

**Estimador de estimador de estado con PMU virtual en GridTeractions**

**Autor:** Jaime Andres Forero Rodriguez

**Asesor:** Mario Alberto Rios Mesias

**Co-Asesor:** David Felipe Celeita Rodríguez

Para obtener el título de Ingeniero Eléctrico

Universidad de los Andes, Colombia 2021

**Abstract**

A partir de las capacidades de la Unidad de Medición Fasorial (PMU) para medir directamente el estado del sistema y la creciente implementación de PMU en la industria de la energía eléctrica, se ha presentado una expansión de las técnicas de estimación de estado para emplear el uso exclusivo de datos de PMU. Este proyecto especial representa una parte del trabajo desarrollado durante la fase inicial del proyecto de investigación. Esto incluye el desarrollo inicial y la prueba de dos aplicaciones: el estimador de estado lineal y el procesador de topología. También se presenta una breve explicación de la estimación de estado y PMU, técnicas tradicionales de estimación de estado y técnicas con datos fasoriales.

**Índice**

[**Abstract**](#_bookmark0) i

[Tabla de Figuras 5](#_Toc63033628)

[1. Introducción 1](#_Toc63033629)

[**1.1.** **Objetivos Generales** 2](#_Toc63033630)

[**1.2.** **Objetivos específicos** 2](#_Toc63033631)

[2. Estimación de Estado 3](#_Toc63033632)

[**2.1** **GridTeractions** 3](#_Toc63033633)

[**2.2** **Estimación de estado por mínimos cuadrados ponderados (WLS)** 4](#_Toc63033634)

[2.2.1 Algoritmos WLS 5](#_Toc63033635)

[**2.3** **Estimación de Estado por Valor mínimo Absoluto (LAV)** 6](#_Toc63033636)

[2.3.1 Algoritmos LAV 7](#_Toc63033637)

[**2.4** **Estimación de Estado Lineal** 8](#_Toc63033638)

[2.4.1 Equivalencia del modelo π 8](#_Toc63033639)

[2.4.2 Estimación con medidas Fasoriales 9](#_Toc63033640)

[3. Implementación de estimador de estado en GridTeractions 10](#_Toc63033641)

[**3.1** **HIL Scada de Typhoon** 10](#_Toc63033642)

[**3.2** **Configuración del Typhoon Virtual HIL Scada con el código de Estimación de estado lineal** 13](#_Toc63033643)

[**3.3** **Comunicación mediante IEC 61850** 14](#_Toc63033644)

[**3.4** **Caso de Estudio** 17](#_Toc63033645)

[4. Prueba del estimador de estado. 19](#_Toc63033646)

[**4.1** **Comparación de los resultados obtenidos con cada método de estimación de estado.** 19](#_Toc63033647)

[**4.2** **Ejecución de Power Factory en modo no interactivo.** 20](#_Toc63033648)

[**4.3** **Estimación de Estado en Typhoon Virtual HiL Scada** 22](#_Toc63033649)

[**4.4** **Configuración del protocolo Sampled Values** 26](#_Toc63033650)

[5. Conclusiones 28](#_Toc63033651)

# Tabla de Figuras

[Figura 1. Arquitectura GridTeractions [2]. 4](#_Toc63033348)

[Figura 2 Modelo π de una línea de transmisión [4]. 9](#_Toc63033349)

[Figura 3. Typhoon Virtual Hil Scada 11](#_Toc63033350)

[Figura 4. Estados de los nodos 2, 4 y 5. 12](#_Toc63033351)

[Figura 5. Estados Estimados 13](#_Toc63033352)

[Figura 6. Unifilar Caso de Estudio 13](#_Toc63033353)

[Figura 7. Configuración de un CheckBox de la interfaz 14](#_Toc63033354)

[Figura 8.Información Paquete Capturado con Wireshark 15](#_Toc63033355)

[Figura 9. Campos en mensaje SV 15](#_Toc63033356)

[Figura 10.Dirección Origen y destino mensaje Capturado. 16](#_Toc63033357)

[Figura 11. APPID mensaje Capturado 16](#_Toc63033358)

[Figura 12. Carga útil mensaje Capturado 17](#_Toc63033359)

[Figura 13 Schematic editor para implementar comunicación SV 17](#_Toc63033360)

[Figura 14. Datos de entrada para el sistema el caso de estudio 18](#_Toc63033361)

[Figura 15. Esquema de Pruebas 19](#_Toc63033362)

[Figura 16. Caso de estudio implementado en Power Factory 20](#_Toc63033363)

[Figura 17. Script de Python para uso no interactivo de Power Factory 21](#_Toc63033364)

[Figura 18. Ejecución Power Factory modo No Interactivo 22](#_Toc63033365)

[Figura 19. Sección de medidores del estimador de estado 23](#_Toc63033366)

[Figura 20. Sección de estados estimados 24](#_Toc63033367)

[Figura 21. Unifilar del caso de estudio con los estados de cada nodo 25](#_Toc63033368)

[Figura 22. Estimación de estado con Typhoon Virtual HIL. 26](#_Toc63033369)

[Figura 23. Configuración de Schematic editor para enviar información 27](#_Toc63033370)

[Figura 24. Schematic editor con bloques para enviar información mediante SV 27](#_Toc63033371)

# Introducción

Monitorear el estado de una red eléctrica es de muy alta importancia para la detección oportuna de situaciones anormales de funcionamiento en el sistema de potencia, esto permite tomar medidas de protección y control ante una eventualidad que ponga el riesgo el funcionamiento del sistema.

El monitoreo de la red puede realizarse con modelos de estimación de estado que utilizan información suministrada por equipos de medición ubicados únicamente en algunos nodos del sistema, para calcular los valores de voltaje y corriente en todos los nodos de la red.

Entre los equipos de medición están las PMUs, las cuales, son dispositivos de medición fasorial con capacidad de tomar datos entre 30 y 60 veces por segundo en sistemas de 60 Hz, a diferencia de otros equipos de medición que muestrean 4 a 6 veces por segundo para un sistema similar. Por tanto, hacer uso de dispositivos PMU en los nodos de la red, permite evidenciar una dinámica del sistema que no es posible obtener mediante el uso de otro dispositivo de medición, por esto, usar la información recopilada por PMUs para estimar el estado de la red, permite tener una mejor aproximación de los valores de corriente y voltaje en los nodos que no cuentan con dispositivos de medición.

La Universidad de los Andes cuenta con la herramienta GridTeractions, que es una plataforma de simulación escalable que permite probar estrategias avanzadas de automatización de distribución en redes inteligentes. La herramienta GridTeractions está enfocada en conducir procesos de aprendizaje en temas relacionados con sistemas de energía, por tanto, tiene la capacidad de hacer el modelado detallado, para emular el comportamiento de sistemas reales.

Gridteractions permite la comunicación a través de una interface hombre-máquina desarrollada en un microprocesador Raspberry PI 2, para adquirir información de los demás terminales disponibles. Las simulaciones en la plataforma GridTeractions, son desarrolladas en un script de Python y administradas con el programa Typhoon HIL Control Center, lo que permite ejecutar simulaciones en series de tiempo y obtener resultados en dominio de fase, para facilitarle al usuario evaluar continuamente las variables eléctricas.

Actualmente, la Universidad de los Andes tiene un convenio de cooperación con Queen´s University at Belfast (QUB), el cual brinda la posibilidad de utilizar OpenPMU y PMUs físicas, para la integración de esos equipos de simulación y medición, con la plataforma GridTeractions.

Con el uso de las plataformas de simulación que la Universidad de los Andes tiene disponible para sus proyectos, se inicia la investigación en estimación de estado aplicada, para un sistema modelable en GridTeractions, haciendo uso de la herramienta PMU virtual suministrada por el convenio con Queen´s University at Belfast, para desarrollar este proyecto. Para lograr lo anterior se plantean los objetivos descritos a continuación.

## **Objetivos Generales**

Implementar un estimador de estado que haga uso de mediciones obtenidas a partir de simulación en GridTeractions con base en PMU virtual.

## **Objetivos específicos**

* Implementar en GridTeractions un algoritmo de estimación de estado con base en mediciones fasoriales obtenidas de PMU virtual.
* Modelar y simular la operación del sistema de prueba de 6 barras de Wood and Wollenberg [1], en GridTeractions.
* Integrar la PMU virtual en la plataforma de simulación de GridTeractions; con el propósito de tomar mediciones virtuales que sean utilizadas en un centro de control para propósitos de estimación de estado.
* Desarrollar un documento de difusión tipo artículo.

# Estimación de Estado

La estimación de estado es el proceso de asignar un valor de estado a una variable desconocida del sistema, basado en las medidas tomadas acorde a algunos criterios de selección. Usualmente, el proceso contiene medidas imperfectas que son redundantes, lo que conlleva a que la estimación de estado del sistema sea basada en un criterio estadístico que estima el verdadero valor de las variables a minimizar o maximizar dependiendo de la selección del criterio. Un comúnmente y familiar criterio usado, es la minimizar la suma de los cuadrados de la diferencia entre el valor estimado y el verdadero [1].

En un sistema de potencia, los estados variables son las magnitudes de voltaje y sus fases de los voltajes relativas al nodo holgura (nodo “slack”). Por tanto, estas medidas son requeridas para estimar el rendimiento del sistema en tiempo real, para el control de la seguridad y limitaciones del despacho económico. Usualmente, las entradas de un estimador de estado son medidas imperfectas del sistema de potencia de las magnitudes de voltaje, potencia activa y reactiva, o flujo de corriente. El estimador es diseñado para producir el “mejor estimador” de los voltajes y ángulos de fase del sistema, reconociendo que estos contienen errores en las medidas cuantitativas y que a la vez sean medidas redundantes. Los datos de salida se utilizan en los centros de control en la implementación del envío y control del sistema con restricciones de seguridad [1].

Como se menciona anteriormente, el monitoreo y control del flujo de potencia y voltajes en un sistema de transmisión es muy importante para mantener la seguridad en el sistema. Por simple verificación del valor medido contra su límite, el operador del sistema de potencia puede decir donde existe problemas en el sistema de transmisión y con esto se espera, que puedan tomar correctivas acciones que alivien la líneas sobrecargadas o voltajes fuera de límites [1].

## **GridTeractions**

GridTeractions es una plataforma simuladora de sistemas de transmisión y distribución que permite el control remoto y distribuido de componentes eléctricos basado en una arquitectura escalable multiplataforma, tal como se muestra en la Figura 1. En la arquitectura del sistema se puede encontrar un servidor que se ejecuta en Typhoon Hil 402 y envía información a varios clientes, cada uno de los cuales se ejecuta en una Raspberry PI 2. El Raspberry PI es un dispositivo útil para aplicaciones basadas en la comunicación cliente y servidor con arquitecturas de múltiples terminales. El servidor y los clientes se comunican a través del protocolo TCP / IP sobre LAN [2].

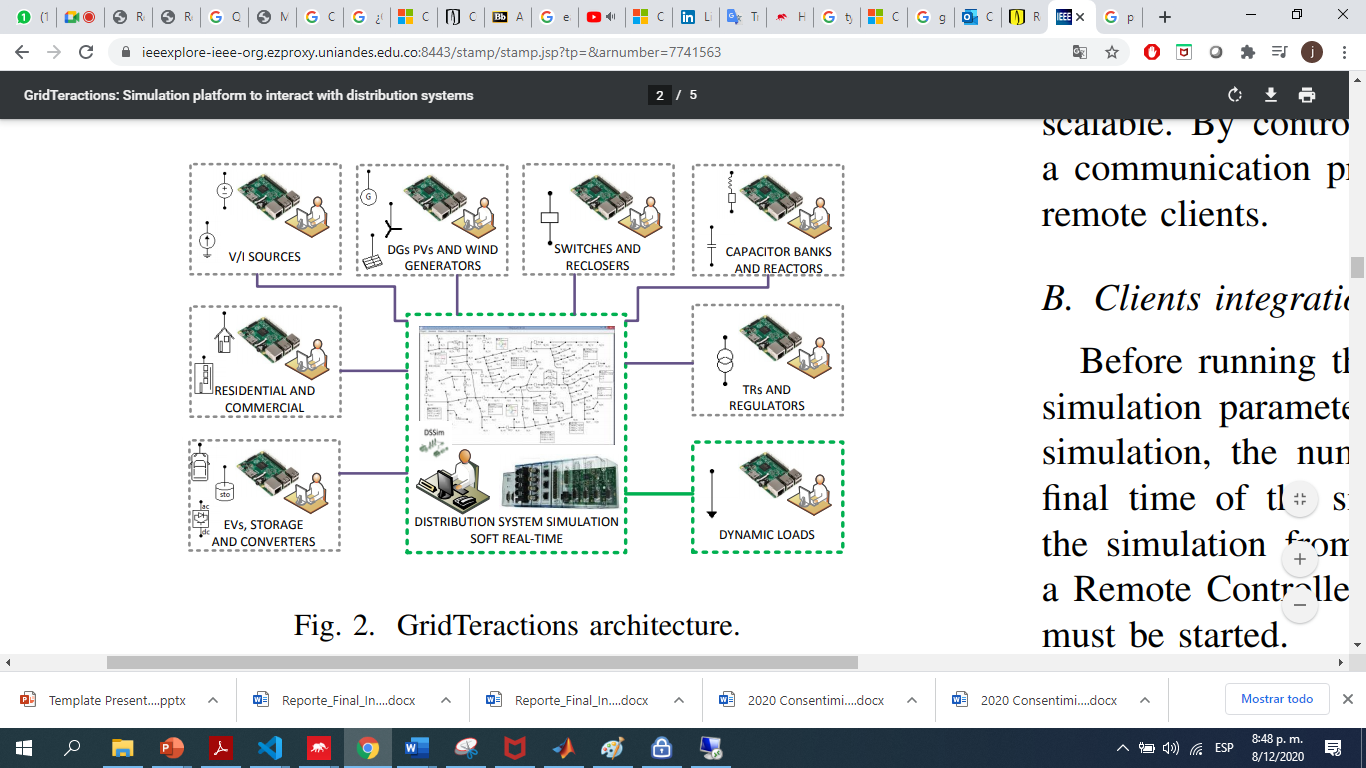


Figura 1. Arquitectura GridTeractions [2].

La plataforma emplea el software Power Factory [3] como simulador del sistema de transmisión para obtener resultados continuos del flujo de energía en el dominio de fase. Este motor, desarrollado por DigSilent, incluye una galería de modelos con dispositivos básicos para el funcionamiento de estrategias de automatización en sistemas de distribución, como interruptores, reguladores, contadores, etc.

El servidor, que se ejecuta en Windows 10 embebido, está implementado en lenguaje gráfico en uno de los ambientes de desarrollo de Typhoon. El sistema de transmisión se simula en Typhoon y se comunica con el servidor a través de la biblioteca de conexión con cada uno de sus ambientes de trabajo. La comunicación entre el servidor y los clientes se basa en un protocolo personalizado diseñado para satisfacer las necesidades del sistema utilizando el protocolo TCP / IP sobre una red local Ethernet [2].

El servidor tiene tres procesos principales para recibir comandos y enviar información al cliente, y un proceso final diseñado para administrar los tiempos de simulación, en el proceso final se pueden implementar estrategias de control procesando todos los datos disponibles desde el Typhoon Hil Scada y los clientes del sistema. Cada uno de los procesos del servidor está codificado como una máquina de estado. De esta forma, la implementación de nuevas funciones o dispositivos es altamente escalable. Controlando estos estados es posible definir un protocolo de comunicación para intercambiar información con los clientes remotos [2].

## **Estimación de estado por mínimos cuadrados ponderados (WLS)**

A partir de la definición de estimación de estado se tienen en cuenta que el conjunto de medidas tomadas del sistema para determinar el estado de cada uno de sus nodos busca obtener los valores más probables del sistema a partir de la información y los supuestos definidos. Por este método, el estimador de estado se convierte en un estimador de mínimos cuadrados ponderado con la inclusión de la matriz de covarianza de error de medida que sirve para ponderar la precisión de cada una de las medidas. La información y las mediciones del modelo del sistema físico se usan para definir las restricciones de igualdad de la optimización básica de mínimos cuadrados ponderados y hace que este estimador sea usado para sistemas de energía [1].

### Algoritmos WLS

La estimación de estado con el algoritmo WLS, consiste en generar los valores estimados con la minimización de la suma de los cuadrados de los errores, A continuación, se da la descripción del algoritmo. Inicialmente se tiene un vector de mediciones Z con m número de medidas y se considera un vector de estado X con las variables de estado [4], como:

Para este método, se emplea los conjuntos de medidas cómo funciones no lineales del vector de estado del sistema y se denotan estas funciones en un vector :

Con estas funciones, luego de ser evaluadas en el estado real del sistema, producirán un conjunto de mediciones con los valores reales de las mediciones. Sin embargo, todas estas medidas tienen cada una su propio error desconocido asociado, denotado por y mostrado en forma vectorial.

Se asume que los errores de medición son independientes entre sí y tiene un valor esperado de cero. Con lo anterior, se puede escribir la ecuación de estado completa.

La solución al problema de estimación de estado se puede formular como una minimización de la siguiente función objetivo:

Esto se puede reescribir de la siguiente forma.

Donde R es la matriz de covarianza de los errores de medición y tiene una estructura diagonal. Cada uno de los elementos diagonales es la covarianza de su medida respectiva y todos los elementos fuera de la diagonal son cero porque se supone que las medidas son independientes. Para encontrar la minimización de esta función objetivo, la derivada debe establecerse en cero. La derivada de la función objetivo se denota por g (x):

Para aplicar este algoritmo, la única información requerida para resolver es la matriz de covarianza de errores de medición, R, y la función de medición, h (x). La medida jacobiana, H (x) es simplemente la derivada de las funciones de medida con respecto al vector de estado. La función de medición y el jacobiano de medición se pueden construir utilizando el modelo de sistema conocido, incluidos los parámetros de rama, la topología de la red y las ubicaciones y el tipo de medición. La matriz de covarianza de error también debe construirse antes de las iteraciones con la información de precisión de los medidores instalados en el sistema.

Para la primera iteración de la optimización, la función de medición y el jacobiano de medición deben evaluarse con un perfil de voltaje o un inicio planos. Un inicio plano se refiere a un vector de estado donde todas las magnitudes de voltaje son 1,0 p.u. y todos los ángulos de voltaje son 0°. Junto con las mediciones, la siguiente iteración del vector de estado se puede calcular una y otra vez hasta que se alcance la tolerancia deseada.

## **Estimación de Estado por Valor mínimo Absoluto (LAV)**

Aunque la estimación por mínimos cuadrado es un método bien conocido y ampliamente utilizado para la estimación de estados, no es un estimador robusto y es necesario realizar pos-procesamiento de los residuos de medición para el análisis de datos incorrectos, lo cual puede llegar a ser computacionalmente más costoso. El uso de estimador LAV puede llegar a competir computacionalmente con el estimador de mínimos cuadrados. El estimador LAV tiene como objetivo minimizar la suma de valores absolutos de los residuos de medición [5].

### Algoritmos LAV

Se define la función objetivo del estimador LAV, debido a que el problema de estimación de estado se puede formular de manera idéntica para los problemas de secuencia cero y positiva / negativa. La formulación se describirá solo para la secuencia cero con el entendimiento de que expresiones idénticas serán válidas también para los otros componentes modales [5].

Donde

y partes reales e imaginarias del componente de secuencia cero de la j-esima medición residual, respectivamente. partes reales e imaginarias de la componente de secuencia cero del valor medido en el sistema. partes reales e imaginarias del componente de secuencia cero del j-esimo valor de medición calculado con estados estimados.

es el vector de estado estimado, y son las partes real e imaginaria de la j-esima variable de estado, es decir, componentes de secuencia cero del bus del fasor de voltaje, respectivamente. El problema de estimación de LAV se puede formular de la siguiente manera.

El problema de optimización de LAV dado se puede expresar como un problema de programación lineal equivalente (LP) reordenando las ecuaciones y definiendo algunas nuevas variables estrictamente no negativas.

## **Estimación de Estado Lineal**

Durante la mayor parte del desarrollo de los estimadores de estado tradicionales, prevaleció la idea de que la recopilación simultánea precisa de mediciones en todo el sistema era algo que nunca podría lograrse [4]. Una gran suposición que mantenía juntas todas las técnicas tradicionales de estimación de estado era que el estado estático del sistema de energía cambiaba muy lentamente y los operadores podían permitirse tener tiempos de exploración significativos. Aunque algunos estimadores actuales tienen tiempos de exploración de solo unos segundos, esto podría ser una eternidad para varias aplicaciones deseables en protección y control. Las PMU permiten la recopilación sincronizada de medidas fasoriales y, con esta tecnología que se está volviendo tan frecuente en las empresas de servicios públicos, es inevitable que se utilice para aplicaciones de estimación de estado [4].

La inclusión de la tecnología PMU en la estimación del estado puede presentarse de varias formas. Las mediciones de PMU pueden incluirse mediante una formulación ligeramente diferente de los mínimos cuadrados ponderados no lineales tradicionales o pueden tenerse en cuenta después de que ya se haya determinado un estado preliminar del sistema [6]. Incluso un pequeño número de estas medidas precisas puede pesar mucho en la precisión del estado general del sistema [7]. Sin embargo, una verdadera aplicación de la tecnología PMU a la estimación de estados tendría todas las mediciones tradicionales de inyecciones de potencia real y reactiva y magnitudes de corriente y voltaje reemplazadas por fasores de voltaje de bus y fasores de corriente de línea. Si solo se utilizan mediciones de PMU, tampoco hay complicaciones por el uso de valores polares y rectangulares en el proceso de estimación de estado, como se haría al incluir mediciones de PMU en estimadores de estado tradicionales [4].

### Equivalencia del modelo π

Para comprender la diferencia fundamental entre las medidas utilizadas en un estimador de estado tradicional y las medidas utilizadas en un estimador de estado lineal, lo mejor es comenzar con el modelo π simple de dos puertos equivalente a una línea de transmisión, el cual se presenta en la Figura 2. El estado de este sistema simple será la magnitud del voltaje y el ángulo en cada extremo de la línea de transmisión. Si hay una PMU en cada extremo de la línea de transmisión, se puede suponer que el conjunto de medidas para este sistema consistirá en los fasores de voltaje en cada extremo de la línea y los flujos de línea que salen de cada extremo de la línea [4].

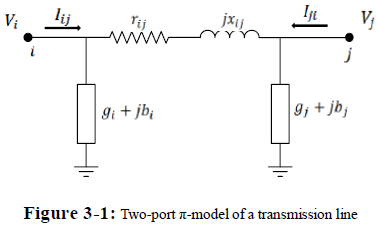


Figura 2 Modelo π de una línea de transmisión [4].

Todos los valores se considerarán rectangulares porque, en el nivel más básico, esto es lo que devolverá la PMU [8]. Esto es lo que le da a la ecuación de estado su propiedad lineal. El estado del sistema es entonces el siguiente vector complejo donde corresponde al voltaje en el nodo i del sistema y al voltaje en el nodo j del sistema.

### Estimación con medidas Fasoriales

La estimación de estado con medidas fasoriales se desarrolla inicialmente con el cálculo del equivalente cartesiano de la medición, que se presenta a continuación:

Se considera que el conjunto de medidas sin errores es la concatenación vertical de los fasores de voltaje en cada extremo de la línea de transmisión y la corriente de la línea fluye desde cada extremo de la línea de transmisión; así:

El estado del sistema claramente puede relacionarse de manera idéntica con las mediciones de voltaje en este vector complejo. Sin embargo, la relación lineal entre el estado del sistema y los flujos de la línea requiere cierto esfuerzo. Primero, se deben definir varias cantidades. La admitancia en serie y la susceptancia en derivación de la línea de transmisión son las siguientes:

La relación entre el estado del sistema y los flujos de corriente de línea para el caso de una línea de transmisión simple es la siguiente:

Entonces, se completa la matriz de estado de tal forma que se tengan en cuenta los voltajes en cada uno de los nodos que tengan medidores y se generaría la siguiente ecuación de estado:

# Implementación de estimador de estado en GridTeractions

La Implementación del estimador de estado lineal se hizo mediante el uso del lenguaje de programación Python, en donde fue posible hacer las pruebas correspondientes para comprobar que el estimador de estado cumpliera con los parámetros necesarios para ser integrado con GridTeractions. Los principales requerimientos que el código debía cumplir son:

* Sea posible implementar el código en el HIL Scada de Typhoon.
* El estado de los nodos donde se ubican las PMUs será ingresado al código por parámetros.
* El código reconozca la cantidad de medidores que tiene el sistema, ya que este valor puede variar dependiendo de la disponibilidad que se asigne a cada nodo.
* Genere un vector con los estados estimados para todos los nodos del sistema.

## **HIL Scada de Typhoon**

Ahora se muestra cómo se adaptó el HIL Scada de Typhoon, haciendo uso de las herramientas que el programa tiene disponible en este ambiente. En la Figura 3, se presenta el Typhoon Virtual HIL Scada con todos los componentes que se configuraron para que el programa pueda mostrar los estados estimados por el código desarrollado.

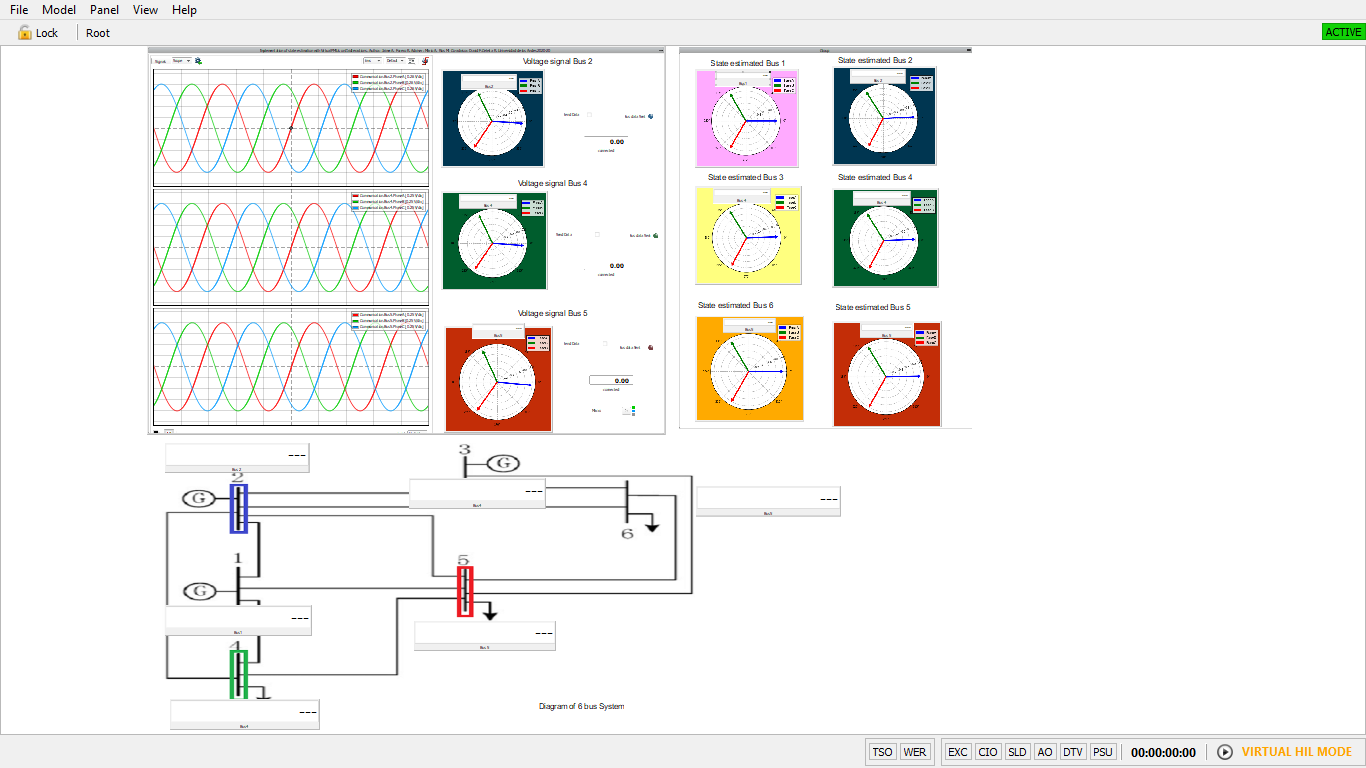
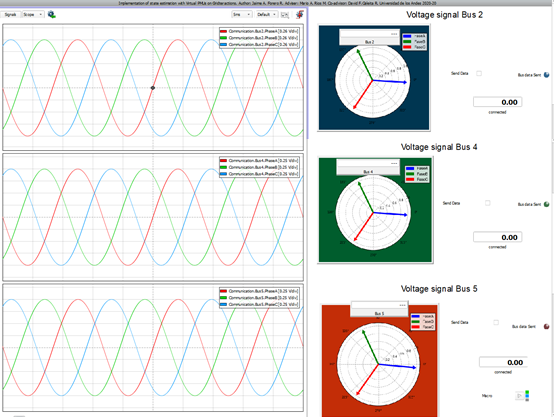


Figura 3. Typhoon Virtual Hil Scada

La ventana se divide en tres secciones. La primera sección muestra el estado de los nodos 2, 4 y 5, en esta sección se grafica las señales para cada una de las fases de los nodos a los cuales se les asigno un medidor, además junto a cada señal se puede encontrar un widget de tipo checkbox, el cual se debe seleccionar para que el estimador de estado tenga en cuenta las mediciones de ese nodo para hacer los cálculos y generar los resultados. En la Figura 4, se puede ver las componentes que se nombraron anteriormente.



**Scope Nodo 2**

**Scope Nodo 4**

**Scope Nodo 5**

**Graph Phasor**

**Graph Phasor**

**Graph Phasor**

**CHECKBOX**

**CHECKBOX**

**CHECKBOX**

Figura 4. Estados de los nodos 2, 4 y 5.

En la segunda sección se configura el widget Graph Phasor para cada uno de los nodos, en esta sección es posible ver el estado estimado por el programa a partir de las mediciones fasoriales seleccionadas del sistema, por lo que hay dos tipos de Widgets, uno muestra gráficamente los fasores de los estados del bus y el segundo es un display donde se escribe cada uno de los valores de magnitud y fase en los nodos. En la Figura 5, se muestran las componentes.

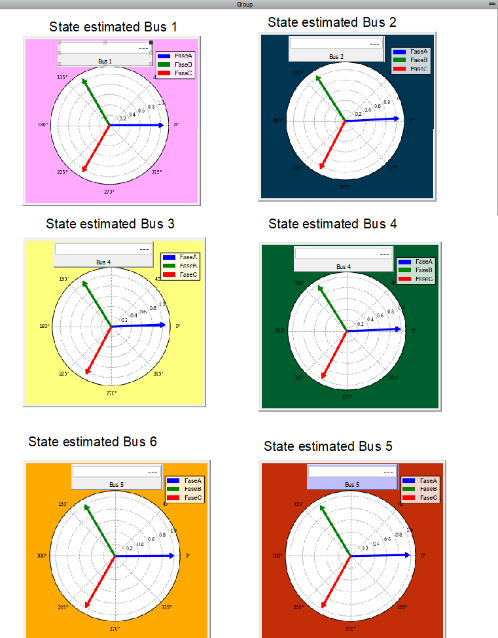


Figura 5. Estados Estimados

Por último, en la Figura 6 se presenta un unifilar del sistema al cual se está haciendo la estimación de estado a los nodos del sistema, allí se marcaron los nodos que tendrán la posibilidad de suministrar datos al programa para hacer la estimación de estado. Este unifilar está en el Typhoon virtual HIL Scada y se podrá ver el estado de cada uno de los nodos del sistema.

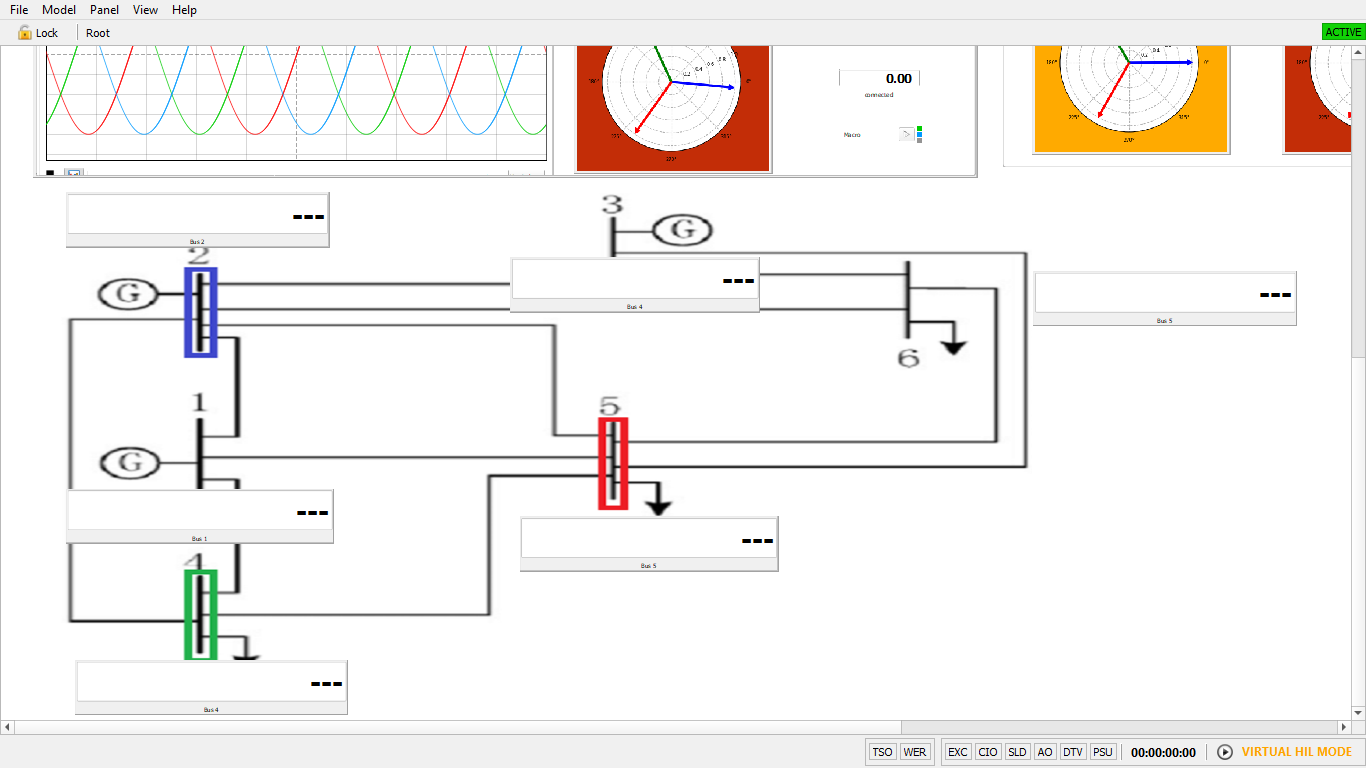


Figura 6. Unifilar Caso de Estudio

## **Configuración del Typhoon Virtual HIL Scada con el código de Estimación de estado lineal**

Luego de tener el código en Python se procedió a vincular el estimador de estado con el HIL Scada de typhoon, para esto se usó el widget *macro* que tiene disponible el programa para vincular en la interfaz, allí se pasa el código del estimador de estado y el widget se encarga de reproducirlo cada 250ms, esto con el fin de actualizar las señales si hay un cambio en la red debido a una falla. Por otro lado, cómo se muestra en la Figura 7 es necesario configurar cada una de los *checkbox*, ya que con esto es posible tomar los valores de las mediciones para ingresarlos al *macro*. Luego, se configuró cada uno de los Graph Phasor para que, a partir de los estados estimados seleccione el que corresponde a cada uno de los nodos del sistema y de esta forma se pueda ver en la interfaz los valores de los fasores.

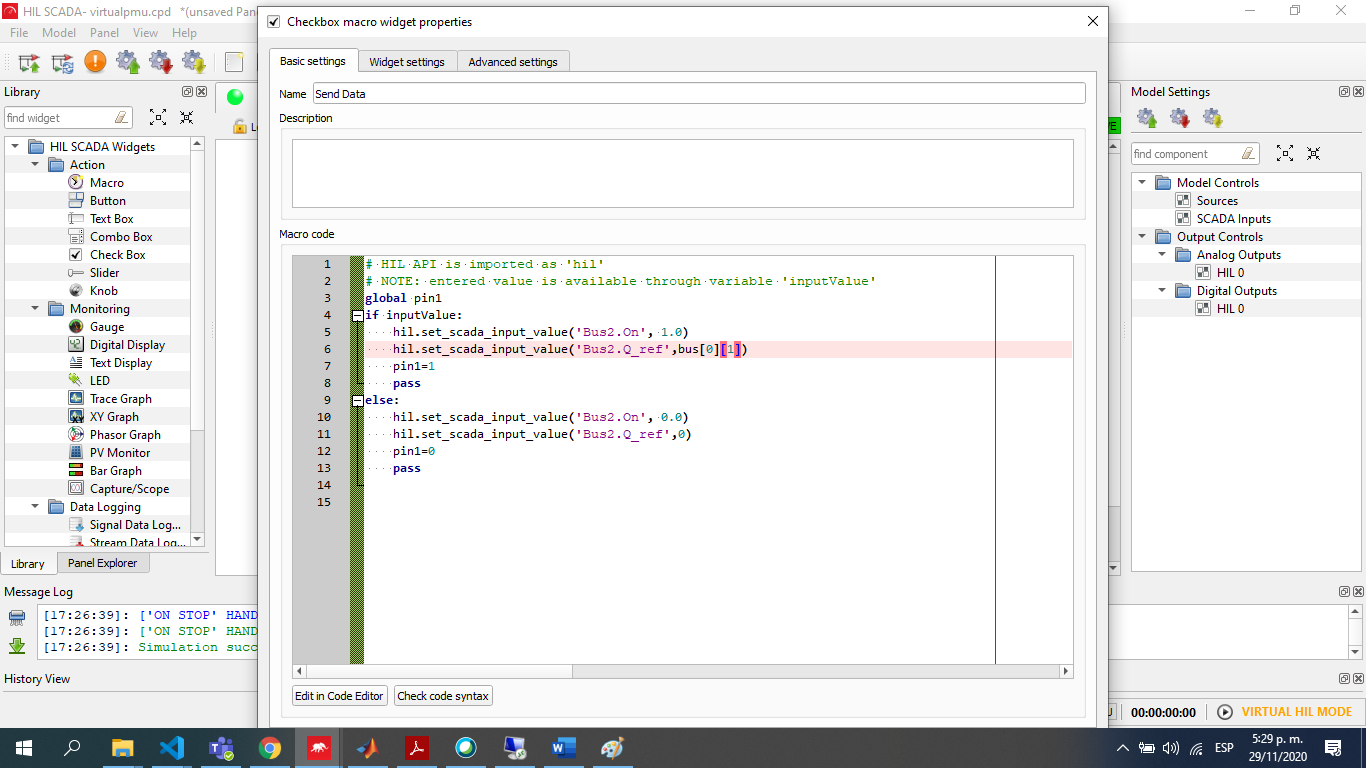


Figura 7. Configuración de un CheckBox de la interfaz

## **Comunicación mediante IEC 61850**

Cómo se planteó inicialmente, es necesario que sea posible enviar los valores estimados mediante una comunicación que cumpla los parámetros del estándar IEC 61850, por tal razón se usa el bloque que tiene disponible Typhoon para comunicarse mediante Sampled Values(SV). En el estándar IEC 61850 se encuentra este protocolo, para entablar una comunicación tipo Publisher/Suscriber con otras máquinas mediante sus puertos Ethernet. Para este proyecto se seleccionó este protocolo de comunicación, ya que es comúnmente usado en el intercambio de datos entre las maquinas instaladas en las subestaciones digitalizadas [9].

La comunicación con el protocolo SV consiste en asignar una maquina cómo Publisher y con ella enviar mensajes periódicamente. Los mensajes son enviados con una descripción, la cual permite que las maquinas que están configuras con Suscriber, seleccionen los mensajes donde la descripción del mensaje coincida con las de ella, así cada máquina filtra y analiza sólo la información que está dirigida a ella.

En el Schematic editor de Typhoon es posible encontrar módulos de comunicación que permiten implementar la comunicación con Sampled Values. Con estos módulos fue posible emular la comunicación y con el programa WireShark se capturaron los paquetes d que tenían de origen el dispositivo Typhoon Hil 402 cómo se muestran en la Figura 8.

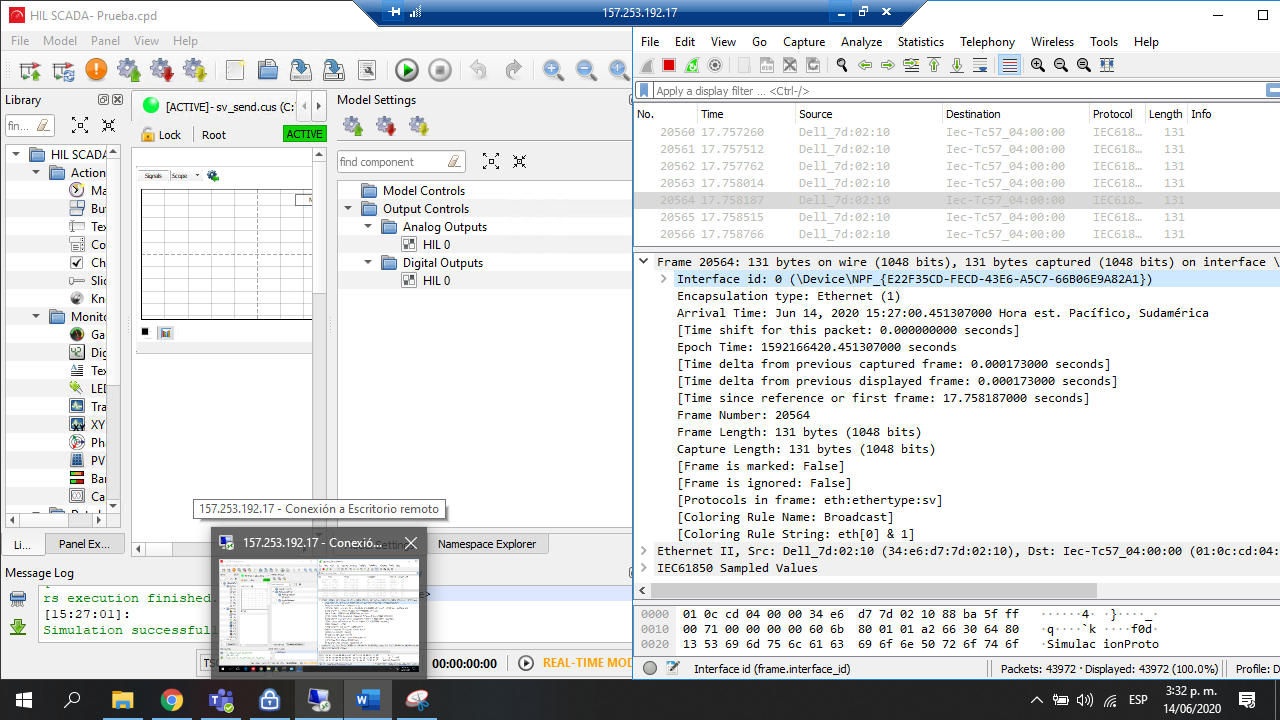


Figura 8.Información Paquete Capturado con Wireshark

La Figura 9 muestra la información que se envía en un mensaje con el protocolo IEC61850.

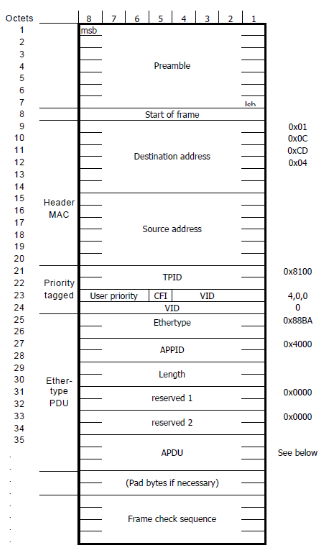


Figura 9. Campos en mensaje SV

Cómo se puede ver en la Figura 9, lo primero que se encuentra en el mensaje es el preámbulo e inicio de trama donde se usan los primeros 7 bits para indicar a las maquinas receptoras que una trama es Ethernet y con un bit adicional se indica que inicia la trama. Posteriormente, se encuentra la dirección de origen y destino, en este espacio solo es posible asignar la dirección Mac que corresponde al equipo Publisher que es el origen del mensaje, y para el destino se puede implementar broadcast único (corresponde a una sola maquina), broadcast múltiple (corresponde a un grupo) o broadcast (corresponde a todos los nodos), según como sea conveniente para la red implementada, en el caso del paquete analizado, el destino es de tipo broadcast único [9].

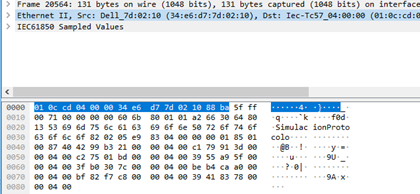


Figura 10.Dirección Origen y destino mensaje Capturado.

Siguiendo con las componentes del mensaje, se encuentran 2 bytes para definir el TPID, el cual es el definido en IEEE802.1Q como 0x8100, con esto es posible identificar el tipo de etiqueta de prioridad que tiene el protocolo. También se debe asignar un valor de EtherType para definir el tipo de protocolo que esta encapsulado en la trama, para el caso del mensaje capturado es 0x8ba y define el protocolo SV. En el mensaje también es necesario definir el tipo de aplicación que tiene la trama, para esto está el APPID el cual está compuesto por dos bytes con 0x4000 y 0x7FFF.

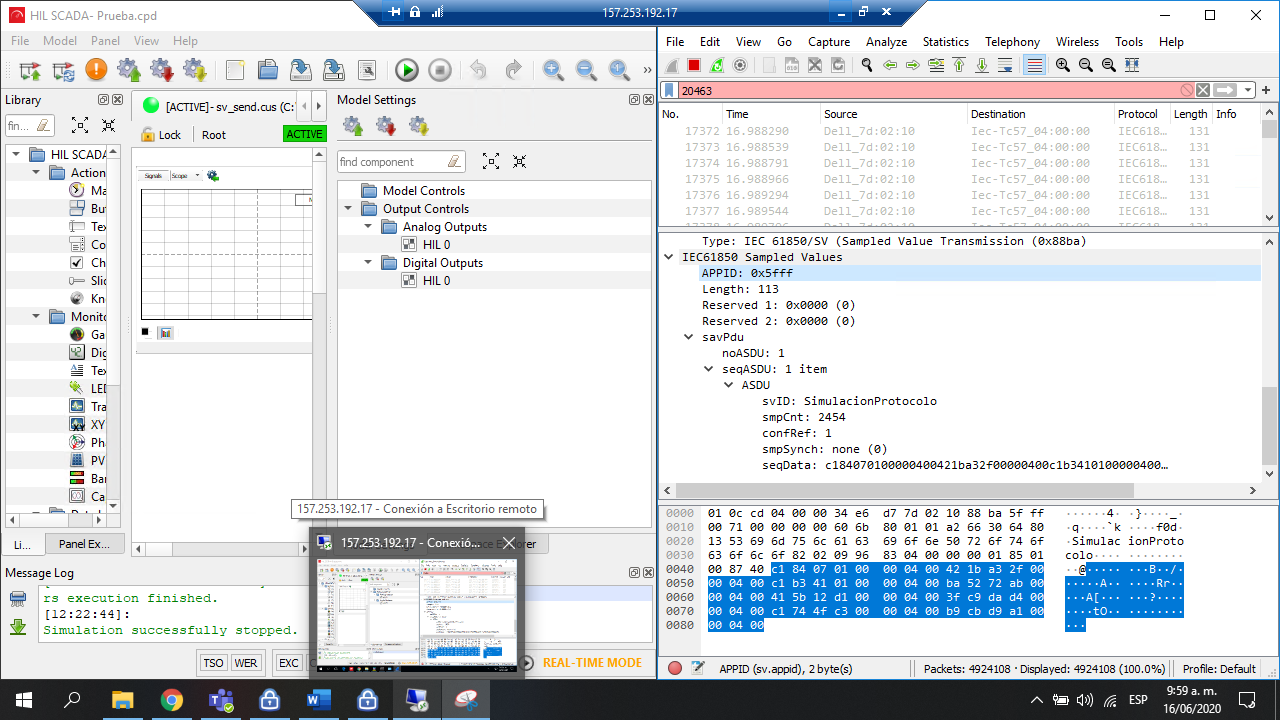


Figura 11. APPID mensaje Capturado

Lo que sigue en el mensaje es el campo APDU, el cual contiene la carga útil de los mensajes SV. En Cada APDU se puede configurar hasta 8 ASDU (Unidad de datos específicos de la aplicación) y cada ASDU tiene un valor de identificación SV único y contiene una medición trifásica de corriente y voltaje, en la emulación que se realizó bajo este protocolo, se configuró SimulacionProtocolo para identificar el mensaje.

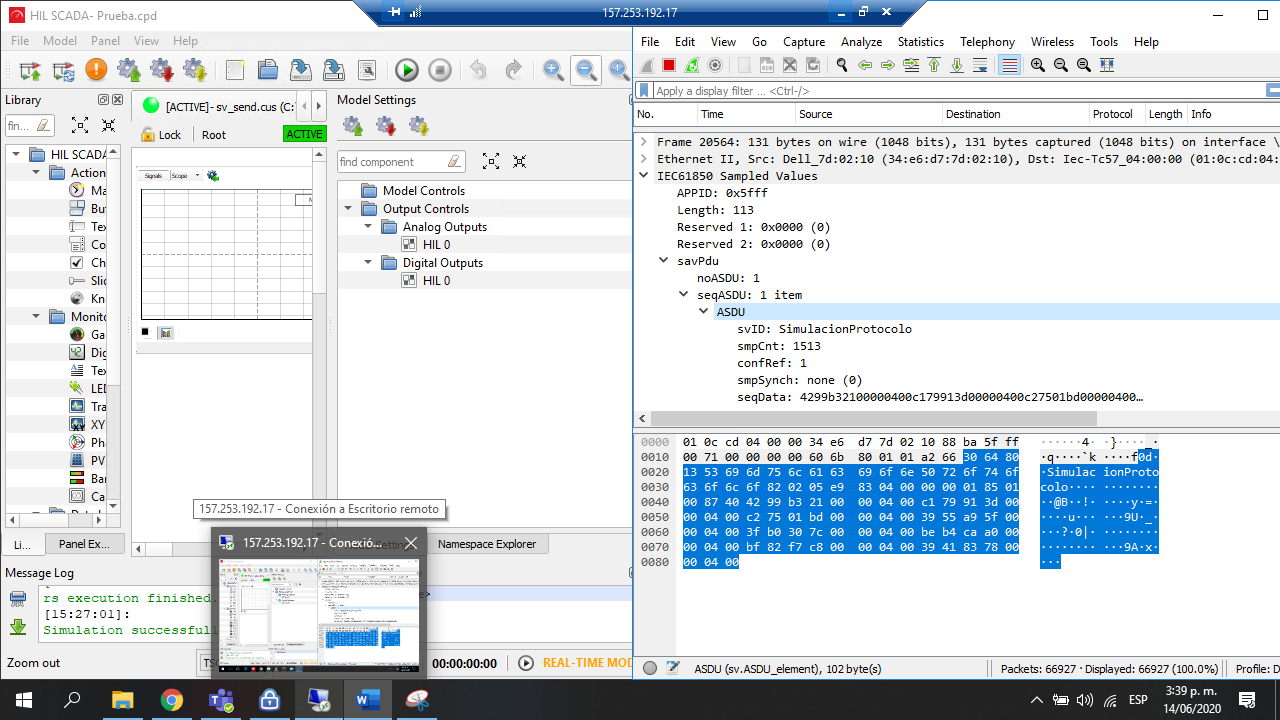


Figura 12. Carga útil mensaje Capturado

Dentro de la información de los ASDU se encuentra SeqData, en donde va la información de voltaje y corriente medidas en la subestación. El esquema para enviar la información de corriente y voltaje con el protocolo SV en Typhoon HIL para cada una de las fases de un nodo, debe ser similar al que se presenta en la Figura 13.

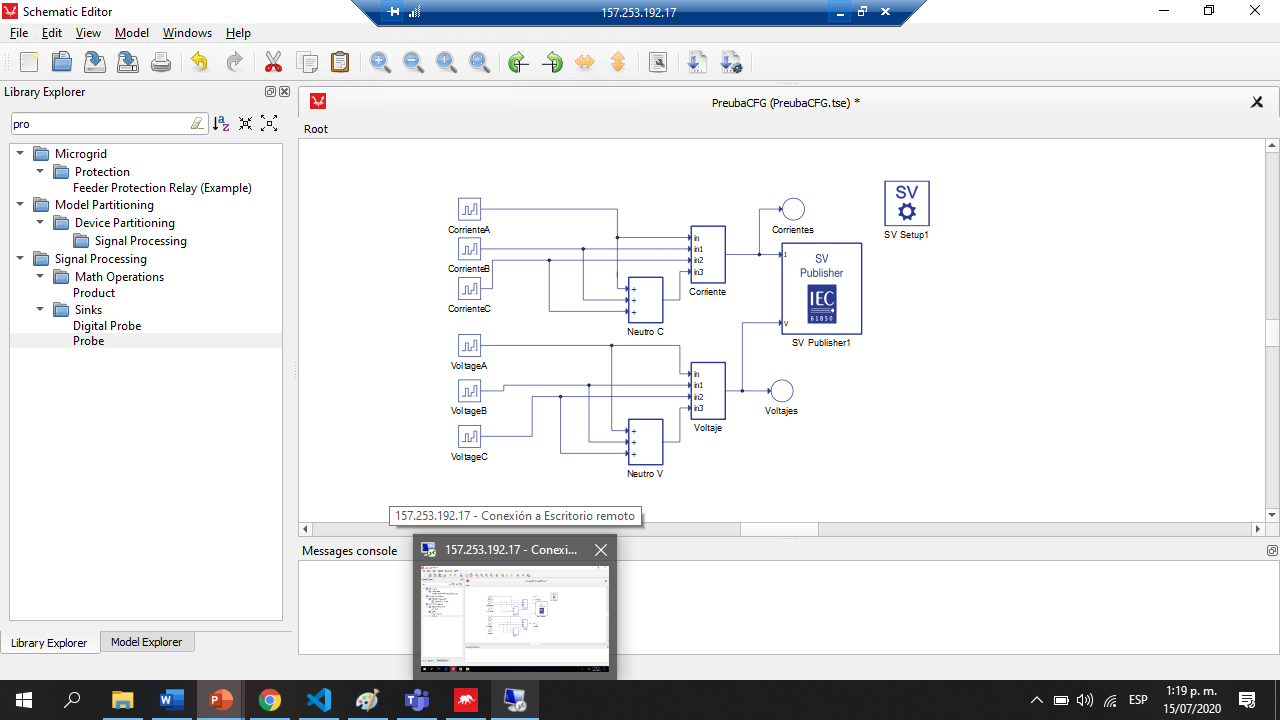


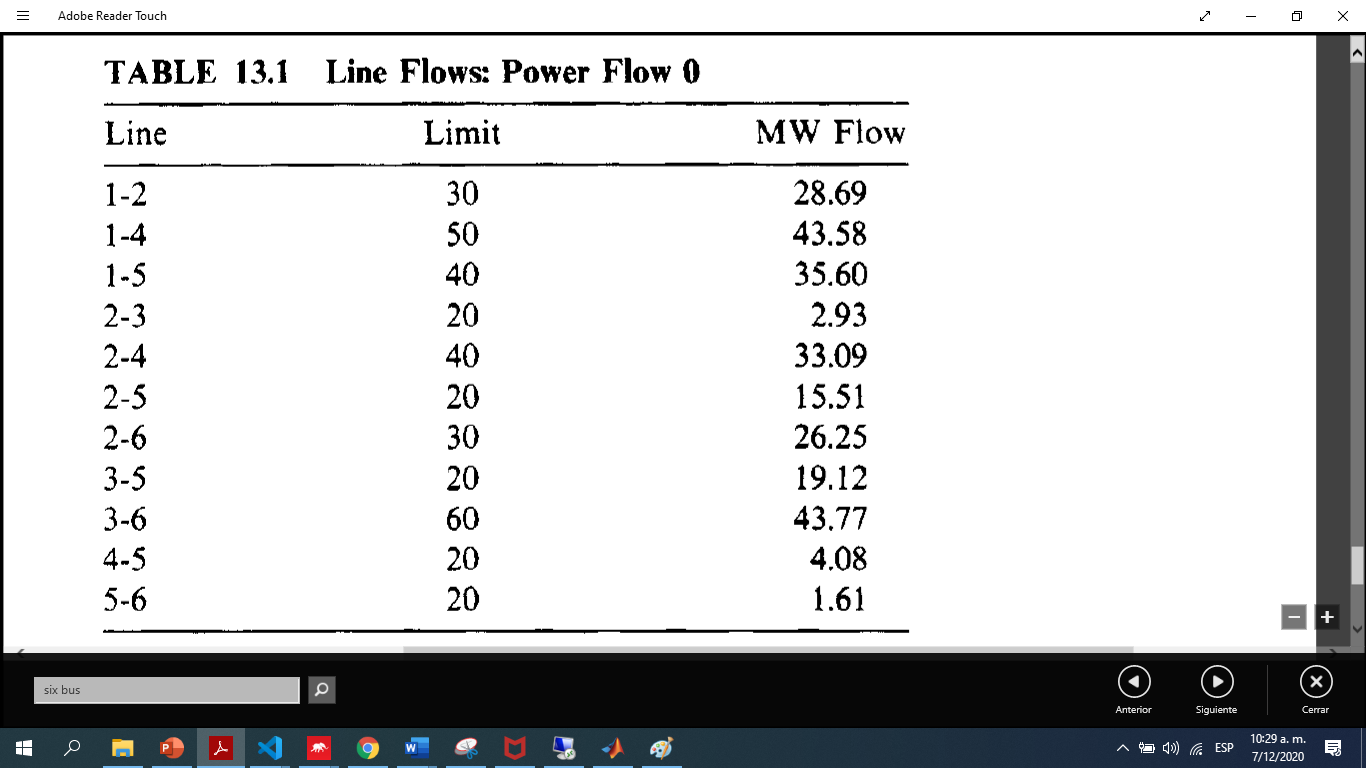
Figura 13 Schematic editor para implementar comunicación SV

## **Caso de Estudio**

El caso de estudio es el sistema de 6 nodos de Wood a Wollenberg el cual presenta la siguiente información.

En la Tabla 1 se presenta la información de los flujos en las líneas de transmisión del sistema de 6 nodos que suministra la referencia [1].

Tabla 1. Flujos de la línea



La información de las líneas de transmisión y las potencias en los nodos también fue posible obtenerlas a partir de los datos que suministra en [1]. La información de la Figura 14 es usada en el estimador de estado para realizar los cálculos.

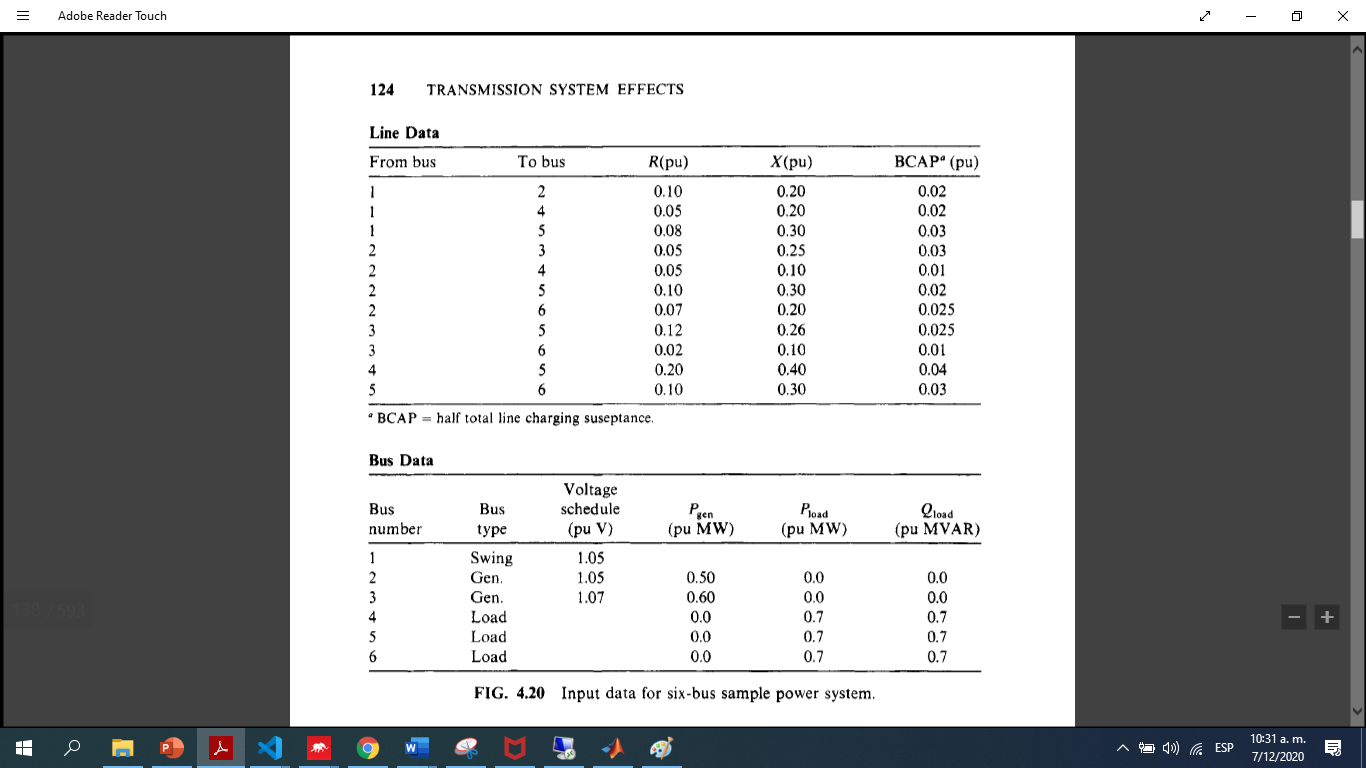


Figura 14. Datos de entrada para el sistema el caso de estudio

# Prueba del estimador de estado.

En esta sección, se presenta los resultados obtenidos luego de implementar el estimador de estado y establecer el esquema de prueba que se presenta en la Figura 15.

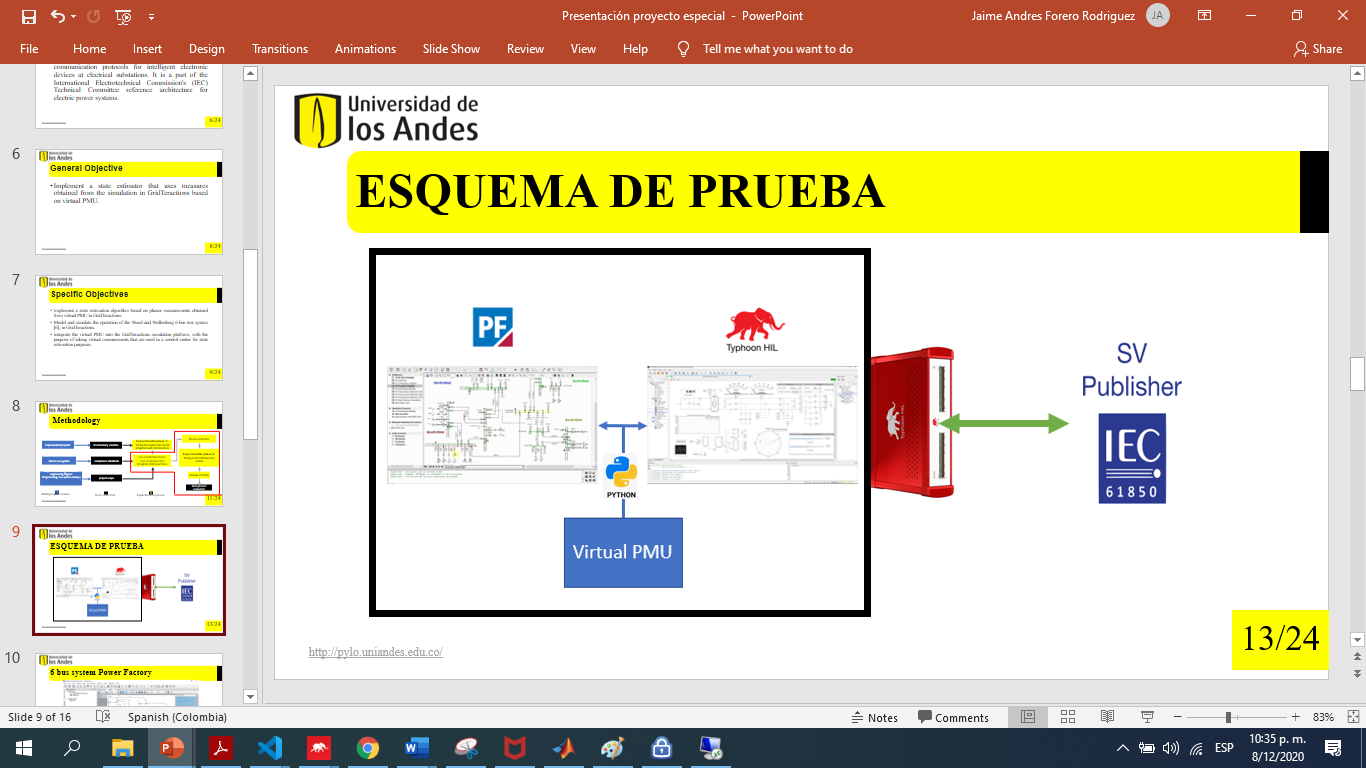


Figura 15. Esquema de Pruebas

## **Comparación de los resultados obtenidos con cada método de estimación de estado.**

Para comprobar la funcionalidad del estimador de estado, se implementaron los tres métodos de estimación de estado en el lenguaje de programación Python, con esto podemos comparar cada uno de los métodos y así decidir si los resultados que tiene un estimador de estados es coherente con el posible estado del sistema. En la Tabla 1 se presentan los resultados para cada uno de los nodos usando los tres métodos de estimación de estado.

Tabla 2. Estimación de estado del caso de estudio con los métodos WLS, LAV y Lineal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nodo | Mediciones | Estimación WLS | Estimación ALV | Estimación Lineal |
| 1 | 1.0500 0° | 1.082 0° | 1.0500 0° | 1.0565 -0.1183° |
| 2 | 1.0500 -3.6710° | 1.0676 -3.9367° | 0.9140 -4.1310° | 1.0516 -3.6767° |
| 3 | 1.0700 -4.2730° | 1.0709 -4.3176° | 1.0405 -4.2540 ° | 1.0723 -4.2929° |
| 4 | 0.9890 -4.1290° | 1.0427 -2.9247° | 1.0020 -3.8770° | 0.9906 -4.1360° |
| 5 | 0.9850 -5.2760° | 1.0103 -5.4631° | 0.9980 -5.3410° | 0.9864 -5.2234° |
| 6 | 1.0040 -5.9470° | 1.0084 -5.9563° | 1.0110 -6.9130° | 1.0064 -6.1290° |

Luego de comparar entre los 3 métodos de estimación de estado y las mediciones de cada nodo en el sistema, se hace la selección del método que se usa en el estimador de estado. Debido a que los métodos de estimación de estado WLS y ALV hacen el cálculo a partir de las potencias del sistema y su complejidad computacional es más alta comparados con el estimador de estado lineal, no se tienen en cuenta para integrarlos con Typhoon.

Cómo se mencionó en capítulos anteriores, el estimador de estado lineal hace el cálculo a partir de las medidas fasoriales de voltaje y corriente, por lo cual es compatible con los requerimientos que se establecieron para integrarlo con GridTeractions. Por otro lado, debido a que se tiene la información del caso de estudio es posible hacer los cálculos de estimación de estado, sin que implique una carga computacional tan alta como con los otros métodos vistos.

## **Ejecución de Power Factory en modo no interactivo.**

Para vincular el sistema implementado en Power Factory con el estimador de estado, se usó la herramienta que tiene disponible este programa, para trabajar con proyectos desde Python sin abrir el programa directamente y así disminuir el tiempo en que se hace esta integración. Por tanto, el caso de estudio se desarrolló en este programa para luego importar toda la información requerida desde Python. En la Figura 16 se presenta el sistema implementado en power Factory.

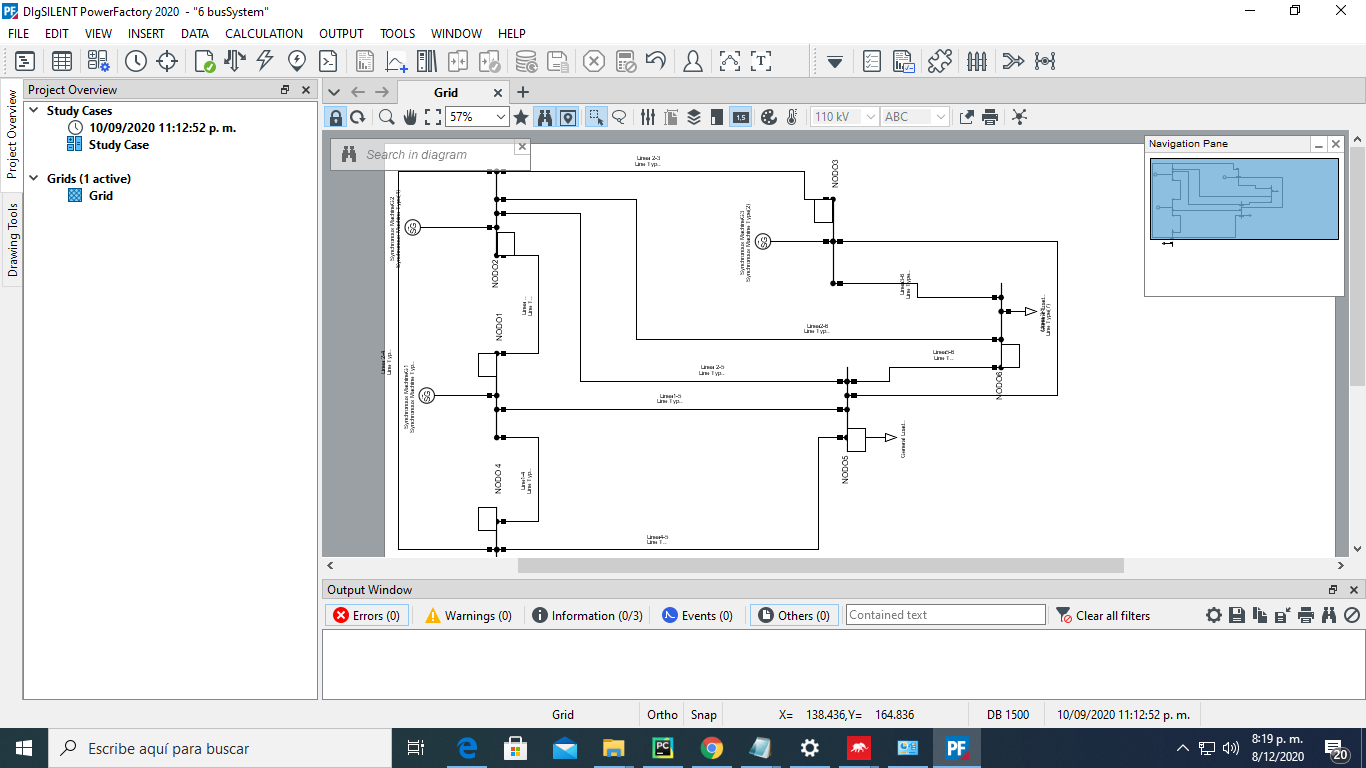
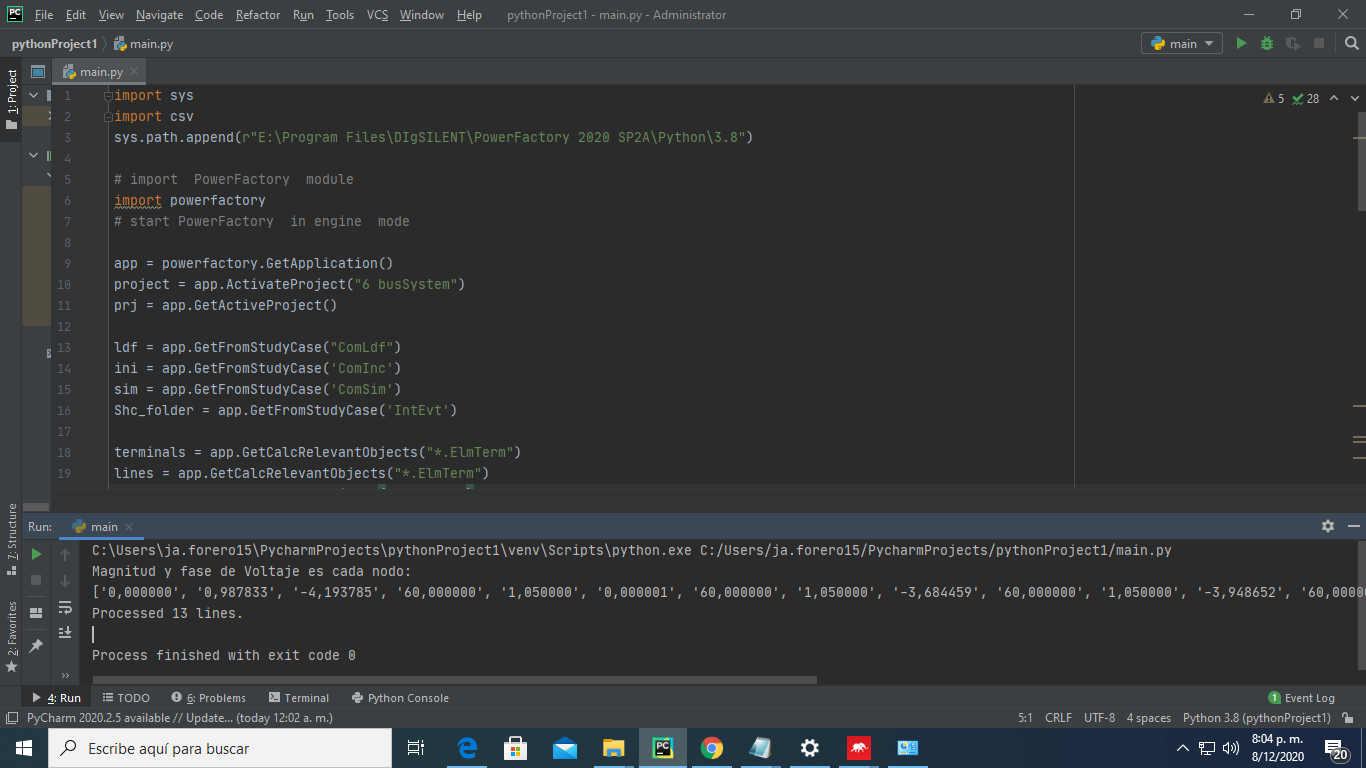


Figura 16. Caso de estudio implementado en Power Factory

Al revisar la información de este programa, se encontró que hay un módulo en las carpetas instaladas de Power Factory que permite ejecutar funciones del programa desde un script de Python. Posteriormente, se procedió a llamar el proyecto donde se realizó el montaje del caso de estudio y así se obtuvo la información en cada uno de los nodos del sistema. Las ventajas de hacer esta integración se verán reflejadas en los casos que se quiera activar algún tipo de breaker en el proyecto y esto se podrá hacer desde la interfaz en Typhool Hil Scada. Entonces desde el estimador de estado se podrá probar los comportamientos del caso de estudio cuando se configure alguna falla o casos de contingencias en las líneas. En la Figura 17 se presenta la forma en que se usa el módulo de Power Factory, se usan las funciones que el módulo provee al usarlo y los valores de magnitud y ángulos del sistema, los cuales son los que se toman cómo mediciones en la Tabla 2.



Funciones aplicadas al proyecto

Valores de Magnitud y Fase en cada Nodo

Modulo Power Factory

Figura 17. Script de Python para uso no interactivo de Power Factory

En la Figura 18, se presenta el diagrama de flujo que resume las actividades que se llevan a cabo para obtener los estados de los nodos en los sistemas de potencia implementados con Power Factory, esto para integrar el programa con el estimador de estado que se está desarrollando.

Diagrama

Descripción generada automáticamente

Figura 18. Ejecución Power Factory modo No Interactivo

## **Estimación de Estado en Typhoon Virtual HiL Scada**

Con el caso de estudio implementado en Power Factory y haciendo uso de la API que tiene disponible este programa para importar los datos con Python, se implementó el estimador de estado con la información proveniente del caso de estudio y de esta forma fue posible sincronizarlo con la interfaz que se presenta en la Figura 19, la cual está implementada en Typhoon y será el ambiente donde se va a realizar la operación del estimador de estado.

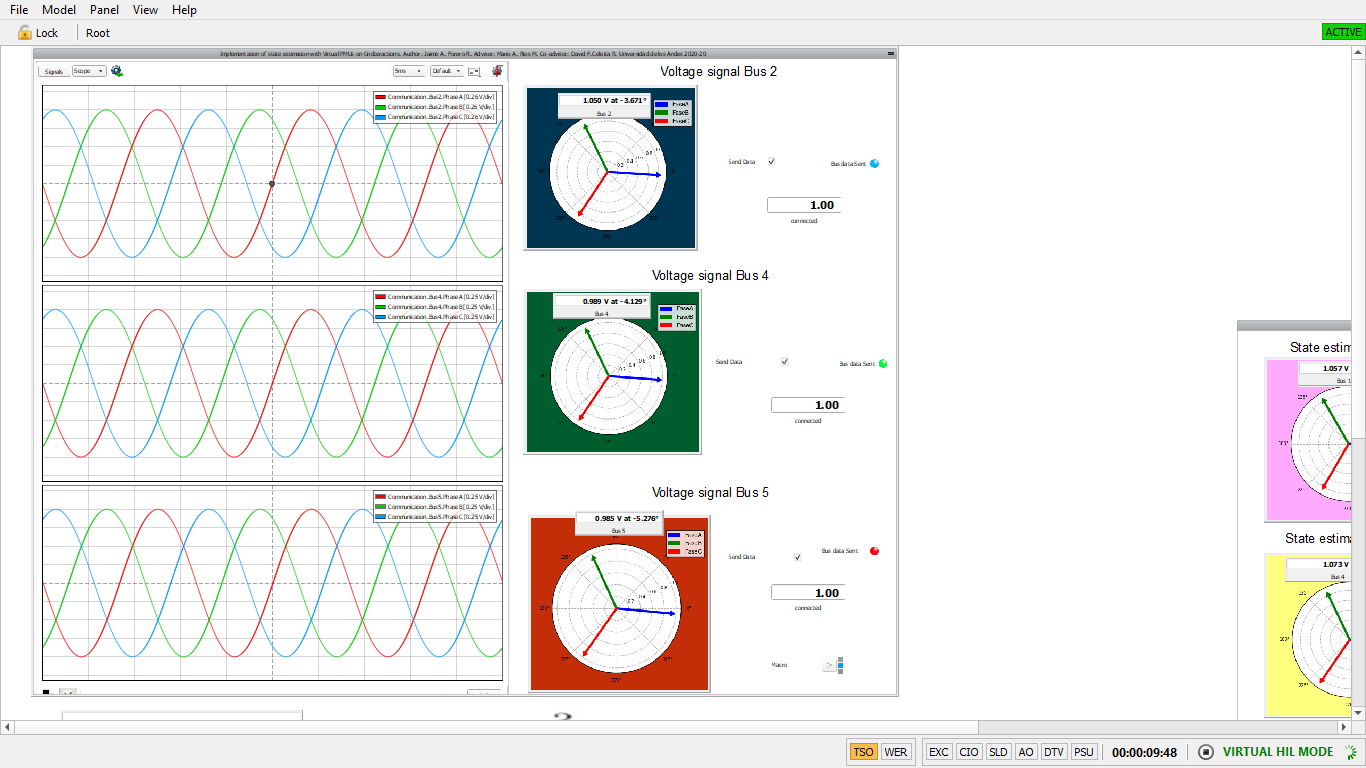


Figura 19. Sección de medidores del estimador de estado

En esta sección se presenta la forma de las señales medidas del sistema, además se selecciona la cantidad de medidores que están involucrados con la estimación de estado, para este caso el estimador de estado toma las mediciones de los nodos 2, 4 y 5, cómo se presenta en la imagen, es posible identificar que estos están seleccionados porque el led se presenta encendido y la checkbox junto a cada medidor está seleccionada.

Con la información recolectada se hace la estimación de estado y sus resultados se presentan en la sección de estados estimados del sistema, en donde se muestra el valor de la magnitud y el fasor estimados para cada uno de los nodos del sistema, incluyendo los nodos donde están ubicados los medidores. En la Figura 20, se presenta está información con el caso de estudio.

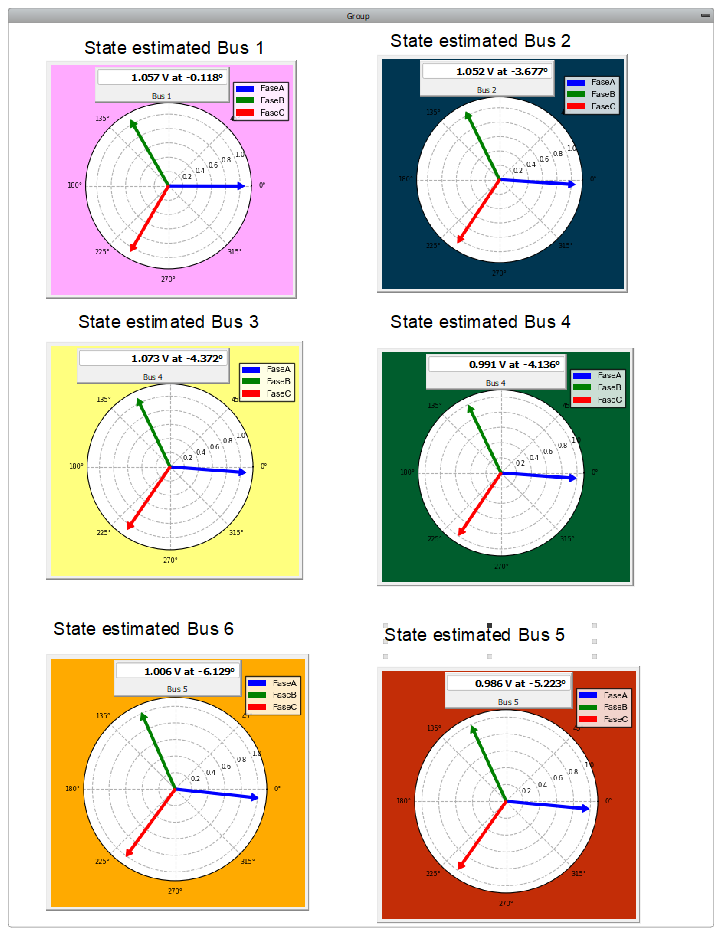


Figura 20. Sección de estados estimados

Por último, en la Figura 21 se presenta el unifilar del sistema al que le están estimando el estado en cada uno de sus nodos. En este unifilar se señalan los nodos del sistema en los que están ubicados los medidores y en cada nodo se muestra el valor del estado en que se encuentra el bus. Esto se implementó para identificar cómo está configurado el sistema y en futuros trabajos decidir cual nodo del sistema sacar de funcionamiento y bajo que tipos de condiciones el sistema presenta fallas.

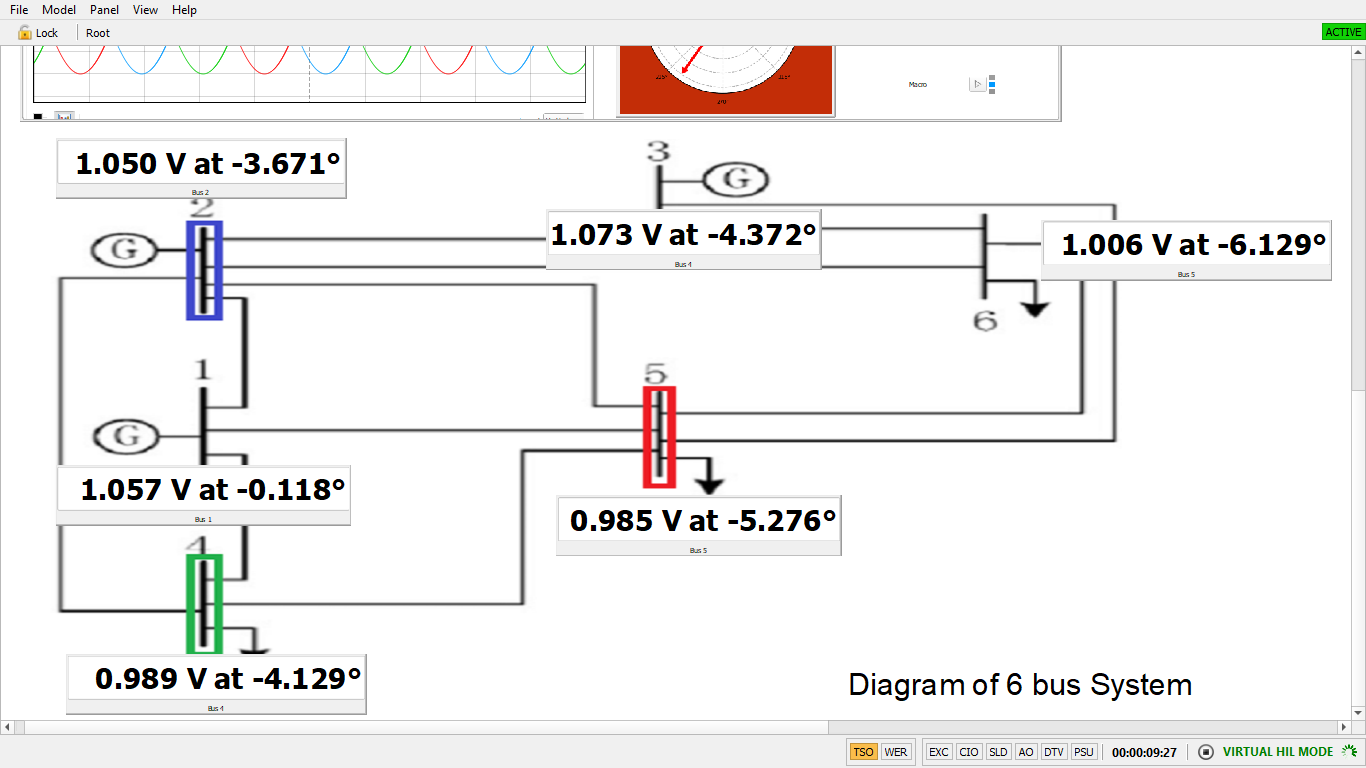


Figura 21. Unifilar del caso de estudio con los estados de cada nodo

En la Figura 22, se presenta un resumen con las actividades que se realizan para lograr estimar el estado del caso de estudio con el ambiente Virtual HIL Scada de Typhoon.

Diagrama

Descripción generada automáticamente

Figura 22. Estimación de estado con Typhoon Virtual HIL.

## **Configuración del protocolo Sampled Values**

Luego de calcular los valores de los estados en la interfaz de Typhoon HIL Scada, se pasa esta información al Schematic Editor para procesar las señales y que sea posible enviar los datos bajo el protocolo IEC 61850 Sampled Values en la salida ethernet del dispositivo Typhoon Hil 402.

Cómo se puede ver en la Figura 23, es necesario configurar tres circuitos diferentes para enviar esas señales bajo SV, para hacerlo se importan los valores de la fase y la magnitud del estado en cada uno de los nodos de la interfaz y ese valor entra por parámetro a la fuente de cada nodo y ahí se calcula el valor de cada una de las fases.

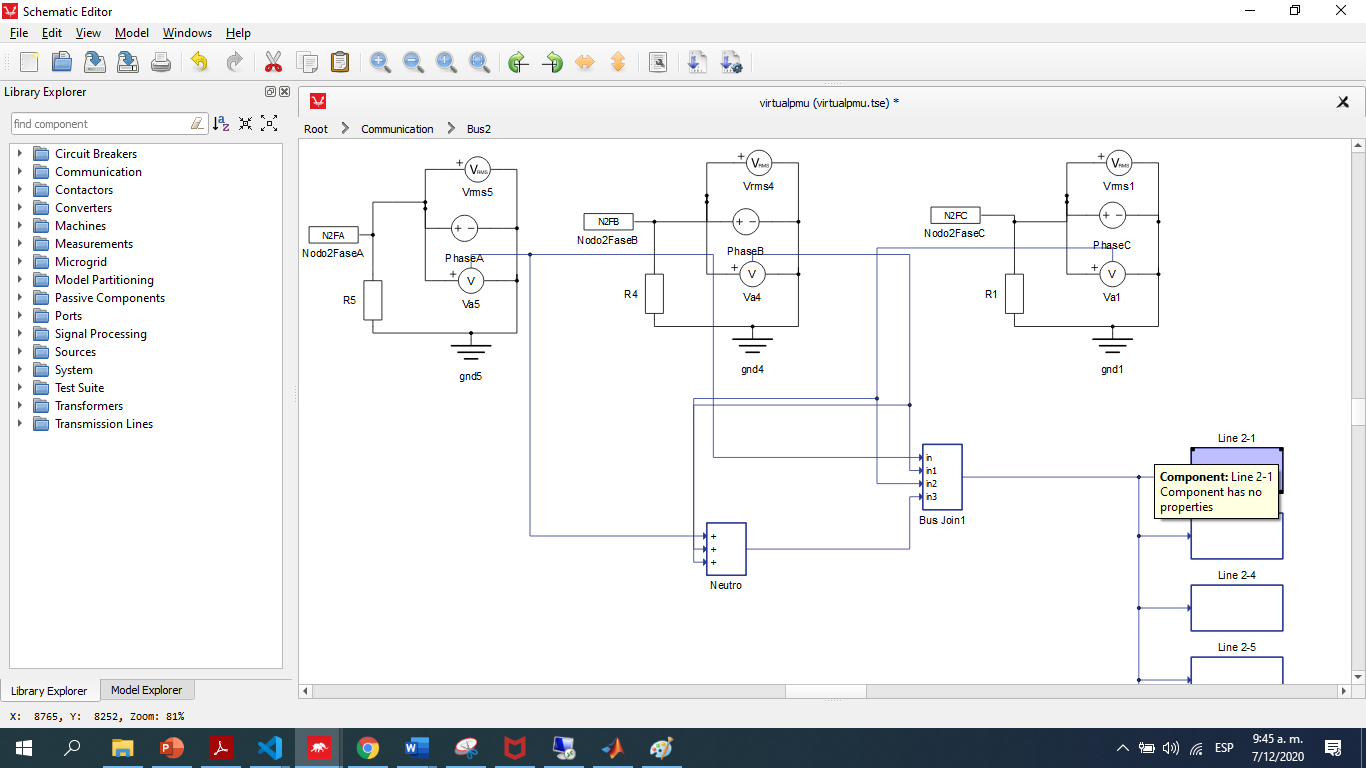


Figura 23. Configuración de Schematic editor para enviar información

Luego la señales de cada una de las fases se suman para crear el Neutro y a partir de eso juntar las cuatro señales para enviarlas al bloque que permite la comunicación, en donde se envía el valor de voltaje en el nodo y la corriente en cada una de las líneas. Para calcular la corriente se toma el valor del voltaje en la dirección destino de la línea usando una etiqueta que permite traer un valor de cualquier punto en el Schematic editor y así es posible enviar la señal completa para cada una de las líneas.

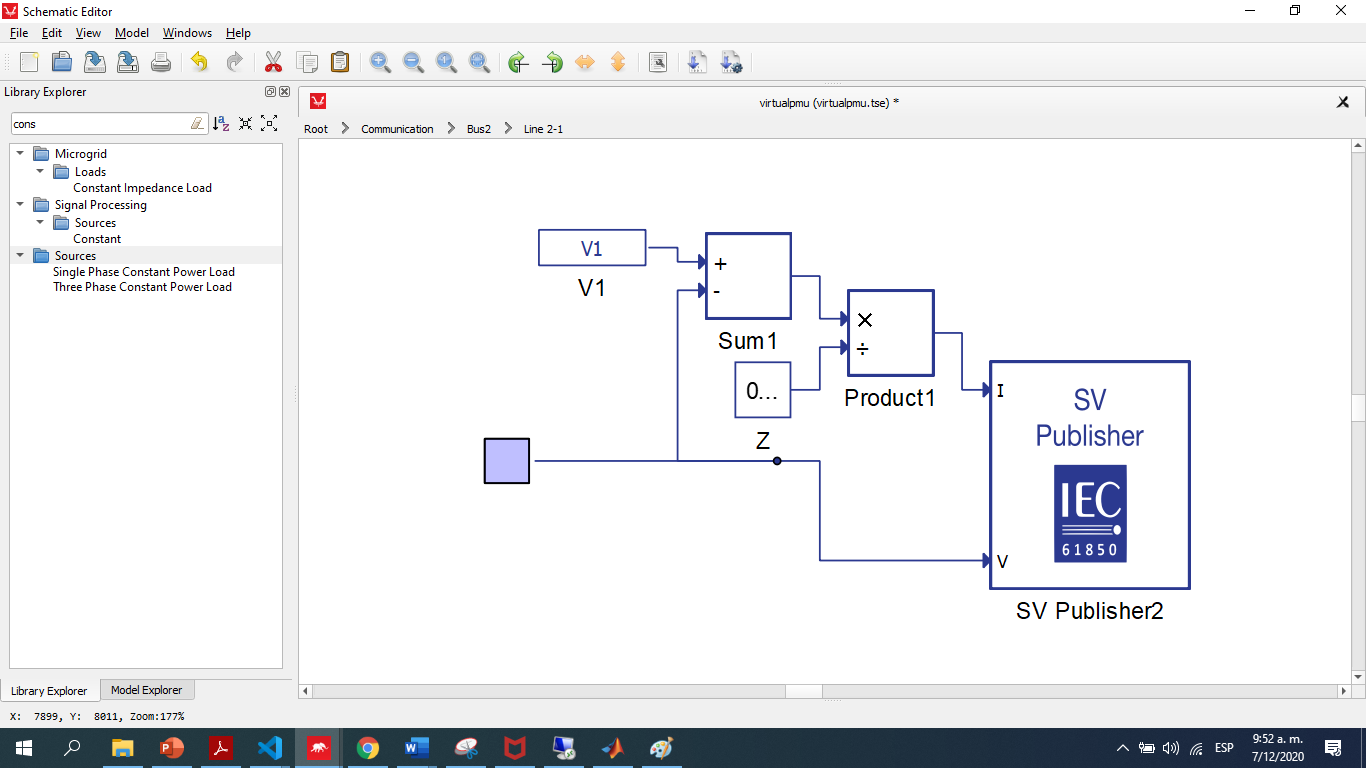


Figura 24. Schematic editor con bloques para enviar información mediante SV

# Conclusiones

En este Documento se presentaron los resultados de implementar un estimador de estado en GridTeractions, el cual es compatible con los programas Typhoon Hil Control Center, Power Factory y trabaja una buena porción de sus cálculos con el lenguaje de programación Python. Los dos principales requerimientos que este estimador de estado cubre, son la estabilidad de los sistemas que pueden integrarse y la vinculación con GridTeractions, además entre las características del estimador está que los sistemas se deben implementar en Power Factory para que sea posible estimar sus estados, debido a que su implementación se hace a través del dispositivo Typhoon Hil 402 las condiciones del sistema deben ser compatibles con las características del dispositivo [10] y al implementarse el algoritmo de estimación de estado en la interfaz del Typhoon Hil Scada, el tiempo entre paso es de 250 ms.

Se seleccionó implementar el estimador de estado en el dispositivo Typhoon Hil 402, ya que este tiene diferentes herramientas integradas que permite la versatilidad para trabajar proyectos con esta máquina. Typhoon le ofrece tres ambientes de trabajo a sus usuarios para el desarrollo de sus proyectos. El primer ambiente es el Schematic Editor en donde se pueden generar circuitos e implementar cálculos, cambios y protocolos sobre sus señales para luego asignarle una de las salidas del dispositivo, el segundo ambiente es la interfaz Virtual Hil Scada que permite la supervisión de las señales y la integración de otros programas con el dispositivo, y el tercer ambiente es el físico, el cual permite desarrollar actividades de hardware in the loop, enviar y recibir señales. La ventaja de usar el dispositivo typhoon hil se centra en la integración de sus tres ambientes de trabajo y esto permite llevar a cabo proyectos con un grado de complejidad más alto, los cuales bajo otras condiciones su desarrollo involucraría mayor cantidad de tiempo y espacio.

La norma IEC 61850 se desarrolló para mantener la integración de las subestaciones digitales con las tradicionales. Para las subestaciones digitales IEC 61850 da una serie de requerimientos que se deben cumplir a la hora de adquirir un equipo para mantener la subestación vigente a lo largo del tiempo. Entre los requerimientos que se encuentran en la norma, los protocolos de comunicación se tienen en cuenta para desarrollar este estimador de estado, ya que se está buscando emular el comportamiento de la información en estos sitios y el protocolo Sampled Values es el utilizado en muchas etapas de la comunicación en los equipos, debido a que a la hora de implementarse presenta ventajas en que es implementable con la mayoría de las máquinas y maneja la comunicación tipo Suscriber- Publisher, lo que permite llegar con un solo mensaje a muchos equipos conectado a la red.

El esquema de prueba planteado para este proyecto se desarrolló para permitir la integración de nuevos componentes y escalar los programas involucrados para su uso en proyectos con sistemas que tengan mayor cantidad de elementos eléctricos. Para añadir un nuevo caso de estudio, es necesario configurar los parámetros de la impedancia, reactancia y susceptancia en cada línea de transmisión, esto porque el estimador de estado usa estas medidas en su algoritmo para dar los valores de salida requeridos por el usuario. El código en Python se implementó para que sea posible cambiar de sistema sin tener que modificarlo.

**Referencias**

|  |  |
| --- | --- |
| [1] | A. Wood, B. Wollenberg and G. Sheble, Power generation, operation, and control, New York: john Wiley y Sons, 1996. |
| [2] | C. Zambrano, C. Trujillo, D. Celeita, M. Hernandez and G. Ramos, «GridTeractions: Simulation platform to interact with distribution systems,» de *IEEE Power and Energy Society General Meeting (PESGM)*, Boston, MA, doi: 10.1109/PESGM.2016.7741563., 2016. |
| [3] | A. DIgSILENT , «DIgSILENT releases PowerFactory 2019 - DIgSILENT,» 2020. [En línea]. Available: https://www.digsilent.de/en/newsreader/powerfactory-2019-release.html.. [Último acceso: 3 06 2020]. |
| [4] | K. D. Jones, J. S. Thorp and R. M. Gardner, «Three-phase linear state estimation using Phasor Measurements,» de *IEEE Power & Energy Society General Meeting*, Vancouver,BC, doi: 10.1109/PESMG.2013.6672516., 2013. |
| [5] | V. Özdemir and M. Göl, «A robust parameter estimation method based on LAV estimator,» de *Modern Electric Power Systems (MEPS)*, Wroclaw,doi: 10.1109/MEPS.2015.7477163, 2015. |
| [6] | M. Zhou, V. A. Centeno, J. S. Thorp and A. G. Phadke, «An Alternative for Including Phasor Measurements in State Estimators,» *IEEE Transactions on Power Systems,* vol. 21, nº 4, pp. 1930-1937, 2006, doi: 10.1109/TPWRS.2006.881112.. |
| [7] | Thorp, A.G. Phadke and J. S., Synchronized Phasor Measurements and Their Aplication, Springer, 2008. |
| [8] | Phadke, A. G., «Synchronized phasor measurements in power systems,» *IEEE Computer Applications in Power,* vol. 6, nº 2, pp. 10-15, 1993, doi: 10.1109/67.207465.. |
| [9] | Typhoon HIL, «IEC 61850 Sampled Values protocol,» 2020. [En línea]. Available: https://www.typhoon-hil.com/documentation/typhoon-hil-schematic-editor-library/References/iec\_61850\_sampled\_values\_protocol.html.. [Último acceso: 24 09 2020]. |
| [10] | Typhoon HIL, «Typhoon HIL Control Center toolchain,» 2020. [En línea]. Available: https://www.typhoon-hil.com/products/hil-software/. [Último acceso: 9 12 2020]. |
| [11] | N. Rodriguez., «A generalized n-bus case study to test the computational efficiency of a PMU-based 4-wire SDDE procedure.,» Universidad de los Andes, Bogotá, 2019. |