse nada en el módulo.

Otra opción para lograr este comportamiento es usar sensores lineales con un transductor de conversión configurable en ganancia y corrimiento, ya sea con un amplificador analógico en la conversión a corriente o con HART si se trata de un sensor inteligente.

La configuración de valores límite, puede ser realizada en unidades de ingeniería, ya que se utiliza la misma función para convertir las unidades. No es necesaria la reconfiguración para cada sensor. Lo mismo aplica para la visualización y comunicación con el sistema de control.

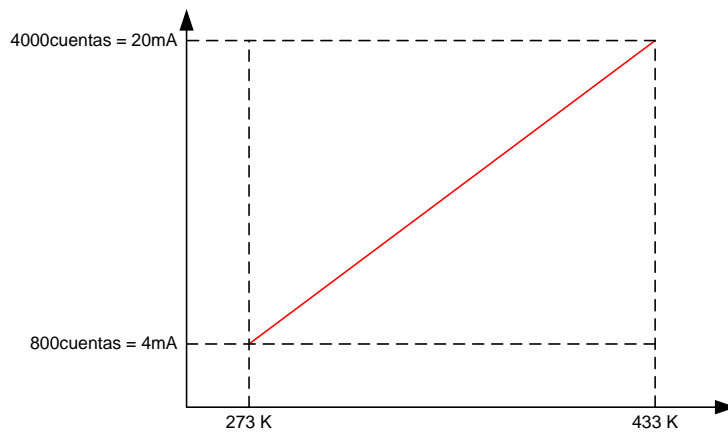


Figura 4 – Función de transferencia de un sensor lineal con valores calibrados

3.1.2 Función lineal sin valores calibrados

Este es el caso de un sensor, de respuesta lineal pero cuyo rango varía de sensor a sensor. Es decir, si la función transferencia es:

$$I = 4mA + A\left[\frac{mA}{K}\right] (T \text{ } ^\circ C - T_0 \text{ } ^\circ C)$$

A y T_0 son variables.

Si se desea realizar la configuración de valores límite en unidades de ingeniería, cada vez que se instala un sensor, se deben fijar las constantes de calibración en el configurador de valores límites. Si los valores Límites se ingresan en porcentaje, cuentas o valores de corriente, este paso no es necesario, ya que la conversión puede ser realizada por el operador o técnico.

Para realizar la visualización en unidades de ingeniería, estas constantes de calibración, deben ser comunicadas a las unidades de visualización y al sistema de

control para una correcta visualización, esto aumenta innecesariamente las tramas de comunicación, entre las distintas unidades del sistema.

Si los valores Límites se ingresan en porcentaje, cuentas, valores de corriente, o en un formato estándar preestablecido, este paso no es necesario, simplificando la operación del sistema.

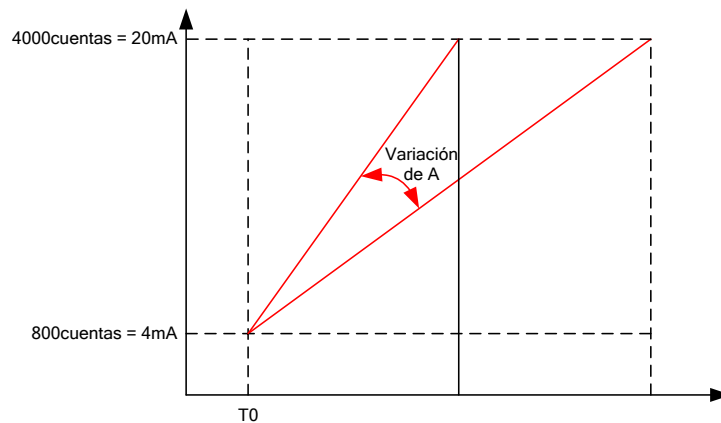


Figura 5 - Función de transferencia de un sensor lineal con valores no calibrados

3.1.3 Función no lineal sin valores calibrados

En este caso la conversión de la variable física a corriente, está dada por una función arbitraria, que se obtiene del ensayo del sensor en particular.

Si se desea realizar la configuración de valores límite en unidades de ingeniería, se debe ingresar la función de calibración en el sistema. Esto significaría, en el caso de un digitalizador de 12bits, ingresar una tabla de 4096 valores.

La configuración de valores límite para estos casos, se realiza haciendo la conversión con la curva de transferencia de manera offline. Ya sea por software o manualmente.

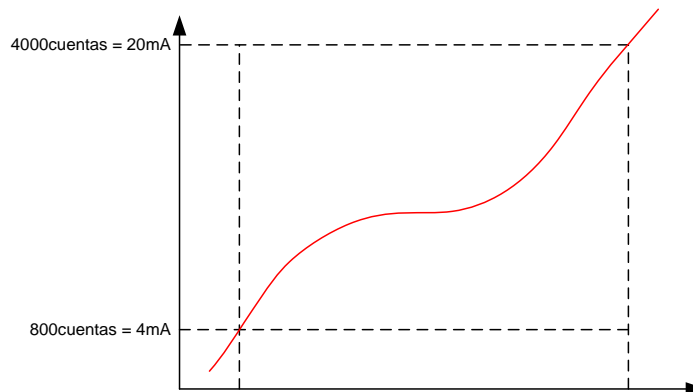


Figura 6 - Función de transferencia de un sensor no lineal

Se prefiere sensores de respuesta lineal, que puedan ser calibrados con offset y ganancia para que las funciones requeridas sean implementables y la operación del sistema sea simple.

3.2 Implementación de digitalización

3.2.1 Precisión

La etapa de entrada de señales analógicas del sistema está diseñada para tener una precisión de aceptación del 1% en total.

Las placas de conversión de corriente a tensión (4-20mA a 2-10V) tienen una precisión del 0,5%. Es decir, que se pueden distinguir cambios de 100uA. La electrónica analógica de los módulos digitalizadores y el conversor Analógico a Digital (ADC) tienen una precisión de 0,5%.

El módulo digitalizador propuesto por IyC es de 12bits. Esto significa que la cantidad de bits del ADC, no es el factor limitante, en cuanto a la precisión del sistema.

Se pueden fabricar módulos con más precisión, usando componentes de tolerancia menor y calibrando cada módulo individualmente. La utilización de este recurso solo debería hacerse si es necesario. Además de estos errores en la medición, hay que tener en cuenta el error propio del instrumento o sensor utilizado.

3.2.2 Cantidad de bits

Los conversores analógico digital, utilizados por los módulos que desarrolla IyC son de 12bits. Se considera un rango útil de 800cuentas a 4mA a 4000 cuentas a 20mA.

Es decir que:

$$\text{Presición} \frac{\mu A}{\text{cuenta}} = \frac{20mA - 4mA}{4000\text{cuentas} - 800\text{cuentas}} = 5 \frac{\mu A}{\text{cuenta}}$$

La calibración de 4mA a 800 cuentas y 20mA a 4000 cuentas se realiza en la etapa de digitalización para cada módulo.

3.2.3 La pérdida de precisión por la mala utilización del rango útil de la medición

Un problema que puede surgir en el proceso de digitalización, es la pérdida de precisión por la mala utilización del rango útil de la medición.

Por ejemplo, se tiene un sensor de 0 a 1000°C. Está calibrado para entregar 4mA a 0°C y 20mA a 1000°C. Sin embargo, la variable física medida cambia de 20°C a 200°C. Considerando que el sistema puede medir con una precisión del 1%, se podría considerar erróneamente que los valores Límites se pueden ingresar con una precisión de 1,8°C.

$$\text{Precisión en } ^\circ\text{C} = 200^\circ - 20^\circ \cdot 0,01 = 1,8^\circ\text{C}$$

Sin embargo, esa precisión se obtiene sobre todo el rango, es decir:

$$\text{Precisión en } ^\circ\text{C} = 1000^\circ - 0^\circ \cdot 0,01 = 10^\circ\text{C}$$

Es decir que la precisión para elegir valores de 20 a 200 se reduce al 5,55%:

$$\text{Precisión en } \% = 100 \cdot \frac{10}{200 - 20} = 5,55 \%$$

Entonces se deberá tener en cuenta que el rango dinámico del sensor coincida con el rango máximo de excursión de la variable a medir.

4. Valores límites de disparo deslizantes

Los VLDD son aquellos valores de comparación, para un comparador por alto o por bajo, que son función de una o varias señales de entrada procesadas.

Para incorporar esta funcionalidad en la arquitectura tradicional de un sistema de protección, se coloca un módulo de procesamiento entre la señal de entrada y el comparador. Este módulo genera la señal de VLDD y opcionalmente, puede transmitir la entrada, sin procesar para ser utilizada en otras comparaciones.

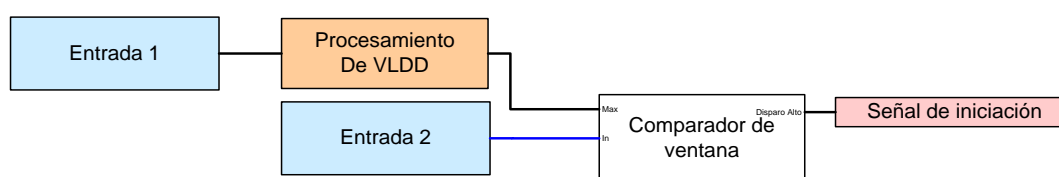


Figura 7 – Diagrama en bloques de la arquitectura elegida para el procesamiento de valores límites deslizantes

Para la implementación en hardware se utilizará tecnología basada en FPGA, con máquinas de estado simples, para el control del ADC y la transferencia de las señales digitales, en una trama serie para el comparador.

En la arquitectura final, el módulo de procesamiento puede generar varios VLDD recibiendo varias señales de entrada simultáneamente. La cantidad de valores calculados depende de la cantidad de señales que puede digitalizar el ADC y la cantidad de celdas que admita el FPGA que se utilizarán en la implementación final.

En la Figura 8 se puede observar una posible implementación del módulo de procesamiento. Las señales entran de forma analógica ya convertidas en señales de tensión 0-10V. Esta conversión se realiza por un módulo de adquisición de señales, que alimenta los sensores y convierte la señal 4-20mA a 0-10V.

Una máquina de estados simple maneja el ADC, para que entregue a través de un protocolo las señales digitalizadas. La implementación del procesamiento depende de cada VLDD generado y se detalla en las siguientes secciones. Finalmente se transmite en una trama serie, todos los datos generados.

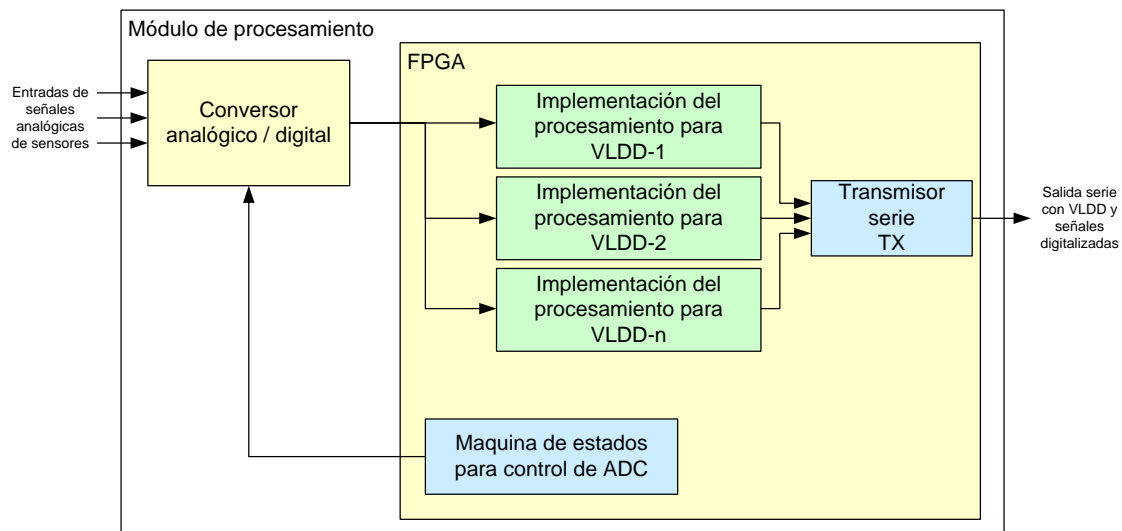


Figura 8 – Diagrama en bloques de implementación final del módulo de procesamiento

4.1 Recuento de señales y parámetros para el cálculo de valores límite deslizantes

Tabla 1– Variables de entrada que generan límites deslizantes

Variables físicas de entrada		Descripción
1	CGV	Caudal del generador de vapor
2	TEN	Temperatura de salida del generador de vapor. Lado primario
3	PP	Presión del primario
4	FNP	Flujo en potencia (Campbell)

Tabla 2 – Valores límite deslizantes generados

Valor límite deslizante	Descripción	
1	AFNP1	Límite por alto en el flujo neutrónico en potencia
2	BCGV1	Límite por bajo en el caudal de alimentación a los generadores de vapor
3	ATEN1 y BTEN1	Límite por alto y bajo en la temperatura de salida de los generadores de vapor del lado primario
4	BPP0 y BPP1	Límites por bajo en la presión del primario

4.2 Valor límite deslizante a partir de presión del primario

En los requerimientos funcionales se genera el VLDD a partir de la presión del primario. Este VLDD se utiliza para:

- Comparar presión del primario (VLDD-BPP0 y VLDD-BPP1)
- Generar el VLDD para flujo en potencia (VLDD-AFNP)
- Generar el VLDD de presión para el posterior cálculo de temperatura de entrada al núcleo (VLDD-PP)

Los valores utilizados en cada componente están en MPa.

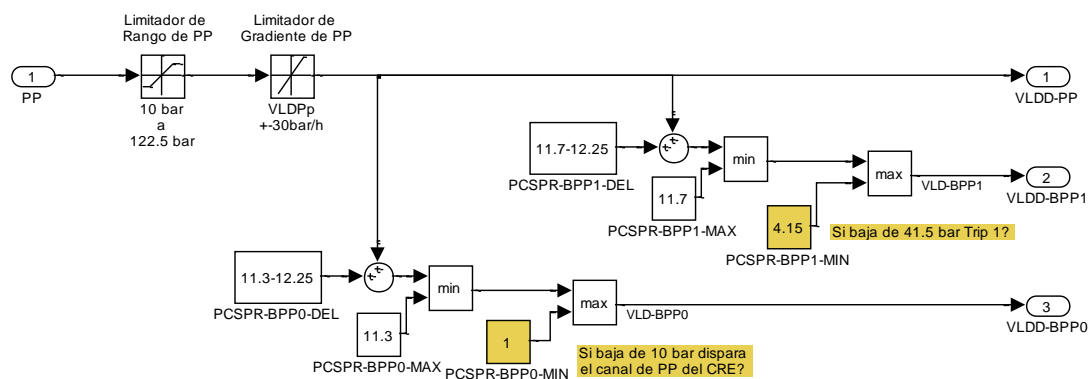


Figura 9 - Especificación de los Límites deslizantes generados a partir de PP según requerimientos en Simulink

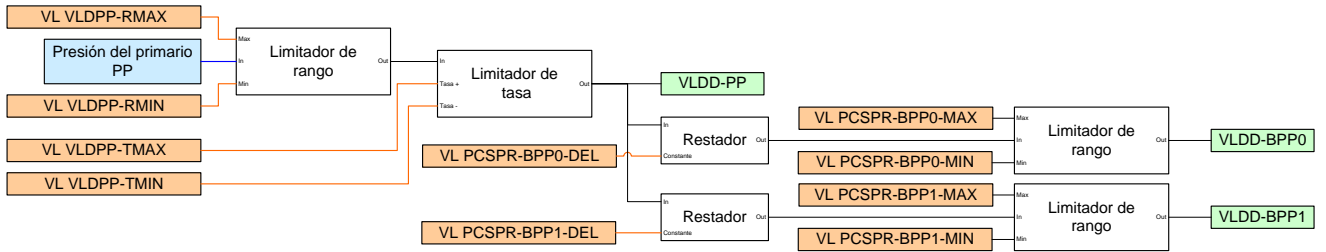


Figura 10 -Diagrama en bloques de la generación de límites deslizantes generados a partir de la PP

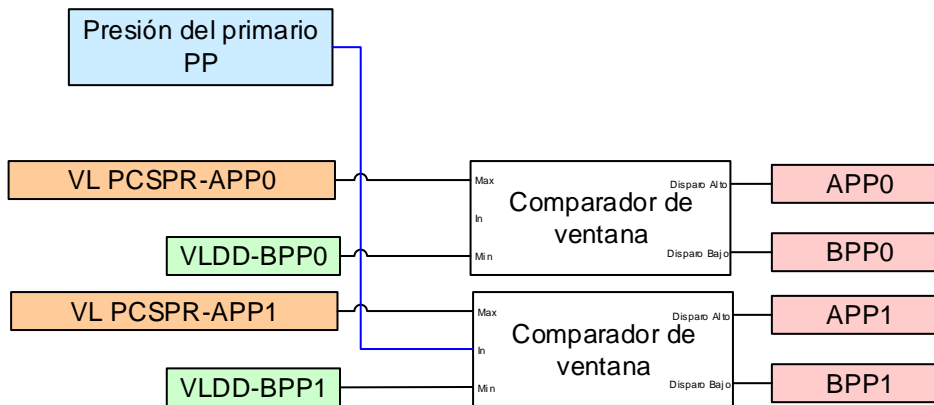


Figura 11 - Diagrama en bloques de los comparadores de PP

4.2.1 Comparación

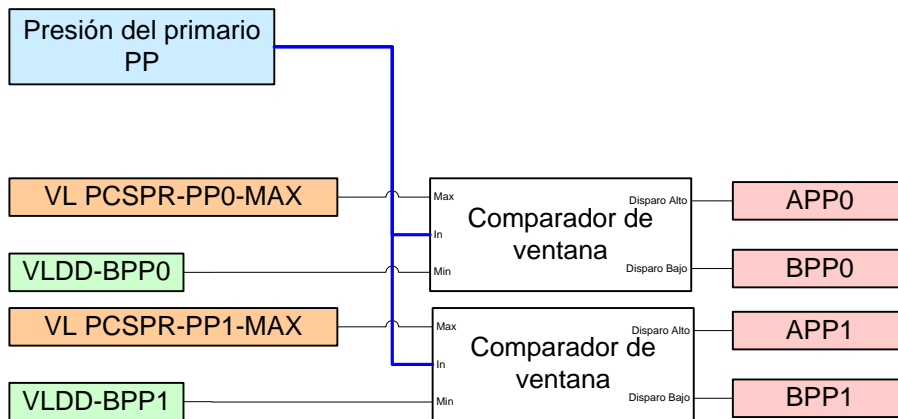


Figura 12 – Uso de VLDD-BPP0 y VLDD-BPP1 en el módulo de comparación

En la Figura 12 se puede ver que la variable de seguridad de presión de primario se compara contra dos máximos fijos como valores límite y dos mínimos que son los límites deslizantes VLDD-BPP0 y VLDD-BPP1.

4.2.2 Adquisición de PP

El limitador de rango está configurado de 10MPa a 12,25MPa. Se puede asumir que el sensor está calibrado de 0MPa = 4mA 13MPa = 20mA

Tabla 3 – Parámetros de adquisición de la Presión del primario

Variable física (MPa)	Corriente del sensor	Cuentas
0	4mA	0
13	20mA	3200

Si bien el ADC es de 12 bits, se debe tener en cuenta el cero vivo de la señal. Entonces, si bien la cantidad de cuentas en 12 bits es de 4096, se reservan 800 cuentas para la señal de cero vivo, es decir, de 0 a 4mA. Luego se calibra la digitalización para que 4 a 20mA corresponda a 800 a 4000. Al valor resultante se le restan las 800 cuentas del cero vivo, para mantener el cero medido en cero cuentas y poder escalar la señal simplemente multiplicando.

4.2.3 Limitador de rango

Se pueden configurar 2 valores, un máximo y un mínimo.

Tabla 4 – Parámetros de configuración del limitador de rango para PP

Variable física (MPa)	Cuentas
1	246
12,25	3015

Para implementar la limitación de rango se utilizará una lógica de saturación por alto y por bajo.

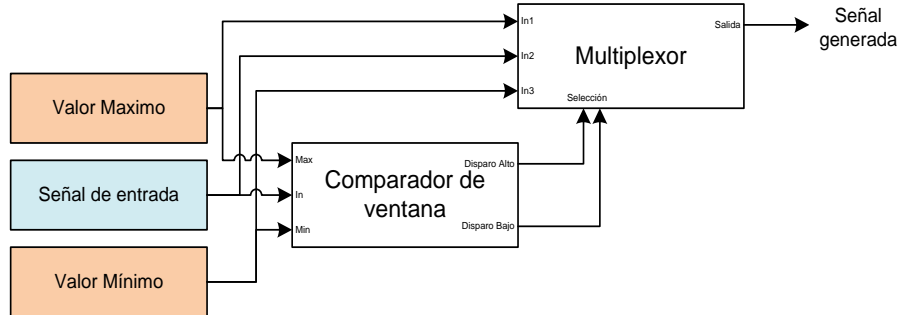


Figura 13 – Diagrama en bloques de la implementación del limitador de rango

4.2.4 Limitador de tasa

Se asume que este limitador trabaja en MPa/hora. Así, solo se configura un valor en MPa/hora.

El valor de 3MPa / hora se traduce a 738 cuentas/hora

$$\frac{\text{cuentas}}{\text{hora}} = \frac{4000\text{cuentas} - 800\text{cuentas}}{13\text{Mpa} - 0\text{Mpa}} \cdot \frac{\text{MPa}}{\text{hora}}$$

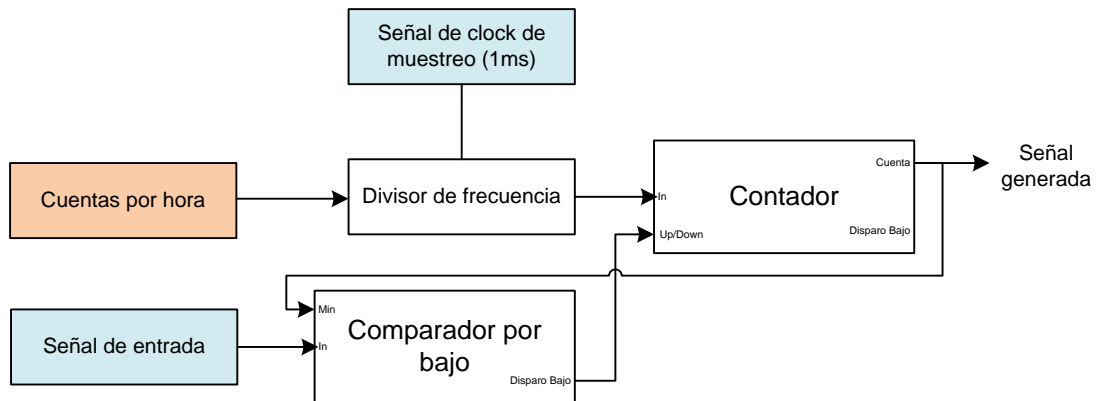


Figura 14 – Diagrama en bloques de implementación de limitador de tasa

Para la implementación del limitador de tasa, se utiliza un contador bidireccional. Los pulsos contados se generan con un divisor de frecuencia, que genera una señal con una frecuencia igual a la máxima cantidad de cuentas por hora, que puede crecer o decrecer la señal de salida. La dirección del contador, arriba/abajo, se controla por un comparador, que compara la señal de entrada con la de salida, de manera que la de salida siempre siga a la entrada.

Esta implementación no considera pendientes distintas para subir o bajar. Para ello se utilizaría un multiplexor luego del divisor de frecuencia.

4.2.5 Valores para VLDD-BPP0 y VLDD-BPP1

Son valores que se restan de la señal generada para establecer un margen de comparación.

Tabla 5 – Parámetros de configuración para la generación de BPP0 y BPP1 (restadores y limitadores de rango)

Variable	MPa	Cuentas
BPP0-DEL (delta)	-0,95	234
Max para BPP0	11,3	2781
Min para BPP0	1	246
BPP1-DEL (delta)	-0,55	135
Max para BPP1	11,7	2880
Min para BPP1	4,15	1022

Sumando todos los parámetros para el cálculo del VLDD-PP, VLDD-BPP0 y VLDD-BPP1 se necesitarían 10 valores de configuración, considerando tasas para subir y bajar distintas

4.2.6 Resultados para el cálculo de VLDD-PP, VLDD-BPP0 y VLDD-BPP1

En la Figura 15 y en la Figura 16 se puede observar como en el caso de enfriamiento normal el valor de PP queda por sobre los límites de presión baja BPP. Sin embargo en el caso de enfriamiento abrupto el valor de PP rápidamente cae por debajo de la señal limitada en tasa, estando por debajo de los límites generados.

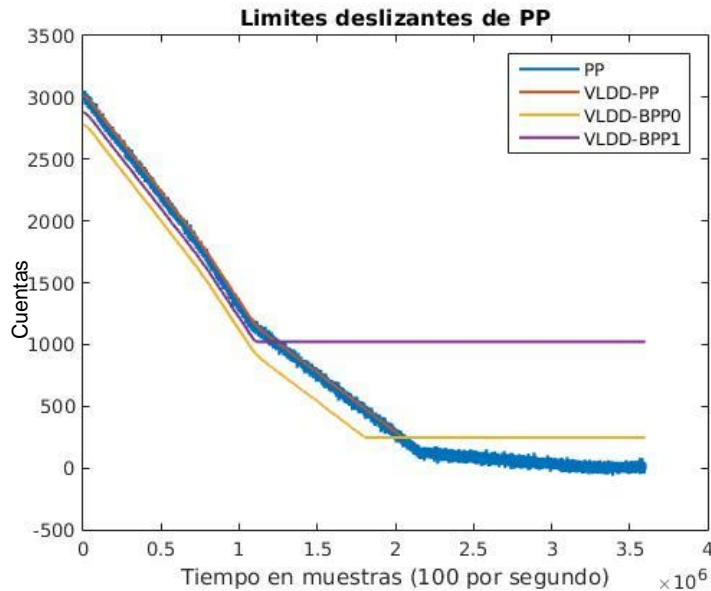


Figura 15 - Límites deslizantes de PP para enfriamiento normal

En la Figura 15 se puede observar que al variar PP tan lentamente VLDD-PP es una copia casi idéntica (sin el ruido) y no se la puede diferenciar.

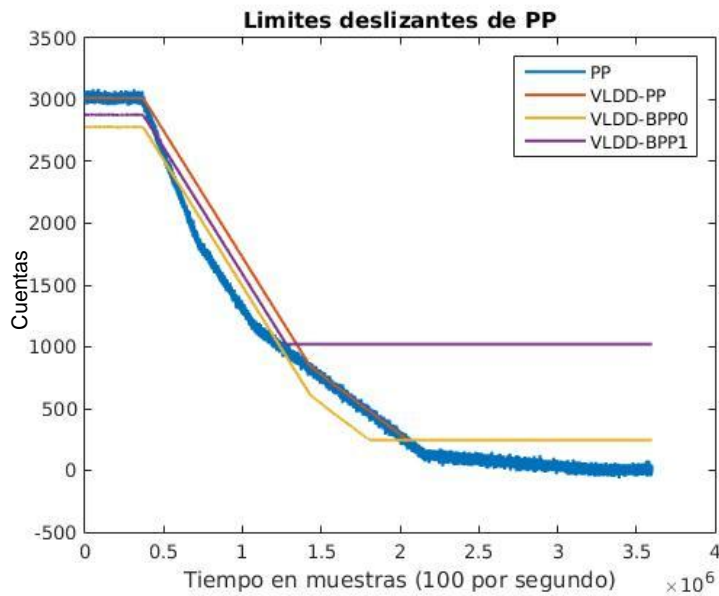


Figura 16 – Límites deslizantes de PP para enfriamiento abrupto

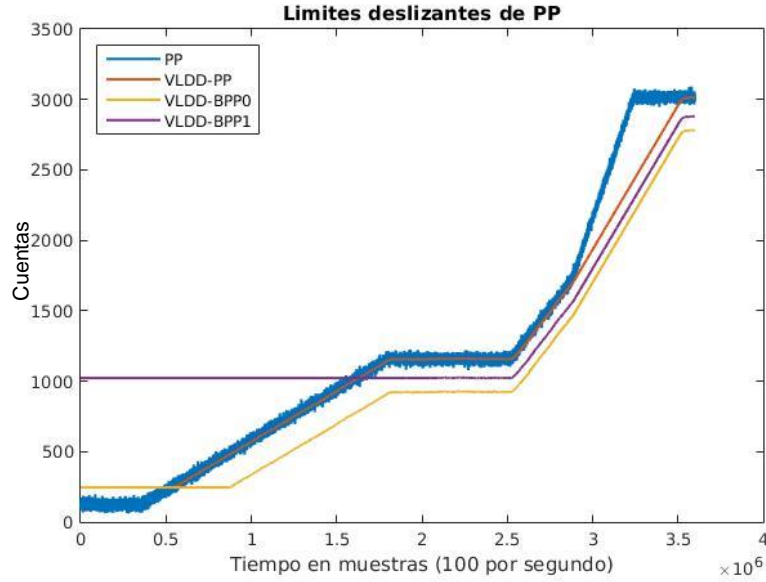


Figura 17 - Límites deslizantes de PP para calentamiento normal

4.2.7 Comparación y cálculo de error para enfriamiento abrupto

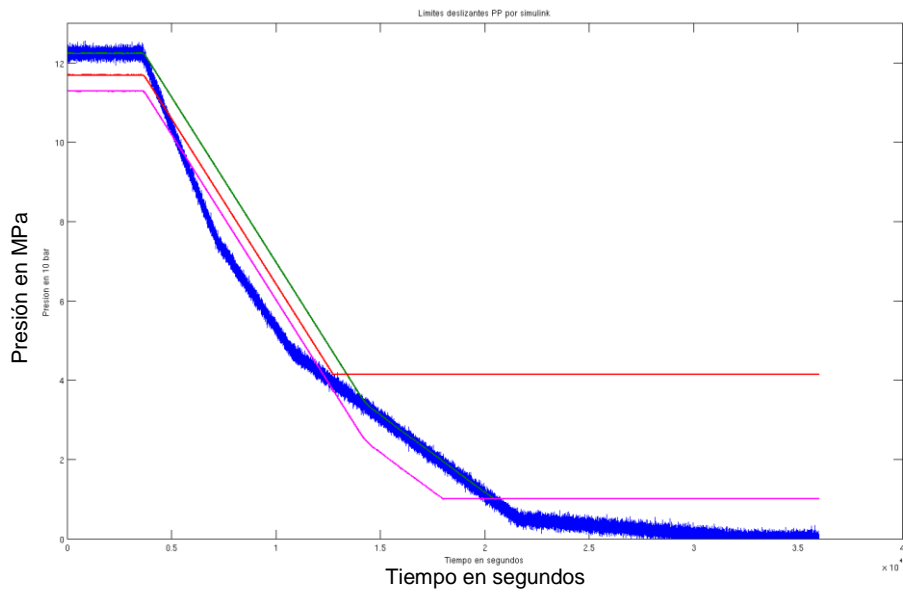


Figura 18 - Límites deslizantes de PP calculados con simulink

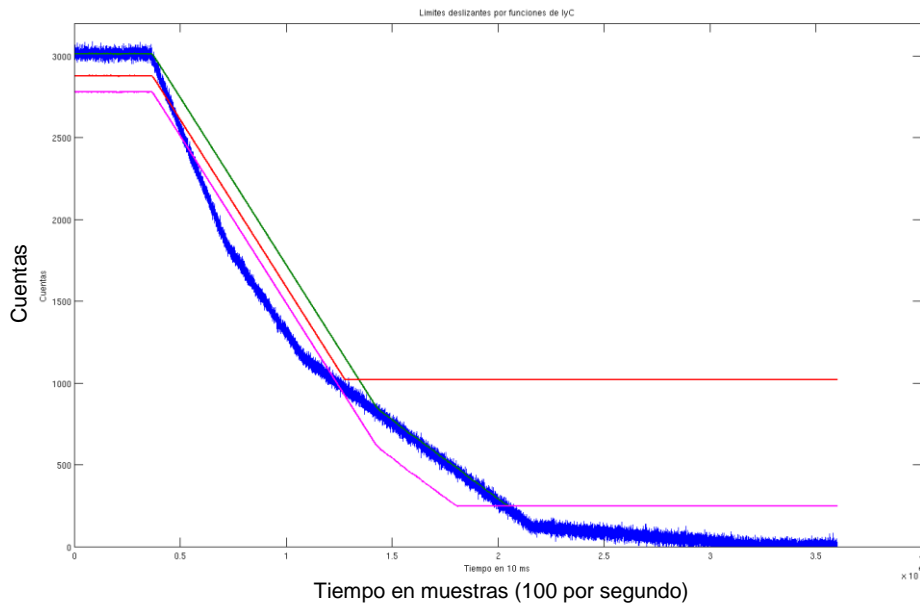


Figura 19 - Límites deslizantes de PP calculados con las funciones de lyC

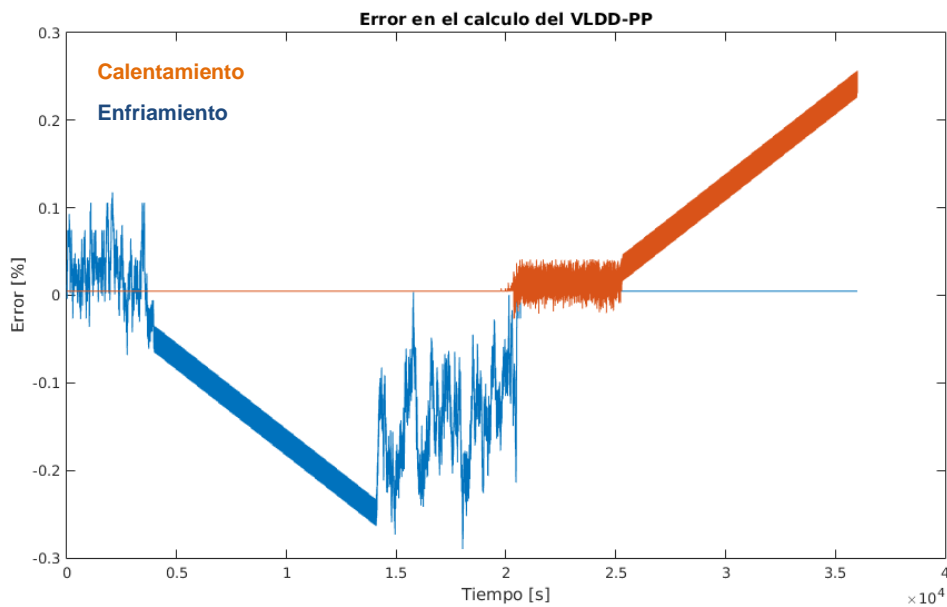


Figura 20 - Error en el cálculo del VLDD-PP para enfriamiento abrupto y calentamiento

Como se puede observar en la Figura 20 el error entre las señales obtenidas con simulink y las obtenidas con las funciones de lyC se mantiene dentro del 0,3%.

Sin embargo, se nota una deriva del error en los casos en que la pendiente de la señal de entrada es muy abrupta, esto se debe a un error acumulativo, por redondeo en el algoritmo. En este caso se simuló el procesamiento con 100 muestras por segundo de la señal de entrada. Este limitador de tasa está pensado para señales lentas. Para el cálculo de la tasa, se realiza la siguiente cuenta:

$$Pendiente = \frac{3600 \cdot frecuencia\ muestreo}{máxima\ tasa}$$

Donde la pendiente es la cantidad de muestras que tiene que esperar el algoritmo para subir o bajar una cuenta la señal de salida. En el procesamiento se debe calcular el divisor por el cual se divide la frecuencia de muestreo y se redondea a un número entero. Es importante que la frecuencia de muestreo sea lo suficientemente alta para que la pendiente tenga la precisión deseada.

Para el sistema final se propone elevar la cantidad de muestras por segundo a 1000 que reduce el error a la mitad. Otra opción, es implementar un divisor de frecuencias con frecuencia fraccionaria, lo que reduciría el error aún más.

Los valores de BPP0-DEL y BPP1-DEL determinan la distancia entre la señal a comparar y los Límites. Se recomienda que estos valores estén a una distancia mínima del 10% para evitar disparos espurios.

4.3 Valor límite deslizante de presión del primario a flujo neutrónico

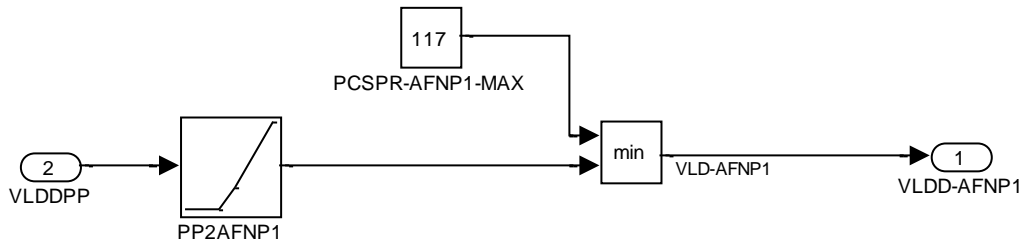


Figura 21 - Especificación de la generación del límite deslizante para alto flujo

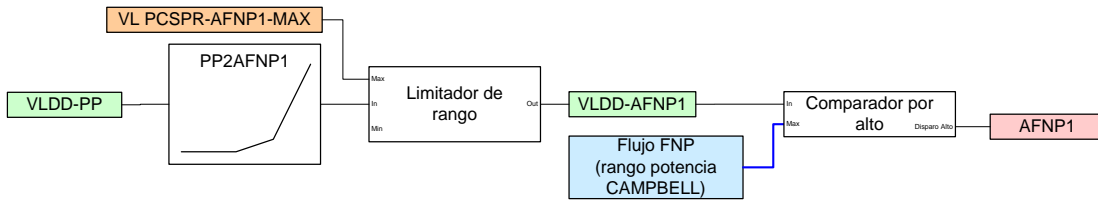


Figura 22-Diagrama en bloques de la generación del límite deslizante para alto flujo

Para generar el límite por flujo alto a partir de la PP se utiliza una tabla para convertir las unidades de entrada (presión en MPa) a porcentaje de potencia de flujo neutrónico. La conversión se realiza basándose en las simulaciones termohidráulicas y neutrónicas del reactor.

4.3.1 Comparación

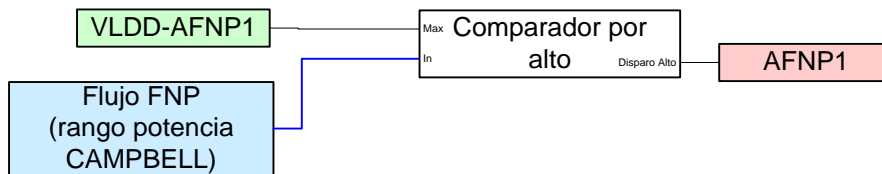


Figura 23- Uso de VLDD-AFNP1 en el módulo de comparación

En la Figura 23 se observa que el límite deslizante VLDD-AFNP1 se utiliza como un máximo para el flujo en potencia. Es decir que el flujo máximo al que puede funcionar el reactor depende directamente de la presión a la que está operando.

4.3.2 Tabla de conversión a flujo

En la tabla del simulink la función llega hasta 12.25MPa. Se expande la interpolación a todo el rango, de 0 a 13 MPa. Se mapea el 0% de flujo como 0 cuentas.

Tabla 6 - Tabla de conversión de Presión a Flujo Neutrónico

X en MPa	0	4,7	7,0	12,25	13
X en cuentas	0	1157	1723	3015	3200
Y en % de flujo	17	17	29	117	130
Y en cuentas	418	418	714	2880	3200

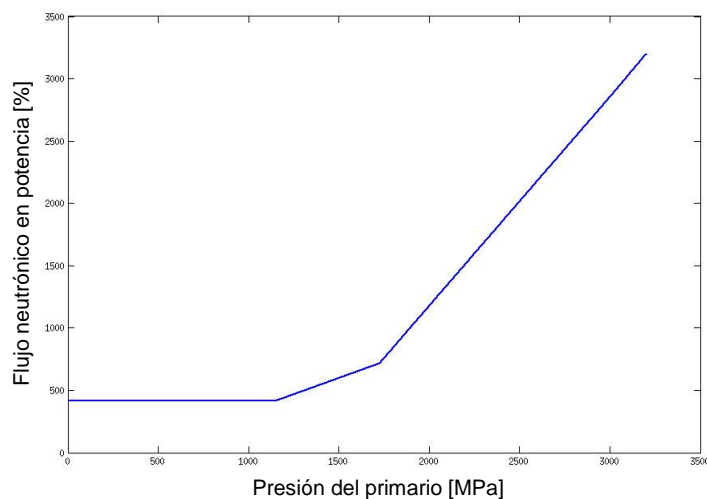


Figura 24 - Curva de conversión entre PP y FNP (Flujo neutrónico en potencia)

La función de conversión se puede implementar en el FPGA como una tabla de 12bits de entrada y 12 bits de salida. La incertidumbre de la señal aumenta al ser amplificada.

En la zona de atenuación leve, de 4,7MPa a 7MPa la ganancia es:

$$\frac{\text{rango de salida en cuentas}}{\text{rango de entrada en cuentas}} = \frac{714 - 418}{1723 - 1157} = 0,52$$

Entonces el error de flujo en ese rango (17 a 29%) es:

$$\text{Presicion original} \cdot \text{Ganancia} = 1\% \cdot 0,52 = 0,52\%$$

En la zona de amplificación, de 7MPa a 13MPa:

$$\frac{\text{rango de salida en cuentas}}{\text{rango de entrada en cuentas}} = \frac{3200\text{cuentas} - 714\text{cuentas}}{3200\text{cuentas} - 1723\text{cuentas}} = 1,68$$

Entonces el error de flujo en ese rango (29 a 130%) es:

$$\text{Presicion original} \cdot \text{Ganancia} = 1\% \cdot 1,68 = 1,68\%$$

Se duplica el error en ese rango.

4.3.3 Implementación de tabla de conversión de unidades

Para la implementación se utilizará una memoria a modo de tabla, donde el valor de entrada es la dirección a leer y el valor de salida, es la señal generada.

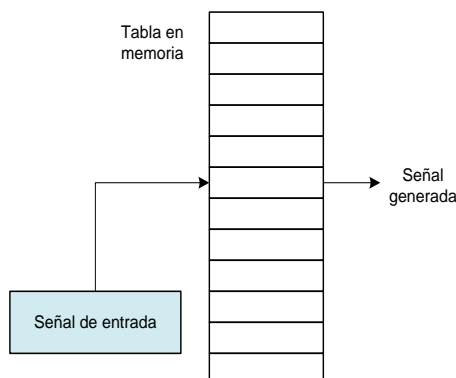


Figura 25 - Diagrama en bloques de la implementación de una conversión de unidades

De esta forma, se pueden realizar tanto conversiones de funciones lineales a tramos como de funciones no lineales, sin necesidad de aumentar la complejidad de las lógicas implementadas en el FPGA.

Las tablas de conversión podrían ser modificables, pero esto implicaría, una intervención demasiado grande por parte del operador / técnico a cargo del sistema. La cantidad de datos a ingresar es demasiado grande, y podría haber errores en el ingreso, disminuyendo la confiabilidad del sistema. Por lo tanto, y debido a que la conversión depende de fenómenos físicos, estas tablas son fijas y se cargan en fábrica.

4.3.4 Limitación de rango

Se implementa un máximo de flujo. Este valor está configurado en 117%, pero según la tabla de conversión anterior, a 13MPa, que es la presión máxima que puede medir dicho sensor, el flujo calculado sería de 112,57%. Esto hace que dicho límite no tenga ningún efecto sobre el valor límite deslizante calculado.

Si se necesita que se pueda calcular un flujo neutrónico hasta ese valor se debería cambiar el rango de la presión medida o la curva de conversión.

Tabla 7 – Parámetro de limitación de rango para VLDD-AFNP

Variable física	% del flujo	Cuentas
Flujo convertido de PP	117	2880

4.3.5 Ejemplos de conversión VLDD-PP a VLDD-AFNP

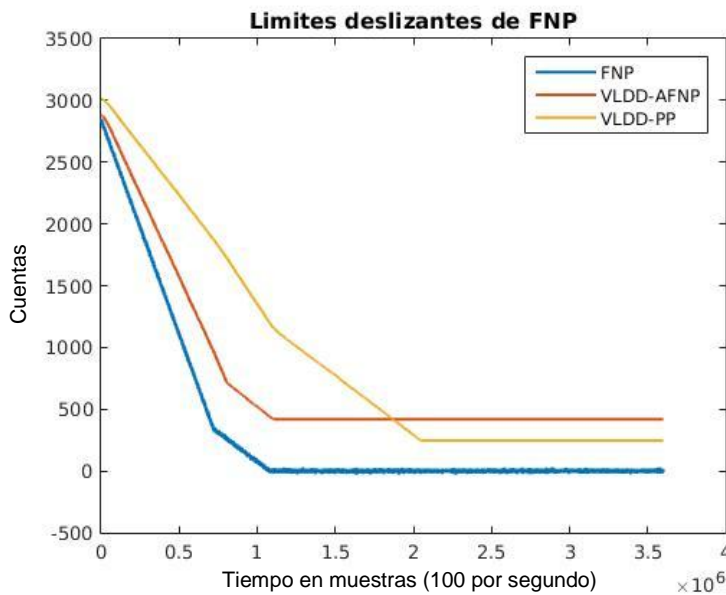


Figura 26 - Flujo neutrónico calculado con PP de enfriamiento abrupto

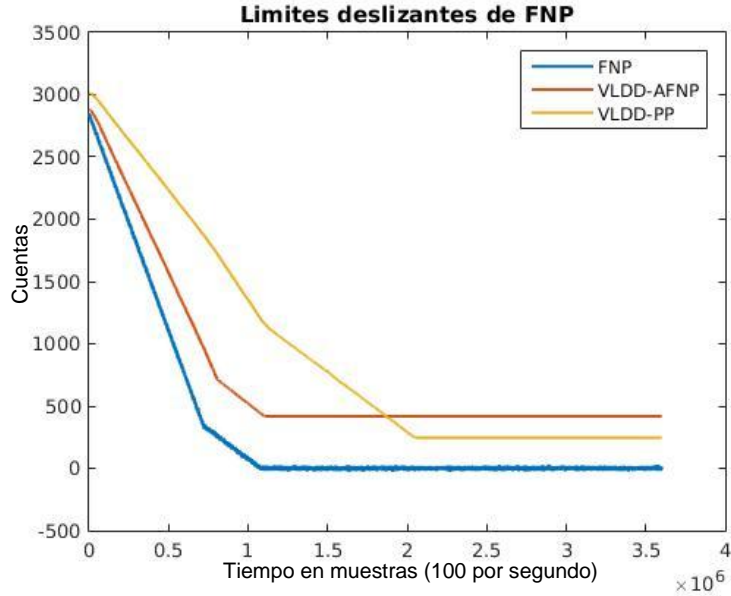


Figura 27 - Flujo neutrónico calculado con PP de enfriamiento normal

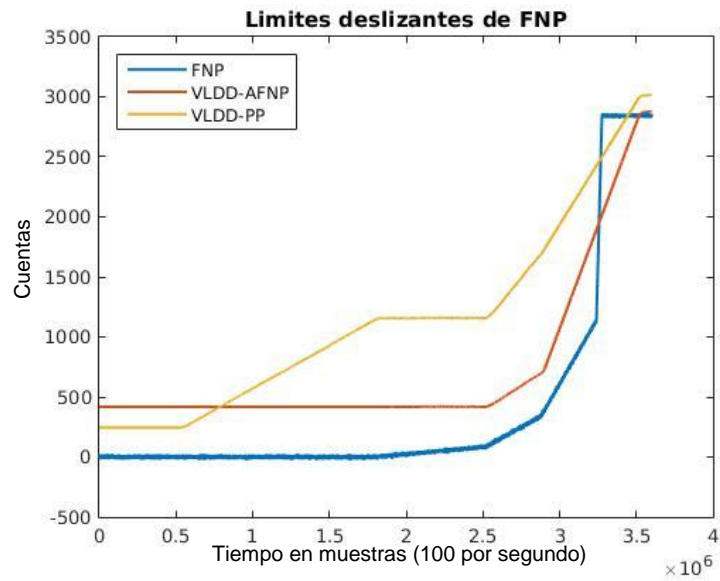


Figura 28 - Flujo neutrónico calculado con PP de calentamiento

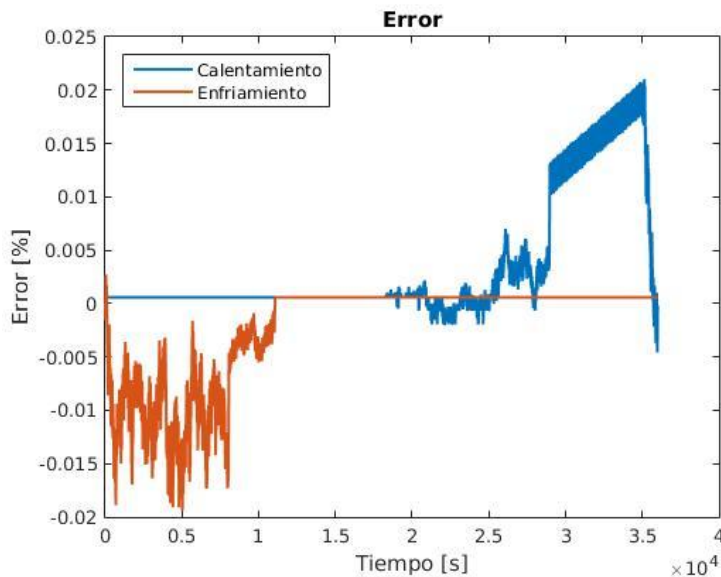


Figura 29 - Error en el cálculo del VLDD-PP para enfriamiento abrupto y calentamiento

Se puede observar también en este caso una deriva en las pendientes abruptas. Aún así el error es menor al 0,025%.

4.4 Valor límite deslizante VLDD-FNP

Este VLDD no se utiliza en ningún comparador sino que sirve para generar:

- Caudal de entrada a los generadores de vapor (VLDD-BCGV)
- Delta T en los generadores de vapor que se utiliza en el cálculo de VLDD-ATEN y VLDD-BTEN

4.5 Digitalización de flujo neutrónico y obtención de VLDD-FNP

El formato final de la señal de flujo neutrónico depende de la implementación del SIN. En este caso se tomará el flujo neutrónico como una variable de 12bits, digitalizada de la siguiente forma:

Tabla 8 – Parámetros de adquisición de flujo neutrónico en potencia

Variable física	% del flujo	Cuentas
Flujo medido en potencia mínimo (Campbell)	0	0
Flujo medido en potencia máximo (Campbell)	130	3200

4.6 Limitación de rango y tasa para VLDD-FNP



Figura 30 – Especificación de la generación del VLDD-FNP según requerimientos en simulink

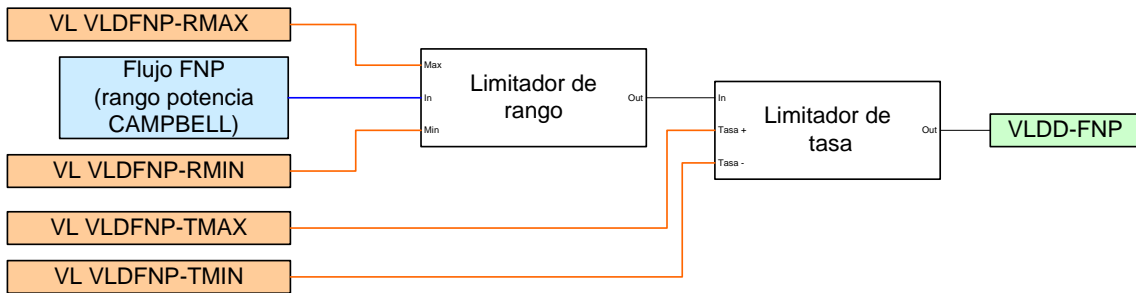


Figura 31 – Diagrama en bloques de la implementación de VLDD-FNP

Al igual que PP, el valor Límite deslizante de flujo neutrónico se construye a partir de las mediciones del sensor, en este caso el SIN, con un límite de rango y uno de tasa. Se toma como máximo flujo medido 130%.

Tabla 9 – Parámetros configurables para el cálculo VLDD-FNP

Parámetro	% de flujo	Cuentas
Límite de rango superior	100	2462
Límite de rango inferior	3	74
Límite de tasa en crecimiento	5 por minuto	123
Límite de tasa en decrecimiento	15 por minuto	369

4.6.1 Resultados de la generación de VLDD-FNP

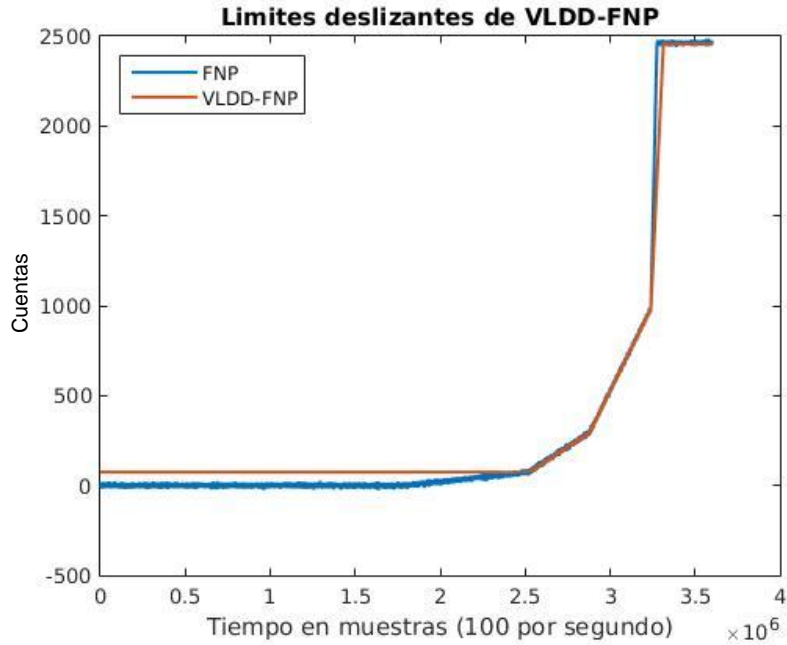


Figura 32 – VLDD-FNP para calentamiento

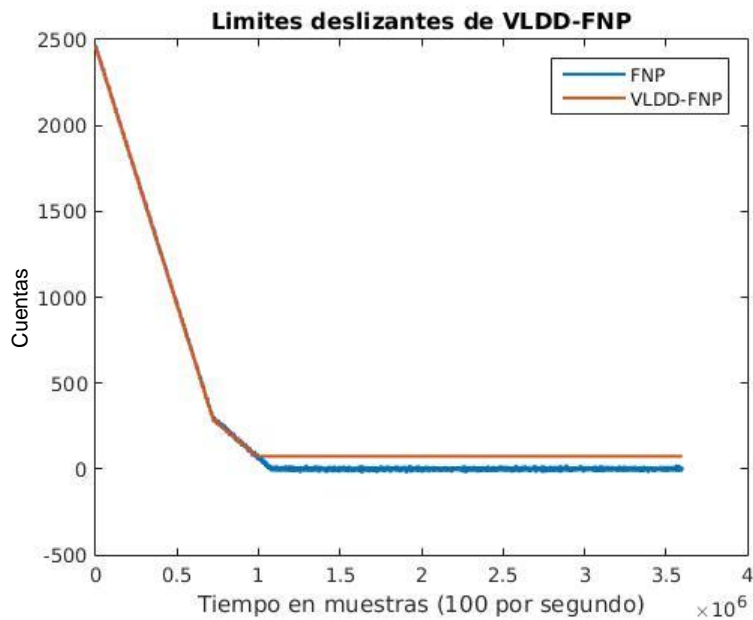


Figura 33 – VLDD-FNP para enfriamiento normal

En esta etapa no se generan errores significativos más allá de los expuestos anteriormente ya que no hay una tabla de conversión.

4.7 Calculo de caudal de generador de vapor a partir del flujo (VLDD-BCGV)

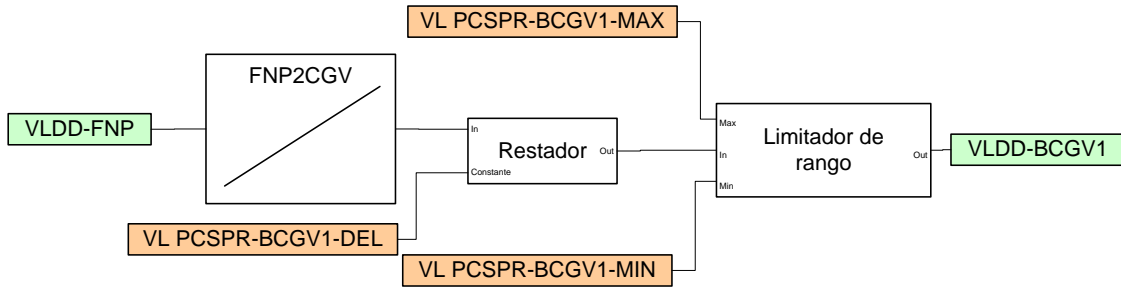


Figura 34 – Diagrama de la implementación de la conversión de VLDD-FNP a límite de Bajo caudal en GV.

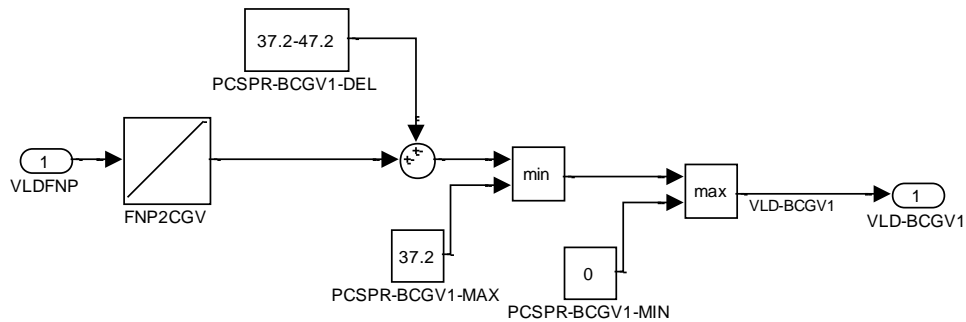


Figura 35 – Especificación de la conversión de VLDD-FNP a VLD-BCGV según requerimientos de simulink

4.7.1 Comparación con caudal del GV

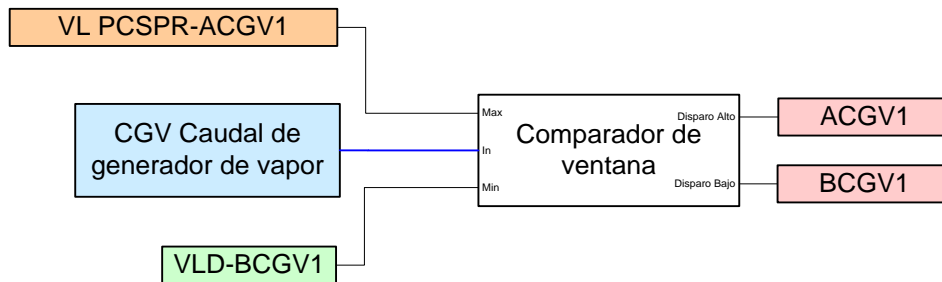


Figura 36 – Comparación de la señal Caudal del generador de vapor

Como se puede observar en la Figura 36, el límite generado VLDD-BCGV se utiliza para comparar el caudal del generador de vapor.

Además del límite deslizante se compara esta señal con un límite fijo por alto.

4.7.2 Rango dinámico del caudal del GV's

Para la digitalización de la variable de CGV (caudal de alimentación de GV's) se elige un rango total de 0 a 50 kg/s.

Este rango debe coincidir con el sensor que se utilice para esta variable, de manera que una determinada cantidad de cuentas, coincida con una misma cantidad de magnitud física, en el caso de la variable convertida y la medida por el sensor.

4.7.3 Conversión a Caudal de alimentación en los GV's

Tabla 10 - Tabla de conversión de Flujo neutrónico a caudal de alimentación en los GV's

X en % de flujo	0	100
X en cuentas	0	2462
Y en °C	0	47,2
Y en cuentas	0	3008

4.7.4 Delta para obtención de límite

Para obtener el límite BCGV (Bajo caudal en los GV's) se resta un delta de 10 kg/s y se limita en rango de 0 a 37.2 kg/s.

Tabla 11–Parámetros para la digitalización del caudal del GV y configuración del límite BCGV

Parámetro	Caudal [kg/s]	Cuentas
Bajo caudal en los GV's mínimo	0	0
Bajo caudal en los GV's máximo	50	3200
Delta para límite - BCGV1-DEL	10	640
Valor del máximo para límite de rango	37.2	2381
Valor del mínimo para límite de rango	0	0

4.7.5 Ejemplos de obtención de BCGV

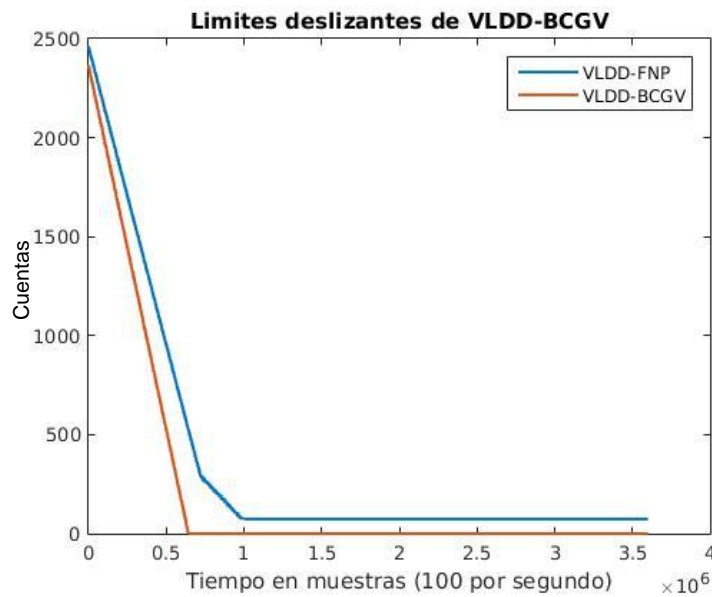


Figura 37 - Calculo de BCGV con flujo de enfriamiento normal

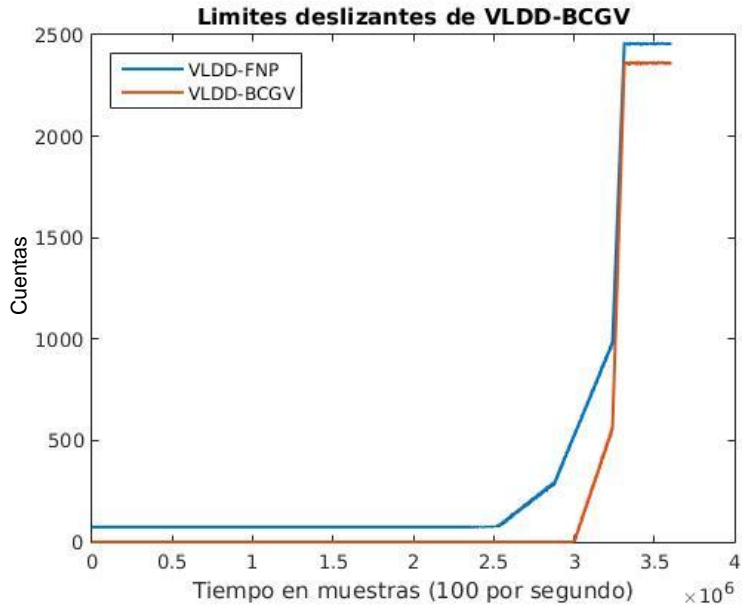


Figura 38 - Cálculo de BCGV con flujo de calentamiento

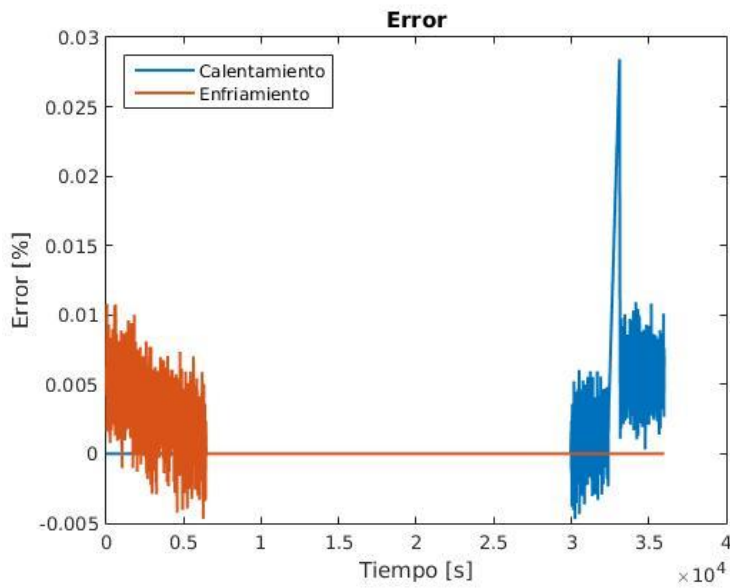


Figura 39 – Error en el cálculo del límite BCGV para calentamiento y enfriamiento normal.

Como se puede observar en la Figura 39, el error produce un pico en la subida rápida de flujo. Esto se debe nuevamente al error en el cálculo del divisor de frecuencia para el limitador de tasa. El error se disminuye a medida que aumenta la tasa de muestreo.

4.8 Cálculo de Temperatura de salida de los GV's (TEN)

Los valores de comparación para alta y baja temperatura se obtienen de la conversión de:

- Valor Límite deslizante de flujo (VLDD-FNP) que se convierte a Variación de temperatura en generadores de vapor
- Valor Límite deslizante de presión (VLDD-PP) que se convierte a temperatura de saturación

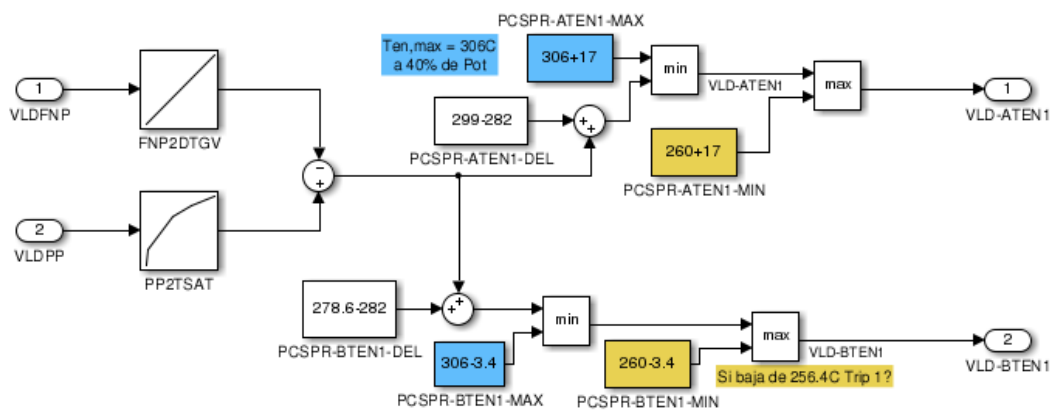


Figura 40 - Especificación de la generación de temperatura de entrada según requerimientos en simulink

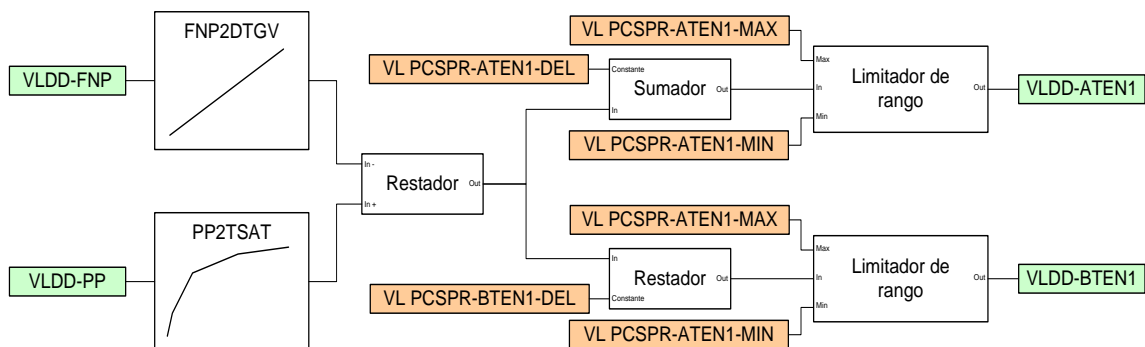


Figura 41 – Diagrama en bloques de la implementación de la generación de temperatura de entrada

4.8.1 Comparación con temperatura de salida de GV's

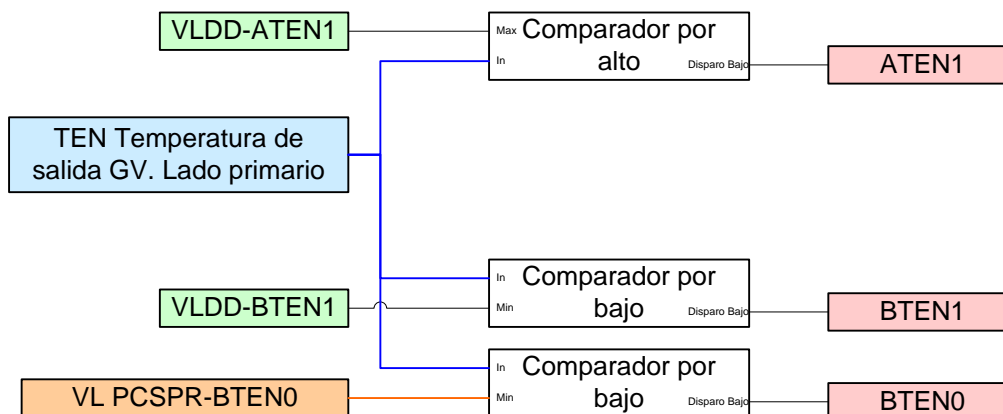


Figura 42– Comparación de la temperatura de salida de GV's

Como se puede observar en la Figura 42, los límites generados VLDD-ATEN y VLDD-BTEN se utilizan para comparar la temperatura de salida de los generadores de vapor. Además de los límites deslizantes se compara esta señal con un límite fijo por bajo.

4.8.2 Conversión de Presión de primario limitado en tasa a temperatura de saturación (VLDD-PP → TSAT)

Para la conversión de la presión del primario a temperatura de saturación se usa una función inversa de la ecuación de Antoine. En esta primera implementación se utiliza una aproximación de la curva utilizando segmentos. En la implementación final se espera utilizar una tabla con los valores correctos para cada punto.

Tabla 12 - Tabla de conversión de Presión a Flujo Neutrónico

X en MPa	0	0,1	0,4	4,7	8,0	12,25	13
X en cuentas	0	25	98	1157	1969	3015	3200
Y en °C	0	100	150	260	295	326	330
Y en cuentas	0	970	1455	2521	2861	3161	3200

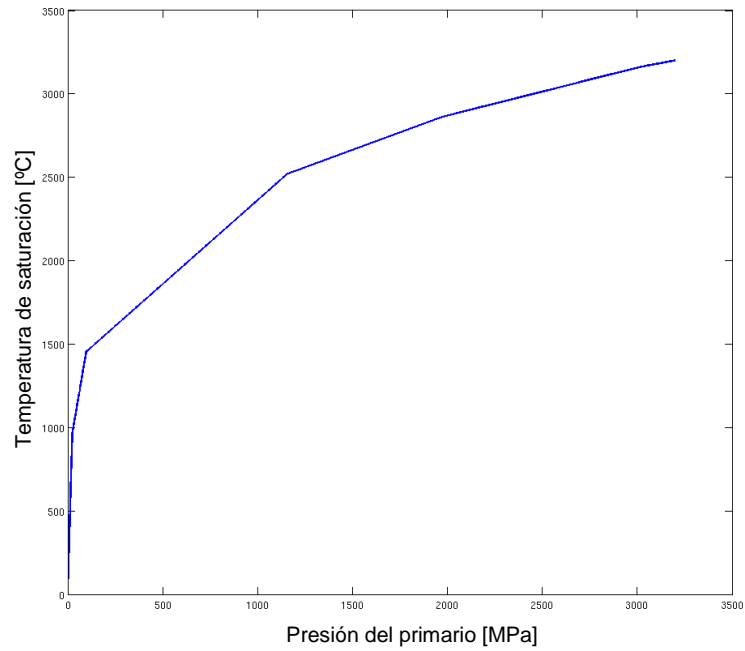


Figura 43 - Curva de conversión entre PP y TSAT (Temperatura de saturación)

4.8.3 Cálculo del error en la conversión PP-TSAT

Para medir el error que inserta la curva en la señal, se toma cada segmento y se calcula la amplificación que genera.

Tabla 13 – Cálculo del error generado por la conversión de Presión a Temperatura de saturación

Rango de entrada en 10xPascal	0-0,1	0,1-0,4	0,4-4,7	4,7-8,0	8,0-12,25	12,25-13
Delta X en cuentas	25	73	1059	812	1046	185
Delta Y en cuentas	970	485	1066	340	300	39
Amplificación	38,8	6,6438	1,0066	0,4187	0,2868	0,2108
Error en la entrada (VLDD_PP)	1%	1%	1%	1%	1%	1%
Error en la salida	38,8%	6,64%	1,01%	0,42%	0,29%	0,21%

Como se puede observar en la tabla, la amplificación de los dos primeros segmentos amplifica también el error de la adquisición del sensor. Esto hace que no se pueda esperar una precisión de menos del 38% en este segmento. Queda limitada la confiabilidad, si dispara después de lo esperado, o bien, la disponibilidad, si dispara antes.

Como la variable PP está limitada en rango desde 1MPa no se ve que el error afecte el resto de la cadena. Si se deseara realizar esta conversión para valores menores a 1MPa, se deberá utilizar otro sensor con una calibración en un rango dinámico más reducido para que el error sea menor.

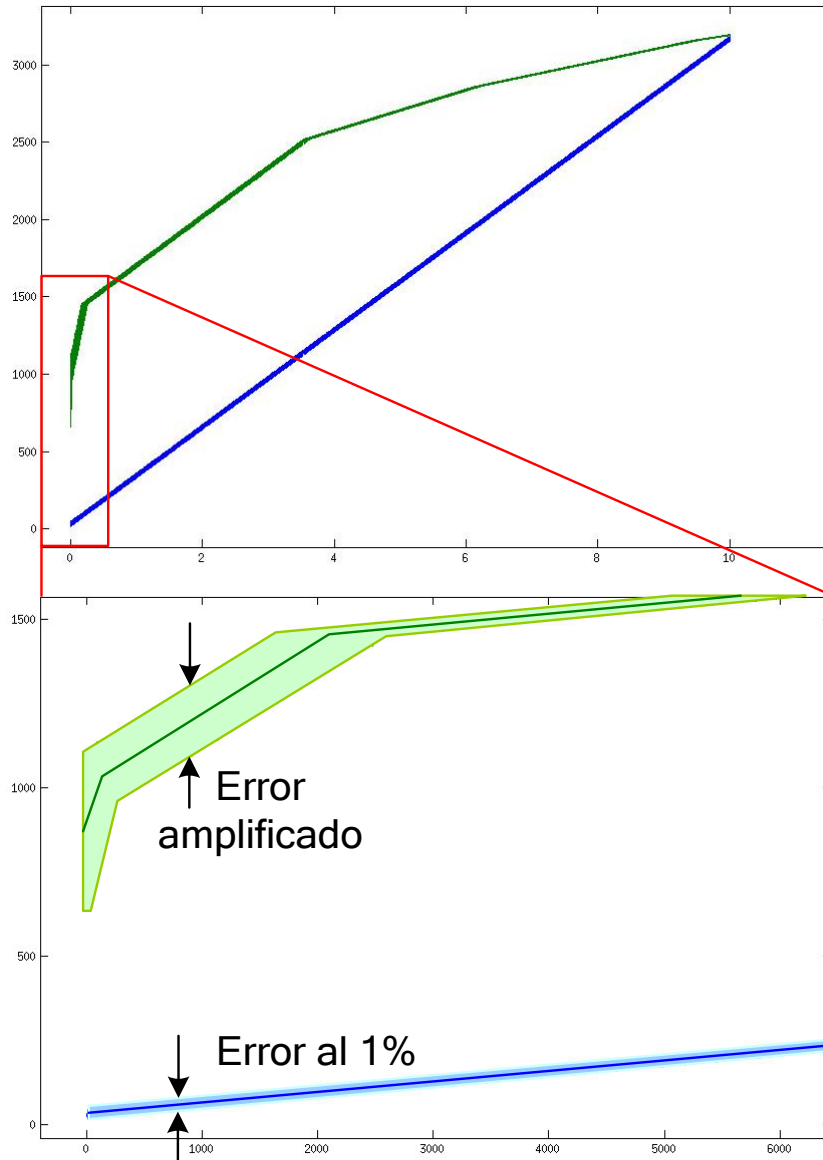


Figura 44 – Ejemplo de amplificación del error en una rampa con error del 1%

En la Figura 44 se puede observar la señal de entrada (azul) con un error del 1%. Para representar el rango de incertidumbre que implica este error se marcó el área alrededor del valor real. Cuando se amplifica la señal al pasar por la tabla de conversión, también se amplifica el error y la incertidumbre aumenta (área en verde claro).

4.8.4 Conversión ejemplo

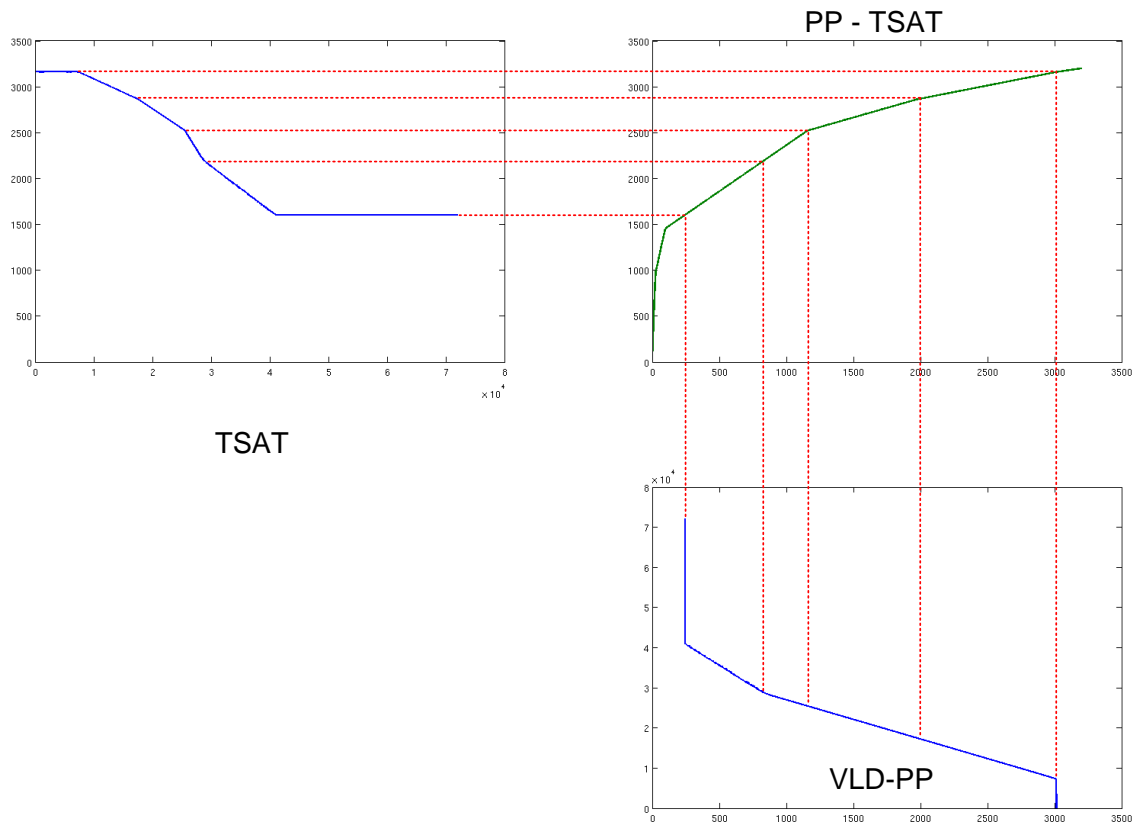
CONVERTIR_IYC es una función que realiza la conversión de unidad de un vector de datos con las muestras de una señal en el tiempo.

$$[\text{salida}] = \text{convertir_iyc}(\text{entrada}, \text{tabla})$$

Entrada: La señal a convertir, debe ser un vector de una dimensión. Deben ser valores de 0 a 3200

Tabla: Es un vector de longitud 3200 donde cada índice corresponde a una entrada de la señal y cada valor guardado corresponde a la salida convertida

$$\text{Tsat} = \text{iyc_convertir}(\text{vldd-pp}, \text{PP2TSAT});$$



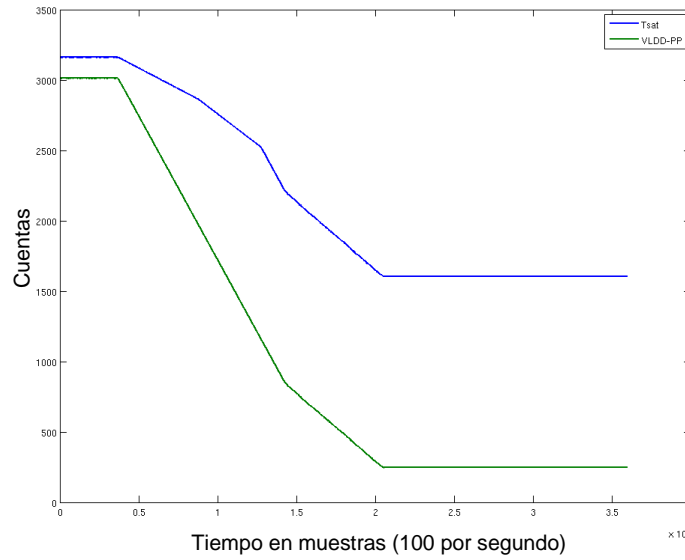


Figura 45 - Calculo de TSAT para Enfriamiento

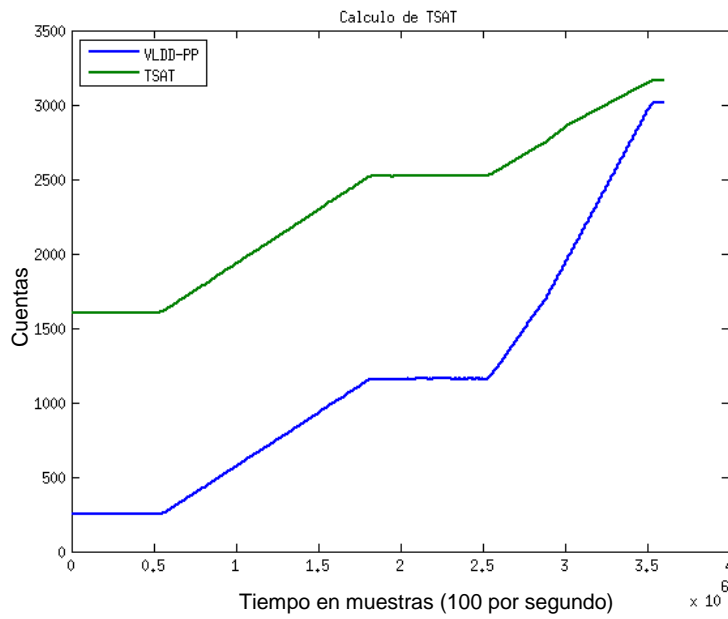


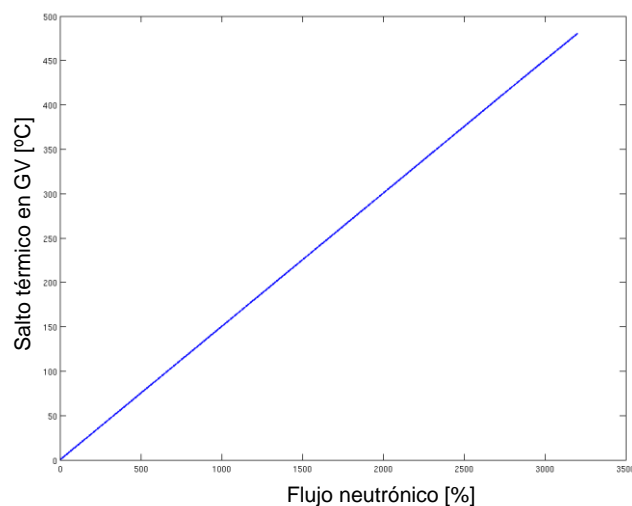
Figura 46 - Calculo de TSAT para calentamiento

4.8.5 Conversión de Flujo neutrónico en potencia a Salto térmico en el GV del lado primario TSAT-TEN (FNP →DTGV)

Para la conversión a salto de temperatura en GV's, se debe tener en cuenta que la escala de digitalización a temperatura tiene que ser compatible con la conversión VLDD-PP → TSAT del punto anterior. Ahí se establecía que la temperatura máxima era de 330°C, es decir que una cuenta equivalía a 0,103125 °C. De otra forma no se podrían restar.

Tabla 14 - Tabla de conversión de Flujo neutrónico a salto térmico en GV

X en % de flujo	0	100	112,57	
X en cuentas	0	2843	3200	
Y en °C	0	44	49,53	330
Y en cuentas	0	427	480	3200



4.8.6 Cálculo de error en la conversión FNP-DTGV

Tabla 15 – Cálculo del error generado en la conversión de FNP a DTGV

Rango de entrada en % de flujo	0-100	100-112.57
Delta X en cuentas	2843	357
Delta Y en cuentas	427	53
Amplificación	0,15	0.15
Error en la entrada (VLDD-FNP)	1%	1%
Error en la salida	0,2%	0,2%

En este caso se puede observar que la amplificación reduce el error en la medición y aumenta la precisión, ya que se trata de una atenuación.

4.8.7 Conversión ejemplo

En la Figura 47 se observa el cambio de proporción es al convertir la señal de flujo neutrónico en diferencia de temperatura en GV's.

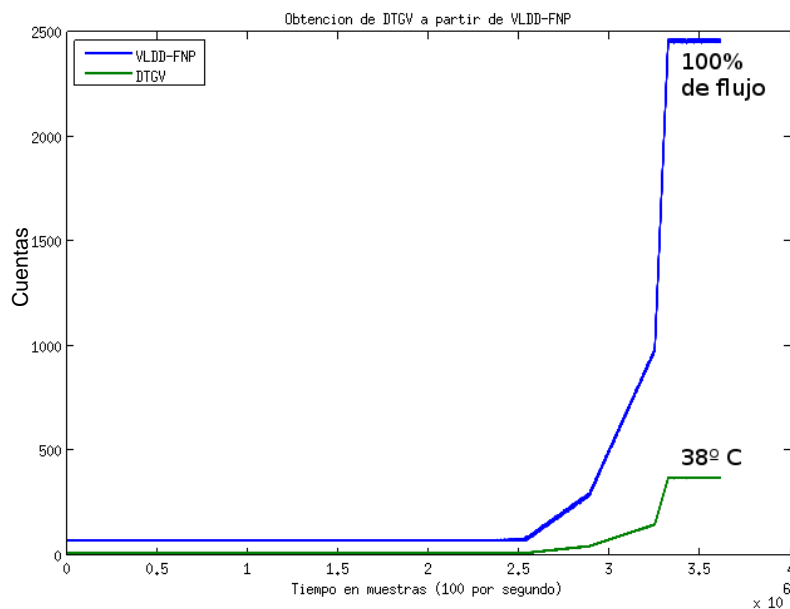


Figura 47 – Cálculo de DTGV para calentamiento

4.8.8 Resta entre Tsat y DtgV

Para la implementación de la resta, se utilizarán restadores convencionales de 16bits. Pueden ser elaborados con la configuración o utilizar los que provee el fabricante en el FPGA.

Se debe tener en cuenta que estos restadores deben saturar por bajo, de manera que si el número resultante es menor a cero, no produzca un error por rollover.

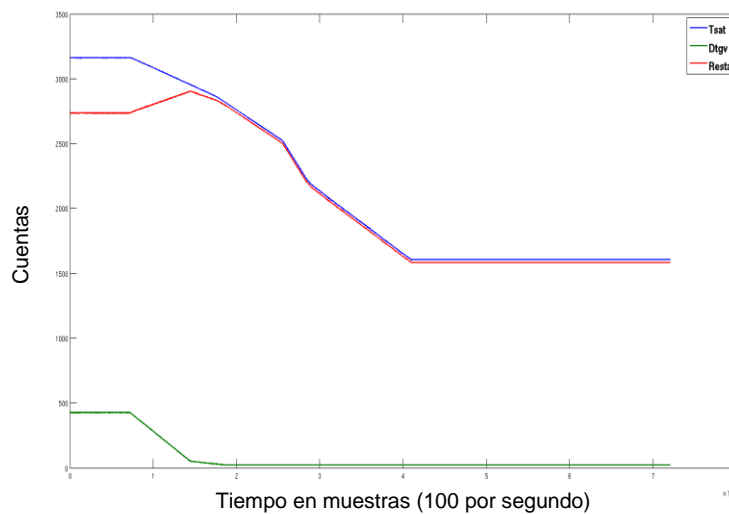


Figura 48 – Resta de Tsat y DtgV para enfriamiento abrupto

4.8.9 Obtención de VLD-ATEN1

Para la obtención de este Límite deslizante se le suma el valor de configuración PCSPR-ATEN1-DEL a la resta entre Tsat y DtgV. En el archivo de simulink este valor está configurado en 17°C.

$$\frac{3200 \text{cuentas}}{330^{\circ}\text{C}} \cdot 17^{\circ}\text{C} = 165 \text{cuentas}$$

Luego se lo limita en rango entre 323 (3132cuentas) y 277°C (2686cuentas).

4.8.10 Obtención de VLD-BTEN1

Para la obtención de este límite deslizante se le resta el valor de configuración PCSPR-BTEN1-DEL a la resta entre T_{sat} y D_{tg}v. En el archivo de simulink este valor está configurado en 3,4°C.

$$\frac{3200\text{cuentas}}{330^{\circ}\text{C}} \cdot 3,4^{\circ}\text{C} = 33\text{cuentas}$$

Luego se lo limita en rango entre 302,6 (2934cuentas) y 256,6°C (2488cuentas).

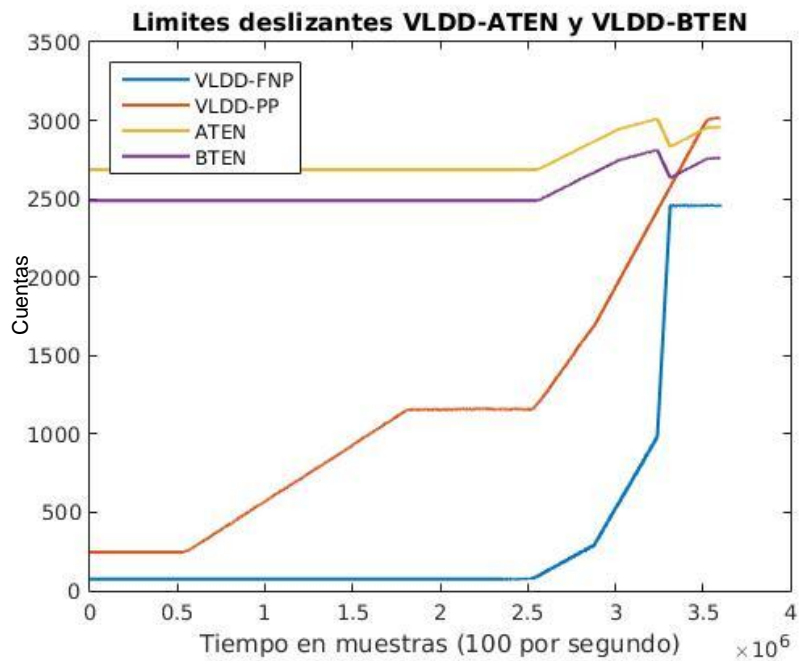


Figura 49 – Calculo de VLDD-ATEN y VLDD-BTEN para FNP y PP de calentamiento

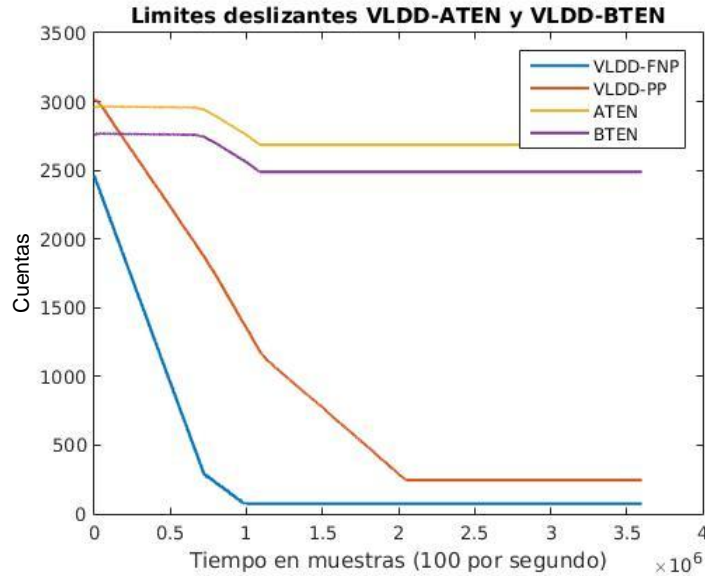


Figura 50 - Cálculo de VLDD-ATEN y VLDD-BTEN para FNP y PP de enfriamiento normal

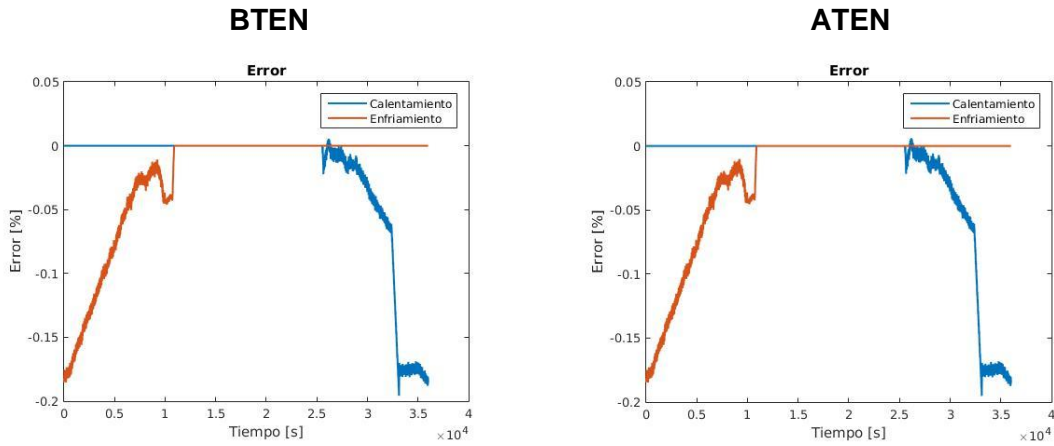


Figura 51 – Error en el cálculo de ATEN y BTEN

Si bien el error obtenido en este caso es menor al 0,2%, se debe aclarar que esta medición, no tiene en cuenta el error generado por la conversión de unidades, ya que es un error inherente de la operación y no depende de la implementación. Afecta tanto a la simulación como a la implementación en FPGA.

5. Conclusiones

El máximo error que agrega la implementación es del 0,3%.

Considerando que los módulos de adquisición de señales analógicas tienen un máximo error del 1% y el error del sensor esta alrededor del 2% a fondo de escala, se puede decir que la implementación utilizando, tecnología basada en FPGA, agrega un error inferior a las otras etapas.

Con este modelo construido se prueba la factibilidad de la implementación de estos cálculos en lógica basada en FPGA, siempre que se cumplan las siguientes condiciones:

1. Los sensores deben ser lineales.

Sensores no lineales implicarían reconfigurar las curvas de conversión entre magnitudes (FNP2CGV, PP2AFNP, FNP2DTGV y PP2TSAT) cada vez que se cambie el sensor o su función de transferencia.

2. Los sensores deben estar calibrados en rango.

Para disminuir el error por cuantificación, se debe utilizar todo el rango útil del digitalizador.

3. Las curvas de conversión entre magnitudes deben ser fijas.

El procedimiento para realizar la configuración de dichas curvas en operación es muy complejo y requeriría un estudio a otro nivel. Las curvas están construidas con tablas de 4096 valores, y es muy difícil que sean configuradas sin errores por un usuario del sistema.

4. La configuración de los parámetros debe ser tal que los VLDD generados tengan un margen del 10% con respecto a la variable contra la cual se comparan para evitar disparos espurios.

5. Se deben muestrear las señales con una frecuencia mayor a 100 veces por segundo.

Esto permite que se reduzca el error en el cálculo del seguimiento de la señal con tasa limitada.

6. El parámetro de configuración VL-VLDD-PP-RMIN en el cálculo de VLDD-PP debe tener un mínimo configurable de 0,4MPa.

Esto se debe a que esta señal se utiliza para el cálculo de TSAT cuya alta pendiente en valores bajos aumenta la incertidumbre del cálculo, al amplificar el error de medición del sensor y la electrónica de adquisición.

7. El parámetro de configuración VL-VLDFNP-RMIN en el cálculo de VLDD-FNP debe tener un mínimo configurable de 0,1%.

Esto se debe a que para que la variable FNP sea válida la cadena Campbell debe estar en rango. Para estos dos últimos puntos, se puede dividir el rango del sensor para duplicar la precisión en cada rango.

6. Referencias bibliográficas

- [1] Rodríguez, F. G, Verrastro, C.A. *Ingeniería básica del sistema de protección para el reactor CAREM25*. CNEA ET-06Y-042, 2013
- [2] Etchepareborda, A. *Requerimientos funcionales del PSPR para el reactor CAREM25*, Proyecto CAREM25, 2014
- [3] Jukka Ranta, *The current state of FPGA technology in the nuclear domain*, VTT Publications, 2012.
- [4] IAEA Nuclear Energy Series Report. *Application of Field Programmable Gate Arrays in Instrumentation and Control Systems of NPPs*, 2014.
- [5] *International Electrotechnical Commission. Nuclear power plants - Instrumentation and control important to safety - General requirements for systems*. IEC 61513 Ed.2, 2011.
- [6] IAEA Safety Standards Series. *Instrumentation and Control Systems Important to Safety in Nuclear Power Plants Safety Guide*. Safety Guide No. NS-G-1.3, 2002.
- [7] *International Electrotechnical Commission. Nuclear power plants - Instrumentation and control important to safety - Development of HDL-programmed integrated circuits for systems performing category A functions*. IEC 62566 Ed.1, 2012.
- [8] Estryk, D.S., Verrastro, C.A. *Lógica de Scram del RA1 – Mod. CNEA T910*. CNEA MU-14Y-002 Rev.0, 2004.
- [9] Ríos, G.E. *Impulsímetro Digital para Canal de Arranque de Reactores de Investigación*. Instituto Dan Beninson, 2012.
- [10] Rodríguez, F. G, Verrastro, C.A. *Ingeniería básica del sistema de protección para el reactor RA10*. CNEA ET-06Y-051, 2013
- [11] Alarcón, J.E., Estryk, D.S., Ferrucci, F.N., Ríos, G.E. *Ingeniería de Detalle del Sistema de Instrumentación Alternativo para el Accionamiento de Inyección de Boro para la CNAII*. CNEA ET-06Y-043 Rev.1, 2013.
- [12] Bakhmach, I., Kharchenko, V., Siora, O., Tokarev, V., Andrashov, A. *Modernization of NPP I&C Systems using FPGA Technologies: Ukrainian experience*. 2nd International Workshop on The Applications of Field-programmable Gate Arrays in Nuclear Power Plants, 2009.

7. Anexos

7.1 Instrucciones para el uso de funciones en MATLAB

7.1.1 Vectores de entrada

Este vector de entrada fue generado con los ejemplos de entradas de flujo y presión en la propuesta de lógica realizada con Simulink. Se trata de vectores de mediciones de dichas variables con un muestreo de 1 por segundo.

```
load ('variables-entrada.mat')
```

Los nombres de los vectores son:

Variables de entrada	
PPCal	Presión de calentamiento en el primario
PPEnfAb	Presión de enfriamiento abrupto en el primario
PPEnfNor	Presión de enfriamiento normal en el primario
FlujoCal	Flujo de calentamiento
FlujoEnfAb	Flujo de enfriamiento abrupto
FlujoEnfNor	Flujo de enfriamiento normal

7.1.2 Calculo de VLDD-PP

Para generar los gráficos y matrices con los resultados de los VLDD a partir de presión del primario se utiliza la función VLDD_PP_iyc

```
VLDD_PP_iyc (PPCal)
```

La función VLDD_PP_iyc (PP) genera una estructura con los siguientes campos:

VLDD_BPP0	Primer VLDD de baja presión
VLDD_BPP1	Segundo VLDD de baja presión
x: [1x3600000 int32]	Vector de ordenadas para plot
VLDD_PP: [1x3600000 int16]	PP limitada en Tasa y rango para otros límites
PP_dig: [1x3600000 int16]	PP digitalizada

7.1.3 Cálculo de VLDD-FNP

Para generar la matriz con VLDD-FNP se utiliza la función VLDD_FNP_ityc

$VLDD_FNP = VLDD_FNP_ityc(\text{FlujoEnfAb})$

La función VLDD_FNP_ityc (Flujo) genera una estructura con los siguientes campos:

VLDD_FNP: [1x3600000 int16]	VLDD-FNP
x: [1x3600000 int32]	Vector de ordenadas para plot
FNP_dig: [1x3600000 int16]	FNP digitalizado

7.2 Obtención de ATEN y BTEN

Para generar los gráficos y matrices de la temperatura de salida de los GV's se utiliza la VLDD_TEN_ityc.

$TEN = VLDD_TEN_ityc(VLDD_PP.VLDD_PP, VLDD_FNP.VLDD_FNP);$

La función VLDD_TEN_ityc (VLDD_PP, VLDD_FNP) genera una estructura con los siguientes campos:

VLDD_ATEN: [3600000x1 int16]	Límite deslizante VLDD_ATEN
-------------------------------------	-----------------------------

VLDD_BTEN: [3600000x1 int16]	Límite deslizante VLDD_BTEN
x: [1x3600000 int32]	Vector de ordenadas para plot

7.2.1 Obtención de VLDD-BCGV

Para generar los gráficos y las matrices con el límite VLDD-BCGV se utiliza la función

```
BCGV=VLDD_BCGV_iyc(VLDD_FNP.VLDD_FNP)
```

La función VLDD_BCGV_iyc(VLDD_FNP) genera una estructura con los siguientes campos:

BCGV: [3600000x1 double]	VLDD-BCGV
x: [1x3600000 int32]	Vector de ordenadas para plot

7.2.2 Obtención de VLDD-AFNP1

La función VLDD_BCGV_iyc(VLDD_PP,Flujo) genera un vector con el valor Límite deslizante AFNP1. El flujo que se recibe como argumento de la función solo se utiliza para realizar el gráfico.

```
AFNP=VLDD_AFNP_iyc(VLDD_PP.VLDD_PP,FlujoCal);
```

7.2.3 Calculo de errores

Para calcular el error de una señal con respecto a la simulación, se debe primero calcular los resultados de la simulación que se incluyen en el archivo simulaciones.mat

```
load ('simulaciones.mat')
```

Luego se utiliza la función error que recibe los siguientes argumentos:

```
error_iyc(sim_cal,sim_enf,impl_cal,impl_enf,max,min)
```

donde:

sim_cal: es una matriz con los datos de la señal simulada en calentamiento

sim_enf: es una matriz con los datos de la señal simulada en enfriamiento

impl_cal: es una matriz con los datos de la señal de la implementación en calentamiento

impl_enf: es una matriz con los datos de la señal de la implementación en enfriamiento

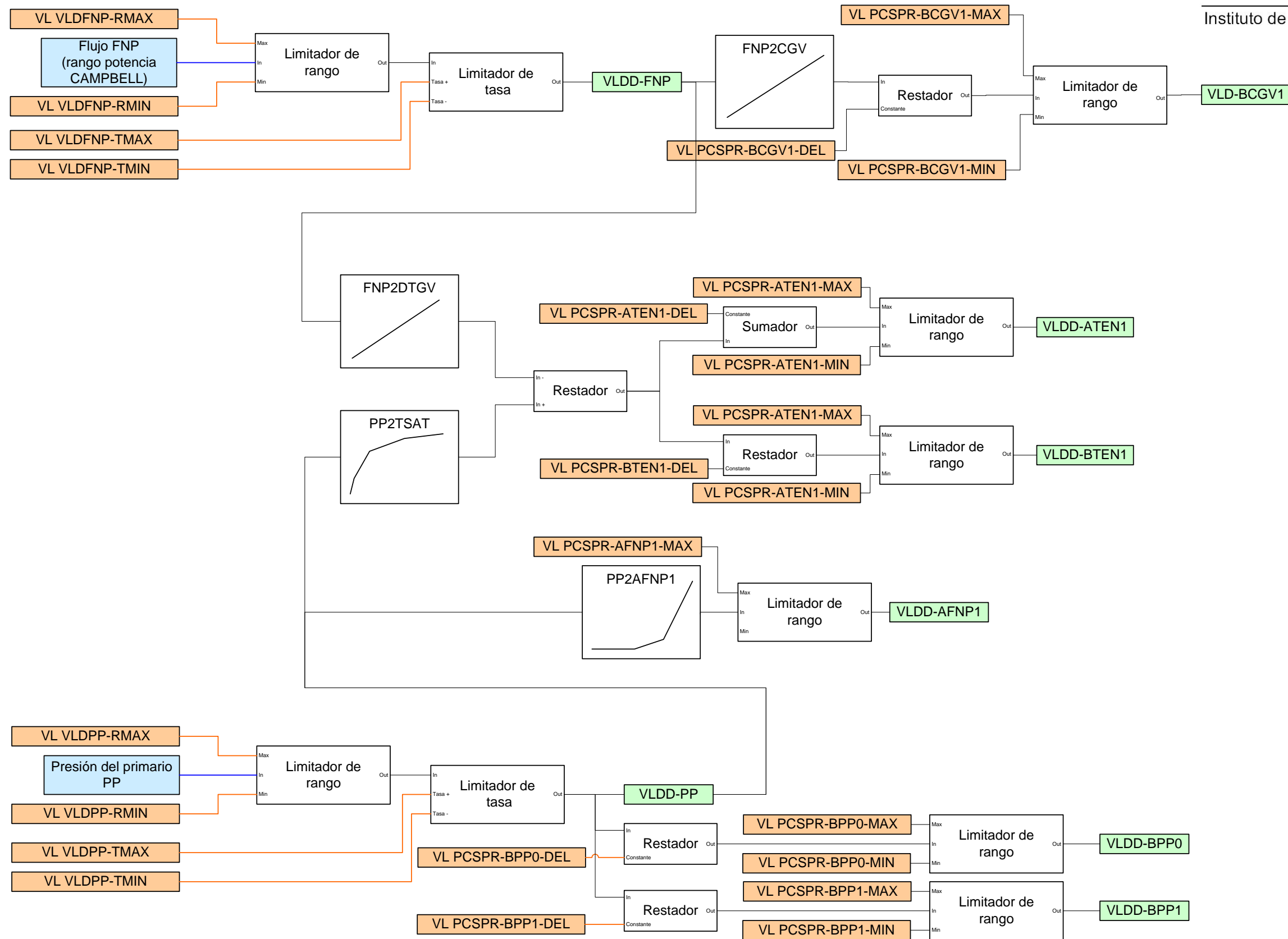
max: es el máximo al que llega la señal convertida en unidades de ingeniería (por ejemplo 330°C)

min: es el mínimo al que llega la señal convertida en unidades de ingeniería (por ejemplo 0°C)

Se genera una matriz con los siguientes campos:

Error_cal	Error en porcentaje de la señal de calentamiento
Compara_cal	Señal de implementación de calentamiento llevada a unidades de ingeniería, para comparación.
Error_enf	Error en porcentaje de la señal de enfriamiento
Compara_enf	Señal de implementación de enfriamiento llevada a unidades de ingeniería, para comparación.

7.3 Esquema general de límites deslizantes



7.4 Código fuente

7.4.1 function salida=digitalizar_iyc(entrada,max,min,sps_in,sps_out)

```
%DIGITALIZAR_IYC es una funcion que realiza un muestreo en numeros
% enteros con una escala fija de 0 a 3200 simulando la digitalizacion
% por el módulode adquisicion y conversion del SPR.
%
% [salida] = digitalizar_iyc (entrada,max,min,sps_in,sps_out)
%
% Entrada: La señal a digitalizar, debe ser un vector de una dimension
%         Puede ser de punto flotante.
%
% Max: El valor de la entrada que se hara coincidir con 3200.
% Min: El valor de la entrada que se hara coincidir con 0.
%
% sps_in: La cantidad de muestras por segundo que tiene la señal de
%         entrada. A simulink le gusta usar 2 muestras por segundo.
% sps_out: La cantidad de muestras por segundo para la señal de
% salida. En el SPR se tiene planificado utilizar 1000.
%
%
unidad=(4000-800)/(max-min);

for i=0:(numel(entrada)/sps_in)-1
for j=1:sps_out
salida(i*sps_out+j)=int16((entrada(i+1)-min)*unidad);
end
end
```

7.4.2 function [salida]=convertir_iyc(entrada,tabla)

```
%CONVERTIR_IYC es una funcion que realiza la conversión de unidad de
% un vector de %datos con las muestras de una señal en el tiempo.
%
% [salida] = convertir_iyc(entrada,tabla)
%
% Entrada: La señal a convertir, debe ser un vector de una dimension
%         Deben ser valores de 0 a 3200
%
% Tabla: Es un vector de longitud 3200 donde cada índice corresponde a
% una entrada % de la señal y cada valor guardado corresponde a la
% salida convertida
%
%
disp(numel(entrada));

for i= 1:numel(entrada)
if entrada(i)<3200
```



UNSAM
UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

```
salida(i)=int16(tabla(entrada(i)));
```

```
else  
disp('Entrada fuera de rango');  
end  
end
```

7.4.3 function salida=sumador_iyc(entrada, suma)

```
% SUMADOR_IYC suma un valor escalar a todo un vector de valores, se  
% comporta como un sumador digital para las señales
```

```
salida=zeros(numel(entrada),1);
```

```
for i=1:numel(entrada)  
salida(i)=entrada(i)+suma;  
end
```

7.4.4 function salida=restador_iyc(entrada, resta)

```
% RESTADOR_IYC resta un valor escalar a todo un vector de valores, se  
% comporta como un sumador digital para las señales
```

```
salida=zeros(numel(entrada),1);
```

```
for i=1:numel(entrada)  
salida(i)=entrada(i)-resta;  
end
```

7.4.5 function

límite=limitador_tasa_iyc(entrada,max_por_hora_up,max_por_hora_down,iteraciones_por_segundo)

```
% LIMITADOR_TASA_IYC es una función que genera una señal que copia la  
% señal de entrada hasta una máxima pendiente para subida y bajada.
```

```
% limitador_tasa_iyc(entrada,max_por_hora_up,max_por_hora_down,  
% iteraciones_por_segundo)
```

```
% entrada: señal de entrada  
% max_por_hora: máximo crecimiento positivo en cuentas por segundo  
% min_por_hora: máximo crecimiento negativo en cuentas por segundo  
% iteraciones_por_segundo: Cantidad de muestras por segundo de la  
% señal  
límite= entrada;
```

```
tiempo_entre_salto_unitario_up= 3600  
*iteraciones_por_segundo/max_por_hora_up;  
tiempo_entre_salto_unitario_down= 3600  
*iteraciones_por_segundo/max_por_hora_down;  
disp(tiempo_entre_salto_unitario_down);
```

```

tiempo_negativo= 0;
tiempo_positivo= 0;

for i = 2:length(entrada)
if(entrada(i)>límite(i-1))
if(tiempo_positivo>tiempo_entre_salto_unitario_up)
límite(i)=límite(i-1)+ 1;
tiempo_positivo= 0;
else
límite(i)=límite(i-1);
tiempo_positivo=tiempo_positivo+ 1;
tiempo_negativo= 0;
end
end

if(entrada(i)<límite(i-1))
%disp 'Es negativo';
if(tiempo_negativo>tiempo_entre_salto_unitario_down)
límite(i)=límite(i-1)- 1;
tiempo_negativo= 0;
else
límite(i)=límite(i-1);
tiempo_negativo=tiempo_negativo+ 1;
tiempo_positivo= 0;
end
end

if(entrada(i)==límite(i-1))
%%La entrada es igual al límite anterior
límite(i)=límite(i-1);
tiempo_positivo= 0;
tiempo_negativo= 0;
end
end

```

7.4.6 function

salida=error_iyc(simu_cal,simu_enf,imple_cal,imple_enf,max,min)

```

% ERROR_IYC es una función que recibe dos pares de señales y las compara entre
% si obteniendo un resultado de la diferencia en forma de porcentaje. Se
% utiliza para calcular el error entre los valores resultantes de la
% simulación en simulink y la implementación de IyC.
%
% [salida] = error_iyc (simu_cal,simu_enf,imple_cal,imple_enf,max,min)
%
%sim_cal: es una matriz con los datos de la señal simulada en calentamiento
% sim_enf: es una matriz con los datos de la señal simulada en enfriamiento
% impl_cal: es una matriz con los datos de la señal de la implementación en
% calentamiento
% impl_enf: es una matriz con los datos de la señal de la implementación en
% enfriamiento
% max: es el máximo al que llega la señal convertida en unidades de ingeniería
% (por ejemplo 330°C)

```

```
% min: es el mínimo al que llega la señal convertida en unidades de ingeniería
% (por ejemplo 0°C)
```

for i=1:36000

```
compara_cal(i)=double(imple_cal(i*100))*(max-min)/3200;
compara_enf(i)=double(imple_enf(i*100))*(max-min)/3200;
end
```

for i=1:36000

```
error_cal(i)=(simu_cal(i)-compara_cal(i))/32;
error_enf(i)=(simu_enf(i)-compara_enf(i))/32;
end
x=1:36000;
```

```
figure('Color',[1 1 1]);
plot(x,error_cal,x,error_enf,'LineWidth',1.5,'LineSmoothing','on');
```

```
title('Error');
xlabel('Tiempo [s]');
ylabel('Error [%]');
```

```
legend('Calentamiento','Enfriamiento')
```

```
salida.error_cal=error_cal;
salida.compara_cal=compara_cal;
salida.error_enf=error_enf;
salida.compara_enf=compara_enf;
```

7.4.7 function salida=VLDD_PP_iyc(PP)

```
% VLDD_PP_IYC es una función que genera la señal de PP limitada
% en tasa y rango VLDD_PP y los límites de presión por bajo
% VLDD_BPP0 y VLDD_BPP1
```

```
PP_dig=digitalizar_iyc(PP,13,0,1,100);
```

```
%Se asume que 0Pa = 4mA 13Pa = 20mA
PP_dig=digitalizar_iyc(PP,13,0,1,100);
```

```
PP_limran=limitador_rango_iyc(PP_dig,246,3015);
```

```
VLDD_PP=limitador_tasa_iyc(PP_limran,738,738,100);
```

```
VLDD_BPP0=VLDD_BPP_iyc(VLDD_PP,234,2781,246);
VLDD_BPP1=VLDD_BPP_iyc(VLDD_PP,135,2880,1022);
```

```
x=int32([1:3600000]);
```

```
figure('Color',[1 1 1]);
plot(x,PP_dig,x,VLDD_PP,x,VLDD_BPP0,x,VLDD_BPP1,'LineWidth',1.5,'LineSmoothing','on');
```

```
title('Límites deslizantes de PP');
xlabel('Tiempo en muestras (100 por segundo)');
legend('PP', 'VLDD-PP', 'VLDD-BPP0', 'VLDD-BPP1')

salida.x=x;
salida.VLDD_PP=VLDD_PP;
salida.PP_dig=PP_dig;
salida.VLDD_BPP0=VLDD_BPP0;
salida.VLDD_BPP1=VLDD_BPP1;
```

7.4.8 function salida=FNP2CGV_iyc(VLDD_FNP)

```
% FNP2CGV_iyc es una función que convierte la señal de flujo
% restringida en tasa a % caudal del generador de vapor
```

```
pp2cgv_x=[0 2462];
pp2cgv_y=[0 3008];
input=[1:3200];
FNP2CGV=interp1(pp2cgv_x,pp2cgv_y,input);

salida=convertir_iyc(VLDD_FNP,FNP2CGV);
```

```
functionsalida=FNP2DTGV_iyc(VLDD_FNP)
```

```
fnp2dtgv_x=[0 2843 3200];
fnp2dtgv_y=[0 427 480];
input=[1:3200];
FNP2DTGV=interp1(fnp2dtgv_x,fnp2dtgv_y,input);
```

7.4.9 function salida=PP2AFNP(VLDD_PP,FNP)

```
% PP2AFNP convierte el valor de presión limitado en tasa a ellímite
% alto de flujo AFNP
```

```
FlujoMax=130;%Flujo maximo que puede medir el SIN
FlujoMin=0;%Flujo minimo que puede medir el SIN
```

```
RangoMax=117;%Valor maximo que puede tomar el límite
RangoMin=0;%Valor minimo que puede tomar el límite
X=[0 1157 1723 3015 3200];
Y=[418 418 714 2880 3200];
Input=[1:3200];
pp2afnp=interp1(X,Y,Input);
```

```
VLDD_AFNP=convertir_iyc(VLDD_PP,pp2afnp);
VLDD_AFNP=limitador_rango_iyc(VLDD_AFNP,3200/(FlujoMax-
FlujoMin)*RangoMin,3200/(FlujoMax-FlujoMin)*RangoMax);
```

```
x=int32([1:numel(VLDD_AFNP)]);
```

```
figure('Color',[1 1 1]);
```

```
plot(x,VLDD_PP,x,VLDD_AFNP,x,FNP,'LineWidth',1.5,'LineSmoothing','on')
;

title('Límites deslizantes de VLDD-AFNP a partir de VLDD-PP');
xlabel('Tiempo en muestras (100 por segundo)');
legend('VLDD-PP','VLDD-AFNP','FNP')

salida.x=x;
salida.VLDD_AFNP=VLDD_AFNP;
```

7.4.10 function salida=PP2TSAT_iyc(VLDD_PP)

```
% PP2TSAT convierte el valor de presión limitado en tasa a temperatura
% de saturacion
```

```
pp2tsat_x=[0 25 98 1157 1969 3015 3200];
pp2tsat_y=[0 970 1455 2521 2861 3161 3200];
input=[1:3200];
PP2TSAT=interp1(pp2tsat_x,pp2tsat_y,input);

salida=convertir_iyc(VLDD_PP,PP2TSAT);
```

7.4.11 function salida=VLDD_FNP_iyc(FNP)

```
% VLDD_FNP es una función que genera el valor de flujo limitado
% en tasa y rango VLDD_FNP
```

```
FlujoMax=130;%Flujo maximo que puede medir el SIN
FlujoMin=0;%Flujo minimo que puede medir el SIN

RangoMax=100;%Valor maximo que puede tomar el límite
RangoMin=3;%Valor minimo que puede tomar el límite
TasaCrec=5;%Maxima velocidad de crecimiento del límite (por minuto)
TasaDecrec=15;%Minimo velocidad de crecimiento del límite (por minuto)

FNP_dig=digitalizar_iyc(FNP,FlujoMax,FlujoMin,1,100);

FNP_limran= limitador_rango_iyc(FNP_dig,3200/(FlujoMax-
FlujoMin)*RangoMin,3200/(FlujoMax-FlujoMin)*RangoMax);

VLDD_FNP= limitador_tasa_iyc(FNP_limran,3200/(FlujoMax-
FlujoMin)*TasaCrec*60,3200/(FlujoMax-FlujoMin)*TasaDecrec*60,100);

x=int32([1:numel(VLDD_FNP)]);
figure('Color',[1 1 1]);

plot(x,FNP_dig,x,VLDD_FNP,'LineWidth',1.5,'LineSmoothing','on');

title('Límites deslizantes de VLDD-FNP');
xlabel('Tiempo en muestras (100 por segundo)');
legend('FNP','VLDD-FNP')
```

```
salida.x=x;
salida.VLDD_FNP=VLDD_FNP;
salida.FNP_dig=FNP_dig;
```

7.4.12 function salida=VLDD_TEN_iyc(VLDD_PP,VLDD_FNP)

VLDD_TEN_IYC genera a partir del VLDD_PP y VLDD_FNP todas las señales necesarias para generar VLDD_ATEN y VLDD_BTEN.

```
TempMax=330;
TempMin=0;

DeltaAten=17;
RangoAtenMax=323;
RangoAtenMin=277;

DeltaBten=3.4;
RangoBtenMax=302.6;
RangoBtenMin=256.6;

DTGV=FNP2DTGV_iyc(VLDD_FNP);
TSAT=PP2TSAT_iyc(VLDD_PP);

Resta=TSAT-DTGV;

ATEN=sumador_iyc(Resta,3200/(TempMax-TempMin)*DeltaAten);
ATEN=limitador_rango_iyc(ATEN,3200/(TempMax-
TempMin)*RangoAtenMin,3200/(TempMax-TempMin)*RangoAtenMax);

BTEN=restador_iyc(Resta,3200/(TempMax-TempMin)*DeltaBten);
BTEN=limitador_rango_iyc(BTEN,3200/(TempMax-
TempMin)*RangoBtenMin,3200/(TempMax-TempMin)*RangoBtenMax);

salida.ATEN=ATEN;
salida.BTEN=BTEN;
x=int32([1: numel(VLDD_FNP)]);
salida.x=x;

figure('Color',[1 1 1]);
plot(x,VLDD_FNP,x,VLDD_PP,x,ATEN,x,BTEN,'LineWidth',1.5,'LineSmoothing',
'on');

title('Límites deslizantes VLDD-ATEN y VLDD-BTEN');
xlabel('Tiempo en muestras (100 por segundo)');
legend('VLDD-FNP','VLDD-PP','ATEN','BTEN');
```