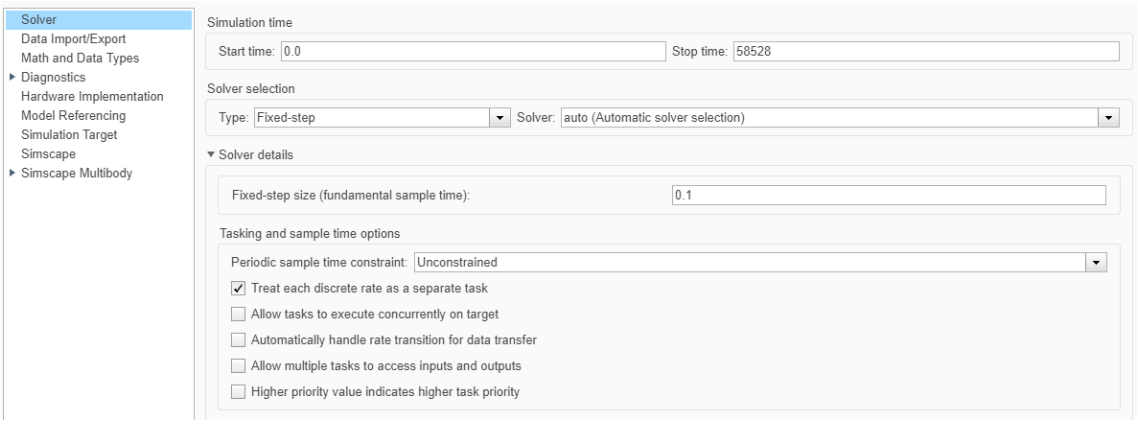
Ayuda para lanzar la optimización del Coursework

1. Cambiar el tiempo de paso máximo en el modelo:

Modeling → Model Settings → Solver → Max step size

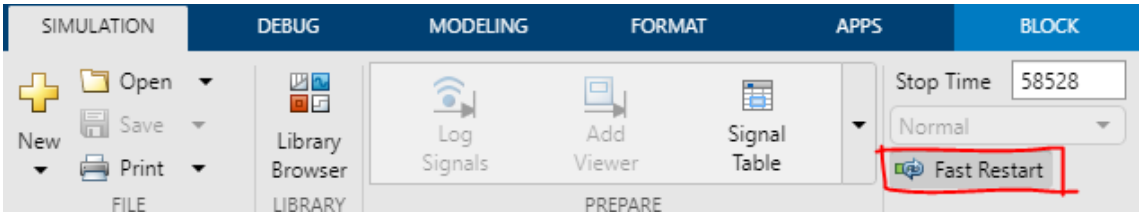
Por defecto, generalmente tenemos este parámetro a 1s. Os recomiendo que lo pongáis más restrictivo, a **0.1s**, ya que sino vais a tener problemas para identificar los parámetros en los pulsos. Si ponéis el solver en type: fixed-step, os irá más rápido también



2. Modelo para las optimizaciones:

Si activáis el ***Fast Restart*** la estructura del modelo se mantiene igual pero las optimizaciones van a ir más rápidas.

Más información en: https://es.mathworks.com/help/simulink/ug/how-fast-restart-improves-iterative-simulations.html



Cuidado: si tenéis ese botón activado no os dejará editar el modelo.

Stop Time → tiempo de ejecución del modelo. Conviene poner este parámetro como un parámetro de inicialización en el script de Matlab.

3. Parámetros a incluir en la optimización:

Paso 1: optimizar solo la curva de tensión

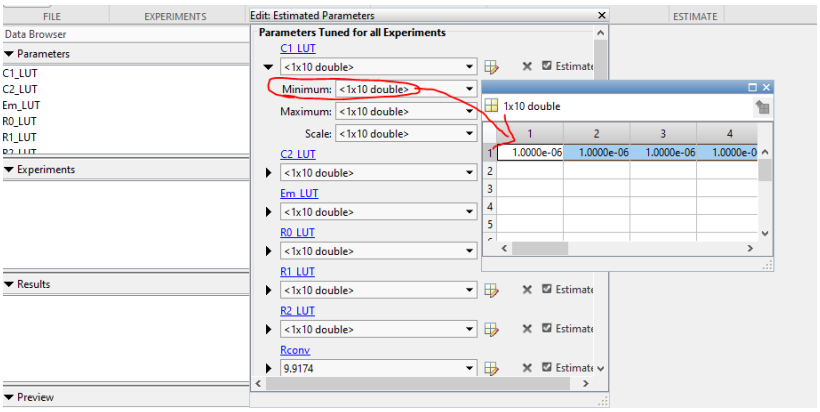
- R0(SoC), R1(SoC), R2(SoC), C1(SoC), C2(SoC) → mínimo estos, sólo en función del SoC, ya que cada grupo hará la simulación a una temperatura concreta.
- Em_LUT(SoC) → tenemos los datos experimentales, pero conviene incluirlo para refinar los datos. No deberíamos incluirlo, pero debido cómo están sacados los datos experimentales os recomiendo añadirlo

Paso 2: optimizar solo la curva de temperatura

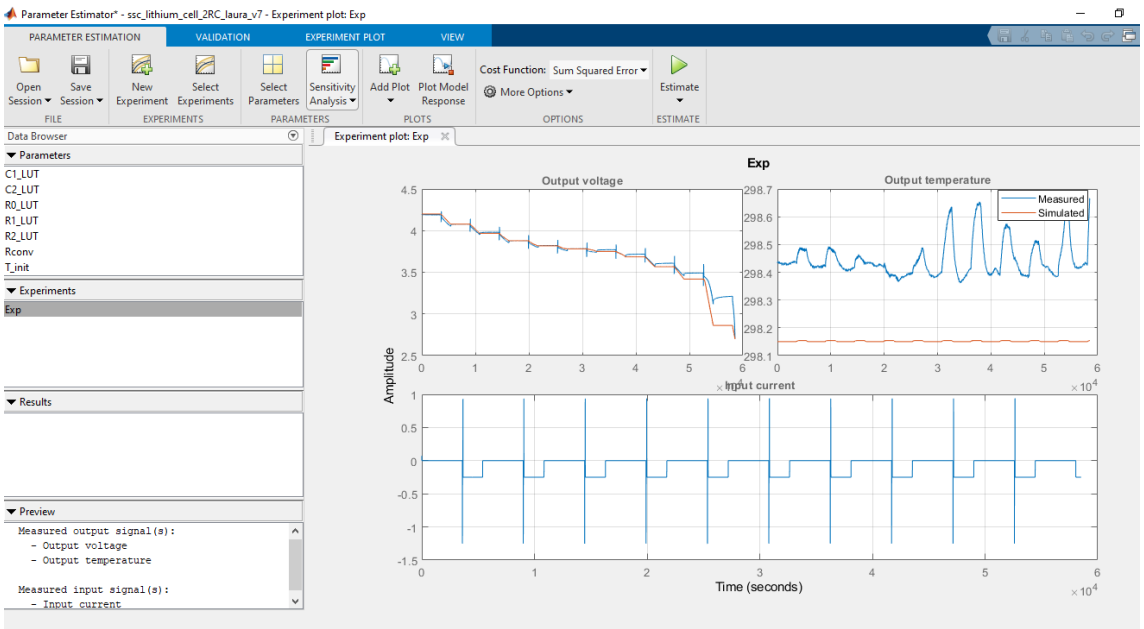
- Refinar el parámetro hconv → Es posible que la mejora no sea muy significativa, ya que las variaciones en temperatura son muy pequeñas (menores de un grado)

Es importante que los límites de los parámetros estén bien definidos, sino se os parará la estimación de parámetros. Para definir los límites inferiores y superiores se abre el desplegable siguiente:

- Límite inferior de C1_LUT, C2_LUT, R0_LUT, R1_LUT, R2_LUT, h_conv -> 10^-6 (siempre superior a cero, estos parámetros no pueden ser negativos)
- Límite superior de R0_LUT, R1_LUT, R2_LUT → = 10 (es mejor tener acotadas las resistencias)
- Límite inferior de Em_LUT = 2.7 V (CoV inferior), Límite superior de Em_LUT = 4.2 V (CoV superior)

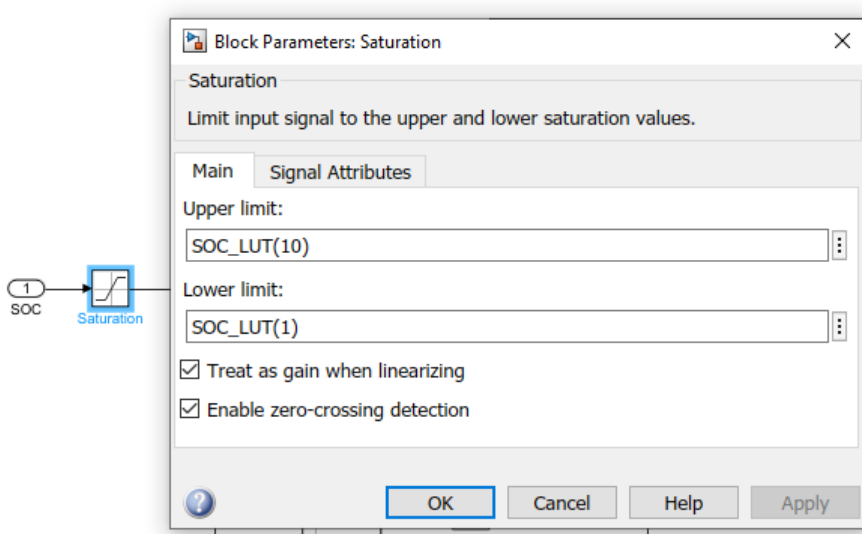


- Para el apartado B del coursework, pondréis los parámetros en función de T y SoC, con los parámetros de vuestros compañeros. Quedando de esta manera: R0(SoC,T), R1(SoC,T), R2(SoC,T), C1(SoC,T), C2(SoC,T), hconv(T), T_init(T)



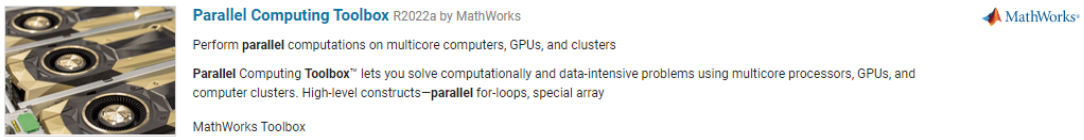
4. Inclusión de un bloque de saturación de SoC **antes del LookUp Table de los parámetros**.

SOC_LUT(10) corresponde a 100% y SOC_LUT(1) a 0%. De esta manera evitaréis problemas durante la optimización pero a la vez, podréis ver en los gráficos que saquéis si el SoC es inferior a 0 o superior a 100 (que no debería pasar nunca).



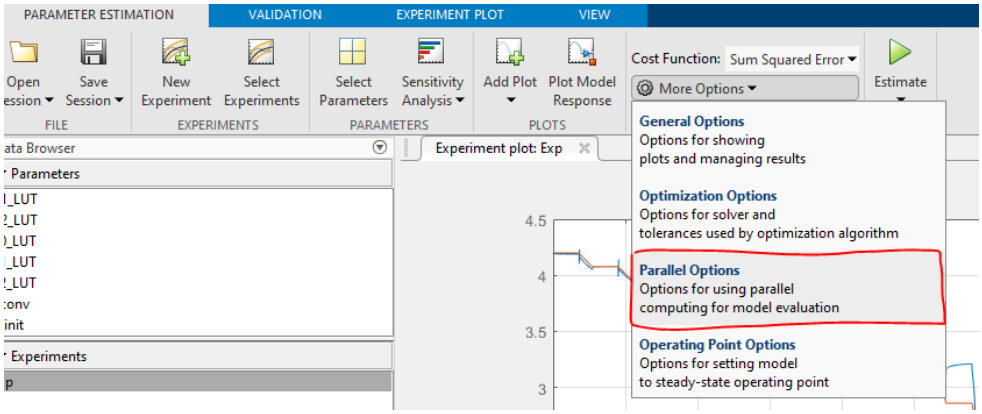
5. Para que las simulaciones ejecuten más rápido

Existe una opción de paralelizar las optimizaciones (recomendable). Para ello deberéis tener instalado el siguiente toolbox de Matlab:

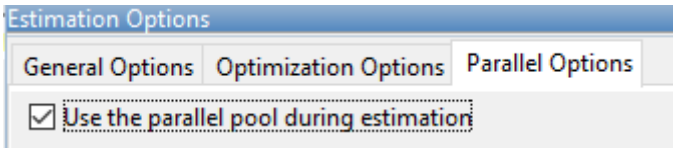


Una vez tengáis ese toolbox instalado, los pasos a dar son los siguientes:

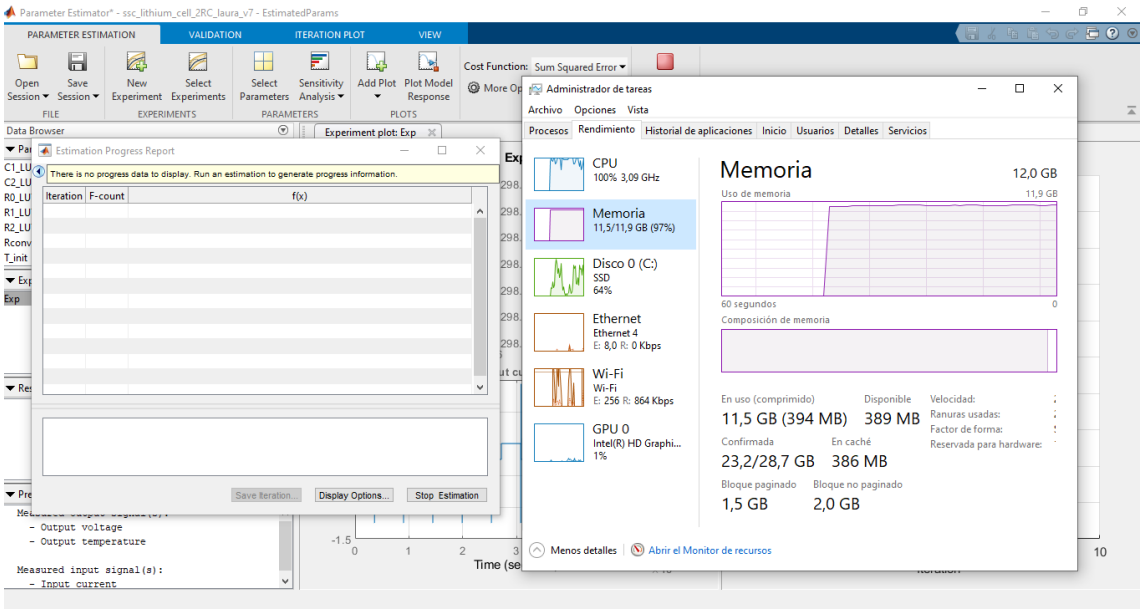
- Dentro del Parameter Estimation → More Options → Parallel Options



- Activar la opción de “Use the parallel pool during estimation”



Es recomendable que esta opción solo la utilicéis cuándo no queráis hacer uso intensivo de vuestro ordenador, ya que os irá más lento porque está cogiendo más recursos para realizar esta optimización (ejemplo de mi ordenador):

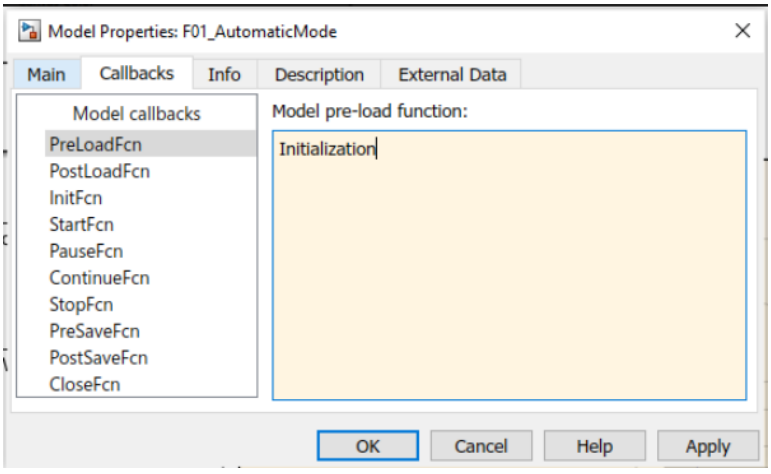


6. Inicialización de los parámetros desde un script de Matlab “Inicialization.m” desde Simulink directamente

Esto es de gran ayuda y recomendable para asegurarnos de que todos los parámetros se cargan desde el .m seleccionado y no has tenido que ejecutar un script de Matlab previamente.

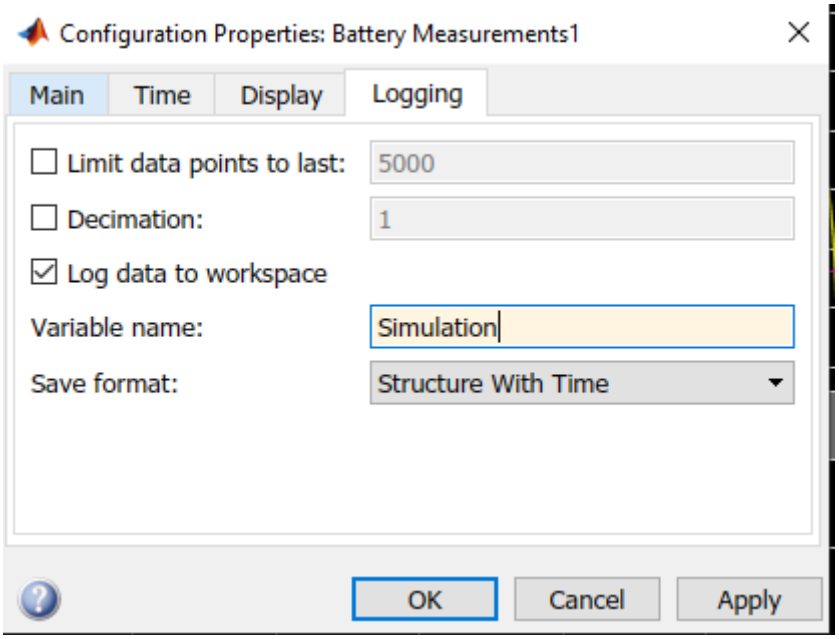
Pasos para activar esta funcionalidad:

- Tener el script de Matlab creado: “Inicialization.m”
- Dentro del modelo de simulink, en una zona en blanco, click derecho, Model properties
- Callbacks\PreLoadFcn y definir el fichero de inicialización que quieres ejecutar al abrir el modelo
- Este Initialization.m tiene que estar en el mismo directorio que el modelo, y estar en el directorio de trabajo de matlab



7. Guardar los datos desde el Scope de simulink al workspace de Matlab para poder calcular los errores

Dentro del Scope de resultados → configuration properties → Logging → log data to workspace



Aparece esta estructura en el workspace: