

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

**DESARROLLO DE UNA LIBRERÍA NEURONAL EN CONTROL
BUILDER 5.1 PARA CONTROLAR EL NIVEL DE MINERAL AL
INTERIOR DE UN MOLINO DE BOLAS**

Tesis para optar el Título de Ingeniero Electrónico, que presenta el bachiller:

Giancarlo Stefano Sotelo Cabrera

ASESOR: José Dávalos Pinto

Lima, Mayo del 2016

RESUMEN

Los equipos utilizados en la molienda de minerales son los molinos de bolas, encargados de realizar la conminución del mineral. Estos equipos desarrollan elevado consumo de energía a una baja eficiencia de operación. Una de las razones de la baja eficiencia se debe a la dificultad existente para identificar el nivel de carga en el molino. Por ejemplo, al incrementar el nivel de carga al nivel óptimo, los espacios existentes entre las bolas de acero se rellenan con mineral, generando un incremento en el ratio de conminución. Por otro lado, podría pensarse que superar el nivel de carga óptimo incrementa el ratio de conminución. Sin embargo, solo satura el molino y disminuye aún más la eficiencia del proceso. Estos son algunos ejemplos que indican la importancia de conocer el nivel de carga durante la operación. El nivel de carga marca la pauta de una operación óptima y permite incrementar la eficiencia del proceso de molienda.

El objetivo de la tesis fue desarrollar un sistema de control neuronal capaz de determinar el nivel de carga instantáneo del molino en base a la experiencia del operador de sala de control y mediciones de variables ampliamente conocidas en la operación de molinos: Presión, Velocidad, Impactos y Tonelaje. La principal ventaja del sistema de control desarrollado es la medición en tiempo real de una variable que no es posible determinar directamente con algún sistema de instrumentación en campo. Adicionalmente, si el molino está operando fuera del nivel óptimo de carga, el sistema de control proporciona las acciones típicas correctivas.

El sistema de control neuronal fue desarrollado en las plataformas Control Builder 5.1 y System800xA (Sistema de Control Distribuido) del fabricante ABB. Las señales de campo se integraron al sistema a través de "Cableado Duro" (4...20mA) y "Modbus TCP". Asimismo, se utilizó el algoritmo neuronal supervisado "Retropropagación" a fin de entrenar la red neuronal. Finalmente, se realizaron pruebas al sistema de control y se obtuvo el nivel de carga del molino con un error máximo de 4.22%.

Dedicatoria

A Dios por haberme permitido llegar a este punto. A mis abuelos Violeta Campos y Juan Sotelo. Por haberme apoyado en todo momento a través de sus consejos y valores, por la motivación constante que me ha permitido ser una persona de bien, pero sobre todo por su amor incondicional. A mis hermanas Margarita Sotelo y Gisella Sotelo. Por los ejemplos de perseverancia y constancia que han infundado siempre en mí y por el valor mostrado para salir adelante en momentos difíciles. A mis Padres. Porque los llevo presentes a todo lugar a donde me dirijo y en cada cosa que hago.

INDICE

| | |
|--|----------|
| INTRODUCCIÓN..... | 1 |
| | |
| CAPÍTULO 1: PROBLEMÁTICA DE LOS MOLINOS DE BOLAS EN UNA PLANTA | |
| 1.1 Planteamiento del problema..... | 2 |
| 1.2 Justificación de la solución propuesta..... | 3 |
| 1.3 Delimitación de la solución..... | 4 |
| 1.4 Formulación del problema..... | 5 |
| | |
| CAPÍTULO 2: MARCO TEÓRICO | |
| 2.1 Control del Circuito de Molienda utilizando Redes Neuronales..... | 6 |
| 2.2 Control de Molinos utilizando Procesamiento de Señales de Vibración..... | 8 |
| 2.3 Control de Molinos utilizando monitorización de Motores Eléctricos..... | 9 |
| 2.4 Control de Molinos utilizando Lógica Fuzzy..... | 10 |
| 2.5. Modelo Teórico..... | 11 |
| 2.6. Objetivos..... | 13 |
| | |
| CAPÍTULO 3: DESARROLLO DE LA LIBRERÍA NEURONAL EN CONTROL BUILDER 5.1 | |
| 3.1 Descripción y Justificación del Diseño..... | 14 |
| 3.2 Requerimientos de Diseño..... | 15 |
| 3.2.1 Selección de Hardware..... | 17 |
| 3.2.2 Selección de Software..... | 18 |
| 3.3 Metodología de Diseño..... | 19 |
| 3.3.1 Configuración en Control Builder 5.1..... | 19 |
| 3.3.2 Estructura de la red neuronal..... | 36 |

CAPÍTULO 4: RESULTADOS

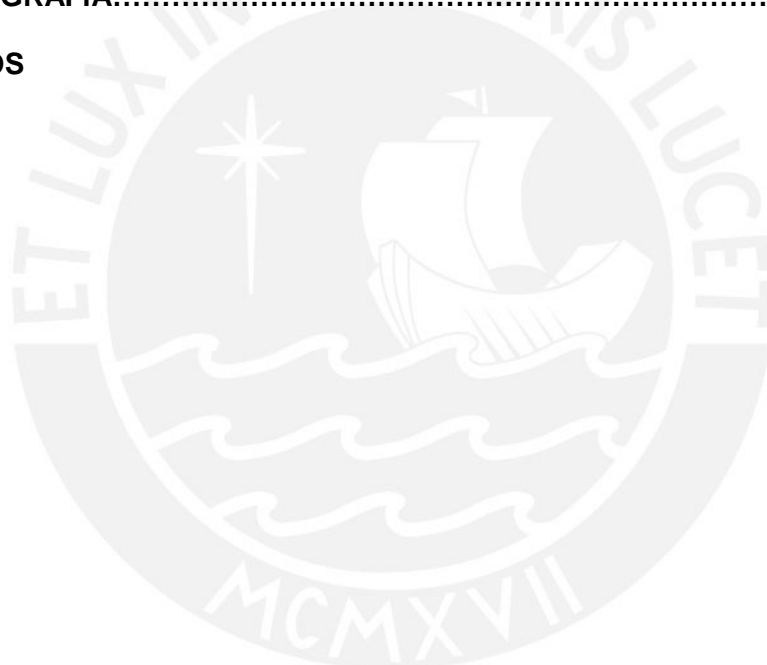
| | |
|--|----|
| 4.1 Entrenamiento de la red neuronal..... | 44 |
| 4.2 Estimación de nivel de carga del molino..... | 47 |
| 4.3 Análisis de resultados..... | 53 |

| | |
|-------------------------|-----------|
| PRESUPUESTO..... | 54 |
|-------------------------|-----------|

| | |
|--------------------------|-----------|
| CONCLUSIONES..... | 55 |
|--------------------------|-----------|

| | |
|-----------------------------|-----------|
| RECOMENDACIONES..... | 56 |
|-----------------------------|-----------|

| | |
|--------------------------|-----------|
| BIBLIOGRAFÍA..... | 57 |
|--------------------------|-----------|

ANEXOS

INTRODUCCIÓN

La Molienda es la operación de mayor consumo energético en el procesamiento de minerales al interior de una Planta Concentradora. La energía consumida asociada al proceso es aproximadamente 40% del costo total de la producción. Sin embargo, solo el 10% es energía efectiva. La baja eficiencia se debe principalmente a que el operador en sala de control desconoce el nivel de carga del molino durante la operación. La eficiencia de un molino depende fundamentalmente de operar en el nivel de carga al cual fue calibrado el equipo, generalmente del 20% al 30% de nivel. Al operar dentro del rango óptimo se reduce la energía consumida, se incrementa la vida útil del equipo y se aumenta la eficiencia de operación. Finalmente, una mínima mejora del performance en el proceso de molienda, genera un gran impacto en los costos de operación de la planta y en la optimización de los recursos energéticos.

La presente tesis tiene como objetivo desarrollar una librería neuronal en Control Builder 5.1 capaz de realizar el control de nivel de carga del molino. La librería neuronal permitirá incrementar la eficiencia del proceso de molienda. El documento se divide en cuatro capítulos. El primer capítulo presenta el planteamiento, justificación y formulación de la problemática relacionada al proceso de molienda. En el segundo capítulo se detalla el estado del arte de investigaciones anteriores realizadas por otros autores y el modelo teórico propuesto de solución. Adicionalmente se plantean los objetivos de la presente tesis. En el tercer capítulo se realiza el desarrollo de la librería neuronal utilizando Control Builder 5.1. Se detalla por completo la configuración del software y se presentan los algoritmos de programación a utilizarse. El último capítulo muestra las simulaciones del control neuronal desarrollado y se analizan los resultados obtenidos. Adicionalmente a los cuatro capítulos desarrollados, se incluyen conclusiones, recomendaciones, bibliografía, presupuesto y anexos.

CAPÍTULO 1

PROBLEMÁTICA DE LOS MOLINOS DE BOLAS EN UNA PLANTA

Hoy en día en el procesamiento de minerales, la operación de molienda es una de las más intensivas en consumo de energía eléctrica. Se estima que el 40% de la energía eléctrica total consumida durante el procesamiento es utilizada en la molienda del mineral. Actualmente no existe un método directo para determinar exactamente el nivel de carga que existe al interior de un molino en un instante de tiempo en particular; por ello, el tiempo de procesamiento es típicamente sobredimensionado. Al sobredimensionar el tiempo de procesamiento, se incrementa drásticamente el consumo energético. De aquí nace la necesidad de plantear una estrategia de control de nivel que permita operar el molino en un nivel de carga óptimo a fin de reducir el consumo eléctrico e incrementar el tiempo de vida útil del equipo.

1.1 Planteamiento del Problema

Actualmente la medición del nivel de carga instantánea en un molino de bolas es un parámetro no disponible para el operador de sala de control debido a múltiples factores. Variaciones en la competencia de mineral, naturaleza de la pulpa e ingreso de agua al chute son algunos de ellos. Dentro de las limitaciones mencionadas, el factor principal es la carencia de un sistema de instrumentación capaz de medir el nivel de carga directamente. Históricamente se han utilizado diversos tipos de señales para diseñar lazos cerrados de control que relacionen el comportamiento de tales variables con un nivel de carga en particular. Las señales más representativas fueron potencia eléctrica (kW), amperaje (A) y voltaje (V) del motor del molino, temperatura de rodamientos (°C), entre otros. A pesar de ello, el performance del control de nivel no resultó en lo esperado debido a la naturaleza multivariable y variante en el tiempo del sistema.

La presente tesis desarrolla una nueva alternativa de control para estimar el nivel de carga del molino en tiempo real. Este enfoque se basa fundamentalmente en la experiencia del operador de sala de control, uso de redes neuronales artificiales y en el análisis de las señales de velocidad, presión, tonelaje e impactos del molino a través del tiempo. El hardware y software utilizado por el sistema de control pertenece a la familia S800 del fabricante ABB.



Figura 1.1 AC800 – Advance Controller ABB

Fuente: <http://www.aotewell.com/wp-content/uploads/2011/12/ABB-AC800M-AOTEWELL.jpg>

1.2 Justificación de la solución propuesta

La cantidad de energía eléctrica utilizada en el procesamiento de minerales se distribuye generalmente de la siguiente forma. El 30 % pertenece a la etapa de chancado primario, 40 % pertenece a la etapa de molienda y el 30% restante pertenece a procesos complementarios. El proceso de mayor consumo eléctrico corresponde a la molienda de mineral; por ende, la industria minera busca continuamente implementar estrategias de control que permitan incrementar la eficiencia y el tiempo de vida útil de estos equipos, a fin de generar ahorro en los costos de producción (M.G.Melero, 2014).

La eficiencia de un molino depende fundamentalmente de operar en el nivel de carga al cual fue calibrado el equipo, generalmente del 20% al 30% de nivel. Por lo tanto, la medición exacta del nivel de carga en el molino es clave para mejorar el performance de la Planta Concentradora.

El control neuronal propuesto permite realizar la ponderación de los siguientes objetivos. La ponderación y/o priorización de estos objetivos conlleva a la operación eficiente y estable del molino.

- Aumento del tonelaje de producción.
- Optimización del uso de energía eléctrica en el molino de bolas.
- Menor variación en la calidad de la fragmentación del mineral.
- Aumento de vida útil del equipo.



Figura 1.2 Molino de Bolas

Fuente: <http://www.johnsonscreens.com/sites/default/files/Dsc00060.jpg>

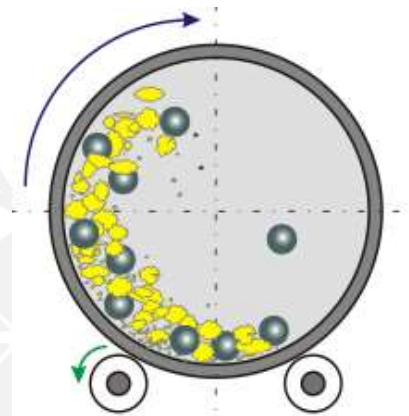


Figura 1.3 Nivel de carga del Molino

Fuente:

http://profesores.elo.utfsm.cl/~agv/elo323/2s06/projects/GerardoAlvarez/proyecto_final_archivos/image009.gif

La Figura 1.2 presenta el molino de bolas utilizado en el estudio. El principio de operación se muestra en la Figura 1.3 en el cual se aprecia que la eficiencia del proceso molienda depende de la cantidad de carga en el molino y de la energía potencial acumulada en cada bola previamente al impacto con la carga.

1.3 Delimitación de la solución

El sistema de control utiliza como punto de partida la instrumentación existente en campo relacionada a las variables utilizadas por el control neuronal. La estimación del nivel de carga se basa en la interrelación de las señales descritas y la experiencia del operador. El nivel de carga estimado permite brindar feedback de operación al operador en sala (aumentar tonelaje, disminuir velocidad, aumentar porcentaje de sólidos, etc.) a fin de proteger el revestimiento del equipo y disminuir el consumo de kW-h.

Señales de campo utilizadas por el control neuronal:

- Presión de Operación del molino (kPa)
- Velocidad del molino (rpm)
- Tonelaje de ingreso al molino (tn/h)
- Impactos sobre el revestimiento del molino (# impactos)

El desarrollo del sistema de control se divide en cuatro partes:

- Selección de Hardware S800 del fabricante ABB.
- Integración de las señales de campo a Control Builder 5.1.
- Desarrollo de la librería neuronal en Control Builder 5.1.
- Desarrollo de la aplicación de control para estimar el nivel de carga del molino en Control Builder 5.1.

1.4 Formulación del problema

¿Cómo la interrelación de estas cuatro variables permite la estimación y control del nivel de carga al interior del molino? Esta interrogante será punto de partida al momento de desarrollar el modelo teórico del control neuronal propuesto.

CAPÍTULO 2

MARCO TEÓRICO

El proceso de molienda es la operación de reducción de tamaño utilizada por la industria minera para segmentar los minerales que serán recuperados en los subsiguientes procesos. La energía consumida por este proceso es aproximadamente 40% del costo total de producción. Sin embargo, solo el 10% de la energía es utilizada de forma efectiva. Adicionalmente, si el tamaño en micras del producto final del circuito de molienda no es el adecuado, afectará las etapas que se encuentren aguas abajo (flotación, filtros, espesadores, etc.), generando que la calidad del concentrado final disminuya. Por ello, la molienda es el proceso clave para incrementar la eficiencia de la Planta Concentradora.

Hoy en día se prioriza la maximización del tonelaje de producción, lo cual ha generado la necesidad de adquirir gradualmente molinos de mayor capacidad de procesamiento. Por ello es necesario un control de alta performance que permita incrementar la eficiencia de estos equipos. A continuación se muestran los antecedentes de trabajos afines a la problemática que han sido realizados previamente por otros autores.

2.1 Control del Circuito de Molienda utilizando Redes Neuronales

El control automático del circuito de molienda es particularmente complejo. Diferentes métodos han sido utilizados para el control automático. Uno de los aportes más simples ha sido utilizar lazos de control de entrada y salida única para cada variable del circuito de molienda, tales como nivel del sumidero, porcentaje de sólidos de alimentación a los hidrociclones, tonelaje de ingreso al molino, perturbaciones, entre otros. Debido a la naturaleza multivariable del proceso es necesaria la interacción de estos lazos de control. Sin embargo, esta técnica no es siempre confiable y puede ser menos eficiente que una técnica multivariable.

Esta investigación (W.STANGE, 2002) presenta el uso de redes neuronales como alternativa de solución multivariable a fin de desarrollar un control eficiente que ajuste automáticamente uno de los parámetros principales del circuito de molienda: Tamaño en micras (um) del producto final. La Figura 2.1 muestra la estrategia de control que el autor propone al circuito de molienda en análisis.

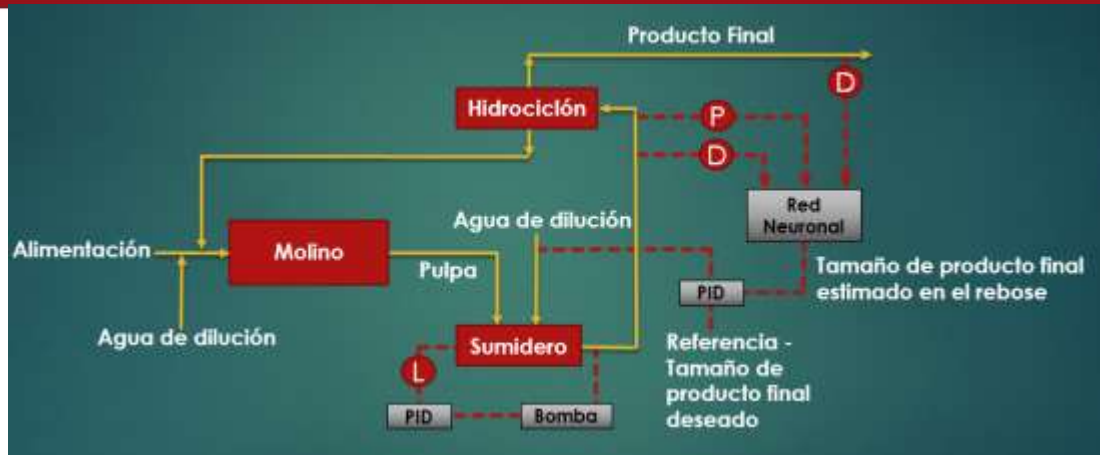


Figura 2.1 Estrategia de Control - Circuito Molienda

Fuente: Using Artificial Neural Networks for the Control of Grinding Circuits (“Uso de redes neuronales artificiales para el control de circuitos de molienda”)

La red neuronal utiliza como entrada las mediciones simples de las siguientes variables: Presión de ingreso al hidrociclón (Sensor “P”) y la densidad de pulpa al ingreso / salida del hidrociclón (Sensores “D”). La Figura 2.2 muestra la estructura de red propuesta. Estas mediciones permiten entrenar la red neuronal a fin de estimar el tamaño de partícula deseado al ingreso del hidrociclón de tal manera que en el producto final se obtenga el 75% de micras pasantes. La estrategia general de control propuesta utiliza la red neuronal para proveer información al sistema de control de una variable que no es físicamente medida en campo, por intermedio de la correlación de variables más sencillas de medir por algún sistema de instrumentación. La mayor ventaja del uso de la red neuronal es que permite estimar el tamaño de partícula del producto final sin necesidad de medir directamente esta variable en tiempo real. A pesar que existen equipos disponibles encargados de realizar tales mediciones, estos son usualmente costosos. Finalmente se integra el modelo neuronal a la estrategia de control de la Figura 2.1.

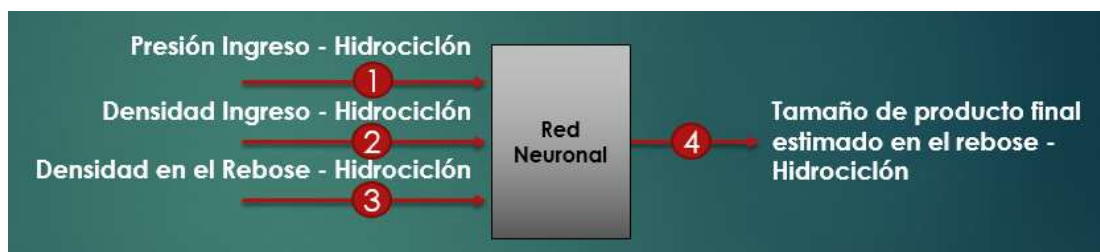


Figura 2.2 Estructura de la Red Neuronal

Fuente: Using Artificial Neural Networks for the Control of Grinding Circuits (“Uso de redes neuronales artificiales para el control de circuitos de molienda”)

El nivel del tanque es controlado por la velocidad de la bomba. Asimismo, la dilución de la pulpa en agua permite controlar el tamaño del producto final estimado por la red neuronal acorde al set point ingresado al PID.

La simulación desarrollada por el autor consideró modelos matemáticos orientados al comportamiento del molino y del hidrociclón bajo condiciones reales de operación, a fin que los resultados obtenidos tengan la mayor exactitud posible. La Figura 2.3 muestra la simulación de la estrategia de control desarrollada.

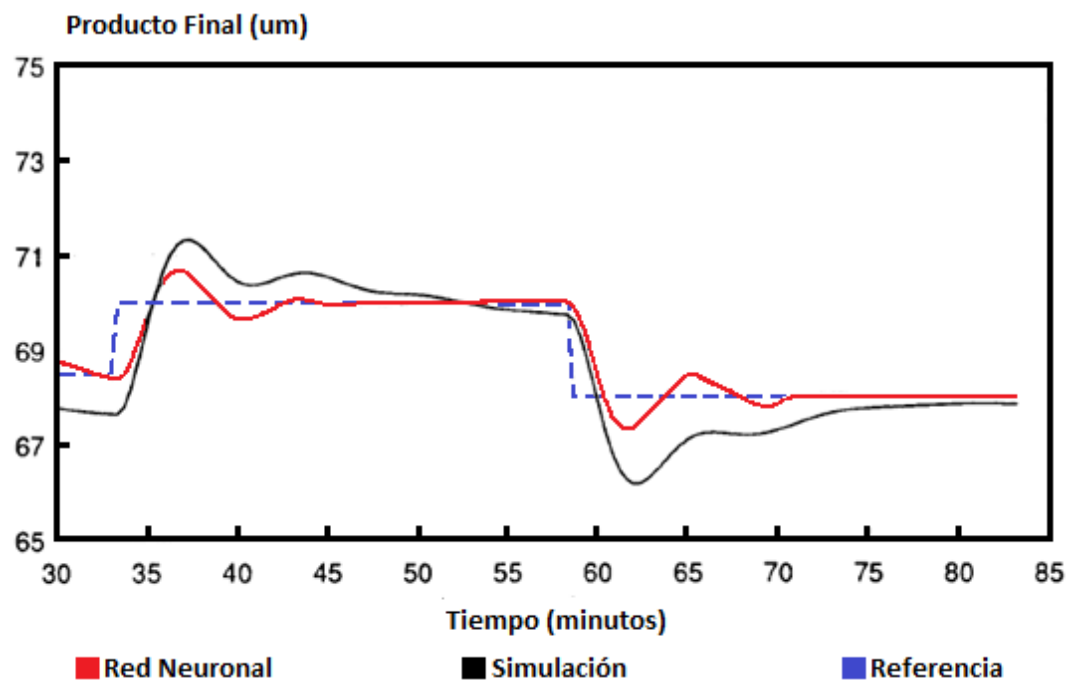


Figura 2.3 Respuesta de la Red Neuronal al cambio de Set Point

Fuente: Using Artificial Neural Networks for the Control of Grinding Circuits (“Uso de redes neuronales artificiales para el control de circuitos de molienda”)

Finalmente, la gráfica muestra los valores estimados por la red neuronal. Notamos que la estimación tiene un porcentaje de aproximación cercano a los cambios de set point deseados para el tamaño del producto final en la molienda.

2.2 Control de Molinos utilizando Procesamiento de Señales de Vibración

Esta investigación (Gugel, 2007) presenta un panorama actualizado del control de nivel en molinos de bolas utilizando procesamiento digital de señales y un acelerómetro montado físicamente sobre el revestimiento del molino, a fin de registrar las vibraciones del equipo. Asimismo realiza la comparación de técnicas clásicas de control versus lazos automáticos, utilizando la vibración para estimar instantáneamente el nivel de mineral presente en el molino. Se muestra mejoras

significativas tanto en el flujo de ingreso de mineral al molino como en la reducción de consumo de kilo – Watt hora por número de toneladas procesadas.

El sistema consta de un transmisor de baja potencia (incluye un conversor analógico – digital) y un acelerómetro. El sensor va instalado sobre el revestimiento del molino, el transmisor se comunica mediante radiofrecuencia con una unidad receptora fija y la fuente de energía proviene de baterías. La Figura 2.4 muestra el esquema general del sistema. La sección transversal se divide en 8 arcos de 45 grados cada uno. Durante la rotación del molino, se estima que las bolas de acero se elevan desde el arco 2 al 4 e impactan sobre los arcos 5 y 6. Determinar el nivel de carga en el molino se enfoca exclusivamente en registrar las vibraciones de impacto entre los arcos 1 y 2, obtener el espectro de frecuencias (kHz) y relacionar cada espectro con un nivel de carga en particular.

Finalmente, la unidad receptora RF posee una salida de 4 – 20 mA que se encarga de enviar la información registrada al sistema de control asociado al proceso. Adicionalmente, esta unidad posee una interfaz RS232, RS485 o Ethernet para realizar la calibración del sistema.



Figura 2.4 Esquema de solución propuesta

Fuente: Automated Control Using Vibration Signal Processing (“Control automático utilizando procesamiento de señales de vibración”)

2.3 Control de Molinos utilizando monitorización de Motores Eléctricos

Hoy en día la monitorización de motores eléctricos es una alternativa que permite controlar la operación de molinos en la gran minería peruana. Asimismo, permite programar mantenimientos predictivos a fin de asegurar mayor vida útil del equipo en cuestión.

Esta investigación (M.G.Melero, 2014) presenta un panorama actualizado relacionado al análisis y control de nivel de carga en molinos monitorizando el drive eléctrico del motor. El control de nivel se realiza monitorizando la potencia eléctrica

instantánea o en su defecto, la corriente eléctrica consumida por el motor eléctrico. Uno de los inconvenientes más resaltantes de esta estrategia de control se debe a que se tiene el mismo consumo de energía cuando el molino está cargado óptimamente que cuando este se encuentra con un nivel de carga muy bajo, lo cual puede llevarnos a lecturas incorrectas del nivel de carga.

El funcionamiento se describe a continuación. El accionamiento del molino es por intermedio de un motor eléctrico y engranajes de reducción (torque). Debido a que el proceso de molienda involucra gran cantidad de masa alrededor de un eje axial de rotación, el drive del motor eléctrico proporciona torques pulsantes vinculados al diagrama de espectros de la corriente, tal como se observa en la Figura 2.5. El método utiliza los armónicos generados para relacionarlos con un nivel de carga en particular.

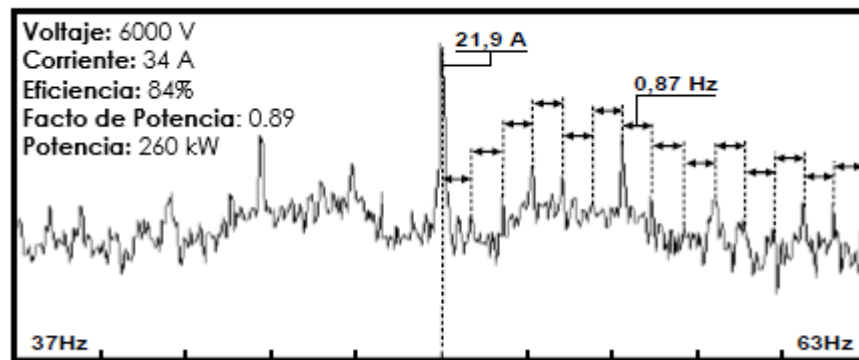


Figura 2.5 Espectro de frecuencia de corriente eléctrica consumida

Fuente: Electric Motors Monitoring (“Monitorización de Motores Eléctricos”)

2.4 Control de Molinos utilizando lógica Fuzzy

La investigación (Cao, 2008) se enfoca en implementar una estrategia de control utilizando lógica Fuzzy. Los controladores clásicos tipo PIDs no son efectivos para este tipo de procesos debido a factores fluctuantes a tomarse en consideración (retardo, naturaleza del material, velocidad del molino, perturbaciones, etc.). Recientemente han surgido nuevos métodos basados en controladores predictivos. Sin embargo, estos métodos de control dependen principalmente del modelo matemático del circuito de molienda, modelos demasiados complejos para implementar en un controlador.

El control Fuzzy no necesita modelos matemáticos. Un controlador Fuzzy está fundamentalmente basado en reglas. Las reglas de control a utilizarse están

relacionadas al conocimiento de expertos en la operación de control de molinos. Mientras más elementos posea el conjunto Fuzzy (es decir, mayor número de reglas), mayor precisión existirá en la estrategia de control propuesta. La salida final del controlador es calculada a través de la interpolación y observación de coeficientes de peso asociados al algoritmo de control Fuzzy.

2.5. Modelo Teórico

La Tesis presenta el desarrollo de una librería neuronal utilizando Control Builder 5.1 del fabricante ABB. La estrategia de control utiliza los impactos registrados en el revestimiento del molino, tonelaje de ingreso, velocidad de giro y presión de operación del molino para determinar el nivel de carga. Dependiendo del nivel estimado el operador puede ajustar los parámetros de molienda (ingreso de tonelaje de alimentación hacia el molino, disminución de velocidad del molino, mayor ingreso de agua por el chute de alimentación, mayor o menor cantidad de sólidos en el molino) a fin de operar eficientemente el molino y aumentar la vida útil del equipo.

Los principales beneficios son los siguientes:

- Aumento de producción de la planta concentradora debido al incremento de tonelaje de alimentación hacia el Molino.
- Los tamaños obtenidos por la molienda son cercanos a los deseados y por ende se reduce el tiempo innecesario de reprocesamiento de mineral.
- Se reduce el tiempo de parada del molino por elevadas presiones de operación.
- Se disminuye la frecuencia de los mantenimientos preventivos debido a la operación eficiente del equipo.
- El sistema de control es 100% digital.
- Se obtiene la medición de una variable (nivel de carga) que no es posible determinar con instrumentación directa debido a la naturaleza del proceso.
- El operador tiene retroalimentación de operación por parte del sistema de control. Es decir, el sistema le indica que variable en particular debe modificar a fin de estabilizar el molino en un punto óptimo.

Dentro de las principales características del sistema tenemos las siguientes:

- Se puede estimar instantáneamente el nivel de carga del molino.
- Las variables de “Presión”, “Velocidad”, “Impactos” y “Tonelaje” se integran al sistema de control a través de Cableado Duro (4...20mA) y MODBUS TCP.

- Es posible estimar una variable de difícil medición en base a mediciones simples de variables conocidas en la operación.
- El entrenamiento del sistema de control neuronal se basa en el ingreso de la data grabada de estas variables durante un día, 1 semana o un mes de operación, a fin de brindarle al sistema la mayor cantidad de información disponible. Basado en el análisis de estas variables y el conocimiento empírico de los operadores de sala de control de molinos (expertos en la operación) se estima el nivel de carga presente en el molino.

El principio de operación se basa en estudiar la interacción de estas cuatro variables. Los operadores de sala de control pueden predecir empíricamente el nivel de carga del molino basándose en el análisis de los históricos de estas variables y en sus años de experiencia como operadores de molinos. Por ejemplo:

- Nivel de impactos menor a 10 nos indica alto nivel de carga en el molino.
- Presión de operación mayor a 8600 kPa nos indica alto nivel de carga en el molino.
- Presión de operación mayor a 8600 kPa y tonelaje de ingreso mayor a 5000 tn/h nos indica alto nivel de carga en el molino.
- Presión de operación mayor a 8600 kPa y velocidad mayor a 8.8 rpm nos indica alto nivel de carga en el molino.
- Numero de impactos mayores a 20 y velocidad del molino menor a 8 rpm nos indica bajo nivel de carga en el molino.

La red neuronal utiliza como entrada las mediciones simples de las siguientes variables: Presión de operación, velocidad, tonelaje e impactos sobre el revestimiento del molino. La Figura 2.6 muestra las entradas y salidas de la red neuronal propuesta. Estas mediciones permitirán entrenar la red a fin de estimar el nivel de carga del molino. La mayor ventaja del uso de la red neuronal es que permite estimar el nivel sin necesidad de medir directamente esta variable en tiempo real. La estimación del nivel se basa en la data registrada durante un periodo de tiempo (1 hora, 1 día, 1 mes, etc.). Para el presente estudio se presenta la data en el Anexo 1.



Figura 2.6 Entrada/Salida Red Neuronal

Fuente: Elaboración propia

Finalmente, un incremento de la eficiencia del proceso puede significar un impacto de gran magnitud en la disminución de costos operativos de la planta, así como en la conservación y optimización de recursos asignados.

2.6. Objetivos

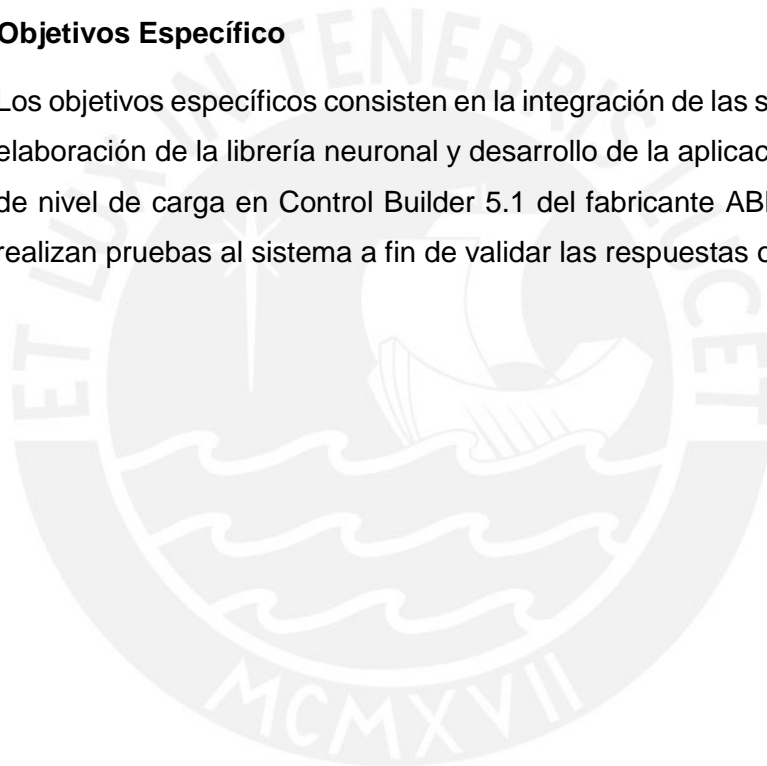
Los objetivos generales y específicos del presente estudio de tesis son:

- **Objetivos General**

El objetivo general de la tesis es desarrollar una librería neuronal en Control Builder 5.1 para controlar el nivel de carga al interior de un Molino.

- **Objetivos Específico**

Los objetivos específicos consisten en la integración de las señales de campo, elaboración de la librería neuronal y desarrollo de la aplicación de estimación de nivel de carga en Control Builder 5.1 del fabricante ABB. Finalmente, se realizan pruebas al sistema a fin de validar las respuestas obtenidas.



CAPÍTULO 3

DESARROLLO DE LA LIBRERÍA NEURONAL EN CONTROL BUILDER 5.1

Se describen los requerimientos de diseño y justificación de los mismos. Se hará uso de diagramas de bloques a fin de ver cada subproceso del sistema. Finalmente se implementa la librería de control neuronal utilizando Control Builder 5.1 del fabricante ABB.

3.1 Descripción y Justificación del Diseño

El diagrama de flujo de la Figura 3.1 muestra el diseño del sistema de estimación con cada subproceso.

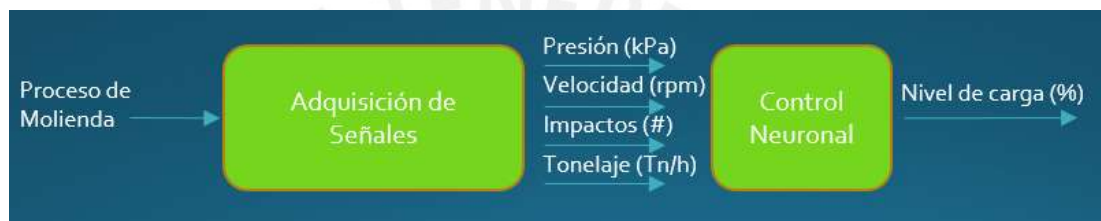


Figura 3.1 Diseño del sistema de estimación

Fuente: Elaboración propia

- **Adquisición de señales:** Las señales de campo que se integran al sistema de control son: presión, velocidad, impactos y tonelaje del molino. Se selecciona hardware e instrumentación pertinente. Las señales se integran al sistema a través de Cableado Duro y MODBUS TCP.
- **Control Neuronal:** Las entradas de la red son las señales de presión, velocidad, impactos y tonelaje del molino. La salida de la red es la estimación del nivel de carga.
- **Presión de Operación:** Este parámetro indica dos pautas. La competencia del mineral y el nivel de carga del molino. Presiones elevadas indican niveles de carga elevados en el molino. Para una estimación exacta se analiza esta variable en conjunto.
- **Velocidad de Operación:** Este parámetro permite controlar el nivel de carga en el molino. Si aumenta la velocidad del molino, se disminuye la carga del molino. Si disminuye la velocidad del molino, se aumenta la carga del molino. Para una estimación exacta es necesario analizar esta variable en conjunto.
- **Nivel de impactos:** El nivel de impactos registrados indica indirectamente el nivel de carga. Mayor número de impactos indica carga baja en el molino (las bolas

impactan mayormente el revestimiento del molino). Menor números de impactos indica alto nivel de carga presente en el molino (existe menor impacto debido a que la bolas colisionan en mayor medida sobre el "riñón" de mineral formado). El impactómetro se calibra con un nivel de carga en particular. Una vez calibrado el instrumento esta es la variable que nos brinda mayor información respecto al nivel de carga.

- **Tonelaje:** Este parámetro ayuda a controlar directamente el nivel de carga del molino. Es el último parámetro que se modifica en la operación debido a que afecta directamente la producción. Para una estimación exacta es necesario analizar esta variable en conjunto.

El principio de operación se basa en la interacción de las cuatro variables de ingreso a la red neuronal. Se puede predecir empíricamente el nivel de carga del molino analizando las tendencias de estas variables a través del tiempo. La estimación depende fundamentalmente de la experiencia del operador. Dependiendo del nivel estimado, se pueden ajustar los parámetros principales de molienda (tonelaje, velocidad, ingreso de agua por el chute de alimentación, porcentaje de sólidos, etc.) a fin de mantener al molino en la zona óptima de operación.

Nota: Consultar Anexo 2 para mayor detalle de la interacción de estas variables para estimar el nivel de carga del molino.

3.2 Requerimientos de Diseño

Una vez definido el sistema es necesario enfocarse en los resultados deseados para cada subproceso. El sistema de control neuronal utiliza como entrada la instrumentación asignada a las variables de presión, velocidad, tonelaje e impactos del molino. Dichas variables se integran al sistema a través de módulos de entradas, seleccionados acorde al tipo de variable (discreta, analógica), protocolo industrial de comunicación y gusto del usuario. Dentro de los fabricantes más destacados tenemos a Siemens, ABB y Rockwell. En cuestiones de hardware la totalidad de marcas ofrecen compatibilidad de equipos entre sí. Por ende, la selección de la marca depende del software y características que pueda ofrecer cada fabricante.

El presente estudio se desarrolla en la plataforma Control Builder 5.1 – System800xA del fabricante ABB. A diferencia de Rockwell y Siemens, presenta un sistema de control distribuido capaz de unificar la totalidad de configuración de servicios en una misma plataforma. Por ejemplo, la configuración de la interfaz de comunicación de dispositivos Rockwell se realiza en la plataforma "RSLinx". Por otro lado, si el mismo

usuario desea configurar los servicios de tendencias de históricos y programación de lógicas de control, debe utilizar la plataforma “RSLogix”. Típicamente en procesos de gran minería se manejan de 14 a 15 controladores, y si la configuración de cada controlador implica el manejo de más de una plataforma de software, puede resultar tedioso para el programador. El System800xA (DCS) del fabricante ABB, proporciona una interfaz amigable, ordenada y unificada para desarrollar aplicaciones de control, configurar servicios y realizar diagnósticos de fallas de los equipos. Asimismo ha sido diseñado para facilitar el ingreso de información unificada de todas las áreas del sistema de automatización, desde los dispositivos de campo hasta la interfaz de optimización de recursos, asegurando información compacta y precisa en tiempo real, lo cual se ve traducido en confiabilidad y un mejor performance de la planta. Por tal motivo el hardware y software seleccionado pertenecen a la gama ABB.

La integración de señales de campo al System800xA se realiza a través de un protocolo de comunicación previamente definido por el sistema de instrumentación asociado a cada variable. La Tabla 3.1 detalla las características de las señales de campo que existen actualmente en la planta concentradora.

| Señal | Descripción | Tipo | Integración al Sistema |
|---------|-------------------------------------|------|------------------------|
| _PIT001 | Presión de Operación - Molino SAG | AI | Cableado duro |
| _SIT001 | Velocidad de Operación - Molino SAG | AI | Cableado duro |
| _IIT001 | Impactos - Molino SAG | AI | MODBUS TCP |
| _WIT001 | Tonelaje de ingreso - Molino SAG | AI | Cableado duro |

Tabla 3.1 Señales de campo (*)

Fuente: Elaboración propia

(*) **Nota:** "AI" representa una entrada analógica. "Cableado duro" hace referencia a una señal que tiene un rango de 4...20 mA.

El performance del sistema depende del tipo controlador seleccionado. En procesos de gran minería generalmente cada controlador puede llegar a manejar de 100 a 200 variables tomando muestras cada 3 segundos por cada variable. Por ello es necesario utilizar controladores de alta gama. Finalmente, hoy en día existen diversos algoritmos para redes neuronales. Uno de los más eficientes y reconocidos es el algoritmo supervisado de “Retropropagación”, algoritmo que será utilizado en el desarrollo del sistema de control. A continuación se detalla la selección de hardware y software ABB que se emplearán en el desarrollo del sistema.

3.2.1 Selección de Hardware

Controlador PM866 - AC800M:

Este módulo es el controlador del sistema encargado de procesar la lógica del sistema en tiempo real. Generalmente los controladores en mina están al 90% de su capacidad de procesamiento, debido a la cantidad de señales provenientes de cada proceso minero. Por lo tanto, se necesita un controlador de alto performance. Dentro de la gama de controladores AC800M, el PM866 pertenece a la clase superior. Se elige el controlador PM866 por la RAM Asignada y frecuencia del procesador.

- 64MB RAM
- 133 MHz
- 2 Puertos Ethernet

Interface de Comunicación MODBUS TCP- CI867:

Es el módulo de comunicación encargado de integrar la señal del “impactómetro” al Control Builder 5.1 a través de MODBUS TCP. El controlador PM866 no puede integrar directamente dispositivos MODBUS TCP. Por tal motivo, se necesita una interface de comunicación entre los dispositivos de campo y el controlador. Tal interface es el módulo CI867 para equipos ABB.

Módulo de entrada Analógica AI815 - ABB:

Este módulo tiene capacidad para 8 entradas analógicas configurables en amperaje o voltaje. Estos canales se configuran de 4...20mA. Un sólo módulo AI815 integra las 3 las señales de campo (“presión”, “velocidad” y “tonelaje”) al sistema de estimación.

- 8 Canales configurables de 4...20mA
- 12 Bits de resolución
- Riel de Montaje DIN
- Comunicación HART

Unidad de Terminación de Módulo TU810 - ABB:

Este módulo se utiliza para la conexión del cableado de las señales de campo hacia los módulos I/O (AI815 en nuestro caso). El sistema de control requiere que se integre 3 señales de campo (presión, velocidad y tonelaje) por lo tanto un solo módulo TU810 cumple los requerimientos de diseño.

- Permite la conexión de 16 señales de campo
- Permite la conexión del AI815 al Module Bus
- Riel de Montaje DIN

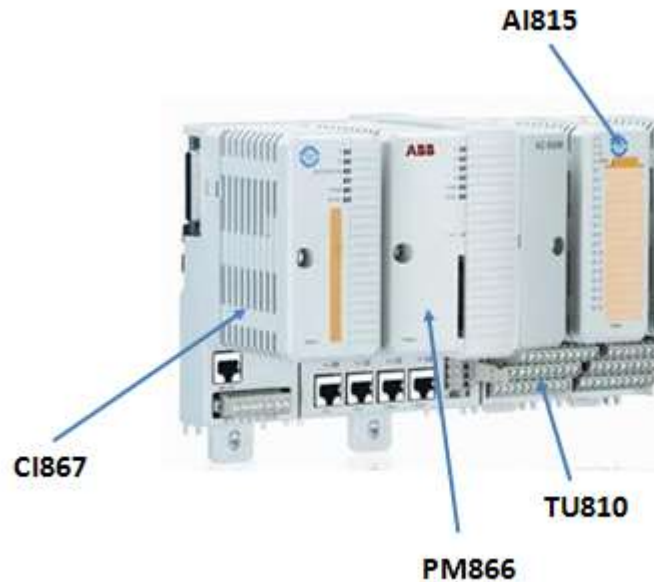


Figura 3.2 Hardware del Sistema

Fuente: Elaboración Propia

3.2.2 Selección de Software

Se utiliza Control Builder 5.1 para la configuración y programación del sistema de control neuronal. Este software provee herramientas para realizar aplicaciones de programación y configuración de hardware para unidades de la familia AC800M (Controlador ABB).

3.3 Metodología de Diseño

En este punto se realiza la implementación del sistema neuronal en Control Builder 5.1. Para revisar en detalle el algoritmo neuronal ver sección 3.3.2. A continuación la secuencia de pasos.

3.3.1 Configuración en Control Builder 5.1.

Paso 1: Se crea el proyecto en Control Builder 5.1.

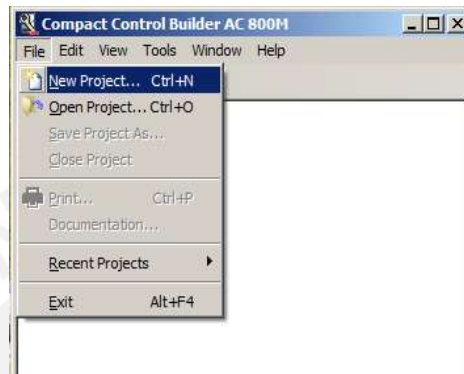


Figura 3.3 Creación del proyecto

Fuente: Elaboración propia

El Proyecto tiene el nombre de "TESIS". Se introduce el nombre y se pulsa "OK".

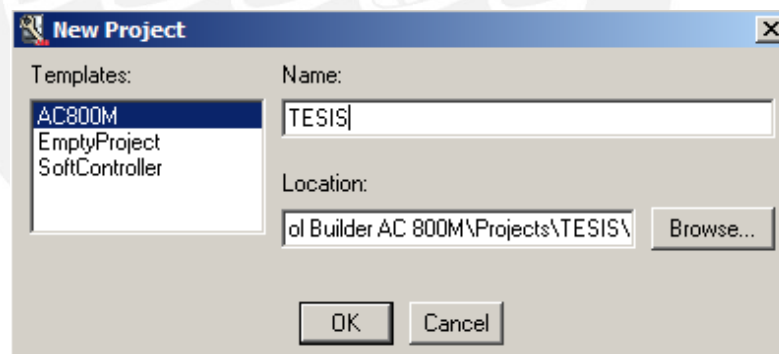
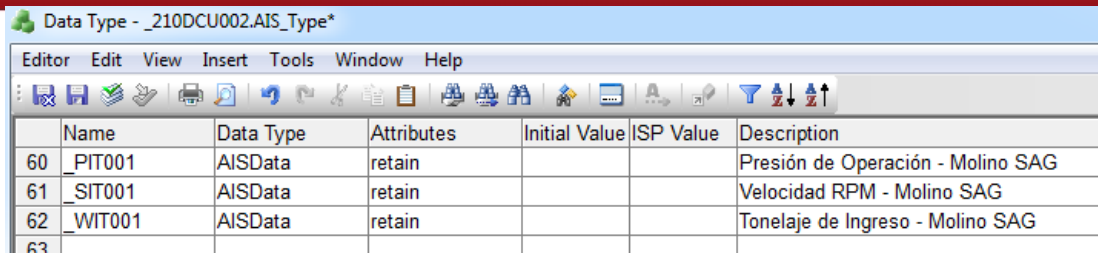


Figura 3.4 Nombre del proyecto

Fuente: Elaboración propia

Paso 2: Integración de señales por Cableado Duro a Control Builder 5.1. Las señales a integrarse son: `_PIT001`, `_SIT001` y `_WIT001` (Revisar Tabla 3.1).

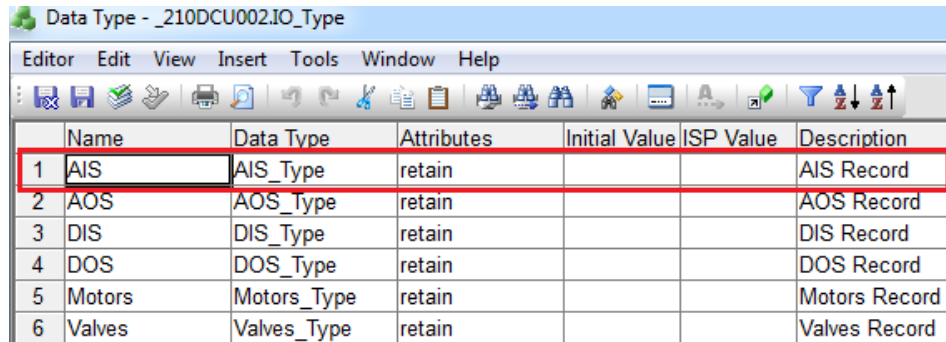
Se definen las variables en Control Builder 5.1. El dato asignado a las variables es del tipo `AISData`, tipo de dato utilizado para integrar señales analógicas al sistema de control. Las señales analógicas se declaran en el directorio "AIS". Asimismo, este directorio es parte de un directorio global denominado `IO_Type`.



| | Name | Data Type | Attributes | Initial Value | ISP Value | Description |
|----|---------|-----------|------------|---------------|-----------|-----------------------------------|
| 60 | _PIT001 | AISData | retain | | | Presión de Operación - Molino SAG |
| 61 | _SIT001 | AISData | retain | | | Velocidad RPM - Molino SAG |
| 62 | _WIT001 | AISData | retain | | | Tonelaje de Ingreso - Molino SAG |
| 63 | | | | | | |

Figura 3.5 Definición de variables

Fuente: Elaboración propia



| | Name | Data Type | Attributes | Initial Value | ISP Value | Description |
|---|--------|-------------|------------|---------------|-----------|---------------|
| 1 | AIS | AIS_Type | retain | | | AIS Record |
| 2 | AOS | AOS_Type | retain | | | AOS Record |
| 3 | DIS | DIS_Type | retain | | | DIS Record |
| 4 | DOS | DOS_Type | retain | | | DOS Record |
| 5 | Motors | Motors_Type | retain | | | Motors Record |
| 6 | Valves | Valves_Type | retain | | | Valves Record |

Figura 3.6 Directorio IO_Type

Fuente: Elaboración propia

Se define el controlador del sistema. El controlador es del tipo PM866 y el nombre es 210DCU002.

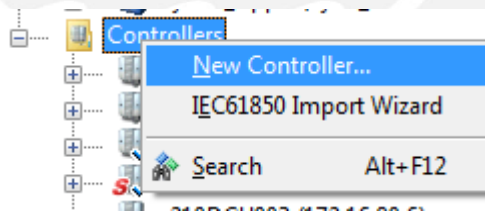


Figura 3.7 Creación del controlador

Fuente: Elaboración propia

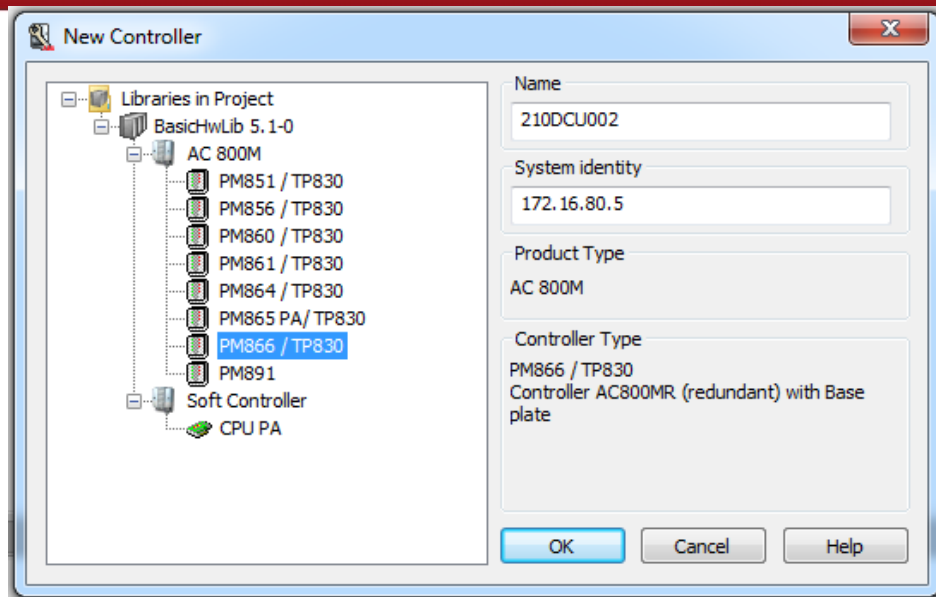


Figura 3.8 Selección del controlador

Fuente: Elaboración propia

Las señales que vienen por cableado duro se integran al sistema a través del "Module Bus" del controlador PM866. En la estructura "Module Bus" se define un módulo AI815. Las señales de campo se conectan a tres de los ocho canales del módulo. Finalmente, se configura los canales analógicos de 4...20mA.

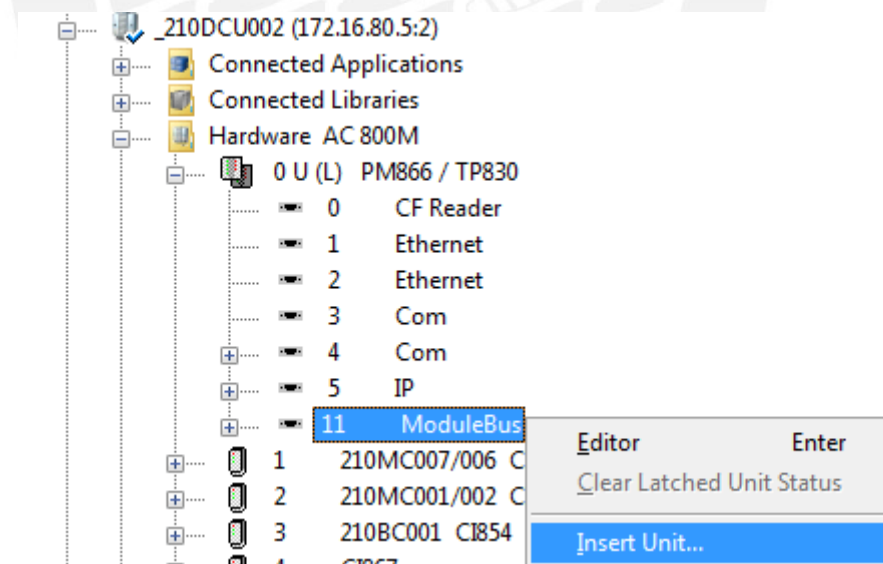


Figura 3.9 Inserción del Módulo de Entrada Analógicas - AI815

Fuente: Elaboración propia

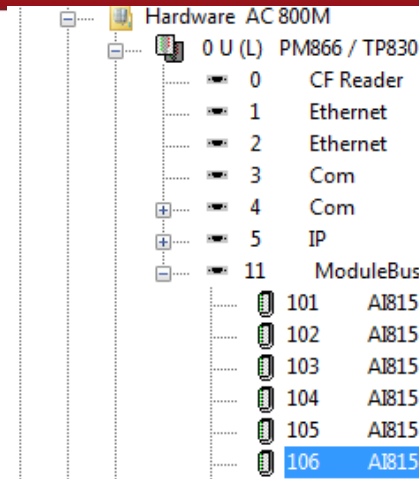


Figura 3.10 Módulo AI815 ubicado en la posición 106

Fuente: Elaboración propia

| Channel | Name | Type | Variable | I/O Description |
|--------------|---------|---------|----------------------------------|-----------------|
| IW0.11.106.1 | Input 1 | ReallIO | _210DCU002.IO.AIS._PIT001.Signal | |
| IW0.11.106.2 | Input 2 | ReallIO | _210DCU002.IO.AIS._SIT001.Signal | |
| IW0.11.106.3 | Input 3 | ReallIO | _210DCU002.IO.AIS._WIT001.Signal | |
| IW0.11.106.4 | Input 4 | ReallIO | | |

Figura 3.11 Asignación de señales de campo al AI815

Fuente: Elaboración propia

| Parameter | Value | Type | Unit | Min | Max |
|------------------------|-------------------------------|------|------|-----|-----|
| Signal range channel 1 | 4..20mA, <2mA signal error | enum | | | |
| Signal range channel 2 | 0..20mA | enum | | | |
| Signal range channel 3 | 4..20mA, <2mA signal error | enum | | | |
| Signal range channel 4 | 4..20mA, <3,8mA signal error | enum | | | |
| Signal range channel 5 | 4..20mA, <3,65mA signal error | enum | | | |
| Signal range channel 6 | 0..5V | enum | | | |
| Signal range channel 7 | 1..5V | enum | | | |
| Signal range channel 7 | 4..20mA sqrt | enum | | | |
| Signal range channel 8 | 4..20mA, <2mA signal error | enum | | | |

Figura 3.12 Configuración de canales a 4...20mA

Fuente: Elaboración propia

Al finalizar este paso, el sistema de control tiene acceso en tiempo real a la data registrada en campo por las señales `_PIT001`, `_SIT001` y `_WIT001`. Es decir, la data ya se encuentra integrada al sistema por cableado duro. Por tanto, el control neuronal puede hacer uso de ella.

Paso 3: Integración de señal de campo por MODBUS TCP. La señal a integrarse es _IIT001 (Revisar Tabla 3.1).

Configuración de Hardware. Se añade el módulo CI867 a la estructura de Hardware del AC800M. El hardware de interfaz de comunicación provee dos redes ethernet (100mbps y 10mbps) sobre las cuales se integran los dispositivos MODBUS TCP. La señal proveniente del impactómetro se integra al sistema a través de la red de 100mbps como un esclavo MODBUS TCP.

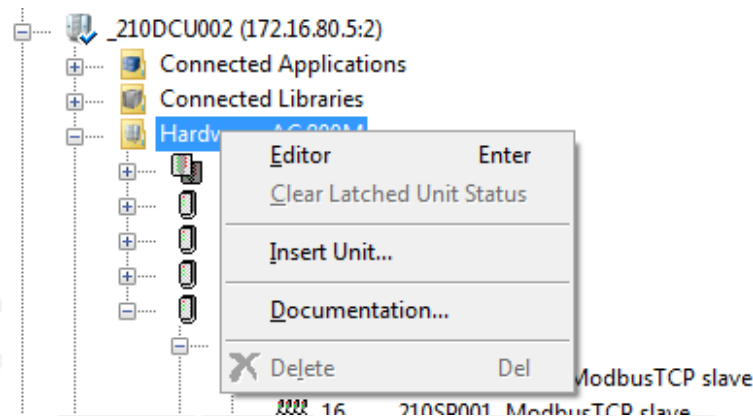


Figura 3.13 Inserción del módulo CI867 - Posición 4

Fuente: Elaboración propia

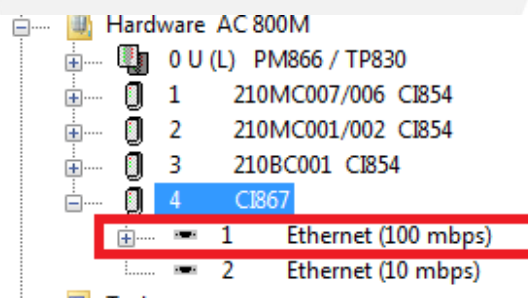


Figura 3.14 Redes Ethernet - CI867

Fuente: Elaboración propia

Una vez añadido el hardware es necesario configurar los parámetros principales del módulo CI867 y de la red ethernet. Se habilita "Modbus slave enabled" en el módulo CI867 y se configura la dirección IP para la red ethernet.

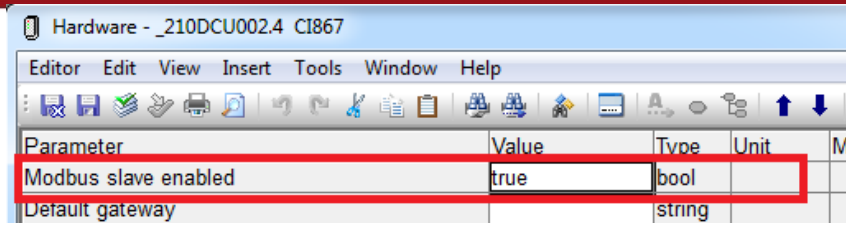


Figura 3.15 Configuración CI867

Fuente: Elaboración propia

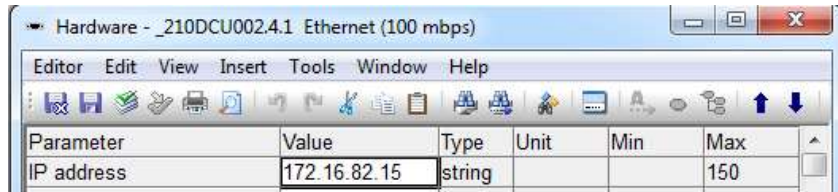


Figura 3.16 Configuración Red Ethernet 100 mbps

Fuente: Elaboración propia

Una vez finalizado la configuración de parámetros del CI867, se integra la señal _IIT001 y se configura su dirección IP.

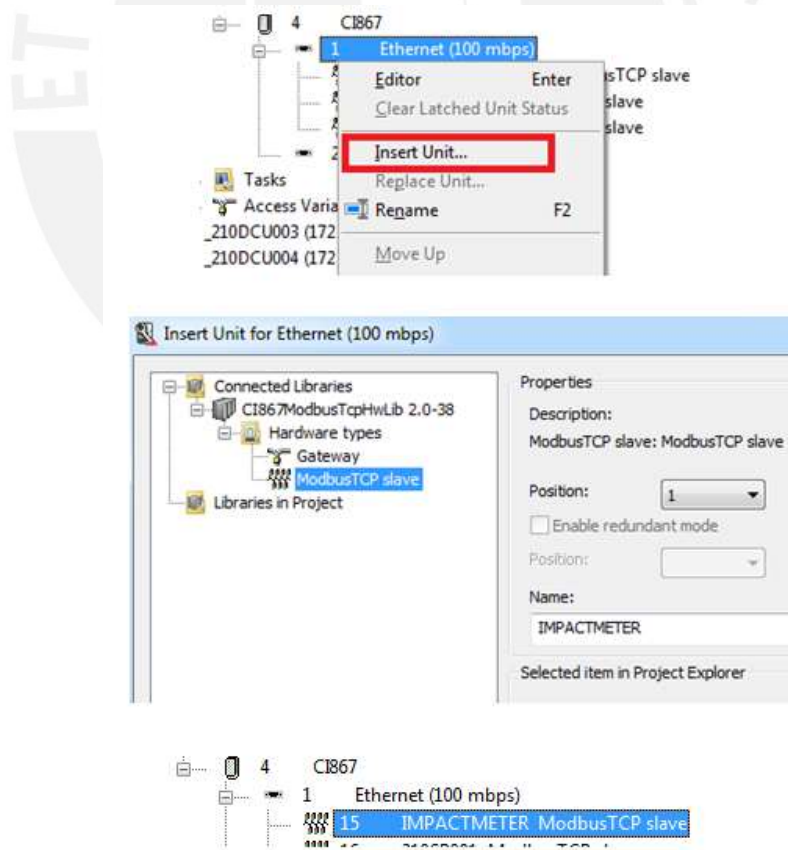


Figura 3.17 Integración – Señal Impactómetro

Fuente: Elaboración propia

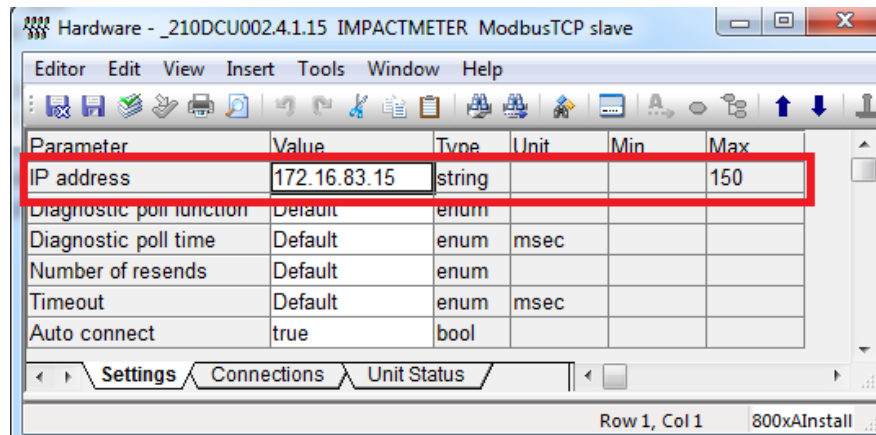


Figura 3.18 Configuración de esclavo MODBUS TCP

Fuente: Elaboración propia

Configuración Software. En este paso se utiliza la librería de control “ModBusTCPCommLib” del fabricante ABB para desarrollar la lógica de comunicación que permita acceder a la data que llega al hardware vía MODBUS TCP. Primero se crea la aplicación 210DCU002_Comm.

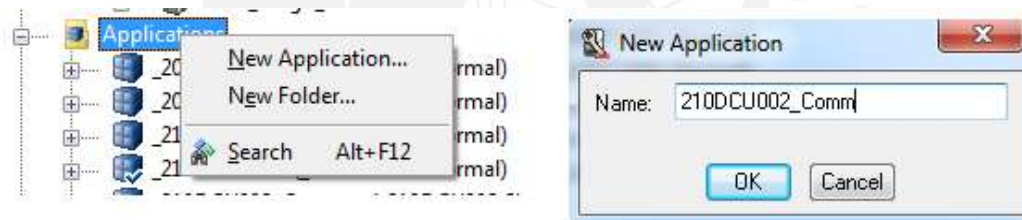


Figura 3.19 Aplicación de Comunicación 210DCU002_Comm

Fuente: Elaboración propia

Una vez creada la aplicación, se define la variable de sistema que almacenará la data del impactómetro en tiempo real. En el directorio AIS_Type se crea la variable "IMPACT_CRITICOS".

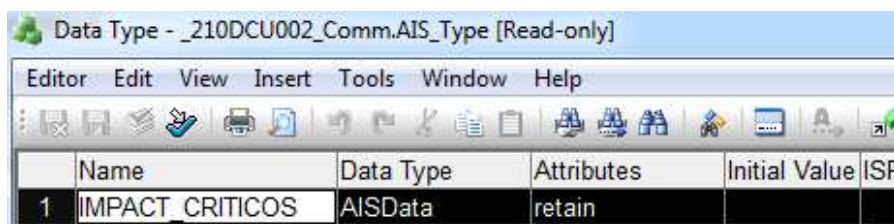


Figura 3.20 Variable de Impactos Críticos

Fuente: Elaboración propia

En este paso se implementa la lógica de comunicación MODBUS TCP. En la estructura "Control Module" se añade un "Single Control Module" de nombre MBTCP. Una vez creado el bloque, click derecho y se selecciona "Editar". Se despliega el editor de programación. La lógica será programada en bloques funcionales.

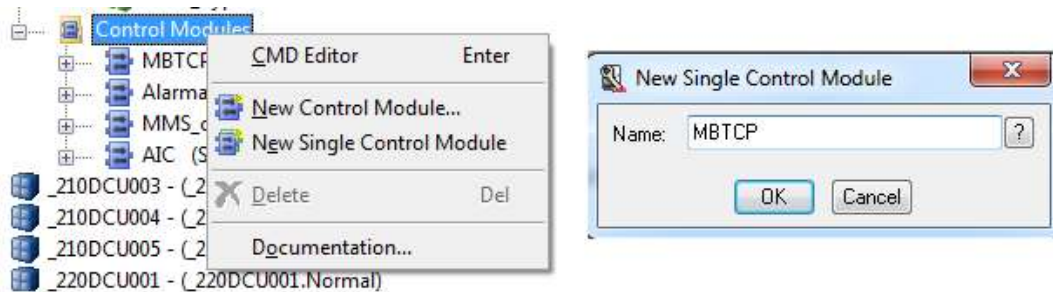


Figura 3.21 Creación del Módulo de Control “MBTCP”

Fuente: Elaboración propia

La comunicación MODBUS se resume en configurar los parámetros de tres bloques funcionales pertenecientes a la librería ModBusTCPCommLib. Estos bloques permiten acceder a la data que viene al hardware vía MODBUS TPC. Los bloques son MBTCPConnect, PulseGenerator y MBTCPRead. Primero, se configura el bloque funcional: "MBTCPConnect".

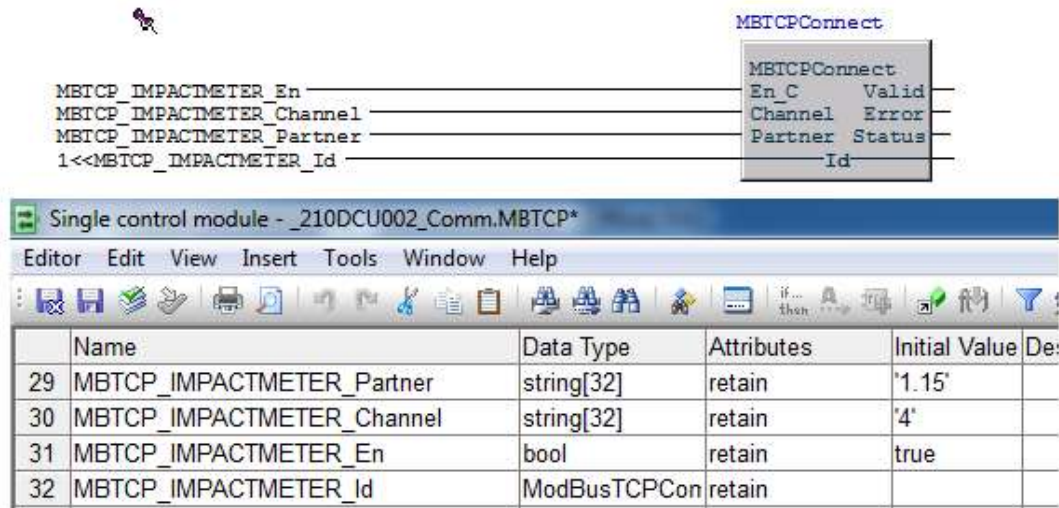


Figura 3.22 Configuración del Módulo de Conexión “MBTCPConnect”

Fuente: Elaboración propia

Configuración del segundo bloque funcional: "PulseGenerator". El parámetro de ingreso "Enable" se conecta al parámetro "Valid" del bloque MBTCPConnect.

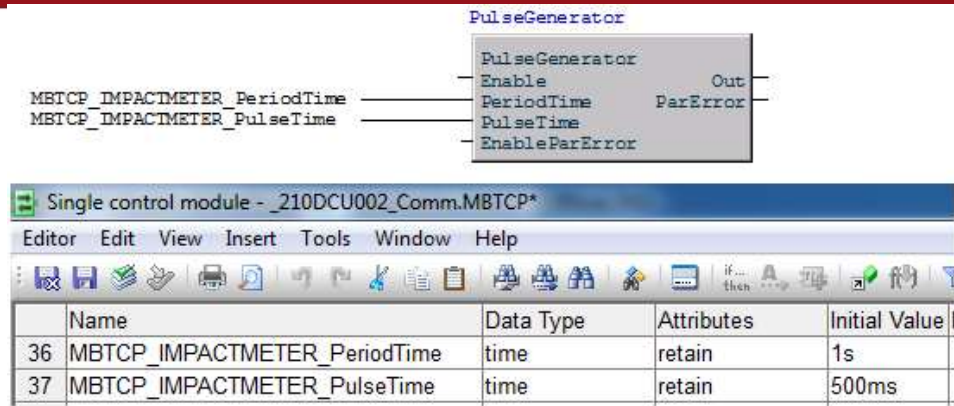


Figura 3.23 Configuración del Generador de Pulso

Fuente: Elaboración propia

Configuración del tercer y último bloque funcional: "MBTCPRead". El parámetro de ingreso "Id" es el mismo que se utilizó para el MBTCPConnect. El parámetro de ingreso "Req" se conecta al parámetro "Out" del bloque "PulseGenerator". El parámetro "StartAddr" se conecta a la variable MBTCP_IMPACTMETER_StartAddrR1. La entrada "Rd[1]" es el parámetro que recibe directamente la data que viene por MODBUS TCP. Este parámetro se conecta a la variable local del Single Control Module: "IMPACT_CRIT". La data registrada en la variable "IMPACT_CRIT" es del tipo doble entero. Por lo tanto se realiza la conversión a tipo de dato "Real" y se almacena en la variable de sistema "IMPACT_CRITICOS". Esta última variable es la que se visualiza en el HMI del sistema.

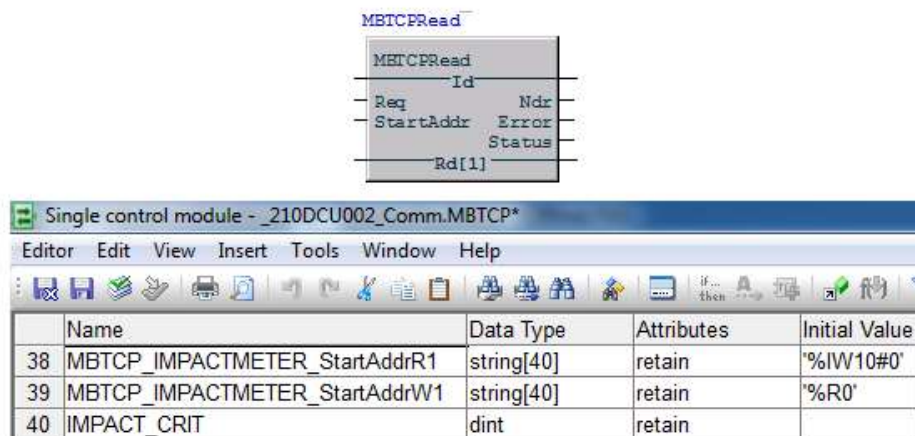


Figura 3.24 Configuración del Módulo de Lectura "MBTCPRead"

Fuente: Elaboración propia

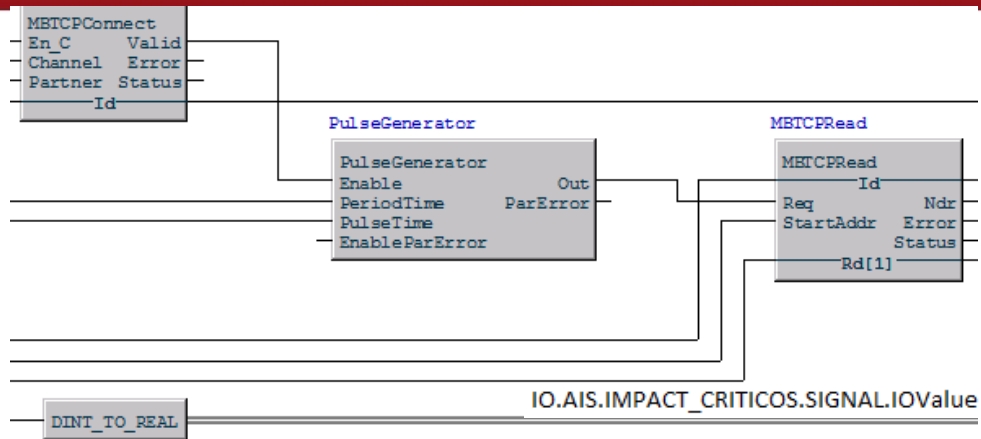


Figura 3.25 Esquema de Comunicación MODBUS TCP

Fuente: Elaboración propia

En este punto se dispone de data en tiempo real de las señales de campo. Esta data ya se encuentra integrada al sistema y se utilizará para implementar la red neuronal.

Finalmente, la siguiente gráfica muestra la lectura en tiempo real de las señales de campo integradas al sistema de control por Cableado Duro y MODBUS TCP.

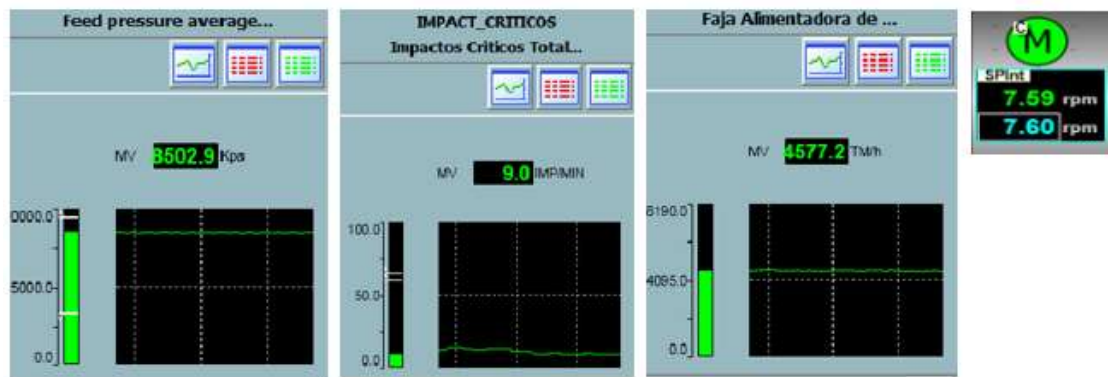


Figura 3.26 Tonelaje de Ingreso - Molino SAG

Fuente: Elaboración propia

Paso 4: Desarrollo de la librería de Control Neuronal.

Se utiliza el algoritmo de entrenamiento supervisado "Retropropagación" para desarrollar la librería neuronal. Las cuatro señales de campo previamente integradas representan las entradas de la red neuronal. La estimación del nivel de carga es la única salida de la red. Se crea la librería "Control_Neuronal" en la estructura "Libraries".

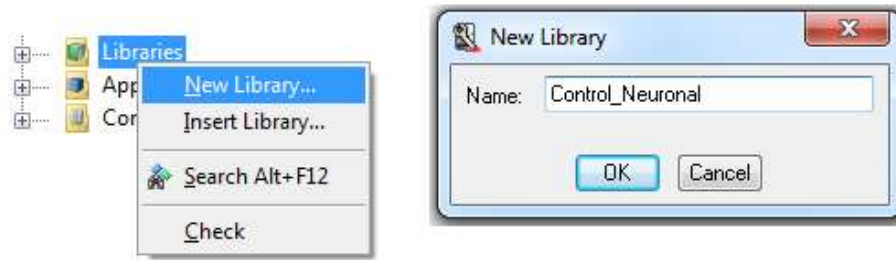


Figura 3.27 Creación de la librería

Fuente: Elaboración propia

La librería neuronal se divide en tres bloques funcionales tal como se muestra en la Figura 3.28. El primer bloque funcional se encarga de enviar los patrones de entrenamiento a la red neuronal. El segundo bloque se encarga de obtener los pesos sinápticos finales de la red neuronal a través del algoritmo de "Retropropagación". El último bloque funcional utiliza los pesos sinápticos obtenidos para estimar el nivel de carga del molino.

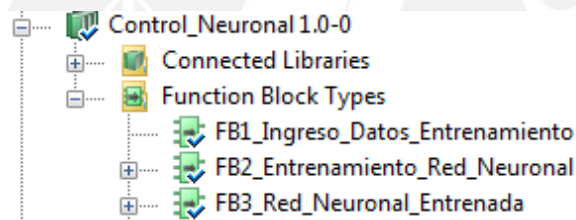
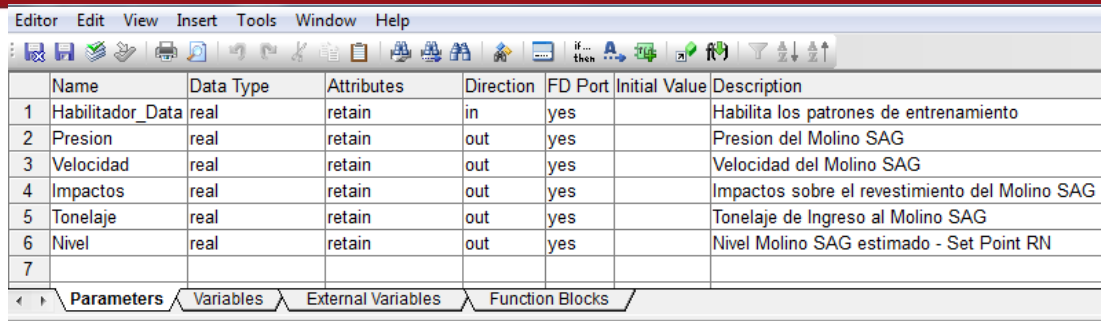


Figura 3.28 Librería de Control - Bloques Funcionales

Fuente: Elaboración propia

Cada uno de los bloques funcionales programados, representan cajas negras que el programador integra en aplicaciones más complejas. Es decir, solo es necesario conectar sus entradas para obtener una salida acorde a la lógica configurada en el bloque funcional.

Una vez definidos los bloques del sistema, se programa cada uno de ellos. Se configura el bloque "FB1_Ingreso_Datos". En la pestaña "Parameters" se definen las entradas y salidas del bloque (entradas y salidas de la caja negra).

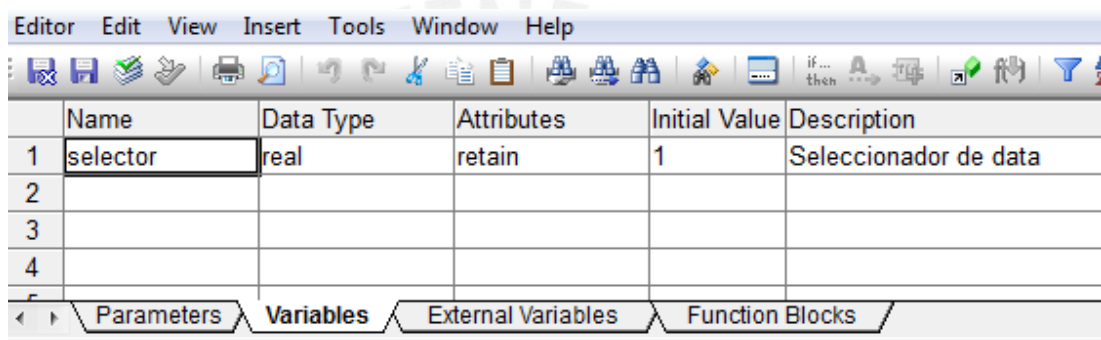


| | Name | Data Type | Attributes | Direction | FD Port | Initial Value | Description |
|---|------------------|-----------|------------|-----------|---------|---------------|--|
| 1 | Habilitador_Data | real | retain | in | yes | | Habilita los patrones de entrenamiento |
| 2 | Presion | real | retain | out | yes | | Presion del Molino SAG |
| 3 | Velocidad | real | retain | out | yes | | Velocidad del Molino SAG |
| 4 | Impactos | real | retain | out | yes | | Impactos sobre el revestimiento del Molino SAG |
| 5 | Tonelaje | real | retain | out | yes | | Tonelaje de Ingreso al Molino SAG |
| 6 | Nivel | real | retain | out | yes | | Nivel Molino SAG estimado - Set Point RN |
| 7 | | | | | | | |

Figura 3.29 Parámetros - FB1_Ingreso_Datos

Fuente: Elaboración propia

En la pestaña "Variables" se definen las variables locales del bloque funcional.



| | Name | Data Type | Attributes | Initial Value | Description |
|---|----------|-----------|------------|---------------|-----------------------|
| 1 | selector | real | retain | 1 | Seleccionador de data |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

Figura 3.30 Variables Locales - FB1_Ingreso_Datos

Fuente: Elaboración propia

En la pestaña "Set_Entrenamiento" implementamos el código fuente asignado al bloque funcional. Revisar sección 3.3.2 para ver el código fuente.

```

(*****
***** DATO 1 *****
*****
If selector = 1 Then
  Presion:=      0.747008496 ;      Velocidad:= 0.9399644 ;
End_if;

(*****
***** DATO 2 *****
*****

```

Figura 3.31 Código Fuente - FB1_Ingreso_Datos

Fuente: Elaboración propia

La Figura 3.32 muestra el bloque funcional programado.

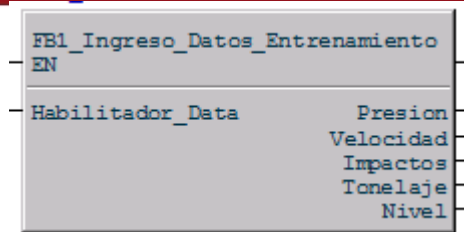
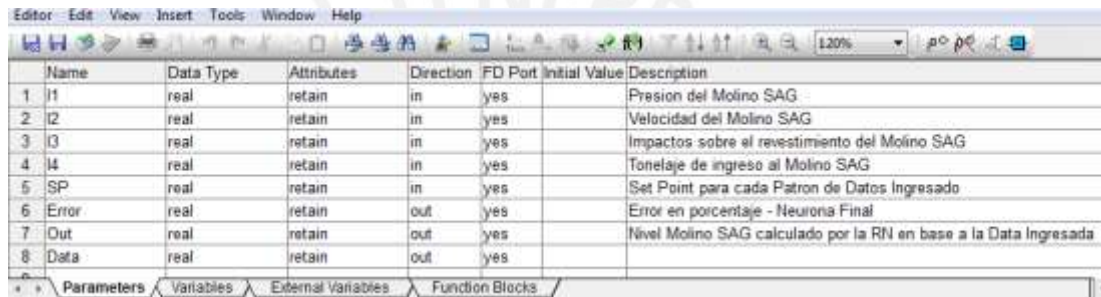


Figura 3.32 Bloque Funcional - FB1_Ingreso_Datos_Entrenamiento

Fuente: Elaboración propia

Se configura el bloque "FB2_Entrenamiento_Red_Neuronal". En la pestaña "Parameters" se definen las entradas y salidas del bloque (entradas y salidas de la caja negra).



| Name | Data Type | Attributes | Direction | FD Port | Initial Value | Description |
|---------|-----------|------------|-----------|---------|---------------|--|
| 1 I1 | real | retain | in | yes | | Presion del Molino SAG |
| 2 I2 | real | retain | in | yes | | Velocidad del Molino SAG |
| 3 I3 | real | retain | in | yes | | Impactos sobre el revestimiento del Molino SAG |
| 4 I4 | real | retain | in | yes | | Tonelaje de ingreso al Molino SAG |
| 5 SP | real | retain | in | yes | | Set Point para cada Patron de Datos Ingresado |
| 6 Error | real | retain | out | yes | | Error en porcentaje - Neurona Final |
| 7 Out | real | retain | out | yes | | Nivel Molino SAG calculado por la RN en base a la Data Ingresada |
| 8 Data | real | retain | out | yes | | |

Figura 3.33 Parámetros - FB2_Entrenamiento_Red_Neuronal

Fuente: Elaboración propia

En la pestaña "Variables" se definen las variables locales del bloque funcional. En la pestaña "Backpropagation" se implementa el código fuente asignado al bloque funcional. Revisar sección 3.3.2 para ver el código fuente.

```

(***)      Perform a forward pass on the network      (***)
(*****)
(***) Assume that neurons have a sigmoid activation function (***)
(*****)

IF step = 1 THEN

    N1_I := I1*W_P1 + I2*W_V1 + I3*W_I1 + I4*W_T1;
    N2_I := I1*W_P2 + I2*W_V2 + I3*W_I2 + I4*W_T2;
    N3_I := I1*W_P3 + I2*W_V3 + I3*W_I3 + I4*W_T3;
    N4_I := I1*W_P4 + I2*W_V4 + I3*W_I4 + I4*W_T4;

END_IF;
    
```

Figura 3.34 Código Fuente - FB2_Entrenamiento_Red_Neuronal

Fuente: Elaboración propia

La Figura 3.35 muestra el bloque funcional programado.

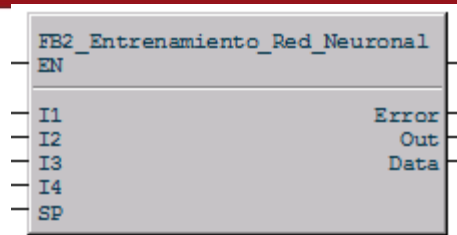
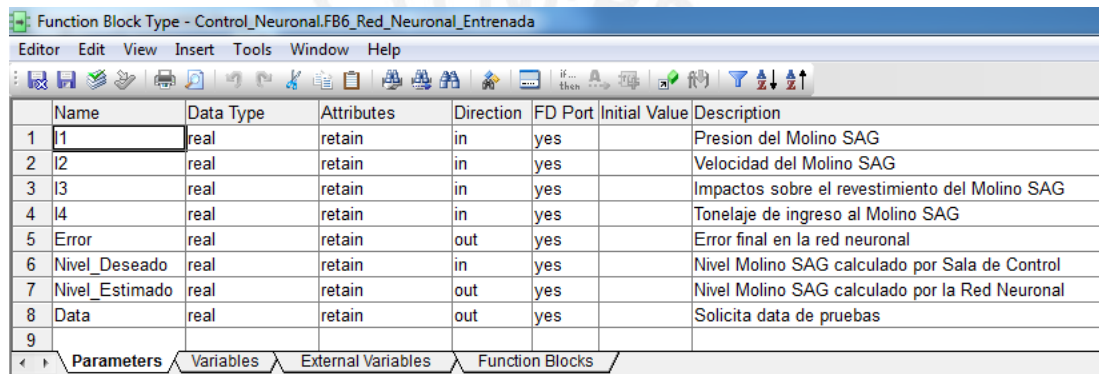


Figura 3.35 Bloque Funcional - FB2_Entrenamiento_Red_Neural

Fuente: Elaboración propia

Por último se configura el bloque "FB3_Red_Neural_Entrenada". En la pestaña "Parameters" se definen las entradas y salidas del bloque (entradas y salidas de la caja negra).



| | Name | Data Type | Attributes | Direction | FD Port | Initial Value | Description |
|---|----------------|-----------|------------|-----------|---------|---------------|--|
| 1 | I1 | real | retain | in | yes | | Presion del Molino SAG |
| 2 | I2 | real | retain | in | yes | | Velocidad del Molino SAG |
| 3 | I3 | real | retain | in | yes | | Impactos sobre el revestimiento del Molino SAG |
| 4 | I4 | real | retain | in | yes | | Tonelaje de ingreso al Molino SAG |
| 5 | Error | real | retain | out | yes | | Error final en la red neuronal |
| 6 | Nivel_Deseado | real | retain | in | yes | | Nivel Molino SAG calculado por Sala de Control |
| 7 | Nivel_Estimado | real | retain | out | yes | | Nivel Molino SAG calculado por la Red Neuronal |
| 8 | Data | real | retain | out | yes | | Solicita data de pruebas |
| 9 | | | | | | | |

Figura 3.36 Parámetros - FB3_Red_Neural_Entrenada

Fuente: Elaboración propia

En la pestaña "Variables" se definen las variables locales del bloque funcional. En la pestaña "ANN" se implementa el código fuente asignado al bloque funcional. Revisar sección 3.3.2 para ver el código fuente.

```

Parameters / Variables / External Variables / Function Blocks /
(*****
*****)
IF step = 1 THEN
    N1_I := I1*W_P1 + I2*W_V1 + I3*W_I1 + I4*W_T1;
    N2_I := I1*W_P2 + I2*W_V2 + I3*W_I2 + I4*W_T2;
    N3_I := I1*W_P3 + I2*W_V3 + I3*W_I3 + I4*W_T3;
    N4_I := I1*W_P4 + I2*W_V4 + I3*W_I4 + I4*W_T4;
END_IF;
(*****
*****)
    
```

Figura 3.37 Código Fuente - FB3_Red_Neural_Entrenada

Fuente: Elaboración propia

La Figura 3.38 muestra el bloque funcional programado.

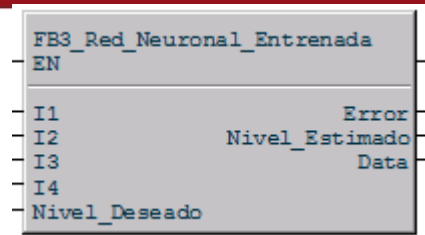


Figura 3.38 Bloque Funcional - FB3_Red_Neuronal_Entrenada

Fuente: Elaboración propia

Paso 5: Desarrollo de la aplicación "_210DCU002" en Control Builder 5.1 para entrenar la red neuronal y estimar el nivel de carga del Molino.

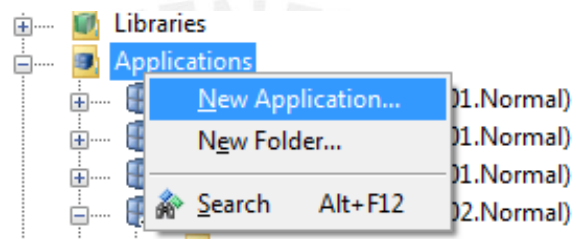


Figura 3.39 Creación de la aplicación "_210DCU002"

Fuente: Elaboración propia

La aplicación creada hará uso de la librería neuronal previamente desarrollada.

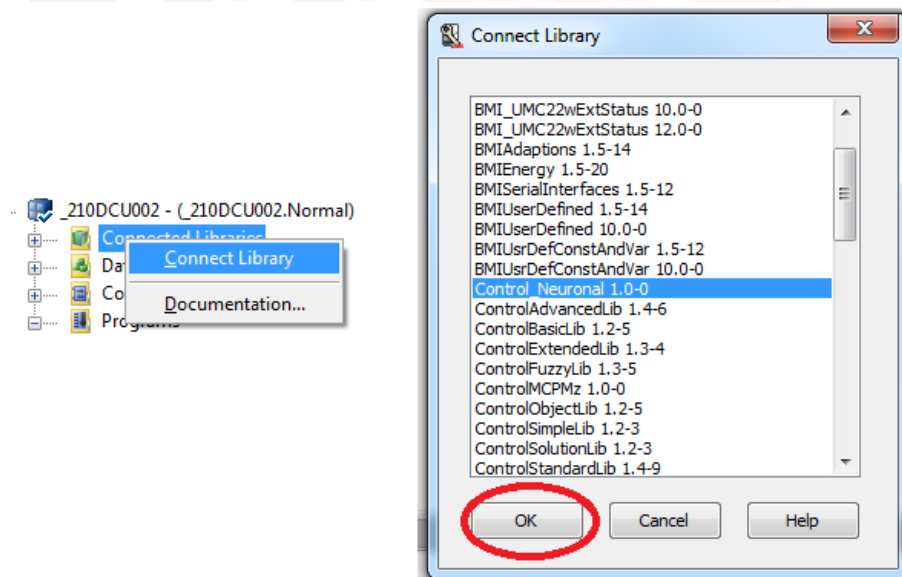


Figura 3.40 Adición de la librería neuronal a la aplicación "_210DCU002"

Fuente: Elaboración propia

Una vez creada la aplicación se elabora el programa encargado de entrenar la red neuronal y estimar el nivel de carga del molino.



Figura 3.41 Programa "Nivel_de_Carga"

Fuente: Elaboración propia

Se ingresa al programa "Nivel_de_Carga" y se define dos instancias de programación: "Entrenamiento" y "Estimacion_Nivel_Carga"

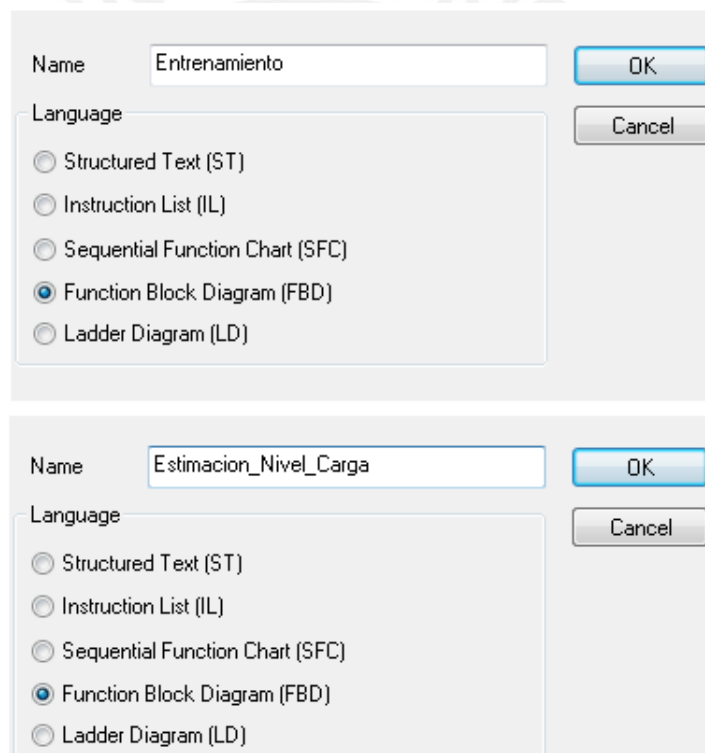
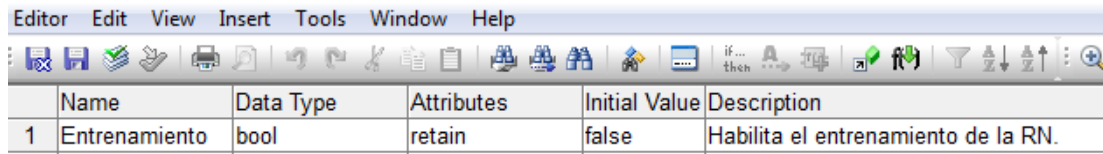


Figura 3.42 Instancias de programación

Fuente: Elaboración propia

Se implementa la lógica de entrenamiento de red utilizando los bloques funcionales "FB1_Ingreso_Datos_Entrenamiento" y "FB2_Entrenamiento_Red_Neuronal" pertenecientes a la librería "Control_Neuronal". En la pestaña "Variables" se definen las variables locales del programa. Esta aplicación permite obtener los pesos sinápticos de la red neuronal una vez que el entrenamiento ha finalizado. El

entrenamiento está completo cuando el error generado por cada patrón de ingreso es mínimo.



| | Name | Data Type | Attributes | Initial Value | Description |
|---|---------------|-----------|------------|---------------|-------------------------------------|
| 1 | Entrenamiento | bool | retain | false | Habilita el entrenamiento de la RN. |

Figura 3.43 Variables Locales - Instancia "Entrenamiento"

Fuente: Elaboración propia

La Figura 3.44 muestra el programa final de entrenamiento. La salida "Error" permite determinar la finalización del entrenamiento. Se analizan los errores registrados a través del tiempo y se detiene el entrenamiento cuando alcanza el valor mínimo.

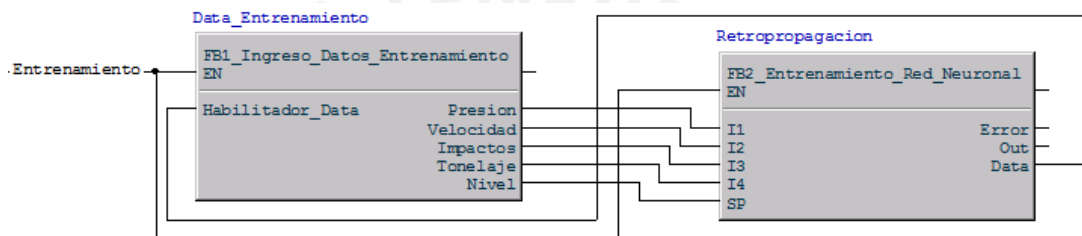


Figura 3.44 Programa Final – Instancia "Entrenamiento"

Fuente: Elaboración propia

Finalmente, se implementa la lógica de estimación de nivel de carga del molino utilizando el bloque funcional "FB3_Red_Neural_Entrenada" perteneciente a la librería "Control_Neural". Esta aplicación permite estimar el nivel de carga analizando la presión, velocidad, impactos y tonelaje del molino. La Figura 3.45 muestra el programa de estimación de nivel de carga en tiempo real.

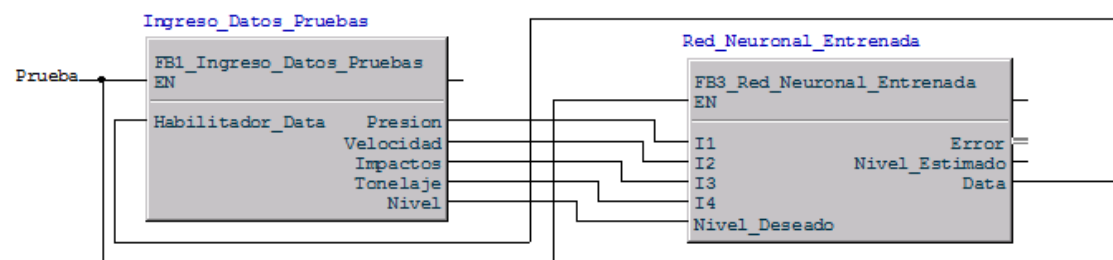


Figura 3.45 Programa – Instancia "Estimación_Nivel_Carga"

Fuente: Elaboración propia

3.3.2 Estructura de la red neuronal y código fuente de los bloques funcionales.

El presente estudio considera la siguiente estructura de red para aplicar el algoritmo de retropropagación.

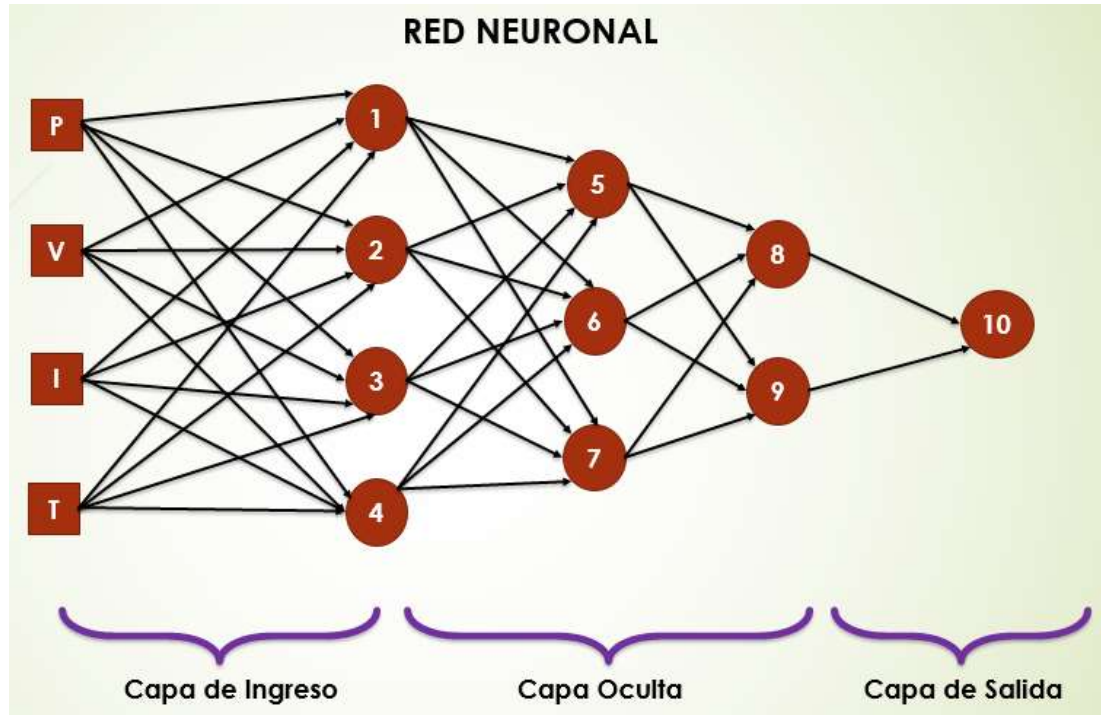


Figura 3.46 Estructura de red

Fuente: Elaboración propia

La red neuronal está conformada por tres capas: “Capa de Ingreso”, “Capa Oculta” y “Capa de Salida”. La capa de ingreso comprende las neuronas del “1” al “4”, la capa oculta comprende las neuronas del “5” al “9” y finalmente la capa de salida posee la neurona “10”. Las entradas de la red neuronal “P”, “V”, “I”, “T” representan las variables de presión, velocidad, impactos y tonelaje del molino y están asociadas a cada una de las señales de campo previamente integradas a Control Builder. La red produce como salida el nivel de carga estimado acorde a los parámetros de ingreso. Finalmente, la nomenclatura utilizada para codificar los pesos sinápticos entre las neuronas es de la forma “ W_{ij} ”, donde que “i” representa la neurona origen y “j” la neurona destino. La figura 3.47 detalla la nomenclatura descrita.

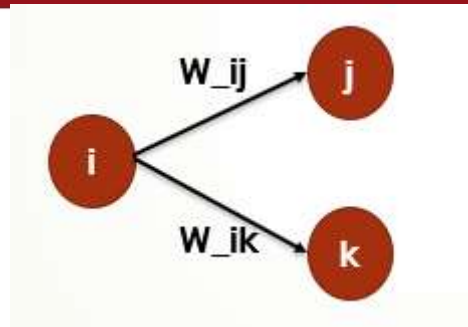


Figura 3.47 Nomenclatura de pesos sinápticos

Fuente: Elaboración propia

I. **CODIGO FUENTE:** FB2_Entrenamiento_Red_Neuronal

```

(*****
***                               Etapa de Propagación                               ***
*****
*** Las neuronas tienen función de activación Sigmoidal ***
*****)

  IF step = 1 THEN

    N1_I := I1*W_P1 + I2*W_V1 + I3*W_I1 + I4*W_T1;
    N2_I := I1*W_P2 + I2*W_V2 + I3*W_I2 + I4*W_T2;
    N3_I := I1*W_P3 + I2*W_V3 + I3*W_I3 + I4*W_T3;
    N4_I := I1*W_P4 + I2*W_V4 + I3*W_I4 + I4*W_T4;

  END_IF;

(*****)

  IF step = 2 THEN

    N1_O := 1/(1+ exp(-1*N1_I));
    N2_O := 1/(1+ exp(-1*N2_I));
    N3_O := 1/(1+ exp(-1*N3_I));
    N4_O := 1/(1+ exp(-1*N4_I));

  END_IF;

(*****)

  IF step = 3 THEN

    N5_I := N1_O*W_15 + N2_O*W_25 + N3_O*W_35 + N4_O*W_45;
    N6_I := N1_O*W_16 + N2_O*W_26 + N3_O*W_36 + N4_O*W_46;
    N7_I := N1_O*W_17 + N2_O*W_27 + N3_O*W_37 + N4_O*W_47;

  END_IF;

(*****)

  IF step = 4 THEN

    N5_O := 1/(1+ exp(-1*N5_I));
    N6_O := 1/(1+ exp(-1*N6_I));
    N7_O := 1/(1+ exp(-1*N7_I));

  END_IF;
  
```



```

(*****
  IF step = 5 THEN
    N8_I := N5_O*W_58 + N6_O*W_68 + N7_O*W_78;
    N9_I := N5_O*W_59 + N6_O*W_69 + N7_O*W_79;
  END_IF;
(*****
  IF step = 6 THEN
    N8_O := 1/(1+ exp(-1*N8_I));
    N9_O := 1/(1+ exp(-1*N9_I));
  END_IF;
(*****
  IF step = 7 THEN
    N10_I := N8_O*W_810 + N9_O*W_910;
    N10_O := 1/(1+ exp(-1*N10_I));
    Out := N10_O;
  END_IF;
(*****
  ***                               Etapa de Retropropagación                               ***
  ***                               de la RN basado en el error de la neurona final                               ***
  *** Termino "N10_O*(1 - N10_O)" por la función Sigmoidal ***
  *****)
  IF step = 8 THEN
    E10 := N10_O*(1 - N10_O)*(SP - N10_O);
    Error := 100*abs(SP - N10_O)/SP;
  END_IF;
(*****
  ***                               Cálculo de nuevos pesos                               ***
  ***                               de la RN basado en el error de la neurona final                               ***
  *****)
  IF step = 9 THEN
    W_810_up := W_810 + E10*N8_O;
    W_910_up := W_910 + E10*N9_O;
  END_IF;
(*****
  IF step = 10 THEN
    E9 := E10*W_910*N9_O*(1 - N9_O);
    E8 := E10*W_810*N8_O*(1 - N8_O);
  END_IF;
(*****

```

```

IF step = 11 THEN

    W_58_up := W_58 + E8*N5_O;
    W_68_up := W_68 + E8*N6_O;
    W_78_up := W_78 + E8*N7_O;

    W_59_up := W_59 + E9*N5_O;
    W_69_up := W_69 + E9*N6_O;
    W_79_up := W_79 + E9*N7_O;

END_IF;

(*****)

IF step = 12 THEN

    E7 := (E9*W_79 + E8*W_78)*N7_O*(1 - N7_O);
    E6 := (E9*W_69 + E8*W_68)*N6_O*(1 - N6_O);
    E5 := (E9*W_59 + E8*W_58)*N5_O*(1 - N5_O);

END_IF;

(*****)

IF step = 13 THEN

    W_15_up := W_15 + E5*N1_O;
    W_25_up := W_25 + E5*N2_O;
    W_35_up := W_35 + E5*N3_O;
    W_45_up := W_45 + E5*N4_O;

    W_16_up := W_16 + E6*N1_O;
    W_26_up := W_26 + E6*N2_O;
    W_36_up := W_36 + E6*N3_O;
    W_46_up := W_46 + E6*N4_O;

    W_17_up := W_17 + E7*N1_O;
    W_27_up := W_27 + E7*N2_O;
    W_37_up := W_37 + E7*N3_O;
    W_47_up := W_47 + E7*N4_O;

END_IF;

(*****)

IF step = 14 THEN

    E1 := (E7*W_17 + E6*W_16 + E5*W_15)*N1_O*(1 - N1_O);
    E2 := (E7*W_27 + E6*W_26 + E5*W_25)*N2_O*(1 - N1_O);
    E3 := (E7*W_37 + E6*W_36 + E5*W_35)*N3_O*(1 - N1_O);
    E4 := (E7*W_47 + E6*W_46 + E5*W_45)*N4_O*(1 - N1_O);

END_IF;

(*****)

```

```

IF step = 15 THEN
    W_P1_up := W_P1 + E1*I1;
    W_P2_up := W_P2 + E2*I1;
    W_P3_up := W_P3 + E3*I1;
    W_P4_up := W_P4 + E4*I1;

    W_V1_up := W_V1 + E1*I2;
    W_V2_up := W_V2 + E2*I2;
    W_V3_up := W_V3 + E3*I2;
    W_V4_up := W_V4 + E4*I2;

    W_I1_up := W_I1 + E1*I3;
    W_I2_up := W_I2 + E2*I3;
    W_I3_up := W_I3 + E3*I3;
    W_I4_up := W_I4 + E4*I3;

    W_T1_up := W_T1 + E1*I4;
    W_T2_up := W_T2 + E2*I4;
    W_T3_up := W_T3 + E3*I4;
    W_T4_up := W_T4 + E4*I4;

END_IF;

(*****
***      Asignación de nuevos pesos de la RN      ***
*****)

IF step = 16 THEN
    W_P1 := W_P1_up; W_P2 := W_P2_up; W_P3 := W_P3_up; W_P4 := W_P4_up;
    W_V1 := W_V1_up; W_V2 := W_V2_up; W_V3 := W_V3_up; W_V4 := W_V4_up;
    W_I1 := W_I1_up; W_I2 := W_I2_up; W_I3 := W_I3_up; W_I4 := W_I4_up;
    W_T1 := W_T1_up; W_T2 := W_T2_up; W_T3 := W_T3_up; W_T4 := W_T4_up;
    W_15 := W_15_up; W_16 := W_16_up; W_17 := W_17_up; W_25 := W_25_up;
    W_26 := W_26_up; W_27 := W_27_up; W_35 := W_35_up; W_36 := W_36_up;
    W_37 := W_37_up; W_45 := W_45_up; W_46 := W_46_up; W_47 := W_47_up;
    W_58 := W_58_up; W_59 := W_59_up; W_68 := W_68_up; W_69 := W_69_up;
    W_78 := W_78_up; W_79 := W_79_up; W_810 := W_810_up;
    W_910 := W_910_up;

END_IF;

```

La variable “Step” permite ejecutar las fases del algoritmo de retropropagación secuencialmente. “Finalmente, “Step” puede tomar valores desde “1” al “16” y la variación se realiza a través del siguiente programa codificado en lógica de bloques funcionales.

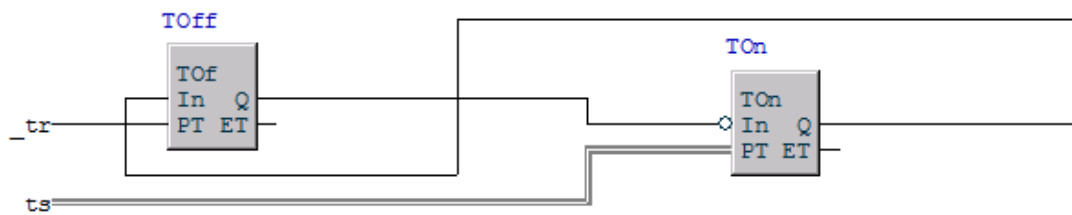


Figura 3.48 Programa Seleccionador de Fase – Algoritmo de “Retropropagación”

Fuente: Elaboración propia

Este programa representa un contador secuencial del “1” al “16”. Cada vez que el temporizador del tipo “T_On” finaliza el tiempo de “Preset”, la variable “Step” aumenta su valor en una unidad. El temporizador del tipo “T_off” reinicializa el temporizador “T_On” a fin de inicializar nuevamente el tiempo de “Preset” e incrementar una vez

más el valor de “Step”. Cuando “Step” llega a alcanzar el valor de “16”, reinicializa su valor a “1” a fin de aplicar el algoritmo a la totalidad de patrones de entrenamiento ingresados (“Presión”, “Velocidad”, “Impactos”, “Tonelaje”, “Nivel”). Es importante resaltar que la lógica de ejecución dentro de un “Control module” es recursiva, eso quiere decir que una vez ejecutada la última línea de código, inmediatamente después se ejecuta nuevamente la primera línea de código sin necesidad de definir un bucle en particular.

II. CODIGO FUENTE: FB3_Red_Neuronal_Entrenada

```
(*****
*****
  IF step = 1 THEN
    N1_I := I1*W_P1 + I2*W_V1 + I3*W_I1 + I4*W_T1;
    N2_I := I1*W_P2 + I2*W_V2 + I3*W_I2 + I4*W_T2;
    N3_I := I1*W_P3 + I2*W_V3 + I3*W_I3 + I4*W_T3;
    N4_I := I1*W_P4 + I2*W_V4 + I3*W_I4 + I4*W_T4;

  END_IF;

(*****
*****
  IF step = 2 THEN
    N1_O := 1/(1+ exp(-1*N1_I));
    N2_O := 1/(1+ exp(-1*N2_I));
    N3_O := 1/(1+ exp(-1*N3_I));
    N4_O := 1/(1+ exp(-1*N4_I));

  END_IF;

(*****
*****
  IF step = 3 THEN
    N5_I := N1_O*W_15 + N2_O*W_25 + N3_O*W_35 + N4_O*W_45;
    N6_I := N1_O*W_16 + N2_O*W_26 + N3_O*W_36 + N4_O*W_46;
    N7_I := N1_O*W_17 + N2_O*W_27 + N3_O*W_37 + N4_O*W_47;

  END_IF;

(*****
*****
  IF step = 4 THEN
    N5_O := 1/(1+ exp(-1*N5_I));
    N6_O := 1/(1+ exp(-1*N6_I));
    N7_O := 1/(1+ exp(-1*N7_I));

  END_IF;
```

```

(*****)
(*****)

    IF step = 5 THEN

        N8_I := N5_O*W_58 + N6_O*W_68 + N7_O*W_78;
        N9_I := N5_O*W_59 + N6_O*W_69 + N7_O*W_79;

    END_IF;

(*****)
(*****)

    IF step = 6 THEN

        N8_O := 1/(1+ exp(-1*N8_I));
        N9_O := 1/(1+ exp(-1*N9_I));

    END_IF;

(*****)
(*****)

    IF step = 7 THEN

        N10_I := N8_O*W_810 + N9_O*W_910;
        N10_O := 1/(1+ exp(-1*N10_I));

    END_IF;

(*****)
(*****)

    IF step = 8 THEN

        E10 := N10_O*(1 - N10_O)*(Nivel_Deseado - N10_O);
        Error := 100*abs(Nivel_Deseado - N10_O)/Nivel_Deseado;
        Nivel_Estimado := N10_O;

    END_IF;

```

Una vez finalizado el entrenamiento supervisado, se aplican los nuevos pesos sinápticos a la red neuronal a fin de evaluar el performance. Consideremos que la evaluación del performance se realiza con un set de datos distinto al set de entrenamiento. Adicionalmente, la variable “Step” nuevamente nos permite ejecutar secuencialmente el proceso de propagación de entradas a la red. Finalmente, la variación de “Step” se realiza nuevamente a través del “Programa Seleccionador de Fase”. El número de fase para esta instancia es “8” y la variable reinicia su valor a “1” siempre y cuando alcance el valor de “8” a fin de procesar la totalidad de entradas ingresadas a la red (“Presión”, “Velocidad”, “Impactos”, “Tonelaje”).

III. **CODIGO FUENTE:** FB1_Ingreso_Datos_Entrenamiento

```
(*****  
(*****          DATO 1          *****  
(*****
```

```
IF selector = 1 THEN
```

```
    Presión      := 0.747008496;  
    Velocidad    := 0.9399644;  
    Impactos     := 0.29929611;  
    Tonelaje     := 0.578142969;  
    Nivel        := 0.213
```

```
END_IF;
```

```
...  
...  
...
```

```
(*****  
(*****          DATO 121          *****  
(*****
```

```
IF selector = 121 THEN
```

```
    Presión      := 0.750779981;  
    Velocidad    := 0.9065368;  
    Impactos     := 0.1731138;  
    Tonelaje     := 0.571757666;  
    Nivel        := 0.224;  
    selector     := 1;
```

```
END_IF;
```

La programación de este bloque funcional posee 121 datos pertenecientes al set de entrenamiento. Asimismo el ingreso de cada patrón se realiza tal cual se ingresó el primer dato (por cuestiones de espacio no se repite “121” veces lo mismo). El selector varía de “1” a “121” y se reinicializa a “1” una vez que se alcance la cuenta máxima. Nuevamente la lógica en el “Control Module” es recursiva a fin de enviar el set de datos completo al bloque funcional de entrenamiento hasta que el error generado por el algoritmo sea mínimo.

CAPÍTULO 4

RESULTADOS

En este capítulo se presentan las simulaciones que verifican el funcionamiento del sistema neuronal. El sistema se simuló a través de Control Builder 5.1 y se dividió en tres etapas: "Entrenamiento de la red neuronal", "Estimación de nivel de carga del molino" y "Análisis de resultados".

4.1 Entrenamiento de la red neuronal

Se realizó la simulación del sistema neuronal utilizando la librería "Control_Neuronal" desarrollada en el capítulo 3. A continuación se describen los pasos para simular el sistema.

Paso 1: Simular el controlador del sistema 200DCU002.

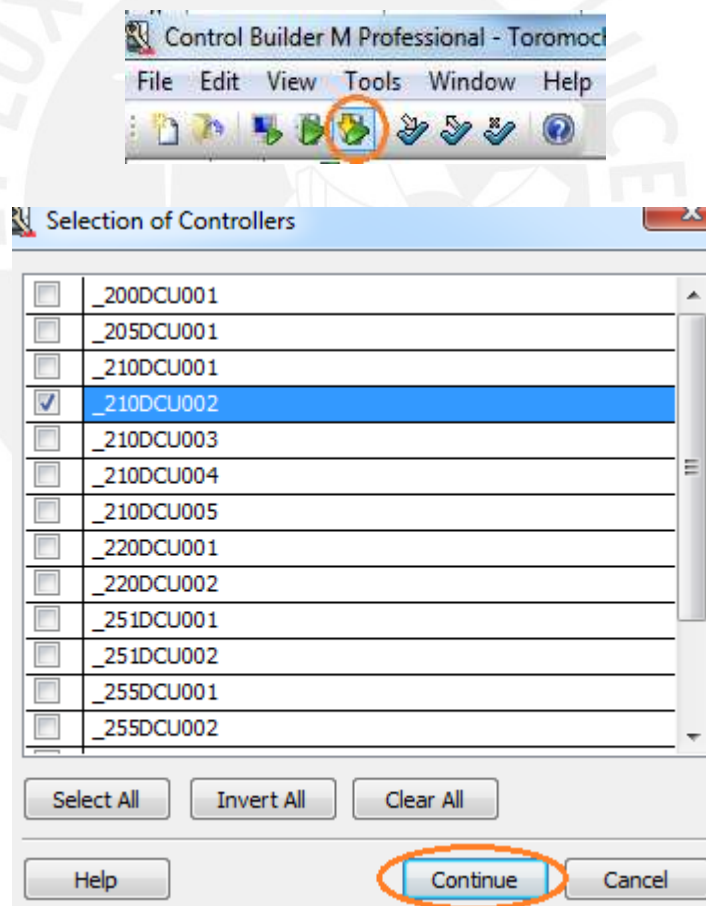


Figura 4.1 Simulación del controlador 200DCU002

Fuente: Elaboración propia

Paso 2: Iniciar entrenamiento de la red neuronal.

La figura 4.2 presenta el programa “Tesis – Entrenamiento”. El sistema procesó un día de data. El nivel de carga se obtuvo a través de la experiencia del operador en sala de control y del análisis de tendencias históricas de las variables características del molino (presión, velocidad, impactos, tonelaje).

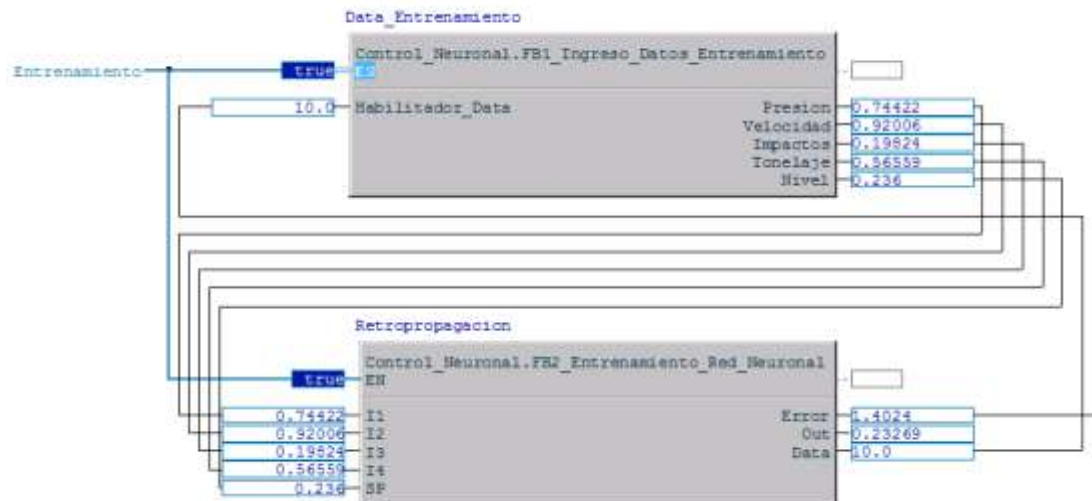


Figura 4.2 Simulación del controlador 200DCU002

Fuente: Elaboración propia

Una vez que todos los patrones se ingresaron, se aplicó nuevamente el primer patrón, segundo patrón, tercer patrón y así sucesivamente hasta que el error final en la neurona de salida “ErrorRN” se redujo al mínimo. Por cada patrón ingresado el programa calculó la actualización de los pesos sinápticos de la red.

Paso 3: Finalizar entrenamiento de la red neuronal.

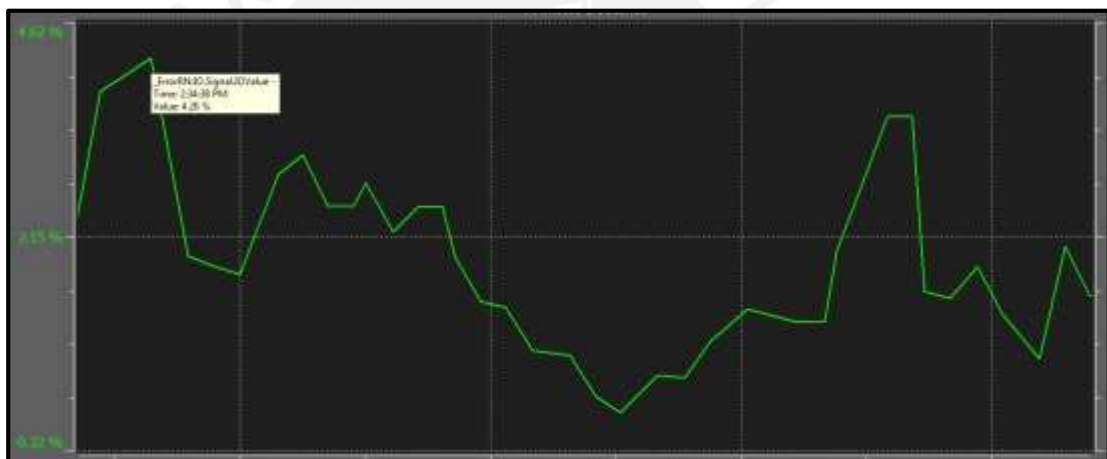
Al inicio del entrenamiento la red neuronal registró valores cercanos al 40% de error, valores que fueron disminuyendo gradualmente a medida que la red adquirió el aprendizaje necesario para procesar la data ingresada.



Figura 4.3 Error inicial – Neurona de salida

Fuente: Elaboración propia

Se estuvo entrenando la red por un lapso de doce horas y se obtuvo la siguiente tendencia histórica del error final en la neurona de salida.



| | Visible | Trace Color | Object Name | Mean Value | Min Value | Max Value |
|---|-------------------------------------|-------------|-------------|------------|-----------|-----------|
| 1 | <input checked="" type="checkbox"/> | | _ErrorRN | 1.90 % | 0.13 % | 4.22 % |

Figura 4.4 Error final – Neurona de salida

Fuente: Elaboración propia

El error final de entrenamiento después de doce horas se redujo a un máximo de 4.22%. A continuación se muestran los pesos sinápticos finales obtenidos.

| Name | Current Value | Name | Current Value |
|------|---------------|-------|---------------|
| W_P1 | 0.79975 | W_17 | 0.7986 |
| W_P2 | 0.79975 | W_25 | 0.99926 |
| W_P3 | 0.79975 | W_26 | 0.99926 |
| W_P4 | 0.79975 | W_27 | 0.79858 |
| W_V1 | 0.7997 | W_35 | 0.99926 |
| W_V2 | 0.7997 | W_36 | 0.99926 |
| W_V3 | 0.7997 | W_37 | 0.79858 |
| W_V4 | 0.7997 | W_45 | 0.99926 |
| W_I1 | 0.79989 | W_46 | 0.99926 |
| W_I2 | 0.79989 | W_47 | 0.79858 |
| W_I3 | 0.79989 | W_58 | 0.98514 |
| W_I4 | 0.79989 | W_59 | 0.98514 |
| W_I1 | 0.79985 | W_68 | 0.98514 |
| W_T2 | 0.99985 | W_69 | 0.98514 |
| W_T3 | 0.99985 | W_78 | 0.98556 |
| W_T4 | 0.99985 | W_79 | 0.98556 |
| W_15 | 0.99924 | W_810 | -0.65858 |
| W_16 | 0.99924 | W_910 | -0.65858 |

Figura 4.5 Pesos sinápticos al finalizar entrenamiento

Fuente: Elaboración propia

4.2 Estimación de nivel de carga del molino

La figura 4.6 representa el programa que se utilizó para la estimación de nivel de carga del molino. Utilizando los pesos sinápticos finales se evaluó el desempeño de la red neuronal.

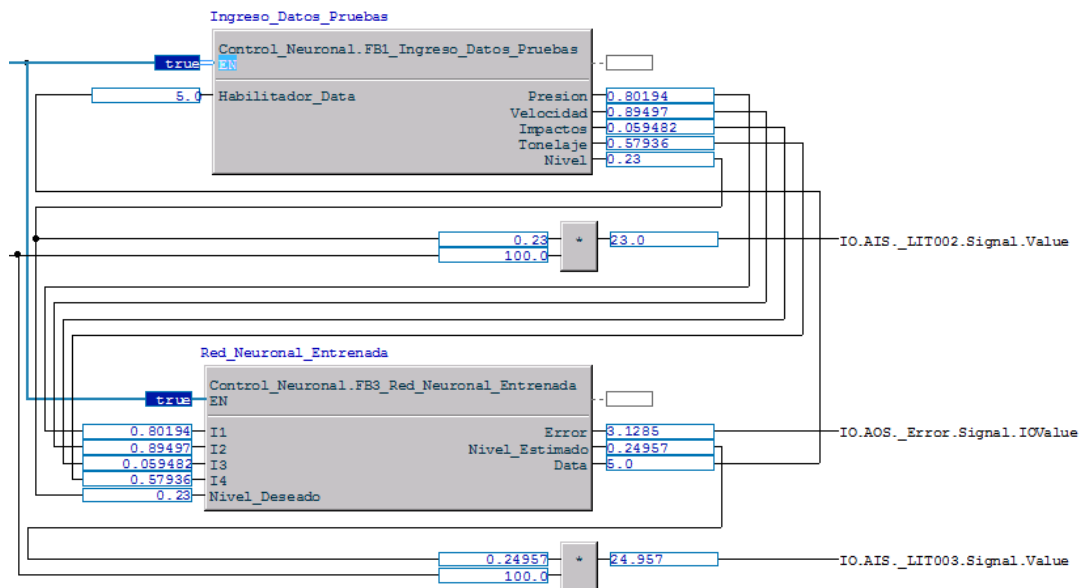


Figura 4.6 Programa para estimar el nivel de carga

Fuente: Elaboración propia

Se ingresó valores de presión, velocidad, impactos y tonelaje relacionados a la operación del molino y la red neuronal calculó el nivel de carga estimado. El nivel estimado por la red neuronal se comparó con el nivel real medido en campo por los operadores y se obtuvo la siguiente gráfica.

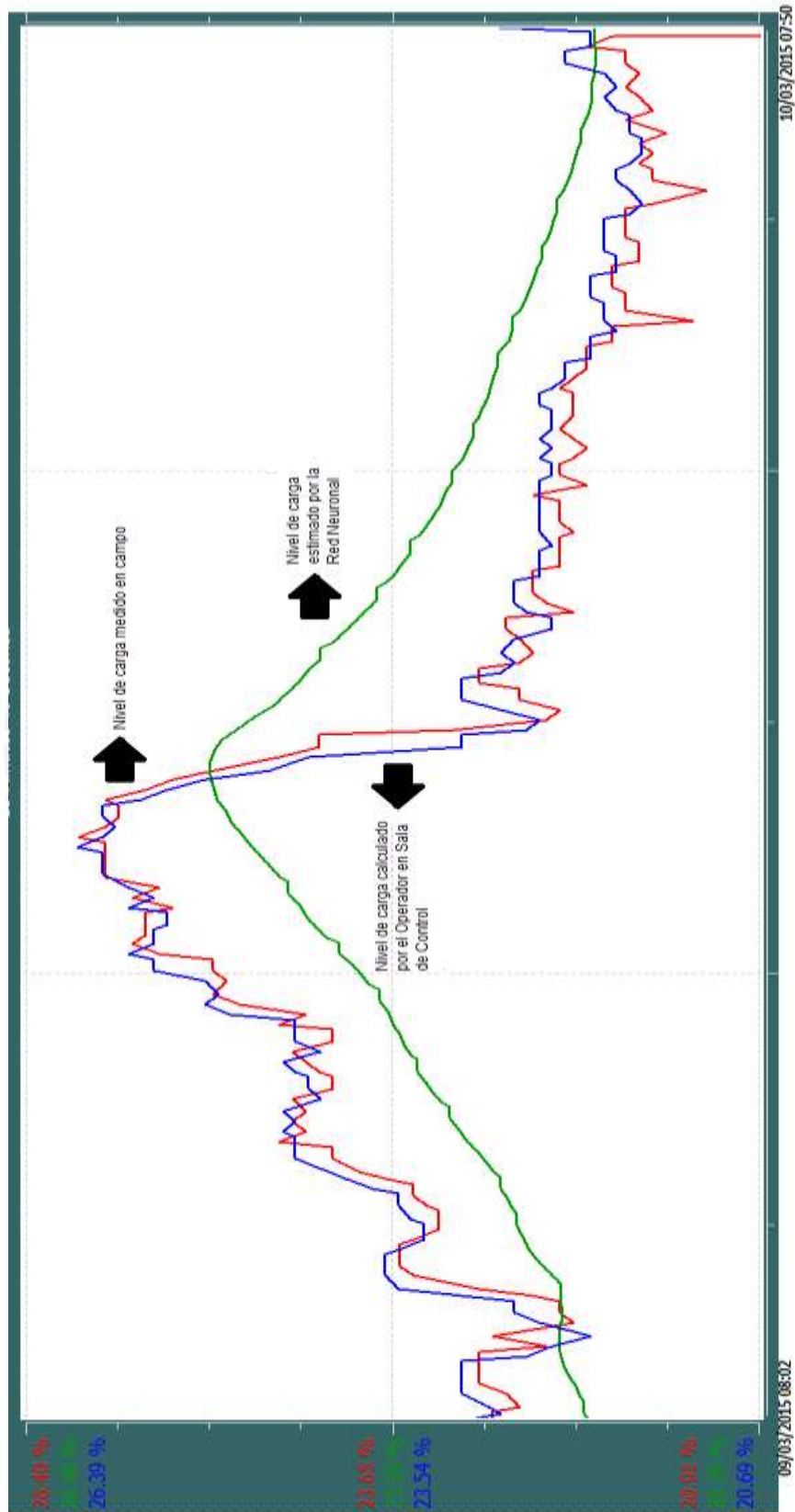


Figura 4.7 Nivel de carga (%) vs. Tiempo(s)

Fuente: Elaboración propia

Se notó que la red neuronal es capaz de estimar el nivel de carga con un error de $\pm 5\%$ el cual se especifica en la Figura 4.4. La “tendencia histórica” asociada al error generada por el System800xA permite al usuario registrar el valor máximo alcanzado durante el tiempo de simulación, lo cual nos ayuda a determinar el error máximo generado por la red neuronal. A medida que la red neuronal procese mayor cantidad de datos en la etapa de entrenamiento (“1” o “2” meses), obtendrá salidas con menor margen de error para datos nunca antes vistos. Es decir, el nivel de generalización e independencia convierten a la red neuronal en una opción tan eficiente como el operador de sala de control. Sin embargo, la principal ventaja sobre el operador de sala es que la red neuronal predice el nivel de carga con mayor rapidez, lo cual permite tomar acciones correctivas en menor tiempo. De esta manera el control se automatiza, disminuyendo los tiempos muertos en la operación y aumentando la eficiencia del proceso. Una vez estimado el nivel de carga, se diseñó el “HMI Screen” encargado de mostrar al operador las señales características del molino, el nivel de carga estimado por la red neuronal, tendencias históricas y los feedbacks de operación pertinentes para situaciones de alta o baja carga en el molino.

El HMI diseñado muestra el típico esquema de operación asociado al proceso de molienda. El mineral junto con las bolas de acero ingresan por la faja transportadora “**210CV001**” hacia al chute de ingreso del molino “**210CH001**”. En el chute de ingreso se añade agua y la pulpa obtenida es procesada por el molino. Luego de procesar la pulpa en el molino, el mineral resultante se deposita en el chute de salida “**210CH010**” a fin de realizar la etapa de clasificación de partículas de mineral a través de la zaranda (elemento clasificador de tamaño de partículas). El mineral que obtenga el tamaño apropiado será ingresado al “Cajón de Pulpa” con la finalidad de ingresar a la siguiente etapa del proceso general. Finalmente, el mineral que no obtenga el tamaño adecuado de molienda será reprocesado nuevamente por el molino. Es decir, se transporta desde la faja “**210CV004**” hacia la faja “**210CV001**” para la etapa de reprocesamiento.

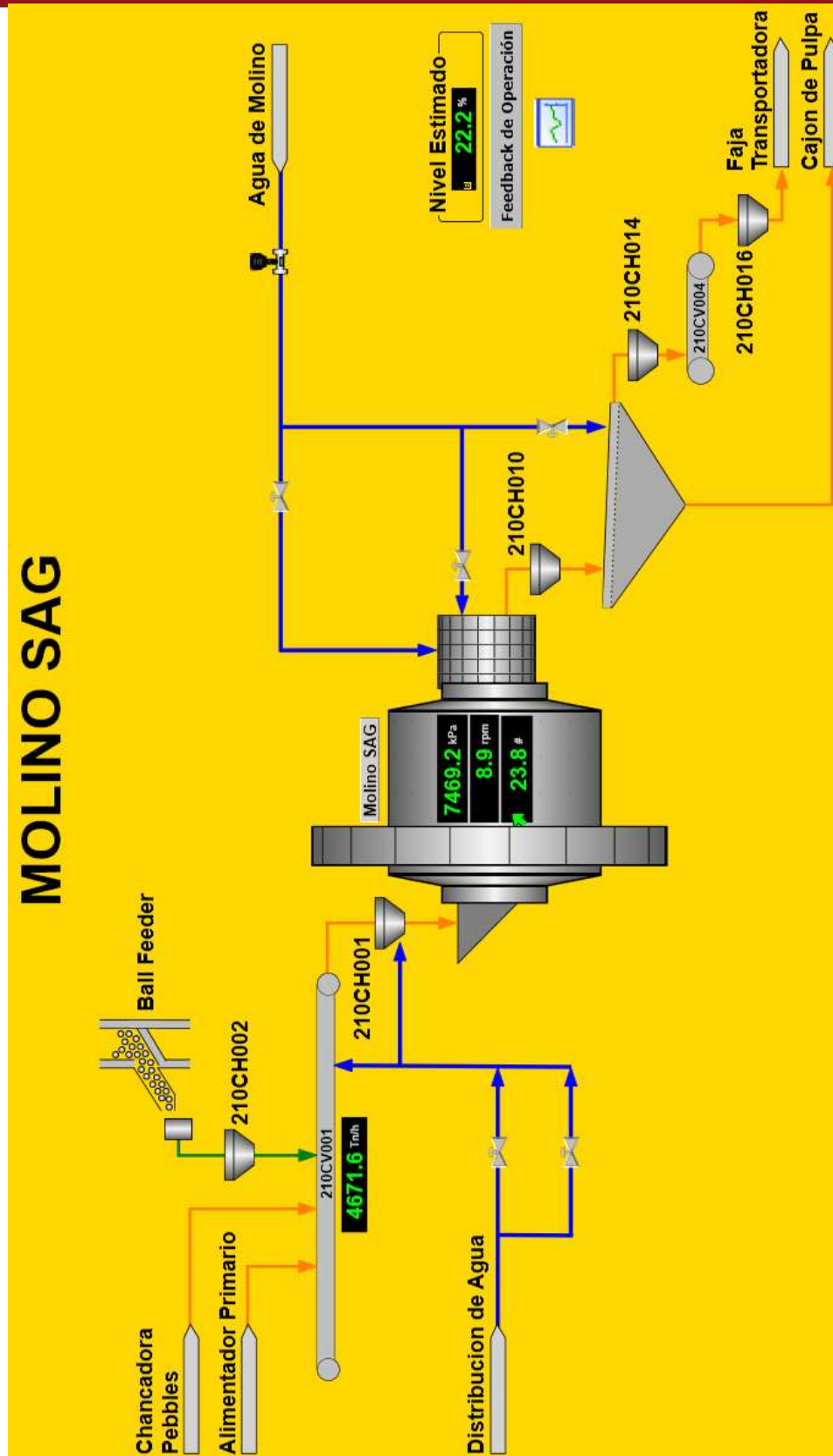


Figura 4.8 HMI Screen – Control de Molinos

Fuente: Elaboración propia

Si en un determinado momento el operador está controlando el molino a baja o alta carga, podrá visualizar las acciones correctivas presionando el vínculo de “Feedback de Operación” en el HMI. Este menú despliega las acciones correcciones típicas que realiza un operador experimentado de molinos.

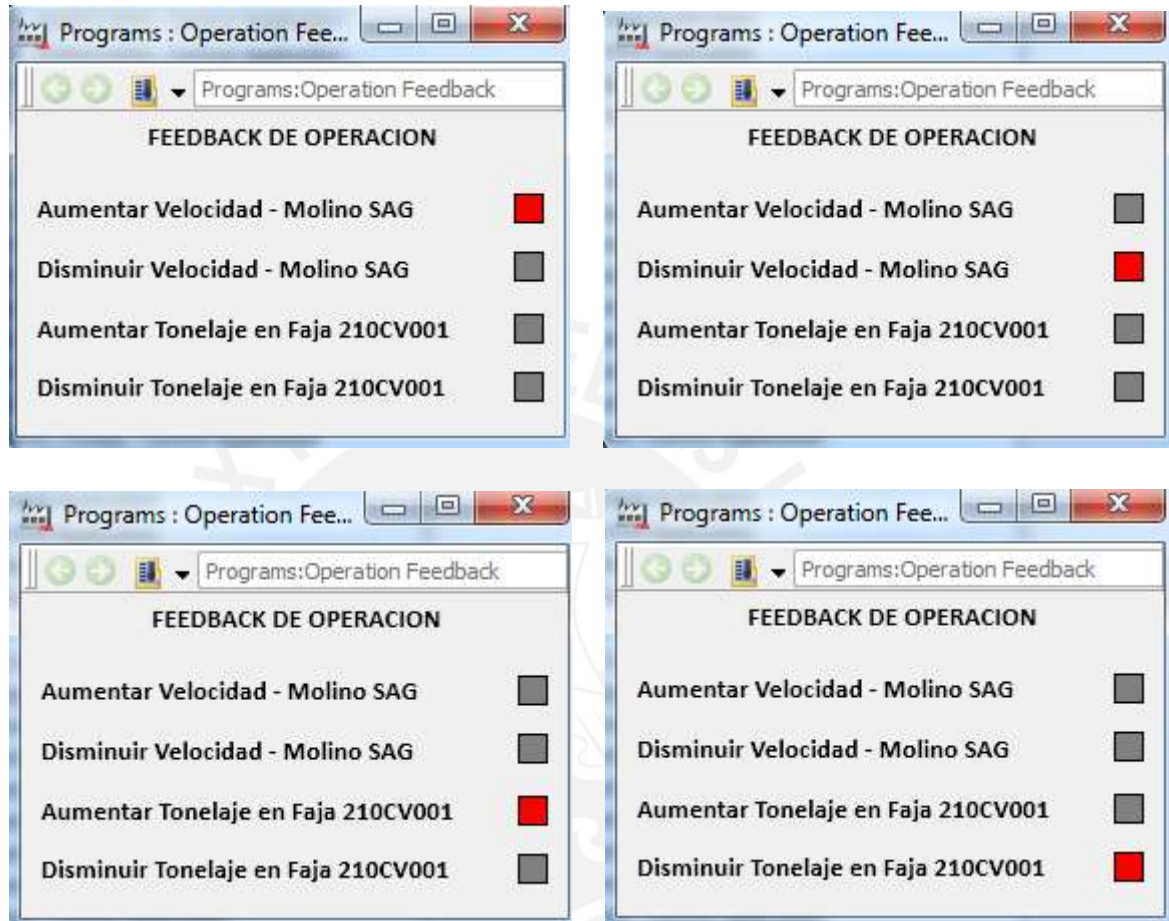


Figura 4.9 Pantalla HMI – Feedback de Operación

Fuente: Elaboración propia

También podrá visualizar las principales variables de operación del molino en tiempo real presionando el vínculo de tendencias en el HMI.



Figura 4.10 Pantalla HMI – Vínculo de Tendencias

Fuente: Elaboración propia

La Figura 4.11 muestra las gráficas que puede obtener el operador de sala de control cuando ingresa al vínculo de tendencias. El gráfico de tendencias brinda detalle de las variables principales de la red neuronal tales como presión, velocidad, impactos

y tonelaje del molino. El operador puede seleccionar la data que desea visualizar al momento y analizar el estado de la operación.

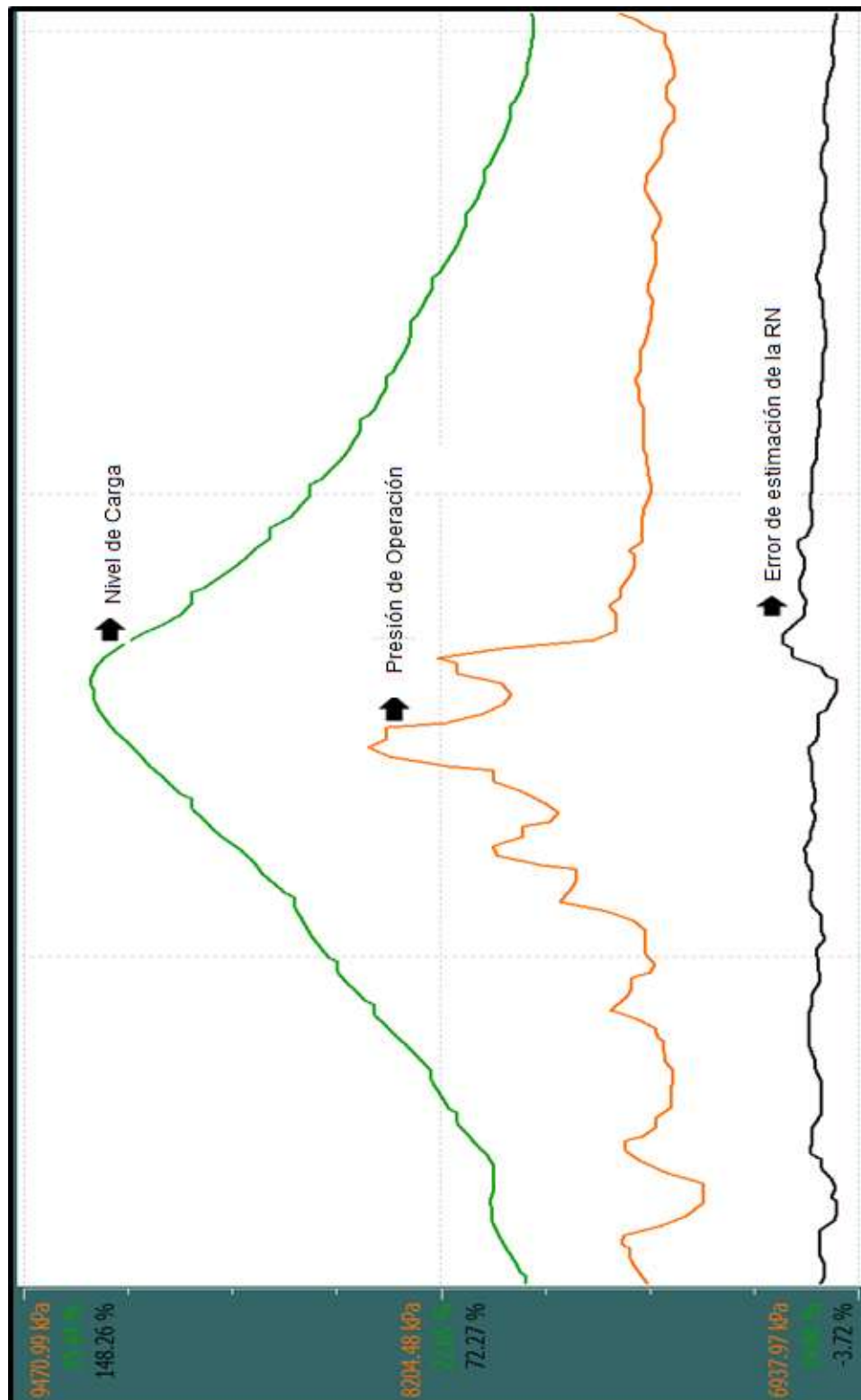
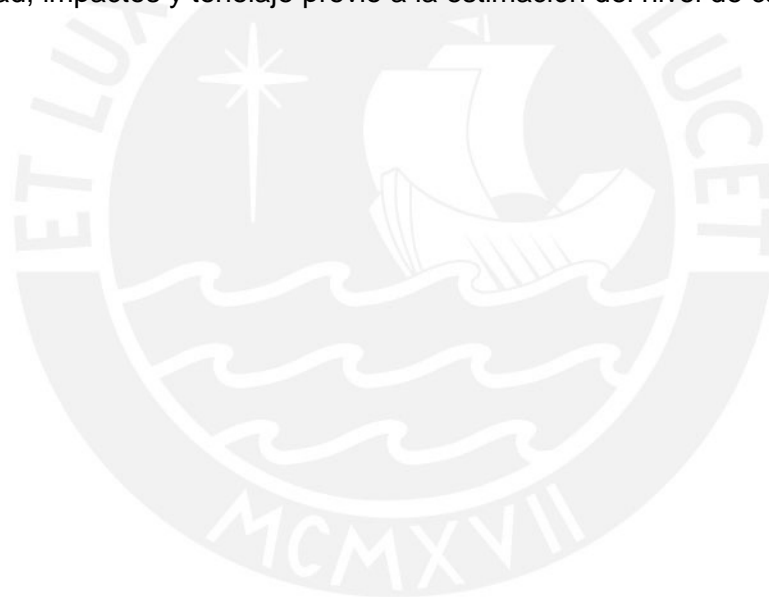


Figura 4.11 Pantalla HMI con tendencias de históricos

Fuente: Elaboración propia

4.3 Análisis de resultados

Puede notarse que el nivel de carga estimado por el operador se asemeja en gran medida al obtenido en cada “Crush Stop”. Por otro lado, la estimación realizada por la red neuronal también predice la tendencia del nivel de carga; sin embargo, presenta un margen de error mayor al del operador de sala de control. Esto se debe principalmente a la limitante de data utilizada durante el entrenamiento (“1” día de información). A medida que la red se entrene con mayor cantidad de datos (“1” semana, “1” mes), tendrá mayor capacidad de generalizar niveles de carga para datos nunca antes vistos durante el entrenamiento. De este modo podrá estimar los niveles de carga igual o mejor que el operador de sala. Finalmente, se puede notar que el sistema neuronal presenta un pequeño “offset” o retraso durante la estimación en relación a las mediciones durante los “Crush stop”. Esto se debe a que la red neuronal debe esperar la evolución en el tiempo de las variables de presión, velocidad, impactos y tonelaje previo a la estimación del nivel de carga.



PRESUPUESTO

En este capítulo se menciona el costo de los equipos utilizados por el sistema de estimación. De esta manera se puede tener referencia de la inversión que implica un sistema de control orientado a minería.

| Item | Cantidad | Descripción | Costo (S/.) | Costo (\$) |
|--------------|----------|---|--------------|--------------|
| 1 | 1 | Módulo de Entrada Analógica AI 815. | 672 | 210 |
| 2 | 1 | Módulo de Comunicación Modbus TCP/IP CI867. | 3974 | 1242 |
| 3 | 1 | Unidad de Procesador PM866 | 48'000 | 15 000 |
| 4 | 1 | Control Builder 5.1. | 3085 | 964 |
| 5 | 500 | Horas Hombre utilizadas durante el proyecto. (3meses) | 10'000 | 3125 |
| TOTAL | | | 65731 | 20541 |

Tabla 5.1 Presupuesto

Fuente: Elaboración propia

La tabla 5.1 muestra equipos utilizados en proyectos de ámbito industrial, por lo cual su costo es elevado. El controlador PM866 es uno de los modelos de mayor capacidad de la gama AC800M, debido a que el proceso de molienda implica el manejo de gran cantidad de información. El modelo de la tarjeta de entrada analógica, es el más económico dentro de la gama AC800M puesto que no permite “cambios en caliente”, esto quiere decir, que si se retira una de estas tarjetas, todo el sistema entra en falla. Finalmente, la instrumentación relacionada a las variables de presión, velocidad, impactos y tonelaje pertenecen al diseño de planta original, por tanto solo se hace uso de la data que registran más no se considera dentro de los gastos del proyecto.

CONCLUSIONES

Las conclusiones del trabajo de tesis son:

- Se integró satisfactoriamente las señales de presión, velocidad y tonelaje a Control Builder 5.1 a través de cableado duro (4...20mA).
- Se integró satisfactoriamente la señal de impactos a través del protocolo industrial de comunicación MODBUS TCP.
- Se desarrolló satisfactoriamente la aplicación en Control Builder 5.1 de un sistema neuronal capaz de estimar el nivel de carga del molino en base a mediciones simples de parámetros de operación ampliamente conocidos.
- Se realizó simulaciones satisfactorias del performance de la red neuronal. Se obtuvo un error final menor al 5%.
- Se logró desarrollar un sistema neuronal capaz de superar las limitaciones del sistema de instrumentación. Hoy en día determinar el nivel de carga directamente y en tiempo real a través de un sistema de instrumentación no es posible.
- Se desarrolló satisfactoriamente la librería de control neuronal utilizando "Control Builder 5.1" y el algoritmo neuronal de "Retropropagación". Se simuló la librería neuronal y se verificó el funcionamiento del algoritmo implementado.
- Se verificó la capacidad de aprendizaje de la librería neuronal sin la necesidad de tener algún modelo matemático explícito que explique la dinámica del sistema.
- La estimación realizada por el experto en la operación se asemeja en gran medida al nivel real medido en campo. La red neuronal adquiere este conocimiento y es capaz de predecir eficientemente.

RECOMENDACIONES

- Se recomienda que el sistema tenga la capacidad de recalibrarse en caso las condiciones de operación varíen. Típicamente durante el procesamiento de minerales, las condiciones de operación tienden a ser fluctuantes. Es decir, la dureza del mineral puede variar de un momento a otro, o la presión de operación puede registrar distintos valores dependiendo del tipo de mineral que se esté procesando, lo cual afectará el performance del sistema de control.
- Se recomienda que el ingreso de data al sistema se realice periódicamente. Mientras más data de procesamiento registre el sistema de control, mayor gama de decisiones podrá realizar de forma autónoma.
- Se recomienda que las reglas de operación del sistema de control sean definidas por los expertos en la operación. El performance de la red es la experiencia del operador depositada en el sistema de control.
- Se recomienda utilizar las paradas de mantenimiento programado para realizar “Crush Stops” a fin de uniformizar criterios entre el nivel de carga real medido en campo y el nivel de carga estimado por el operador en sala de control.
- Se recomienda utilizar sensores en redundancia dentro de la lógica de control. Típicamente los sensores en campo tienden a tener fluctuaciones debido a la contaminación, interferencias, entre otros. Debido a ello, si se perdiera la comunicación en alguno de los sensores, el sistema podría continuar la lógica de control utilizando el sensor redundante. Esta medida de control proporcionaría un sistema de respaldo en caso de fallas.

BIBLIOGRAFÍA

1. ABB. (2010). *AC800M Communication Protocols*. Switzerland. Retrieved Abril 2015
2. ABB. (2010). *Compact Control Buider AC800M 5.1*. Switzerland. Retrieved Abril 2015
3. ABB. (2010). *System800xA Configuration Version 5.1*. Switzerland. Retrieved Abril 2015
4. ABB. (2010). *System800xA Configuration Version 5.1 Getting Started*. Switzerland. Retrieved Abril 2015
5. ABB. (2010). *System800xA Installation Version 5.1*. Switzerland. Retrieved Abril 2015
6. ABB. (2010). *System800xA Tools Version 5.1*. Switzerland. Retrieved Abril 2015
7. Bertona, L. (2005). Entrenamiento de redes neuronales basado en algoritmos evolutivos. Retrieved Enero 2015, from <http://laboratorios.fi.uba.ar/lsi/bertonatesisingeneriainformatica.pdf>
8. Blender Data Communications. (n.d.). Modbus TCP/IP Protocol Manual. Retrieved Febrero 2015, from http://www.maguire.com/ps_image/pdf/Modbus_TCPIP_Protocol.pdf
9. Cao, H. (2008). Load Control of Ball Mill by a High Precision Sampling Fuzzy Logic Controller with Self-Optimizing. *Asian Journal of Control*, 10(6), 621-631. Retrieved Octubre 2014, from http://www.readcube.com/articles/10.1002%2Fasjc.63?r3_referer=wol&tracking_action=preview_click&show_checkout=1&purchase_referrer=onlinelibrary.wiley.com&purchase_site_license=LICENSE_DENIED
10. FLS Smidth. (2011). Installation, Operation and Maintenance Manual Toromocho Project. Retrieved Febrero 2015
11. Gugel, K. S. (2007). Automated Mill Control using Vibration Signal Processing. *Charleston World Cement Conference*. Retrieved Octubre 2014, from http://www.digitalcontrollab.com/documents/AutoContro_Vibe_IEEE07.pdf
12. M.G.Melero. (2014). Electric Motor Monitoring: An alternative to increase the efficiency of ball mills. 10(12). Retrieved Octubre 2014
13. MILLSCAN. (2010). Using Vibrations to Measure Mill Fill Level. *L.V. Technology*. Retrieved Octubre 2014, from 1324540259
14. Monov, V. (n.d.). Grinding in Ball Mills: Modelling and Process Control. *Cybernetics and Information Technologies*, 12(2). Retrieved Octubre 2014,

from http://www.cit.iit.bas.bg/CIT_2012/v12-2/Monov,Sokolov,Stoenchev-pp51-68.pdf

15. Peng, H. (2010). New Method to Measure the Fill Level of the Ball Mill I - Theoretical Analysis and DEM Simulation. *Chinese Journal of Mechanical Engineering*, 23(3). Retrieved Octubre 2014, from <http://124.205.29.37/xuebaoenglish/papersed/allpapered/newaccepted/fulltext/2009-101.pdf>
16. Schlanz, J. w. (2006). Grinding: An Overview of Operation and Design. *Mineral Research Laboratory*. Retrieved Octubre 2014, from http://mrl.ies.ncsu.edu/reports/87-31-P_Grinding_Operations_Design.pdf
17. Valencia, M. (2006). Algoritmo Backpropagation para Redes Neuronales: Conceptos y aplicaciones. Retrieved Diciembre 2014, from <http://www.repositoriodigital.ipn.mx/bitstream/handle/123456789/8628/Archivo%20que%20incluye%20portada%2c%20%C3%ADndice%20y%20texto.pdf?sequence=1>
18. W.STANGE. (2002). Using Artificial Neural Networks for the Control of Grinding Circuits. *Minerals Engineering*.
19. ZHAO, D. (n.d.). Intelligent Optimal Control System for Ball Mill Grinding Process. *Journal Control Theory*, 11(3). Retrieved Octubre 2014, from <http://link.springer.com/article/10.1007%2Fs11768-013-1210-3#/page-1>