PROYECTO SIMULACIÓN SISTEMAS DE EV EN MATLAB

Jimmy Alcivar[1], Carlos Rojas[2]

Resumen

Este trabajo presenta el modelado y la simulación de un vehículo eléctrico (EV) mediante el uso de herramientas de MATLAB–Simulink, específicamente Virtual Vehicle Composer, Battery Builder y Motor Control Blockset. Como caso de estudio se selecciona el vehículo eléctrico BYD Seagull, a partir del cual se construye un modelo de dinámica longitudinal que integra los subsistemas principales del tren motriz eléctrico: batería, inversor, motor síncrono de imanes permanentes (PMSM) y chasis vehicular.

El Virtual Vehicle Composer se emplea para generar una arquitectura vehicular coherente y parametrizable, permitiendo la simulación del comportamiento dinámico y energético del vehículo en una ruta predefinida. Posteriormente, se desarrolla el paquete de baterías mediante Battery Builder, considerando las limitaciones propias de la herramienta y adaptando la configuración para alcanzar el voltaje nominal del sistema. Finalmente, se modela y controla el motor PMSM utilizando una estrategia de control V/f, evaluando su respuesta en términos de par, velocidad y potencia.

Los resultados obtenidos evidencian un comportamiento consistente entre las variables eléctricas, mecánicas y energéticas, validando la utilidad de estas herramientas para el análisis preliminar y el dimensionamiento de sistemas de propulsión eléctrica.

Palabras Clave: Vehículo eléctrico, MATLAB–Simulink, Virtual Vehicle Composer, Battery Builder, PMSM, control V/f.

Abstract

This paper presents the modeling and simulation of an electric vehicle (EV) using MATLAB–Simulink tools, specifically Virtual Vehicle Composer, Battery Builder, and Motor Control Blockset. The BYD Seagull electric vehicle is selected as a case study to develop a longitudinal dynamics model that integrates the main subsystems of the electric powertrain, including the battery pack, inverter, permanent magnet synchronous motor (PMSM), and vehicle chassis.

Virtual Vehicle Composer is used to generate a structured and parametrized vehicle architecture, enabling the simulation of the dynamic and energetic behavior of the EV along a predefined driving route. Subsequently, the battery pack is modeled using Battery Builder, taking into account the software limitations and adapting the cell arrangement to achieve the nominal battery voltage. Additionally, the PMSM is modeled and controlled using a V/f control strategy, allowing the evaluation of motor speed, torque, and power response.

The simulation results show coherent interactions among electrical, mechanical, and energy variables, demonstrating the effectiveness of MATLAB–Simulink as a reliable platform for preliminary analysis and early-stage validation of electric vehicle propulsion systems.

Keywords: Electric vehicle, MATLAB–Simulink, Virtual Vehicle Composer, Battery Builder, PMSM, V/f control.

Vehículos Eléctricos e Híbridos I, Universidad Politécnica Salesiana – Ecuador.

1. Introducción

El uso de entornos de simulación como MATLAB–Simulink se ha consolidado como una solución ampliamente empleada en la industria y la academia, debido a su capacidad para integrar modelos físicos, eléctricos y de control dentro de una misma plataforma. Herramientas especializadas como Virtual Vehicle Composer facilitan la creación rápida de arquitecturas vehiculares, permitiendo la parametrización de subsistemas clave y la evaluación de su interacción bajo distintos escenarios de conducción, sin la necesidad de desarrollar modelos desde cero.

Asimismo, el modelado de sistemas de almacenamiento de energía representa un aspecto crítico en los vehículos eléctricos, ya que la batería influye directamente en la autonomía, el desempeño y la seguridad del vehículo. Battery Builder permite diseñar paquetes de baterías a partir de celdas individuales, considerando configuraciones en serie y paralelo, así como variables fundamentales como voltaje, corriente, capacidad y estado de carga. No obstante, las limitaciones inherentes a las herramientas de simulación requieren realizar adaptaciones que permitan aproximarse a las condiciones reales del sistema.

Por otro lado, el motor síncrono de imanes permanentes (PMSM) es ampliamente utilizado en aplicaciones vehiculares debido a su alta densidad de potencia, elevada eficiencia y excelente respuesta dinámica. El uso de estrategias de control como V/f permite un arranque suave y un control estable en lazo abierto, siendo adecuado para estudios preliminares del comportamiento del motor dentro del tren motriz eléctrico.
En este trabajo se presenta el modelado y la simulación de un vehículo eléctrico basado en el BYD Seagull, integrando Virtual Vehicle Composer, Battery Builder y Motor Control Blockset, con el objetivo de analizar de forma conjunta el desempeño dinámico, energético y eléctrico del sistema, demostrando la utilidad de estas herramientas para el análisis y validación inicial de vehículos eléctricos.

2. Marco teórico

- **Virtual Vehicle Composer**

Es una herramienta orientada a acelerar la creación de modelos de vehículo en Simulink mediante la selección de una arquitectura vehicular y la parametrización de sus subsistemas principales. Su propósito es reducir el tiempo de configuración inicial, estandarizar la estructura del modelo y permitir comparaciones rápidas entre configuraciones.

En términos funcionales, el Composer actúa como un generador de modelo: a partir de entradas de usuario (tipo de tren motriz, motor, transmisión, batería, inversor, ciclo de conducción, etc.), crea un diagrama con bloques preconfigurados, conexiones correctas entre subsistemas y, en muchos casos, parámetros iniciales coherentes para simulación.

Arquitecturas vehiculares y alcance: Un vehículo puede representarse como la interacción de varios dominios físicos y de control. De manera general, el Composer organiza el sistema en:

a) Fuente de energía: tanque/combustible (vehículos convencionales) o batería/almacenamiento eléctrico (EV/HEV).
b) Conversión de energía: motor térmico y/o máquina eléctrica (PMSM, IM, etc.), con su electrónica de potencia (inversor).
c) Transmisión: reductor, caja, diferencial y ejes.
d) Chasis longitudinal: modelo de masa equivalente, resistencia a la rodadura, resistencia aerodinámica y pendiente.
e) Ruedas y neumáticos: relación fuerza–tracción, radio efectivo, pérdidas y adherencia (según el nivel de detalle).

f) Control: lazos para velocidad/par, estrategia de reparto de potencia (en híbridos), control del inversor y gestión de energía.

El alcance del Composer típicamente se centra en dinámica longitudinal (aceleración, velocidad, fuerzas resistivas y demanda de par/potencia), siendo común en fases de dimensionamiento y validación temprana del tren motriz.

- **Battery Builder**

Sistemas de Gestión de Baterías (BMS) y Modelado Térmico. El desarrollo de sistemas de movilidad eléctrica, como las bicicletas o autos eléctricos, requiere un conocimiento profundo sobre la configuración de celdas, la gestión de la carga y el comportamiento térmico de los paquetes de baterías.

Arquitectura del Paquete de Baterías

Un paquete de baterías no es una unidad sólida, sino un conjunto organizado de celdas individuales. Su capacidad y voltaje total dependen de su configuración:

Conexión en Paralelo: Se utiliza para aumentar la capacidad total (Ah). Al conectar celdas en paralelo, el voltaje se mantiene igual al de una celda, pero la corriente disponible se suma.

Conexión en Serie: Se utiliza para aumentar el voltaje total (V). La capacidad se mantiene igual a la de una unidad, pero el potencial eléctrico se suma.

Notación de Configuración: Se utiliza la nomenclatura XpYs (por ejemplo, 3p4s), donde X representa el número de celdas en paralelo y Y el número de grupos en serie.

Estado de Carga (SOC) y Gestión de Carga

State of Charge (SOC): Es una medida porcentual que indica el nivel de energía restante en la batería. Para mantener la integridad de las celdas, un BMS debe implementar algoritmos de carga específicos:

Carga CC-CV (Constant Current - Constant Voltage): Es el método estándar donde primero se aplica una corriente constante hasta alcanzar un umbral de voltaje, seguido de un voltaje constante mientras la corriente disminuye gradualmente.

Límites Operativos: El sistema debe asegurar que el SOC permanezca dentro de rangos seguros para evitar la degradación prematura o fallos catastróficos.

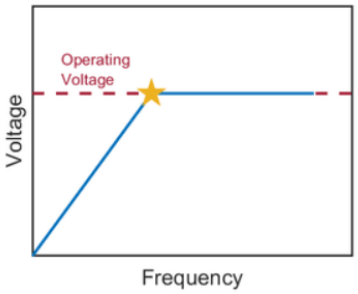
- **Introducing to Motor Control**

Para modelar el motor síncrono de imanes permanentes (PMSM) para un vehículo eléctrico (EV), se necesita conocer parámetros clave del motor que describen las propiedades físicas del motor y ayudan a los bloques integrados a crear modelos precisos y diseñar controladores eficaces. Los fabricantes proporcionan una hoja de datos con información sobre las propiedades y el rendimiento del motor. Estos datos pueden servir como referencia. Sin embargo, es posible que falte algún parámetro necesario o que se requiera un mayor nivel de precisión. Motor Control Blockset ofrece herramientas para estimar los parámetros del motor mediante la identificación del sistema. Estas herramientas generan una estructura de datos que contiene los parámetros estimados, que luego se pueden cargar en un espacio de trabajo de MATLAB.

Control de tensión por frecuencia: Al arrancar un motor parado, aplicar repentinamente la tensión máxima puede provocar comportamientos negativos. En el caso de los motores eléctricos, el control de tensión por frecuencia, o V/F, es una estrategia de control de bucle abierto para aumentar de forma constante la tensión aplicada y la frecuencia de la señal a un motor de CA.

En el control V/F, el voltaje de fase y la frecuencia de la señal enviada al motor se incrementan

proporcionalmente. A medida que aumenta la frecuencia, también lo hace el voltaje, hasta alcanzar el voltaje deseado. A continuación, el voltaje se mantiene constante mientras la frecuencia sigue aumentando, lo que incrementa la velocidad del motor. En el caso de un motor parado, se puede aumentar gradualmente el voltaje y la frecuencia para garantizar un arranque suave del motor.



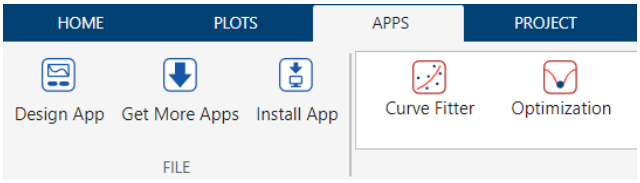
3. Desarrollo

Vehículo EV seleccionado: BYD SEAGULL

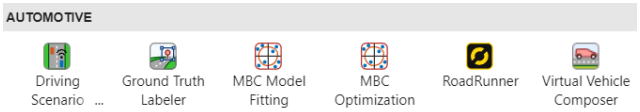
3.1. Virtual Vehicle Composer

- **Creación del modelo**

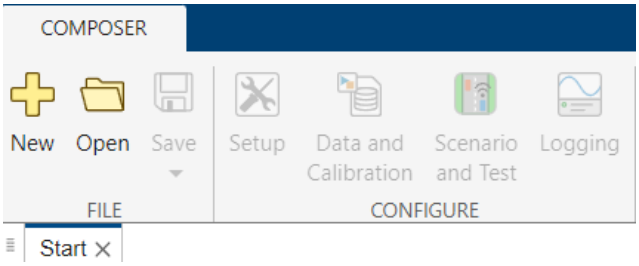
a) En el workspace de Matlab, se selecciona la opción de APPS.



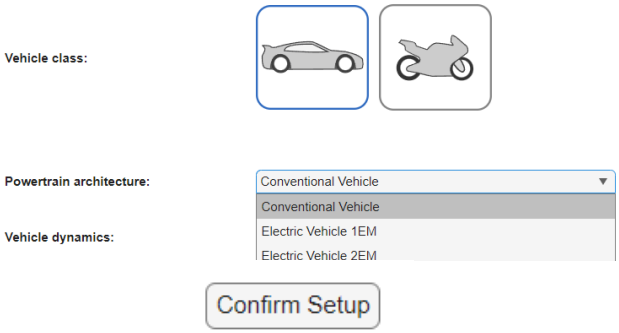
b) En el apartado de automotive, se selecciona la opción Virtual Vehicle Composer.



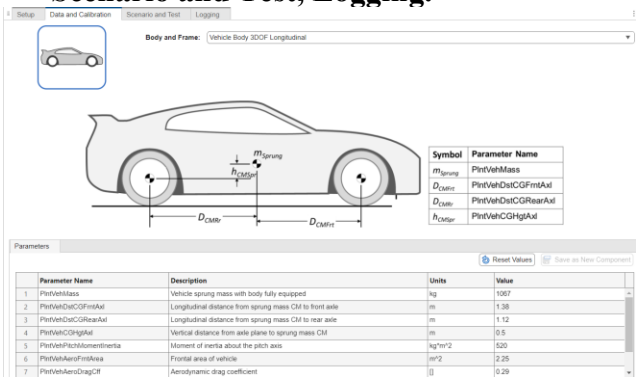
c) En la nueva pestaña se selecciona New.



d) En el apartado **clase de vehículo** se selecciona **vehículo de pasajero,** y en el **tren motriz** se selecciona **vehículo eléctrico 1EM.** Se confirma la selección del vehículo en la misma pestaña en **Confirm Setup.**



e) Se abre una nueva pestaña con las opciones por ingresar para la simulación del vehículo, que son **Setup, Data and Calibration, Scenario and Test, Logging.**



f) En **Data** and **Calibration**, se ingresan los datos referentes al chasis del vehículo que se desea simular, en el caso del BYD SEAGULL los valores a ingresar son los siguientes:

Parámetro	**Valor**
PlntVehMass	**1067 kg**
PlntVehDstCGFrntAxl	**1.38 m**
PlntVehDstCGRearAxl	**1.12 m**
PlntVehCGHgtAxl	**0.50 m**

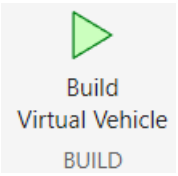
PlntVehPitchMomentInertia	**520 kg·m²**
PlntVehAeroFrntArea	**2.25 m²**
PlntVehAeroDragCff	**0.29**
PlntVehAeroLiftCff	**0.10**
PlntVehAeroPitchCff	**−0.08**

g) En el apartado de **Logging** se ingresan los valores que se desea obtener al momento de realizar la simulación de ruta por parte del vehículo. Seleccionando en la lista de señales la opción de **EM** y **Battery**.

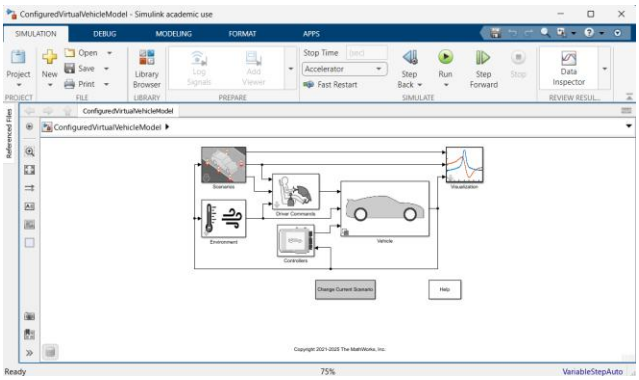
h) Los datos que se desean obtener son los siguientes:

Variable	Dato
Battery.BMSSensors.CellTemperatures	Temperaturas medidas por el BMS en cada celda
Battery.BMSSensors.CellVoltages	Voltajes medidos por el BMS en cada celda
Battery.BMSSensors.PackCurrent	Corriente total medida del pack
Battery.BMSSensors.PackVoltage	Voltaje total medido del pack
Battery.BMSSensors.VoutInvtr	Voltaje de salida hacia el inversor
Battery.BattInfo.BattAmpHr	Capacidad disponible de la batería
Battery.BattInfo.BattCurr	Corriente instantánea de la batería
Battery.BattInfo.BattPwr	Potencia eléctrica de la batería
Battery.BattInfo.BattSoc	Estado de carga de la batería
Battery.BattInfo.BattTemp	Temperatura promedio de la batería
Battery.BattInfo.BattVolt	Voltaje total de la batería
EM.EMMechPwr	Potencia mecánica del motor
EM.EMPwrElec	Potencia eléctrica del motor
EM.EMSpd	Velocidad del motor eléctrico
EM.EMTemp	Temperatura del motor eléctrico
EM.EMTrq	Par del motor eléctrico

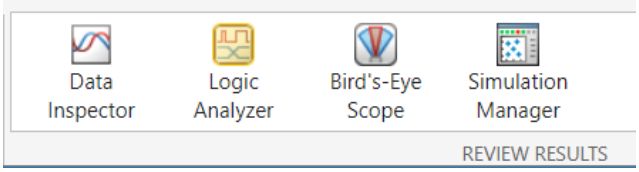
i) Se selecciona la opción de **Buil Virtual Vehicle**, para realizar la simulación del vehículo con la ruta establecida por defecto en el Matlab.



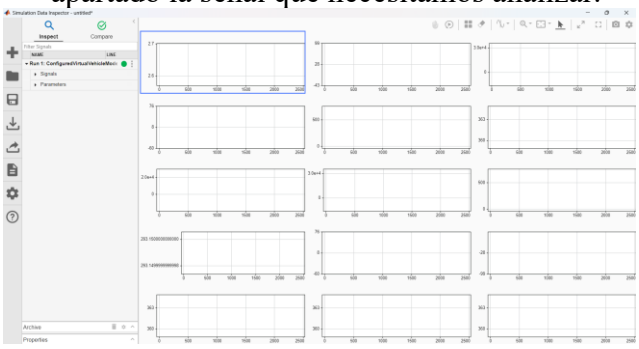
j) En la nueva pestaña, se selecciona **Run** para correr el sistema.



k) En el apartado de **View Results**, se da clic en **Data Inspector.**

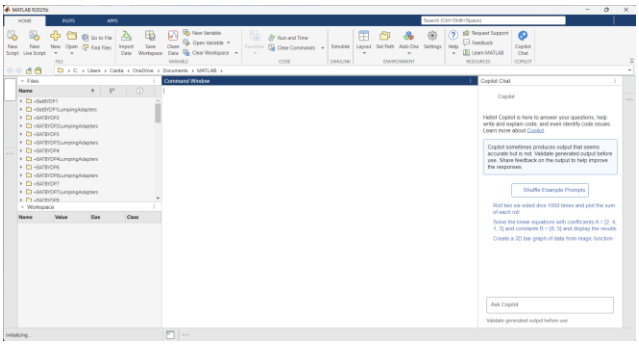


l) En la nueva pestaña, en la opción de **Signals,** se escoge para cada grafica que permite el apartado la señal que necesitamos analizar.

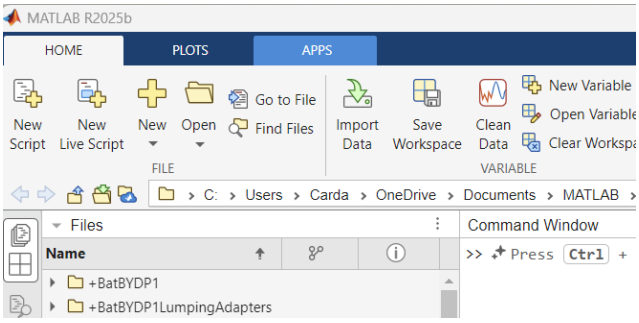


3.2. Battery builder

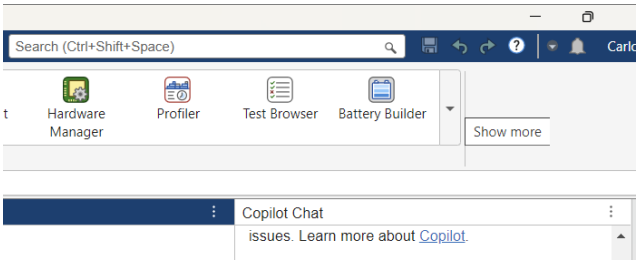
a) Se abre la aplicación de Matlab.



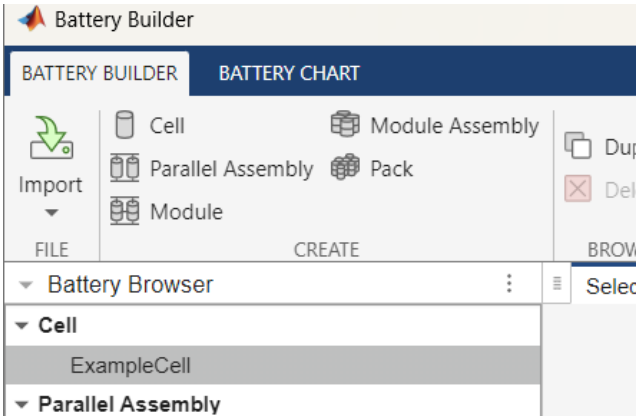
b) En el menú nos dirigimos a la pestaña de **APPS.**



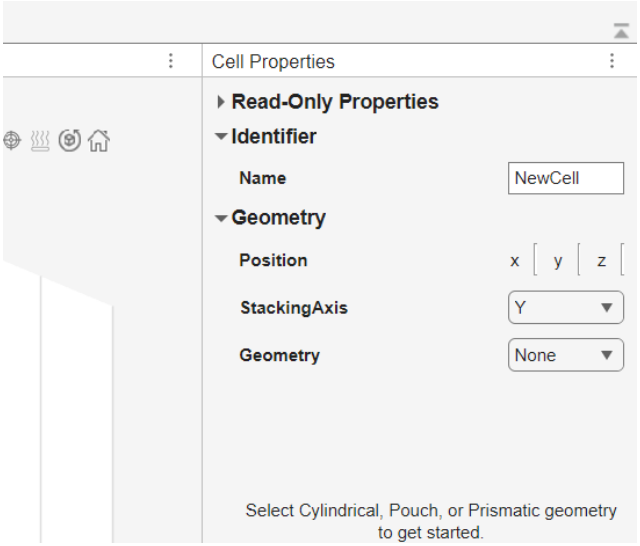
c) En la barra de herramientas nos dirigimos a la aplicación llamada Battery builder y le damos clic.



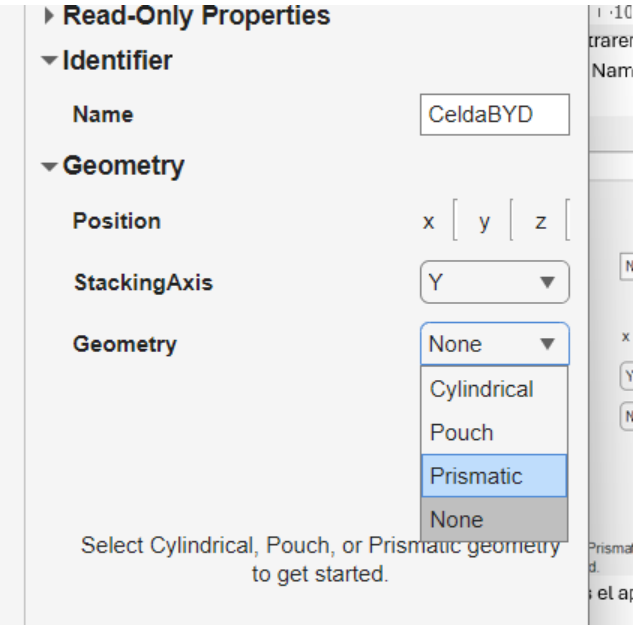
d) En la nueva ventana que se abre nos dirigimos a **Cell** y le damos clic para generar una nueva celda.



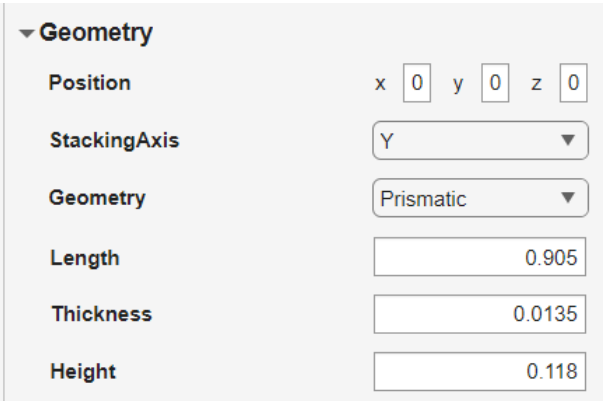
e) En el lado derecho en **Cell Properties** encontraremos el apartado **Identifier**, y dentro de este encontraremos el apartado **Name**, aquí cambiaremos el nombre de la Celda.



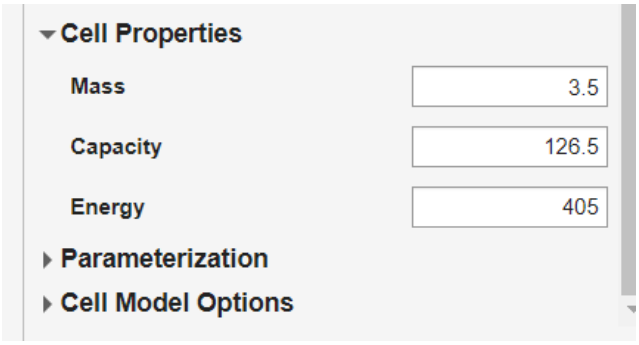
f) En el apartado de **Geometry** encontraremos el apartado **Geometry**, en la lista desplegable de este seleccionaremos **Prismatic**.



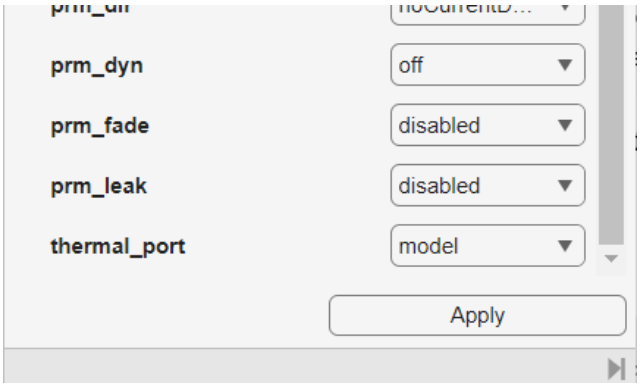
g) Tras seleccionar la geometría más abajo se desplegarán tres apartados **Length**, **Thickness** y **Height** aquí introduciremos los valores de largo, espesor y alto de nuestra celda en metros.



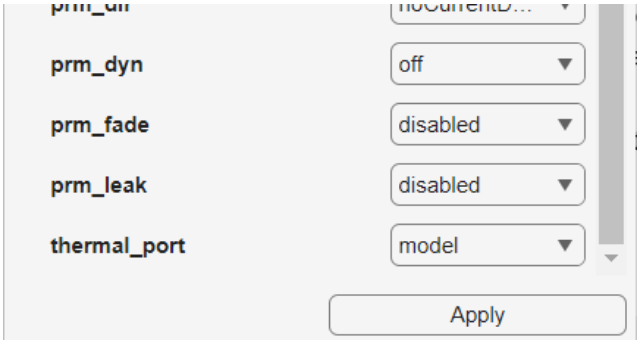
h) Después más abajo en el apartado de **Cell properties** daremos clic y se desplegaran tres apartados más **Mass**, **Capacity** y **Energy** aquí introduciremos los valores de masa en kg, capacidad en A*hr y energía en W*hr respectivamente.



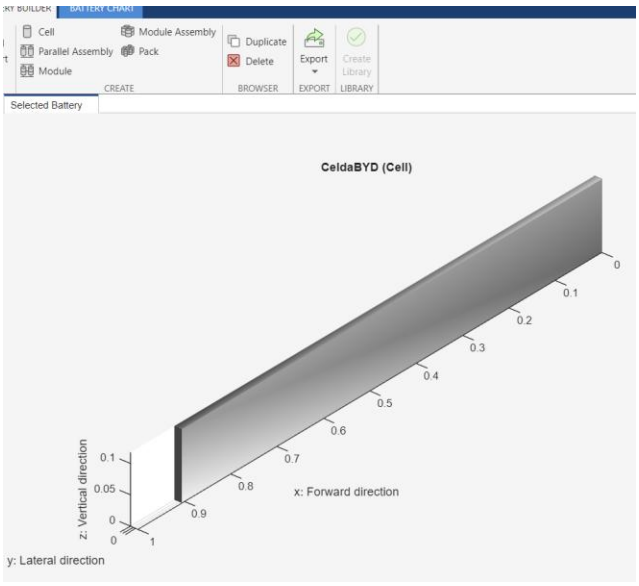
i) Más abajo en el apartado de **Cell model options** daremos clic y se desplegaran varios apartados, en el apartado **SOC_port** en la lista desplegable seleccionamos **Yes**.



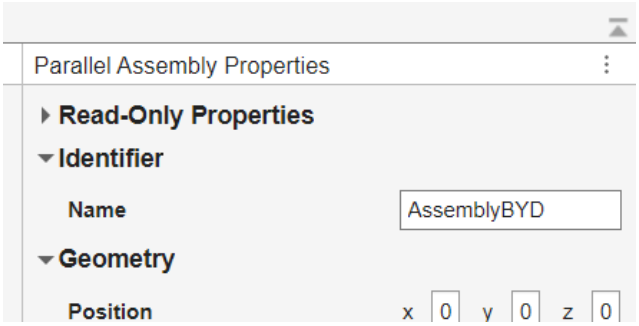
j) Más abajo damos clic en **Apply** y Matlab generara automáticamente la celda.



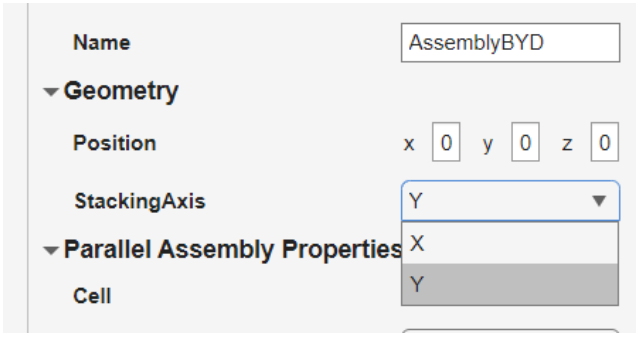
k) En la barra de herramientas damos clic en **Parallel Assembly** para generar un nuevo ensamble de celdas.



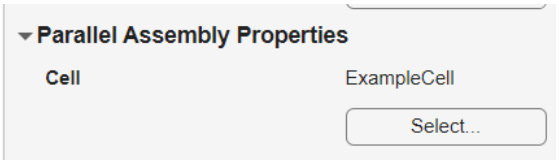
l) En el lado derecho en **Parallel Assembly Properties** encontraremos el apartado **Identifier** y dentro de este encontraremos el apartado **Name**, aquí cambiaremos el nombre del ensamble de celdas.

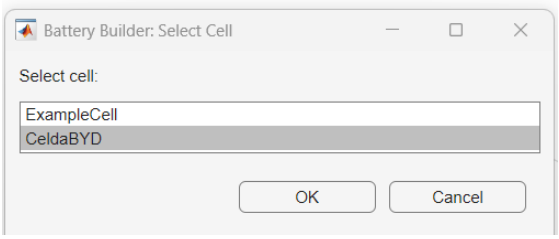


m) Más abajo en el Apartado **Geometry** encontraremos el apartado **StackingAxis** en la lista desplegable de este seleccionamos **Y**.

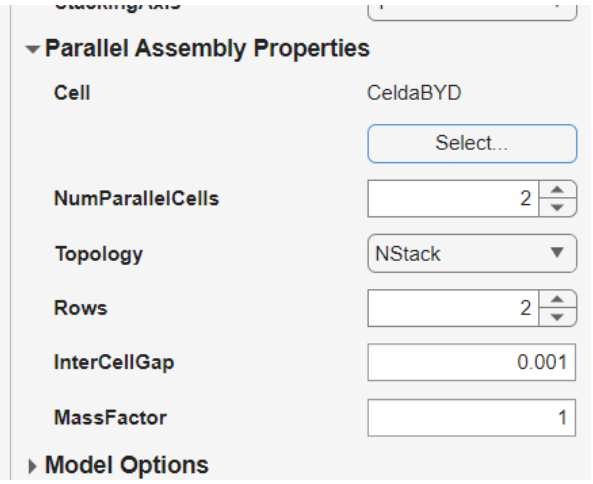


n) Más abajo en el apartado **Parallel Assembly Properties** encontraremos el apartado **Cell** aquí daremos clic en **Select** y en la ventana que se abre seleccionamos la celda que creamos anteriormente y damos clic en ok.

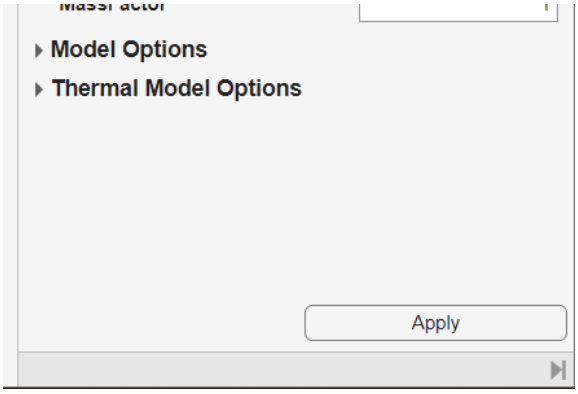




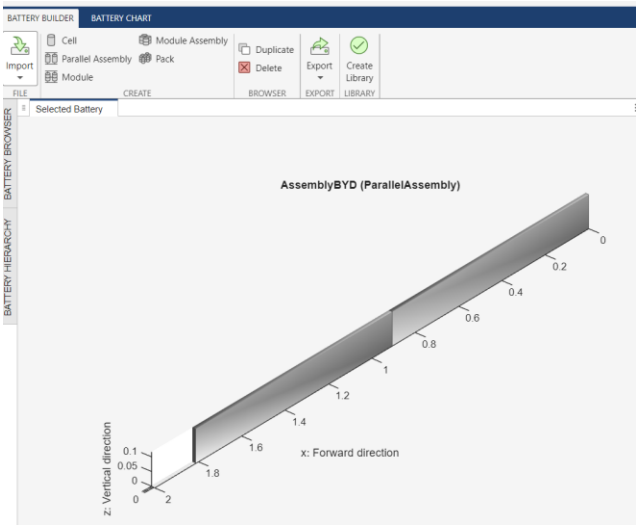
o) Dentro del mismo apartado más abajo en **NumParallelCells** cambiamos el valor a 2, en **Topology** en la lista desplegable seleccionaremos **Nstack**, en **Rows** cambiaremos el valor a 2 y en **InterCellGap** introduciremos el valor de espacio entre celdas en metros.



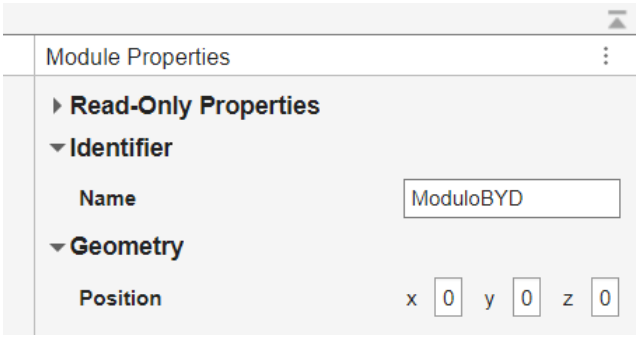
p) Más abajo daremos clic en **apply** para guardar esta configuración y Matlab automáticamente generará el ensamble de celdas.



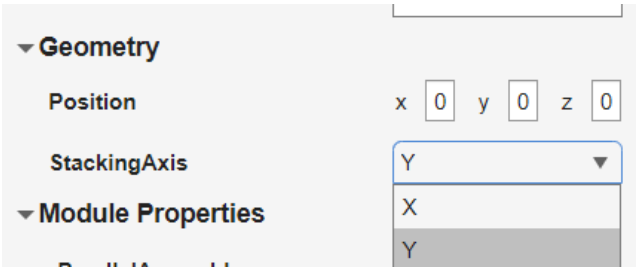
q) En la barra de herramientas daremos clic en Module para generar un nuevo módulo.



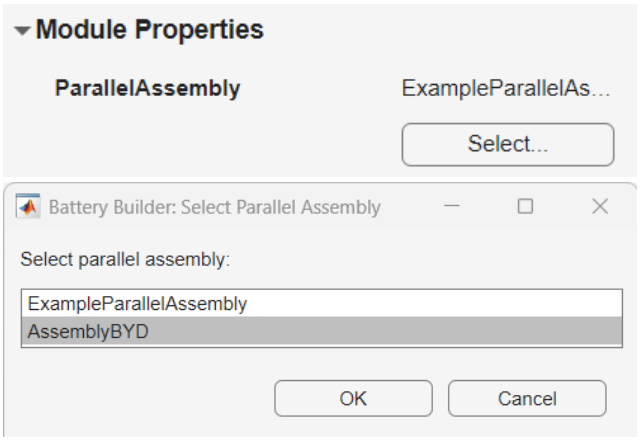
r) En el lado derecho en **Module Properties** encontraremos el apartado **Identifier** y dentro de este encontraremos el apartado **Name**, aquí cambiaremos el nombre del módulo.



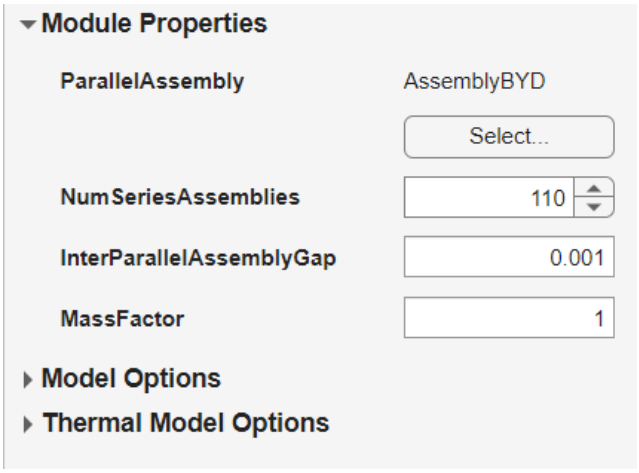
s) Más abajo en **Geometry** encontraremos el apartado **StackingAxis** en la lista desplegable seleccionaremos **Y**.



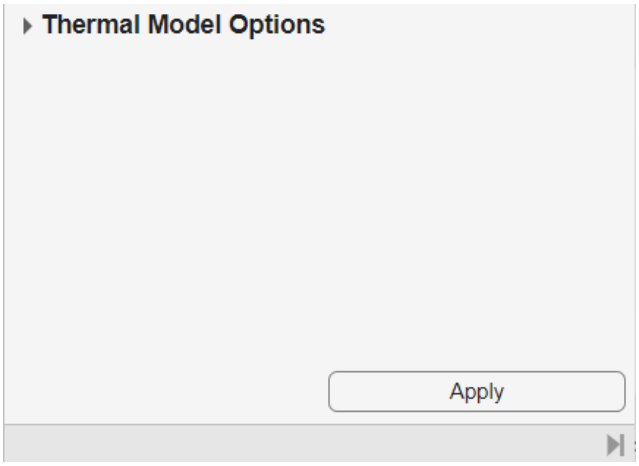
t) Más abajo en el apartado **Module Properties** encontraremos el apartado **Parallel Assembly** aquí le daremos clic en **select** y en la nueva ventana que se abre seleccionamos el ensamble de celdas que creamos y damos clic en ok.



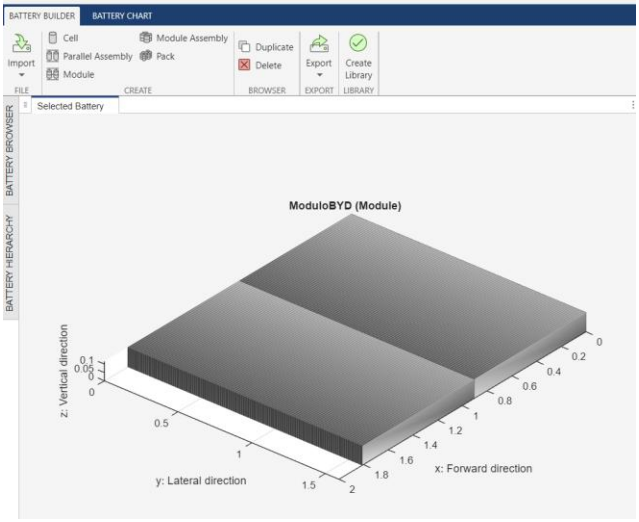
u) Más abajo en el apartado **NumSeriesAssemblies** cambiaremos el valor a 110.



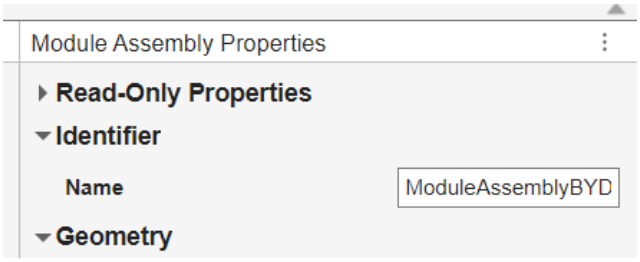
v) Más abajo daremos clic en **Apply** para guardar esta configuración y Matlab automáticamente generara el módulo.



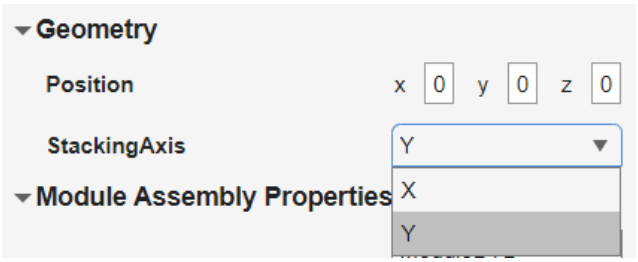
w) En la barra de tareas daremos clic en **Module Assembly** para generar un nuevo ensamble de módulos.



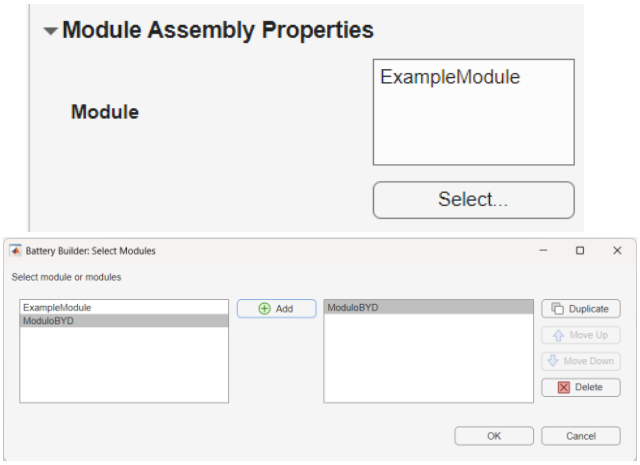
x) En el lado derecho en **Module Assembly Properties** encontraremos el apartado **Identifier** y dentro de este encontraremos el apartado **Name**, aquí cambiaremos el nombre del ensamble de módulos.



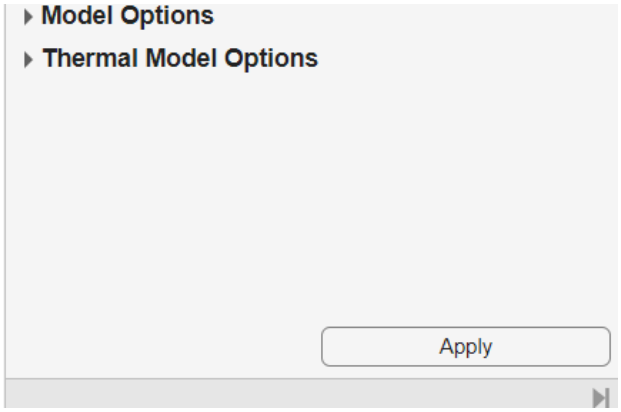
y) Más abajo en el apartado **Geometry** encontraremos el apartado **StackingAxis** en la lista desplegable seleccionaremos **Y**.



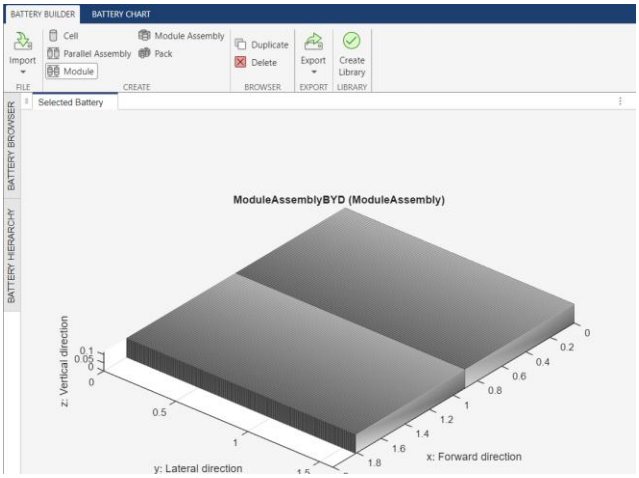
z) Más abajo en el apartado **Module Assembly Properties** encontraremos el apartado **Module**, aquí le daremos clic en **Select** y en la nueva ventana que se abre escogeremos el módulo que creamos anteriormente y le damos clic en **Add** y luego en **ok**.



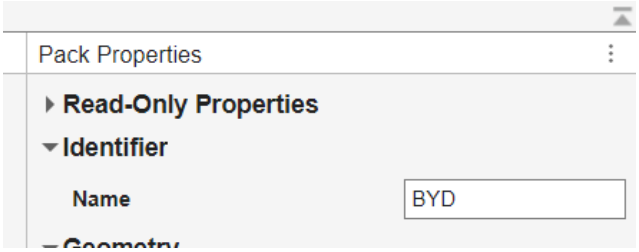
aa) Más abajo le damos clic en **Apply** para guardar esta configuración y Matlab automáticamente generara el ensamble de módulos.



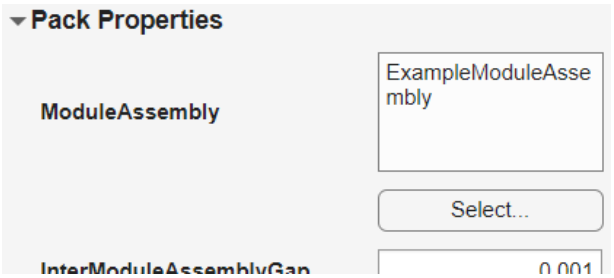
bb) En la barra de tareas daremos clic en **Pack** para generar un nuevo paquete de baterías.

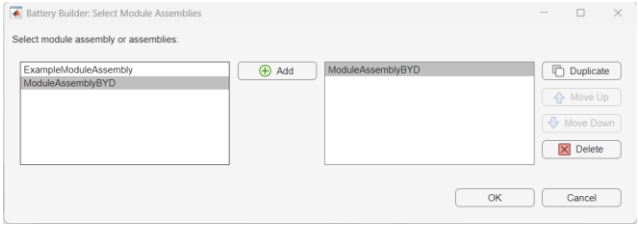


cc) En el lado derecho en **Pack Properties** encontraremos el apartado **Identifier** y dentro de este encontraremos el apartado **Name**, aquí cambiaremos el nombre del paquete de baterías.

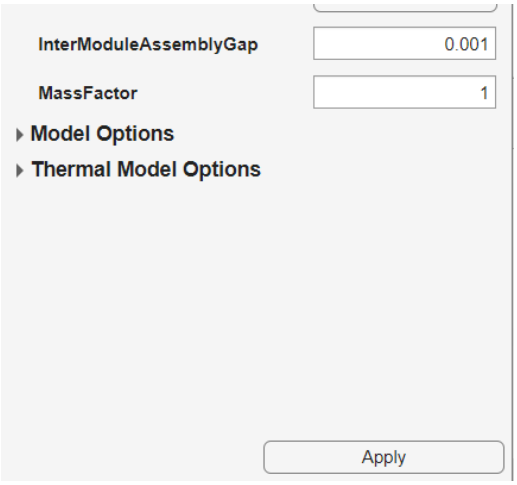


dd) Más abajo en el apartado **Pack Properties** encontraremos el apartado **ModuleAssembly**, aquí le daremos clic en **Select** y en la nueva ventana que se abre escogeremos el ensamble de módulos que creamos anteriormente le daremos clic en **Add** y luego en **Ok**.

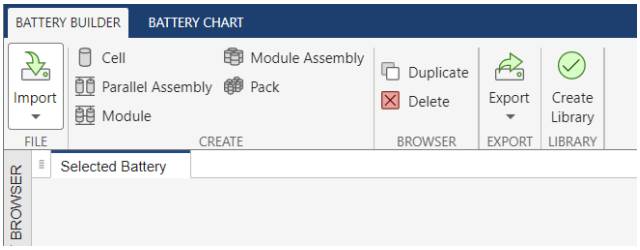




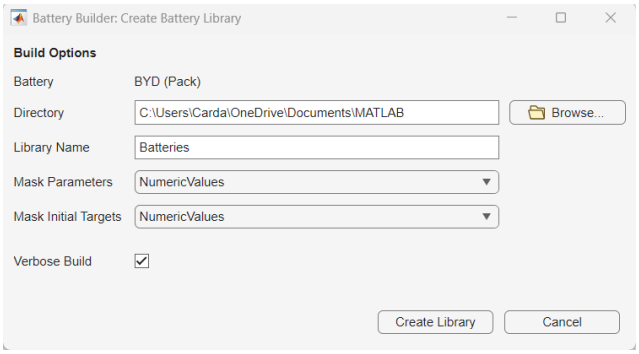
ee) Más abajo daremos clic en Apply para guardar esta configuración y Matlab automáticamente generará el paquete de baterías.



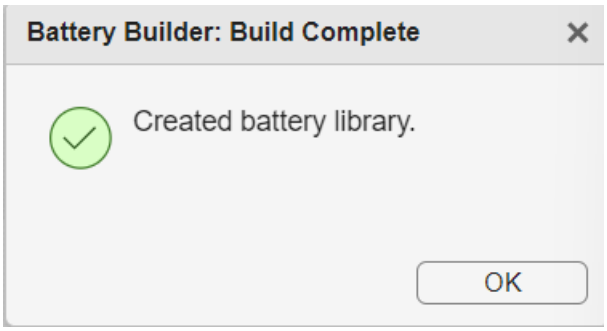
ff) Para finalizar en la barra de tareas le daremos clic en **Create Library** para crear una librería desde la cual poder llamar a la batería con facilidad al entorno de trabajo de Simulink.



gg) En la nueva ventana que se abre en **Directory** seleccionaremos la carpeta en la que deseamos guardar la librería y en **Library Name** cambiamos el nombre de la librería, luego de esto le damos clic en **Create Library**.



hh) Tras unos instantes la librería se guardará y abrirá una nueva ventana para indicar que la librería fue creada con éxito, le damos clic en **ok**.

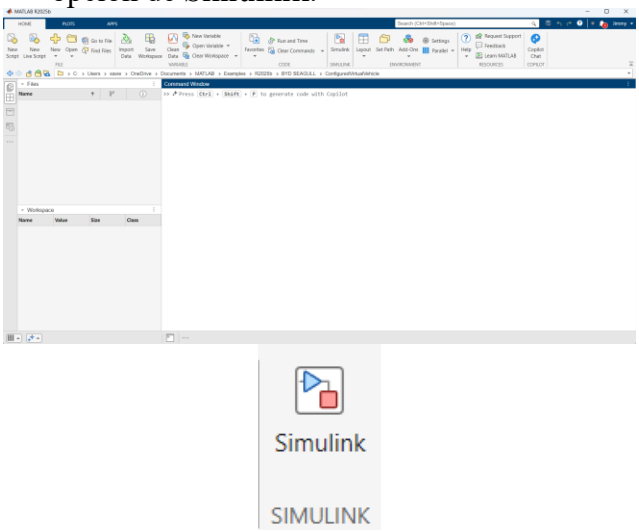


3.3. Introducing to motor control

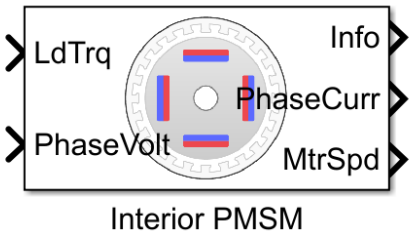
a) El motor por simularse es un TZ180XSH perteneciente a un BYD SEAGULL; el cuál posee los siguientes valores de fabricación, utilizados para esta simulación.

Descripción	**Valores**
Number of Pole Pairs	8
Rotational Inertia	**0.09 kg·m²**
Stator Resistance	**0.013 Ω**
D-Axis Inductance	**150 µH**
Q-Axis Inductance	**250 µH**
Permanent Magnet Flux	**0.175 Wb**
Rated Voltage	**400 V**
Rated RPM	**9600 RPM**
LdrTqr	**135 Nm**

b) En el espacio de trabajo de Matlab, escoger la opción de **Simulink**.



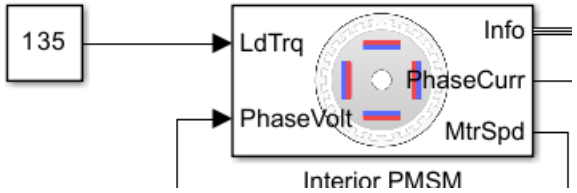
c) En la ventana de Simulink, en Library Browser ir a: **Motor Control Blockset > Electrical Systems-Motors y** arrastrar el bloque Interior PMSM al modelo.



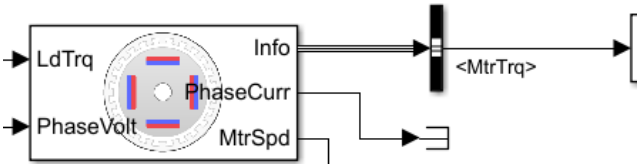
d) Abrir el **Block Parameters** y configurar los datos según la tabla mencionada en el literal a.

e) Desde **Simulink > Sources**, agregar un bloque **Constant**. Conectar al bloque **PMSM**

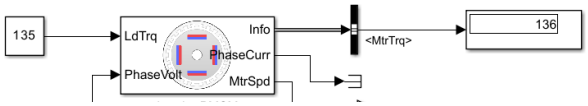
a la entrada **LdrTqr** son los valores del literal a.



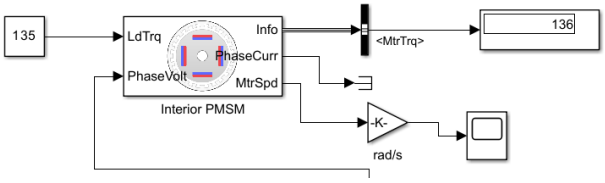
f) Desde **Simulink > Signal Routing**, agregar un **Bus Selector**. Conectar su entrada al puerto **Info** del PMSM. Abrir el **Bus Selector** y seleccionar **MtrTrq** posterior presionar **Add to output** Eliminar las señales sobrantes signal1 y signal2.



g) Desde **Simulink > Sinks**, agregar un bloque **Display**. Conectar la salida del Bus Selector al **Display**.

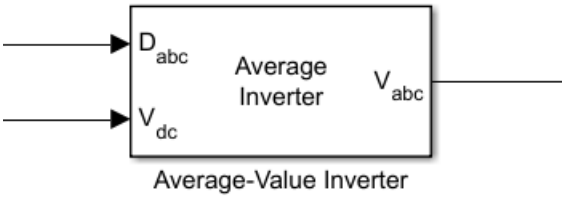


h) Conectar la salida **PhaseCurr** a un bloque **Terminator**. Insertar un bloque **Gain**. Configurar el valor del Gain para convertir de rad/s a RPM. Conectar **MtrSpd-Gain**.

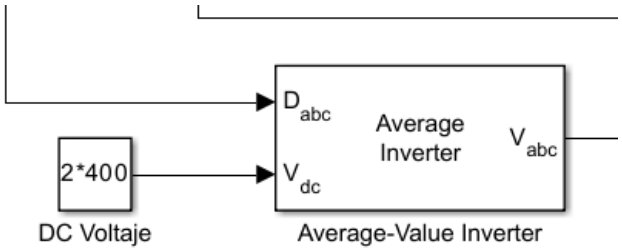


i) Desde **Simulink > Sinks**, agregar un bloque **Scope** y conectar la salida del Gain al mismo.

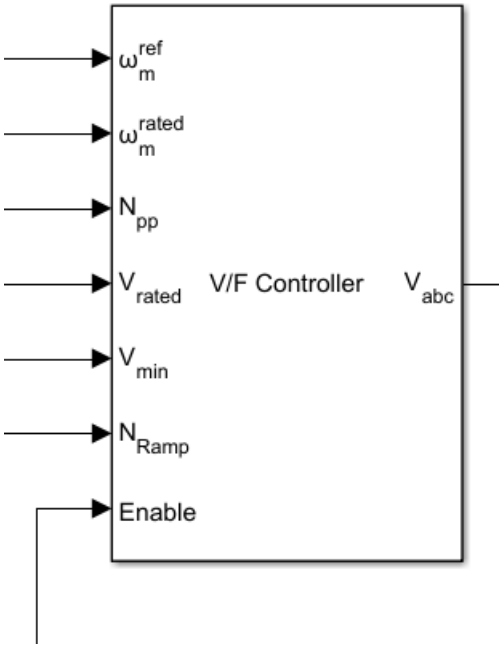
j) Ir a **Motor Control Blockset > Electrical Systems > Inverters** y agregar el bloque **Average-Value Inverter** al modelo.



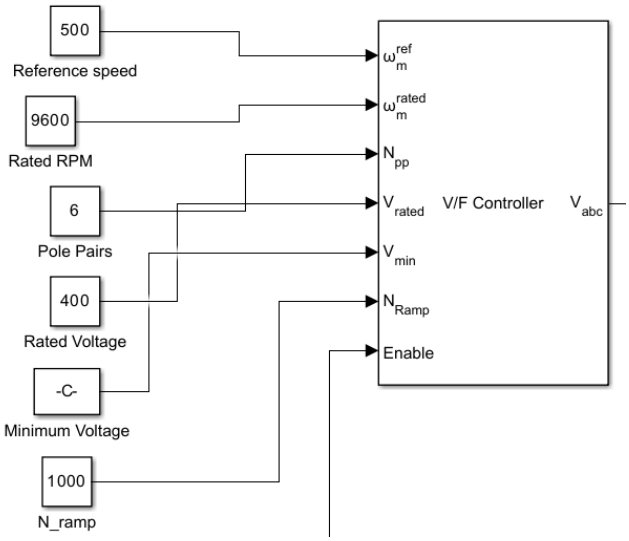
k) Agregar un bloque **Constant** y conectar al puerto **V_dc** del **Average-Value Inverter**, con el valor de **Rated Voltage*2** entregado en el literal a.



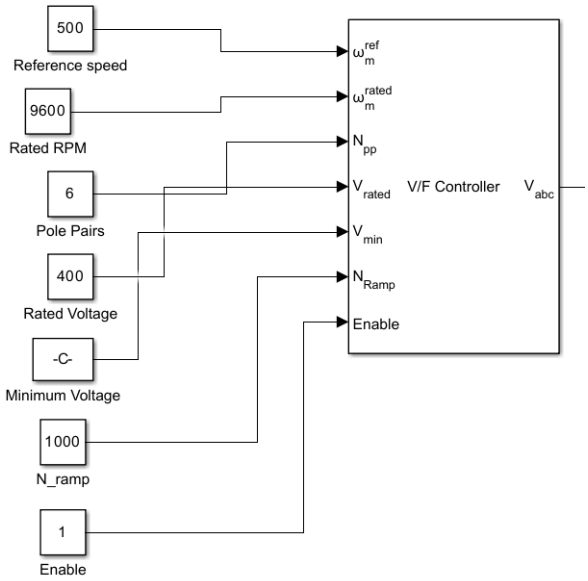
l) Arrastrar el bloque **VbyF Controller** al modelo. Abrir el **Block Parameters** y configurar el **Input speed unit** en RPM y **Discrete step size** en 2*25e-6 s.



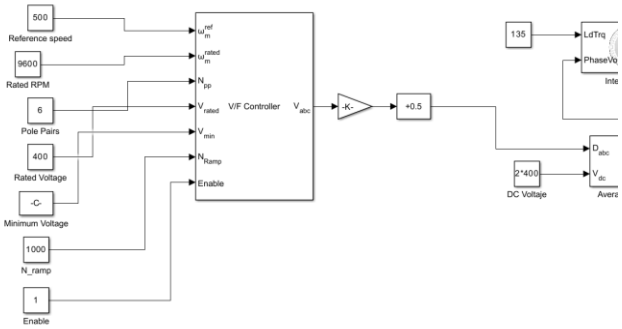
m) Desde **Simulink > Sources**, agregar bloques **Constant** con cada uno de los valores solicitados por el **VbyF**, la mayoría de estos valores están dados en la tabla del literal a.



n) Insertar un bloque **Constant** asiganarle un valor de 1 y conectarlo a la entrada **Enable** del controlador VbyF; para la activación del controlador VbyF.



o) Insertar un bloque **Gain** configurar el valor en 1 / (2*V_rated). Conectar la salida **Vabc** del controlador VbyF al **Gain** e Insertar un bloque **Bias** con valor 0.5 y Conectar la salida final al puerto **Dabc** del subsistema.



4. Resultados

4.1. Gráficas Virtual Composer

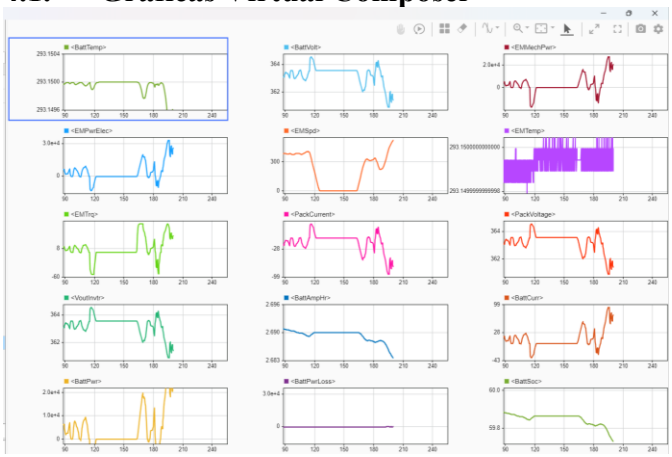


Figura. Gráficas Virtual Composer

4.2. Battery Builder

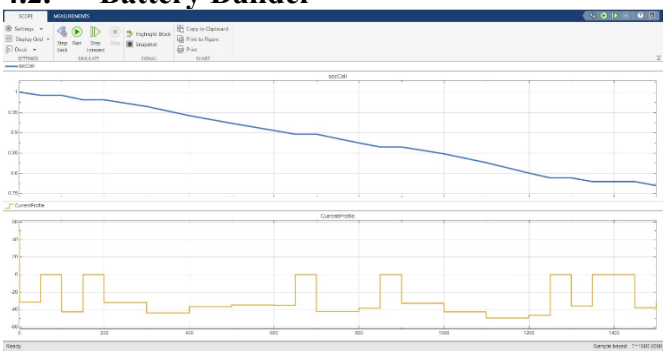


Figura. Gráficas SOC y Corriente de carga

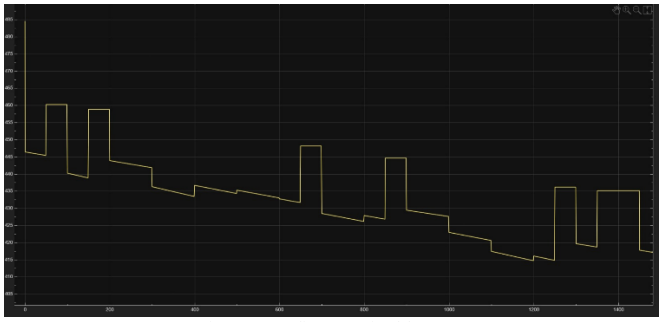


Figura. Gráfica nominal batería

4.3. Motor

Figura. Gráfica funcionamiento motor a velocidad nominal

5. Análisis

5.1. Virtual Vehicle Composer

En la simulación del vehículo eléctrico realizada en MATLAB mediante Vehicle Builder Composer se observa un comportamiento coherente con una ruta que incluye una fase inicial de circulación con baja demanda, seguida de una detención y posteriormente una reaceleración. Este comportamiento se evidencia principalmente en la velocidad del motor eléctrico (EMSpd), la cual primero se mantiene distinta de cero, luego cae prácticamente a cero durante el alto del vehículo y finalmente aumenta nuevamente cuando se retoma la marcha.

Durante la fase de reaceleración se presentan incrementos claros en el par del motor (EMTrq) y en la potencia eléctrica y mecánica del motor (EMPwrElec y EMMechPwr), alcanzando picos característicos de una mayor exigencia de tracción. En estos mismos instantes, el voltaje del pack de baterías (BattVolt y PackVoltage) experimenta caídas moderadas, un comportamiento esperado debido al aumento de la corriente demandada y a la resistencia interna de la batería.

La corriente del pack (PackCurrent y BattCurr) aumenta su magnitud conforme crece la demanda energética del sistema, mientras que la potencia de la batería (BattPwr) refleja este proceso con valores asociados a la entrega de energía durante la tracción y posibles transitorios relacionados con desaceleraciones o eventos de regeneración. Estos cambios confirman una correcta interacción entre la batería, el inversor y el motor eléctrico a lo largo de la ruta simulada.

Finalmente, el estado de carga de la batería (BattSoc) y la capacidad en amperios-hora (BattAmpHr) muestran una disminución leve y progresiva, lo que indica que el recorrido simulado es relativamente corto y que el consumo energético global es reducido. Por otro lado, las temperaturas de la batería y del motor (BattTemp y EMTemp) permanecen prácticamente constantes alrededor de la temperatura ambiente, lo que sugiere que el modelo térmico no entra en un régimen dinámico significativo o se encuentra simplificado, manteniendo un funcionamiento estable y físicamente consistente del vehículo eléctrico.

5.2. Battery builder

Debido a los limites propios de Matlab en cuanto a cantidad de elementos para la creación de las baterías del vehículo Seagull de BYD, se tuvo que realizar la batería en un formato de 2 columnas contrario a lo indicado en la ficha técnica del vehículo para poder alcanzar los valores de voltaje nominal reales de la batería.

El vehículo originalmente cuenta con baterías tipo Blade organizadas de forma distinta a lo convencional en un modelo que la marca denomino Cell-to-Pack, en donde todas las celdas individuales forman directamente el paquete de baterías saltándose toda la organización por bloques y módulos con la finalidad de permitir un mantenimiento menos costoso y fácil de realizar en caso de fallo de las celdas individuales de la batería.

5.3. Introducing to motor control

En el funcionamiento del motor síncrono de imanes permanentes (PMSM) modelado, cada uno de los parámetros definidos por el fabricante influye directamente en su comportamiento eléctrico, mecánico y dinámico dentro del sistema motor–inversor–control implementado en Simulink. Estos parámetros no actúan de forma aislada, sino que determinan conjuntamente la respuesta del motor frente a las señales trifásicas generadas por el inversor y las estrategias de control aplicadas.

El número de pares de polos establece la relación entre la frecuencia eléctrica aplicada y la velocidad mecánica del rotor, de modo que un mayor número de polos reduce la velocidad para una misma frecuencia, pero incrementa la capacidad de generar par a bajas revoluciones, característica clave en aplicaciones vehiculares. La resistencia del estator condiciona las pérdidas por efecto Joule, influyendo en la eficiencia, la disipación térmica y el par disponible, especialmente durante el arranque y las aceleraciones iniciales.

Las inductancias en los ejes d y q definen la dinámica de la corriente, el rizado y la estabilidad del sistema, además de influir en la producción de par por saliencia propia de los PMSM interiores. Por su parte, el flujo magnético de los imanes permanentes determina la constante de par y la fuerza contraelectromotriz, donde valores elevados mejoran la capacidad de generación de par, pero limitan la velocidad máxima del motor debido a la saturación del voltaje disponible.

La inercia del rotor controla la rapidez con la que el motor puede acelerar o desacelerar, justificando la necesidad de rampas de velocidad suaves para evitar picos de corriente y esfuerzos mecánicos excesivos. El voltaje nominal del motor define el rango seguro de operación del inversor y es utilizado para normalizar las señales de control, garantizando una alimentación adecuada y una modulación correcta del sistema de potencia.

En este contexto, el controlador VbyF cumple un rol fundamental en el modelo, ya que permite un control de velocidad en lazo abierto seguro y progresivo, manteniendo constante la relación voltaje–frecuencia durante el arranque y la aceleración. Esta estrategia asegura un flujo magnético prácticamente constante en el estator, reduce sobrecorrientes, evita esfuerzos bruscos en el motor y reproduce el comportamiento real de un vehículo eléctrico al aplicar gradualmente el acelerador, constituyendo una base robusta y confiable para validar el funcionamiento del PMSM antes de implementar estrategias más avanzadas como el control vectorial (FOC).

6. Conclusiones

6.1. Virtual Vehicle Compositer

El Virtual Vehicle Composer permite integrar de forma estructurada y coherente los subsistemas principales del vehículo eléctrico batería, inversor, motor eléctrico, transmisión y vehículo, facilitando una visión global del comportamiento dinámico y energético del EV sin necesidad de desarrollar cada modelo desde cero.

La herramienta resulta especialmente útil para analizar escenarios de conducción realistas, ya que permite evaluar de manera simultánea variables mecánicas, eléctricas y energéticas, como velocidad, par, potencia, corriente, voltaje y estado de carga, asegurando la coherencia física entre ellas.

Los resultados obtenidos evidencian que el Virtual Vehicle Composer es eficaz para identificar transitorios importantes del sistema, como aceleraciones, detenciones y posibles eventos de regeneración, permitiendo evaluar su impacto directo sobre el consumo energético y el desempeño del tren motriz.

6.2. Battery Builder

La aplicación Battery builder de Matlab tiene límites en cuanto a la cantidad de elementos generables para el diseño de baterías, por lo que para la creación de la batería del vehículo Seagull de BYD se tuvo que modificar ligeramente su

construcción con la finalidad de alcanzar los 314V reales que entrega la batería.

A causa de la ausencia de datos en la ficha técnica del vehículo publicada por BYD, el modelo realizado es de baja fidelidad por complejidad de diseño, al no contar con todos los datos necesarios para la creación del sistema de carga y descarga.

6.3. Virtual Vehicle Compositer

El uso de MATLAB–Simulink permitió desarrollar un modelo estructurado, coherente y representativo del comportamiento real de un motor síncrono de imanes permanentes en este caso el de un BYD SEAGULL integrando de manera eficiente los subsistemas eléctricos, mecánico y de control dentro de un mismo entorno de simulación. Esta integración facilitó el análisis simultáneo de variables críticas como velocidad, par, voltajes y corrientes, permitiendo evaluar de forma clara la interacción entre los parámetros definidos por el fabricante, el inversor y la estrategia de control implementada.

A partir del análisis de los datos obtenidos en simulación, se comprobó que Simulink permite identificar de manera temprana el impacto de cada parámetro del motor sobre la respuesta dinámica del sistema, tales como aceleraciones, estabilidad de la velocidad y comportamiento ante cambios de frecuencia. La posibilidad de modificar valores de forma paramétrica y observar su efecto inmediato en las señales de salida constituye una ventaja significativa frente a ensayos físicos, reduciendo tiempos de desarrollo y evitando riesgos asociados a pruebas experimentales iniciales.

Referencias

[1] MathWorks, *Motor Control Blockset™ User's Guide*, Natick, MA, USA: The MathWorks, Inc., 2024. [En línea]. Disponible en: https://www.mathworks.com/products/motor-control.html

[2] MathWorks, *Simscape Electrical™ Documentation*, Natick, MA, USA: The MathWorks, Inc., 2024. [En línea]. Disponible en: https://www.mathworks.com/help/physmod/sps/