



Le moteur à courant continu

Ajuster les paramètres du modèle causal à partir des mesures

Table des matières

1. Création du signal d'entrée du modèle causal.....	2
2. <i>Modèle causal du motoréducteur pour recherche des paramètres.....</i>	10
3. <i>Recherche des paramètres.....</i>	15

1. Création du signal d'entrée du modèle causal

Les résultats d'une simulation ne sont pas ne concordent pas en général avec les mesures car les paramètres du modèle utilisé n'ont pas les bonnes valeurs. Ceci en supposant bien sur que le montage expérimental soit le plus correct possible et que les erreurs de mesures ne soient que de type aléatoire et non pas provoquées par des erreurs de manipulation ! Matlab permet d'ajuster les paramètres d'un modèle en faisant des simulations successives et en variant petit à petit ses paramètres jusqu'à ce que la différence entre le résultat obtenu et les mesures soit inférieure à une certaine valeur imposée par l'utilisateur.

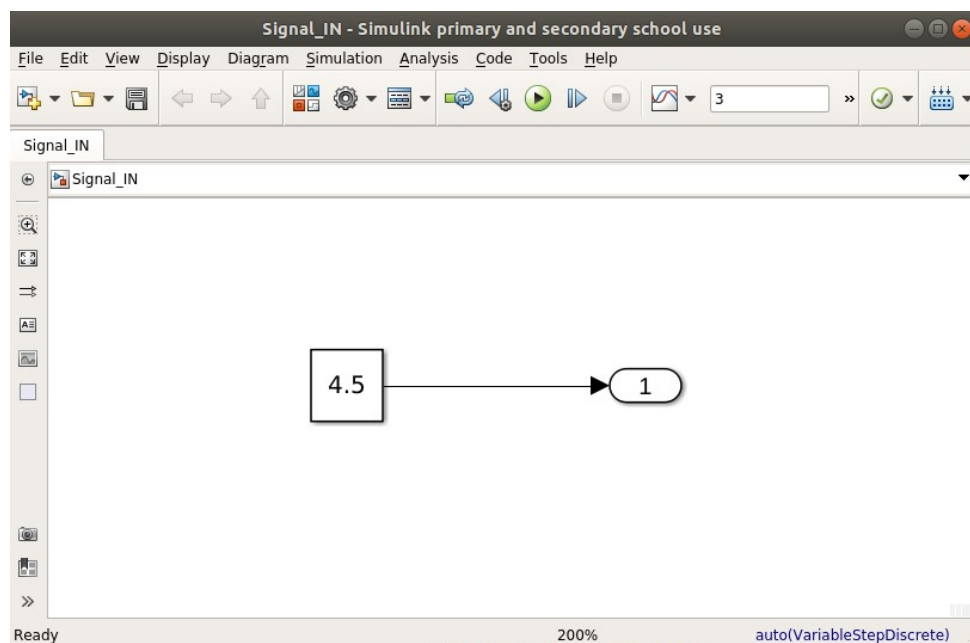
Un modèle causal comme celui utilisé dans le « TP Modélisation causale d'un motoréducteur » doit être légèrement modifié :



- Le modèle doit recevoir les signaux initiaux sous la forme des vecteurs présents dans le Workspace :
 - o la tension aux bornes du moteur en fonction du temps
 - o la vitesse de rotation mesurée en fonction de temps
- Les paramètres à rechercher doivent avoir un nom et une valeur initiale

La tension appliquée aux bornes du moteurs est associée à deux vecteurs :

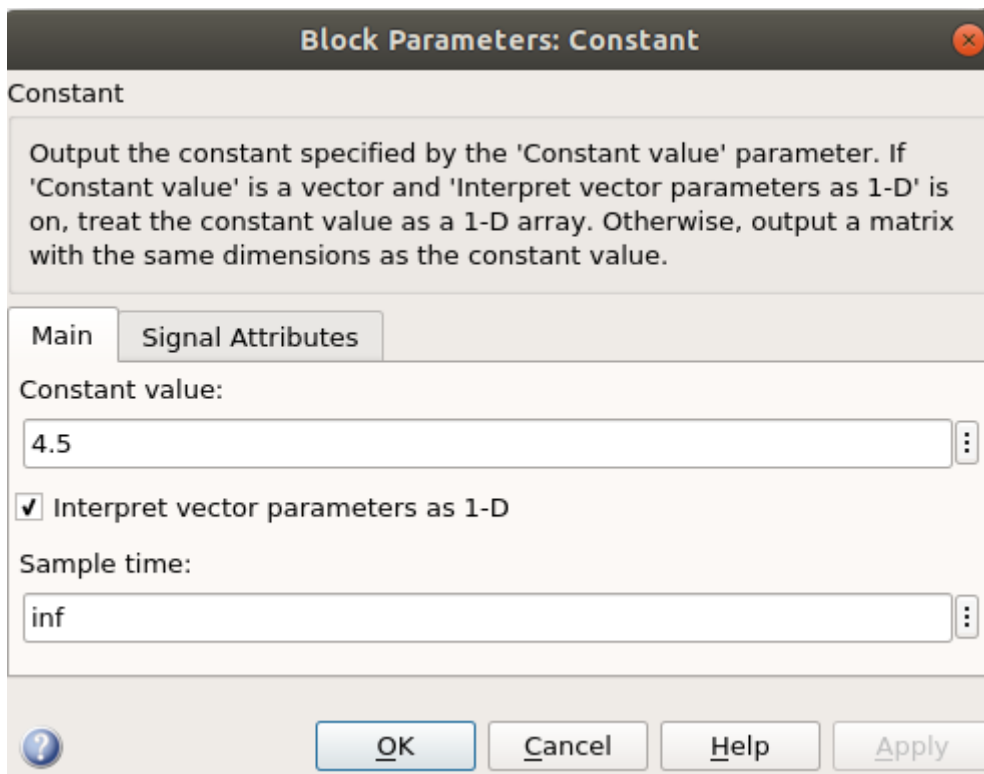
- Un vecteur contenant les instants de temps des mesures : $\text{time_IN}=[0.01 \ 0.02 \ 0.03 \dots 3.00]$
- Un vecteur contenant la tension appliquée aux bornes du moteur : $\text{Signal_IN}=[4.5 \ 4.5 \dots 4.5]$

Le modèle Matlab (« Signal_IN.slx ») qui crée ces vecteurs est présenté dans la figure ci-dessous :

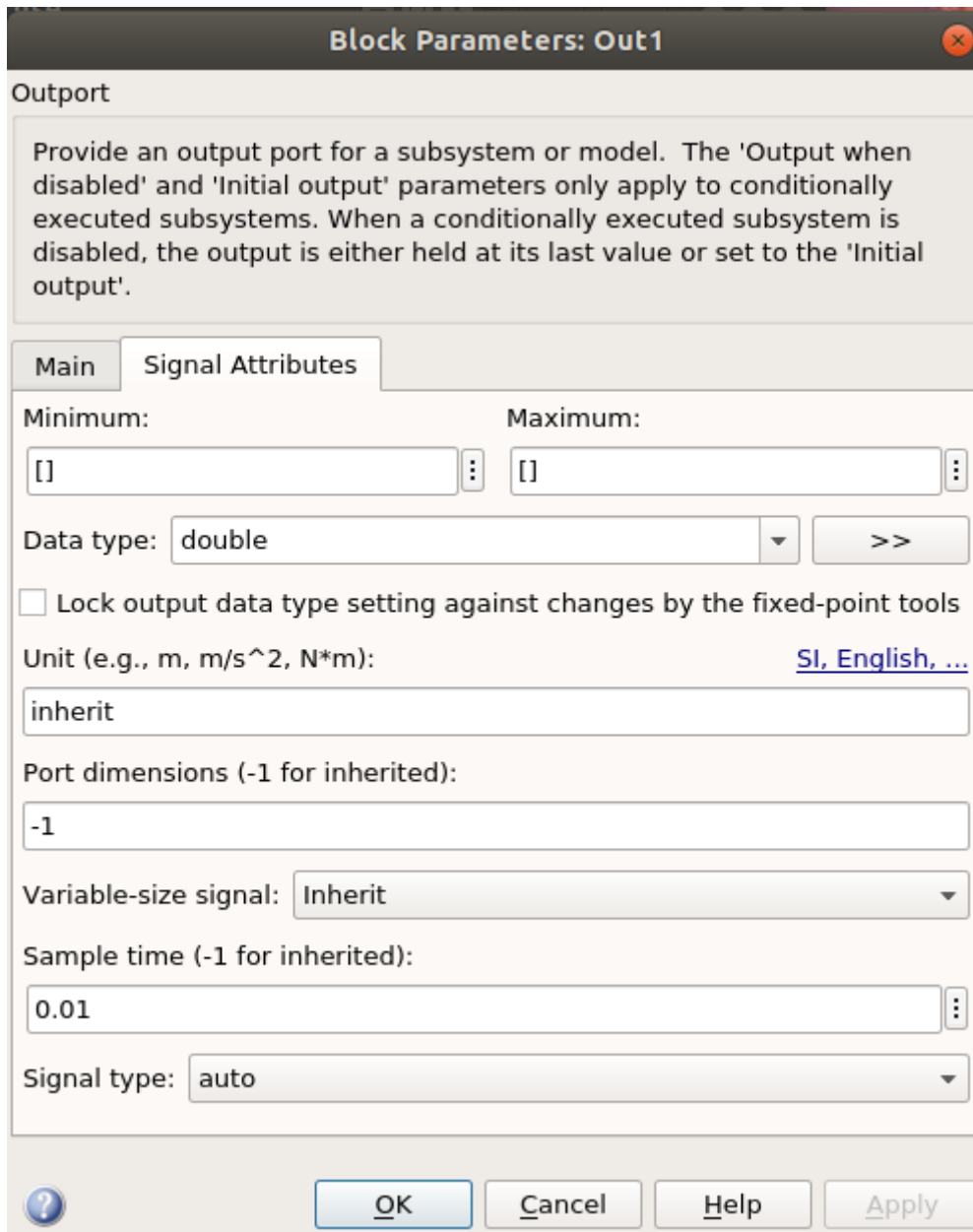


Nom du bloc et rôle	Bloc Matlab	Bibliothèque
« Constant » Fournit la valeur de la tension appliquée aux bornes du moteur et inscrite dans le vecteur yInSignal	 Constant	Simulink/Commonly Used Blocks
« Out1 » Crée dans le workspace tInSignal et yInSignal	 Out1	Simulink/Commonly Used Blocks

- Paramétrisation du bloc « Constant »

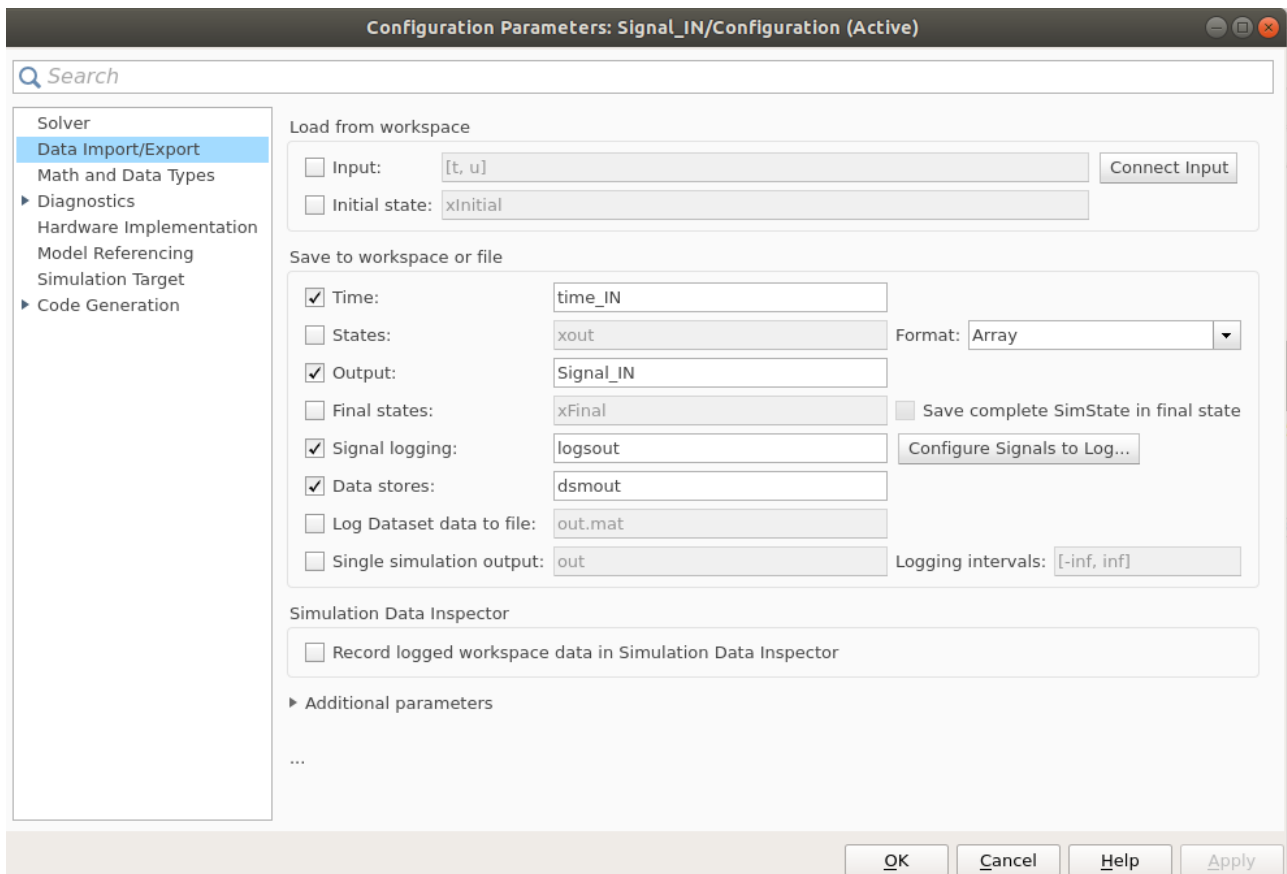


- Paramétrisation du bloc « Out1 »



Le pas de temps doit être égal à 0,01s car le pas d'échantillonnage des mesures est égal à 0,01s.

Les vecteurs time_IN et Signal_IN doivent être définis dans le « Model Configuration Parameters » :



Le temps de simulation doit être égale à 3s afin de correspondre aux mesures du fichier « mesures.mat ».

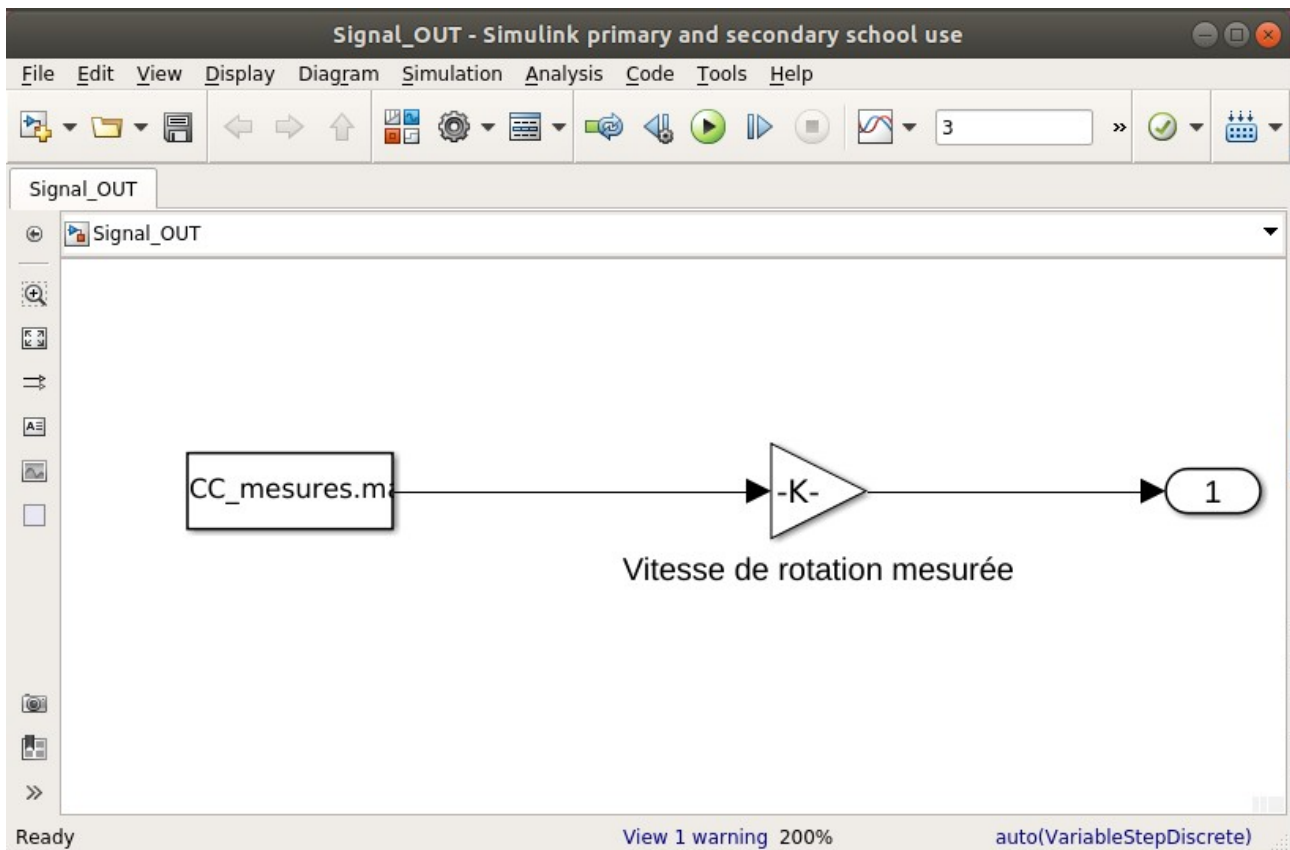
On lance ce modèle et dans le Workspace on doit obtenir :




Workspace	
Name	Value
Signal_IN	301x1 double
time_IN	301x1 double

La vitesse de rotation mesurée en fonction de temps est associée à deux vecteurs:

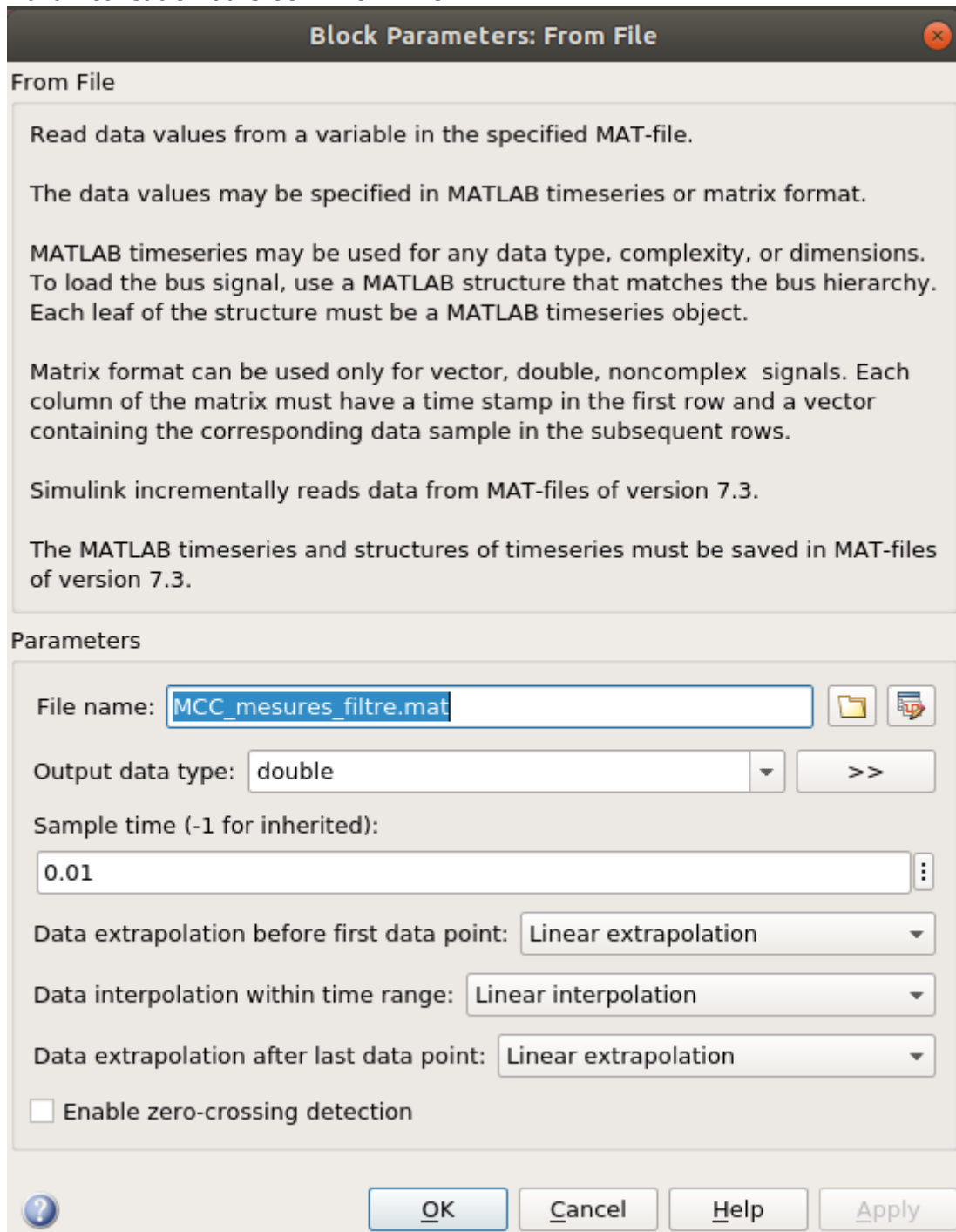
- Un vecteur contenant les instants de temps des mesures : $\text{time_OUT} = [0.01 \ 0.02 \ 0.03 \dots 3.00]$
- Un vecteur contenant la vitesse de rotation mesurée toutes les 10 ms: Signal_OUT .

Le modèle Matlab (« Signal_OUT.slx ») qui crée ces vecteurs est présenté dans la figure ci-dessous :

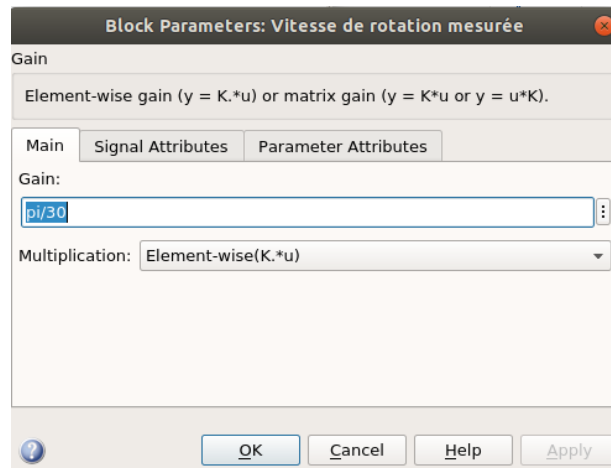


Nom du bloc et rôle	Bloc Matlab	Bibliothèque
« From File » Lit les valeurs de vitesse de rotation mesurées du fichier correspondant (« MCC_mesures.mat »)	 From File	Simulink/Sources
« Out1 » Crée dans le workspace time_OUT et Signal_OUT	 Out1	Simulink/Commonly Used Blocks
« Gain » Multiplie le signal par $\pi/30$ pour transformer les tours/min en rad/s	 Gain	Simulink/Commonly Used Blocks

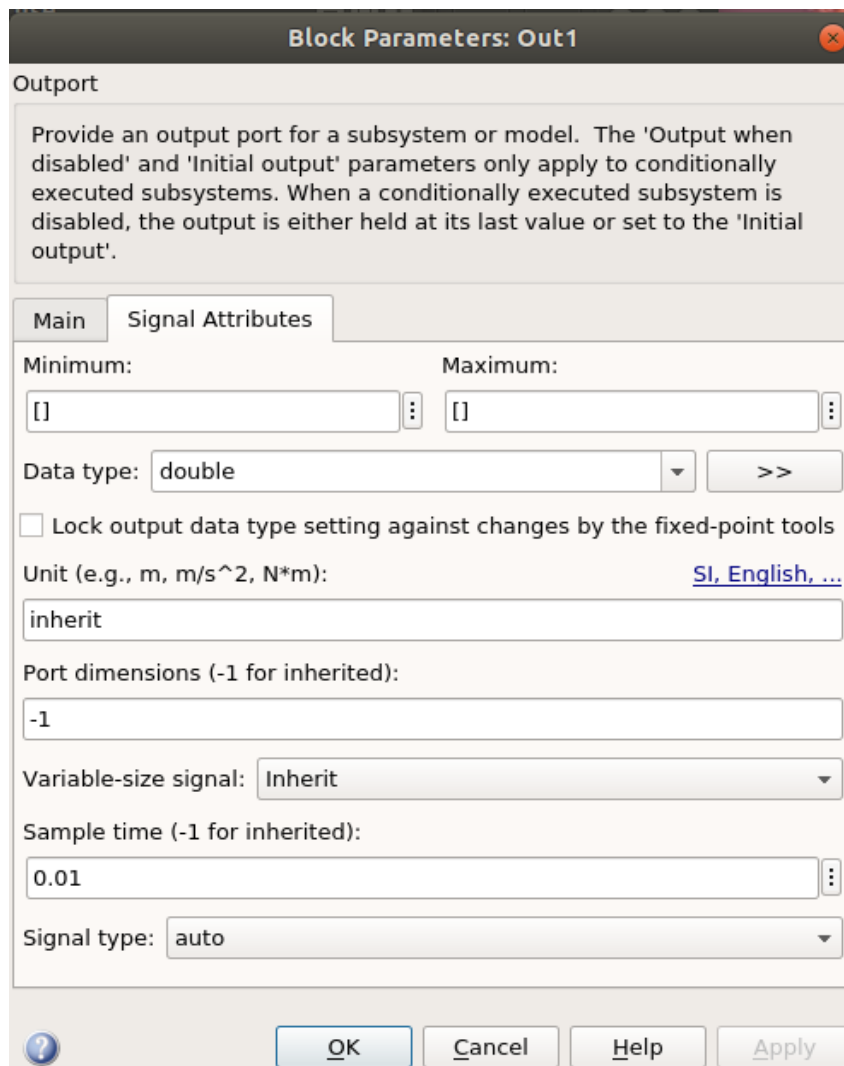
- **Paramétrisation du bloc « From File »**



- **Paramétrisation du bloc « Gain »**

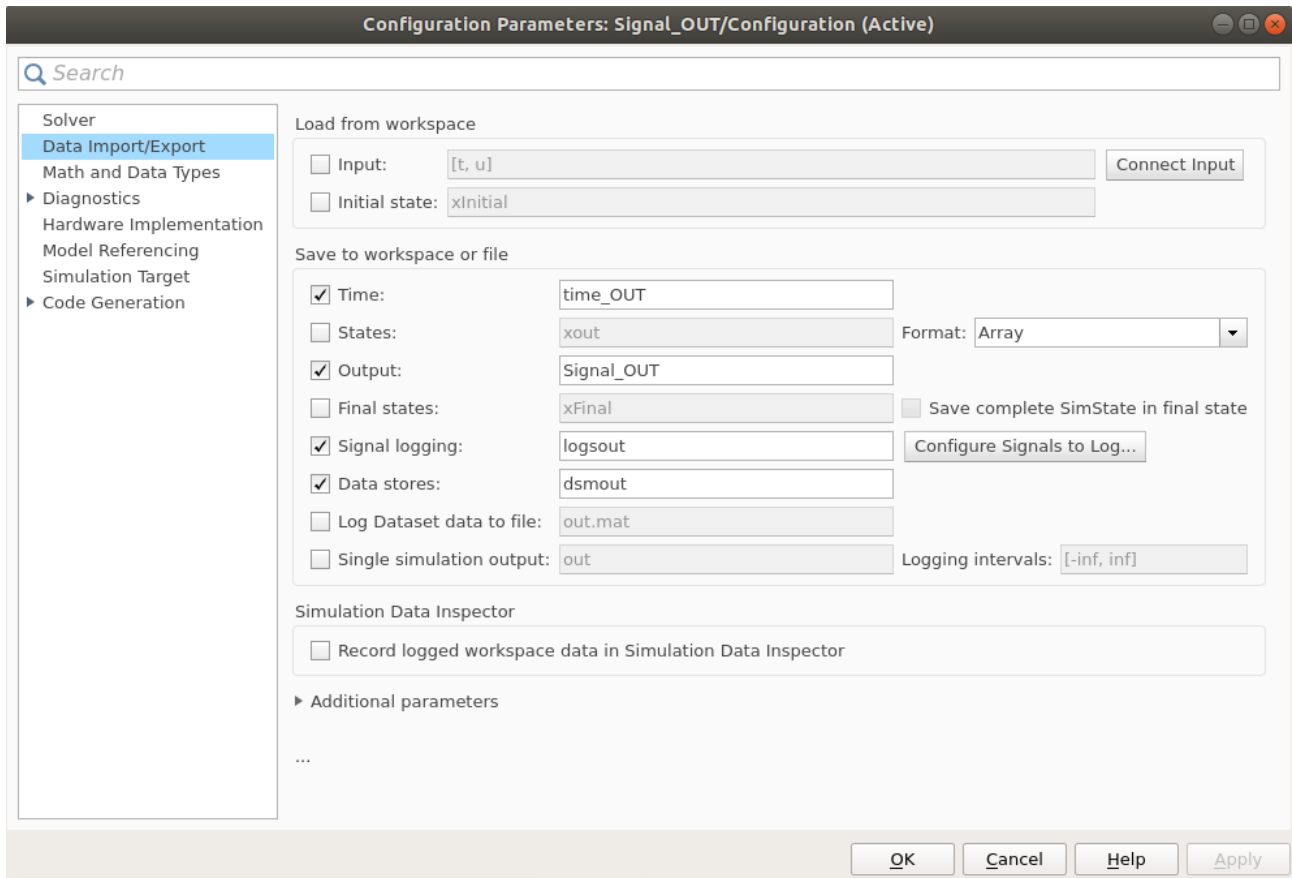


- **Paramétrisation du bloc « Out1 »**



Le pas de temps doit être égal à 0,01s car le pas d'échantillonnage des mesures est égal à 0,01s.

Les vecteurs time_OUT et Signal_OUT doivent être définis dans le « Model Configuration Parameters » :



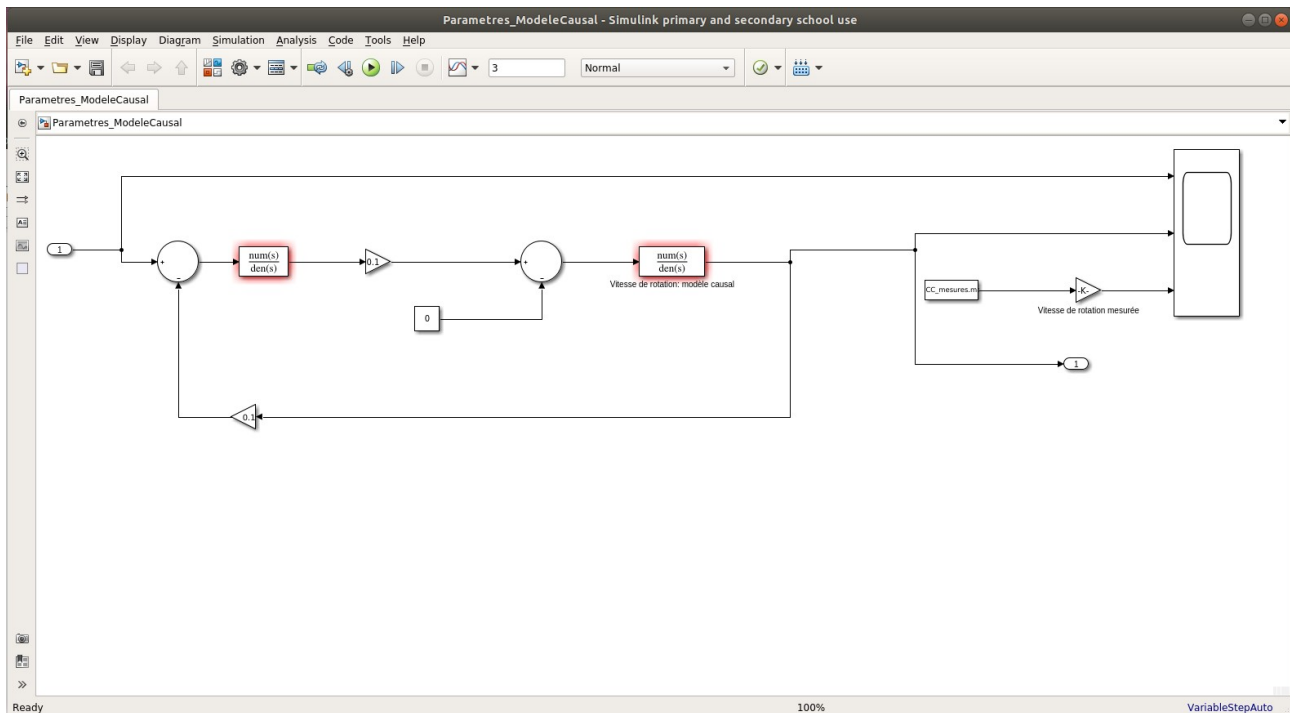
Le temps de simulation doit être égale à 3s afin de correspondre aux mesures du fichier « MCC_mesures_filtre.mat ».



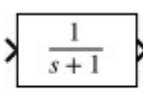
On lance ce modèle et dans le Workspace on doit obtenir :






Name	Value
Signal_OUT	1x1x301 double
time_OUT	301x1 double

2. Modèle causal du motoréducteur pour recherche des paramètres

Le modèle causal à utiliser est présenté dans la figure ci-dessous :



Nom et rôle du bloc	Bloc Simulink	Bibliothèque
« In1 » Signal d'entrée lu du Workspace	 In1	Simulink/Commonly Used Blocks
« Sum » Soustraction ou Addition	 Sum	Simulink/Commonly Used Blocks
« Transfer Fcn » Fonction de transfert	 Transfer Fcn	Simulink/Continuous

<p>« Gain » Multiplication par une constante</p>	 Gain	<p>Simulink/Commonly Used Blocks</p>
<p>« Scope » Oscilloscope</p>	 Scope	<p>Simulink/Commonly Used Blocks</p>
<p>« Constant » Constante pour la transformée de Laplace du couple résistant</p>	 Constant	<p>Simulink/Commonly Used Blocks</p>
<p>« From File » Le fichier qui contient les mesures</p>	 From File	<p>Simulink/Sources</p>
<p>« Out 1 » Crée la variable « Out1 » dans le Workspace</p>	 Out1	<p>Simulink/Commonly Used Blocks</p>

- **Paramètres du bloc « In1 »**

Les paramètres par défaut du bloc

- **Paramètres du bloc « Out1 »**

Les paramètres par défaut du bloc

- **Paramètres des blocs « Gain » et « Constant »**

Les paramètres du modèle causal du motoréducteur « Motoreducteur_ModeleCausal.slx »

- **Paramètres du bloc « From File »**

Block Parameters: From File

From File

Read data values from a variable in the specified MAT-file.

The data values may be specified in MATLAB timeseries or matrix format.



MATLAB timeseries may be used for any data type, complexity, or dimensions. To load the bus signal, use a MATLAB structure that matches the bus hierarchy. Each leaf of the structure must be a MATLAB timeseries object.



Matrix format can be used only for vector, double, noncomplex signals. Each column of the matrix must have a time stamp in the first row and a vector containing the corresponding data sample in the subsequent rows.


Simulink incrementally reads data from MAT-files of version 7.3.


The MATLAB timeseries and structures of timeseries must be saved in MAT-files of version 7.3.


Parameters


File name:  

Output data type:  


Sample time (-1 for inherited): 

Data extrapolation before first data point: 

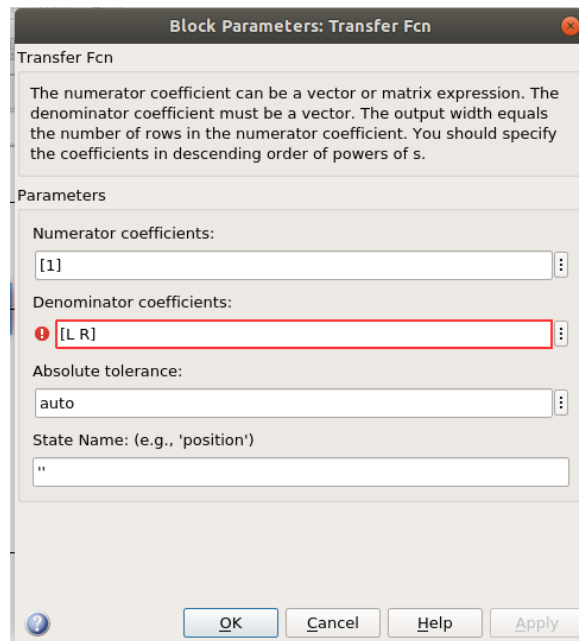
Data interpolation within time range: 

Data extrapolation after last data point: 

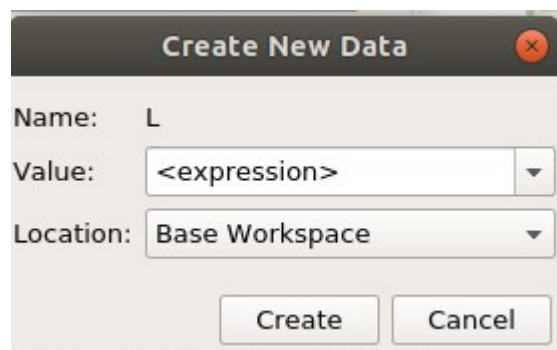
Enable zero-crossing detection



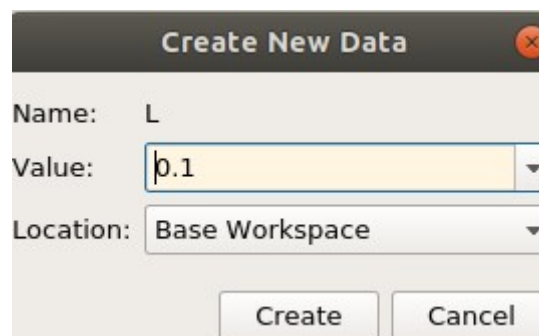
Les blocs « Transfert Function » et « Gain » sont encadrés en rouge, car leurs paramètres ne sont pas définis. Il faut commencer par donner des valeurs initiales :



- Dans « **Denominator coefficients** » cliquer sur les pointillées à droite, ensuite « L : Create ». On obtient :



Donner une valeur de départ en « Value » . Par exemple, 0,1 H .

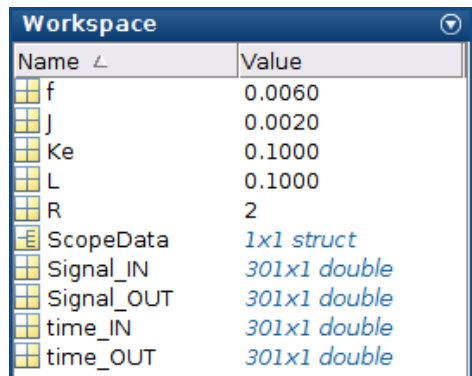


Ensuite « Create » .

On répète les mêmes étapes pour donner des valeurs aux autres paramètres :

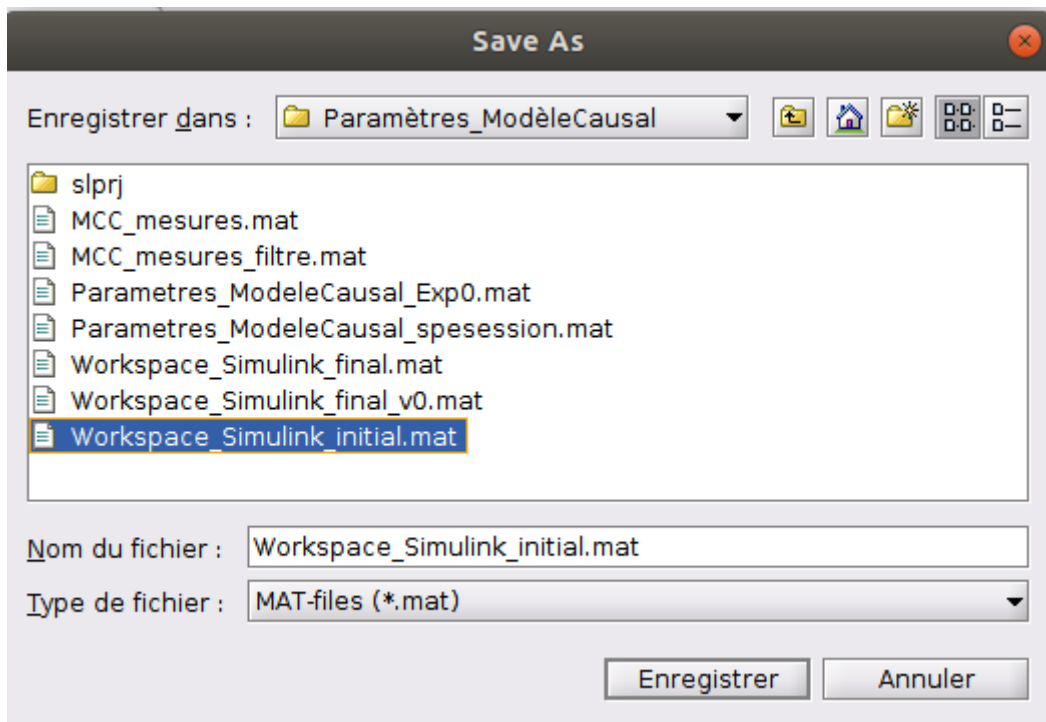
J= 0.002 kg·m²
R=2 Ω
Ke=0.1 Nm/A
f=0.006 N.m.s
L=0.1 H

Le Workspace devient :

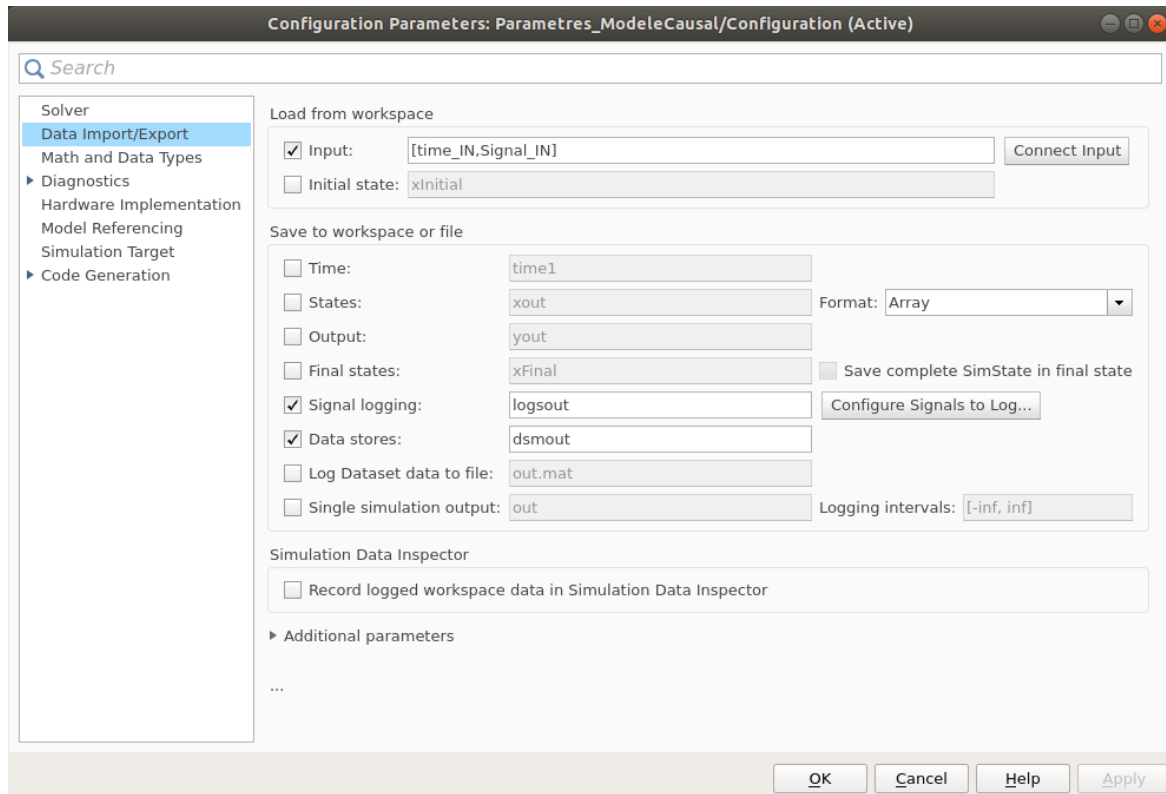


Name	Value
f	0.0060
J	0.0020
Ke	0.1000
L	0.1000
R	2
ScopeData	1x1 struct
Signal_IN	301x1 double
Signal_OUT	301x1 double
time_IN	301x1 double
time_OUT	301x1 double

On peut sauvegarder ce workspace dans un fichier pour des simulations ultérieures. On peut ainsi sauter les étapes de création des signaux time_IN, Signal_IN, time_OUT, Signal_OUT ainsi que l'étape de déclaration des paramètres des fonctions de transfert (L, R, f et J) :



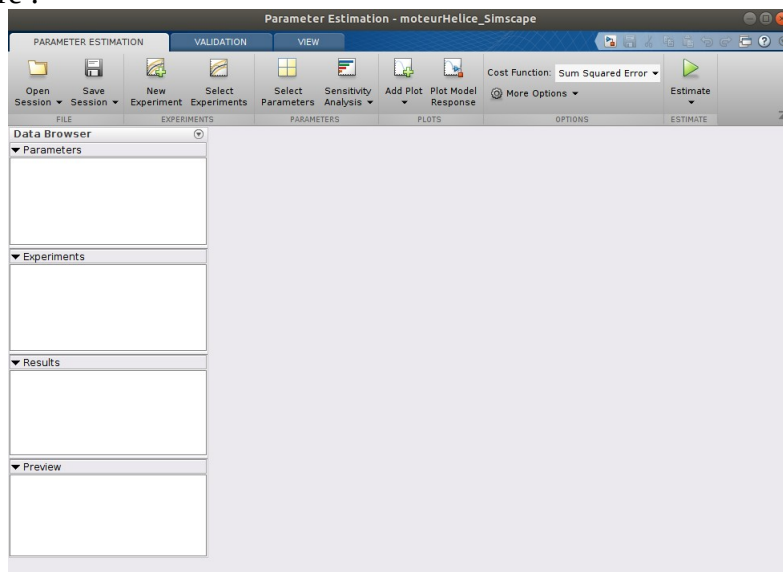
Les paramètres du modèle sont :



3.Recherche des paramètres

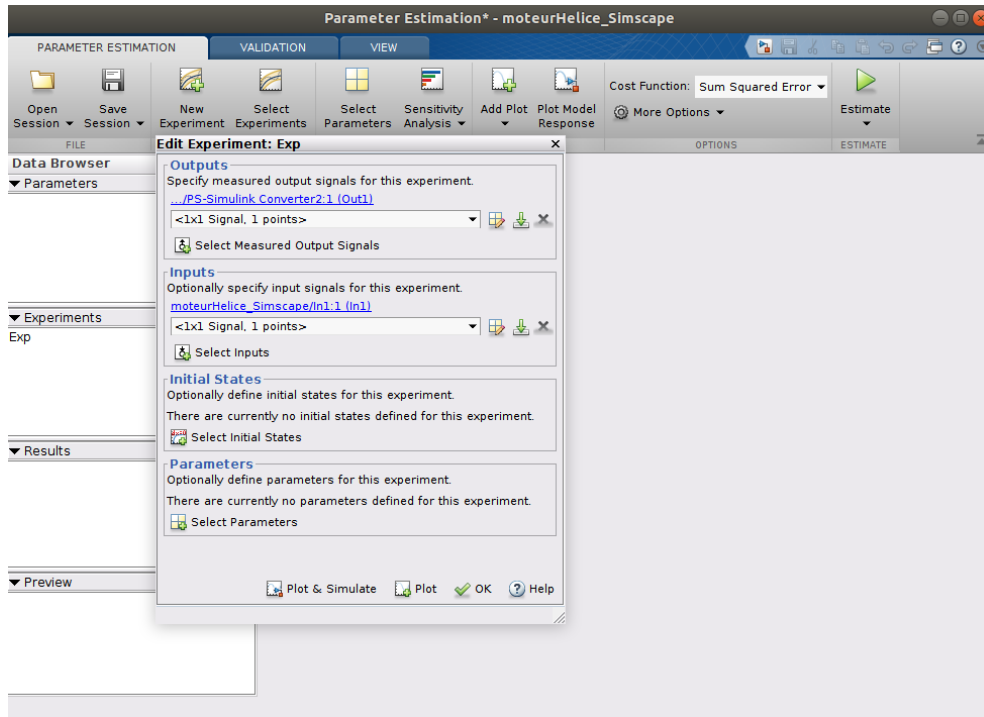
Pour lancer l'algorithme censé à déterminer L,R,J et f on sélectionne :
Analysis/Parameter Estimation/

On obtient la fenêtre :



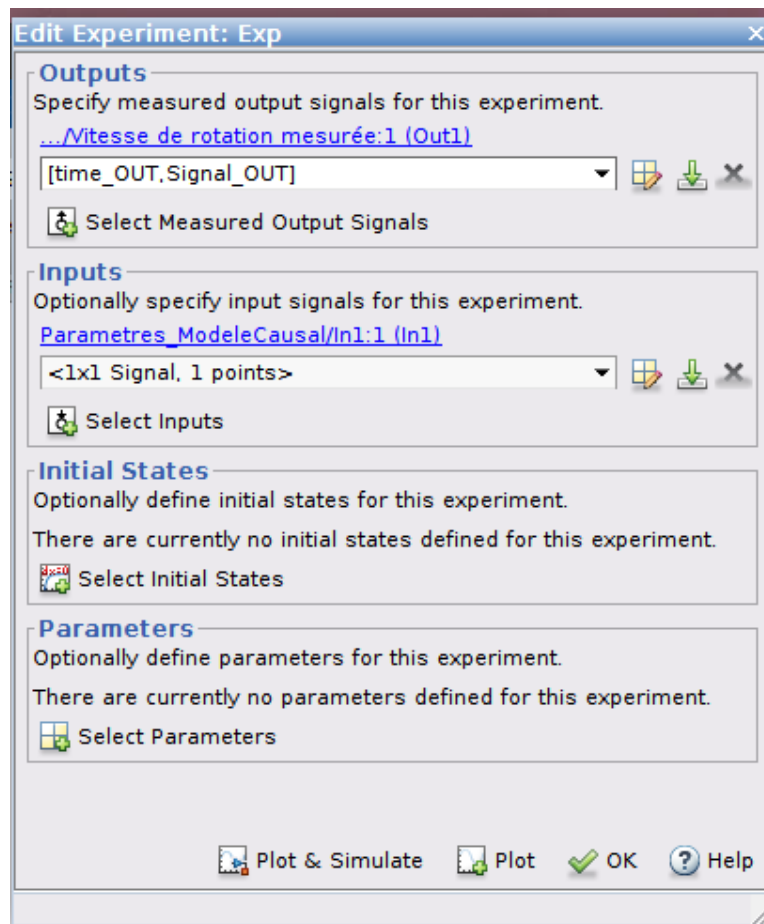
Dans cette fenêtre il faudra configurer les blocs : « New Experiment », « Select Parameters » et « Add Plot ».

Un clic sur « New Experiment » ouvre la fenêtre :



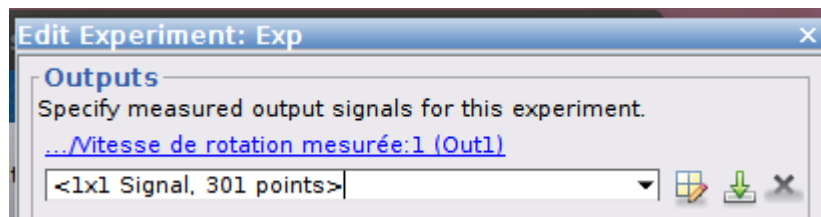
Dans cette fenêtre on spécifiera les champs « Outputs » et « Inputs ». « Inputs » représente le signal d'entrée [time_IN, Signal_IN], « Outputs » le signal de sortie, c'est-à-dire, le signal des mesures [time_OUT, Signal_OUT].

Pour la partie « Outputs » on introduit [time_OUT, Signal_OUT] :

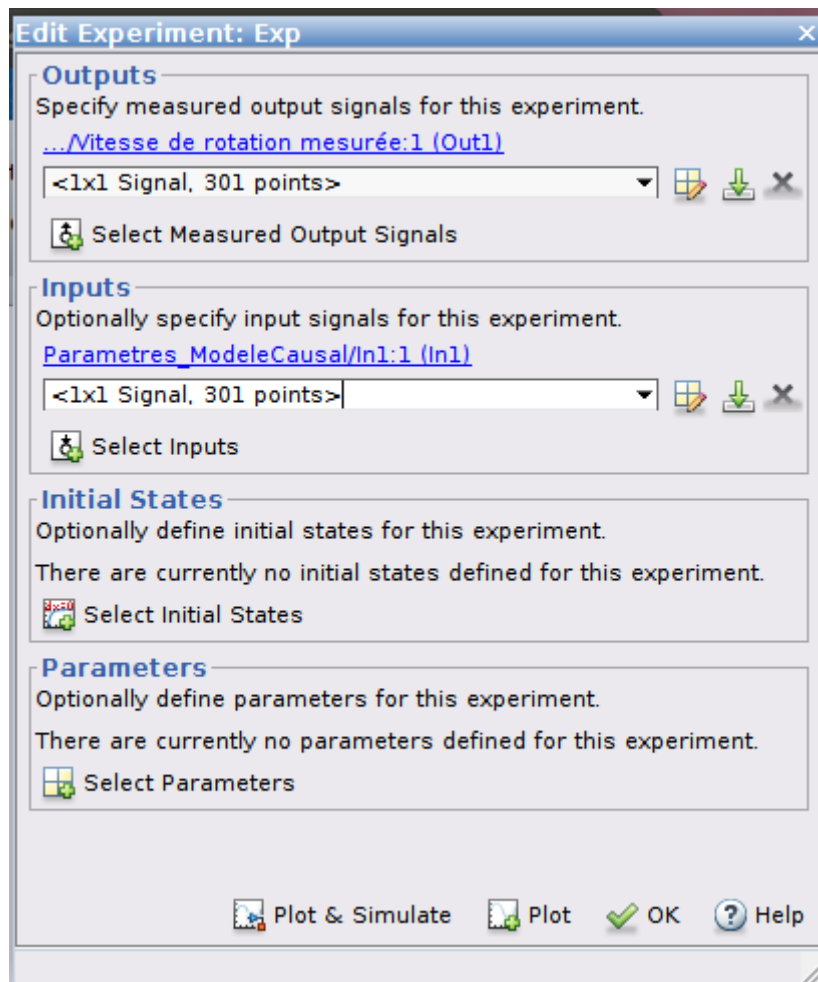


Ensuite « Enter » :

et dans la fenêtre, à la place de [time_OUT,Signal_OUT] on obtient :

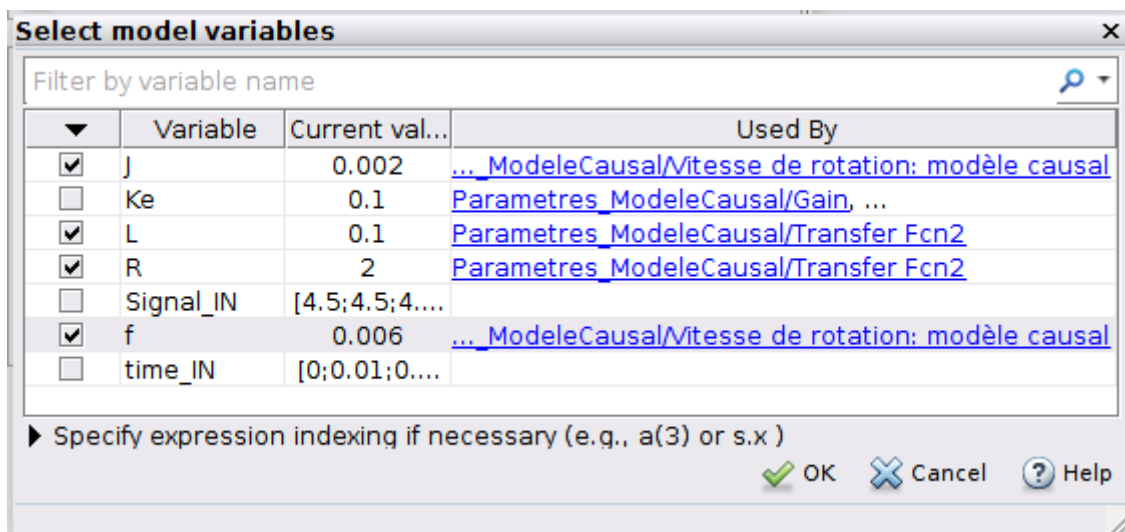


Une fois les deux signaux introduits on aura :

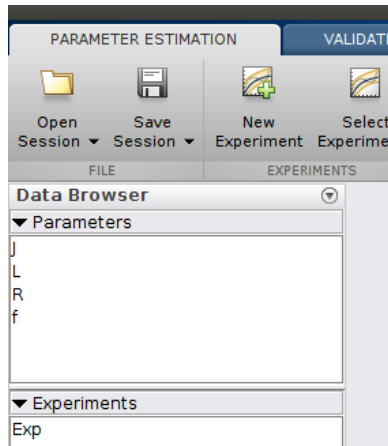


On peut confirmer les signaux : « OK ».

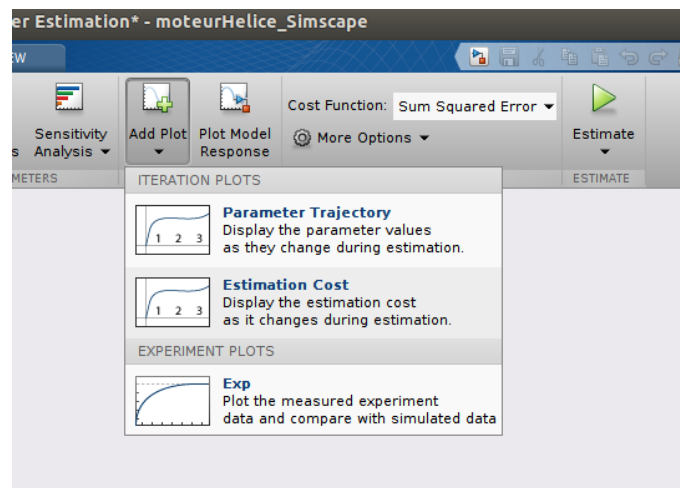
Dans le bloc « Select Parameters » on sélectionne L,R, J et f :



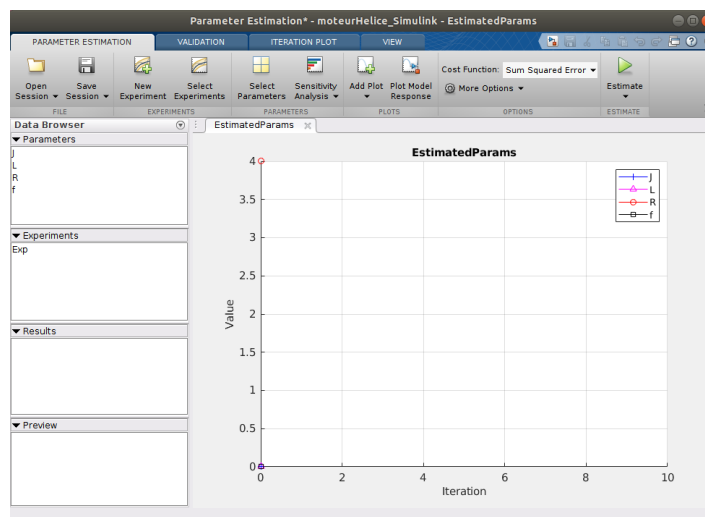
Ces paramètres vont apparaître dans l'onglet « Parameters » :



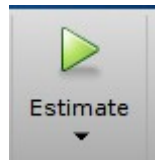
Dans la fenêtre « Add Plot » on sélectionne « Parameter Trajectory » :



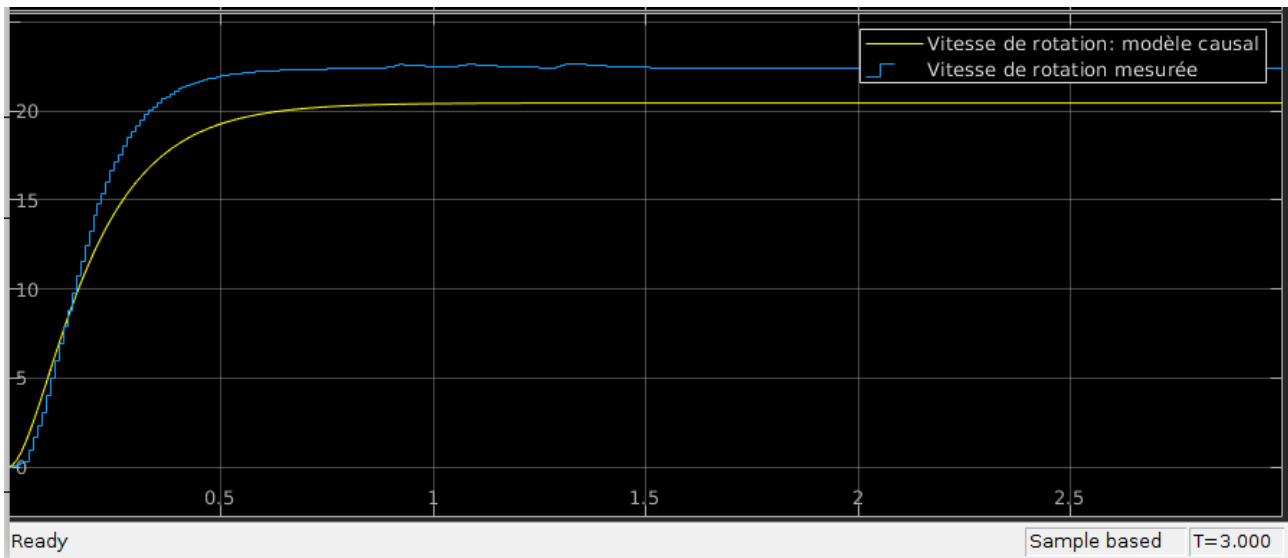
On obtient :



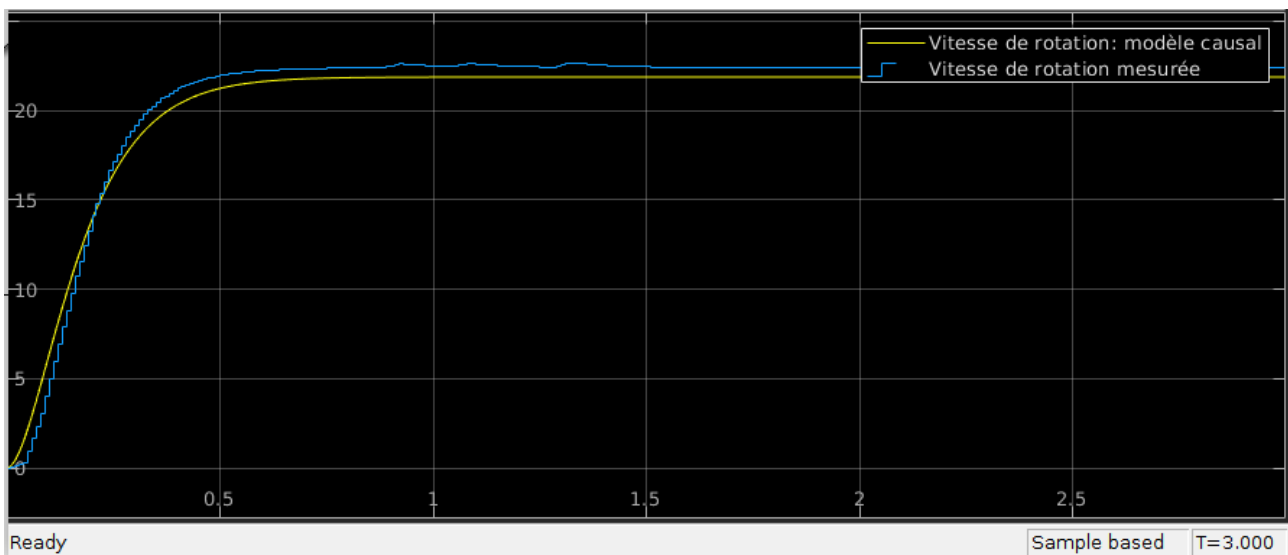
On peut maintenant lancer la simulation avec « Estimate » :



Le modèle sera lancé plusieurs fois avec des valeurs L,S et J différentes. Ci-dessous, les réponses indicielles initiale et finale (à la fin de l'algorithme) :



Réponses indicielle initiale et mesurée.



Réponses indicielle finale (après convergence de l'algorithme) et mesurée.