

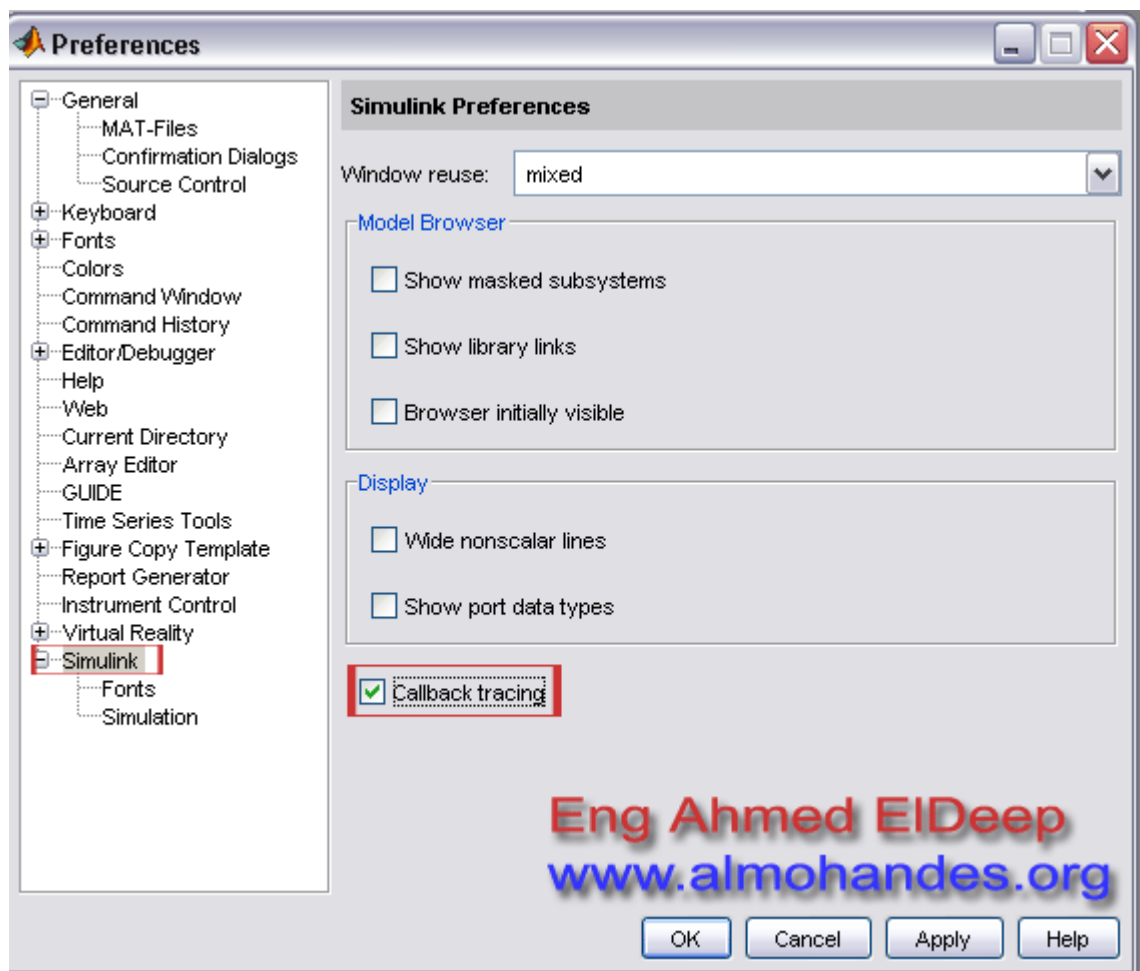
الدرس الثامن

Using Callback Functions

يمكنك تعريف أوامر ماتيابل يتم تنفيذها عند تفعيل نموذج معين وتسمى هذه الاوامر بي callback functions

Tracing Callbacks

وهذه الخاصية تمكنا من معرفة الاوامر المستدعا (Tracing Callbacks) التي يستحضرها السميولينك وما هي كيفية تنفيذ هذه الاوامر وتوقيتها ولتفعيل هذه الخاصية ادخل على Simulink Preferences ثم اختر Callback tracing كما يلي :



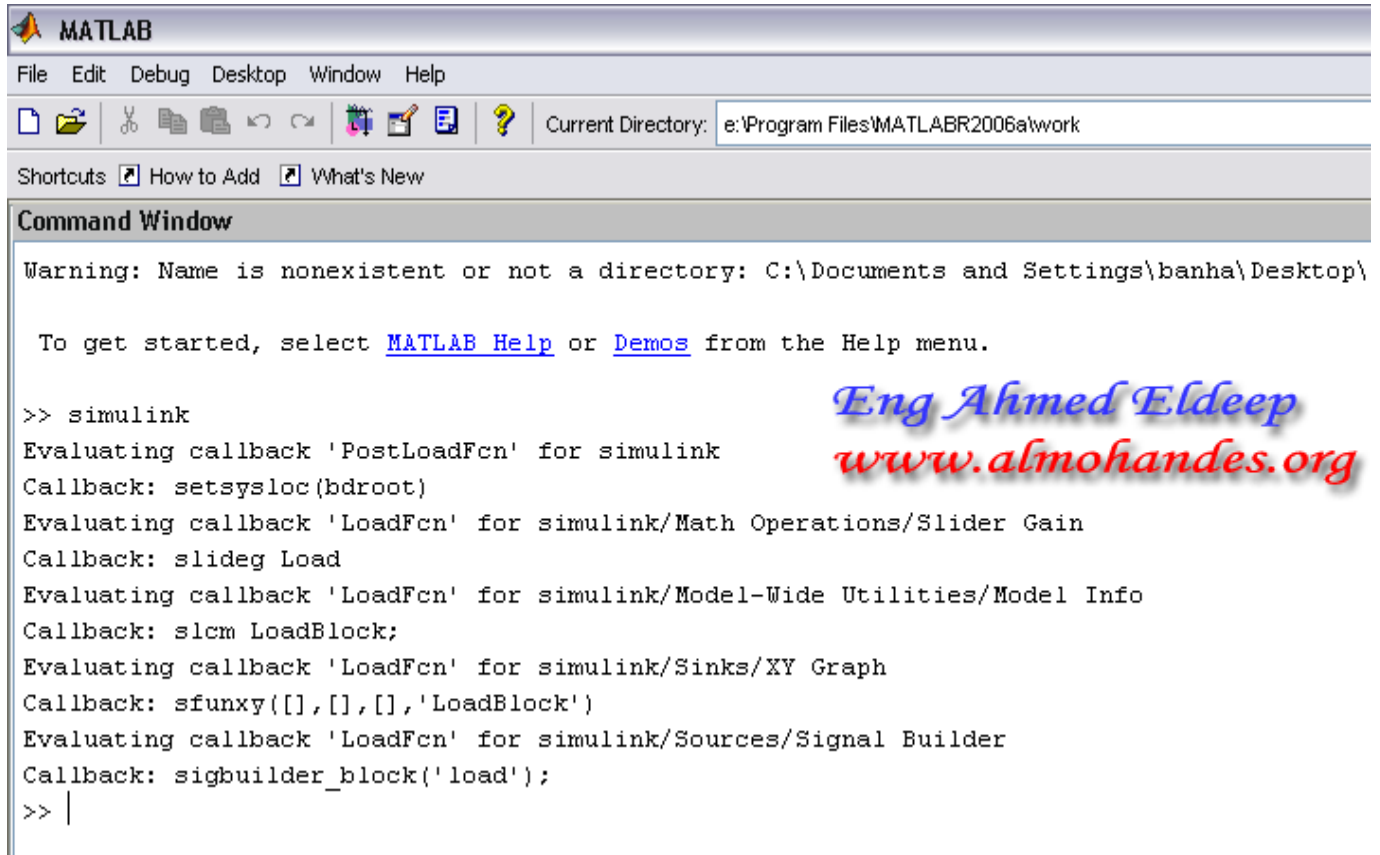
او من خلال تنفيذ الامر الاتي في سطر أوامر الماتلاب

رمز:

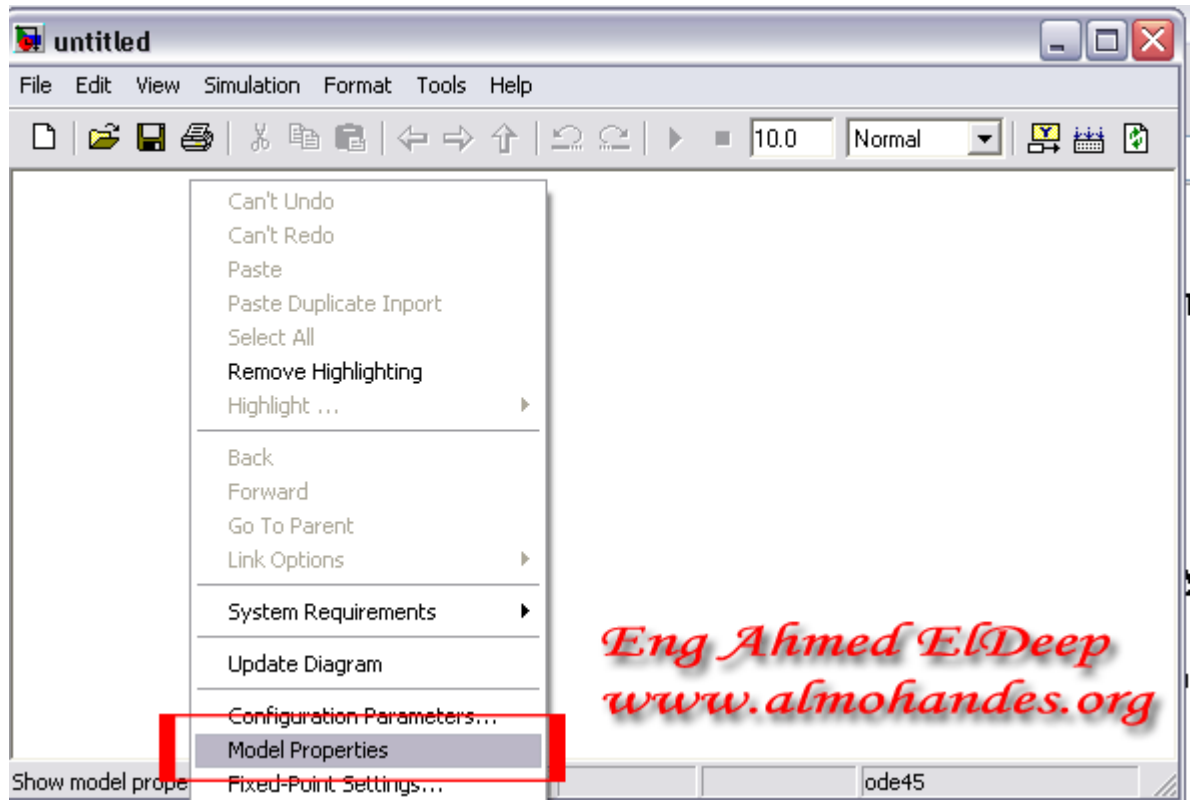
كود

```
>> set_param(0,'callbacktracing','on')
```

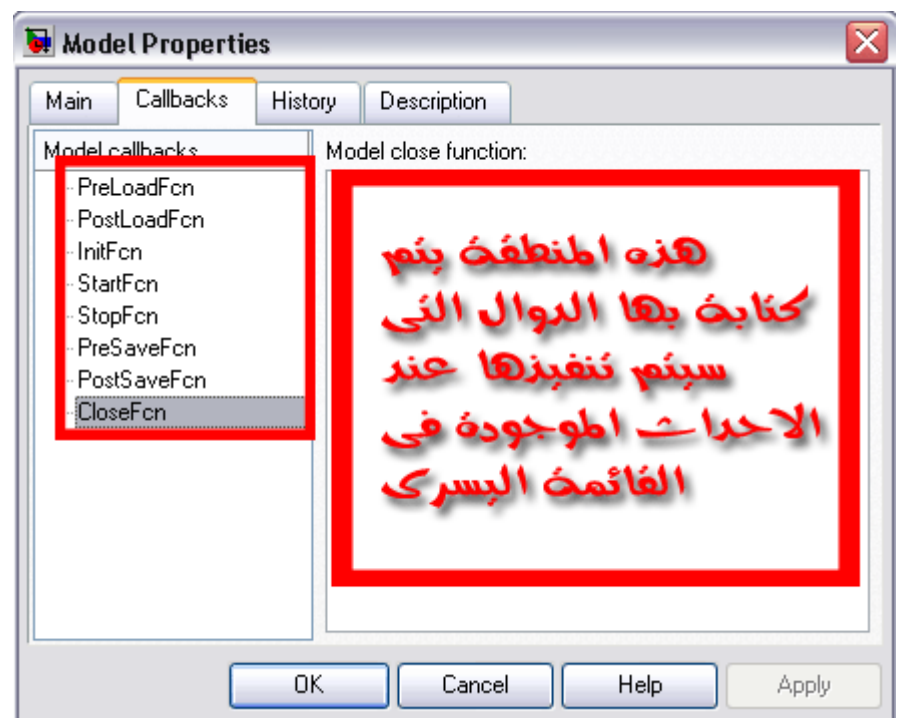
ويتم عرض الاوامر التي يستحضرها السميولينك عند تنفيذ عمل معين في سطر
أوامر الماتلاب
فمثلا قم باغلاق السميولينك واعادة تشغيله مرة اخرى وستلاحظ ما يلي :



والان سنتعلم كيفية عمل هذه الاوامر
وتكون بطريقتين وهما باستخدام السميولينك او باستخدام البرمجة
والان سنبدأ في التعرف على **Callbacks pane** وهى التى تظهر عند اختيار
Model Properties عند الضغط كليك يمين



وستظهر لنا النافذة الآتية



وفي الجدول التالي نجد توقيت حدوث كل أمر من أوامر ال Callbacks pane

Parameter	When Executed
CloseFcn	Before the block diagram is closed.
PostLoadFcn	After the model is loaded. Defining a callback routine for this parameter might be useful for generating an interface that requires that the model has already been loaded.
InitFcn	Called at start of model simulation.
PostSaveFcn	After the model is saved.
PreLoadFcn	Before the model is loaded. Defining a callback routine for this parameter might be useful for loading variables used by the model.
PreSaveFcn	Before the model is saved.
StartFcn	Before the simulation starts.
StopFcn	After the simulation stops. Output is written to workspace variables and files before the StopFcn is executed.

ويمكن ايضا عمل Callbacks خاصة ببلوك معين وايضا عمل Callbacks خاصة بـ port معين فى هذا البلوك

Working with Model Workspaces

يكون لكل نموذج Workspaces خاصة به لتخزين متغيراته بها والتي تشبه ال Workspaces الموجودة فى الماتلاب وتكون هذه المتغيرات مرئية باستخدام scope .

ويمكننا من خلال ال Workspaces عمل متغيرات لنموذج معين وعند فتح تحميل هذا النموذج فان قيم هذه المتغيرات يتم تحميلها من مصدر للبيانات ومصادر البيانات هى ::

MDL-file وهو الملف الخاص بالموديل

MAT-file

M-code

و انواع ال objects المستخدمة فى تخزين البيانات الموجودة فى ال Workspaces هى

Simulink.Parameter objects

Simulink.Signal objects

وكل هذه المتغيرات الموجودة فى Workspaces لا يمكن تعديلها اثناء عملية المحاكاة اى انها تعتبر **not tunable**

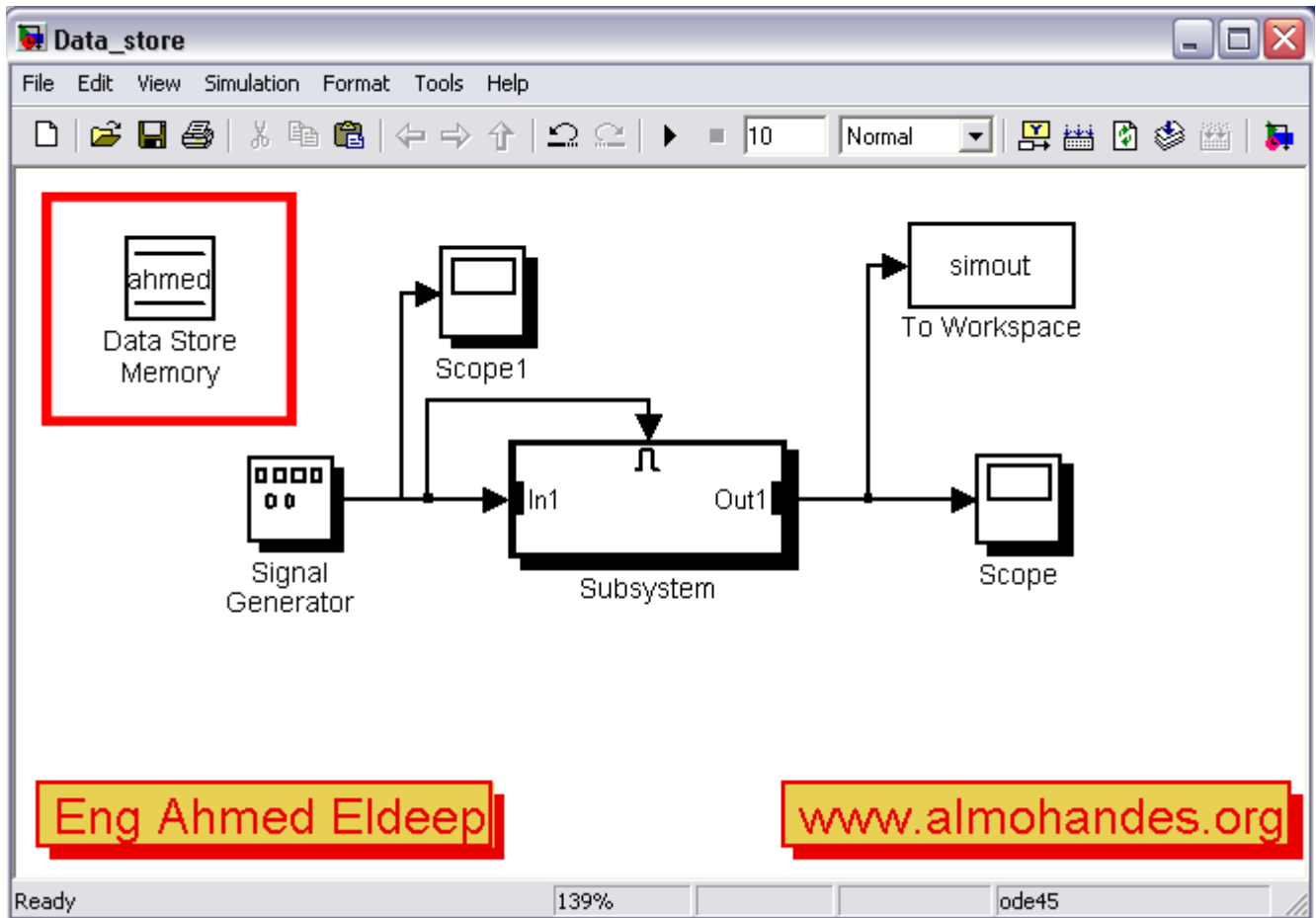
ماعدات المتغيرات التي نستخدمها لل model arguments لعمل نماذج مرجعية .

Working with Data Stores

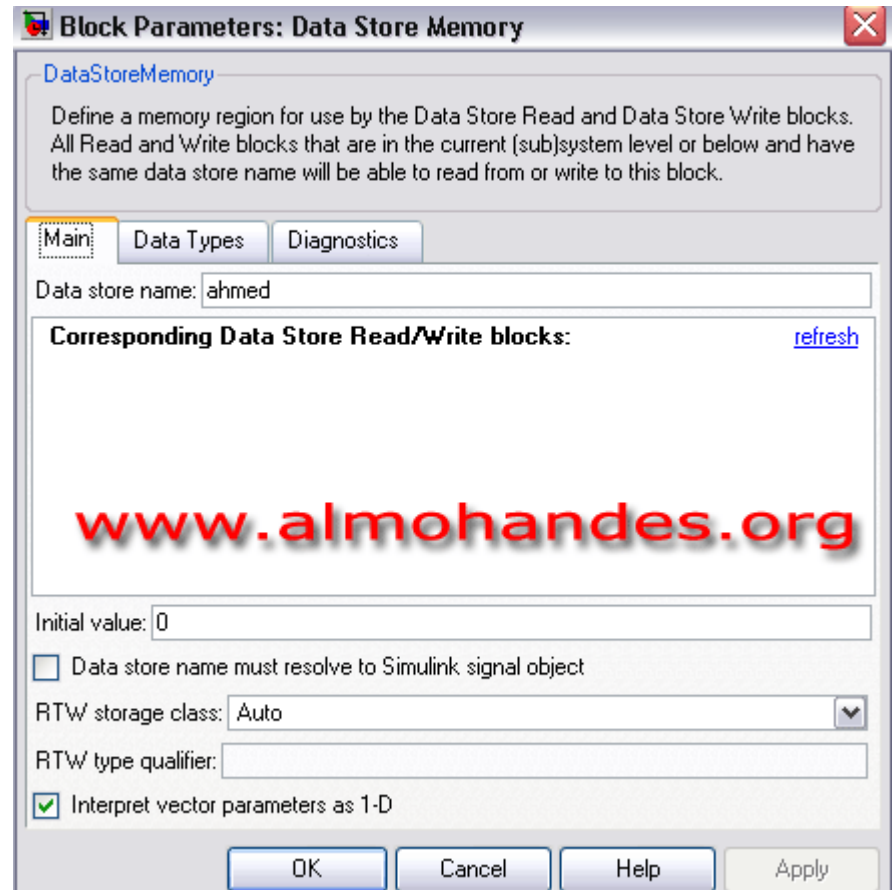
مخازن البيانات هي اشارات تكون متاحة في اى نقطة في النموذج سواء في النموذج الاصلى النموذج الفرعى او النموذج المرجعى ومن مميزاتا ان تسمح بتبادل البيانات بين النموذج الفرعى او المرجعى وبين النموذج الاصلى (فى مستوى أعلى)

ويمكننا استخدام هذه المميزات عن طريق اضافة بلوك Data Store Memory او استخدام Simulink.Signal class

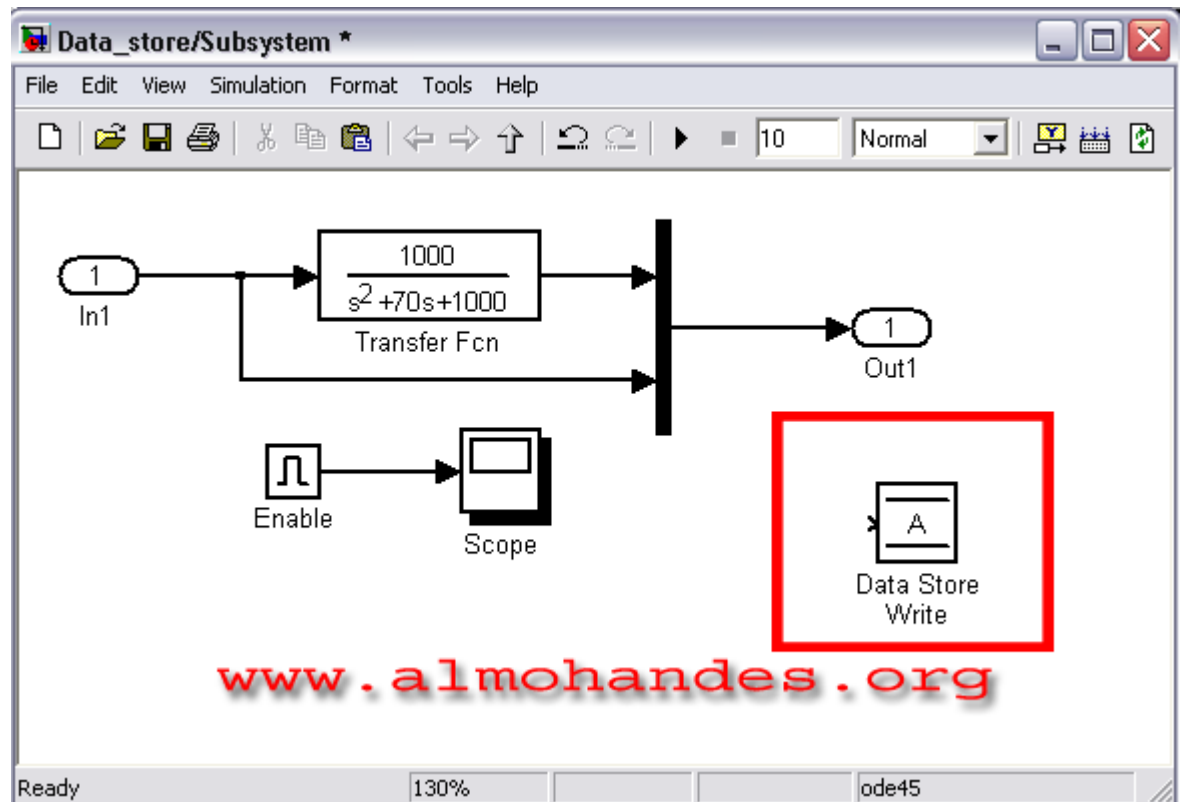
والان سنبدأ فى اضافة بلوك data store ونقوم بوضعه فى اى مكان فى النموذج وحتى تكون هذه البيانات ظاهرة فى جميع أجزاء الموديل يجب وضع هذا البلوك فى اعلى مستوى من النموذج وهو مستوى ال root
والان قم بالدخول الى signal routing
من على اليمين اسحب بلوك data store الى الموديل كما بالشكل الاتى



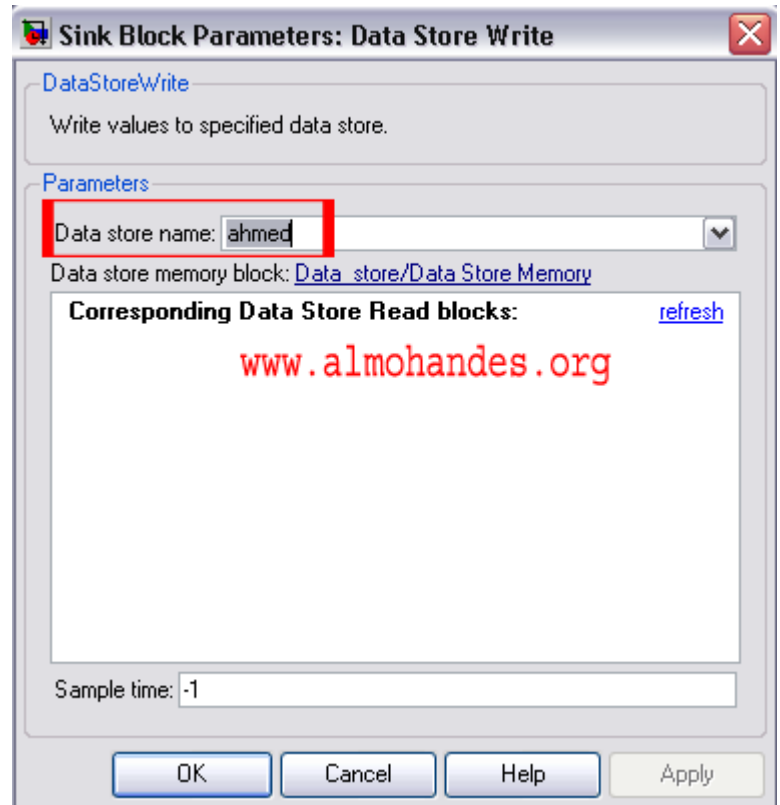
ثم قم بالضغط عليه لتحديد اسمه كما بالشكل الاتي :



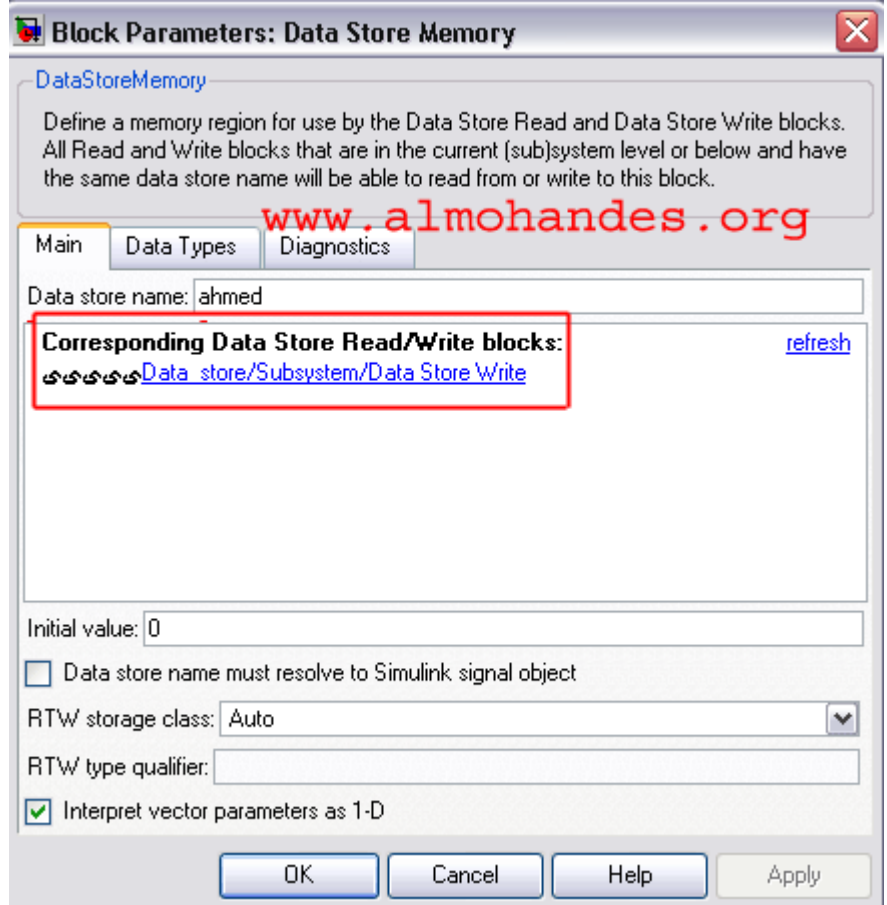
والان فى النظام الفرعى الموجود لدينا نقوم بوضع بلوك **Data Store Write** او فى اى نظام موجود لدينا فى النموذج نريد تخزين بياناته كما بالشكل الاتي :



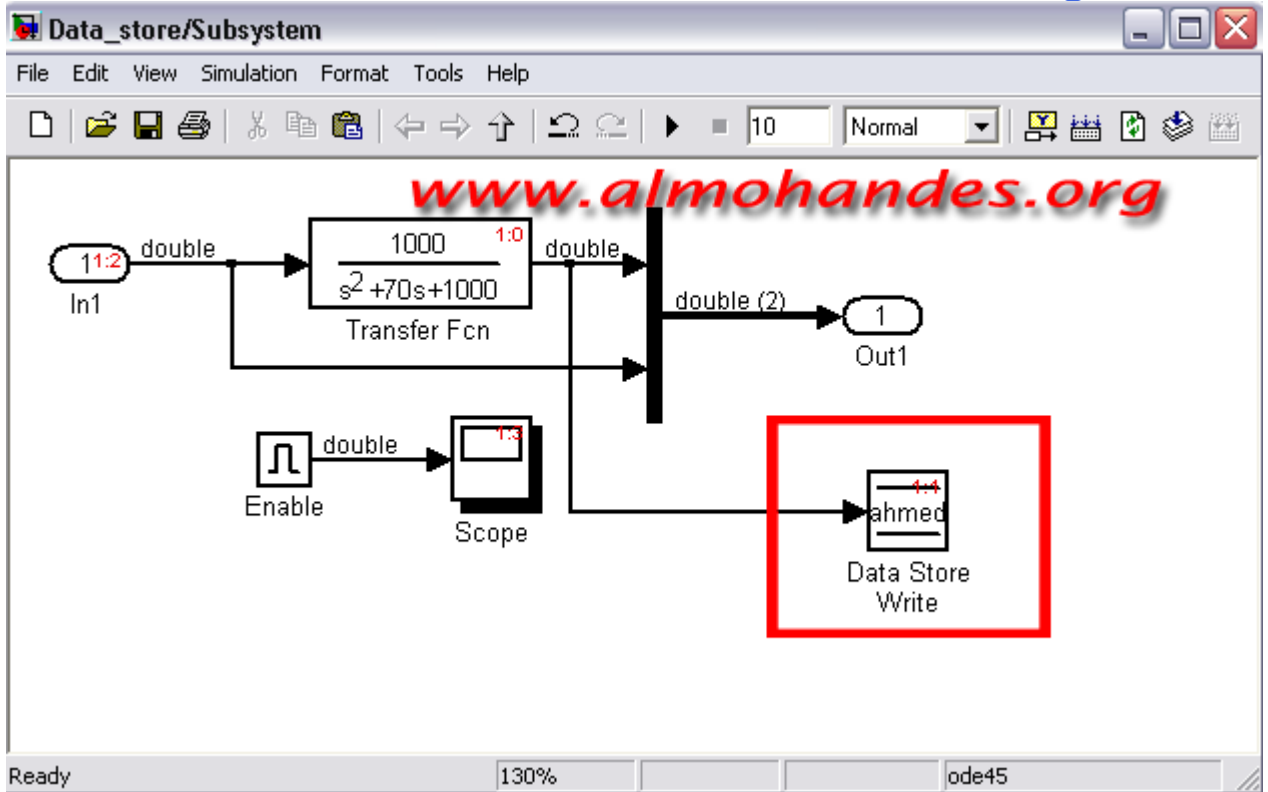
ثم نقوم بالدخول الى خصائصه بالضغط عليه مرتين وكتابة اسم **data store** **name** مثل التي تم تحديدها في بلوك **data memory store** الموجود في المستوى الاعلى



والان قم بالضغط على بلوك **data memory store** الموجود في المستوى الاعلى
وسنلاحظ انه تم التعرف على بلوك **Data Store Write** الذي قمنا بوضعه في
المستوى الفرعى وبالضغط عليه سوف يوجهنا اليه

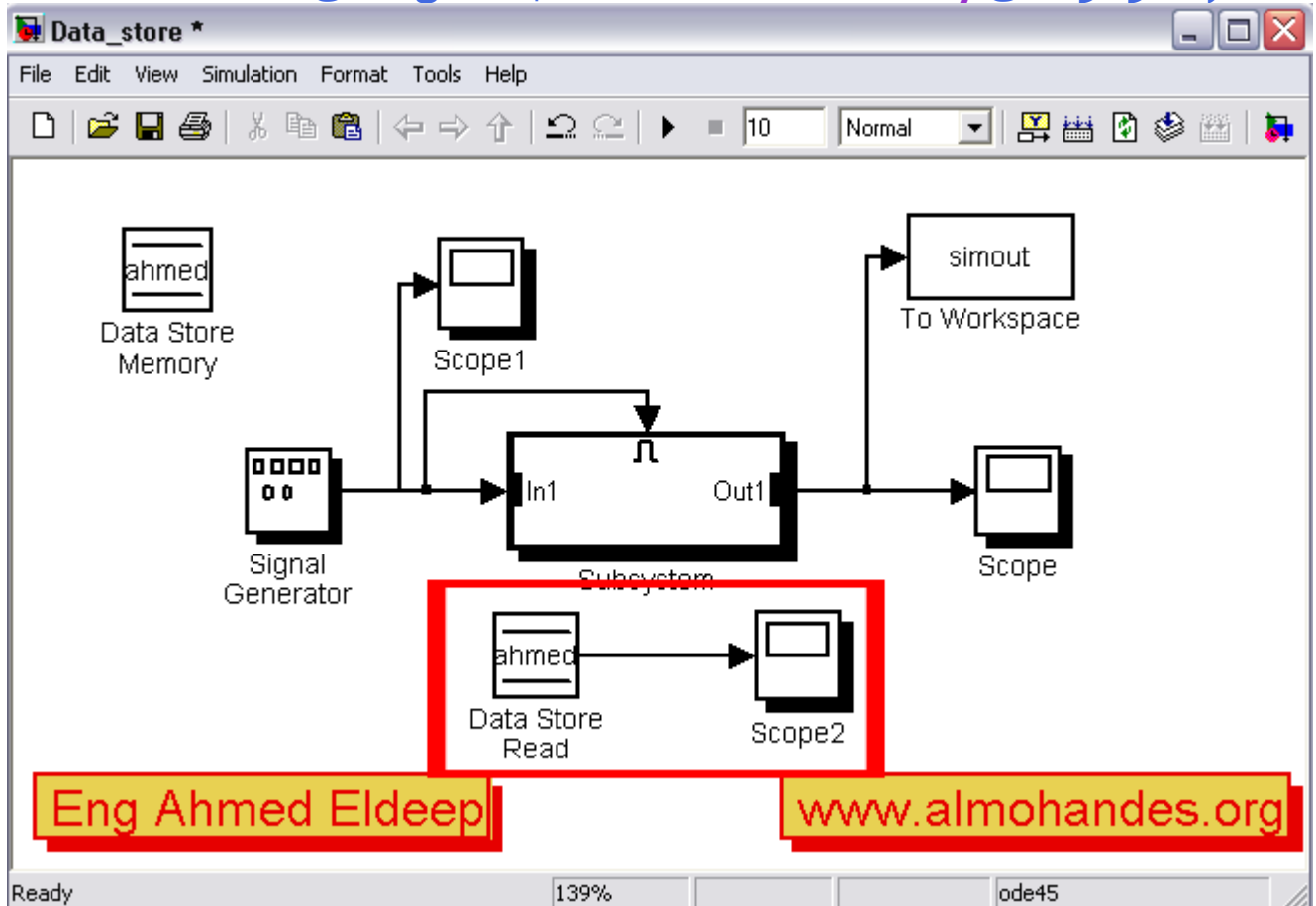


والان قم بتوصيل بلوك **Data Store Write** بمصدر البيانات المطلوب كما بالشكل الاتي :

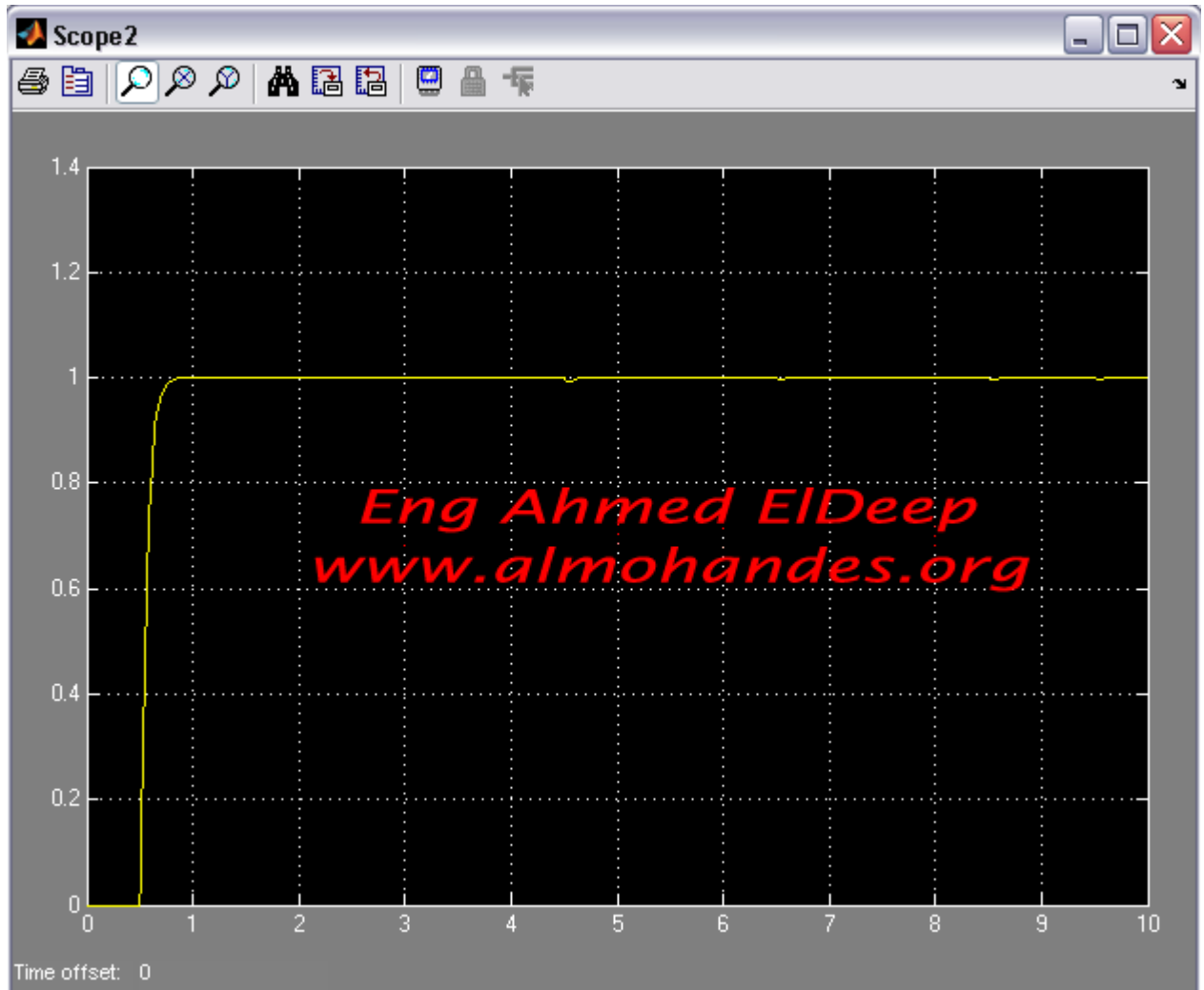


وللحصول على البيانات التي يتم تخزينها عند كل فترة زمنية نقوم باضافة بلوك **Data Store Read** فى اى مستوى من النموذج نريد فيه استخدام هذه القيمة ولا يشترط ان يكون فى نفس المستوى مع **Data Store Write** فمثلا سنقوم بوضع البلوك فى المستوى الاعلى root ونوصله بى scope لعرض البيانات مع مراعاة الدخول الى خصائصه وتغيير اسم **data store name** الى

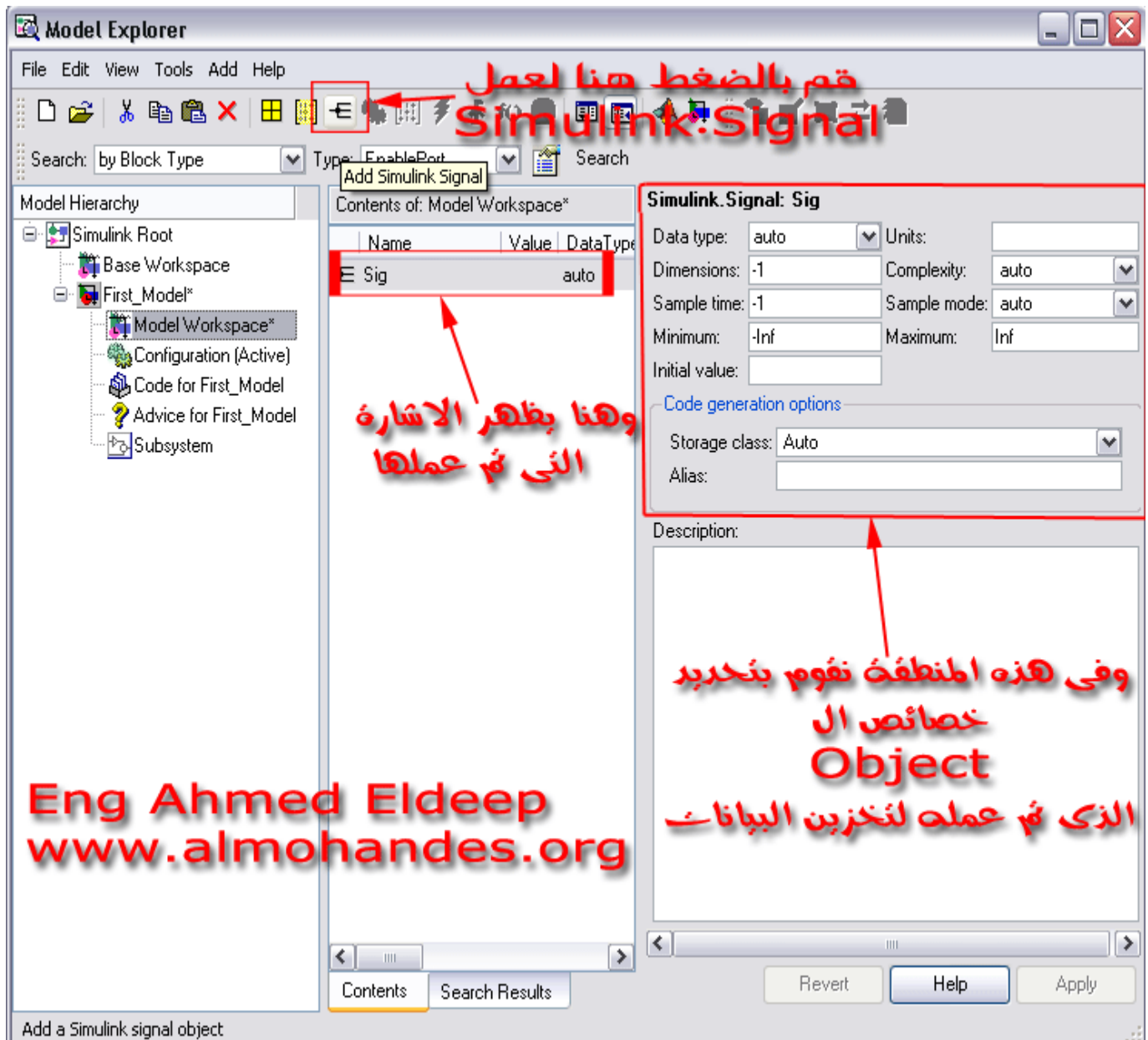
الاسم الموجود في data store memory كما بالشكل الاتي :



والان أصبحنا جاهزين لان نقوم بعملية المحاكاة
قم بالذهاب الى المستوى الاعلى ثم أضغط على بداية المحاكاة وسيكون شكل
ال scope الموجود بعد بلوك Data Store Read كما بالشكل التالى :



ولاستخدام `Simulink.Signal` class
قم بعمل `Simulink.Signal` فى ال `workspace`. وتكون عن طريق الدخول الى
Model Explorer كما بالشكل الاتى :



ومثلا نقوم بعمل object له الخصائص الاتية : data store name التي تم وضعها في المستوى الاعلى من النموذج

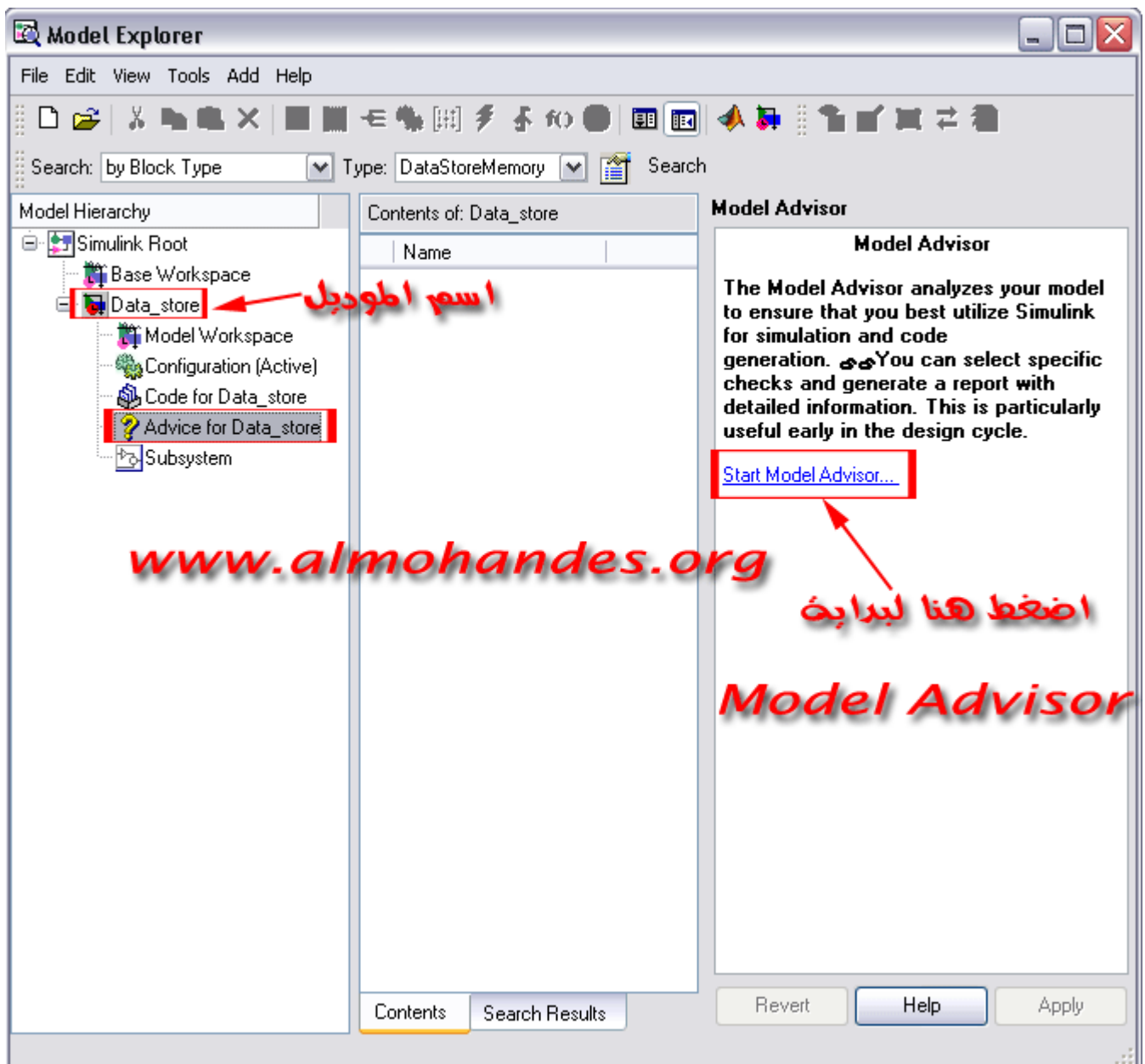
```
ahmed = Simulink.Signal;  
  
ahmed.Description = 'Use to signal that subsystem output is  
  
invalid';  
  
ahmed.DataType = 'boolean';  
  
ahmed.Complexity = 'real';  
  
ahmed.Dimensions = 1;  
  
ahmed.SamplingMode='Sample based';  
  
ahmed.SampleTime = 0.1;
```

ويمكننا تخصيصها ايضا بكتابة الاوامر السابقة فى سطر الماتلاب
اما بالنسبة للجزء السفلى من الخصائص والخاص بى **StorageClass** فيجب ان
يكون **auto** فى حالة تخزين البيانات فى مستوى واحد **local**

Consulting the Model Advisor

من أهم خصائص برنامج السميولينك هي وجود ال **Model Advisor** وهو يقوم بعمل اختبار للنموذج او النظام الفرعى والتأكد من خصائصه و اخطاءه التى قد تسبب عدم دقة فى الحسابات وظهور نتائج غير متوقعة خلال عملية المحاكاة وبعد الانتهاء من هذا الاختبار يقدم لنا **Model Advisor** تقرير يحتوى على جميع ما يخص النموذج والاقتراحات المطلوبة لتحسين الاداء .

ولبدء عمل **Model Advisor** قم بالدخول الى قائمة **tools** ثم اختر **Model Advisor** او من خلال الدخول الى **contents** فى ال **The Model Explorer** ثم اختر **Advice for** مع تحديد اسم النموذج كما بالشكل الاتى

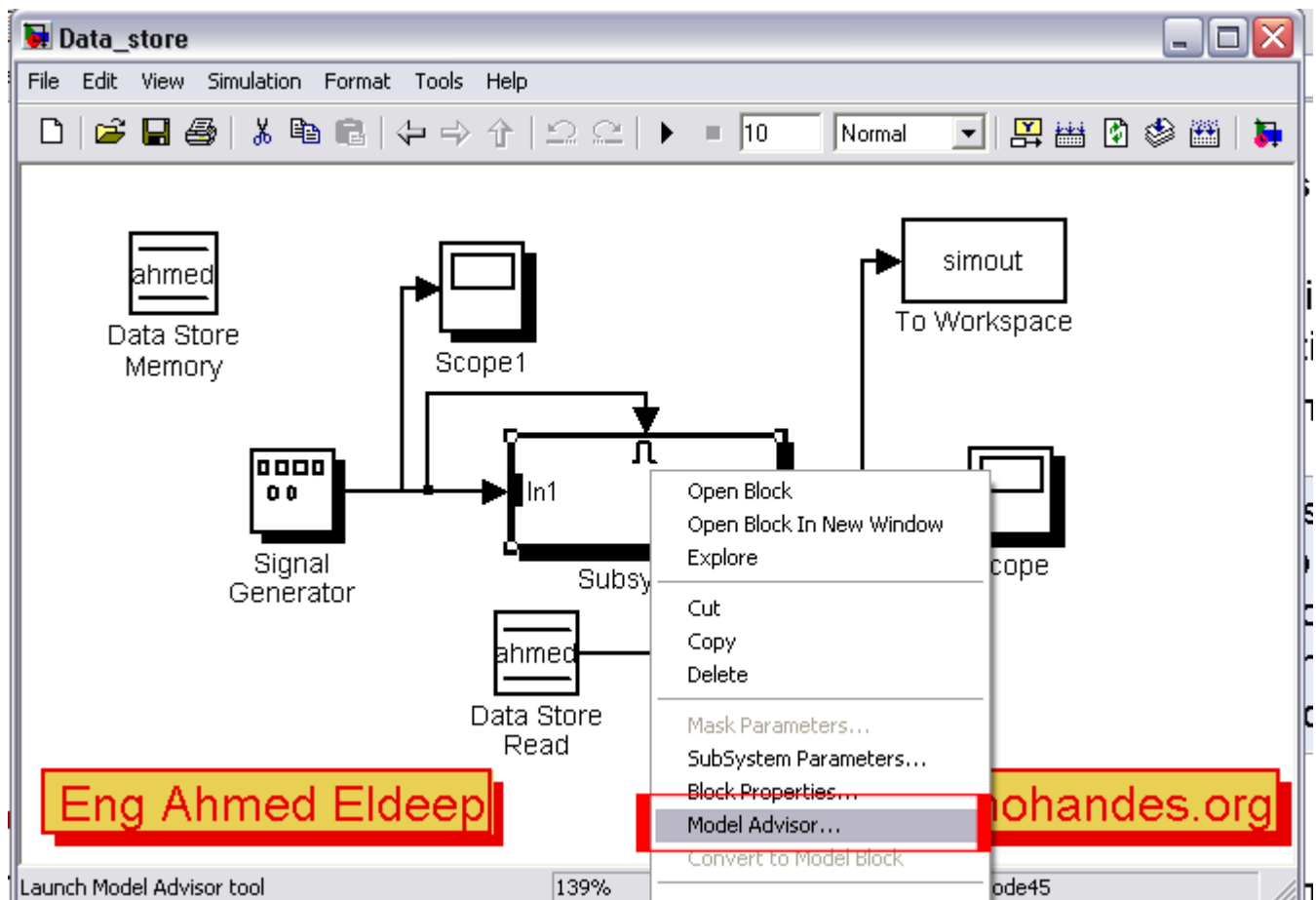


وايضا من خلال سطر أوامر الماتلاب يمكننا كتابة الامر الاتى

رمز:
كود

```
>>modeladvisor(model_name)
```

وايضا من خلال الضغط كليك يمين على النموذج الفرعى واختيار **Model Advisor** كما فى الصورة الاتية



ويجب ملاحظة ان ال **Model Advisor** يقوم بتخزين التقارير فى الملف **slprj** الذى يكون موجود مع اى ملف موديل

وعند بداية ال **Model Advisor** يكون على الشكل الاتى



ويمكننا أيضا تشغيل ال Model Advisor من خلال برنامج M-file