

Modélisation Multiphysique avec Simscape

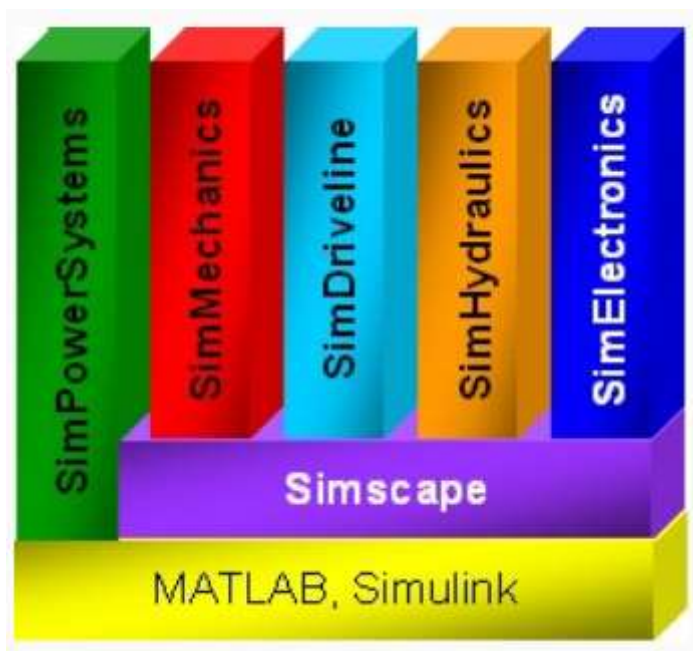
Mécanisme entraîné par un moteur à courant continu

I. But

La modélisation d'un moteur à courant continu (MCC) est un bon exemple pour démarrer dans la simulation multiphysique. En effet, un MCC est un convertisseur électromécanique avec un domaine électrique d'induit et un domaine mécanique dans lequel on peut insérer un moment d'inertie, un couple de frottement visqueux, une charge résistante, etc. La simulation peut alors nous donner la vitesse de rotation, la consommation de courant, la puissance utile... Dans ce tutoriel, nous allons apprendre à modéliser un petit mécanisme entraîné par un moteur à courant continu et visualiser les conséquences d'un effort résistant sur l'entraînement du moteur.

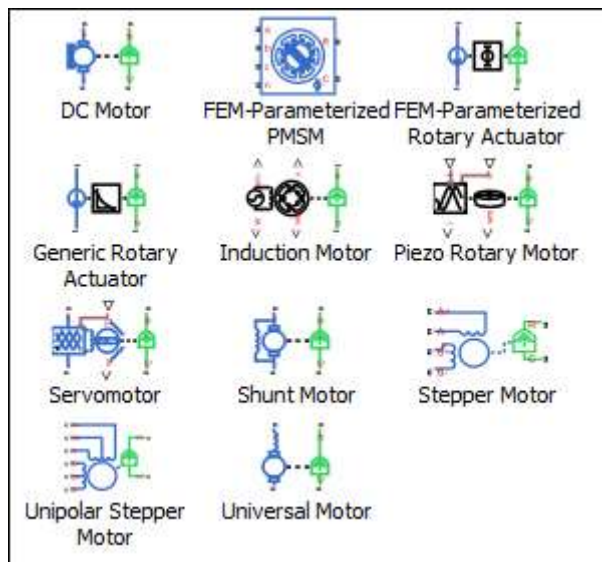
II. Environnement de Simscape

Simscape est une extension de Matlab/Simulink dédiée à la modélisation multiphysique avec des modules spécialisés dans les domaines mécaniques, électriques, hydrauliques... SimPowerSystems, une autre extension, est dédié à la modélisation et à la simulation de circuits électriques de puissance.

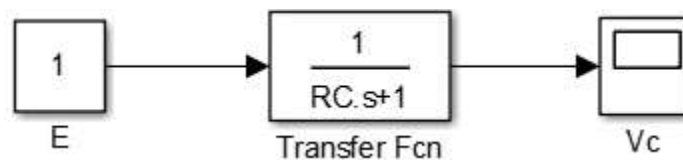


Chaque module comporte une palette de composants graphiques de base (une résistance électrique, un vérin, une pompe, un réducteur à engrenages, etc.) à faire glisser dans la fenêtre de travail et à relier entre eux. Chaque composant constitue un modèle de comportement avec des équations mathématiques faisant intervenir les grandeurs physiques. Les connexions entre composants établissent les relations de flux et d'efforts.

Exemple de composants (bibliothèque : « Rotational Actuators ») :

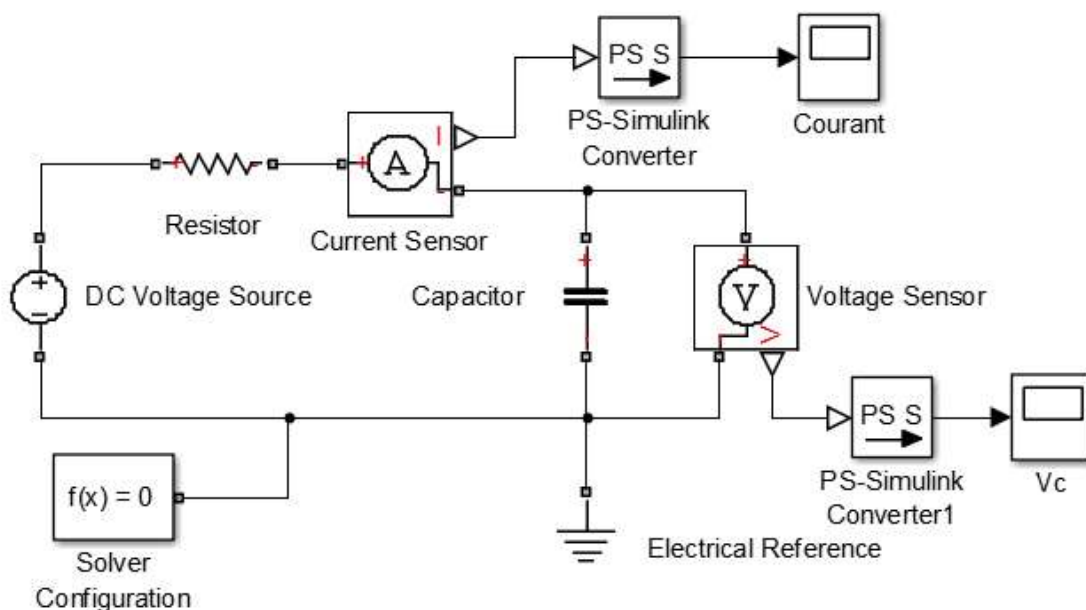


Simulink permet de créer des modèles « causaux ». La sortie « d'une boîte » est « causée » par l'entrée. En aucun cas la sortie n'influe sur l'entrée. En d'autres termes, le flux est unidirectionnel.



Ci-dessus, un modèle causal d'un circuit RC : il n'y a qu'une sortie « Vc », la tension aux bornes du condensateur.

Simscape permet de créer des modèles acausaux. Les liens entre les diverses « boîtes » sont des liens de puissance ($U \times I$ ou $F \times V$, par exemple). Il n'y a pas de sens imposé pour les flux, ceci permet, par exemple, que le couple résistant imposé sur un arbre moteur influe sur le courant du moteur.



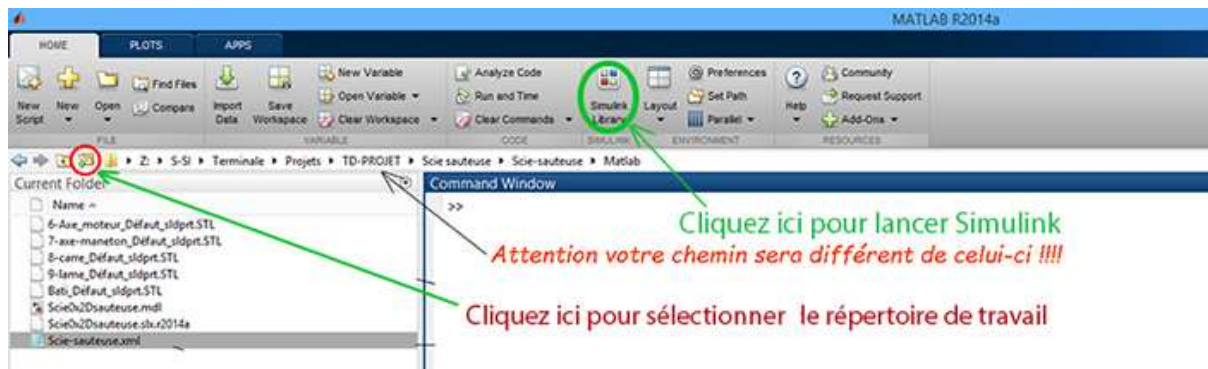
Le même circuit avec Simscape permet un placement de chaque élément tel qu'il le serait dans la réalité (comme le bloc « Current Sensor », un ampèremètre monté en série). Ce schéma permet d'obtenir la tension Vc et le courant I. Le bloc « PS-S Simulink Converter » fait la passerelle avec la

couche Simulink en convertissant un signal physique d'un diagramme physique acausal en un signal causal vers un bloc Simulink (ici un « Scope » pour tracer la courbe du signal en fonction du temps).

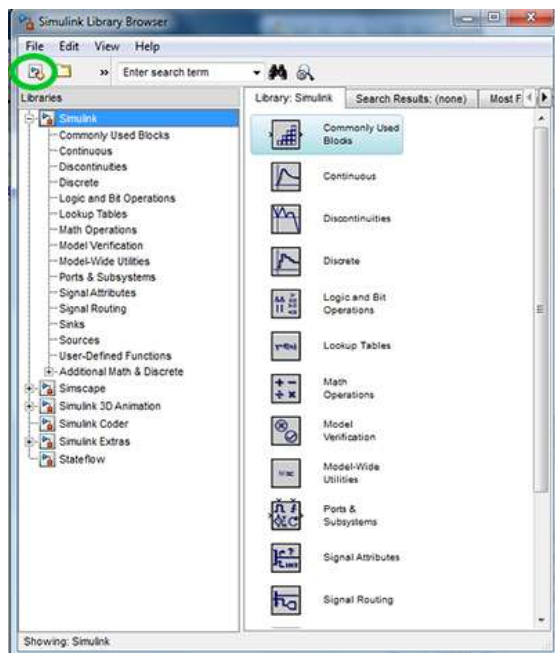
III. Création du projet▲

Lancez Matlab.

La fenêtre ci-dessous va s'ouvrir, vous devez dans un premier temps vous positionner dans votre dossier de travail, puis lancer Simulink.



Pour créer un nouveau modèle, cliquez sur l'icône entourée en vert.

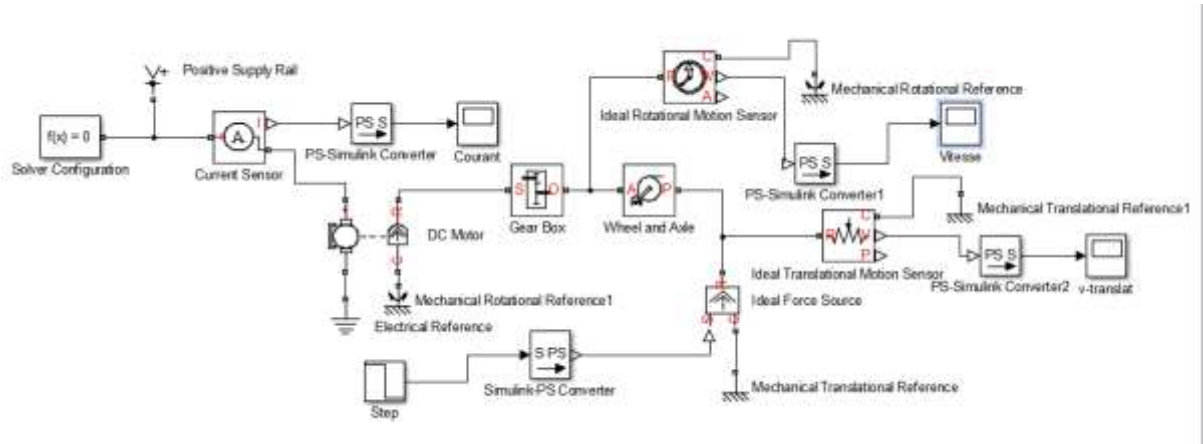


La fenêtre de gauche contient l'arborescence des divers modules que vous avez achetés, celle de droite vous montre les composants disponibles dans chaque bibliothèque. Pour placer un composant sur le modèle, il suffit de cliquer dessus et de le déplacer dans la fenêtre du modèle que vous avez précédemment créé (cercle vert de l'image ci-dessus).

IV. Le modèle▲

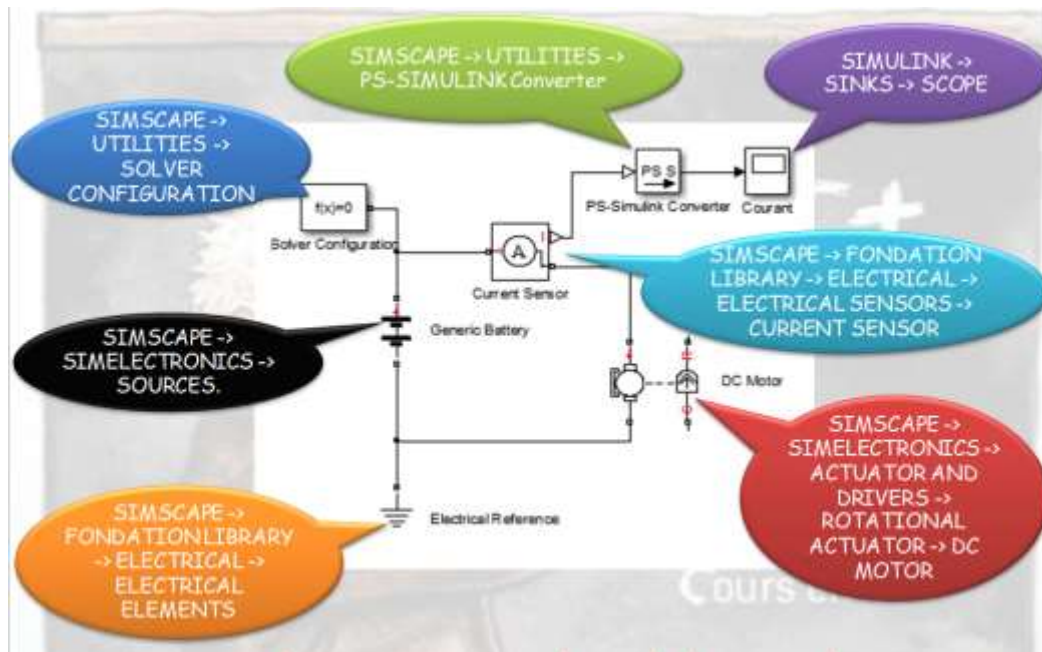
Le modèle que nous allons réaliser, est composé :

1. D'un moteur CC ;
2. D'un réducteur ;
3. D'une transformation de mouvement (rotation en translation).



V. Domaine électrique▲

Nous allons dans un premier temps créer la partie du domaine électrique.



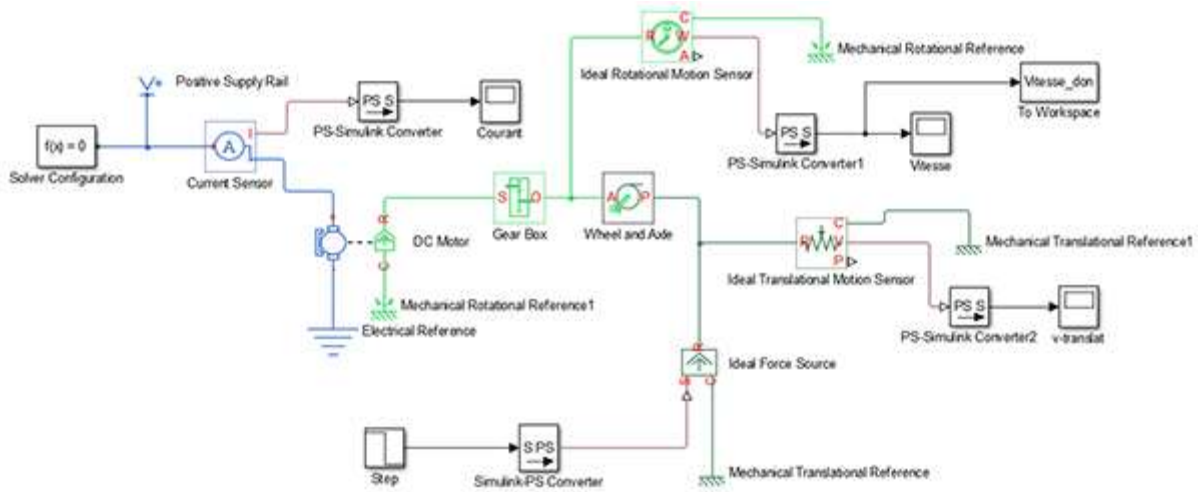
Sur le schéma ci-dessus, la source de tension (« supply rail ») a été remplacée par une batterie. Cette deuxième solution permet la simulation de la décharge de la batterie.

Vous trouverez sur l'image ci-dessus, la position de chaque élément dans les diverses bibliothèques.

Il existe une autre solution qui peut être dans certains cas plus rapide : Tapez le nom du bloc dans la barre de recherche, puis cliquez sur les jumelles.

Placez les composants sur le modèle et effectuez les liaisons.

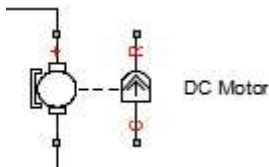
Remarque : à partir de la version 14b, Matlab différencie les flux physiques en leur affectant une couleur différente. À chaque type de flux physique est associé une couleur :



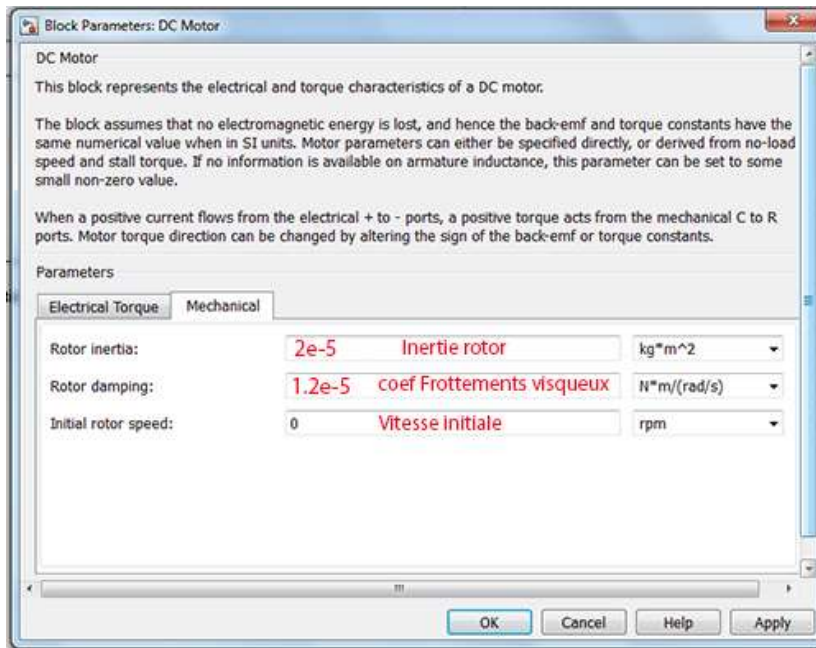
La rédaction du document ayant été réalisée à l'aide de la version 2012, les couleurs n'apparaîtront pas sur les captures d'écran.

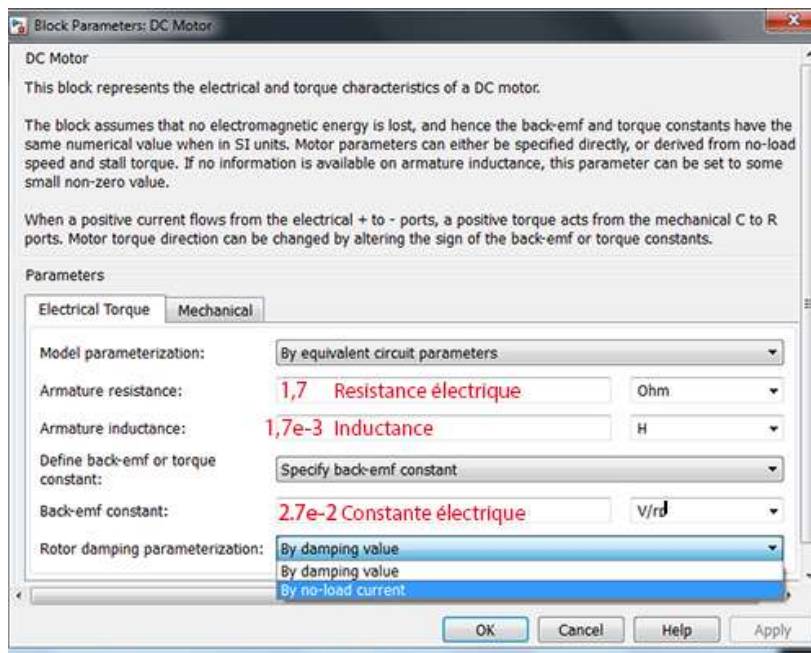
V-A. Configuration

V-A-1. Le moteur



Pour accéder aux paramètres, double-cliquez sur le modèle :



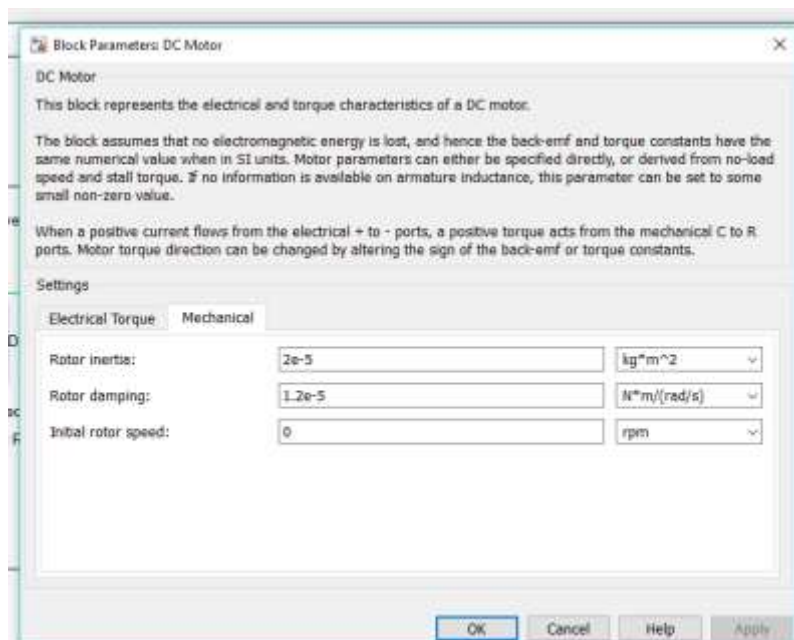


Vous pouvez choisir les paramètres à renseigner en fonction des données que vous possédez sur votre moteur.

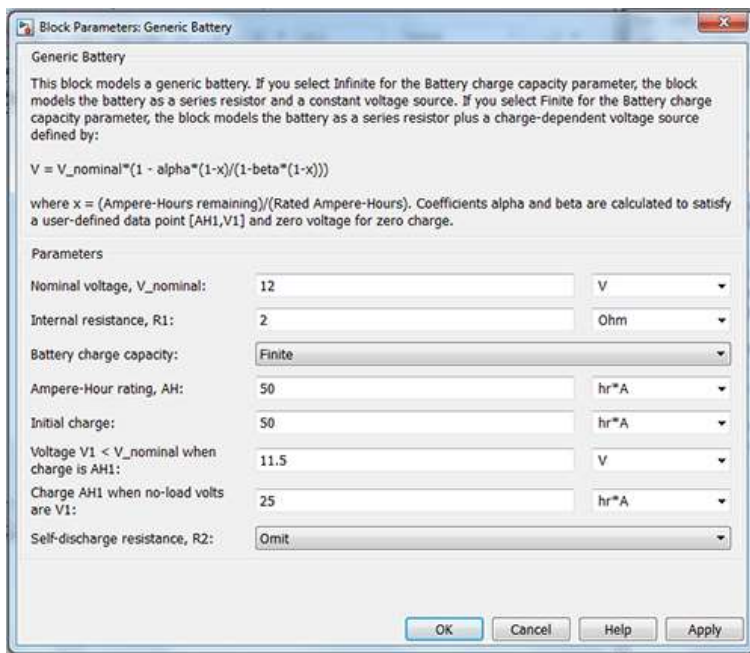
Nous choisirons :

1. « By equivalent circuit parameters » ;
2. « By damping value ».

Dans l'onglet Mechanical :



V-A-2. La batterie



En sélectionnant « Finite », vous pouvez simuler la décharge de la batterie. Si cette simulation ne vous intéresse pas, sélectionnez : « Infinite ».

Il est possible de remplacer la batterie par une source de tension.

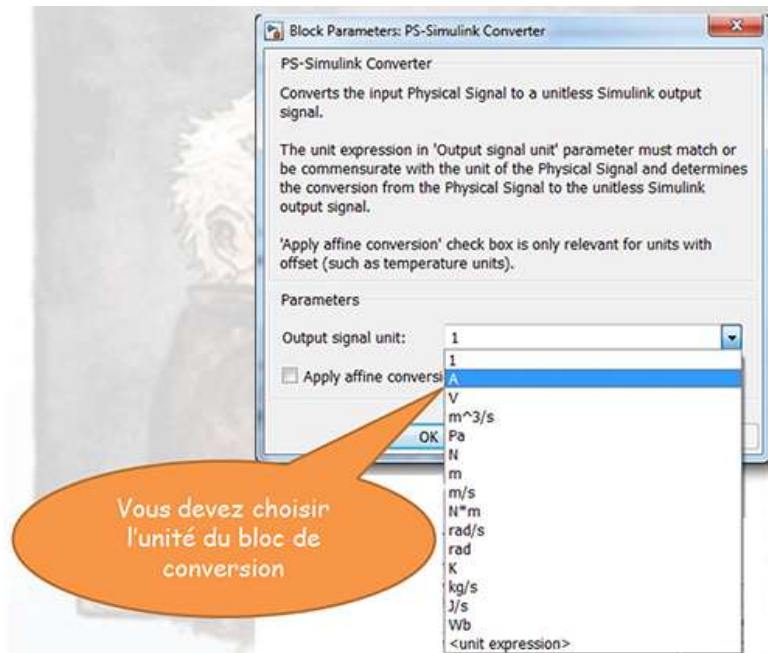
V-A-3. Passerelle signaux physiques ⇒ valeurs numériques

Comme vous l'avez déjà compris, plusieurs natures de signaux cohabitent dans Simscape :

1. Des signaux physiques (électrique, pneumatique, mécanique) ;
2. Des signaux numérique (Simulink).



Les valeurs numériques sont manipulées par Simulink, il est nécessaire d'utiliser une interface entre ces deux flux.



Certaines unités existent alors qu'elles ne sont pas listées, comme :

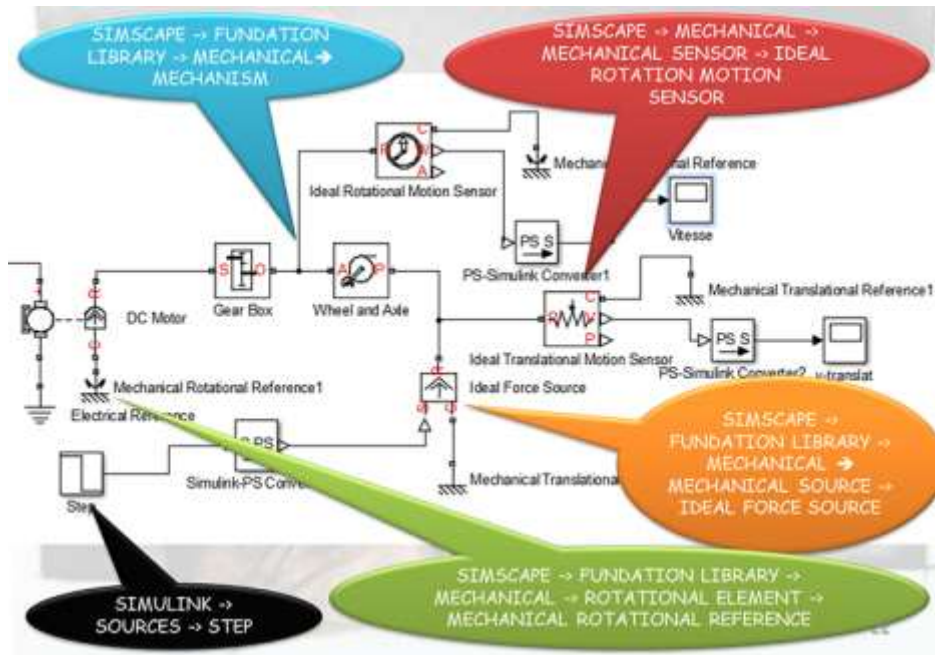
1. deg ==> pour degrés ;
2. rpm ==> tour par minute.

VI. Domaine mécanique▲

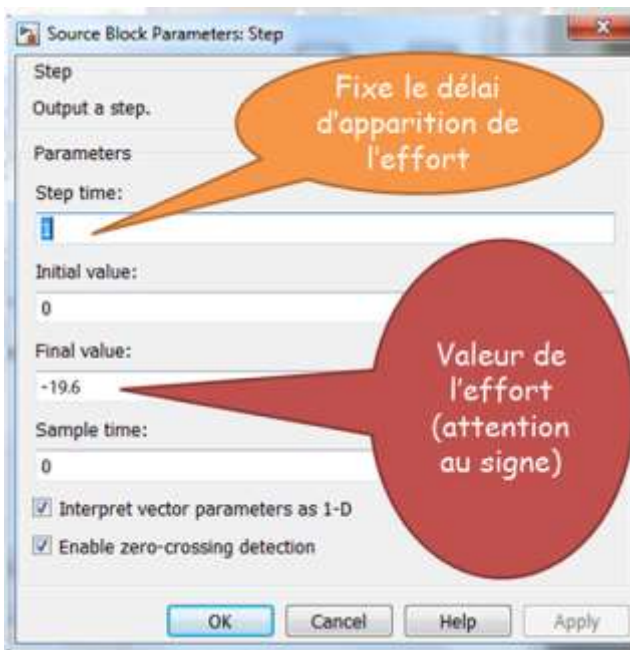
Vous allez maintenant ajouter les composants mécaniques. Les liens sont donnés sur le dessin ci-dessous.

Sur ce schéma, nous trouvons après le moteur :

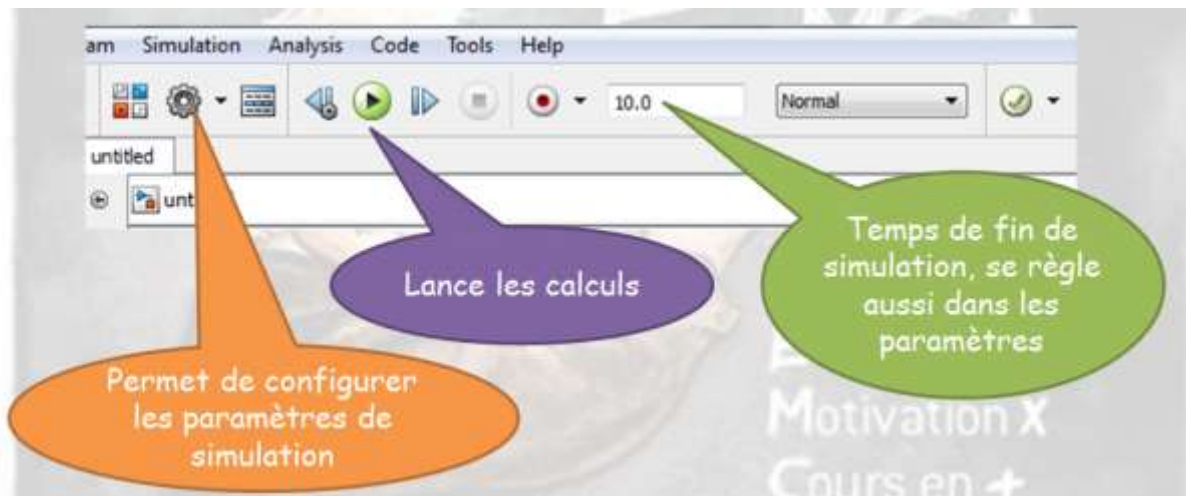
1. Un réducteur (« gear box ») : il se paramètre en donnant la valeur du ratio, pour une réduction 1/50, entrez « 50 » ;
2. Un transformateur de mouvement (« wheel and axle ») : il se paramètre en donnant la valeur du rayon, par exemple 0,03 m ;
3. Un capteur de vitesse (« ideal translational motion sensor ») :
 1. Vitesse = V,
 2. Position = P ;
4. un capteur de rotation (« ideal rotational motion sensor ») :
 1. Vitesse = W,
 2. Angle = A ;
5. Un générateur d'efforts (« ideal force source ») ;
6. Un « step » qui permet de générer un front avec un décalage temporel.



VI-A. détail du « step » ▲



VII. Configuration du solver ▲



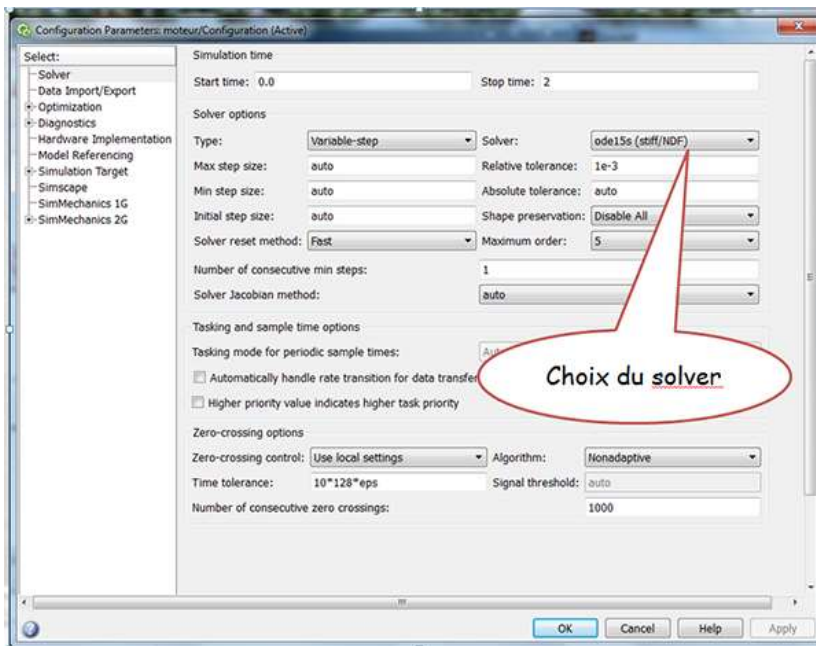
Avant de lancer votre simulation, vous devez contrôler la configuration du « solver ».

Le bloc du solver a été ajouté dans la première partie. Ce bloc est indispensable si on utilise une autre bibliothèque que Simulink.

Le choix du solver est lié au type de simulation que vous effectuez et vous pouvez utiliser pour cette simulation :

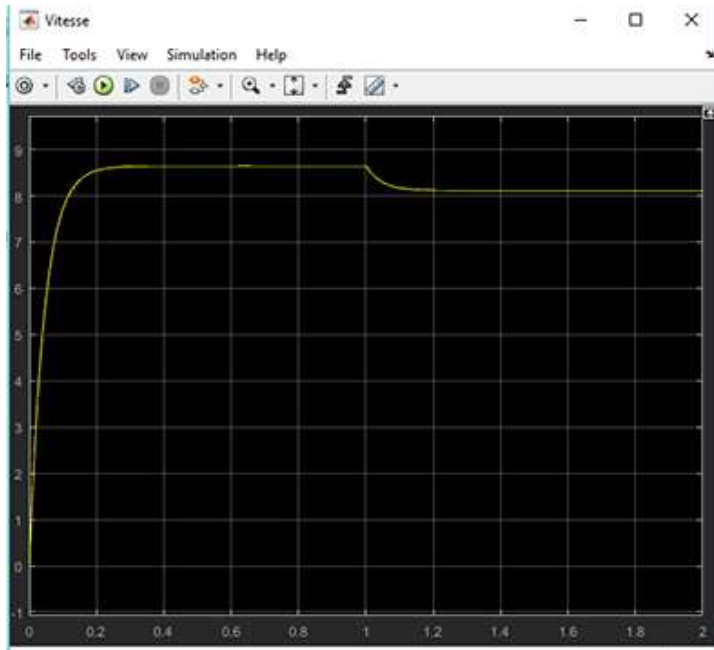
1. ode23t ;
2. ode15s.

Cliquez sur la petite roue dentée, modifiez le solver, (la version 2015 propose de choisir automatiquement le solver) :



VIII. Résultats▲

Vous pouvez maintenant lancer la simulation en fixant un temps de fin à 2 s, vous obtiendrez une courbe similaire à l'image ci-dessous :



La pente de la montée en vitesse (régime transitoire), est liée aux valeurs :

1. Inductance ;
2. Inertie ;
3. Coefficient d'amortissement.

La réduction de vitesse à 1 seconde, est causée par la charge de 19,6 N. (step).

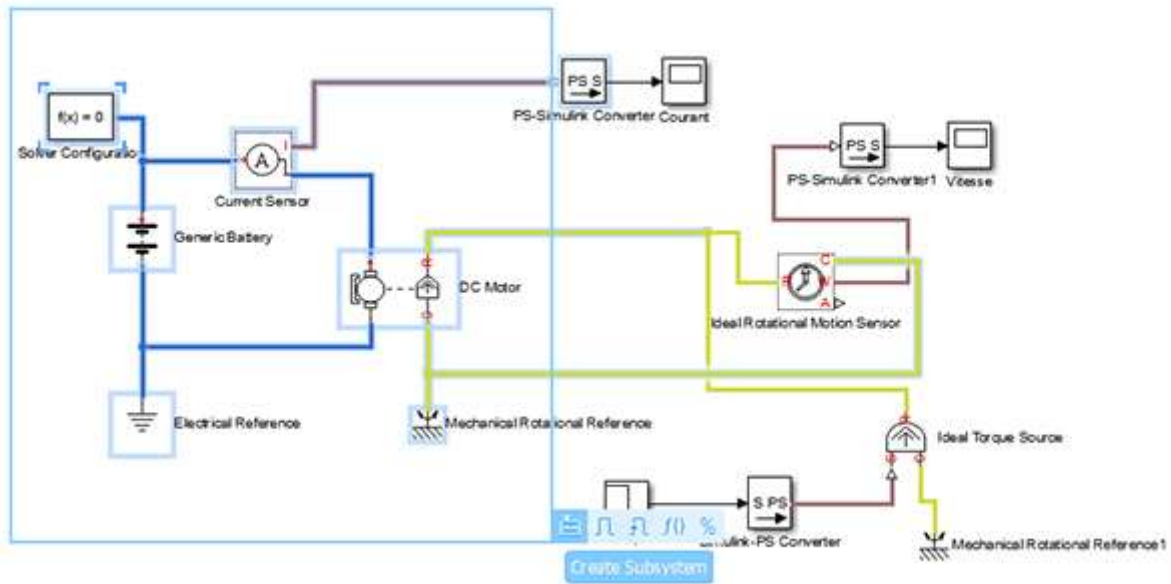
IX. Les sous-systèmes▲

Dans le but de rendre plus lisible le modèle, il peut être judicieux de regrouper par fonction un ensemble de composants.

Bien que dans ce cas particulier, au vu de la faible complexité du modèle, cela a peu d'intérêt, nous allons créer un sous-système afin de connaître la méthodologie.

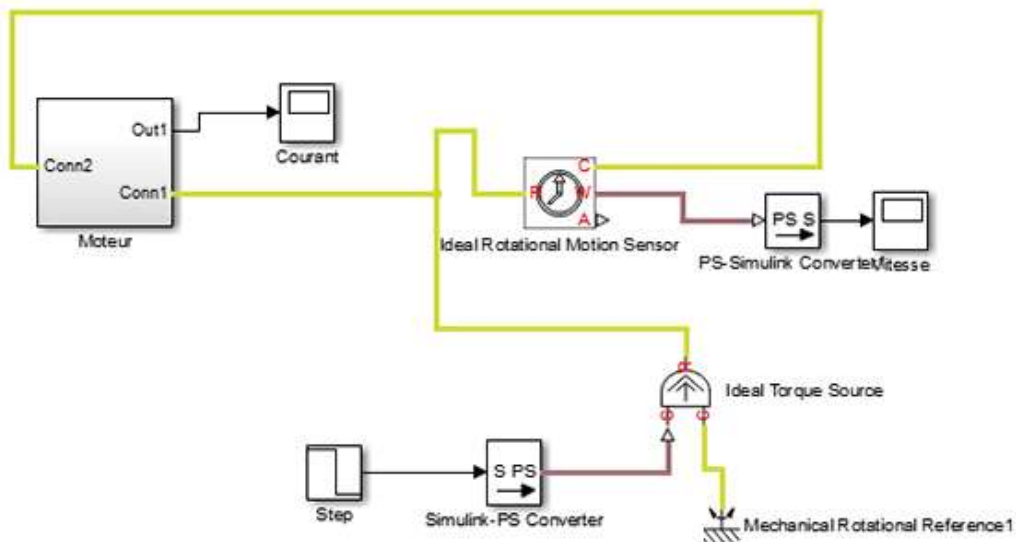
IX-A. Sélection des blocs ▲

À l'aide de la souris, sélectionnez les blocs que vous désirez regrouper :

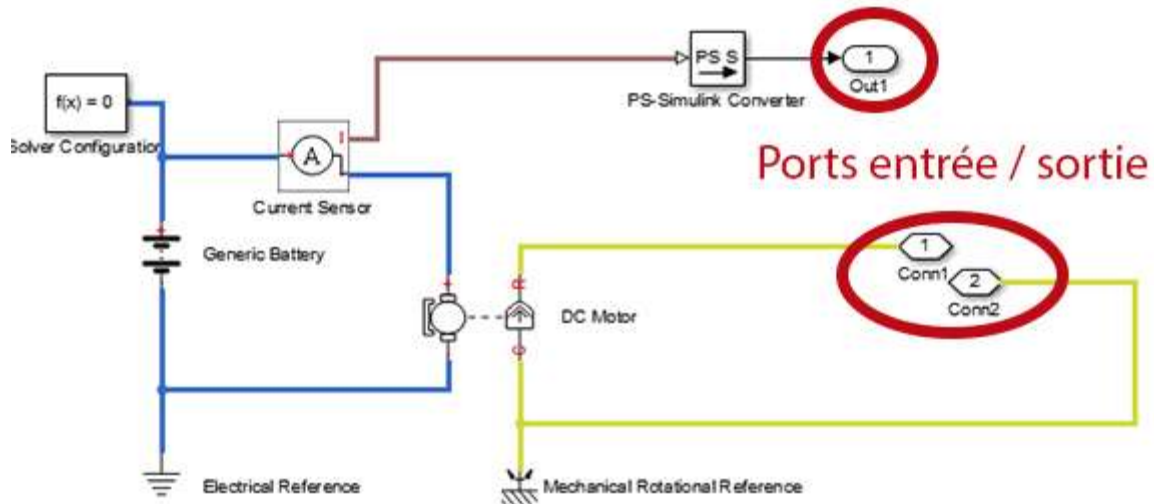


Cliquez dans le coin inférieur droit de la fenêtre sur : « Create subsystem ».

Vous obtiendrez quelque chose de similaire à l'image ci-dessous :



En double-cliquant sur le sous-système que vous venez de créer, vous le développez. Vous pouvez maintenant changer les noms des divers ports.

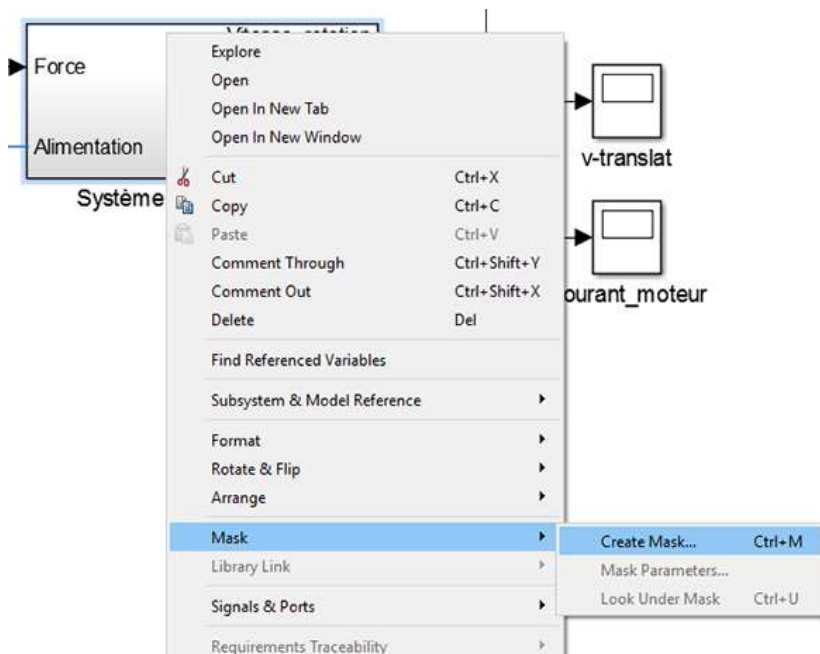


IX-B. Ajout d'une image▲

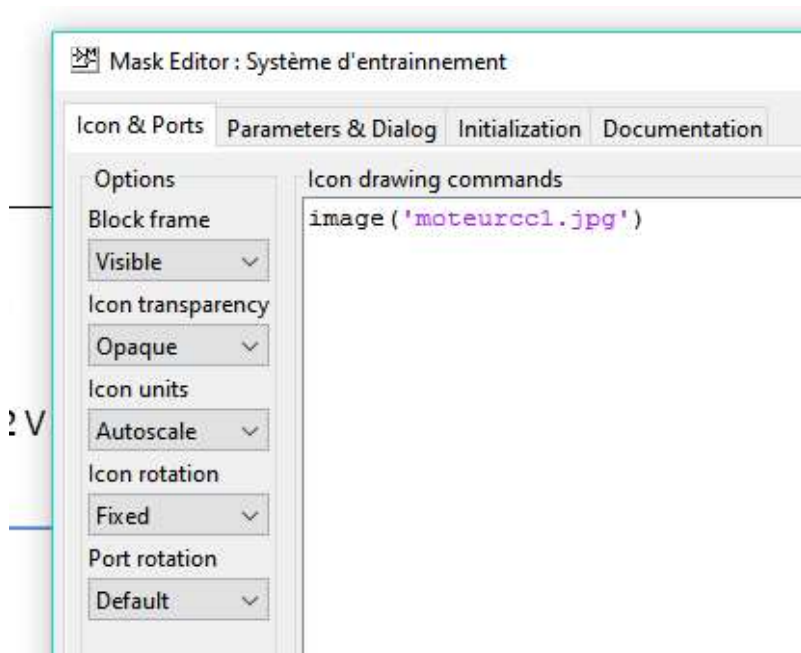
Pour rendre le modèle plus compréhensible, et plus agréable à regarder, il peut être intéressant d'ajouter des images sur les sous-systèmes. Cela présente aussi l'avantage d'identifier plus rapidement la fonction.

Vous devez au préalable avoir ajouté dans le dossier où se trouve votre modèle, la ou les images que vous souhaitez incorporer.

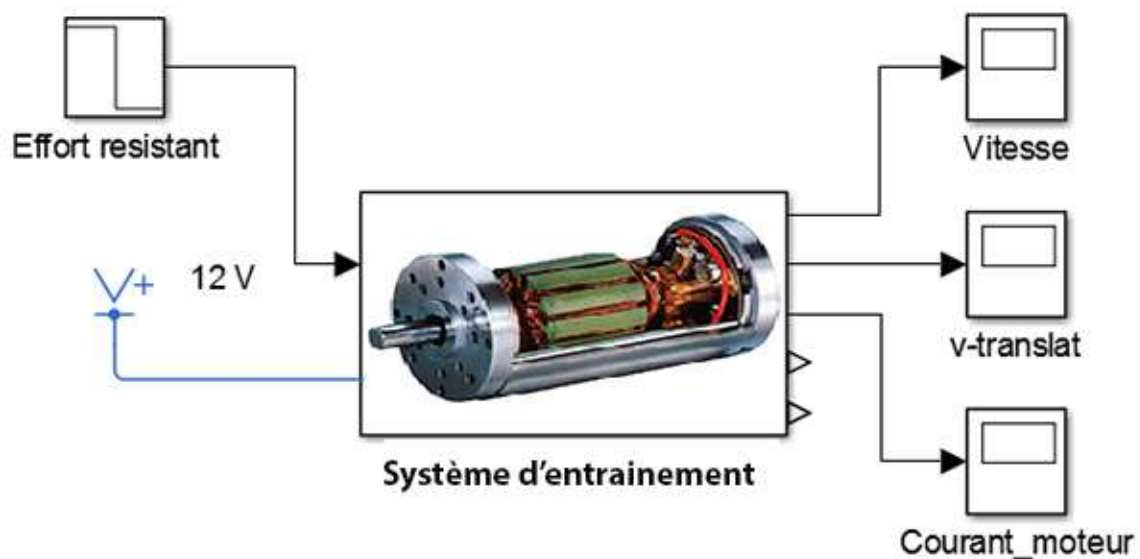
Effectuez un clic droit sur le sous-système, un menu déroulant apparaît. Cliquez sur « create mask » :



Dans la fenêtre, ajoutez : `image('moteurcc1.jpg')` :



Ne pas omettre l'extension du fichier. Si le fichier ne se trouve pas dans le même répertoire, vous devrez préciser le chemin.



X. Conclusion▲

Cette simulation nous permet de visualiser l'effet d'une modification d'un des divers paramètres moteur.

En augmentant l'inertie du rotor par exemple, vous constaterez que l'ensemble demande plus de temps pour atteindre la vitesse de régime établi.

Techniquement on crée le modèle, puis on le consolide (mise en correspondance des résultats de la simulation avec les mesures effectuées sur système réel).

L'ingénieur pourra à loisir tester toutes les modifications qu'il envisage sur le modèle, sans risquer d'abîmer le mécanisme.

Matlab intègre des « outils » pour la consolidation.