

## Chapitre 2

### Librairies de Simscape

---

#### I. La librairie « Foundation Library »

- I.1. Electrical
- I.2. Physical Signals
- I.3. Magnetic
- I.4. Mechanical
- I.5. Thermal

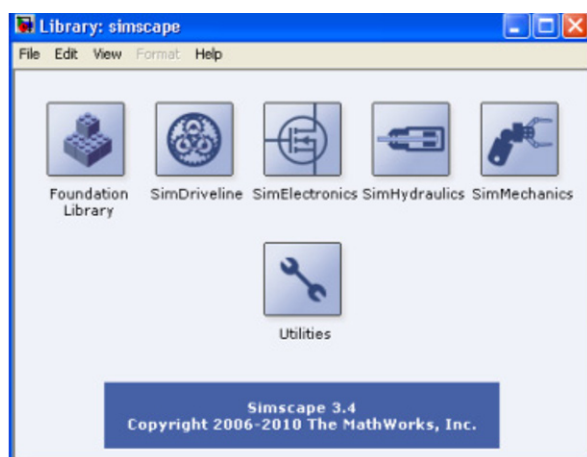
#### II. Utilities

#### III. SimElectronics

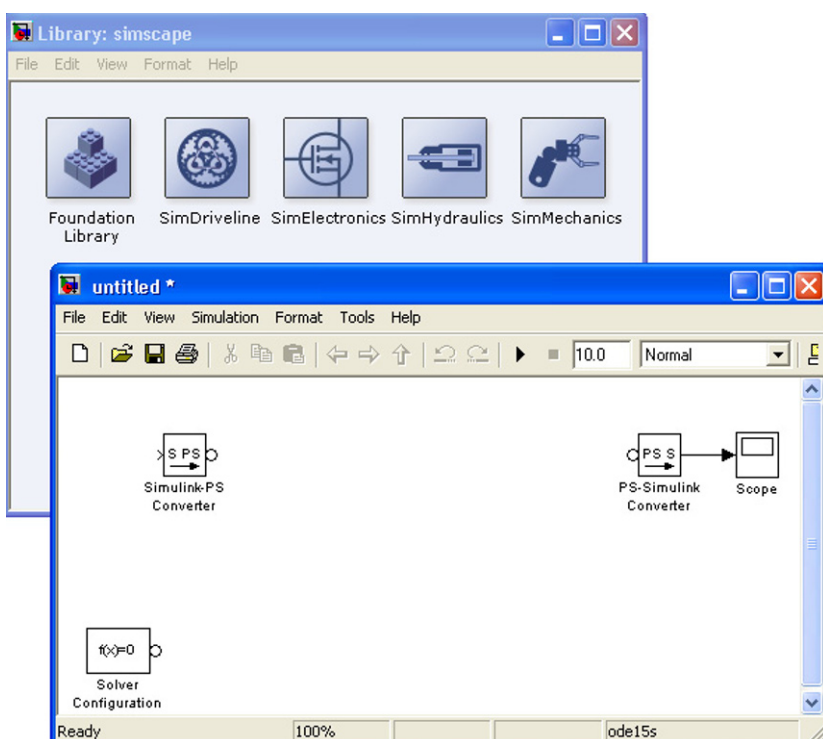
- III.1. Actuators & Drivers
  - III.2. Integrated circuits
  - III.3. Passive Devices
  - III.4. Semiconductor Devices
  - III.5. Additional Components
  - III.6. Sensors
- 

Pour visualiser les différentes librairies de Simscape, on peut lancer la commande `simscape` à partir du prompt de Matlab.

```
>> simscape
```

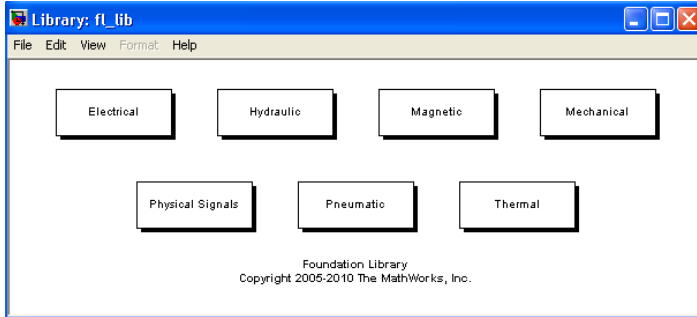


La commande `ssc_new` permet la création d'un nouveau modèle Simscape avec quelques blocs indispensables.



## I. La librairie « Foundation library »

Dans cette librairie, on trouve les éléments essentiels des domaines de l'électricité, la mécanique, la thermique, les signaux physiques, etc.

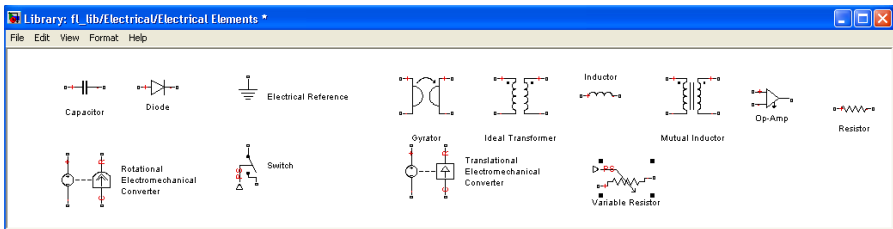


### I.1. Electrical

On y trouve les éléments, les sources et les capteurs électriques.

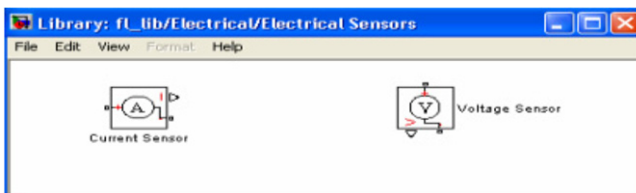
#### I.1.1. Electrical Elements

C'est une bibliothèque dans laquelle on trouve les éléments de base de l'électricité tels des résistances, des capacités, diode, self, amplificateur opérationnel, switch, ainsi que des éléments électromécaniques, de translation et de rotation.



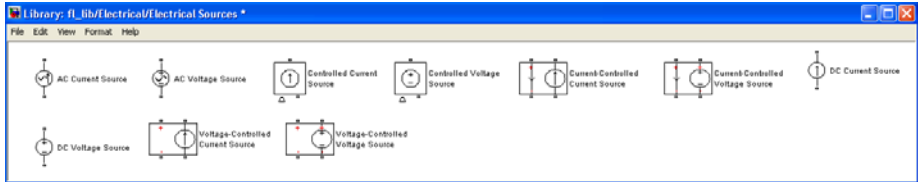
#### I.1.2. Electrical Sensors

Pour mesurer le courant et la tension, nous avons besoin de capteurs de tension et de courant qu'on trouve dans cette bibliothèque.



### 1.1.3. Electrical Sources

Dans cette bibliothèque on trouve différentes sources de tension, de courant, continues ou alternatives et des sources de tension, de courant contrôlées par un courant ou tension.



### 1.1.4. Applications

#### • Circuit RC

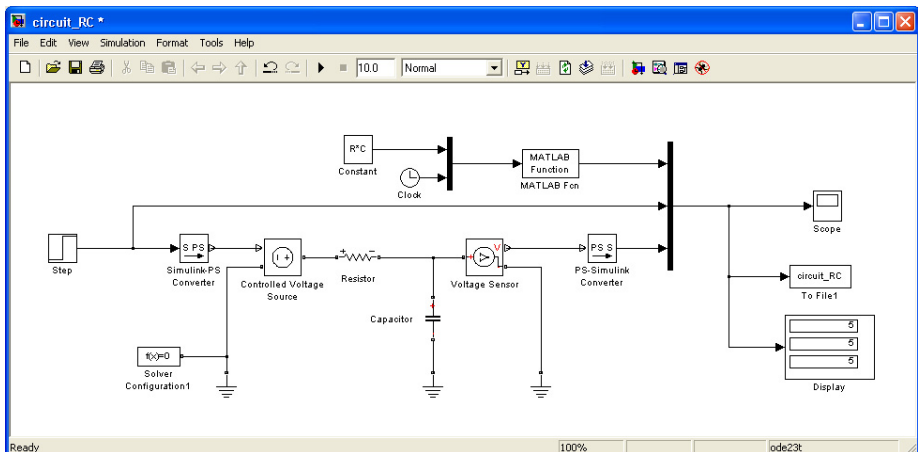
Dans le modèle Simulink suivant, on étudie la charge d'une capacité à travers une résistance. On envoie dans le fichier binaire `circuit_RC.mat`, les 3 signaux multiplexés suivants:

- la courbe théorique de la tension  $V_c$  aux bornes de la capacité,
- l'échelon d'entrée,
- la tension aux bornes du composant physique.

La courbe théorique, donnée par l'expression  $V_c = U_0 (1 - e^{-t/RC})$  est programmée dans le bloc MATLAB Function,

MATLAB function:

```
U0*(1-exp(-(u(2)/u(1))))
```

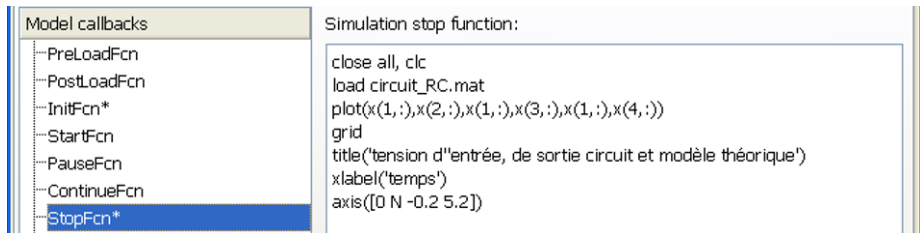


La tension d'entrée du circuit RC est un échelon unité de Simulink. Afin qu'il puisse être relié aux composants physiques de Simscape, nous utilisons le bloc  $S \rightarrow PS$  (bibliothèque Utilities de Simscape) pour le passage de Simulink à Physical Systems.

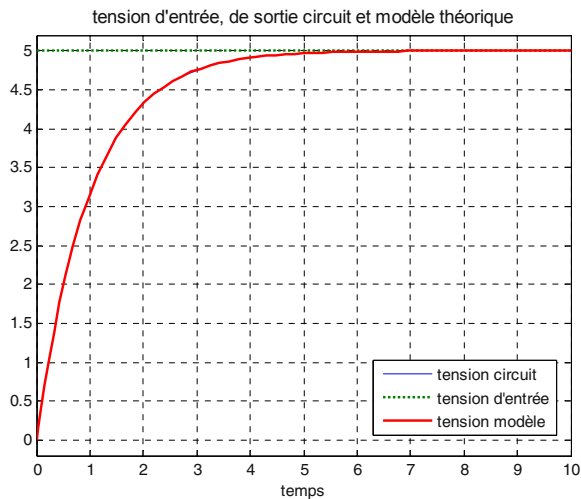


Après ce passage, il faut le transformer en tension électrique qu'on appliquera au circuit RC. Ceci se fait grâce à la source de tension contrôlée « Controlled Voltage Source » de la librairie « Foundation Library/Electrical/Electrical Sources ».

La tension aux bornes de la capacité est mesurée par le voltmètre ou capteur de tension « Voltage Sensor » de « Foundation Library/Electrical/Electrical Sensors ». Dans la fonction Callback InitFcn nous avons spécifié les valeurs de la résistance et de la capacité et dans StopFcn (fin de simulation) nous avons tracé les différentes courbes.



La tension mesurée aux bornes de la capacité est confondue avec la courbe de l'expression théorique.

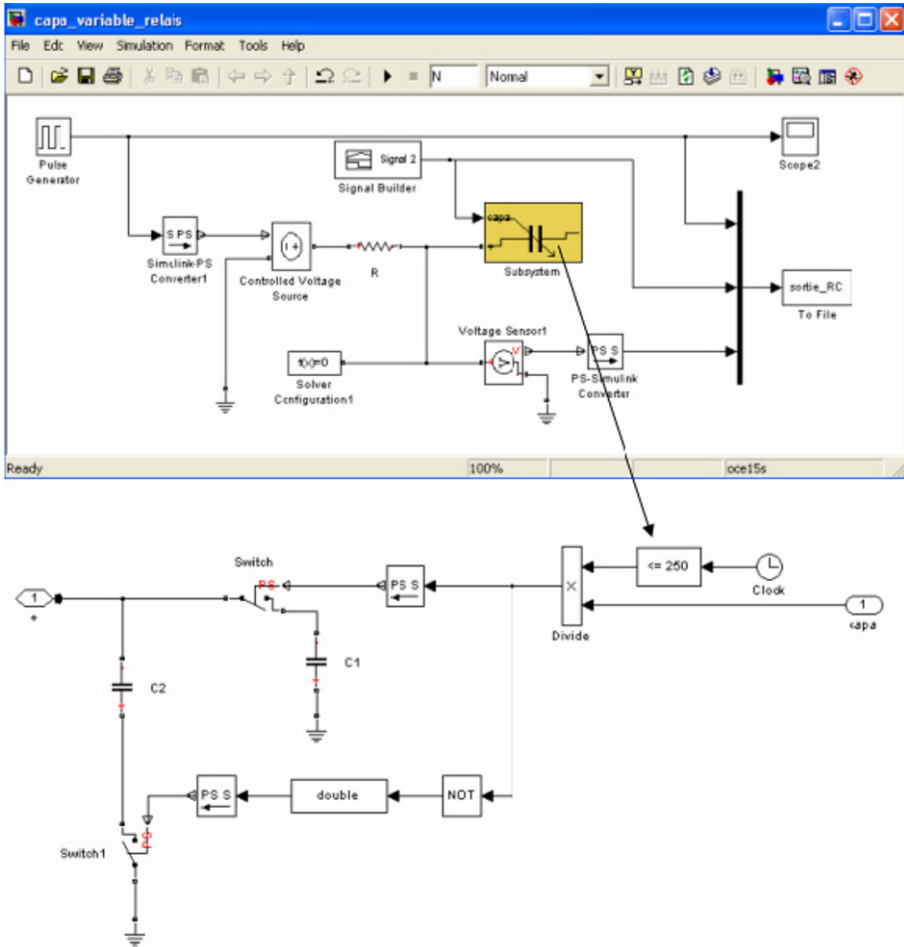


Dans la figure suivante, nous remarquons que la tension aux bornes de la capacité rejoint bien l'échelon d'entrée selon un régime du 1<sup>er</sup> ordre de constante de temps  $\tau = RC$ .

#### • Capacité variable avec relais

Dans l'exemple suivant, nous considérons une capacité variable (2 valeurs) en utilisant des relais. Soit le modèle suivant dans lequel le sous-système Subsystem est une capacité reliée à la résistance  $R=1\text{ k}\Omega$ , laquelle est attaquée par le signal carré Pulse Generator. Avant d'attaquer la résistance, le signal de type Simulink doit être converti en signal physique par le convertisseur S→PS et transformé en tension par le bloc Controlled Voltage Source.

Nous avons alors le cas simple d'une charge/décharge d'une capacité à travers la résistance  $R$  en réponse à un signal carré. Si on observe le contenu du sous-système, nous remarquerons la commutation de deux capacités par le jeu de 2 relais.

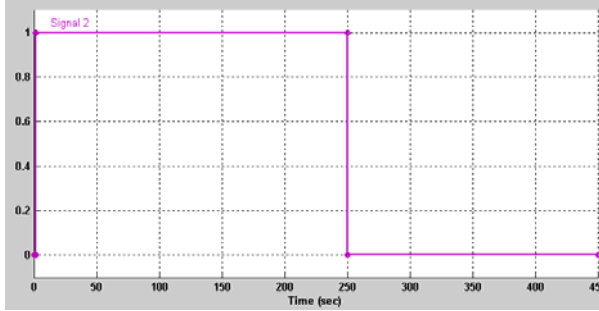


Lorsque  $t \leq 250$ , la valeur du signal fourni par le bloc Signal Builder vaut 1. Le block Switch reçoit une valeur logique 1 et se ferme, inversement au Switch1 qui s'ouvre grâce à porte logique NOT.

La borne inférieure de  $C2$  n'est pas à la masse. Ainsi, il n'y a que  $C1$  qui est reliée à la résistance  $R$ .

Lorsque  $t > 250$ , l'interrupteur Switch s'ouvre, la borne supérieure de  $C1$  n'est plus connectée au circuit, Switch1 se ferme, ainsi seule  $C2$  est reliée à la résistance  $R$ .

Le signal de commutation qui reliera successivement l'une ou l'autre des deux capacités est généré par le bloc Signal Builder.



Entre les instants  $t=0$  et 250, ce bloc génère une valeur 1 qui permet au relais `switch` de se fermer et de relier la capacité  $C1=2,7\text{mF}$  au signal d'entrée.

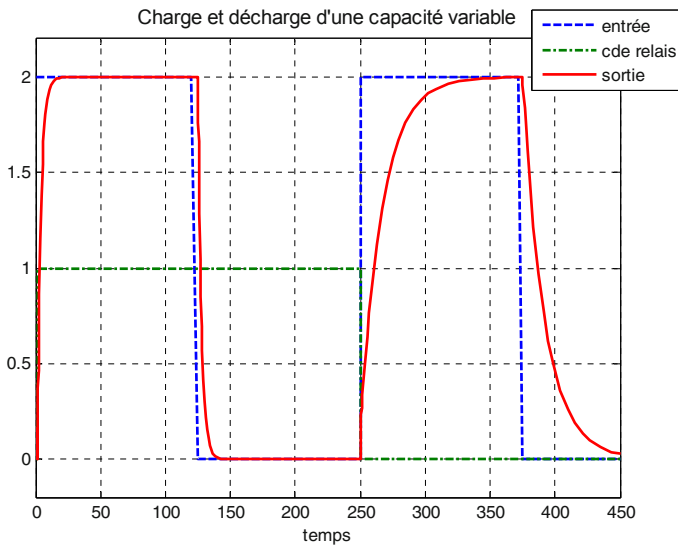
Grâce au bloc logique NOT, la capacité  $C2=17\text{mF}$  sera déconnectée du circuit en isolant une de ses pattes de la masse par l'ouverture du relais `switch1`.

Pendant cette durée, le circuit RC possède une constante de temps  $\tau = R C_1 = 10^3 * 2,7 \cdot 10^{-3} = 2,7\text{s}$ .

Au-delà de l'instant  $t=250$ , c'est la capacité  $C2=17\text{mF}$  qui est activée et  $C1$  désactivée, soit une constante de temps de 17s.

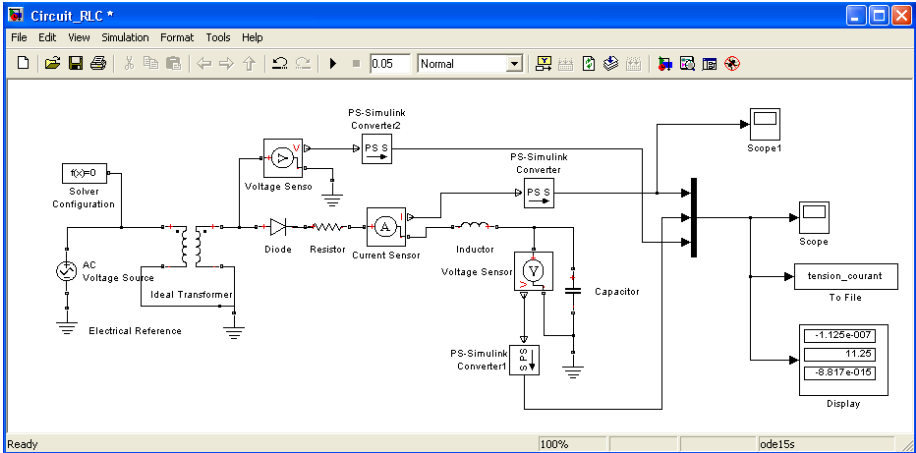
On observe dans la figure suivante, une charge/décharge plus lente au-delà de l'instant  $t=250$  grâce au changement de capacité reliée à la résistance, ceci grâce à la commutation simultanée des deux relais.

En sortie de la porte NOT, nous devons utiliser un convertisseur de type de signal : du type binaire à double. Ce convertisseur, `Data Type Conversion`, se trouve dans la bibliothèque `Signal Attributes` de Simulink.



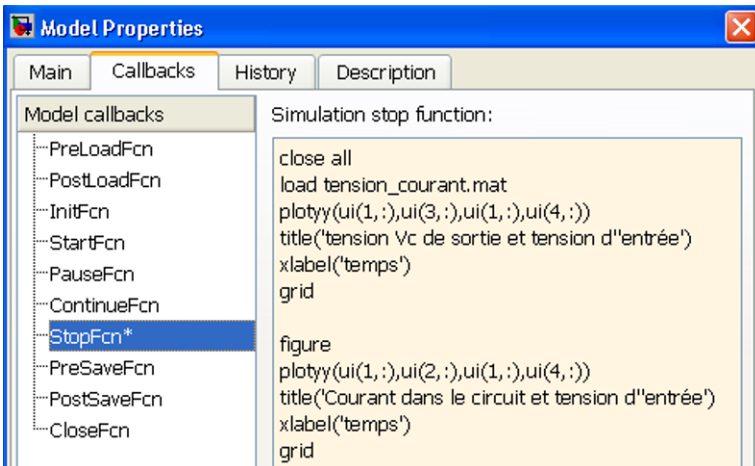
• *Circuit RLC, régime sinusoïdal*

Dans cette application, nous avons utilisé les éléments des 3 bibliothèques de la librairie Foundation Library/Electrical (sources, éléments et capteurs).



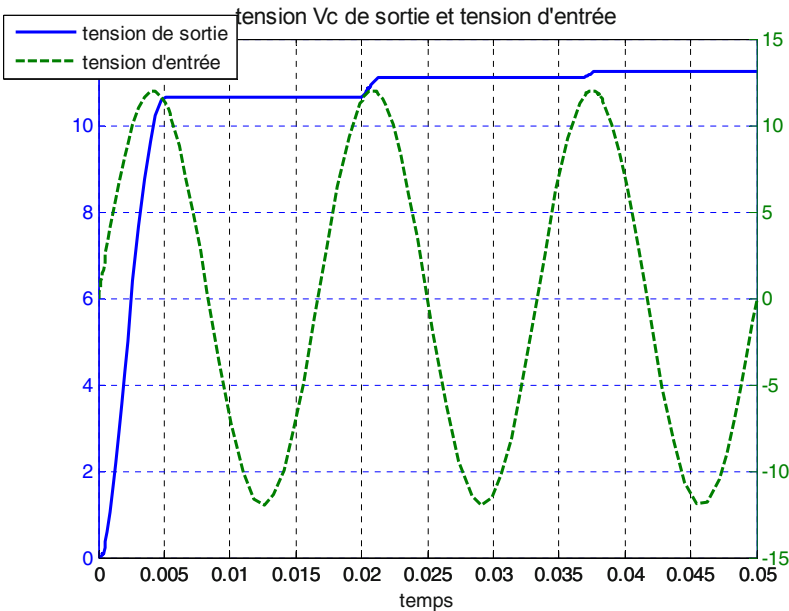
Le transformateur permet d'abaisser l'amplitude de 220V à 12V. L'ampèremètre ou capteur de courant (Current Sensor) est placé en série entre la résistance et la self pour mesurer le courant qui circule dans le circuit.

Dans les commandes de la fonction Callback `StopFcn`, nous lisons le fichier binaire `tension_courant.mat` et nous traçons les courbes de la tension d'entrée, de sortie et le courant circulant dans le circuit.

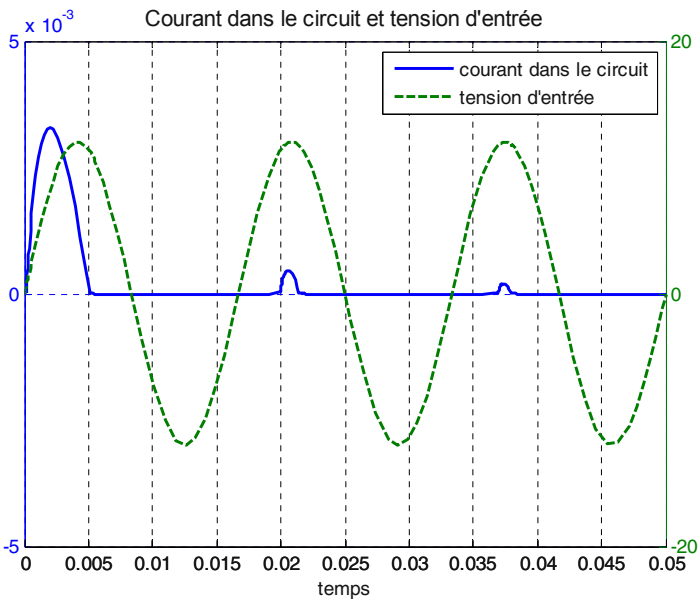


Dans les courbes suivantes, nous remarquons que le courant s'annule progressivement et la tension aux bornes de la capacité monte à chaque alternance positive et devient constante.

La tension de sortie augmente à chaque alternance et se stabilise à 11.25V.



Le courant s'annule au fur et à mesure de la charge du condensateur.



### • Redressement

Le relais permet de brancher ou de débrancher la capacité  $C$  en parallèle à la résistance  $R$ . Lorsque le relais est « énergisé » soit qu'il reçoit une tension supérieure à 0 à son entrée  $PS$ , la borne commune  $C$  est branchée à la borne  $S2$  (capacité branchée), autrement la borne commune  $C$  est branchée à la borne  $S1$  (masse) et la capacité débranchée du circuit.

En double-cliquant sur le bloc de ce relais nous obtenons la boîte de dialogue suivante.

**Relay**

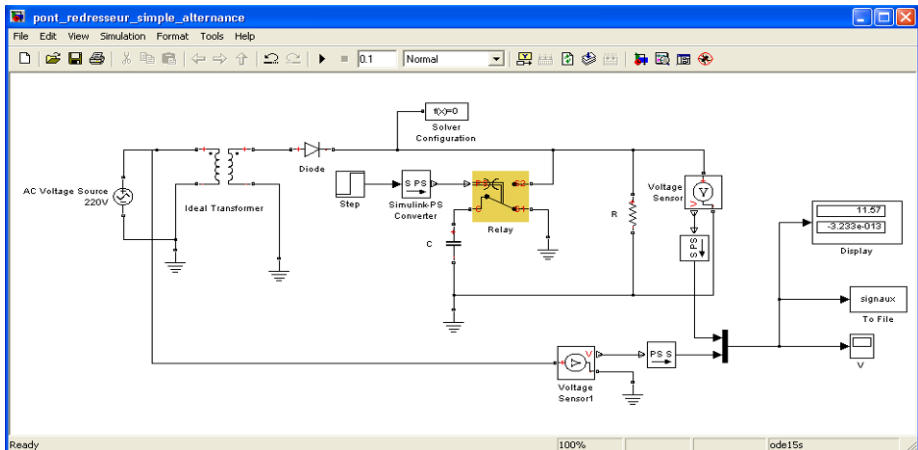
The block represents a relay. If the external physical signal  $PS$  is greater than the threshold, then the common terminal  $C$  gets connected to terminal  $S2$ , otherwise terminal  $C$  is connected to terminal  $S1$ .

The Connected resistance  $R$  and Open-circuit conductance  $G$  must be greater than zero.

**Parameters**

Time-to-break C-S1 connection:	0	s
Time-to-make C-S1 connection:	0	s
Time-to-break C-S2 connection:	0	s
Time-to-make C-S2 connection:	0	s
Connected resistance $R$ :	0.01	Ohm
Open-circuit conductance $G$ :	1e-08	1/Ohm
Threshold:	0	
Initial connection:	C to S1 closed, C to S2 open	

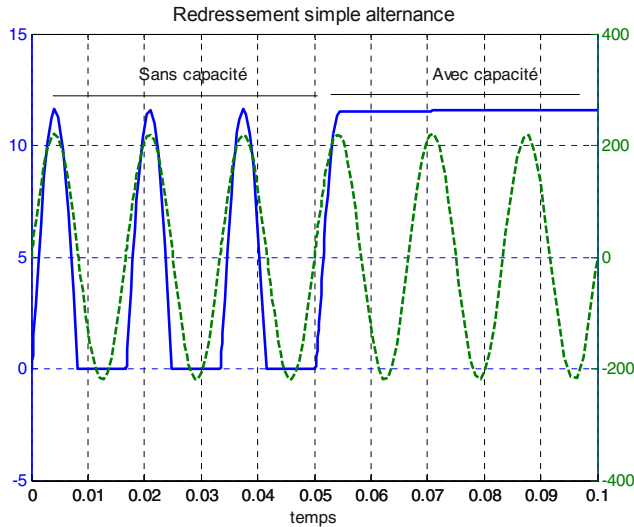
On peut spécifier les temps de la connexion de  $C$  à  $S1$  ou à  $S2$ , la résistance du relais fermé (ici  $0.01\Omega$ ) et la conductance lorsqu'il est ouvert (ici  $10^{-8}\Omega^{-1}$ ) soit une résistance de  $100M\Omega$ .



Dans la fonction Callback `StopFcn`, les commandes suivantes permettent de lire le fichier binaire `signaux.mat` et de tracer les signaux d'entrée et de sortie avec et sans capacité de filtrage.

```
load signaux.mat
plotyy(x(1,:),x(2,:),x(1,:),x(3,:)), grid
title('Redressement simple alternance'), xlabel('temps')
```

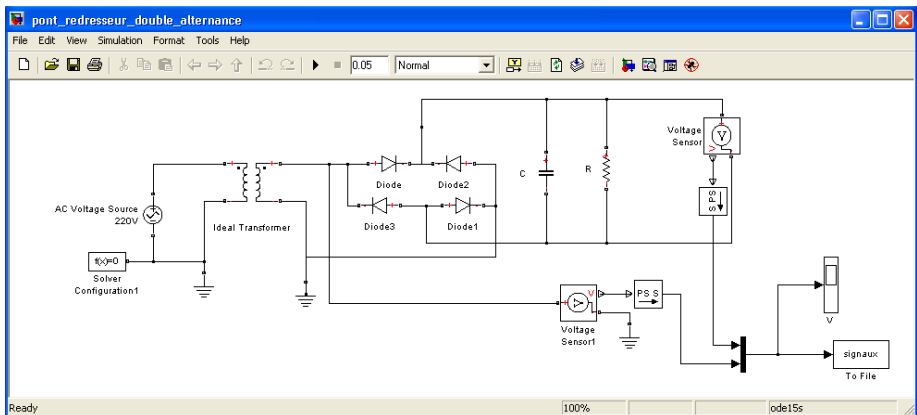
Les signaux sont tracés dans le même graphique avec des axes d'ordonnés différents grâce à la commande `plotyy`.



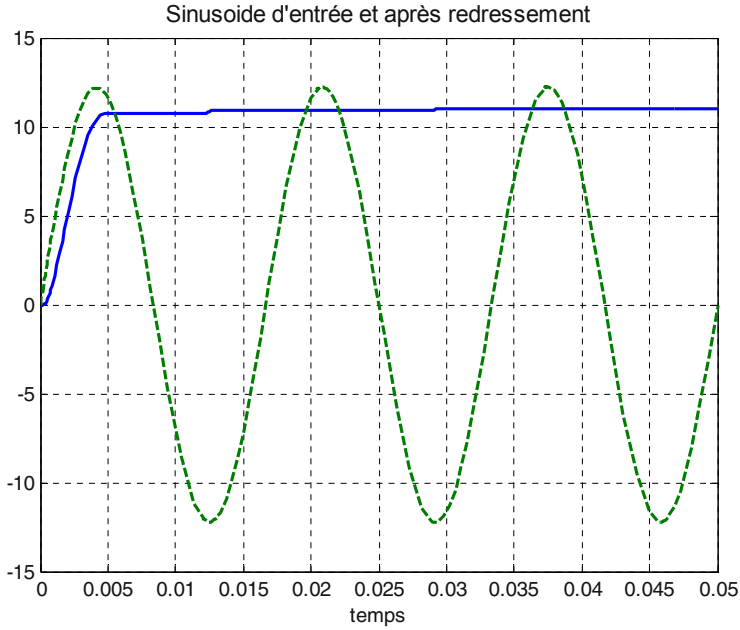
Grâce au filtrage par la capacité, la tension de sortie est constante et égale à 11.6V. Sans la capacité, nous n'obtenons que les alternances positives.

#### • Redressement double alternance

Le redressement d'une tension sinusoïdale permet d'avoir une tension continue. Il existe des redressements, simple comme le précédent et double alternance. Pour avoir un redressement double alternance, nous devons utiliser un pont à 4 diodes.



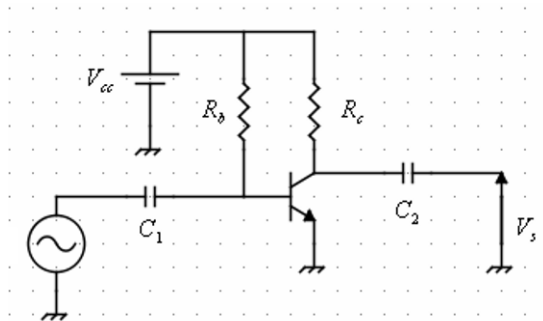
La capacité permet de faire un lissage et d'obtenir un signal presque continu en sortie.



• Schéma équivalent d'un transistor

Dans cette application, nous allons utiliser des sources contrôlées de tension par un courant et de courant par une tension.

Considérons le montage suivant d'un transistor monté en émetteur commun et polarisé en courant par la résistance  $R_b$ .



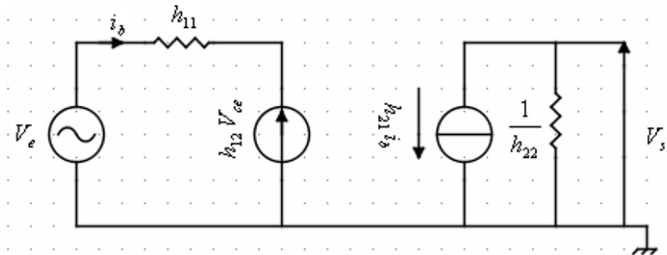
Le gain en tension de ce montage est donné par :

$$G = -\frac{\beta R_c}{h_{11}}$$



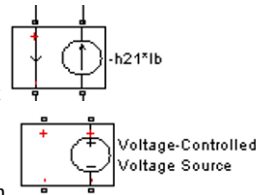
$\beta$  est le gain en courant et  $h_{11}$  l'impédance d'entrée.

Le schéma équivalent en régime sinusoïdal des petits signaux est le suivant :



La source de courant  $h_{21} i_b$  est contrôlée par le courant de base  $i_b$  et la source de tension  $h_{12} V_{ce}$  par la tension collecteur-émetteur  $V_{ce}$ . Ces 2 sources peuvent être simulées par les suivantes :

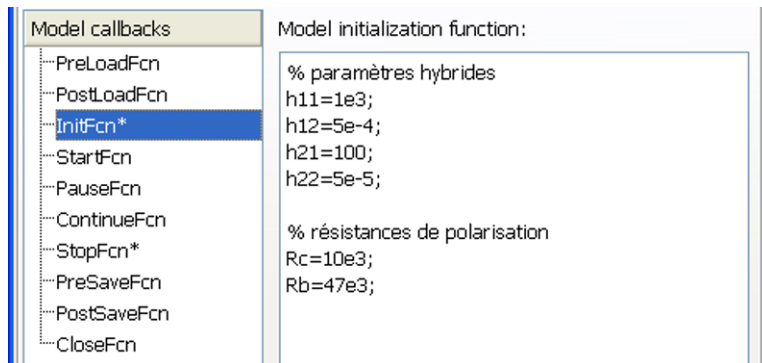
- les sources contrôlées de courant par un courant
- les sources de tension contrôlées par une tension ( $h_{12} V_{ce}$ )



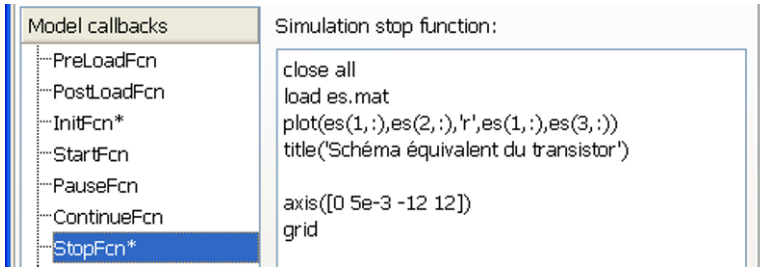
Les deux autres sources contrôlées de cette bibliothèque sont celle de courant contrôlée par une tension et de tension contrôlée par un courant.

Le modèle Simulink suivant simule le schéma équivalent du transistor. Son signal de sortie, ainsi que celui donné par la théorie sont envoyés dans le fichier binaire `es.mat` sous le nom de variable `es`.

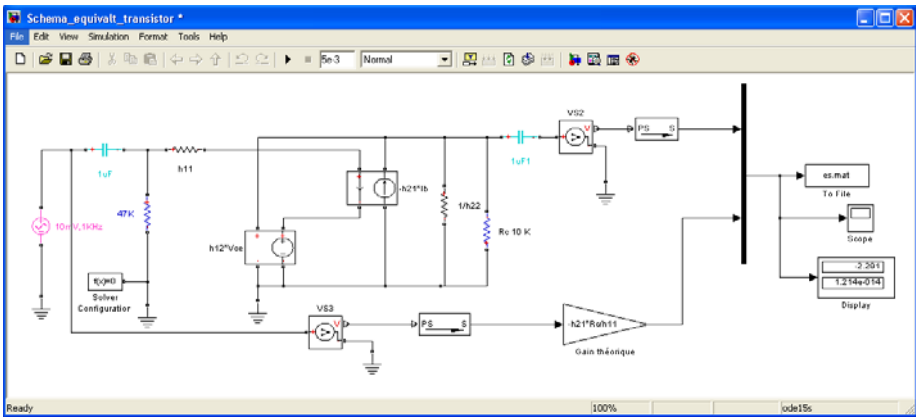
La fonction Callback `InitFcn` (Initialisation de la simulation) spécifie les valeurs des résistances et des paramètres du transistor.



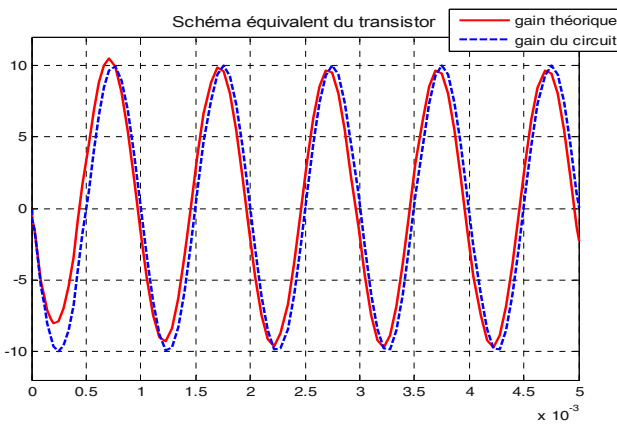
Dans le Callback `StopFcn` (fin de la simulation), on lit le fichier binaire et on trace les 2 signaux de sortie.



Le schéma équivalent est attaqué par un signal sinusoïdal d'amplitude 10 mV et de fréquence 1 kHz. Dans ce modèle, nous comparons la sortie du circuit physique et celle obtenue par l'expression théorique du gain.



La figure suivante représente la tension de sortie donnée par le circuit formé des sources contrôlées et celle donnée par le gain théorique.



Les 2 signaux, quasiment identiques, ont une amplitude de 10 V, soit un gain en tension de 1000.

Le tracé se fait dans la fonction Callback `StopFcn` dans laquelle on lit le fichier binaire `es.mat` pour récupérer ces 2 tensions.

On observe un léger régime transitoire ainsi qu'un infime déphasage entre la courbe du schéma réel et celle donnée par la théorie.

- **Utilisation des sources contrôlées de tension et de courant**

Pour ce nouveau schéma équivalent, la source de tension  $h_{12} V_{ce}$  est réalisée en ramenant la tension  $V_{ce}$  grâce à un voltmètre.

Cette tension sera multipliée par un gain (composant physique de la bibliothèque `Physical Signals/Functions`) pour contrôler la source de tension par une tension (`Controlled Voltage Source`).

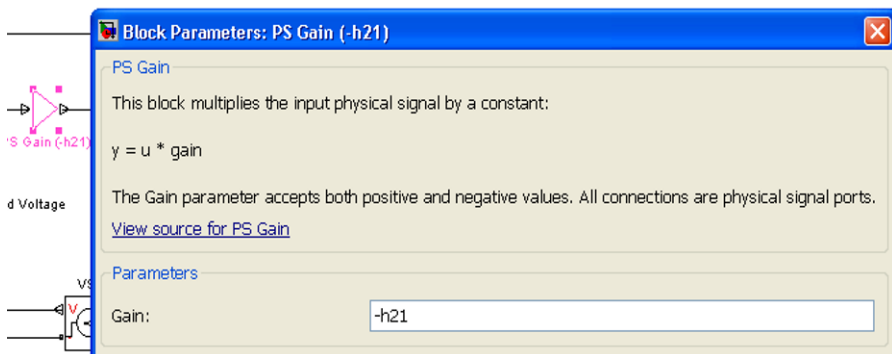
Les sources contrôlées, de tension et de courant sont dans la librairie `Foundation Library/Electrical/Electrical Sources`.

La source de courant  $h_{21} i_b$  est réalisée de la même façon grâce à l'ampèremètre.

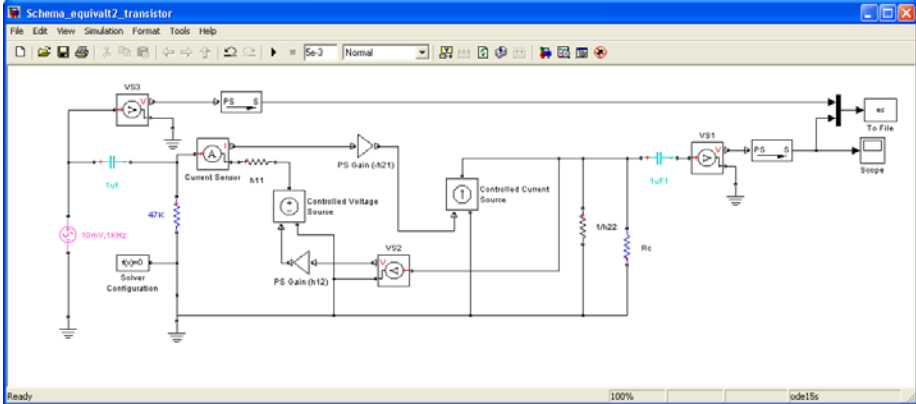
Ce courant, multiplié par le même gain, commande la source de courant contrôlée (`Controlled Current Source`).

Nous avons utilisé, en même temps, que les sources contrôlées, des composants physiques de gain dans lesquels nous avons mis respectivement les variables `h12` et `-h21` dont les valeurs sont spécifiées dans la fonction Callback `InitFcn`.

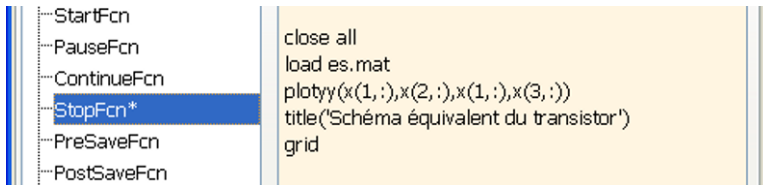
Dans la boîte de dialogue, les gains sont définis par les variables spécifiées dans cette fonction Callback `InitFcn`.



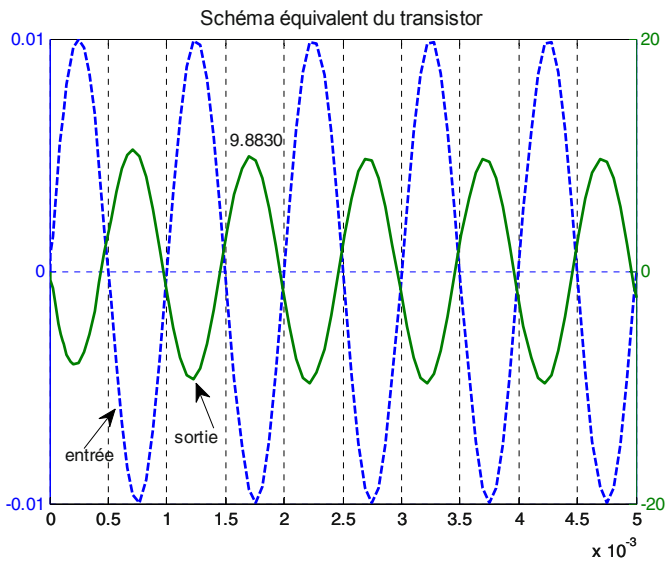
Le modèle hybride du transistor est défini par les gains  $-h_{21}$ ,  $h_{12}$  et les sources contrôlées de tension ( $h_{12} V_{ce}$ ) et de courant ( $-h_{21} i_b$ ).



La fonction Callback `StopFcn` permet de lire le fichier binaire et de tracer, dans deux ordonnées différentes, les signaux d'entrée et de sortie de ce nouveau schéma équivalent. Les résistances  $h_{22}^{-1}$  et  $R_c$  sont en parallèle.



Les 2 signaux sont quasiment en opposition de phase et nous retrouvons le même gain de 1000. Les 2 tensions sont tracées dans 2 axes d'ordonnées différents.

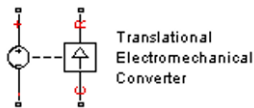


### • Mouvements mécaniques de translation et de rotation

Nous allons étudier les systèmes mécaniques de translation et de rotation, définis dans la librairie Foundation Library/Mechanical qui contient 5 bibliothèques :

- Mechanical Sources : Sources de couple, force, etc.
- Mechanisms : boîte à vitesse, levier, etc.
- Mechanical Sensors : capteur de force, vitesse, position et vitesse (translation et rotation), etc.
- Rotational Elements : Eléments de rotation (ressort de torsion, inertie, etc.).
- Translational Elements : Eléments de translation (masse, ressort de rappel, etc.).

#### - Mouvement de translation



Le bloc de la librairie Electrical/Electrical Elements, permet de convertir une tension  $U$  en mouvement de translation. Il réalise ainsi une conversion entre les domaines, électrique et mécanique. La vitesse de translation est donnée par :  $V = \frac{U}{K}$  avec  $K$  le coefficient de proportionnalité que l'on peut spécifier dans sa boîte de dialogue en double-cliquant sur ce bloc.

Parameters

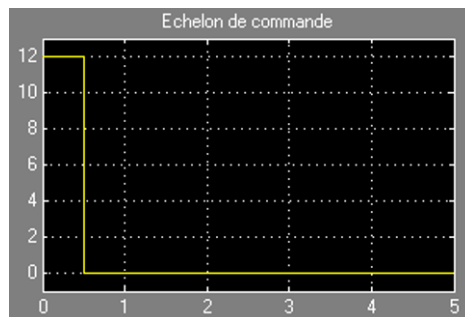
Constant of proportionality K:  s\*v/m

La force est proportionnelle au courant traversant la partie électrique.

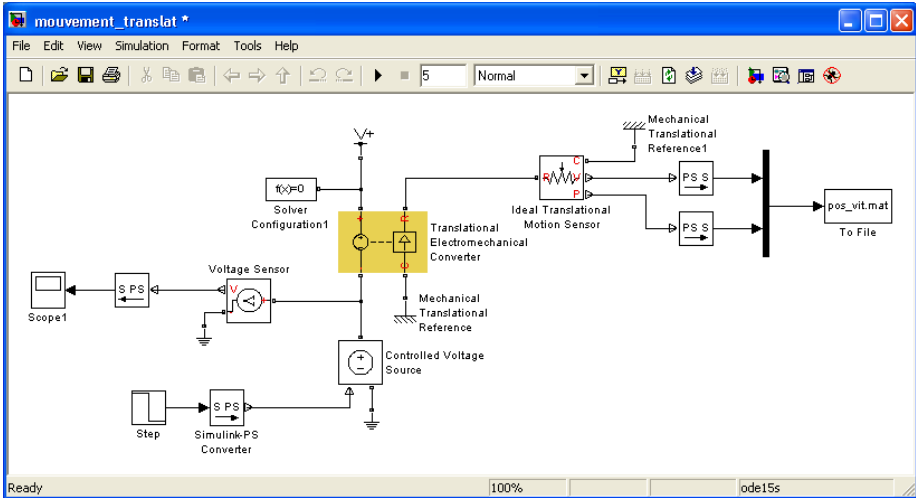
Dans le modèle suivant, nous appliquons un échelon de valeur 12V au pôle négatif de l'induit, le pôle positif étant relié à l'alimentation entre  $t=0$  à 2s afin d'appliquer 0V entre les ports électriques.

Nous utilisons pour cela une source de tension contrôlée commandée par une tension en échelon.

L'échelon revient à la valeur 0, ce qui applique 12V entre les ports électriques.



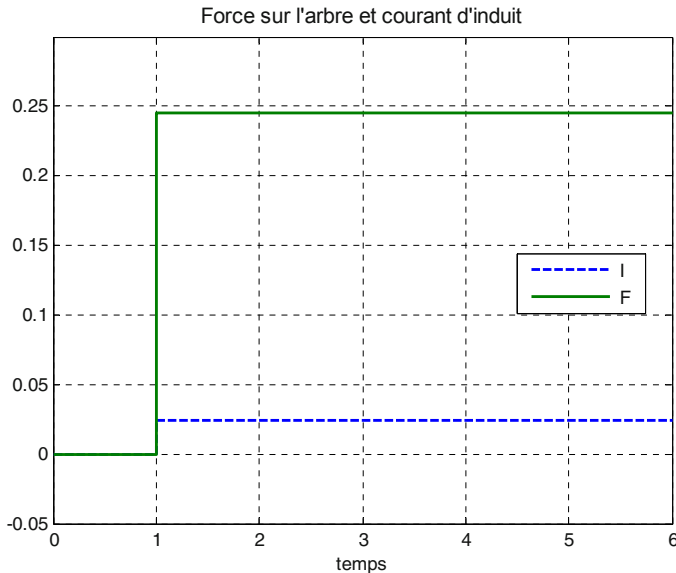
Nous obtenons une vitesse constante, proportionnelle à la tension, soit 120 rad/s et une position qui augmente linéairement.



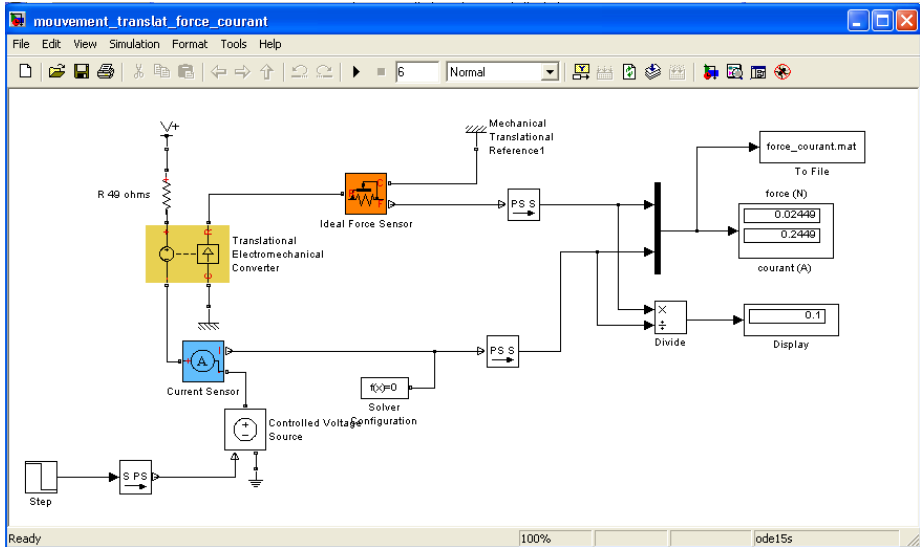
Dans la fonction Callback `StopFcn`, nous traçons la vitesse et la position qu'on lit préalablement du fichier binaire `pos_vit.mat`.

Si nous désirons vérifier la proportionnalité entre la force mécanique et le courant électrique, nous devons utiliser un capteur de force et un ampèremètre, comme on le montre dans le modèle suivant.

La courbe suivante montre la proportionnalité de la force au niveau du rotor et du courant d'induit.



Nous avons inséré une résistance de  $49\Omega$  entre les ports électriques, symbolisant la résistance d'induit et nous avons négligé son inductance.

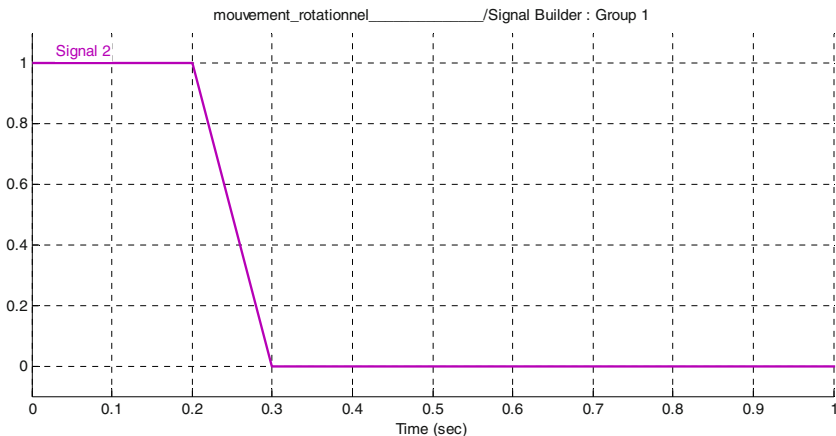


Les commandes insérées dans la fonction Callback `StopFcn` permettent de lire le fichier binaire `force_courant.mat`. Nous vérifions bien la valeur du rapport de 0.1 entre la force mécanique et le courant électrique.

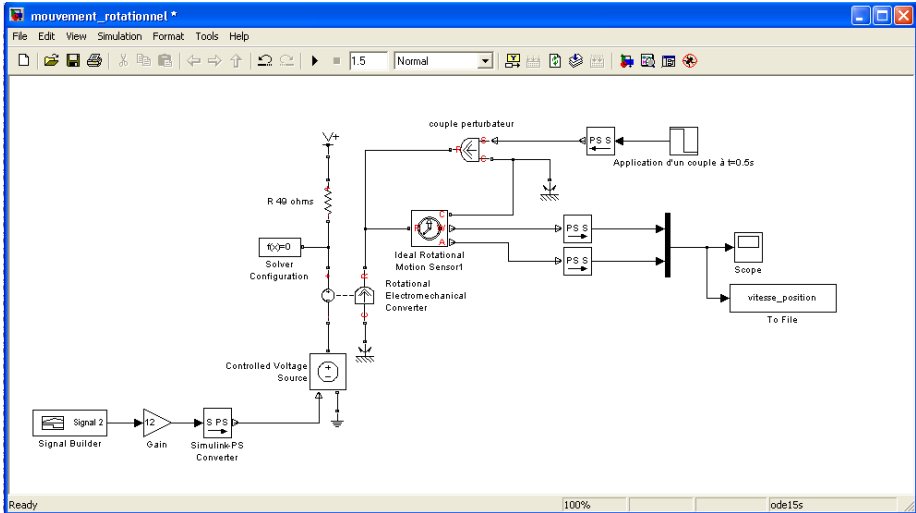
#### - Mouvement de rotation

Dans cet exemple, nous étudions le convertisseur électromécanique de rotation (Rotational Electromechanical Converter).

Nous imposons sur le port électrique inférieur une tension générée par le bloc Signal Builder qui a la forme suivante :



Ce signal est multiplié par un gain de 12 avant de commander la source de tension contrôlée (Controlled Voltage Source).



Nous notons que les références mécaniques pour les mouvements de translation et de rotation sont différentes, ainsi que les capteurs qui mesurent les vitesses et les positions (linéaires et angulaires).

Entre les instants 0 et 0.2, on applique 0 V comme différence de potentiel entre les 2 ports électriques, ce qui provoque le non démarrage du moteur (position et vitesse angulaires nulles).

Dès l'instant  $t=0.2$ , toute la tension d'alimentation de 12 V est appliquée sur les ports électriques, ce qui provoque une augmentation de la vitesse angulaire jusqu'à 120 rad/s.

A l'instant  $t=0.5$ , nous appliquons sur l'arbre mécanique un couple négatif (opposé au sens de rotation) de valeur  $-15 \cdot 10^{-3}$  N.m.

La vitesse de rotation chute alors à 46.96 rad/s.

Suite à cette perturbation, nous observons un point d'inflexion sur la courbe de position angulaire.

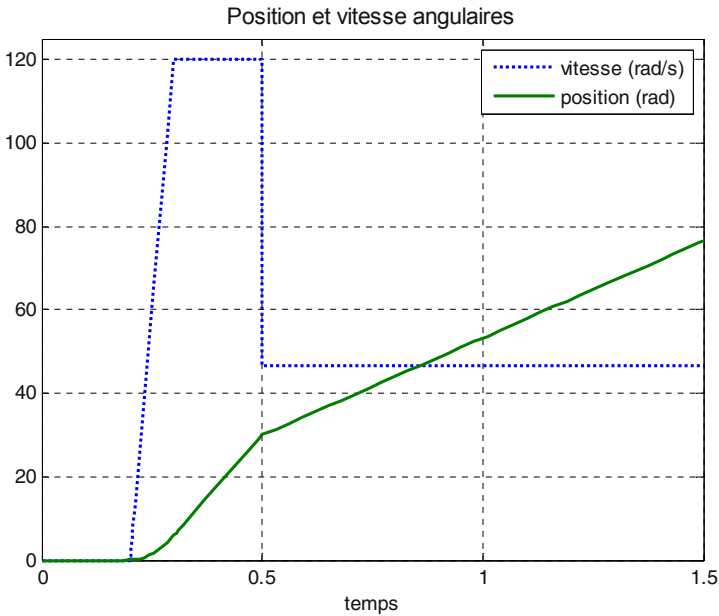
En régime permanent, le potentiel de la borne électrique inférieure devient nul ; ce qui a pour effet d'appliquer 12V aux bornes de l'induit.

La vitesse devient constante et la position continue de croître.

Les courbes de position et de vitesse angulaire sont données par la figure suivante.

Dans la fonction Callback `InitFcn`, on lit le fichier binaire et on trace les courbes de position angulaire et de vitesse de rotation.

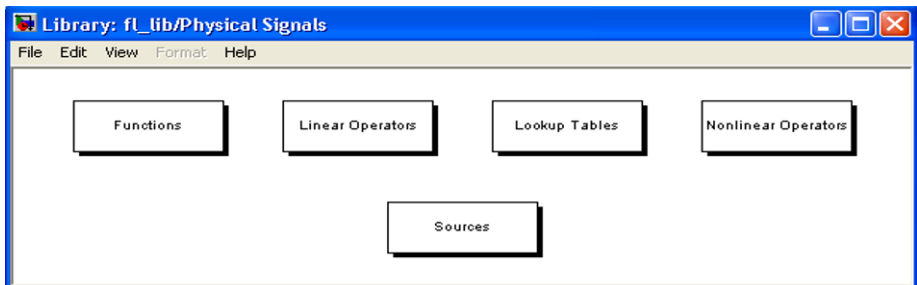




## I.2. Physical Signals

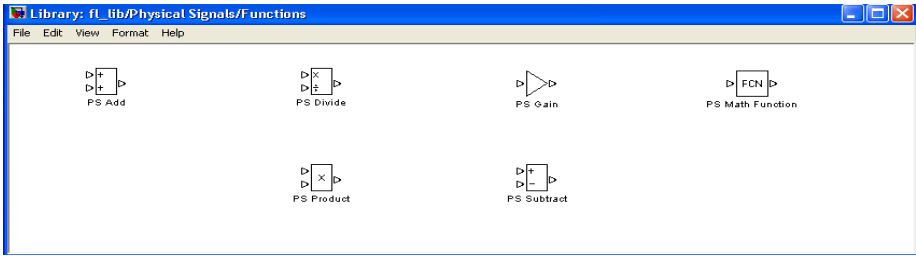
Cette librairie reprend, pour les signaux de type physique de Simscape, quelques blocs équivalents à ceux de Simulink pour réaliser:

- des opérations mathématiques,
- une intégration,
- une constante,
- des opérateurs non linéaires,
- ainsi que des tables d'interpolation, etc.



### I.2.1. Functions

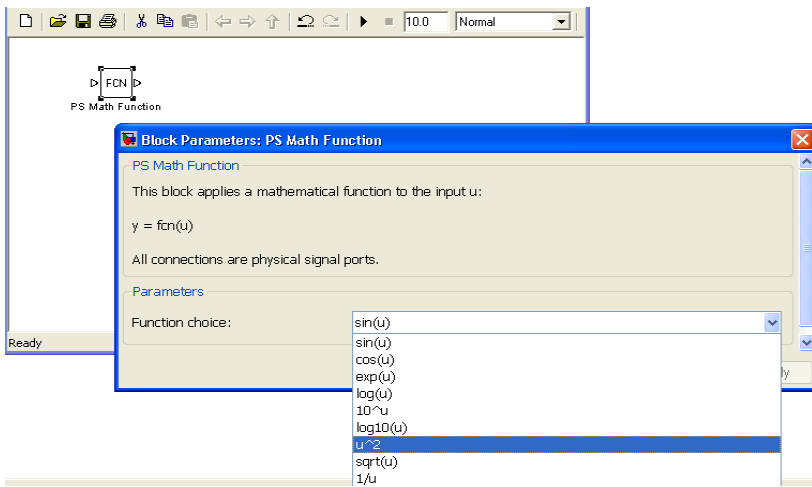
Ce sont des fonctions mathématiques de base qui opèrent sur des signaux physiques (entrées/sorties).



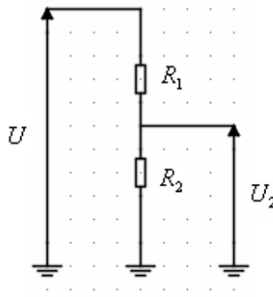
Le bloc `Fcn` équivaut au même type de bloc de Simulink dont il propose quelques unes de ses fonctions, mais en agissant sur des signaux physiques de Simscape.

Dans cette librairie nous avons des opérations simples telles que la somme, produit, gain, ainsi que le bloc `PS Math Function` qui réalise diverses opérations mathématiques.

Dans le modèle suivant, ce bloc est utilisé pour calculer le carré de la tension et la puissance au niveau de la résistance.



Considérons un exemple très simple du pont diviseur de tension suivant.



Dans le modèle suivant, nous calculons la puissance au niveau de la résistance  $R_2$  de 2 façons différentes.

Les valeurs des résistances  $R_1$  et  $R_2$  sont spécifiées dans la fonction Callback d'initialisation `InitFcn`.

Nous utilisons pour cela l'opérateur physique `PS Divide` de Simscape.

- La puissance par l'expression  $\frac{U_2^2}{R_2}$ .

```
>> U=12 ;
>> U2=(R2/(R1+R2))*U; % pont diviseur de tension
>> % Calcul de la puissance
>> P=U2^2/R2
P =
    0.0032
```

Le bloc `FCN` (`PS Math Function`) réalise l'opération  $u^2$  tandis que le gain `PS Gain` possède la valeur  $1/R_2$ .

- La puissance par  $U_2 * I = U_2 * \frac{U}{R_1 + R_2}$

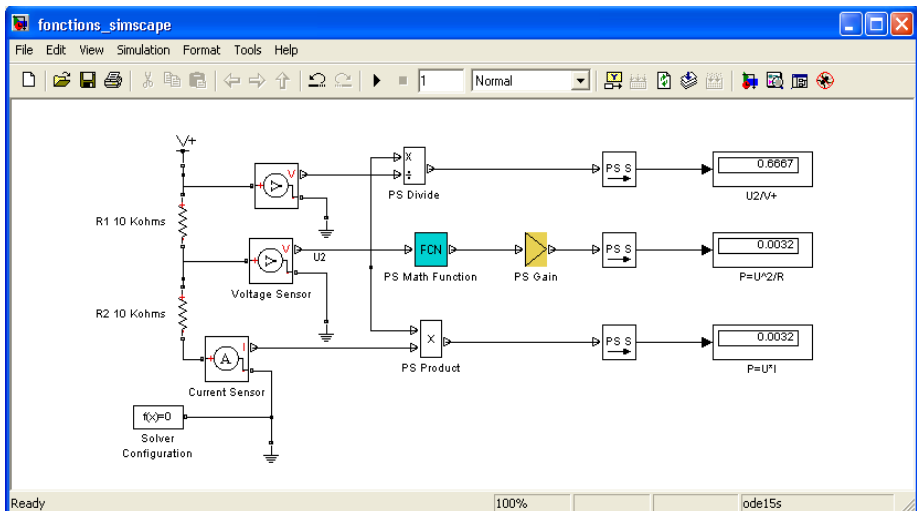
```
>> P=U2*U/(R1+R2)
P =
    0.0032
```

La tension aux bornes de  $R_2$  et le courant la traversant sont mesurés par des capteurs, de tension et de courant.

Il suffit juste d'utiliser l'opérateur produit `PS Product`.

Dans le modèle suivant, nous calculons la tension aux bornes de la résistance  $R_2$  et la puissance qu'elle consomme, en utilisant les 2 façons précédentes.

Le but de ce modèle étant d'utiliser les opérateurs de Simscape.



### 1.2.2. Linear Operators

Cette librairie se résume à l'intégrateur analogique PS Integrator.

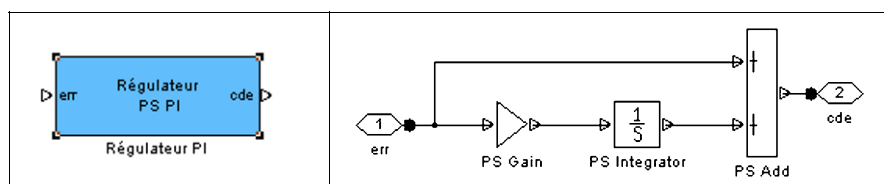


• *Application à la régulation de la position linéaire (translation)*

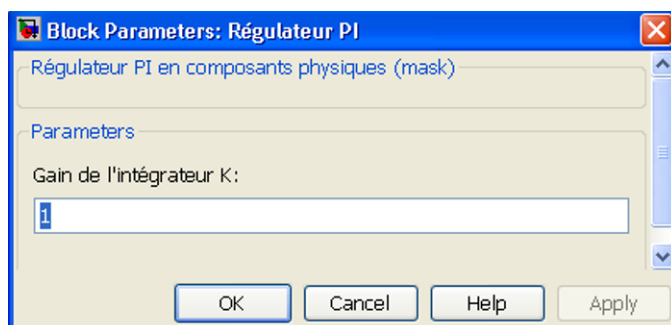
On utilise cet opérateur d'intégration pour réaliser le régulateur Proportionnel et Intégral PI d'expression :

$$D(p) = 1 + \frac{K}{p}$$

On réalise alors un masque pour programmer ce régulateur (Cf. Annexe Masques) pour réguler la position linéaire d'un convertisseur électromécanique.



En double-cliquant sur le masque, on obtient la boîte de dialogue dans laquelle on spécifie le gain K de l'intégrateur.

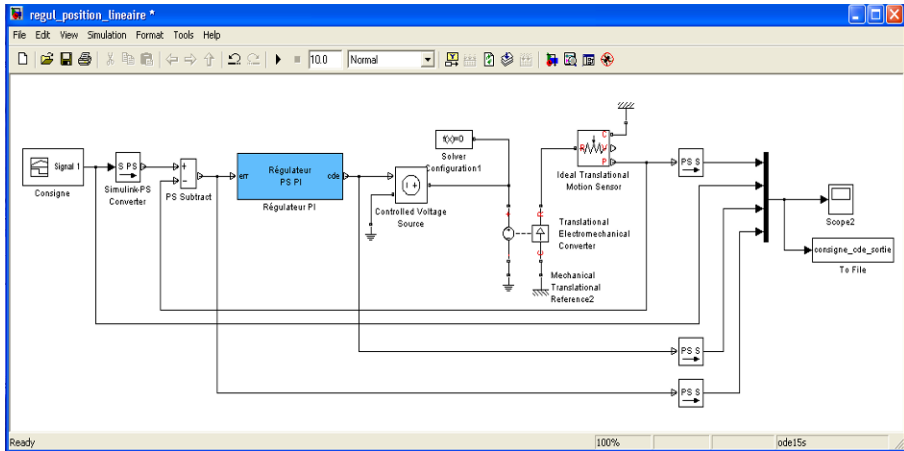


Le modèle suivant montre l'utilisation du régulateur pour contrôler la position du convertisseur électromécanique de translation.

La sortie A, angle, du capteur Ideal Translational Motion Sensor revient à l'entrée pour être comparée à la consigne.

Cette erreur constitue l'entrée du régulateur.

Le signal de consigne est généré par le bloc Signal Builder de Simulink.

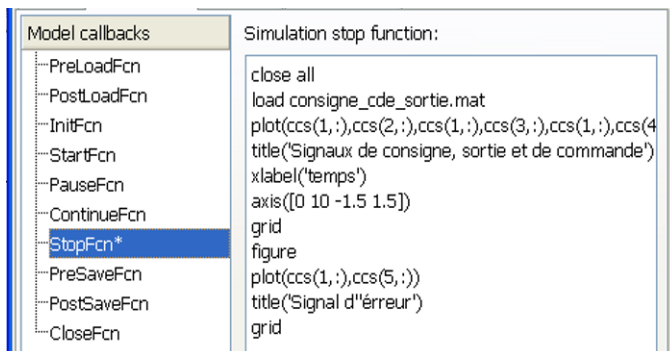


Après le régulateur, afin de transformer le signal de commande en tension qui s'applique au côté électrique du convertisseur électromagnétique, on utilise la source contrôlée de tension (Controlled Voltage Source).

Le signal à réguler est la position linéaire de la partie mobile du côté mécanique d'un convertisseur électromécanique défini par les 2 relations suivantes :

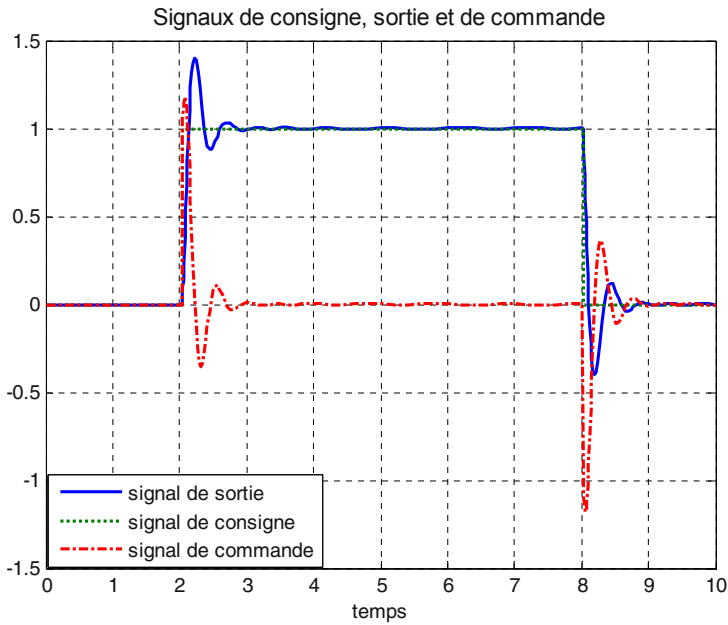
- $F = K I$ , avec  $F$  la force mécanique et  $I$  le courant dans la partie électrique,
- $V = K v$ , avec  $V$  la tension aux bornes de la partie électrique et  $v$  la vitesse de la partie mécanique.

Dans la fonction Callback `StopFcn`, nous récupérons les signaux sauvegardés dans le fichier binaire `consigne_cde_sortie.mat` afin de les tracer.



Dans l'exemple suivant, nous choisissons un gain  $K=20$ .

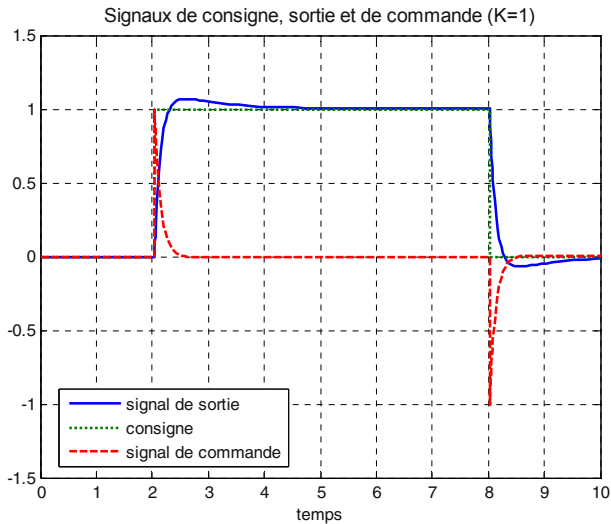
Nous observons, pour ce gain, des oscillations amorties du signal de sortie en régime transitoire pour rejoindre le signal de consigne en régime permanent.



Le système en boucle fermée est, au moins, du second degré.

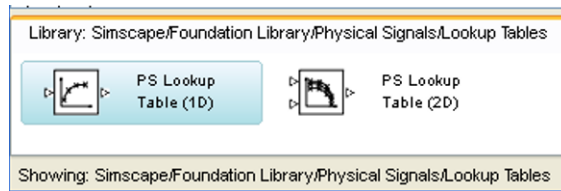
Le signal de commande oscille avec la même pulsation non amortie et s'annule en régime permanent.

Avec un gain  $K=1$ , le signal de position atteint la consigne avec un seul dépassement. En statique, lorsque la valeur constante de consigne est atteinte, le signal de commande s'annule.



### 1.2.3. Lookup Tables

Ce sont, comme dans Simulink, des tables d'interpolation à une et deux dimensions mais en agissant sur des signaux physiques de Simscape.



Dans les exemples suivants, on se propose, d'une part, d'utiliser les opérateurs mathématiques précédents et d'autre part, d'obtenir différentes valeurs d'une thermistance en fonction de sa température dans le but d'utiliser une table d'interpolation pour en faire un capteur.

Dans ce premier modèle, nous nous limitons au calcul de la valeur de la thermistance en fonction de sa température.

En série à la thermistance, nous avons placé une résistance  $R=10\text{ k}\Omega$  pour réaliser un pont diviseur de tension.

L'alimentation possède une valeur de 5 V.

La tension aux bornes de la thermistance est alors, donnée par l'expression :

$$U_{R_{th}} = \frac{R_{th}}{R_{th} + R} V_{cc}$$

De cette expression, nous tirons celle de la valeur de la thermistance.

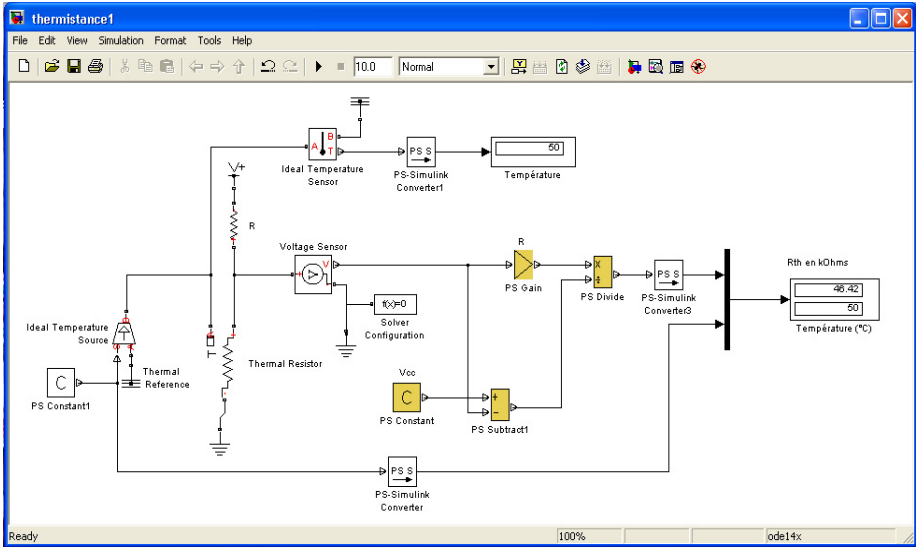
$$R_{th} = R \frac{U_{R_{th}}}{V_{cc} - U_{R_{th}}}$$

Cette relation est programmée par les blocs physiques de calculs élémentaires de Simscape dont le gain PS Gain vaut la valeur de R et la constante PS Constant est égale à la tension d'alimentation  $V_{cc}$ .

La borne T (température) du composant physique de la thermistance est soumise à une température donnée par la constante PS Constant1 qui commande une source de température (Ideal Temperature Source).

Nous obtenons la valeur de  $46.42\text{ k}\Omega$  de la thermistance lorsqu'on la soumet à une température de  $50^\circ\text{C}$ , ce qu'on peut observer à la sortie T du capteur (Ideal Temperature Sensor).

D'autres blocs thermiques seront étudiés dans la partie correspondante de cette librairie.



Pour avoir un certain nombre de points dans un domaine spécifié de température, nous allons créer un script dans lequel on simulera ce modèle et on sauvegardera les différentes valeurs dans un fichier texte.

Nous obtiendrons 2 vecteurs correspondants aux températures et résistances de la thermistance. Ces vecteurs seront utilisés par la table d'interpolation linéaire PS\_Lookup Table (1D).

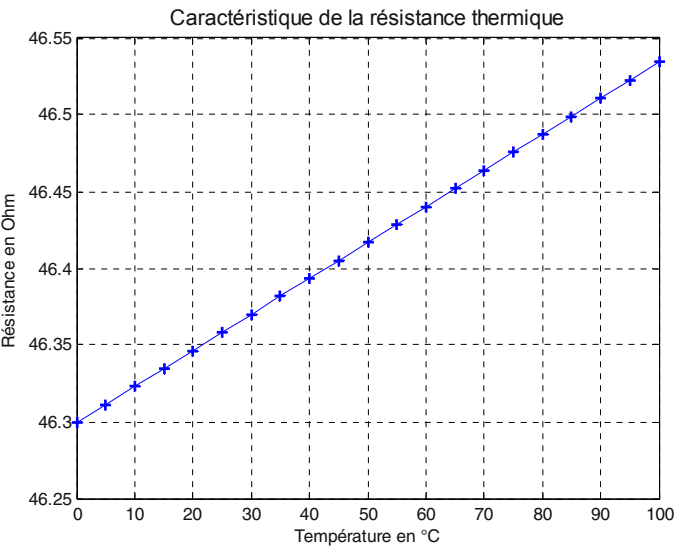
```
fichier caract_thermistance.m
```

```
clear all
clc
close all
R=10;
Vcc=5;
temp=0;
x=[];

for i=0:1:20
    temp=i*5;
    sim('thermistance2')
    RT=RT(1,:)';
    x=[x,RT]
end

plot(x(2,:),x(1,:))
hold on
plot(x(2,:),x(1:,:),'+')
title('Caractéristique de la résistance thermique')
xlabel('Température en °C')
ylabel('Résistance en Ohm')
grid
```

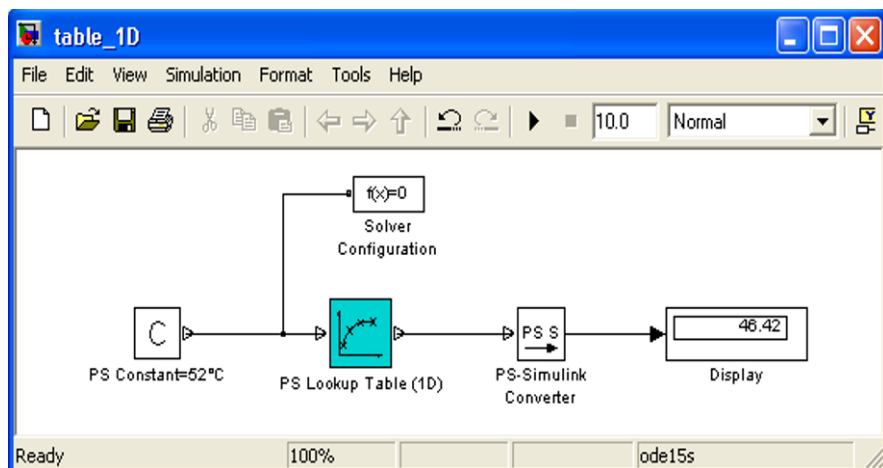




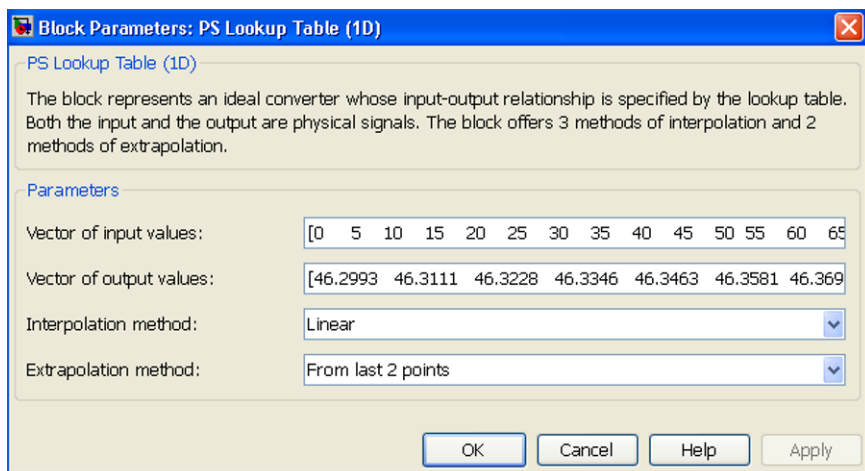
Grâce à ce script, nous obtenons la variable `x` qui contient la valeur de la résistance et la température correspondante, comme suit.

R(Ω)	T(°C)
46.2993	0
46.3111	5.0000
46.3228	10.0000
46.3346	15.0000
46.3463	20.0000
46.3581	25.0000
46.3698	30.0000
46.3816	35.0000
46.3933	40.0000
46.4051	45.0000
46.4168	50.0000
46.4286	55.0000
46.4403	60.0000
46.4521	65.0000
46.4638	70.0000
46.4756	75.0000
46.4873	80.0000
46.4991	85.0000
46.5108	90.0000
46.5226	95.0000
46.5343	100.0000

Nous allons utiliser une table à une dimension pour fournir la valeur de la résistance en fonction de la température.



En double-cliquant sur le bloc PS Lookup Table (1D), on entre les valeurs de la température comme celles du vecteur d'entrée et les valeurs correspondantes de la résistance de la thermistance comme éléments du vecteur de sortie.



Dans le modèle précédent, la constante vaut 52°C.

La valeur estimée de la résistance est donnée par l'interpolation linéaire entre les points de température 50 et 55°C dont les valeurs de la résistance sont respectivement de 46.4168Ω et 46.4286Ω. Pour trouver les coefficients  $a$  et  $b$  de la droite passant par ces 2 points, nous nous proposons de résoudre le système suivant à 2 inconnues:

$$\begin{bmatrix} 50 & 1 \\ 55 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 46.4168 \\ 46.4286 \end{bmatrix} \iff A\beta = B$$

Nous le résolvons de 2 manières différentes :

$$\beta = (A^T A)^{-1} A^T B$$

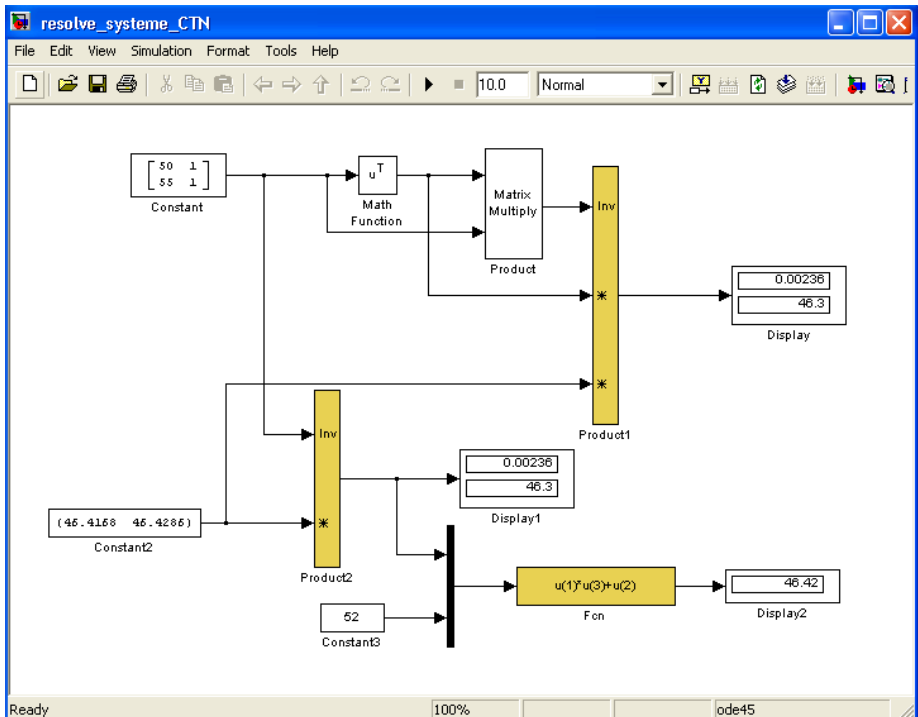
et

$$\beta = A^{-1} B$$

Comme ce système est déterminé, ces 2 méthodes donnent le même résultat que celui de la table d'interpolation.

```
>> inv(A'*A)*A'*B

ans =
    0.0024
   46.2988
```



La droite qui passe par les points extrêmes a pour équation :

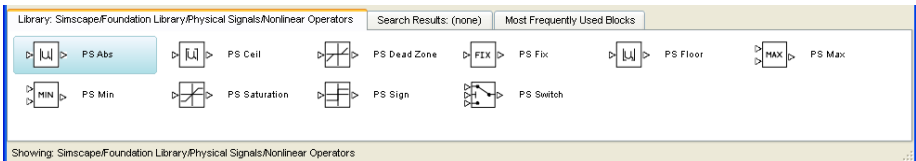
$$y = 0.00236 x + 46.3$$

Pour  $x = 52$ , nous obtenons bien la même valeur de  $46.42 \Omega$ .

Dans ce modèle, nous avons trouvé la même valeur de différentes façons.

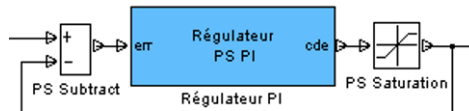
### 1.2.4. Nonlinear Operators

Ces opérateurs effectuent les mêmes opérations que leurs homologues de Simulink : valeur absolue (PS Abs), calcul des valeurs, maximale et minimale d'un signal (PS Max et PS Min, etc.).

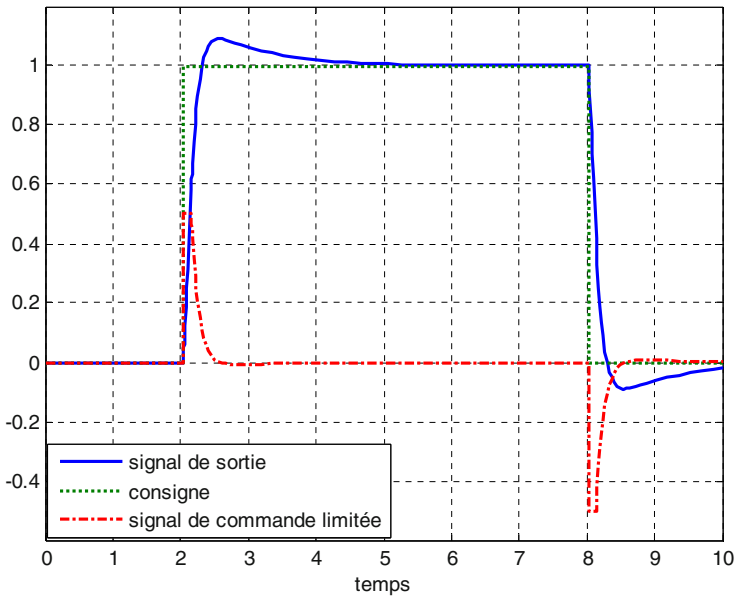


Dans le cas de la régulation de la position précédente (Application à la régulation de la position linéaire, translation), nous remarquons que le signal de commande évolue brutalement au changement de consigne.

Si l'on veut diminuer la hauteur du saut du signal de commande, on peut le saturer en utilisant le bloc de saturation PS Saturation.



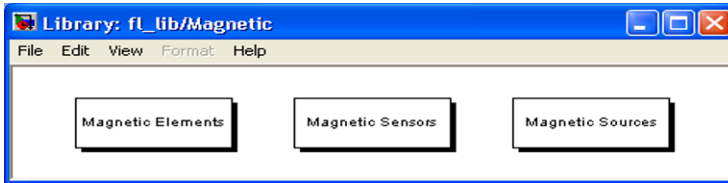
Signaux de consigne, sortie et de commande limitée entre -0.5 et +0.5



Si on limite le signal de commande à  $\pm 0.5$ , comme précédemment, on remarque que la sortie met plus de temps pour rejoindre, en régime statique, le signal de consigne.


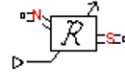
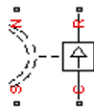
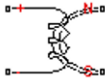

### I.3. Magnetic

Cette librairie concerne le domaine magnétique et comporte, comme certaines autres librairies étudiées précédemment, les éléments, les sources et les capteurs magnétiques.

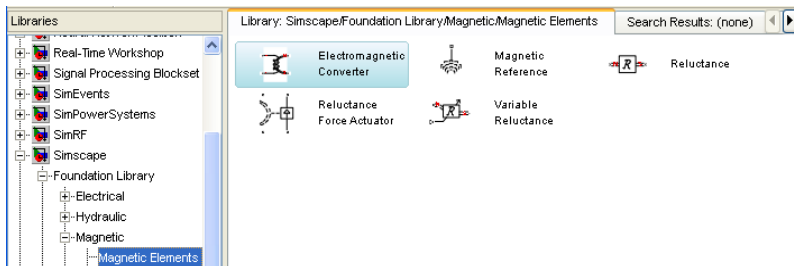


#### I.3.1. Magnetic Elements

Comme éléments magnétiques nous trouvons les blocs suivants.

Une réluctance	 Reluctance
Une réluctance variable	 Variable Reluctance
Un actionneur	 Reluctance Force Actuator
Un convertisseur électromagnétique	 Electromagnetic Converter
Une référence (masse) magnétique	 Magnetic Reference

Cette librairie se présente comme suit dans le browser de Simulink.



La réluctance est homogène à la résistance pour les circuits électriques. Elle quantifie la résistance (aptitude) d'un circuit magnétique à s'opposer à sa traversée par un champ magnétique.

Pour un circuit magnétique homogène (un seul matériau, section homogène), la réluctance dépend de la nature du matériau (perméabilité magnétique  $\mu$  et de ses dimensions, longueur et section).

La réluctance est définie par l'expression suivante, pour un circuit magnétique homogène, en fonction du matériau qui le constitue et de ses dimensions :

$$\mathfrak{R} = \frac{1}{\mu} \frac{l}{s}$$

$l$  : longueur en m,

$s$  : section en  $\text{m}^2$ ,

$\mu$  : perméabilité magnétique en  $\text{kg.m.A}^{-2}.\text{s}^{-2}$ .

L'unité de la réluctance est le  $\text{H}^{-1}$  (H étant le Henry).

Le bloc `Reluctance` modélise une réluctance magnétique, soit un composant qui résiste au flux magnétique comme une résistance résiste à un courant électrique.

C'est le rapport de la force magnétomotrice sur le flux résultant qui traverse le composant.

### ***1.3.2. Magnetic Sources***

Cette librairie contient 2 paires de types de blocs : source contrôlée de flux magnétique et source de flux, en Weber (Wb), ainsi qu'une source contrôlée de force magnétomotrice et une source de force magnétomotrice (MMF).

Le Weber (symbole Wb) est l'unité dérivée de flux d'induction magnétique du système international (SI).

Le Weber est le flux d'induction magnétique qui, traversant un circuit d'une seule spire, y produit une force électromotrice de 1 Volt .

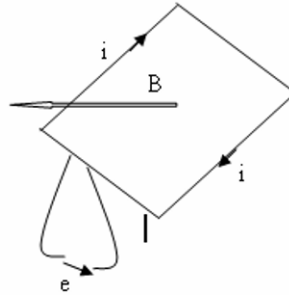
$1 \text{ Wb} = 1 \text{ T} \cdot \text{m}^2$ , soit le produit du champ magnétique B par la surface traversée.

La loi de Lenz montre que si une variation de flux  $d\phi(t)$  apparaît dans un cadre constitué d'un conducteur électrique, une force électromotrice  $e(t)$  apparaît à ses bornes.

Cette f.e.m s'oppose à la variation de flux, d'où le signe moins dans la formule suivante,  $\phi(t)$  étant le flux magnétique.

$$e(t) = - \frac{d\phi(t)}{dt}$$

Le Weber peut s'exprimer aussi en V.s, si  $e(t)$  est en Volts et le temps en secondes.



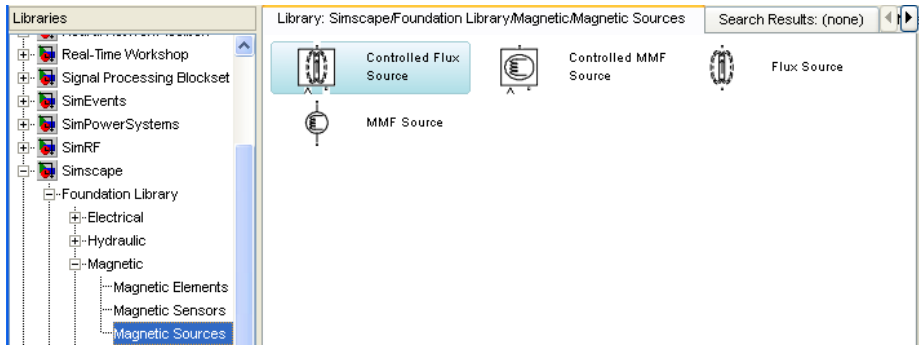
La variation d'un champ magnétique  $B$  dans un cadre conducteur crée un courant et l'apparition d'une différence de potentiel  $e$  aux bornes des extrémités de ce cadre.

Le flux magnétique que traverse ce cadre de section  $S$  est alors :

$$\phi(t) = B(t)S$$

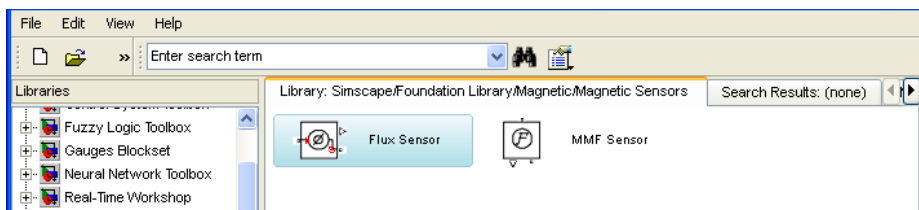
Une force magnétomotrice (MMF, Ampère) est toute force motrice qui peut produire un flux magnétique.

Dans le browser de Simulink, la librairie *Magnetic Sources* se présente comme suit :



### 1.3.3. Magnetic Sensors

Comme pour certaines librairies étudiées, il y a 2 sortes de capteurs : un capteur de flux et un capteur de force magnétomotrice.



1.3.4. Applications des circuits magnétiques

L'exemple suivant utilise un convertisseur électromagnétique qui sert d'interface entre le coté électrique possédant un certain nombre de spires  $N$  et la partie magnétique définie par ses pôles Nord et Sud. Si  $I$  et  $V$  sont les courants parcourant le coté électrique et  $N$  le nombre de spires (number of electrical winding turns), le flux  $\phi(t)$  et la MMF (force magnétomotrice) sont donnés par :

$$\begin{aligned} \text{MMF} &= N \cdot I \\ V &= -N \frac{d\phi(t)}{dt} \end{aligned}$$

Dans le modèle suivant, nous utilisons un convertisseur électromagnétique auquel on applique une tension sinusoïdale à son coté primaire avec un nombre de spires  $N=2000$ .

Le secondaire est relié à la réluctance  $\mathfrak{R}$ .

Par analogie avec les circuits électriques, la MMF (A) correspond à une tension (V) et la réluctance ( $H^{-1}$ ) équivaut à une résistance ( $\Omega$ ). L'équivalence de la loi d'Ohm  $U = R I$  s'écrit, dans le domaine magnétique par :  $F = \mathfrak{R} \phi$ .

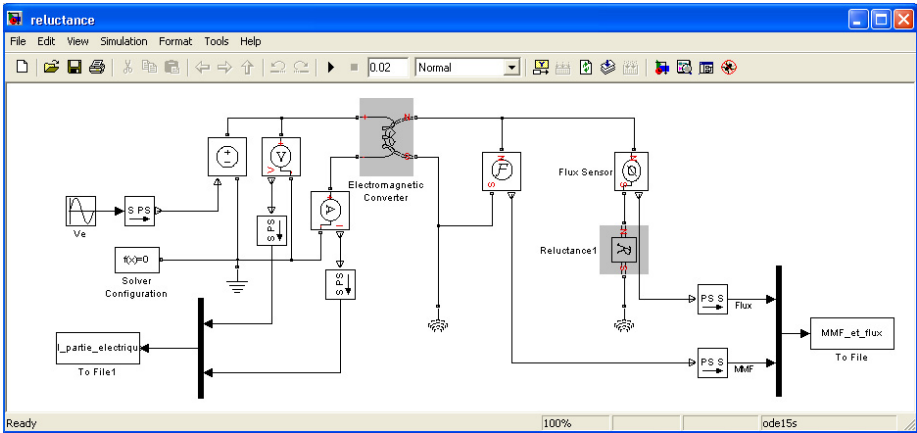
Le tableau suivant montre quelques équivalences entre les domaines, électrique et magnétique.

Paramètre électrique	Paramètre équivalent magnétique
Intensité électrique $I$ (A)	Flux magnétique $\phi$ (Wb)
Tension électrique (V)	Force magnétomotrice $F$ (A)
Résistance $R$ ( $\Omega$ )	Réluctance $\mathfrak{R}$ ( $H^{-1}$ )
Conductivité électrique $\delta$ (S/m)	Perméabilité magnétique $\mu$ (H/m)

• Réluctance, flux et MMF

Dans le modèle suivant, nous étudions un système composé d'un convertisseur électromagnétique auquel on branche, sur le coté magnétique, une réluctance.

On mesure le flux qui la traverse par un capteur de flux, la force magnétomotrice par le capteur MMF sensor, le flux par le capteur Flux Sensor qu'on envoie dans le fichier binaire MMF\_et\_flux.mat, le courant  $I$  et la tension  $V$  du circuit primaire qu'on envoie dans le fichier binaire VI\_primaire.mat à travers un multiplexeur.



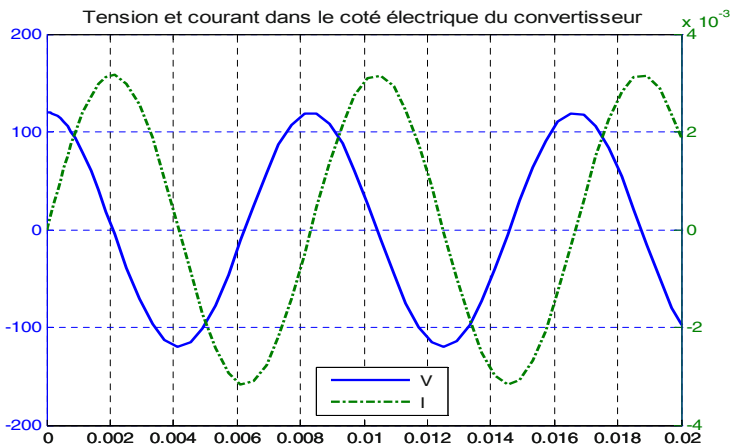


La figure suivante représente les courbes de tension et courant du circuit primaire du convertisseur électromagnétique.

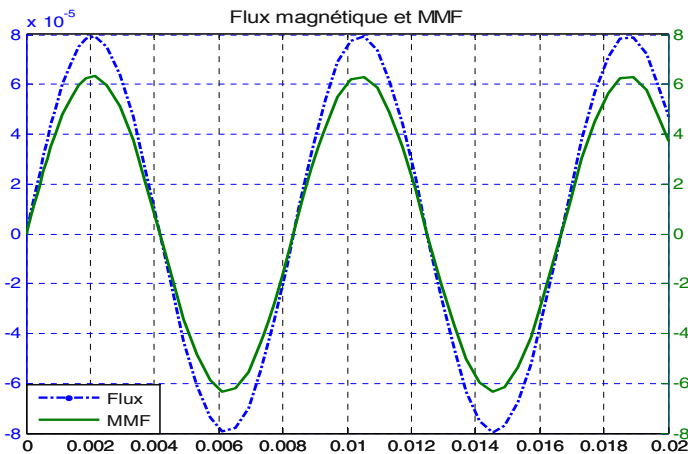
On retrouve bien la tension sinusoïdale, d'amplitude  $120\sqrt{2}$  V, appliquée au primaire.

Les figures sont tracées grâce aux lignes de commande suivantes, programmées dans la fonction Callback StopFcn.

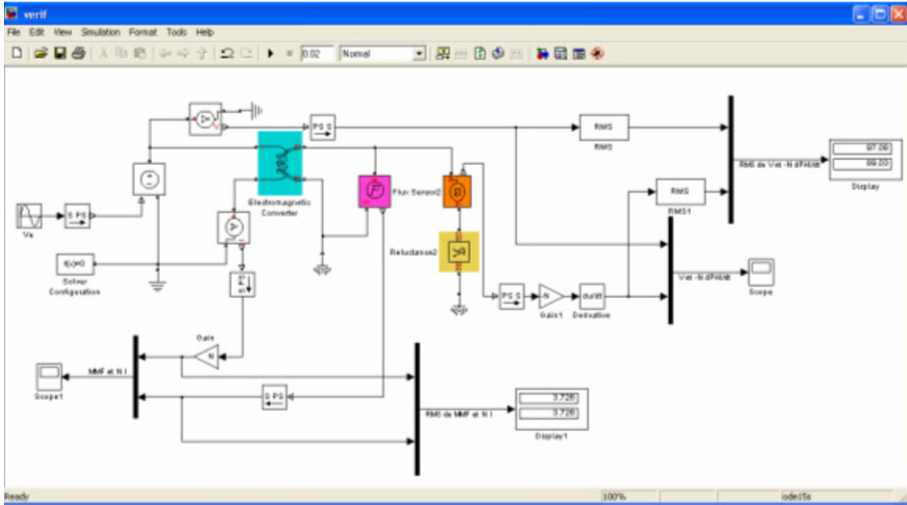
```
close all, load VI_primaire.mat
plotyy(x(1,:),x(2,:),x(1,:),x(3,:)), grid
title('Tension et courant au primaire du convertisseur')
figure, load MMF_et_flux.mat
plotyy(y(1,:),y(2,:),y(1,:),y(3,:)), grid
title('Flux magnétique et MMF')
```



Le flux traversant la réluctance et la MMF sont donnés par les figures suivantes.



Nous comparons et vérifions les formules théoriques, à savoir les signaux MMF et N.I. d'une part et  $V$  et  $-N \frac{d\phi(t)}{dt}$  d'autre part.

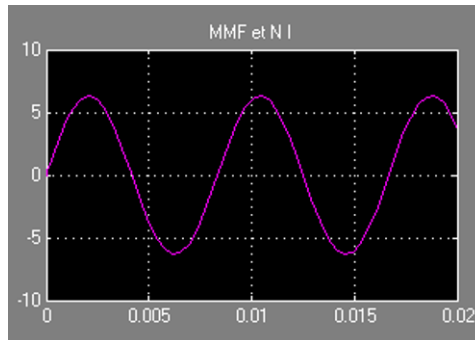


On utilise le bloc RMS de l'outil Signal Processing Blockset de la bibliothèque Statistics.

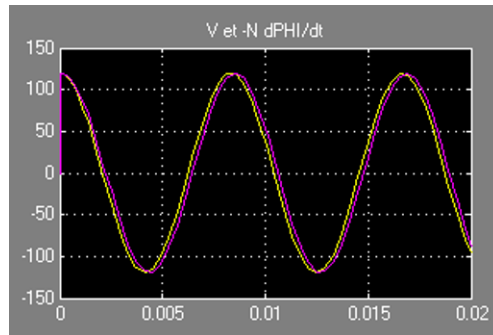
Ce bloc calcule la moyenne des carrés des valeurs d'un signal.

Si le signal est centré (moyenne nulle), la valeur obtenue correspond à la variance de ce signal.

Les signaux MMF et N.I. sont confondus dans l'oscilloscope et leurs valeurs RMS sont parfaitement égales à 3.726.



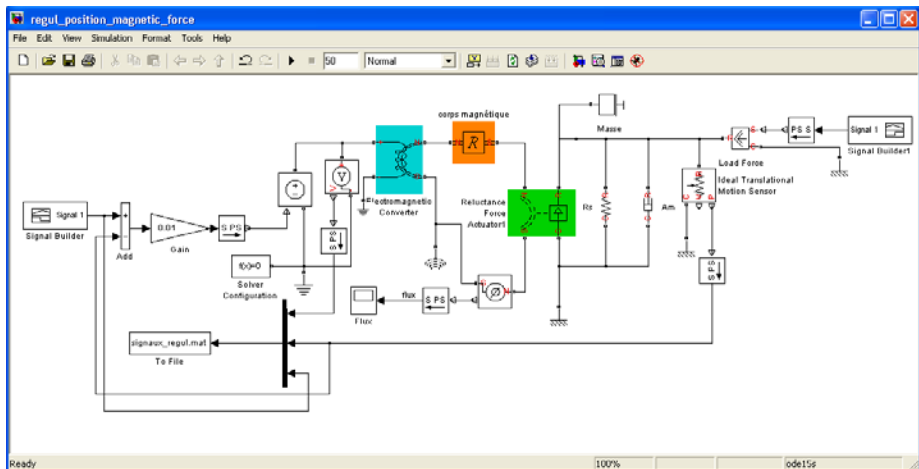
Celles de  $V$  et  $-N \frac{d\phi(t)}{dt}$  sont légèrement différentes comme le montre leur oscilloscope.



### ✱ Régulation de position d'une masse avec le bloc « Reluctance Force Actuator »

Dans le modèle suivant, on réalise une régulation de la position d'une masse de 1 kg reliée à un amortisseur et un ressort. Comme on régule une position, le système possède une intégration et un simple régulateur P proportionnel suffit.

La masse est mue par l'actionneur électromagnétique Reluctance Force actuator.



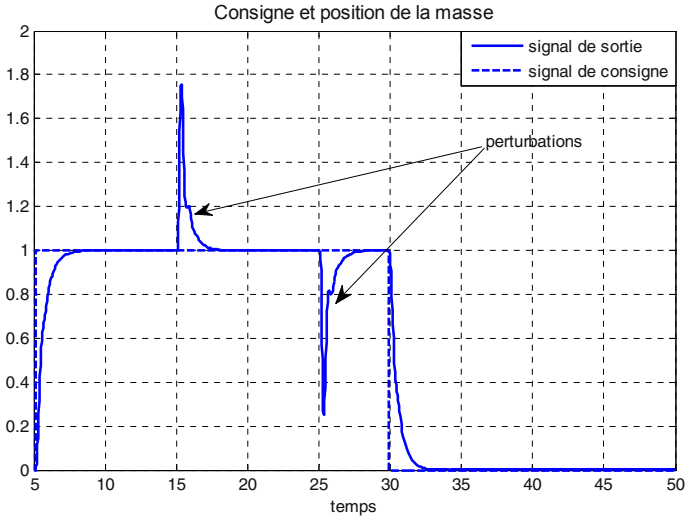
Nous appliquons des perturbations de force aux instants  $t=15$  et  $25$ . Ces perturbations sont rejetées rapidement.

Les signaux sont tracés dans la fonction Callback `StopFcn` où sont programmées les lignes de commandes suivantes :

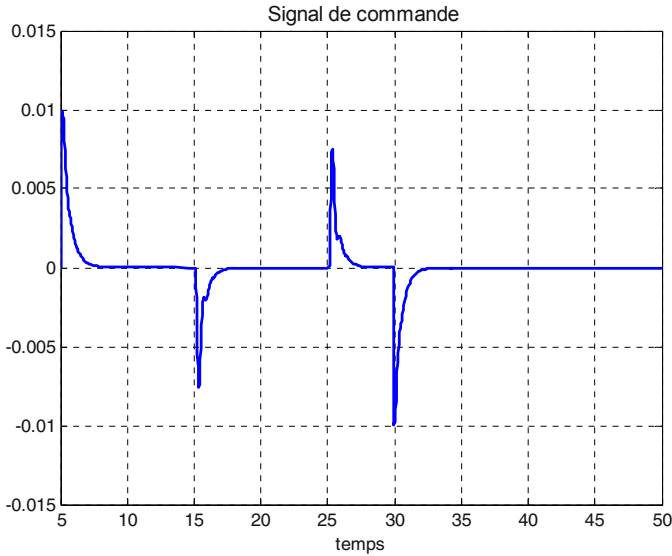
```
close all, load signaux_regul.mat
plot(xyz(1,:),xyz(3,:)), hold on
plot(xyz(1,:),xyz(4,:))
axis([5 50 0 2]), xlabel('temps'), grid
title('Consigne et position de la masse')
figure
plot(xyz(1,:),xyz(2,:)), grid
```

```
title('Signal de commande')
axis([5 50 -0.015 0.015]), xlabel('temps')
```

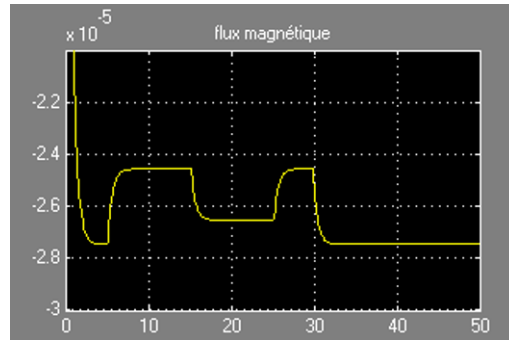
Avec la valeur du gain choisi, les perturbations sont rejetées sans dépassement.



Le signal de commande est toujours nul sauf au moment de rejeter les perturbations ou du changement du signal de consigne (fronts montants ou descendants).



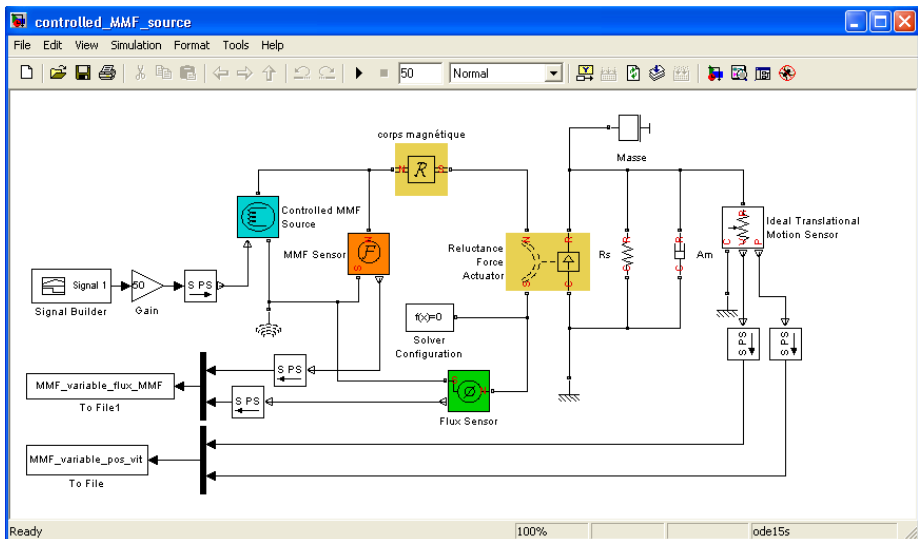
La variation correspondante de flux est donnée par l'oscilloscope suivant.



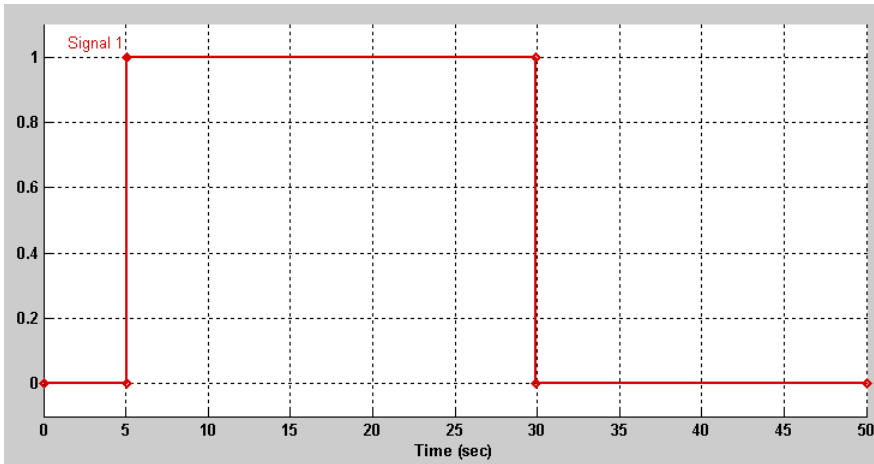
• *Utilisation de la source contrôlée de MMF (force magnétomotrice)*

Dans le modèle suivant, nous utilisons une source contrôlée de force magnétomotrice (MMF) qu'on applique, à travers la réluctance  $\mathcal{R}$ , à la partie magnétique d'un actionneur linéaire. Cet actionneur déplace une masse soumise à un amortisseur et rattachée à un ressort. La source est contrôlée par le signal issu du bloc Signal Builder que l'on multiplie par 50.

Nous mesurons cette MMF par le capteur MMF Sensor ainsi que le flux qui traverse la réluctance  $\mathcal{R}$  et la partie magnétique de l'actionneur, par le capteur de flux magnétique Flux Sensor. Ces deux signaux sont envoyés vers un fichier binaire via un multiplexeur. Il en est de même pour la position et la vitesse linéaire de la masse.



Le signal fourni par le bloc `Signal Builder` que l'on applique à la source contrôlée de force magnétomotrice est le suivant.



Le tracé des signaux se fait par la fonction `Callback StopFcn` dans laquelle sont programmées les commandes suivantes.

```
load MMF_variable_pos_vit.mat

close all
clear all

% Tracé des signaux de position et de vitesse
plotyy(x(1,:),x(2,:),x(1,:),x(3,:))
grid
title('Position et vitesse de la masse')

figure
load MMF_variable_flux_MMF.mat

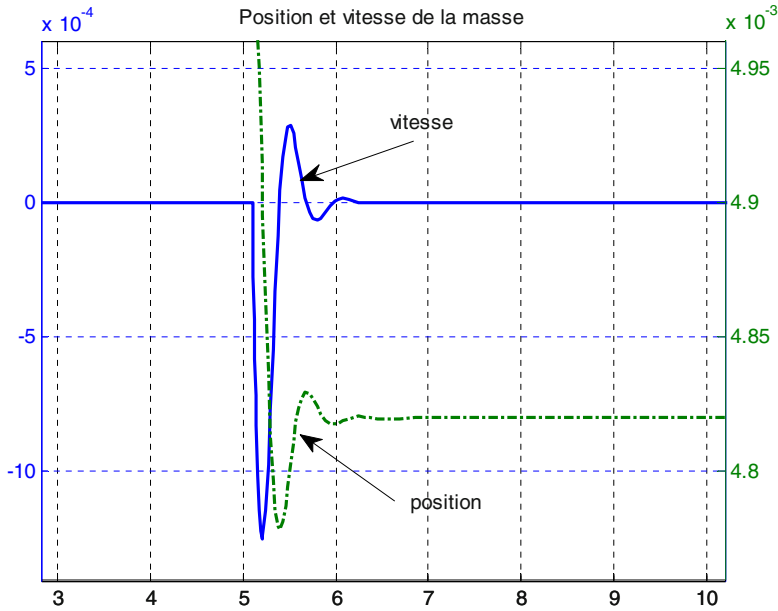
% Tracé des signaux de MMF et du flux magnétique
plotyy(y(1,:),y(2,:),y(1,:),y(3,:))
grid
title('Flux magnétique et MMF')

delete *.mat
```

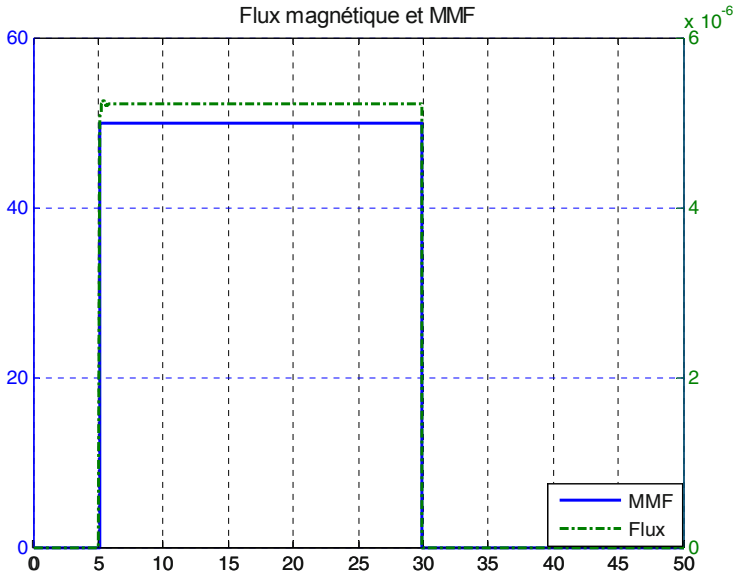
Les signaux de la position et de la vitesse linéaires de la masse sont donnés par la figure suivante.

Selon la dynamique du système et celle du signal de commande appliqué, la réponse en boucle ouverte, en position ou en vitesse, réagit avec des dépassements que l'on observe mieux en faisant un zoom (figure suivante).

Si l'on fait un zoom sur la partie transitoire, la position et la vitesse subissent 4 dépassements avant de se stabiliser.



La MMF et le flux sont tous deux des créneaux qui sont tracés dans la même figure, dans 2 axes d'ordonnées différents, grâce à la commande `plotyy`.



## I.4. Mechanical

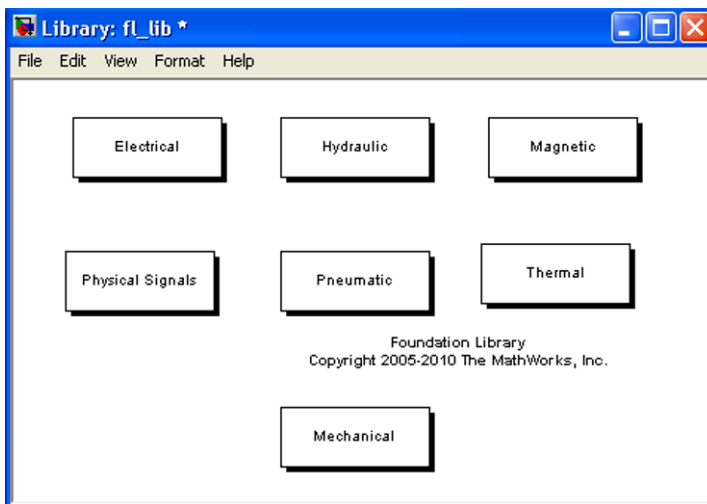
### I.4.1. Eléments des différentes bibliothèques

Cette librairie de mécanique, comporte, comme celle de thermique :

- Des capteurs (Mechanical Sensors)
  - capteur de force (Ideal Force Sensor),
  - capteur de couple (Ideal Torque Sensor),
  - capteur de vitesse et de position (mouvement de translation, Ideal Translational Motion Sensor et de rotation, Ideal Rotational Sensor).
- Des sources mécaniques
  - source de vitesse, angulaire (Ideal Angular Velocity Source), de translation (Ideal Translational Velocity Source),
  - source de force (Ideal Force source),
  - source de couple (Ideal Torque Source).
- Des éléments mécaniques (de translation, Translational Elements et de rotation, Rotational Elements). Ces éléments comportent des ressorts, des frottements, une inertie ou une masse, un amortisseur, un limiteur de mouvement et une masse de référence.

En plus de la librairie de thermique, celle-ci comporte les mécanismes suivants :

- boîte de vitesse (Gear Box),
- un levier (Lever),
- une roue avec axe (Wheel and Axle) qui permet de transformer un mouvement de rotation en mouvement linéaire.

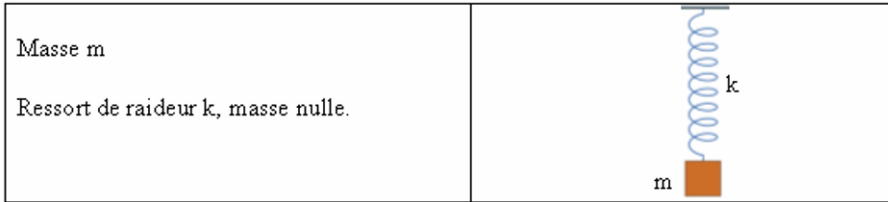




### 1.4.2. Applications

#### • Système masse – ressort sans frottement

On considère le système mécanique suivant composé d'une masse  $m$  accrochée à un ressort de raideur  $k$  dont l'autre extrémité est reliée à un support fixe.



C'est un mouvement de translation qui peut être horizontal ou vertical et identique en considérant que dans chacun des deux cas, il n'y a pas de frottement. On obtient des oscillations harmoniques.

Dans le cas général, le mouvement de la masse est dû à 3 forces :

- une force de rappel  $F_r$ ,
- une force d'amortissement  $F_a$ ,
- une force extérieure  $F_e$ .

Si  $x$  est la position de la masse  $m$  à partir de sa position d'équilibre, l'équation du mouvement s'écrit :

$$m \ddot{x} = F_r + F_a + F_e$$

La force de rappel est une fonction de la position qui varie en sens inverse de l'excursion, tout en passant par l'origine lorsque les excursions sont comptées à partir du point d'équilibre, soit :  $F_r = -kx$ .

L'amortissement dépendant de la vitesse est considéré linéaire, soit  $F_a = -f \dot{x}$  avec  $f$  la constante de raideur du ressort.

L'équation du mouvement donne :

$$m \ddot{x} + f \dot{x} + kx = F_e$$

Dans le cas de notre exemple, dans lequel on suppose qu'il n'y a pas de force extérieure ni d'amortissement, l'équation du mouvement se réduit à :

$m \ddot{x} + kx = 0$ , dont la solution est, selon les équations d'Euler :

$$x = a \cos\left(\sqrt{\frac{k}{m}} t\right) + b \sin\left(\sqrt{\frac{k}{m}} t\right),$$

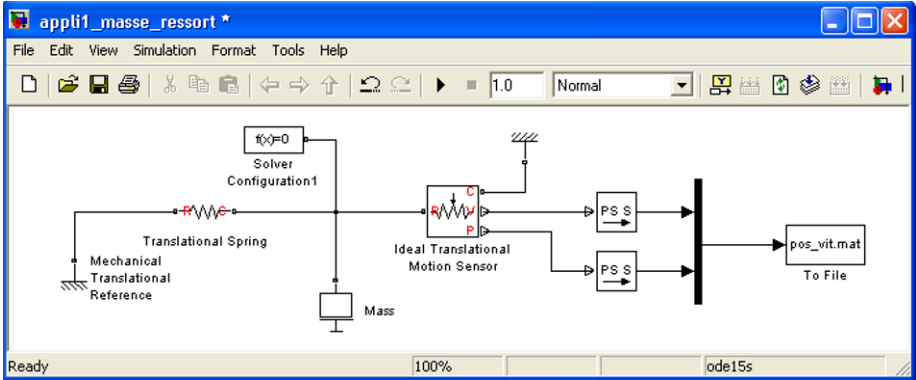
En posant  $a = A \cos \varphi$ ,  $b = -A \sin \varphi$ , on obtient :

$$x = A \cos\left(\sqrt{\frac{k}{m}} t + \varphi\right).$$

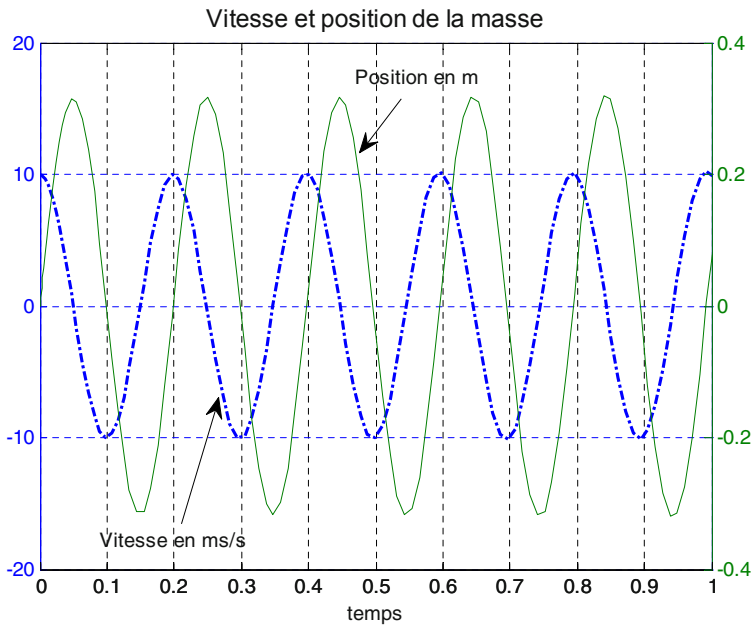
C'est un signal sinusoïdal d'amplitude  $A$ , de phase  $\varphi$  et de pulsation  $\omega_0 = \sqrt{\frac{k}{m}}$ , soit une période  $T = 2\pi \sqrt{\frac{m}{k}}$ .

Avec  $m = 1 \text{ kg}$  et  $k = 1000 \text{ N/m}$ , la période vaut :  $T = 2\pi \sqrt{10^{-3}} = 0.1987 \text{ s}$ .

Nous simulons ce système par le modèle `appli1_masse_ressort.mdl`, dans lequel nous avons spécifié une vitesse initiale de  $10 \text{ m/s}$  qui deviendra la limite maximale en valeur absolue.



Les oscillations harmoniques de la position et de la vitesse sont données par la figure suivante.



Nous vérifions bien, dans la figure suivante la valeur de la période  $T \approx 0.2 \text{ s}$ .

• *Système masse – ressort avec amortisseur*

Nous avons ajouté au système précédent, un amortissement de coefficient  $f = 10 \text{ N} / (\text{m} / \text{s})$ . Dans le cas réel, le système est régi par l'équation différentielle :

$$m \ddot{x} + f \dot{x} + k x = 0$$

Son équation caractéristique est :

$$m r^2 + f r + k = 0$$

Le déterminant de cette équation étant :

$$\Delta = f^2 - 4 m k .$$

Si ce déterminant est négatif, on a deux racines complexes conjuguées qui conduisent à une solution oscillante mais amortie.

La sortie du système oscille en tendant vers zéro, avec la pulsation  $w_0 = \sqrt{\frac{k}{m}}$ .

Le coefficient d'amortissement est :

$$\zeta = \frac{1}{2} \frac{f}{k} w_0 = \frac{1}{2} \frac{f}{k} \sqrt{\frac{k}{m}} = \frac{1}{2} f \sqrt{\frac{1}{k m}}$$

Pour obtenir la réponse impulsionnelle de ce système, nous utilisons la transformée de Laplace de la fonction de transfert de ce système.

$$\begin{aligned} H(p) &= \frac{\frac{1}{k}}{\frac{m}{k} p^2 + \frac{f}{k} p + 1} \\ &= \frac{K}{1 + \frac{2\zeta}{w_0} p + \frac{1}{w_0^2} p^2} \end{aligned}$$

avec :

- $K$  : gain du système,
- $\zeta$  : coefficient d'amortissement,
- $w_0$  : pulsation propre.

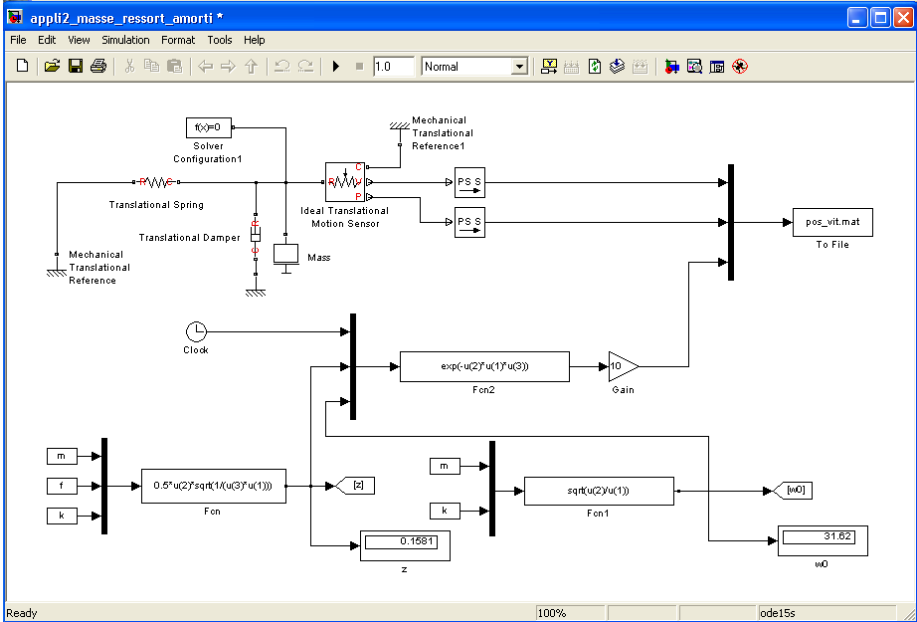
La réponse impulsionnelle est donnée par :

$$s(t) = \frac{K w_0}{\sqrt{\zeta^2 - 1}} e^{-\zeta w_0 t} \sin(w_0 \sqrt{1 - \zeta^2} t + \theta)$$

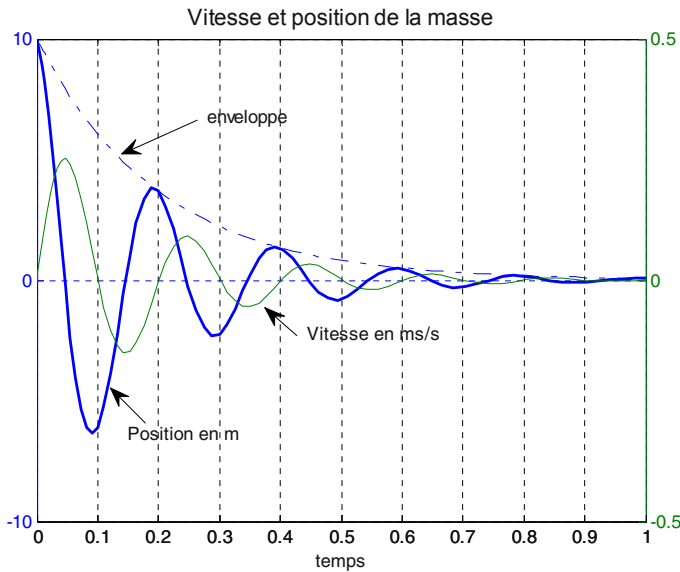
La réponse est pseudo-périodique. Elle comporte des oscillations dont la période, appelée pseudo-période est :

$$T = \frac{2\pi}{w_0 \sqrt{1 - \zeta^2}}$$

Dans le modèle suivant, nous représentons le système mécanique précédent ainsi que la solution théorique que nous avons obtenue.



Il est impératif d'utiliser l'amortisseur et un ressort de la bibliothèque des éléments de translation ainsi que la référence (masse) correspondante. Nous obtenons une position et une vitesse qui décroissent vers 0 avec la même pseudo-période de 0.2s. Dans la figure suivante, nous traçons la vitesse et la position avec 2 ordonnées différentes.



La courbe en pointillés représente l'enveloppe de décroissance du signal de vitesse évoluant selon  $v_0 e^{-\zeta \omega_0 t}$  avec  $v_0$  la vitesse initiale, égale à 10 comme on l'a spécifié pour la masse.

Parameters

Mass:

kg

Initial velocity:

m/s

Dans ce modèle, on calcule le coefficient d'amortissement dans la fonction `Fcn`, la pulsation  $w_0$  par `Fcn1`.

La fonction `Fcn2` calcule l'expression de l'enveloppe, soit  $e^{-\zeta w_0 t}$ .

Dans la fonction Callback `StopFcn`, nous avons tracé les différentes courbes par les lignes de commandes suivantes :

```
load pos_vit.mat
plotyy(pv(1,:),pv(2,:),pv(1,:),pv(3,:))
gtext('Vitesse en ms/s'), gtext('Position en m')
title('Vitesse et position de la masse')
xlabel('temps')
grid
hold on
plot(pv(1,:),pv(4,:))
gtext('enveloppe')
```

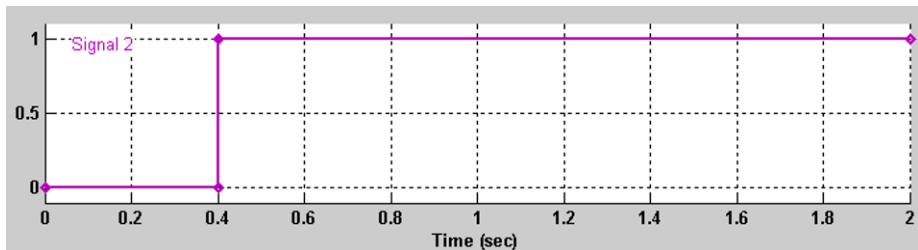
Dans `InitFcn`, nous avons spécifié les valeurs de la masse, le coefficient de raideur du ressort ainsi que le coefficient d'amortissement.

```
f=10;
m=1;
k=1000;
```

#### • Système de rotation

Dans le modèle suivant, nous étudions 2 exemples, l'un de translation et l'autre de rotation d'un système équivalent.

Nous appliquons un couple sous forme de l'échelon suivant avec lequel on commande la source contrôlée de couple `Ideal Torque Source`.

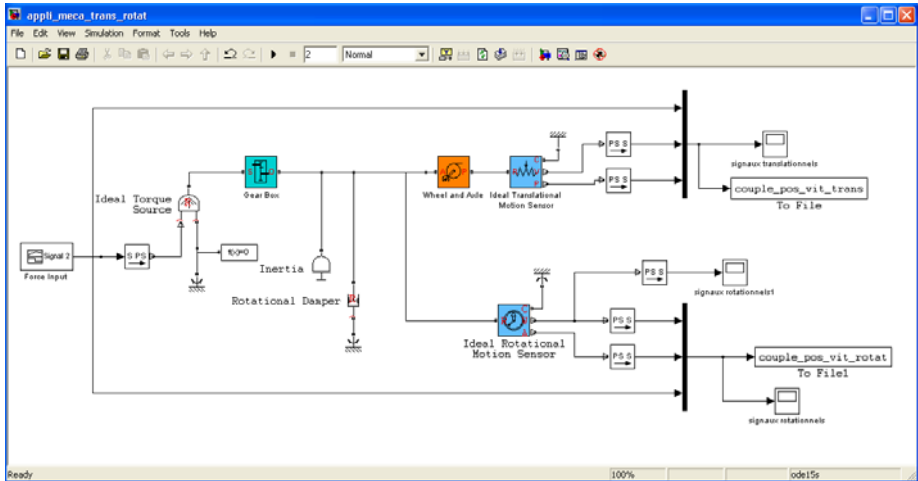


On utilise, ensuite, un réducteur de vitesse de rapport 5, ce qui consiste à obtenir, en sortie, un couple 5 fois plus grand qu'à l'entrée, mais à une vitesse d'autant plus réduite.

Le réducteur de vitesse est utilisé comme multiplicateur de couple.

Ce couple est appliqué à un système formé d'une inertie, *Inertia*, un amortisseur, *Rotational Damper* et un ressort de torsion, *Rotational Spring*.

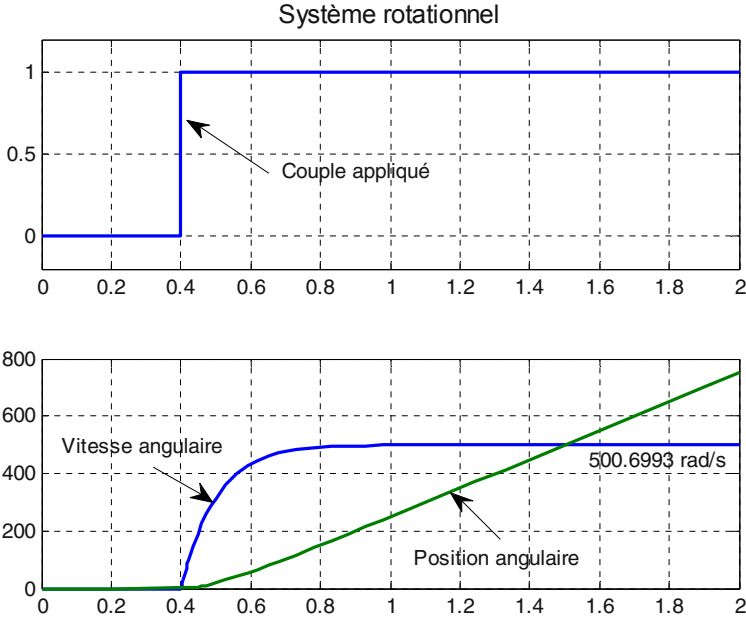
Ce mouvement de rotation est transformé en mouvement linéaire grâce au bloc *Wheel and Axle*. La vitesse et la position linéaire, obtenues à la sortie du capteur *Ideal Translational Motion Sensor*, sont envoyées simultanément vers un oscilloscope et le fichier binaire `couple_pos_vit_trans.mat`.



Dans la partie basse de ce modèle, on utilise un capteur de mouvement de rotation, *Ideal Rotational Motion Sensor*, pour obtenir la vitesse et la position angulaires que l'on envoie vers un oscilloscope et le fichier binaire `couple_pos_vit_rotat.mat`.

Dans la fonction Callback `StopFcn`, nous lisons les fichiers binaires et traçons les différentes courbes.

```
% Lecture des 2 fichiers binaires
close all, clc
load couple_pos_vit_rotat.mat
load couple_pos_vit_trans.mat
subplot 211
plot(x(1,:),x(4,:)), axis([0 2 -0.2 1.2])
title('Système rotationnel')
gtext('Couple appliqué'), grid
subplot 212
plot(x(1,:), x(2,:),x(1,:),x(3,:)), grid
gtext('Position angulaire')
gtext('Vitesse angulaire')
figure
plot(y(1,:), y(3,:),y(1,:),y(4,:))
title('Système linéaire'), grid
gtext('Position linéaire'), gtext('Vitesse linéaire')
```



#### • *Système de translation*

Le mouvement linéaire est dérivé du système de rotation qui entraîne une roue et un axe (Wheel and Axle).

Si le rayon de cette roue est noté  $r$ , la vitesse linéaire est donnée par :

$$V = w r$$

avec  $w$  : la vitesse angulaire de la roue.

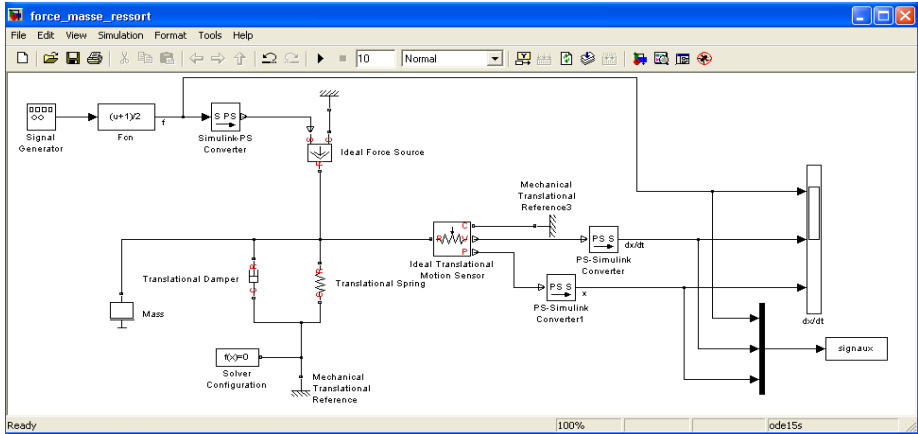
De même que pour le système de rotation, on mesure la vitesse et la position linéaire par le capteur `Ideal Translational Motion Sensor`.

#### • *Force appliquée à un système masse-ressort*

Dans le modèle suivant, nous appliquons une force sous forme d'un signal rectangulaire de fréquence 0.2 Hz d'amplitude 1.

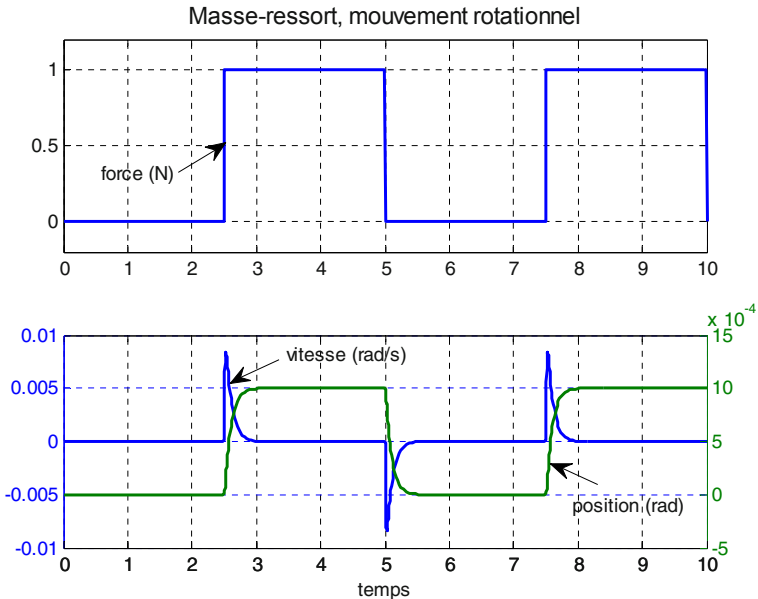
Le bloc `Fcn, (u+1)/2`, permet de ramener le signal qui varie entre -1 et 1, vers des valeurs positives comprises entre 0 et 1.

Ce signal contrôle une source de force par le bloc `Ideal Force Source`. On applique ainsi au système, une force de 1 N, sous la forme d'un signal carré de fréquence 0.2 Hz.



Dans la fonction Callback `StopFcn`, nous traçons les courbes de la force appliquée ainsi que les variations de la vitesse et de la position angulaire. La variable `x` correspond à celle qui contient les signaux sauvegardés dans le fichier binaire `signaux.mat`.

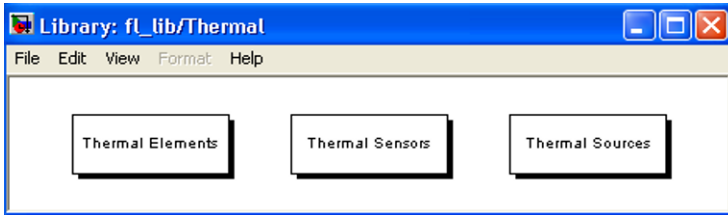
```
load signaux.mat
subplot 211, plot(x(1,:),x(2,:))
title('Masse-ressort, mouvement rotationnel'), grid
axis([0 10 -0.2 1.2])
subplot 212
plotyy(x(1,:),x(3,:),x(1,:),x(4,:))
grid
xlabel('temps')
```





## I.5. Thermal

Cette librairie possède les 3 bibliothèques suivantes :

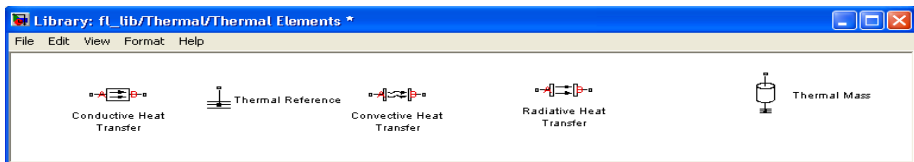


- Thermal elements ou éléments thermiques (masse thermique, différents types de transfert thermiques, masse et référence thermique),
- Thermal sensors ou capteurs de température,
- Thermal Sources ou sources thermiques.

### I.5.1. Thermal Elements

Cette bibliothèque contient les éléments thermiques, comme :

- la masse thermique,
- les coefficients d'échange thermique par rayonnement, conduction et convection,
- la masse thermique ou référence des circuits thermiques.



La librairie Thermal de Foundation Library comporte les 3 bibliothèques des sources, éléments et capteurs thermiques.

Les 2 sources de température et de flux thermique doivent être commandées par un signal issu de Simulink (échelon, etc. ...) qui doit être ensuite converti en signal physique de Simscape.

Parmi les éléments, nous trouvons une masse thermique qui symbolise l'inertie thermique, les coefficients d'échange par conduction, convection et rayonnement. Parmi ces éléments, nous trouvons la référence ou masse pour les signaux des circuits modélisant les systèmes thermiques.



- **Masse thermique**

La masse thermique symbolise l'inertie thermique d'un matériau quelconque.

Une boule de grande masse, à l'inverse d'une petite bille, sera difficile à mettre en mouvement, mais une fois lancée, elle sera difficile à arrêter.

Ceci est dû à l'inertie propre à sa masse.

L'inertie thermique, de la même manière, est liée à la « masse thermique » des matériaux. Certains matériaux sont difficiles à monter en température que d'autres comme le métal, mais une fois chauffés, il est plus difficile de les refroidir.

Le bloc de masse thermique représente une masse thermique, qui reflète la capacité d'un matériau ou une combinaison des matériaux à stocker l'énergie interne.

Cette propriété est caractérisée par la masse du matériau et de sa chaleur spécifique. La masse thermique est décrite avec l'équation suivante :  $Q = m c \frac{dT}{dt}$ .

avec :

- Q : flux thermique,
- c : chaleur spécifique du matériau,
- m : masse du matériau,
- T : température du matériau,
- t : temps.

Ce bloc est défini par la boîte de dialogue dans laquelle, on spécifie la masse, la chaleur spécifique et la température initiale du matériau.

Parameters

Mass:	1	kg
Specific heat:	447	J/kg/K
Initial temperature:	300	K



Conductive Heat Transfer

#### • Echange par conduction

Le bloc de transfert de chaleur par conduction représente le transfert par conduction entre 2 couches du même matériau.

Le transfert est régi par la loi de Fourier définie par l'équation suivante :  $Q = k \frac{A}{D} (T_A - T_B)$

où :

- Q : flux de chaleur,
- K : conductivité thermique du matériau,
- A : aire ou surface d'échange, normale à la direction du flux de chaleur,
- D : épaisseur des couches,
- $T_A, T_B$  : températures des couches A et B.

Le flux est considéré positif de A vers B. Sa boîte de dialogue est la suivante :

Parameters

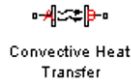
Area:	1e-4	m <sup>2</sup>
Thickness:	0.1	m
Thermal conductivity:	401	W/(m*K)

Area : surface d'échange de chaleur, normale à la direction d'écoulement de la chaleur.

La valeur par défaut est 0.0001 m<sup>2</sup>,

Thickness : épaisseur des couches. La valeur par défaut est de 0.1 m,

Thermal conductivity : conductivité thermique, valeur par défaut 401 W/(m\*K)



#### • Echange par convection

Le bloc de transfert par convection représente le transfert de chaleur entre deux corps, dont au moins un est fluide.

Ceci est représenté par la loi de Newton, décrite par l'équation :  $Q = k A (T_A - T_B)$

Q : flux de chaleur,

k : coefficient du transfert de chaleur par convection,

A : surface d'échange,

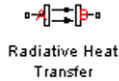
T<sub>A</sub>, T<sub>B</sub> : températures des 2 corps.

Lorsqu'on double-clique sur ce bloc, on obtient la boîte de dialogue dans laquelle on spécifie l'aire et le coefficient de transfert, dont les valeurs par défaut sont, respectivement de 10<sup>-4</sup> m<sup>2</sup> et 20 W/((m<sup>2</sup>\*K).

**Parameters**

Area:  m<sup>2</sup> ▼

Heat transfer coefficient:  W/(m<sup>2</sup> \* K) ▼



#### • Echange par rayonnement

Ce bloc de transfert par rayonnement représente le transfert entre deux surfaces de telle sorte que l'énergie, émise par un corps, est complètement absorbée par le corps récepteur.

Le transfert est régi par la loi de Stefan-Boltzmann et décrit par l'équation :

$$Q = k A (T_A^4 - T_B^4)$$

Q : débit de chaleur ou flux,

k : constante de Stefan-Boltzmann,

A : aire de transfert,

T<sub>A</sub>, T<sub>B</sub> : températures des 2 corps.

La boîte de dialogue de ce bloc est la suivante :

**Parameters**

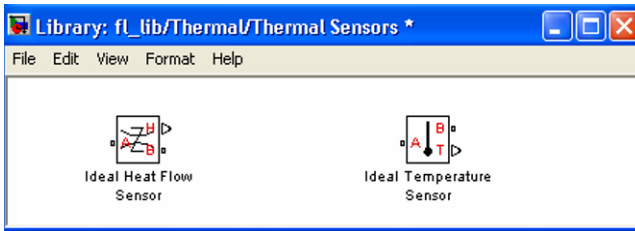
Area:  m<sup>2</sup> ▼

Radiation heat transfer coefficient:  W/(m<sup>2</sup>\*K<sup>4</sup>) ▼

On peut spécifier la surface qui est ici par défaut, de 10<sup>-4</sup> m<sup>2</sup>.

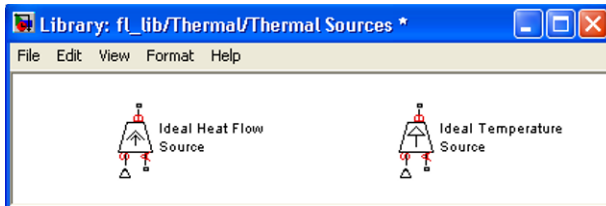
### I.5.2. Thermal Sensors

Nous avons 2 capteurs thermiques, un capteur de température et un capteur de flux thermique.



### I.5.3. Thermal Sources

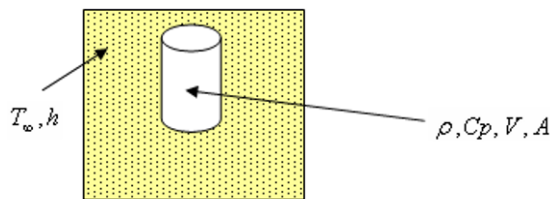
Comme pour les capteurs, nous avons 2 sources, une de température et une autre de flux thermique.



### I.5.4. Applications de thermique

#### ▲ Immersion d'un corps mince dans un fluide

Dans cet exemple, nous étudions le cas d'un cylindre métallique très mince à la température  $T_0 = 20^\circ\text{C}$  qu'on immerge dans un fluide à la température  $T_\infty = 100^\circ\text{C}$ .

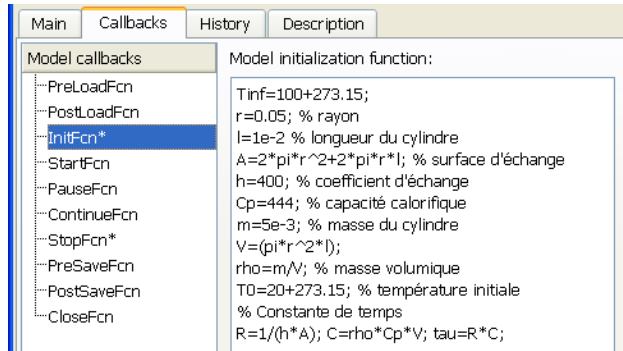


Le bilan thermique ou équation de la chaleur :  $Ah(T_\infty - T(t)) = \rho V Cp \frac{dT}{dt}$

Ceci donne :  $\frac{dT(t)}{dt} + \frac{Ah}{\rho Cp V} (T(t) - T_\infty) = 0$

En posant  $m = \frac{Ah}{\rho Cp V}$ , nous obtenons :  $\frac{T(t) - T_\infty}{T_0 - T_\infty} = \frac{\theta(t)}{\theta_0} = e^{-mt}$

Dans le modèle suivant, on définit les paramètres du cylindre ainsi que ceux du fluide, dans la fonction Callback `InitFcn`.



On a :

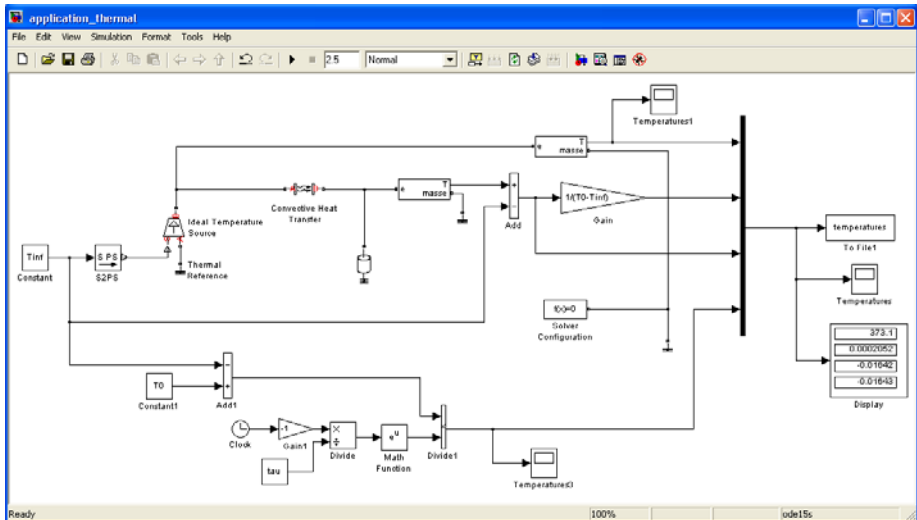
$$\frac{\theta(t)}{\theta_0} = e^{-mt} = e^{-t/\tau}$$

C'est un système du 1<sup>er</sup> ordre de constante de temps :

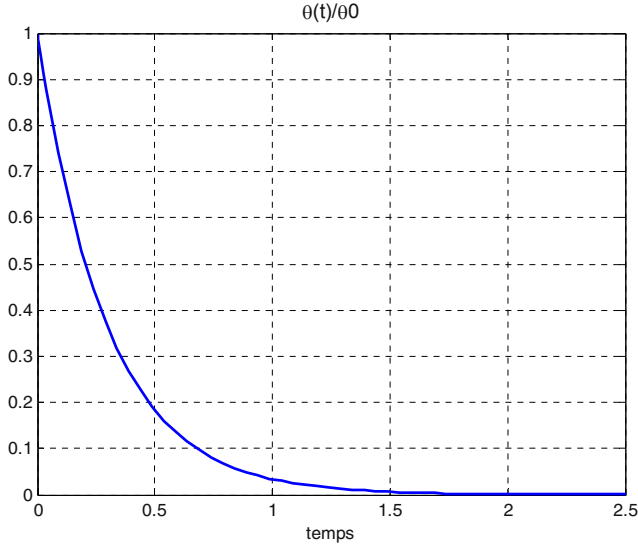
$$\tau = \frac{1}{Ah} \rho CpV.$$

On voit bien l'analogie de ce système et la décharge d'un condensateur dans une résistance dans le cas d'un circuit RC avec :

$$R = \frac{1}{Ah} \text{ et } C = \rho CpV.$$



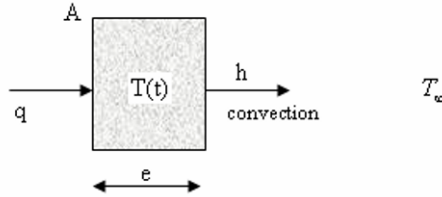
L'évolution, avec le temps, du rapport  $\frac{\theta(t)}{\theta_0}$  est donné par la courbe suivante.



• *Système à conditions mixtes*

On considère un solide de surface transversale  $A$  et d'épaisseur  $e$ , initialement à la température  $T_0$ .

Pour  $t > 0$ , on fournit de la chaleur à l'une des surfaces transversales ( $q$  en  $W/m^2$ ), l'autre étant en contact avec l'air ambiant à  $T_\infty$ .



Le bilan thermique s'écrit :

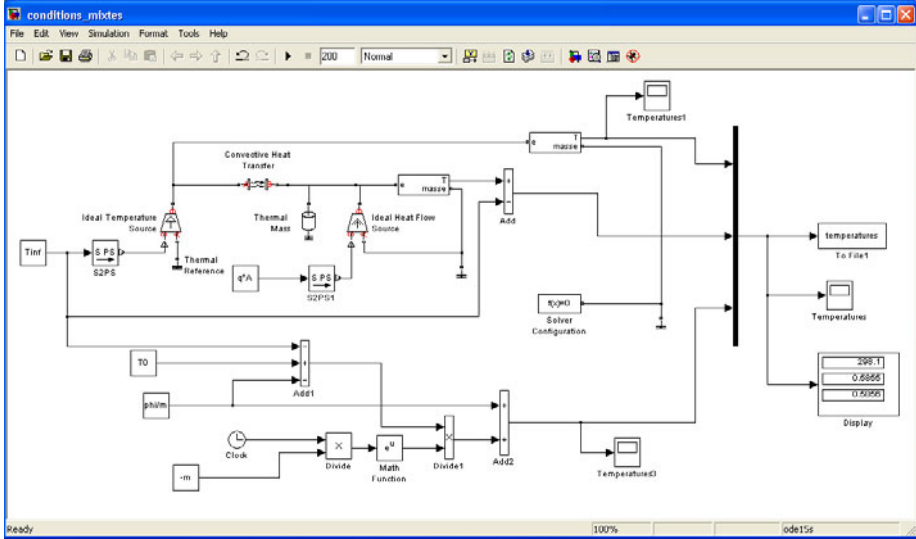
$$Aq - hA[T(t) - T_\infty] = \rho C_p A e \frac{dT}{dt}$$

En posant  $\theta(t) = T(t) - T_\infty$ ,

$$\frac{d\theta}{dt} + \frac{h}{\rho C_p e} \theta(t) = \frac{q}{\rho C_p e}$$

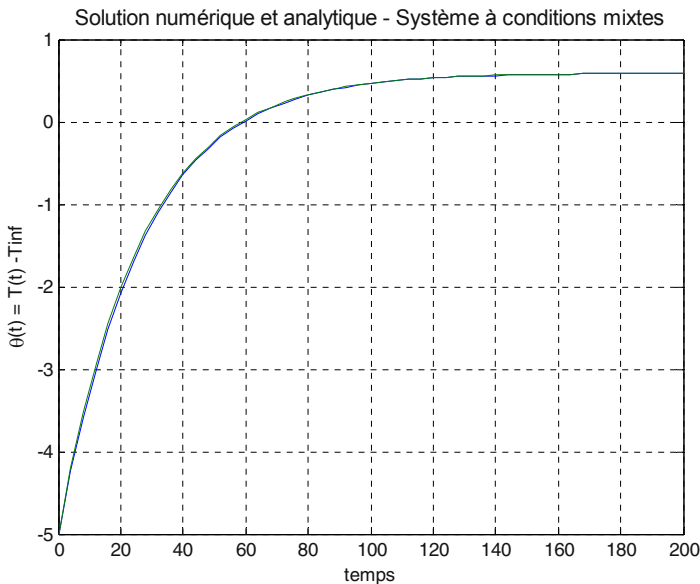
Avec  $m = \frac{h}{\rho C_p e}$  et  $\phi = \frac{q}{\rho C_p e}$ ,

la solution est donnée par :  $\theta(t) = \theta_0 e^{-mt} + \frac{\phi}{m}(1 - e^{-mt})$ , où  $\theta_0 = T_0 - T_\infty$ . Dans le modèle suivant, nous appliquons d'un côté la température  $T_\infty$ , de l'autre le flux  $q$ .



L'échange convectif entre le système et l'air ambiant, par un coefficient  $h$ , se fait grâce au bloc Convective Heat Transfer. L'accumulation de chaleur dans le solide est simulée par la masse thermique Thermal Mass.

Dans cet exemple, nous avons utilisé 2 conditions aux limites, la température de l'air ambiant  $T_{\infty}$  (Ideal Temperature Source) et le flux  $q$  (Ideal Heat Flow Source). Les solutions obtenues par les blocs thermiques de Simscape et la théorie qui sont tracées dans la fonction Callback StopFcn sont parfaitement identiques.



## II. Utilities

Dans cette librairie, nous avons les blocs qui permettent le passage du domaine Simulink à celui des composants physiques de Simscape et inversement ( $S \rightarrow PS$  et  $PS \rightarrow S$ ).

Nous remarquons que contrairement à Simulink, les ports d'entrée *In* et de sortie *Out* d'un sous-système Simscape ne sont pas différents et peuvent être réalisés par le même port



Connection Port

L'élément très important de cette bibliothèque est le bloc



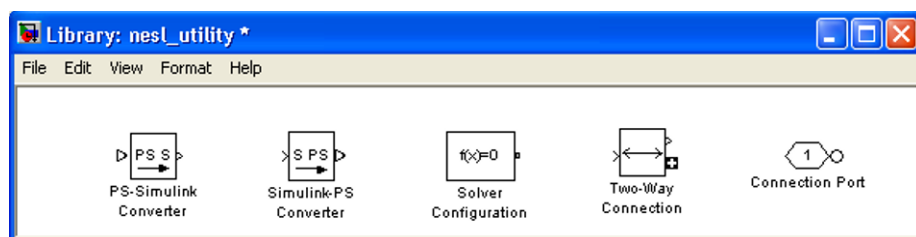
Solver  
Configuration

figuration

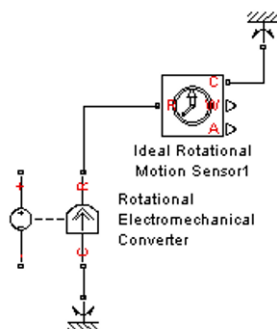
Ce dernier doit être raccordé à n'importe quel point du système.

Il spécifie des informations globales et fournit les paramètres dont a besoin le solveur Simulink avant tout début de simulation.

Chaque modèle a besoin d'un bloc unique de ce type.



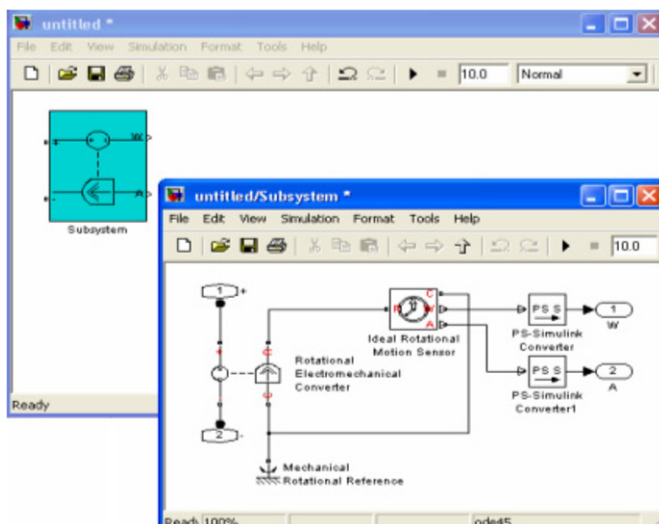
Considérons le système suivant, obtenu de l'exemple du mouvement rotationnel.



En sélectionnant tout ce système, on peut en obtenir un sous-système par *Edit/Create Subsystem*, qui comprend un moteur à courant continu et ses capteurs de position angulaire et de vitesse de rotation.

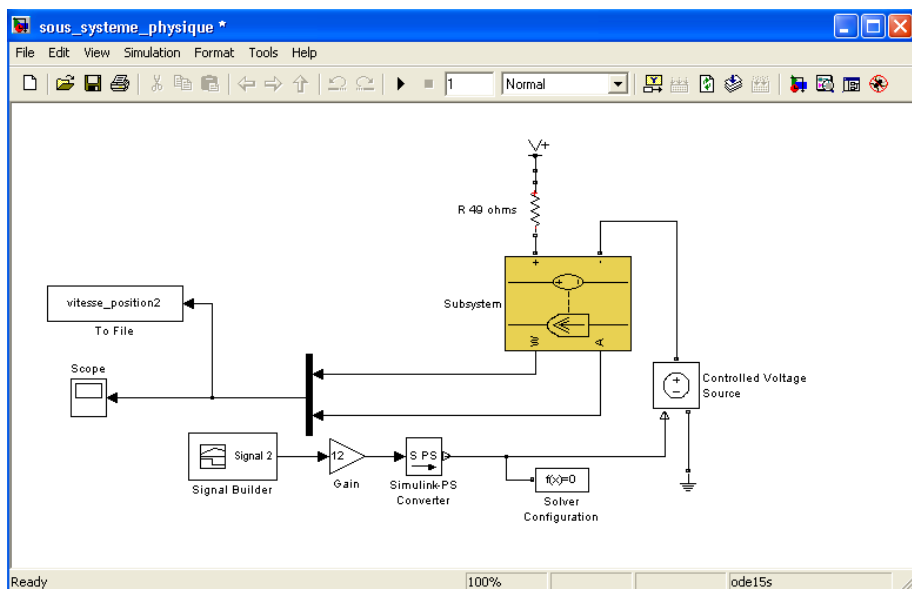
Ces mesures physiques sont transformées en signal de type Simulink. Par contre, les entrées électriques de l'induit sont de type physique de Simscape.



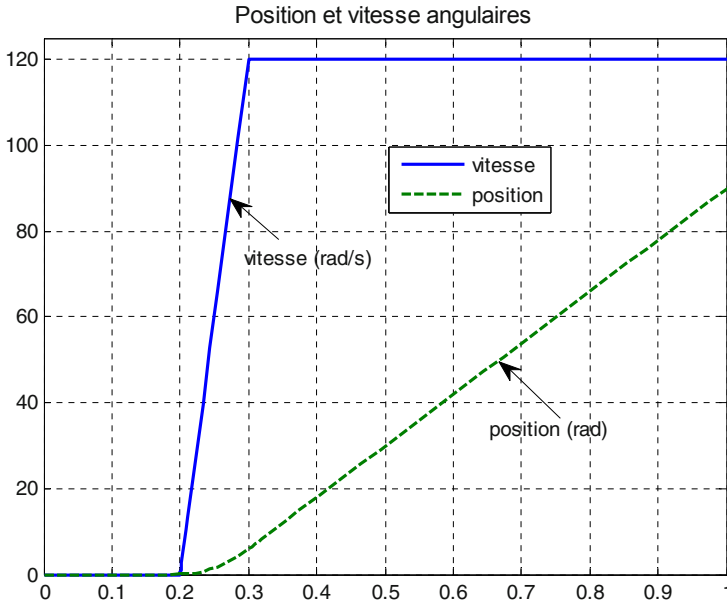


Les masses, électrique et mécanique, sont incluses dans le sous-système. Lorsqu'on crée le sous-système de cette façon les connecteurs se placent automatiquement.

Mais rien n'empêche, si l'on veut créer une autre entrée ou sortie, de copier un connecteur, de renommer son nom et de le connecter à l'entrée ou sortie désirée.



Nous obtenons les mêmes résultats que dans l'exemple précédent.

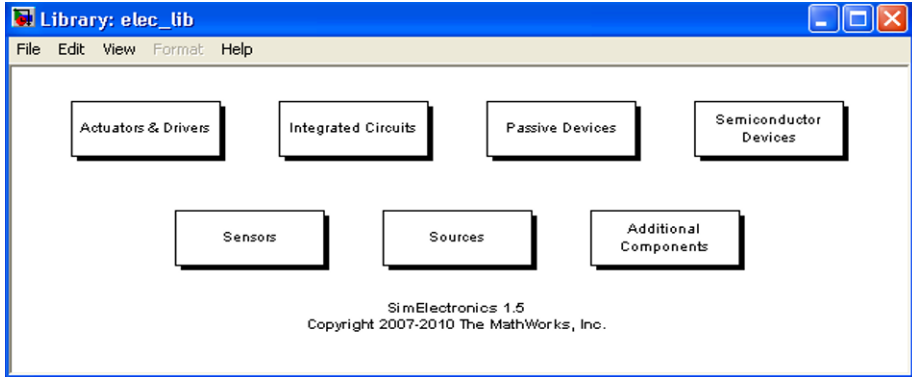


### III. SimElectronics

SimElectronics est un outil additionnel à Simscape permettant de simuler des composants électriques (moteurs, composants actifs ou passifs, capteurs, etc.).

Cette librairie contient les bibliothèques suivantes :

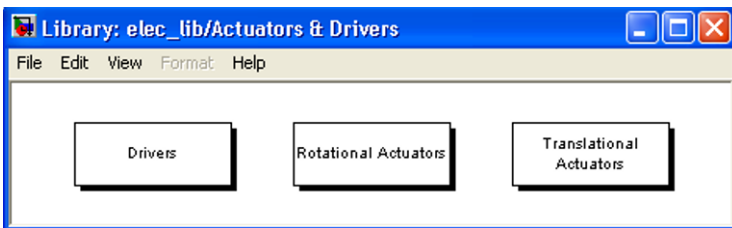
- `Actuators & Drivers`: actionneurs (différents types de moteurs) et drivers (PWM, ponts en H, ...),
- `Additional Components`: composants compatibles SPICE,
- `Integrated Circuits`: amplificateurs opérationnels, bibliothèque de portes logiques CMOS, etc.
- `Passive Devices`: 2 switches (commande en tension et en courant), fusible, quartz, relais, capacité et self variable, etc.
- `Semiconductor Devices`: des transistors (bipolaires, MOSFET, JFET), diode, etc.
- `Sensors`: thermocouple, thermistance, jauge de contraintes, capteur de proximité, photodiode, etc.
- `Sources`: une batterie générique, une cellule solaire, des bornes – et + d’une alimentation continue.



Le contenu de cette fenêtre (contenu de SimElectronics) peut être obtenu directement du prompt de Matlab :

```
>> elec_lib
```

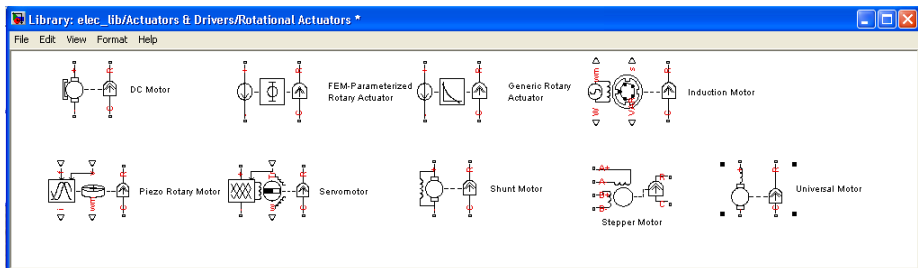
### III.1. Actuators & Drivers

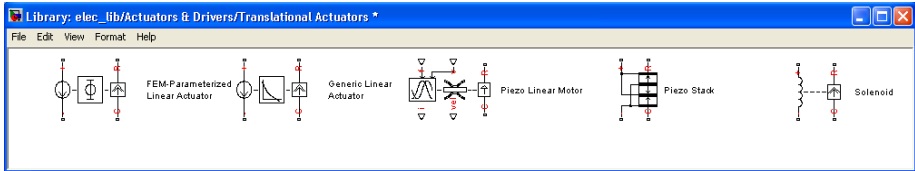


#### III.1.1. Translational et Rotational Actuators

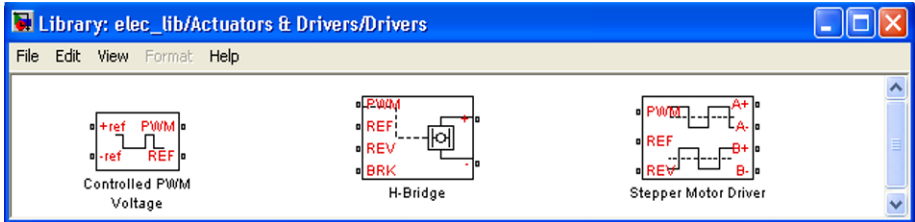
Dans ces librairies, nous trouvons différents types de modèles de moteurs à mouvement linéaire et de rotation:

- Moteur à courant continu,
- Servomoteur,
- Moteur à induction, etc.



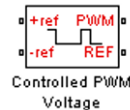


### III.1.2. Drivers



Nous trouvons, dans cette librairie, des drivers qui génèrent un signal PWM, pont en H et moteurs pas à pas.

### III.1.3. Applications à la commande d'un moteur à courant continu



#### • Application, générateur de signal PWM

Le bloc **Controlled PWM Voltage** permet de générer un signal PWM en fonction de la tension appliquée à son entrée **+ref**.

L'entrée **-ref**, la référence négative est reliée à la masse, ainsi que sa sortie **REF**.

En double-cliquant sur ce bloc, on peut spécifier la fréquence du signal, les valeurs du signal d'entrée correspondant, respectivement, aux valeurs du rapport cyclique, minimal 0 % et maximal 100%. Pour le mode **PWM** ou **Averaged**, la sortie est une valeur constante égale à la moyenne du signal PWM.

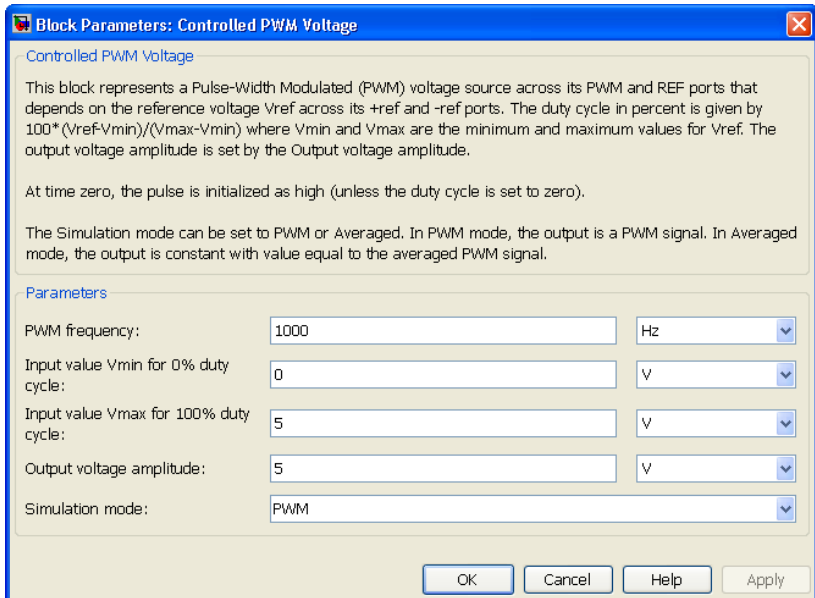
Dans cette boîte de dialogue, on spécifie aussi l'amplitude de la tension de sortie (**Outputvoltage amplitude**).

Nous pouvons choisir 2 modes pour le signal de sortie :

- **PWM** : le signal de sortie est un signal PWM (Pulse Width Modulation)
- **Average** : le signal de sortie est une moyenne du signal PWM.

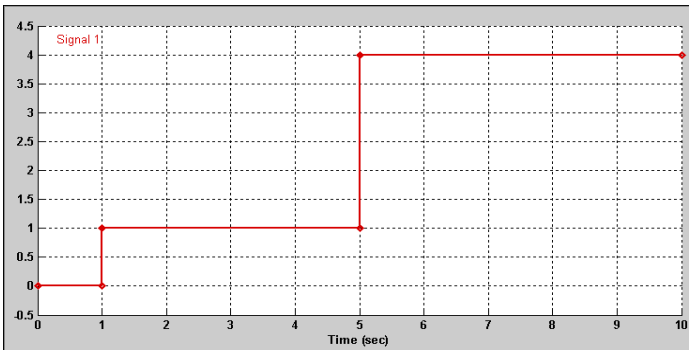
Dans l'exemple suivant, le signal PWM aura un rapport cyclique de 100% pour un signal de référence de 5 V et 0% pour 0 V.

Sa fréquence est de 1 kHz et son amplitude sera de 5 V.

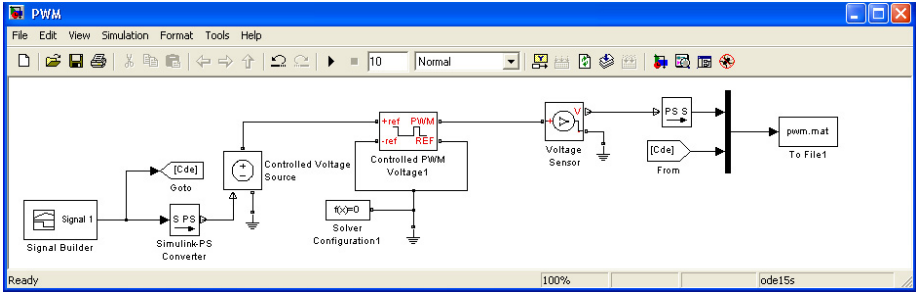


Dans le modèle suivant, l'entrée est générée par le bloc `Signal Builder` qui commande une source de tension contrôlée. L'entrée appliquée à l'entrée `ref` correspond, respectivement à 5V pour un rapport cyclique de 100% et 0V pour 0%. Ainsi, comme spécifié dans les valeurs du signal d'entrée suivant, le tableau donne le rapport cyclique en fonction du temps.

Instants	Rapport cyclique
0-1	0%
1-5	20%
5-10	80%



Le signal de commande ou de référence est donné par ce signal généré par le bloc `Signal Builder`.

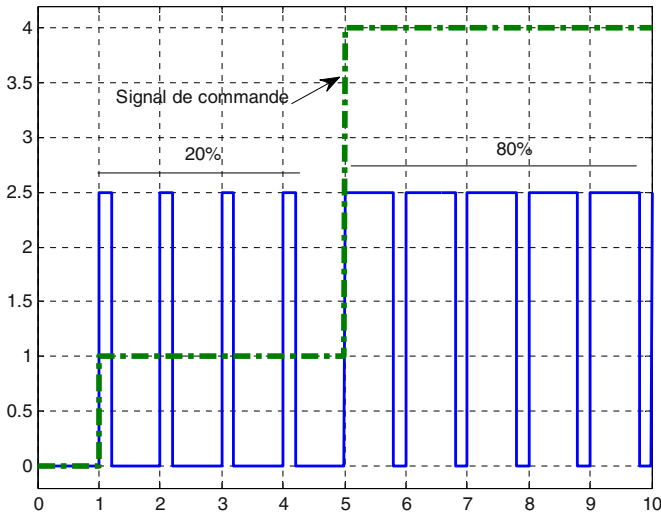


La tension de sortie est mesurée grâce au capteur de tension Voltage Sensor et sauvegardée dans le fichier binaire `pwm.mat`. Nous avons spécifié une amplitude de sortie de 2.5 V comme on le montre ci-dessous :

**Parameters**

PWM frequency:	<input type="text" value="1"/>	Hz
Input value Vmin for 0% duty cycle:	<input type="text" value="0"/>	V
Input value Vmax for 100% duty cycle:	<input type="text" value="5"/>	V
Output voltage amplitude:	<input type="text" value="2.5"/>	V
Simulation mode:	PWM	

Le signal PWM obtenu, est représenté par la figure suivante.



Nous vérifions que le signal de sortie PWM vaut 0 entre les instants 0 et 1 (rapport cyclique de 0 %).

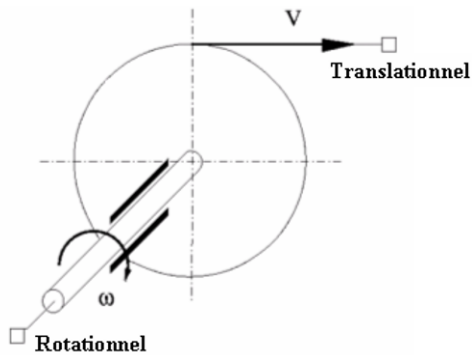
Entre 1 et 5, le rapport cyclique est de 20 % et 80 % entre  $t=5$  et 10.

### Régulation de vitesse d'un moteur à courant continu

On se propose de réaliser la régulation de la vitesse d'un moteur à courant continu (DC Motor) à l'aide d'un régulateur proportionnel et intégral PI que l'on programme à partir de composants physiques de Simscape.

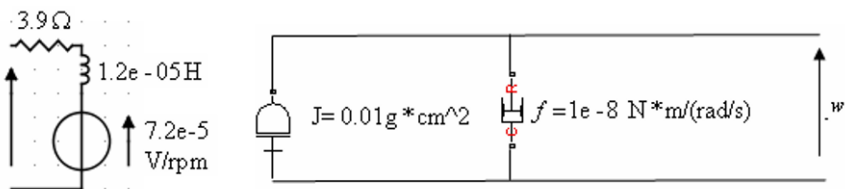
Au bloc DC Motor, nous avons ajouté une inertie (Inertia) de  $10 \text{ g.cm}^2$  et un couple de frottements (Friction) de  $0.02 \cdot 10^{-3} \text{ N.m}$ .

La vitesse est divisée par 5 en sortie de l'arbre moteur grâce au bloc réducteur de vitesse (Gear Box). Le couple mécanique est multiplié par la même valeur que le taux de réduction de la vitesse. Le bloc Wheel and Axle (Roue et Axe) permet de passer du mouvement de rotation au mouvement linéaire.



La vitesse linéaire  $V$  est égale à la vitesse angulaire  $\omega$  multipliée par le rayon de la roue,  $V = r\omega$ .

Le modèle électromécanique du moteur est le suivant.



Les couples d'inertie et de frottements du moteur sont négligeables devant ceux qu'on a rajoutés.

$$U(p) = E(p) + R I(p) + L p I(p)$$

avec :

$I$  : courant d'induit,

$L$  : self d'induit,

$E$  = force électromotrice d'induit (f.e.m).

$$E(p) = K \Omega(p)$$

$\Omega(p)$  : vitesse de rotation.

Le couple moteur est donné par :

$C_m = K I(p)$  si le flux est constant (excitation constante),

$C_m = J p \Omega(p)$  si on néglige les frottements,

$$U(p) = \Omega(p) (K + r J p / K + L p J p / K)$$

Ainsi la fonction de transfert de la vitesse de rotation sur la tension d'entrée est donnée par :

$$\begin{aligned} \frac{\Omega(p)}{U(p)} &= \frac{\frac{1}{K}}{1 + \frac{RJ}{K^2} p + \frac{LJ}{K^2} p^2} \\ &= \frac{k}{1 + \frac{2\zeta}{w_0} p + \frac{1}{w_0^2} p^2} \end{aligned}$$

avec :

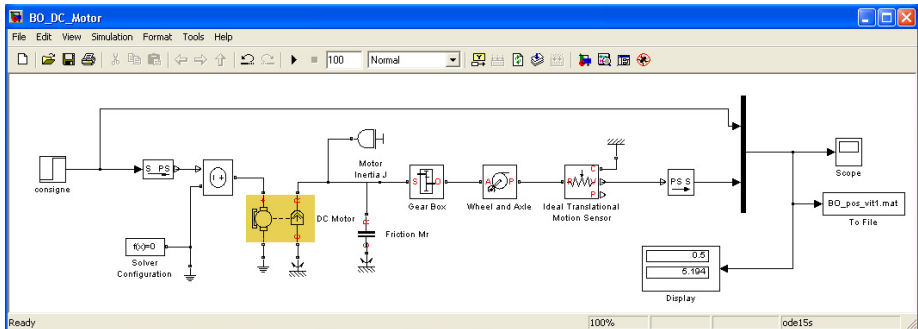
$\zeta$  : coefficient d'amortissement,

$w_0$  : pulsation propre,

$k$  : gain.

#### • Modélisation du moteur en boucle ouverte

Le modèle suivant consiste à tracer la réponse en boucle ouverte du moteur pour obtenir le type de modèle (identification par modèle de conduite).

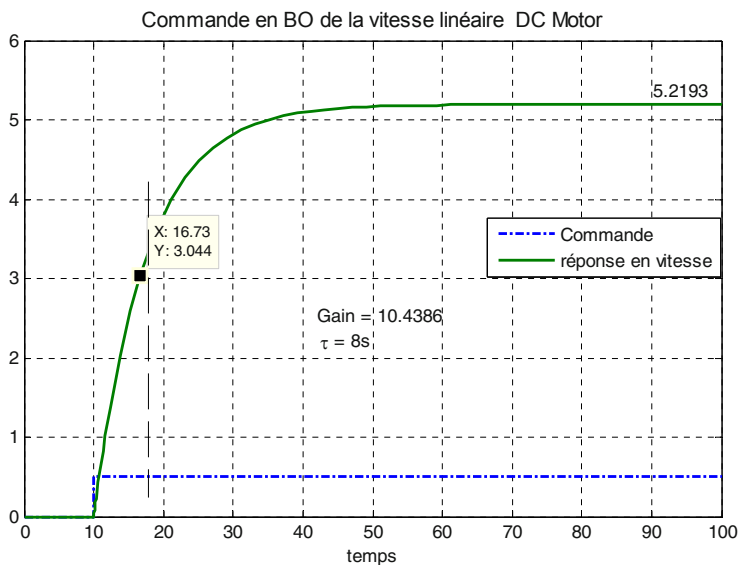


La courbe suivante représente la réponse en boucle ouverte de la vitesse linéaire du moteur suivi d'une boîte de vitesses (réduction de valeur 5) et d'une roue de rayon de 0.05 m avec axe qui permet de transformer le mouvement de rotation en mouvement linéaire.

La vitesse linéaire est alors donnée par  $v = r w$ ,  $r$  étant le rayon de la roue et  $w$  la vitesse angulaire.

Dans ce cas, le capteur qu'on doit utiliser est celui qui permet d'avoir la vitesse et la position linéaire (Ideal Translational Motion Sensor).

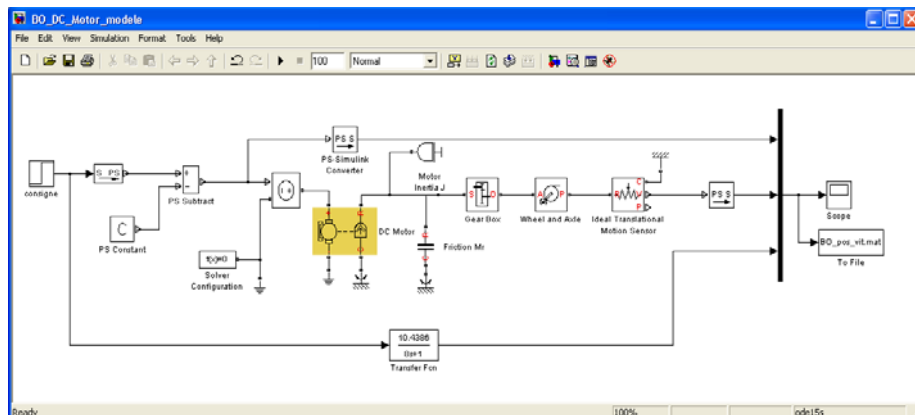




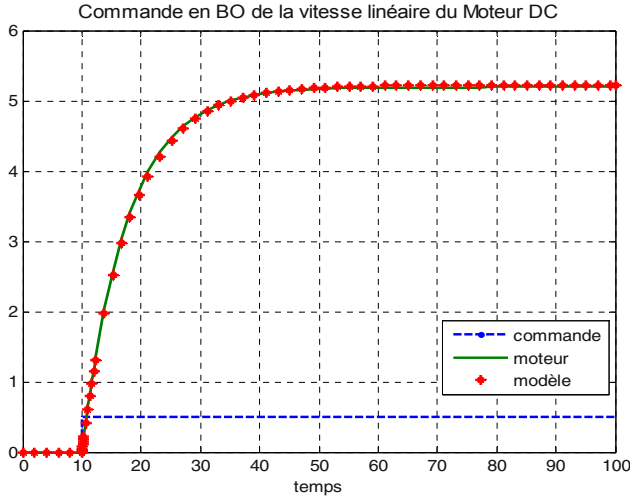
La fonction de transfert du système est identifiée comme celle du 1<sup>er</sup> ordre d'expression suivante :

$$H(p) = \frac{G}{1 + \tau p} = \frac{10.4386}{1 + 8p}$$

Dans ce modèle, nous traçons les 2 réponses indicielles, celle du moteur et celle de son modèle pour comparaison.



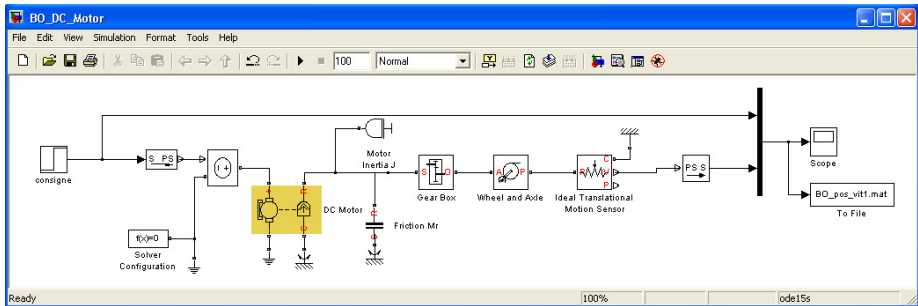
La courbe suivante, représente la réponse en boucle du moteur et celle de son modèle en vitesse.



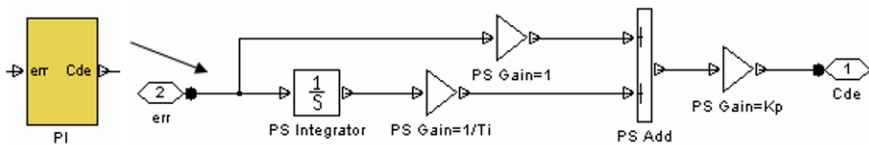
Il y a quasi similitude entre les réponses indicielles du moteur et de son modèle identifié en boucle ouverte.

#### • Régulation proportionnelle et intégrale, PI

Dans le modèle suivant, on réalise une régulation de type proportionnel et intégral, PI. Le moteur est à courant continu. Son arbre, constitué d'une inertie  $J$  et un couple de frottements  $M_r$  est couplé à un réducteur de vitesse et une roue qui permet de passer du mouvement de rotation à un mouvement rectiligne. La consigne est un échelon de 0.5 m/s.

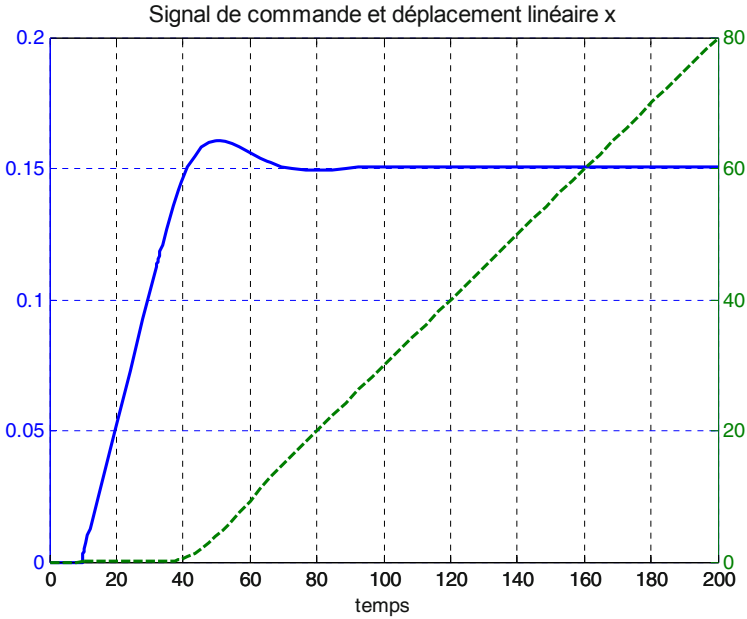


Le régulateur, proportionnel et intégral, PI est réalisé par des composants physiques de Simscape qu'on a programmés dans le sous-système PI suivant.





Le signal de commande issu du régulateur et le déplacement linéaire sont donnés dans la figure suivante.



• *Régulation par compensation de pôles*

On réalise maintenant une régulation par la compensation du pôle du système en utilisant un régulateur PI, d'expression :

$$D(p) = k \left( 1 + \frac{1}{\tau p} \right) = \frac{k}{\tau p} (1 + \tau p),$$

La fonction de transfert en boucle ouverte est alors donnée par :

$$BO(p) = 10.4386 \frac{k}{\tau p}$$

Ce qui donne en boucle fermée :

$$BF(p) = \frac{10.4386 k}{10.4386 k + \tau p} = \frac{1}{1 + \frac{\tau}{10.4386 k} p}$$

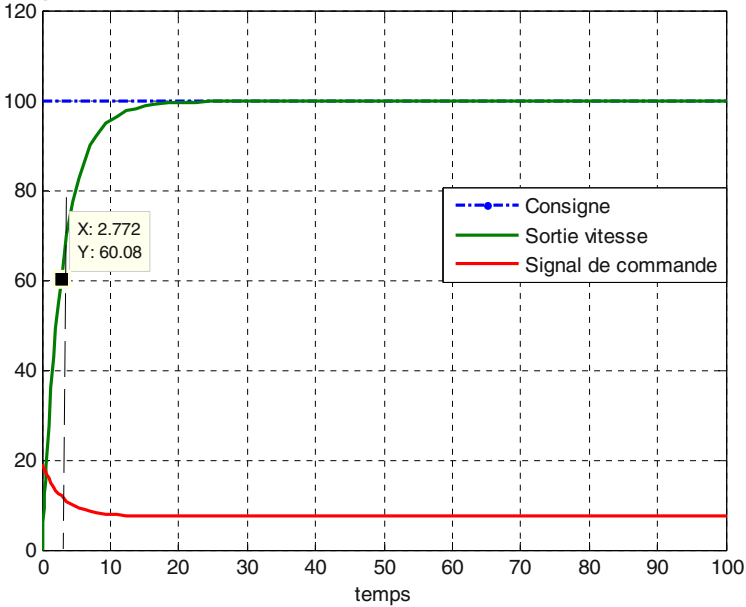
En boucle fermée, le système reste du 1<sup>er</sup> ordre dont on peut fixer la valeur de sa constante de temps par le paramètre de réglage  $k$ .

Si nous voulons rendre le système plus rapide, par exemple réduire de moitié sa constante de temps, il faut choisir le paramètre  $k$  de façon à ce que

$$10.4386 k = 2, \text{ soit } k = 0.1916.$$

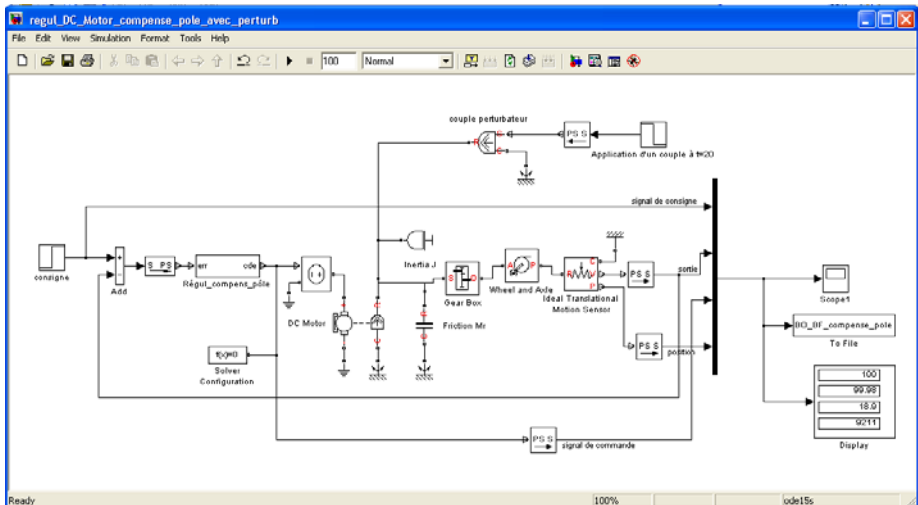


Régulation par compensation de pôle de la vitesse linéaire du Moteur DC



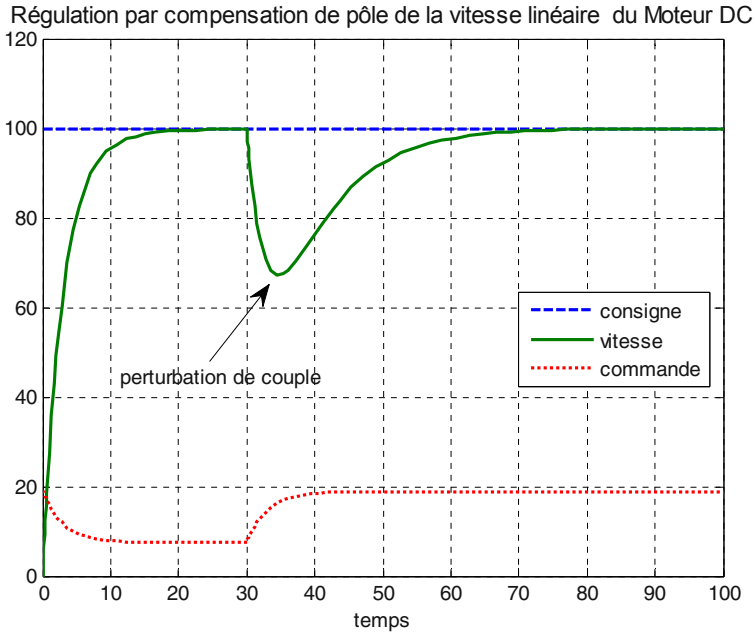
Cette courbe montre, approximativement, que le système se comporte comme un système du premier ordre, dont la constante de temps a été divisée par 2 par rapport au système en boucle ouverte.

Dans le modèle suivant, on applique une perturbation de couple.



Dans la figure suivante, nous représentons la réjection de la perturbation de couple à l'instant  $t=30$ .

La sortie rejoint toujours le signal de consigne grâce à une augmentation du signal de commande.

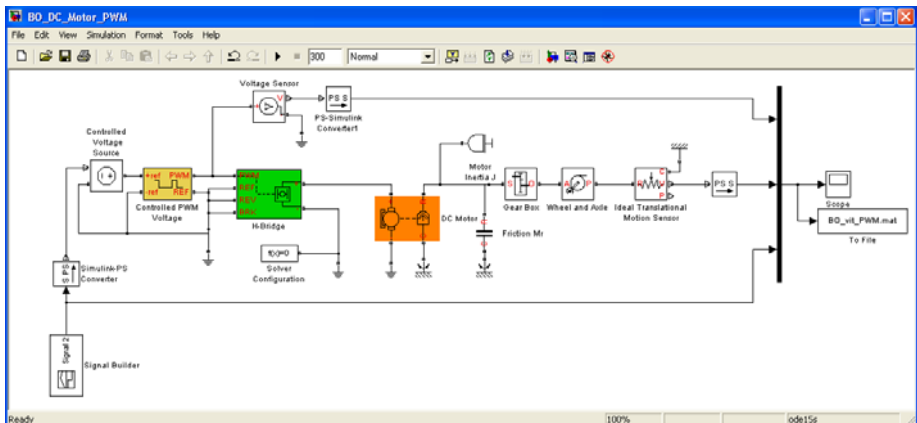


La perturbation de vitesse est rejetée au bout de 30s environ.

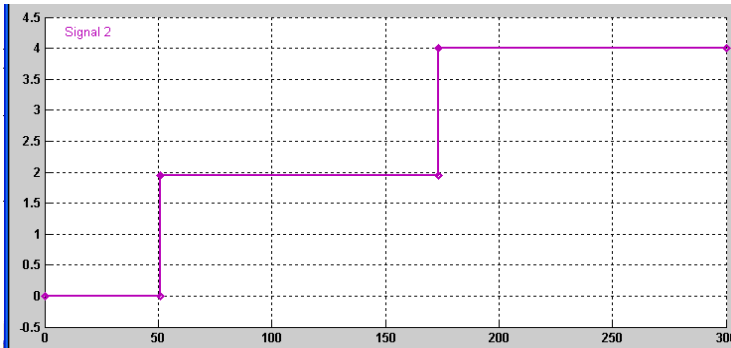
#### ▲ Commande par un pont en H, moteur en boucle ouverte

Le modèle suivant simule la commande du même moteur avec inertie, frottements, le réducteur de vitesse rapport 5, la transformation du mouvement de rotation en mouvement linéaire.

Dans ce cas présent, cette commande, en boucle ouverte, se fait grâce à un bloc générateur de signal PWM qui attaque un pont en H auquel est relié l'induit du moteur.

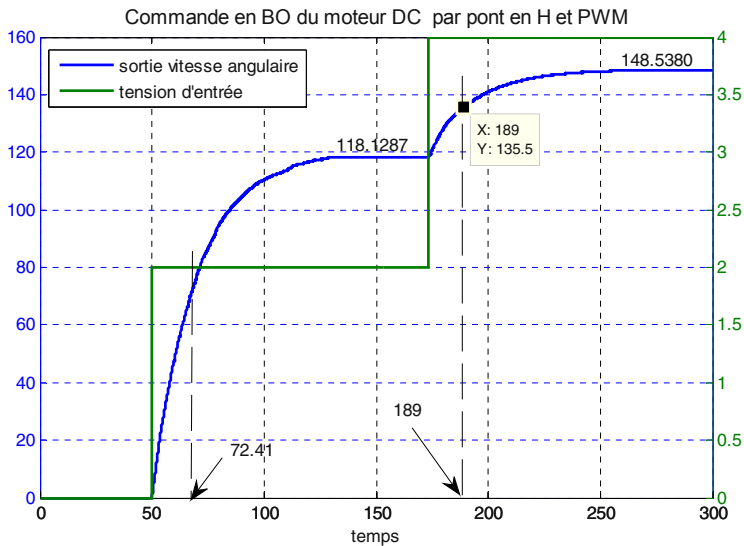


Le bloc Signal Builder fournit le signal d'entrée suivant formé d'échelons correspondants aux valeurs suivantes du rapport cyclique, 40% et 80%.



Ce signal est transformé en tension par le bloc Controlled Voltage Source pour commander la tension contrôlée sous forme PWM (Controlled PWM Voltage). Ce dernier bloc est le pont en H qui alimente le moteur à courant continu (H-Bridge). Les figures sont tracées grâce aux lignes de commandes suivantes, programmées dans la fonction Callback StopFcn.

```
load BO_vit_PWM.mat
plotyy(x(1,:),x(3,:), x(1,:),x(4,:)), grid
title('Commande en BO du moteur DC par pont en H et PWM')
xlabel('temps')
figure
plot(x(1,:),x(2,:)), xlabel('temps')
title('Signal PWM')
```



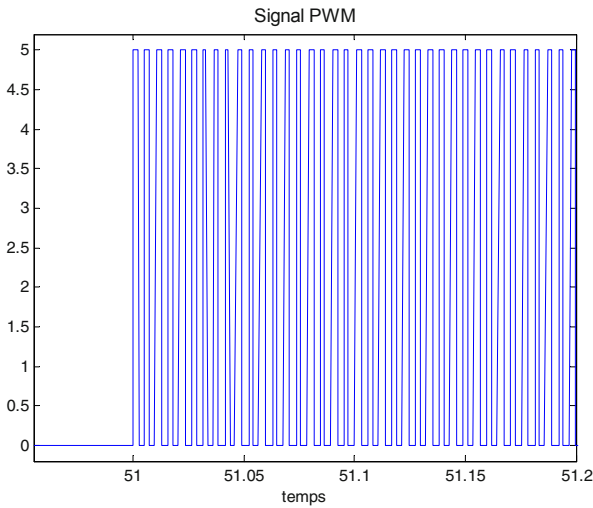


Nous avons représenté pour chaque échelon la valeur de la vitesse en régime établi. Si on estime pour chaque échelon, la réponse comme celle d'un système du 1<sup>er</sup> ordre, nous remarquons que lors du premier échelon, le gain est égal à 59.0643 et 74.2690 pour le deuxième échelon.

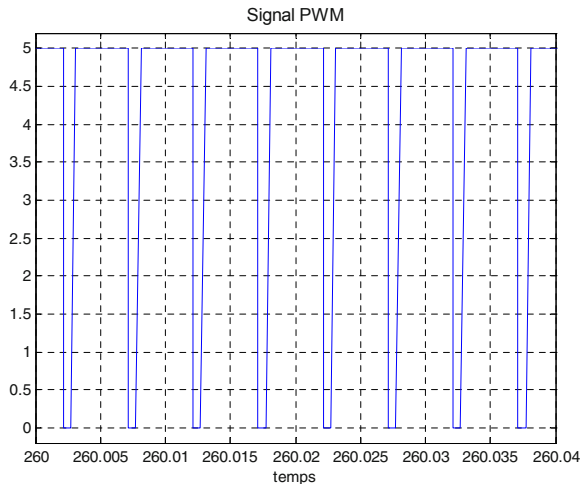
Les 2 constantes de temps sont approximativement de 22.4s et 19s, respectivement, pour le 1<sup>er</sup> et 2<sup>ème</sup> échelon. Nous pouvons estimer, alors, qu'en moyenne le modèle du système est de

$$\text{gain égal } 67 \text{ et de constante de temps de } 20.7\text{s} : H(p) = \frac{67}{1 + 20.7 p}$$

Au début de la simulation, nous avons appliqué un échelon de 2V correspondant à un signal PWM de rapport cyclique de 40% (presque un signal carré) que l'on observe dans la figure suivante.



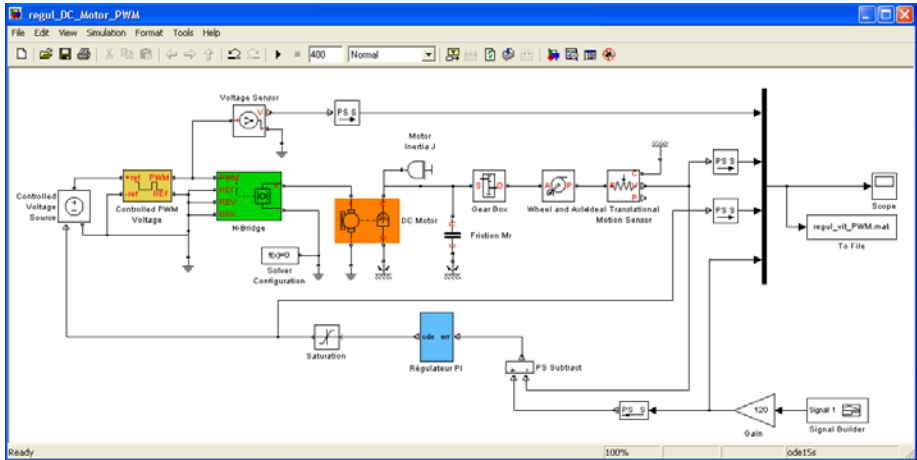
Au-delà de l'instant  $t=170$ , nous appliquons un échelon d'amplitude 4V, ce qui correspond, en sortie du bloc Controlled PWM Voltage, à un signal PWM de rapport cyclique de 80%, comme le montre la figure suivante.



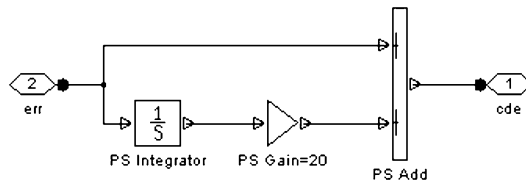
• *Moteur en boucle fermée*

Le modèle suivant représente la régulation en vitesse du moteur en appliquant sur son induit un signal PWM fourni par le bloc **Controlled PWM Voltage**, lequel commande le pont en H.

La consigne est réalisée par le bloc **Signal Builder** qui fournit un échelon d'amplitude unité après sa multiplication par le gain de valeur 120.



Le régulateur possède la forme suivante,



avec l'expression

$$D(p) = 1 + \frac{20}{p}$$

Les lignes de commande suivantes permettent de lire le fichier binaire et de tracer les différents signaux de consigne, sortie, signal de commande issu du régulateur et signal PWM qu'on applique à l'induit du moteur.

```
load regul_vit_PWM.mat

plot(x(1,:), x(3,:), x(1,:), x(5,:))
grid
title('Consigne et sortie vitesse - moteur DC par pont en H et PWM')
xlabel('temps')
figure
plot(x(1,:), x(4,:))
grid
```

```

title('Signal de commande')
xlabel('temps')

% tracé du signal de commande
figure
plot(x(1,:),x(2,:))
grid
axis([0 N -0.2 5.2])
title('Tension de commande PWM')
xlabel('temps')

```

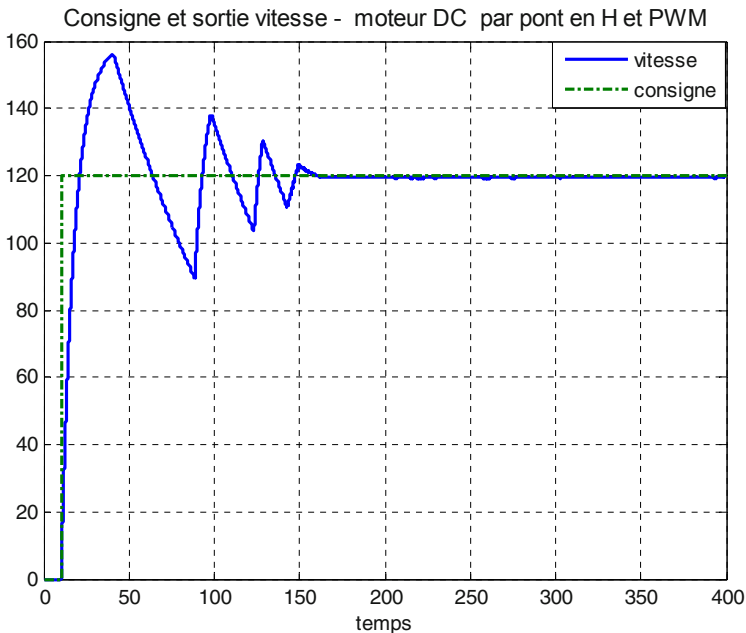
avec N, la durée de simulation, définie dans la fonction Callback `InitFcn` à laquelle on fait appel lors de l'étape d'initialisation.

```

% effacement de l'écran et suppression des fenêtres
% graphiques
clc
close all
N=400;

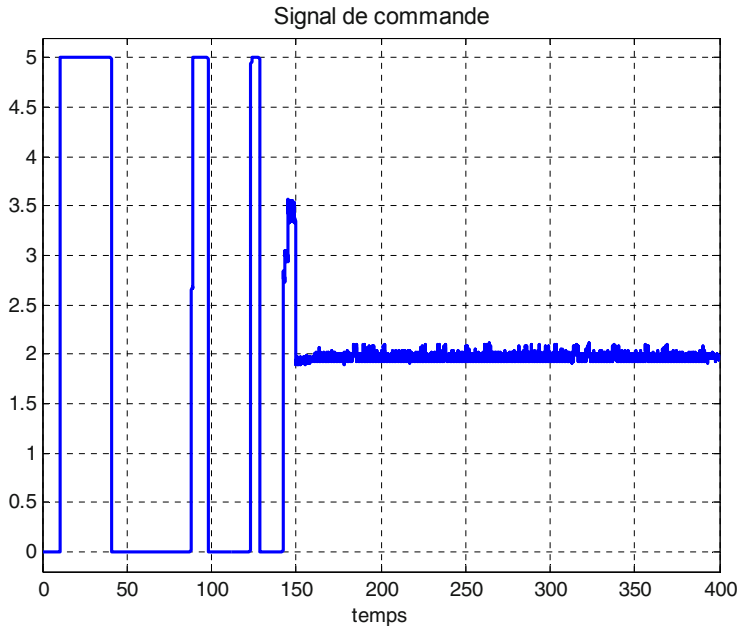
```

La figure suivante montre l'évolution du signal de sortie qui rejoint la consigne au bout de 150 échantillons après un régime transitoire oscillant (7 oscillations décroissantes).

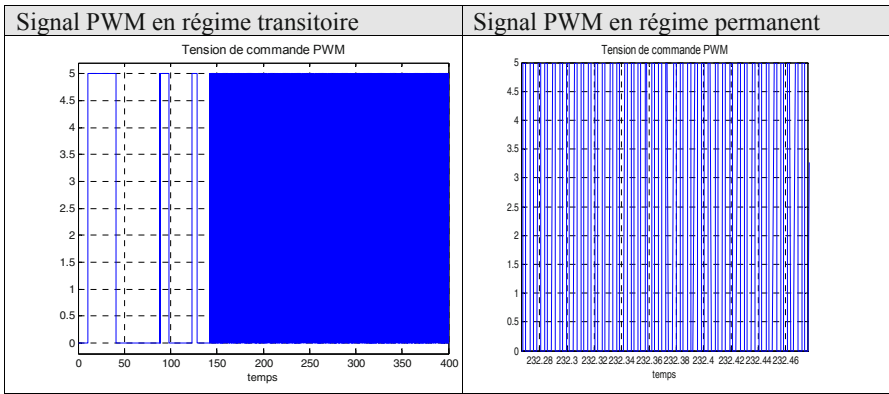


La figure suivante représente le signal de commande fourni par le régulateur.

Celui-ci va attaquer le bloc `Controlled PWM Voltage` qui fournira un signal PWM qui sera appliqué au pont en H.



Nous observons dans la courbe suivante du signal PWM, comme dans la précédente du signal issu du régulateur, les régimes, transitoire et permanent.



### III.1.4. Moteur à induction

Le moteur est constitué de deux parties principales:

- une partie fixe appelée « STATOR »,
- une partie mobile appelée « ROTOR ».

Ces deux parties sont séparées par un espace appelé «entrefer»

La figure suivante montre le modèle de circuit équivalent du bloc du moteur à induction.

Dans cette figure, les différents éléments sont :

- R1 : résistance d'une des phases du stator,
- R2 : résistance du rotor,
- L1 : inductance du stator,
- L2 : inductance de rotor,
- Lm : réactance réelle servant à représenter la puissance réactive requise pour produire le champ magnétique tournant ou impédance magnétisante,
- s : glissement du rotor.

$U$ ,  $I$  : tension d'alimentation sinusoïdale et courant dans l'une des phases.

Le « ROTOR » est un circuit fermé, donc court-circuité, qui se trouve dans le champ magnétique tournant créé par le « STATOR ». Il est alors traversé par des courants induits ou de Foucault qui provoquent sa rotation.

Le rotor ne tourne pas à la fréquence du champ magnétique, d'où la notion de glissement.

Si on note,  $\omega_s$  la pulsation du signal de commande du stator,  $p$  le nombre de paires de pôles et  $\omega_m$  la vitesse de rotation du rotor, le glissement est défini par :

$$g = 1 - p \frac{\omega_m}{\omega_s}$$

Le glissement est une grandeur qui rend compte de l'écart de vitesse de rotation d'une machine asynchrone par rapport à une machine synchrone hypothétique construite avec le même stator. La tension induite dans les bobines du rotor ne sera pas à fréquence constante et l'amplitude de cette tension sera proportionnelle à l'écart de vitesse entre celle du champ et la vitesse du rotor.

Les pertes par effet Joule évoluent avec le glissement et sont maximales au démarrage ( $g = 1$ ).

Si on note  $V_2$ , la tension au niveau des bobines du rotor ouvert (donc sans rotation), et  $f_2$  la fréquence du signal (rotor ouvert).

Nous avons, pour chaque phase du rotor en rotation :

$$\begin{aligned} V_r &= g V_2, \\ f_r &= g f_2. \end{aligned}$$

Chaque phase du rotor est une bobine que l'on peut représenter par une impédance inductive :  $Z_r = R_r + j X_r$

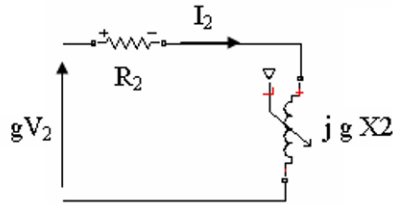
$R_r$  : résistance totale de la phase,

$X_r$  : réactance totale de la phase.

$$X_r = 2 \pi f_r L_r = 2 \pi s f_2 L_r = g X_2$$

$X_2$  est la réactance du rotor à circuit ouvert (sans rotation et même fréquence que le signal appliqué au stator).

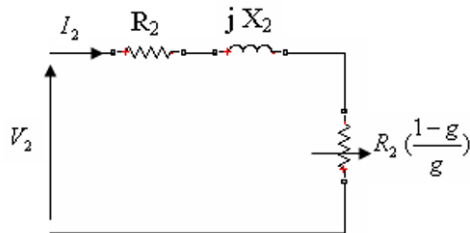
Une phase du rotor en court-circuit peut être modélisée par le schéma suivant :



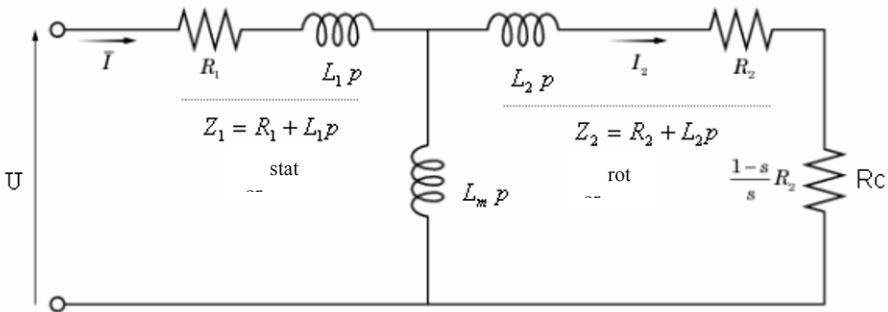
Nous avons alors, en divisant par  $g$  :

$$V_2 = I_2 \left( \frac{R_2}{g} + j X_2 \right)$$

Si l'on veut faire apparaître le terme  $Z_2 = R_2 + j X_2$ , le schéma de la phase devient :



Le modèle du moteur à induction est alors :



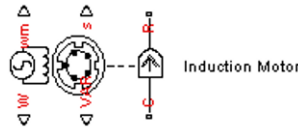
Le couple moteur est inversement proportionnel à la vitesse de rotation du rotor et proportionnel au carré de la valeur efficace de la tension d'entrée du stator par rapport au neutre.

Il possède une valeur maximale pour une certaine valeur du glissement.

$$T = \frac{n p R_2}{g w} \frac{V_{eff}^2}{(R_1 + R_2 + \frac{1-g}{g} R_2)^2 + (X_1 + X_2)^2}$$

$n$  étant le nombre de phases et  $p$  le nombre de paires de pôles.

Le bloc du moteur à induction dans Simscape est le suivant :



Sa boîte de dialogue possède 3 onglets.

Parameters

Electrical Torque | Power Supply | Mechanical

Model parameterization: By motor ratings

Magnetizing inductance  $L_m$ : 0.5 H

Rated mechanical power: 825 W

Rated speed: 3.5e+03 rpm

Rated RMS line-to-line voltage: 200 V

Rated supply frequency: 60 Hz

Rated RMS line current: 2.7 A

L1+L2 parameterization: From starting current

RMS starting (or locked rotor) line current: 7.5 A

R1 parameterization: From motor efficiency

Motor efficiency (percent): 95

Number of pole pairs: 1

Number of phases: 3

Stator connections: Star configuration

Le moteur est triphasé avec ses 3 phases montées en étoile. Par défaut , il possède 1 paire de pôles, soit un pôle Nord et un pôle Sud.

#### • *Electrical Torque*

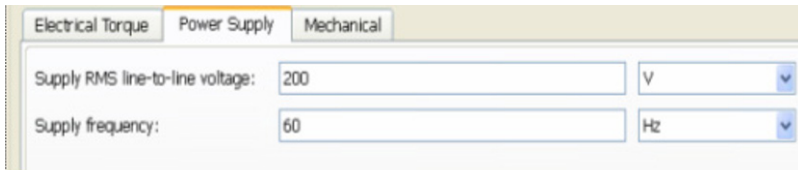
Dans cet onglet, on peut spécifier, entre autres,

- l'inductance magnétisante  $L_m$ ,
- la valeur efficace de la tension spécifiée de l'alimentation du stator,
- la fréquence du signal d'entrée spécifiée pour le moteur,
- le courant de ligne en ampères,
- la puissance mécanique estimée.

Les valeurs spécifiées sont celles définies pour le moteur.

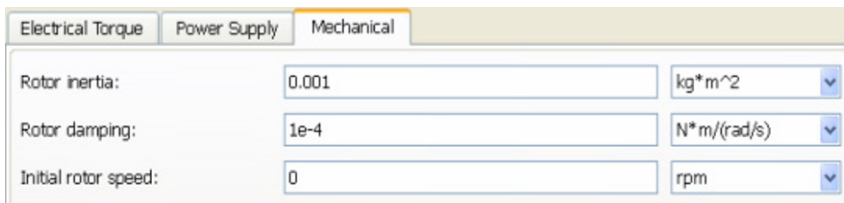
#### • Power Supply

Dans cet onglet, on retrouve les éléments réellement appliqués au moteur (tension efficace et fréquence).



#### • Mechanical

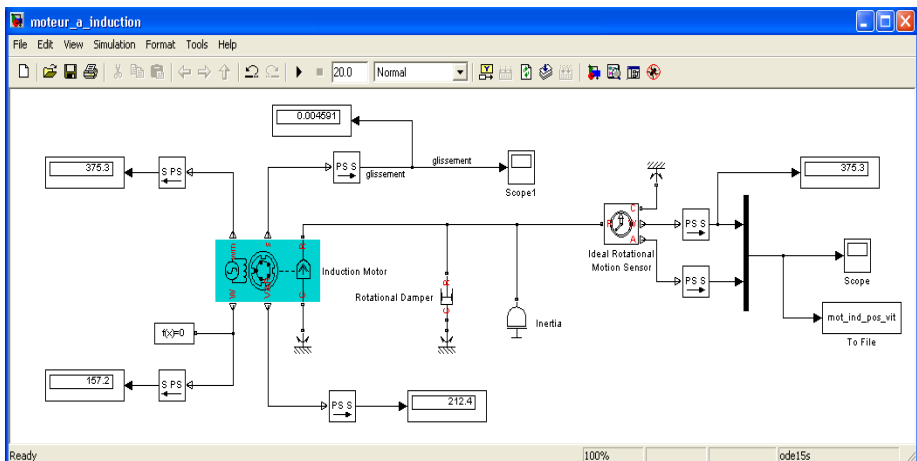
Dans cet onglet, on définit l'inertie et l'amortissement au niveau du rotor, ainsi que sa vitesse de rotation initiale.



#### • Application

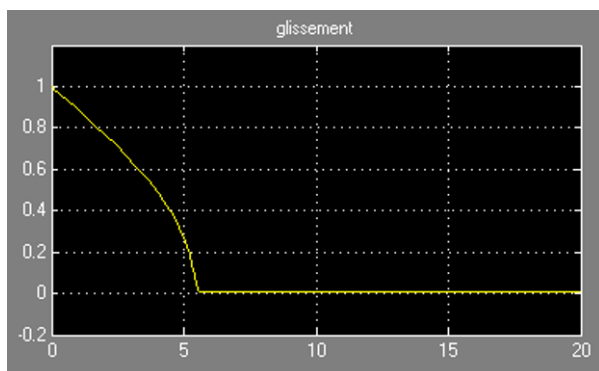
Le modèle suivant représente un moteur à induction dont on affiche l'évolution du glissement dans un oscilloscope et dont on trace celle de la vitesse et de la position angulaire qu'on récupère dans le fichier binaire `mot_ind_pos_vit.mat`.

Nous avons rajouté une inertie et un amortissement supplémentaires.

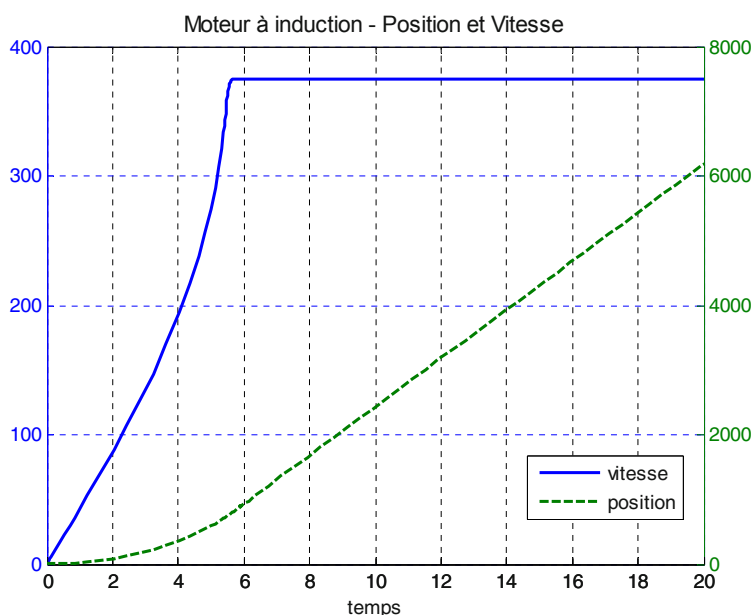




L'oscilloscope suivant montre l'évolution du glissement :



Le glissement, maximal et égal à 1 au départ, vaut 0.0046 à la fin de la période du démarrage. La figure suivante montre l'évolution de la vitesse angulaire du rotor ainsi que l'angle de rotation.



Après le démarrage qui dure un peu moins de 6s, la vitesse est constante et égale à 375.3 rad/s.

Si l'on choisit l'option «By equivalent circuit parameters», nous pouvons spécifier les valeurs des éléments du circuit électrique équivalent, comme les résistances  $R_1$ ,  $R_2$ , les selfs  $L_1$ ,  $L_2$  du stator et du rotor ainsi que la self magnétisante  $L_m$ .

Nous pouvons aussi imposer le nombre de paires de pôles ainsi que le nombre de phases.

Dans cette application, nous cherchons à tracer l'évolution du couple moteur.

**Parameters**

Electrical Torque | Power Supply | Mechanical

Model parameterization: By equivalent circuit parameters

Stator resistance R1: 1 Ohm

Rotor resistance R2: 1 Ohm

Stator inductance L1: 0.02 H

Rotor inductance L2: 0.02 H

Magnetizing inductance Lm: 0.5 H

Number of pole pairs: 1

Number of phases: 3

Stator connections: Star configuration

Les valeurs du circuit équivalent sont représentées par des variables dont on spécifie les valeurs dans la fonction Callback `InitFcn`.

**Model Properties**

Main | Callbacks | History | Description

Model callbacks

- PreLoadFcn
- PostLoadFcn
- InitFcn\*
- StartFcn
- PauseFcn
- ContinueFcn
- StopFcn\*
- PreSaveFcn
- PostSaveFcn
- CloseFcn

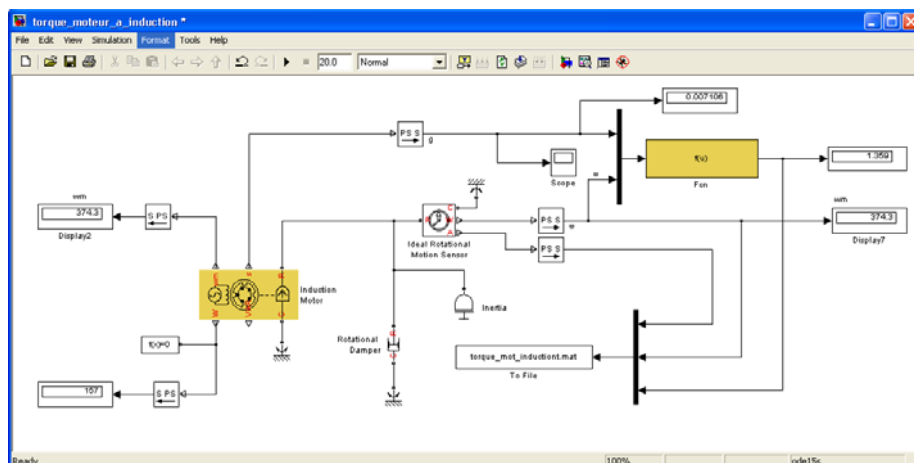
Model initialization function:

```
% valeurs des composants électriques
R1=1;
R2=1;
L1=0.02;
L2=0.02;
Lm=0.5;
n=3;
p=1;

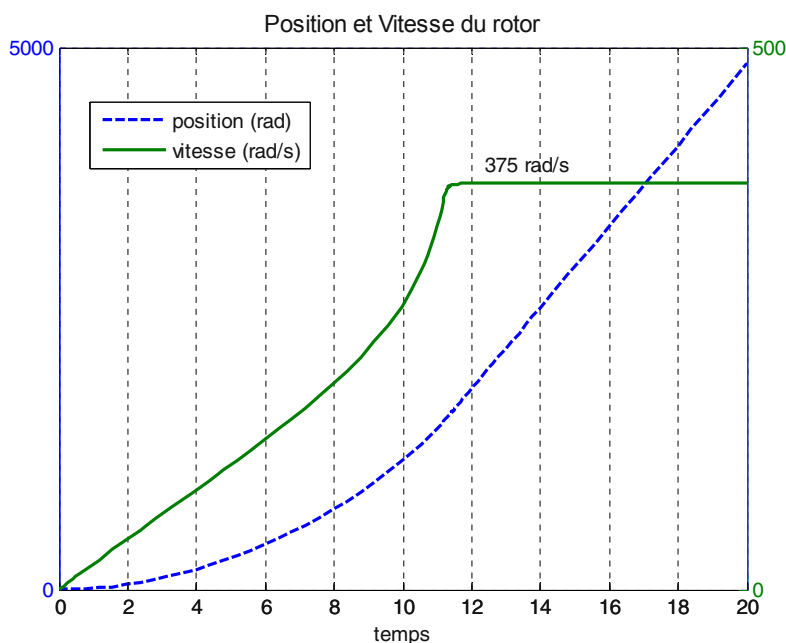
% signal d'entrée du stator
Vrms=200;
f=60;
```

Dans le bloc `Fcn`, nous avons programmé l'expression qui donne le couple en fonction de :

- la vitesse de rotation,
- le glissement,
- le nombre de paires de pôles,
- le nombre de phases,
- des valeurs de résistances et inductances au niveau du stator et du rotor.

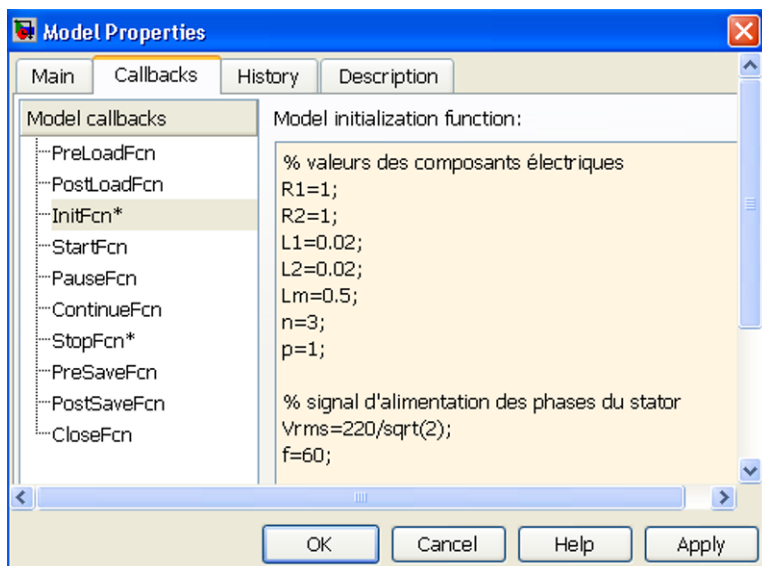


La fonction Callback `StopFcn` permet de tracer les courbes de vitesse et position angulaire ainsi que celle du couple sur le rotor.



Après un régime transitoire d'un peu moins de 12 secondes, la vitesse se stabilise à la valeur 375 rad/s.

Les paramètres du moteur ainsi que son alimentation électrique sont définis dans la fonction Callback `InitFcn`.



La tension triphasée appliquée à chacune des phases du stator est de 220V d'amplitude, soit  $220/\sqrt{2}$  V de valeur efficace qu'on spécifie dans la boîte de dialogue du moteur. Comme il y a une paire de pôles, la vitesse de rotation sera égale à la fréquence de l'alimentation :

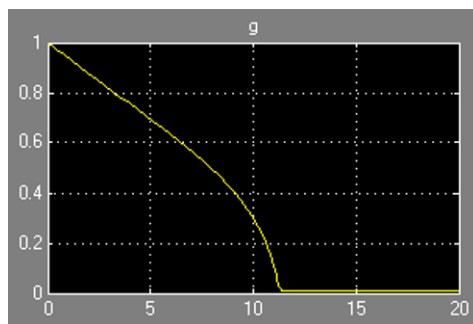
```
>> wth=2*pi*60
wth =
    376.9911
```

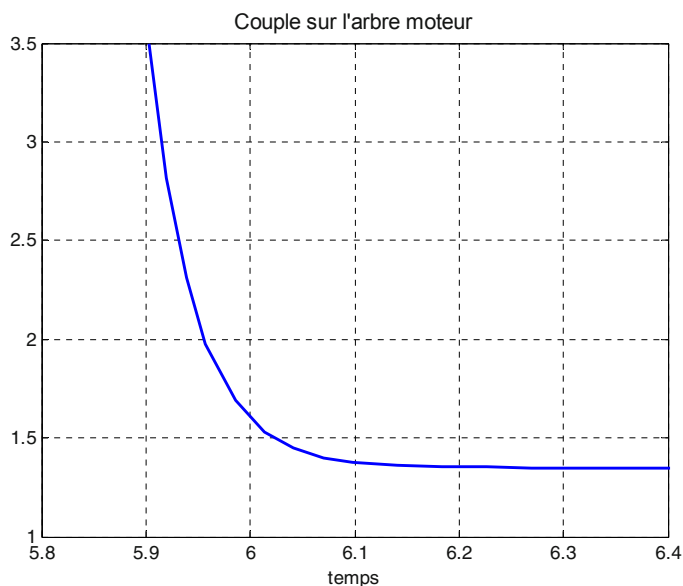
Si la valeur finale du glissement n'est pas nulle, la vitesse de rotation est alors :

$$w_m = (1 - g) w$$

```
>> wth=(1-7e-3)*2*pi*60
wth =
    374.3522
```

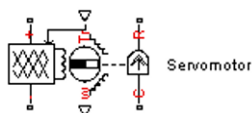
L'évolution du glissement, de 1 à 7e-3, est donné par l'oscilloscope suivant :





### III.1.5. Servomoteur

Le bloc du servomoteur est le suivant :



Le servomoteur est un moteur à courant continu sans balais (brushless). Le couple est contrôlé en circuit fermé.

Pour simuler un type de moteur, nous devons spécifier une enveloppe couple-vitesse qui délimite les valeurs admissibles soit les valeurs des vitesses des couples maximums correspondants.

Le block limite automatiquement le couple réel aux demandes de couple.

Nous spécifions aussi dans la boîte de dialogue suivante, la résistance électrique du modèle correspondant aux pertes par effet Joule.

Cette résistance est calculée selon l'efficacité du moteur utilisé.

La puissance appliquée  $U I$  est égale à la somme des pertes par effet Joule,  $R I^2$  et de la puissance mécanique utile  $T w$ ,  $T$  étant le couple.

$$U I = R I^2 + T w$$

On définit l'efficacité du moteur par:

$$E = \frac{T w}{U I}$$

En remplaçant le courant  $I$  dans la première relation, nous obtenons :  $R = \frac{EV^2}{T_w}(1-E)$

L'efficacité étant comprise entre 0 et 1, nous trouvons toujours une valeur  $R > 0$ .

En double-cliquant sur le bloc *Servomotor*, on spécifie certains paramètres (Electrical Torque) et mécaniques (Mechanical) qui le définissent; notamment les valeurs du couple et de la vitesse de l'enveloppe définissant les limites que doit englober ce couple. Ces 3 points ont pour coordonnées la vitesse du moteur et le couple maximum admissible, comme on les fixe dans la figure suivante. Dans cette boîte de dialogue, on spécifie aussi des paramètres concernant l'efficacité et les pertes du moteur.

Parameters

Electrical Torque Mechanical

Vector of rotational speeds: [ 0 3.75e+3 7.5e+3 8e+3 ] rpm

Vector of maximum torque values: [ 0.09 0.08 0.07 0 ] N\*m

Torque control time constant, Tc: 0.02 s

Motor and driver overall efficiency (percent): 100

Speed at which efficiency is measured: 3.75e+3 rpm

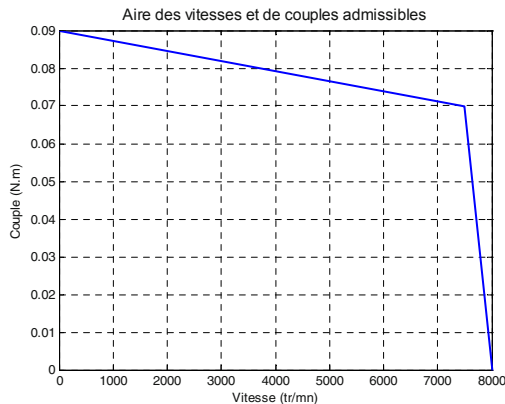
Torque at which efficiency is measured: 0.08 N\*m

Torque-independent electrical losses: 0 W

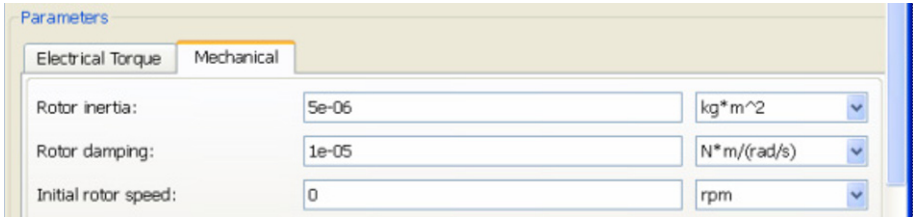
Supply series resistance: 0 Ohm

L'enveloppe couple-vitesse des couples admissibles est donnée, par défaut dans la figure suivante, selon les valeurs spécifiées dans la boîte de dialogue ci-dessus.

Cette enveloppe couple-vitesse montre, en effet, que si l'on veut augmenter le couple on doit généralement baisser la vitesse (par un réducteur de vitesse, par exemple) et que si l'on augmente la vitesse, il n'en suit une diminution du couple, pour une même puissance donnée.

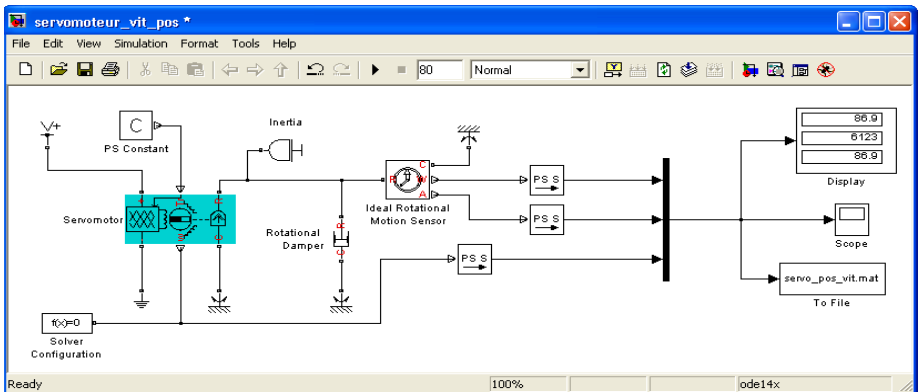


Dans l'onglet Mechanical, nous trouvons les valeurs par défaut de l'inertie du rotor, son amortissement et la valeur initiale de la vitesse de rotation qu'on peut modifier selon le type de servomoteur dont on dispose.

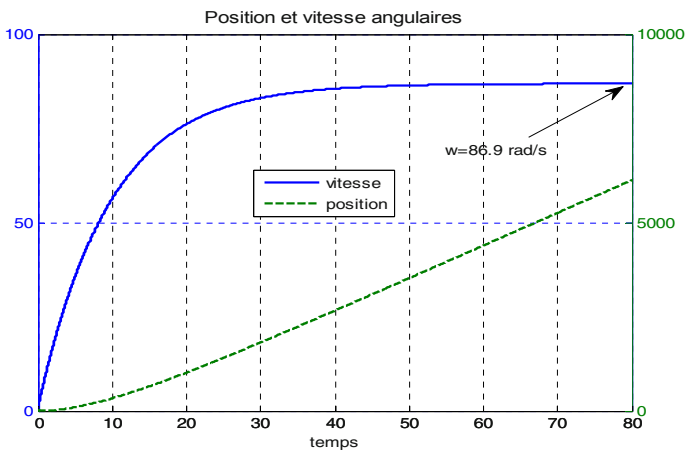


Dans le modèle suivant, nous appliquons une tension de 12 V à l'induit (stator) du moteur et nous avons ajouté des couples extérieurs d'inertie de  $0.01 \text{ kg}\cdot\text{m}^2$  et un amortisseur de coefficient  $0.001 \text{ N}\cdot\text{m}/(\text{rad}/\text{s})$  en sortie de l'arbre.

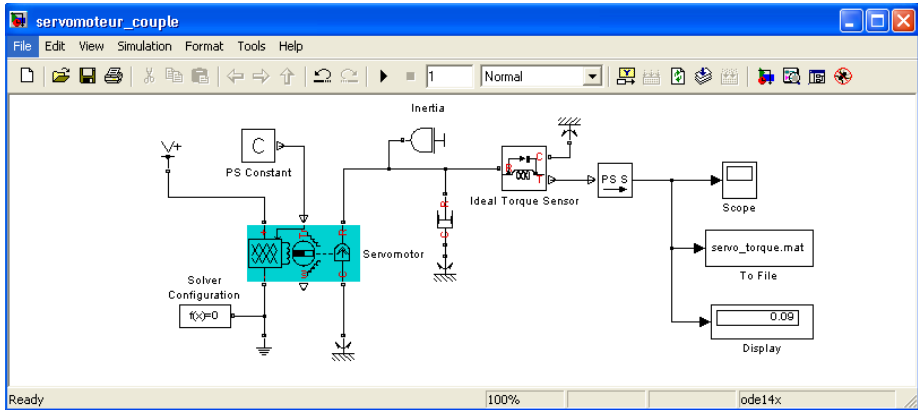
Nous spécifions une demande de couple de 5 N.m, défini par le bloc PS Constant.



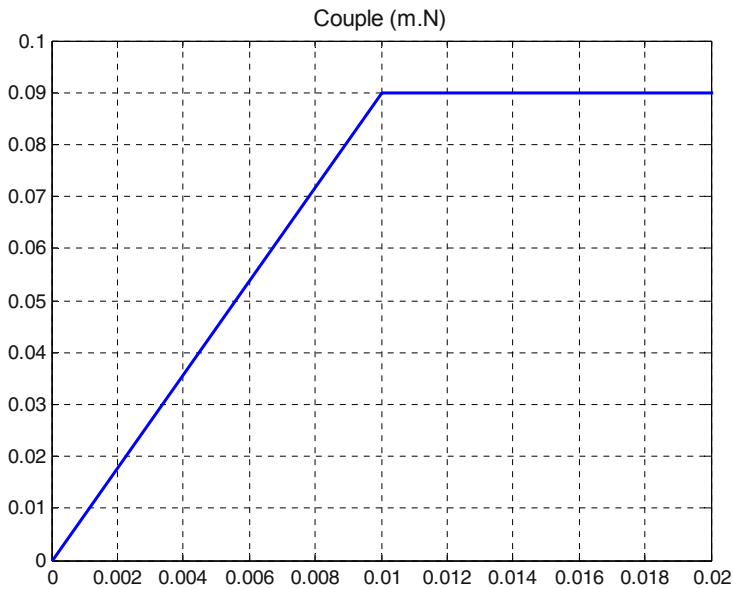
La vitesse et la position, obtenues grâce au capteur « Ideal Rotational Motion Sensor », sont tracées par les commandes de la fonction Callback StopFcn.



Pour connaître le couple réel, nous utilisons le capteur de couple « Ideal Torque Sensor ».



Comme le montre la figure suivante, le couple est limité très rapidement à 0.09 N.m.



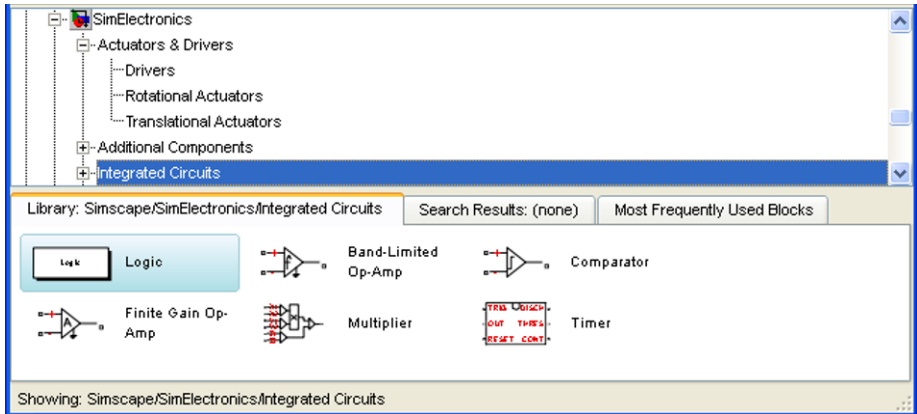
### III.2. Integrated circuits

Cette librairie contient une bibliothèque de composants logiques et 2 types d'amplificateurs opérationnels :

- amplificateur opérationnel à bande limitée,
- amplificateur opérationnel à gain fini.

La fenêtre suivante représente cette librairie dans le browser Simulink.

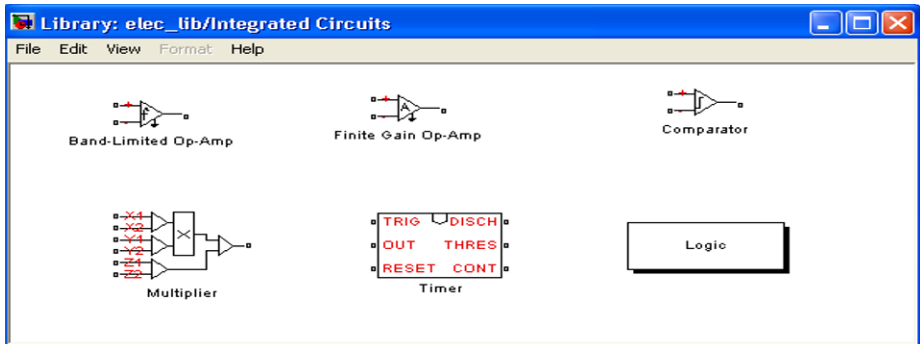




Avec la commande qu'on lance du prompt de Matlab,

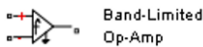
```
>> simscape
```

cette bibliothèque se présente comme suit :



### III.2.1. Amplificateurs opérationnels

#### ✦ Amplificateur opérationnel à bande limitée



La tension de sortie se comporte comme un filtre passe bas du 1<sup>er</sup> ordre vis-à-vis de la différence de tension entre ses bornes, positive et négative.

$$V_s = \frac{A(v_+ - v_-)}{1 + \frac{1}{2\pi f_c}} - R_s I_s$$

avec :

A : gain,

$f_c$  : fréquence de coupure à -3 dB,

$R_s$  : résistance de sortie,

$I_s$  : courant de sortie.

Le courant d'entrée est donné par :

$$I_e = \frac{v_+ - v_-}{R_e}$$

$R_e$  : résistance d'entrée

En double-cliquant sur ce bloc, on peut spécifier toutes ces valeurs.

Parameters		
Gain, A:	<input type="text" value="1000"/>	
Input resistance, Rin:	<input type="text" value="1e+6"/>	Ohm ▾
Output resistance, Rout:	<input type="text" value="100"/>	Ohm ▾
Minimum output, Vmin:	<input type="text" value="-15"/>	V ▾
Maximum output, Vmax:	<input type="text" value="15"/>	V ▾
Maximum slew rate, Vdot:	<input type="text" value="1000"/>	V/s ▾
Bandwidth, f:	<input type="text" value="1e+5"/>	Hz ▾
Initial output voltage, V0:	<input type="text" value="0"/>	V ▾

#### • Amplificateur à gain fini



Finite Gain Op-Amp

Cet amplificateur est défini par :

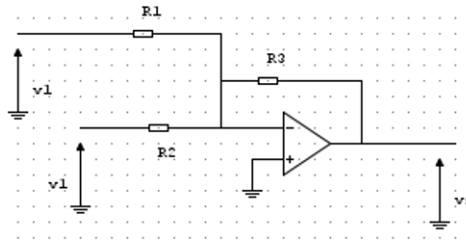
$$V_s = A(v_+ - v_-) - R_s I_s$$

Le courant d'entrée est donné par l'expression précédente de l'amplificateur à bande limitée.

Sa boîte de dialogue permet la spécification du gain A, les résistances d'entrée  $R_{in}$ , de sortie  $R_s$  et les tensions, minimale  $V_{min}$  et maximale  $V_{max}$  d'alimentation.

Parameters		
Gain, A:	<input type="text" value="1000"/>	
Input resistance, Rin:	<input type="text" value="1e+6"/>	Ohm ▾
Output resistance, Rout:	<input type="text" value="100"/>	Ohm ▾
Minimum output, Vmin:	<input type="text" value="-15"/>	V ▾
Maximum output, Vmax:	<input type="text" value="15"/>	V ▾

Le modèle suivant simule un montage sommateur.

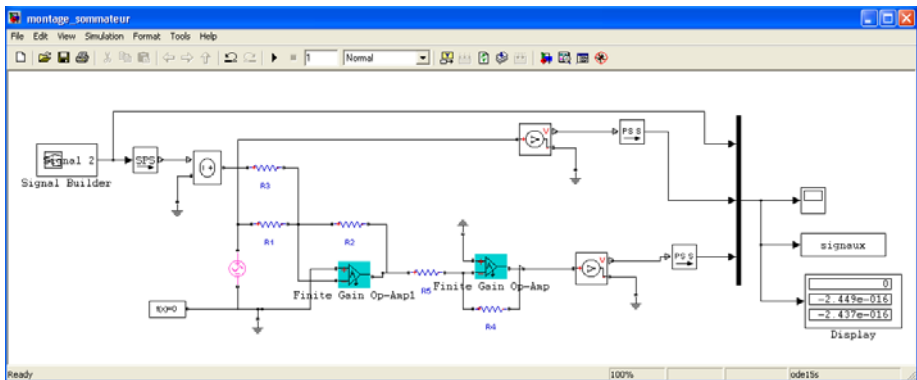


Le théorème de Millman, appliqué à l'entrée  $v_-$ , donne :  $\frac{v_- - v_1}{R_1} + \frac{v_- - v_2}{R_2} + \frac{v_- - v_s}{R_3} = 0$

D'où :  $v_s = -R_3 \left( \frac{v_1}{R_1} + \frac{v_2}{R_2} \right)$

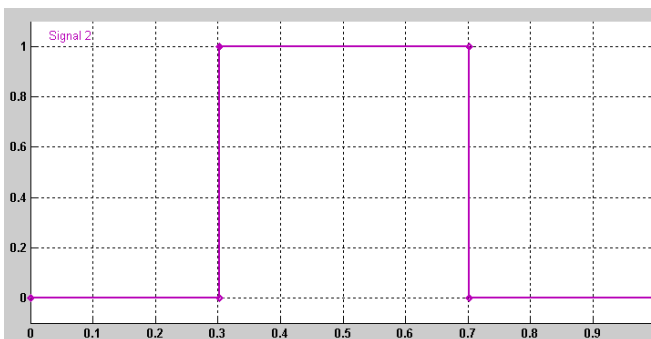
Si l'on choisit  $R_1 = R_2 = R_3$ , nous obtenons  $v_s = -(v_1 + v_2)$

Dans le modèle suivant, nous avons fait suivre le sommateur par un circuit inverseur pour éliminer le signe moins.

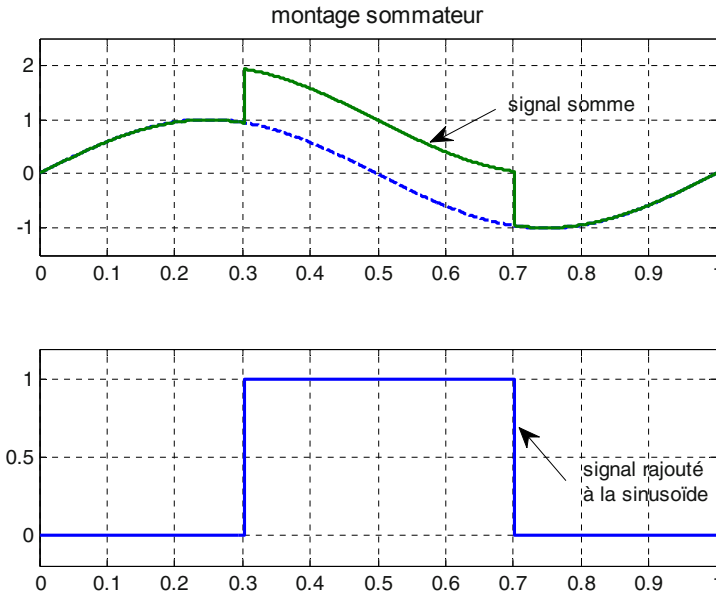


Dans ce montage, toutes les résistances sont égales à  $10\text{ k}\Omega$ .

Le signal qu'on ajoute à la sinusoïde est donné par le bloc Signal Builder ; c'est un créneau de valeur 1 entre les instants  $t=0.3$  et  $0.7$ .



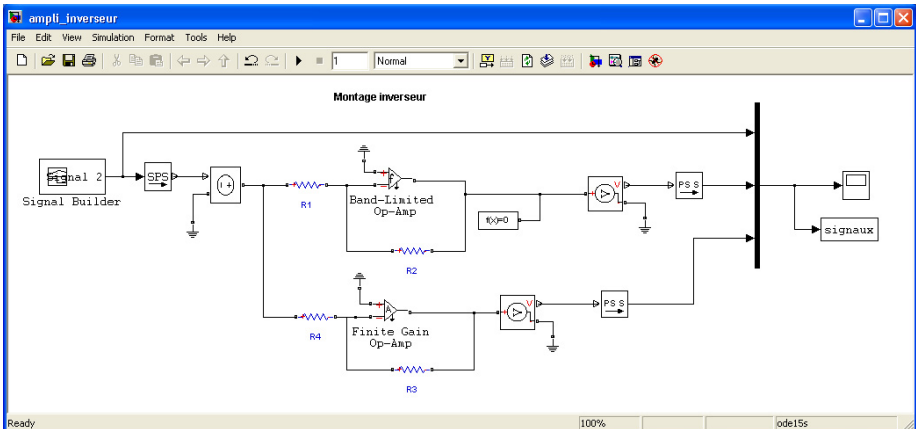
L'amplificateur opérationnel Finite Gain Op-Amp permet d'éliminer le signe « - » et de réaliser une vraie sommation. Le créneau est ajouté à la sinusoïde entre les instants 0.3 et 0.7, comme le montre la figure suivante.



Entre les instants 0.3 et 0.7, la sinusoïde est décalée de 1 vers le haut suite à l'ajout du créneau à la sinusoïde d'origine.

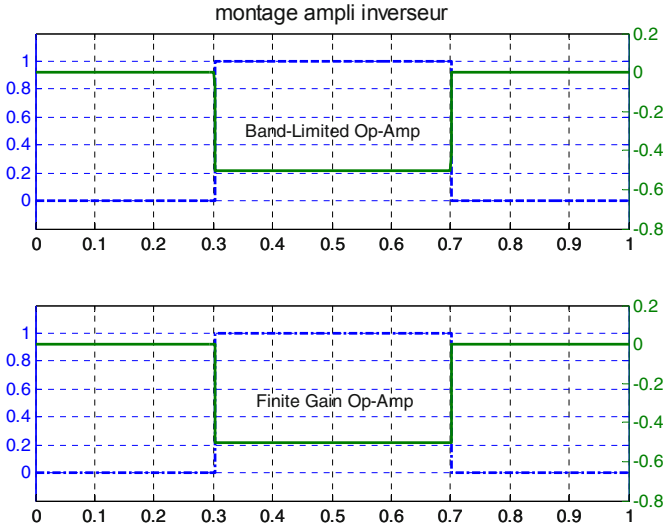
#### • Comparaison des 2 amplificateurs opérationnels

Dans l'exemple suivant, nous programmons le même type de circuit inverseur, le premier utilisant l'amplificateur à bande limitée et le deuxième, celui à gain limité.



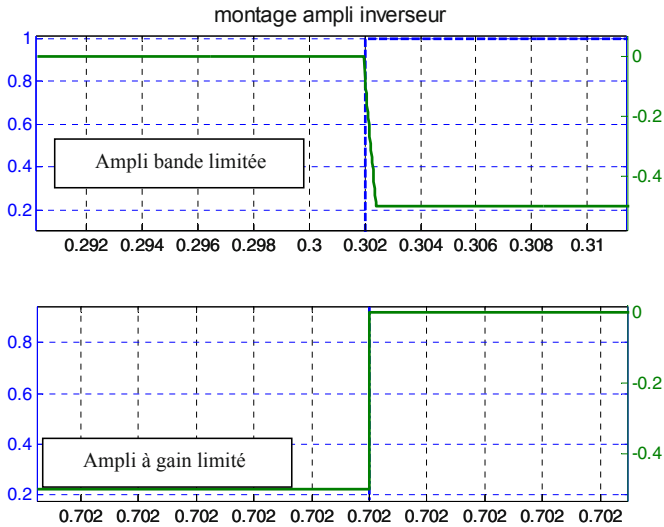
Le même signal d'entrée est fourni par le bloc `Signal Builder`.

Nous cherchons à comparer la réponse en fréquences des deux montages en examinant les réponses aux fronts du signal d'entrée.



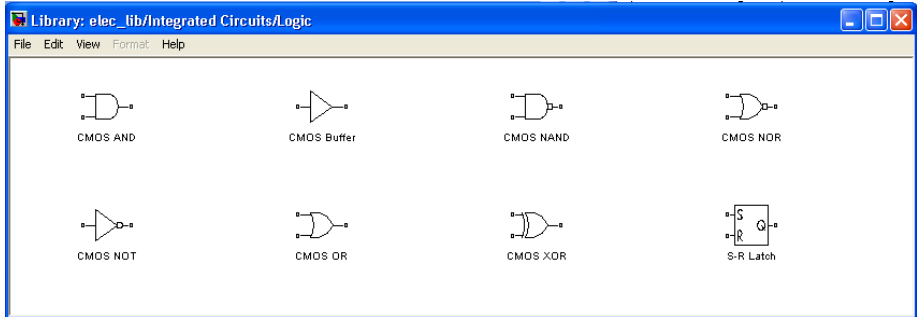
A première vue, les deux montages donnent la même réponse.

Si l'on fait un zoom horizontal, nous remarquons qu'avec une étendue d'abscisse plus large, la réponse du signal de sortie de l'amplificateur opérationnel à bande limitée accuse un fléchissement alors que pour celui à gain limité, le signal de sortie possède la même montée raide que le signal d'entrée.



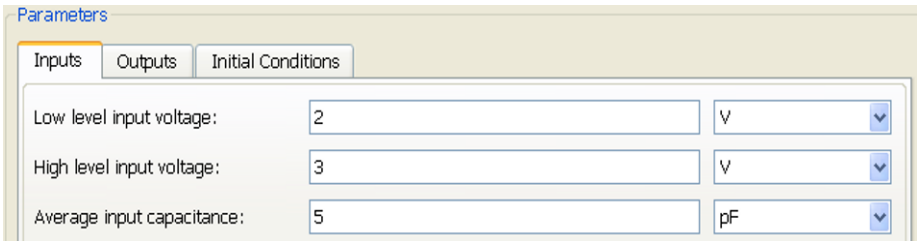
### III.2.2. Logic

La bibliothèque `Logic` comporte des portes logiques MOS complémentaires, CMOS.

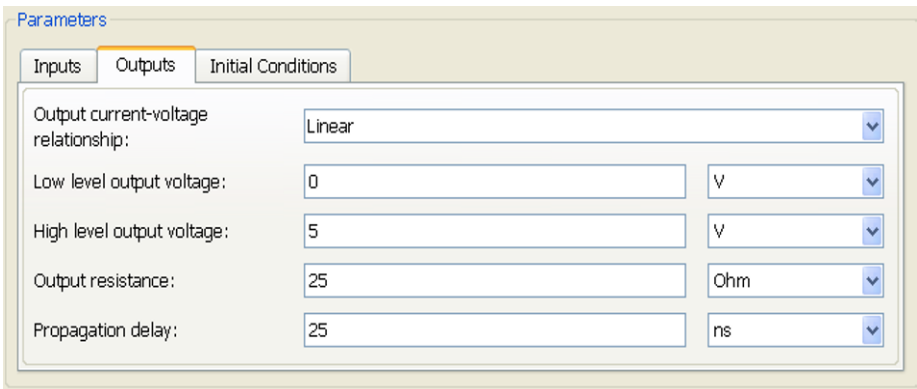


En double-cliquant sur le bloc de la porte ET CMOS AND par exemple, nous obtenons la boîte de dialogue suivante à 2 onglets où l'on définit les caractéristiques des signaux d'entrée et de sortie.

À l'entrée de la porte, en dessous de 2V, le signal est interprété comme étant du niveau bas (0 logique) et au-dessus de 3V, il est considéré comme étant du niveau (1 logique).

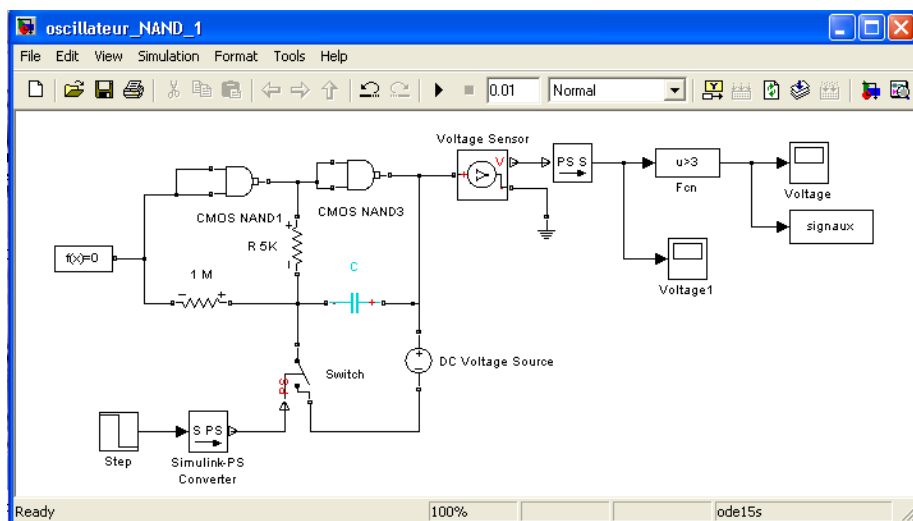


En sortie, les niveaux de la tension, correspondants aux niveaux logiques 0 et 1 sont de 0V et 5V.

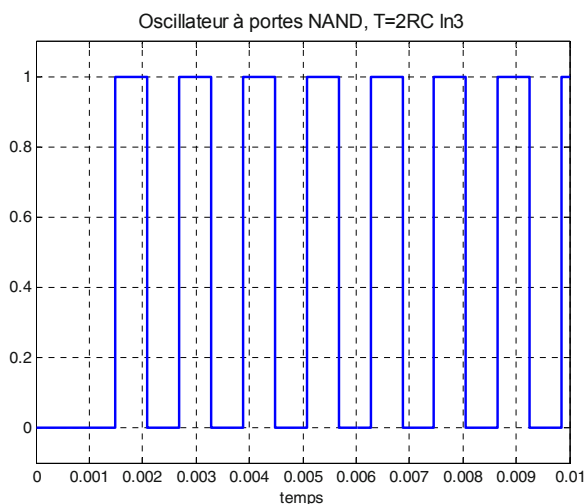


### • Oscillateur astable à portes logiques

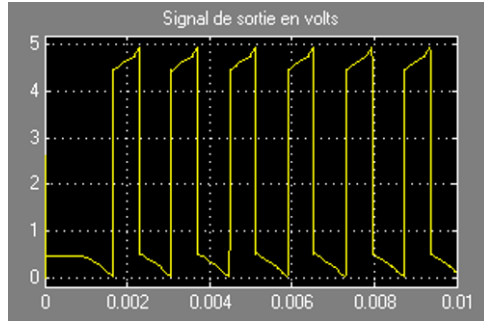
Le modèle suivant permet de réaliser un oscillateur à l'aide des portes NAND. On peut aussi utiliser des portes NOR, car le but de ces portes est de réaliser une inversion. La résistance de  $1M\Omega$  dite de protection permet de limiter le courant. La période d'oscillation est donnée par :  $T \approx 2RC \ln(3)$ . Dans le cas pratique ci-dessous,  $T = 2 \cdot 10^{-4} = 0.2 \text{ ms}$ .



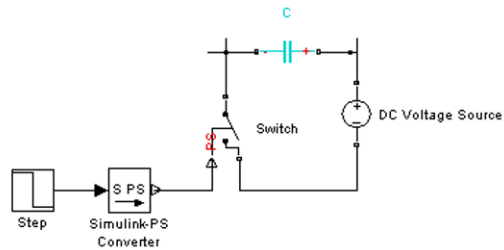
Le bloc  $F_{cn}(u>3)$  permet d'avoir des valeurs logiques à partir de la tension de sortie de la deuxième porte logique, 1 si la tension est supérieure à 3V et 0 dans le cas contraire. Nous vérifions bien la valeur de la période théorique de 0.2 ms.



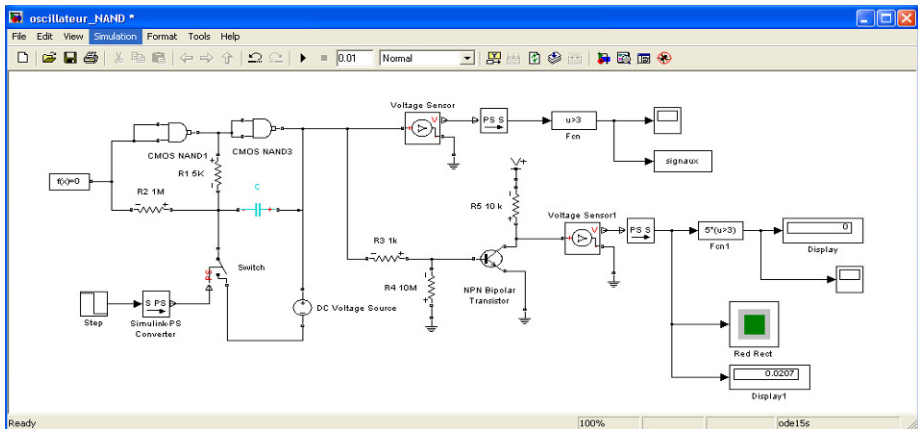
La tension de sortie en volts est donnée par l'oscilloscope Voltage1.



Comme dans les applications précédentes, on utilise un relais pour appliquer une tension continue aux bornes de la capacité. Le relais est fermé pendant 1 ms au départ puis reste ensuite ouvert.



La charge de départ de la capacité peut être spécifiée plus simplement en précisant sa charge initiale dans sa boîte de dialogue dans le champ *Initial voltage*.



On utilise un transistor dans le régime saturé/bloqué.

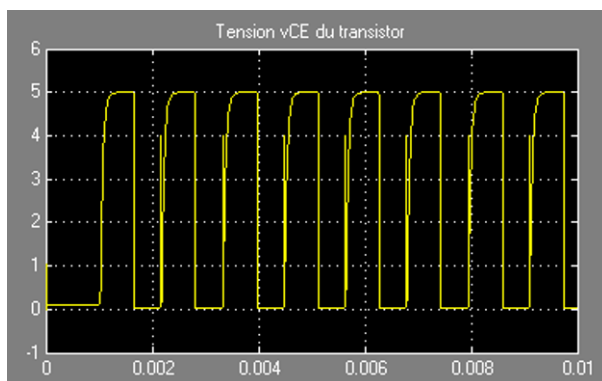
Lorsque la sortie du système est à l'état 1, le courant de base du transistor est imposé par la résistance  $R_3$  par  $\frac{V_{sat}}{R_3} \approx 3\text{ mA}$  au minimum, ce qui permet de saturer le transistor.



Les blocs `Fcn` et `Fcn1` permettent, respectivement, d'afficher les valeurs 1/0 et 5/0 selon l'état haut et bas de la sortie du système.

L'état 1 ou 0 est aussi représenté par la LED « Red Rect » de l'outil additionnel Gauges Blockset.

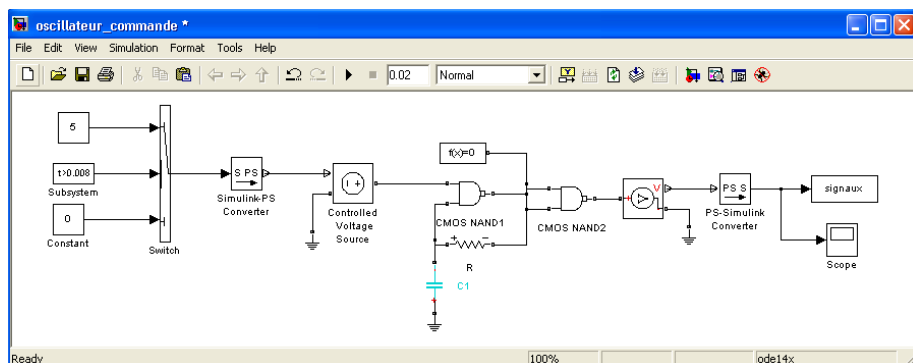
La tension de sortie au niveau du collecteur du transistor est visualisée dans l'oscilloscope `Voltage1` suivant.



### • *Oscillateur triangulaire commandé*

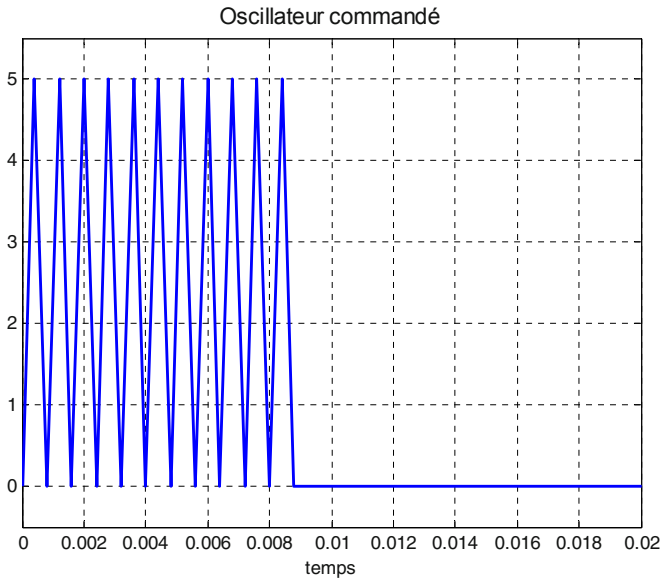
Avant l'instant  $t=0.008s$ , la condition logique  $t>0.008$  est fausse, le switch reçoit alors un 0 logique à sa 2<sup>ème</sup> entrée. C'est alors la 1<sup>ère</sup> entrée du switch qui commande l'état de sortie. La constante `PS Constant` qui vaut 5 permet de mettre un 1 logique à la 1<sup>ère</sup> entrée de la porte `CMOS NAND1`.

La capacité, non chargée au départ, se charge à travers la résistance `R` jusqu'à mettre un 1 logique sur la 2<sup>ème</sup> entrée de cette porte logique. La sortie se met alors à 0. La capacité se décharge et ainsi de suite.

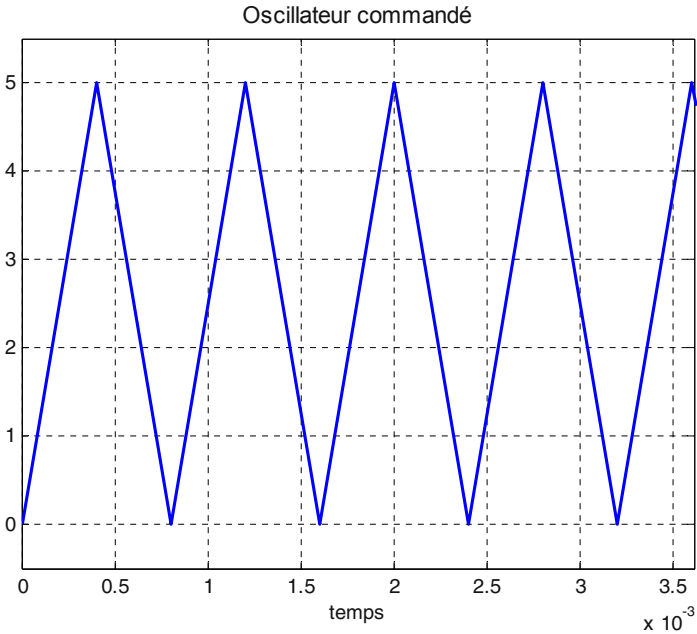


Après l'instant  $t=0.008$ , la valeur logique 0 est constamment appliquée à la 2<sup>ème</sup> entrée de la porte `NAND`.

Quelle que soit la valeur logique prise par la charge de la capacité, la sortie vaut toujours 1. Par la présence de la porte CMOS NAND2, la sortie est toujours égale à 0 comme le montre la figure suivante.

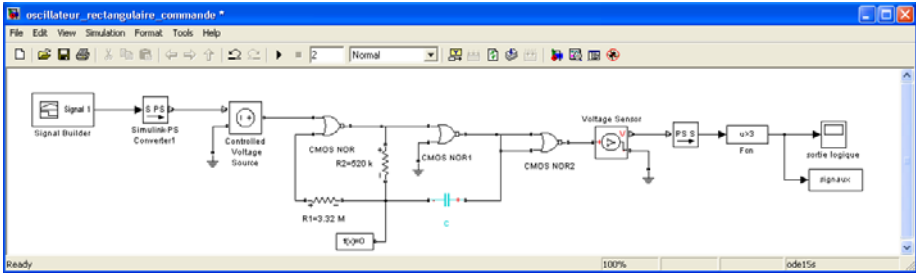


Avec les valeurs des composants du circuit RC, le signal est de forme quasi triangulaire, comme le montre la figure suivante, obtenue après un zoom horizontal.

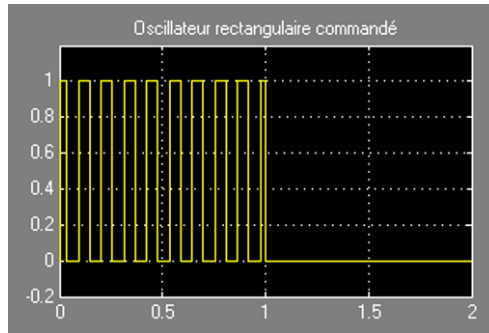


### • Oscillateur rectangulaire commandé

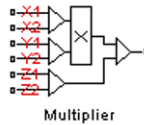
Le modèle suivant correspond à cet oscillateur utilisant les portes NOR.



L'oscilloscope suivant montre le signal obtenu en sortie.



### III.2.3. Multiplieur



L'entrée de ce circuit consiste en 3 paires de signaux X, Y et Z. Les paires  $(x_1, x_2)$ ,  $(y_1, y_2)$  et  $(z_1, z_2)$  sont des entrées de différentiateurs. Les sorties des 2 premiers différentiateurs sont multipliées et le résultat divisé par un facteur K.

A ce résultat on soustrait la sortie du 3<sup>ème</sup> différentiateur. Le résultat final est ensuite multiplié par un gain A.

Le signal de sortie est alors donné par l'expression suivante :

$$V_s = A \left\{ \frac{(x_1 - x_2)(y_1 - y_2)}{K} - (z_1 - z_2) \right\}$$

Les signaux  $x_i$ ,  $y_i$  et  $z_i$  sont des tensions d'entrée du multiplieur, A est le gain et K un facteur d'échelle.

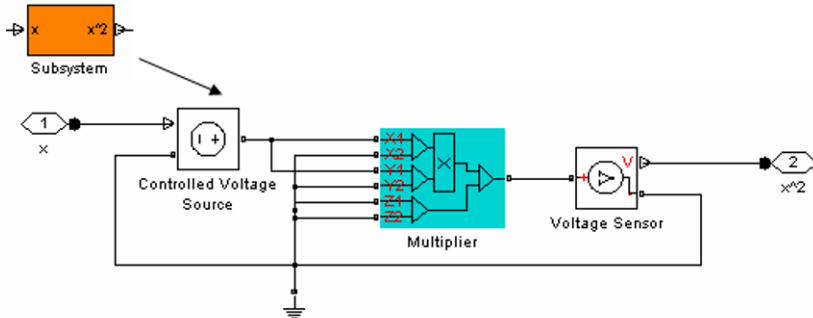
Ce bloc permet d'effectuer de nombreuses opérations.

L'application suivante effectue l'élévation au carré d'un signal. Pour cela, on relie ensemble les entrées  $x_1$  et  $y_1$  pour former le même signal d'entrée  $u$ , pendant qu'on relie toutes les autres entrées à la masse. L'expression précédente devient alors :  $V_s = A \frac{u^2}{K}$ , avec  $u = x_1 = y_1$ .

Si l'on choisit  $A = K$ , nous obtenons parfaitement  $V_s = u^2$ .

En double-cliquant sur ce bloc, on trouve divers onglets, dans lesquels on peut spécifier les valeurs du gain  $A$ , du facteur d'échelle  $K$  et les paramètres liés aux amplificateurs opérationnels (saturation, le slew rate, etc.).

Ce composant qui élève un signal au carré est programmé dans le sous-système suivant :

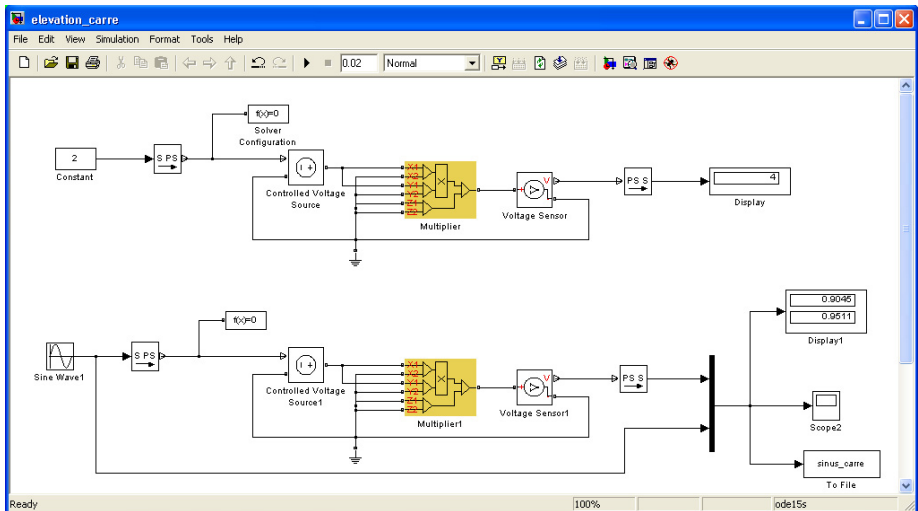


#### ▲ Élévation au carré d'un signal

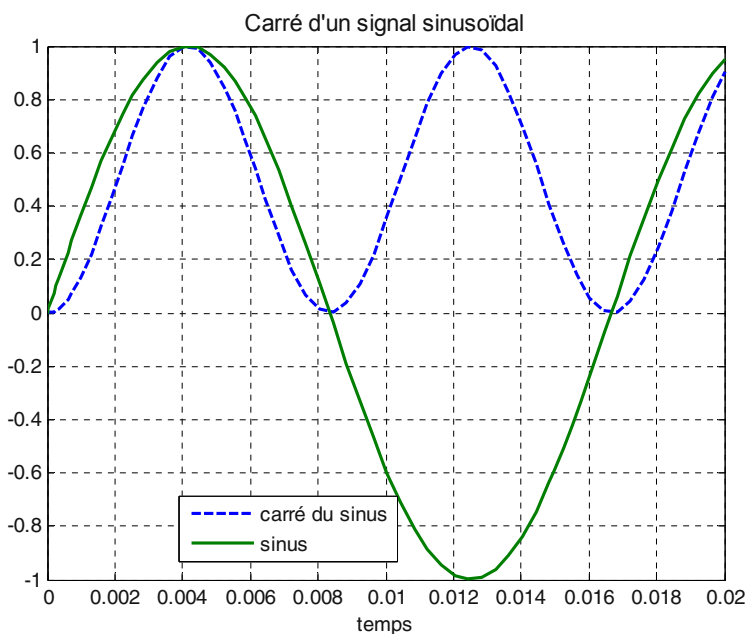
L'application à un signal constant donne parfaitement son carré.

On l'applique ensuite à l'élévation d'un signal variant dans le temps, comme la fonction sinusoïdale.

Le signal  $\sin^2(t)$  est affiché dans le même oscilloscope que la sinusoïde d'origine. Le modèle Simulink suivant représente ces 2 applications (carré d'une constante et du signal sinusoïdal).

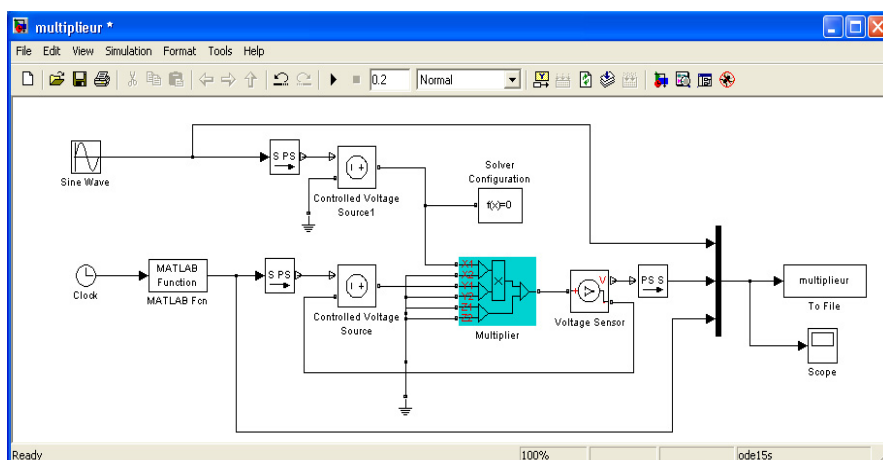


La courbe suivante représente une sinusoïde et son carré par l'utilisation du bloc Multiplier.



### • Amplificateur à gain variable

Dans cette application nous réalisons un amplificateur de tension dont le gain peut être modifié par les valeurs d'un autre signal.



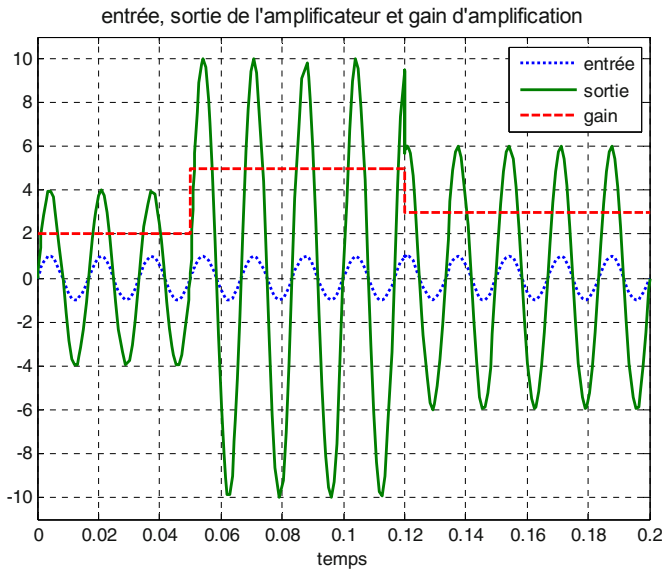
Toutes les entrées du multiplieur sont branchées à la masse sauf l'entrée  $x_1$  qui reçoit une sinusoïde servant d'entrée de l'amplificateur à gain variable.

La valeur du gain est calculé par le bloc Matlab Function en fonction du temps d'horloge par l'expression relationnelle suivante :

$$2*(u \leq 0.05) + 5*((0.05 < u) \& (u \leq 0.12)) + 3*(u > 0.12)$$

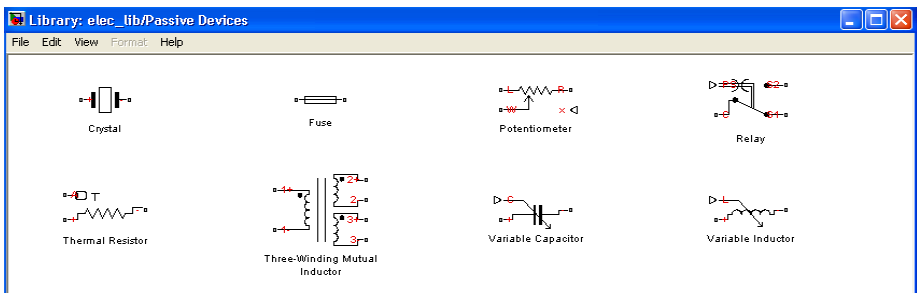
$$\begin{cases} 2 & t \leq 0.05 \\ 5 & 0.05 < t \leq 0.12 \\ 3 & 0.12 < t \end{cases}$$

Cette expression relationnelle permet d'avoir le signal suivant, en fonction du temps :



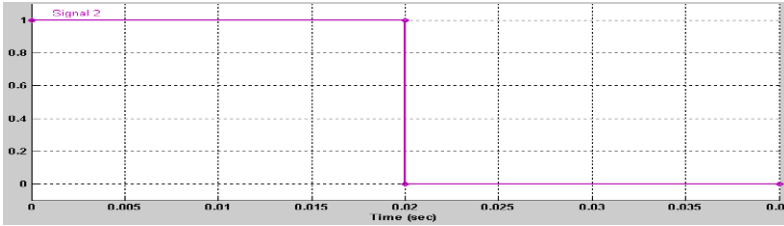
### III.3. Passive Devices

Dans cette librairie de composants passifs, nous trouvons, entre autres, un quartz, des interrupteurs, un relais, une thermistance, etc.

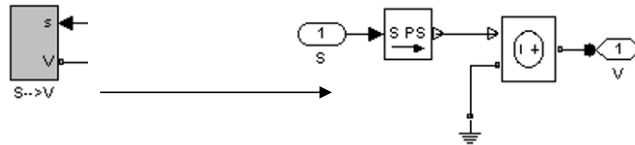


### III.3.1. Switch commandé en tension

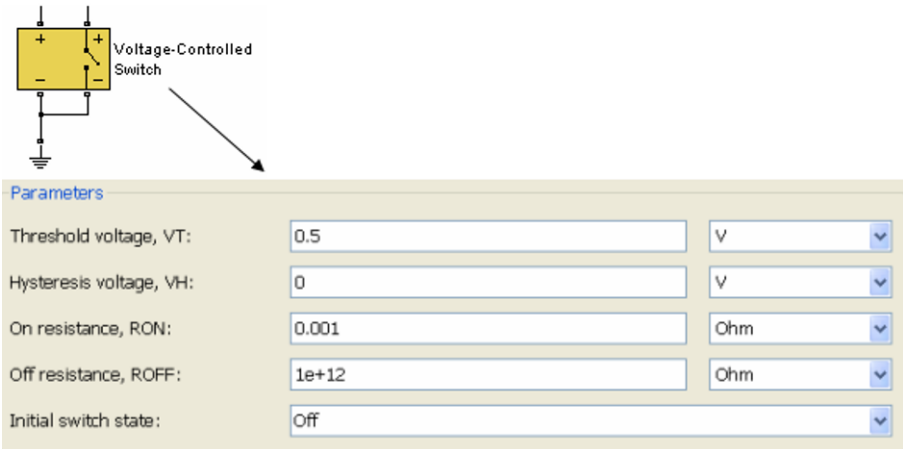
Le modèle suivant permet d'utiliser le switch commandé en tension. Le signal de commande est généré par le bloc Signal Builder de Simulink.



Ce signal est transformé en tension par le sous-système  $S \rightarrow V$ .



Pour mettre en parallèle les deux résistances  $R_2$  et  $R_3$  nous utilisons le switch commandé en tension suivant dont nous avons spécifié le seuil à 0.5 V. Ce switch appartient à la librairie SPICE-Compatible Components/Passive Devices. Ce switch présente une résistance que nous avons spécifiée, respectivement, à 0.001 lorsqu'il fermé et à  $10^{12} \Omega$  quand il est ouvert.



Le switch est, initialement, initialisé à Off donc ouvert.

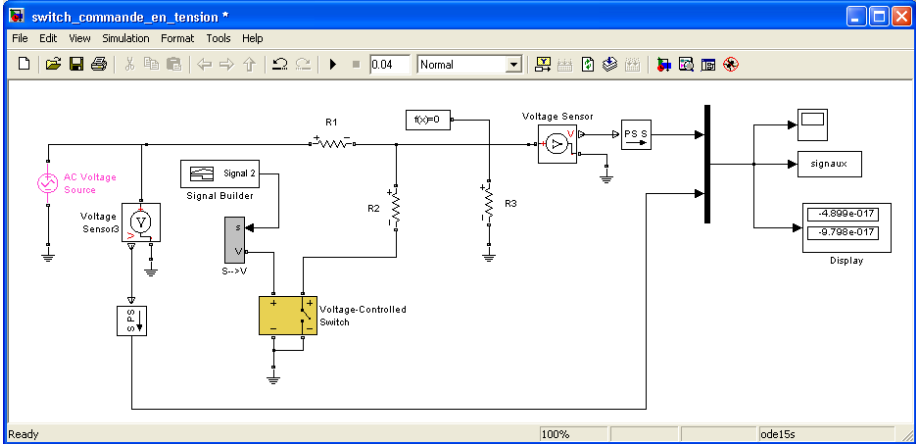
Ce bloc est commandé par le signal Signal2 provenant du bloc Signal Builder, un créneau de 1 V entre les instants  $t=0$  et 0.02.

Ce signal est appliqué à une résistance  $R_3$  de  $10k\Omega$ . Cette résistance est ainsi mise en parallèle à la résistance  $R_2$  aux bornes de laquelle la tension possède le 1/4 de l'amplitude du signal sinusoïdal.

En effet la tension à ses bornes vaut :

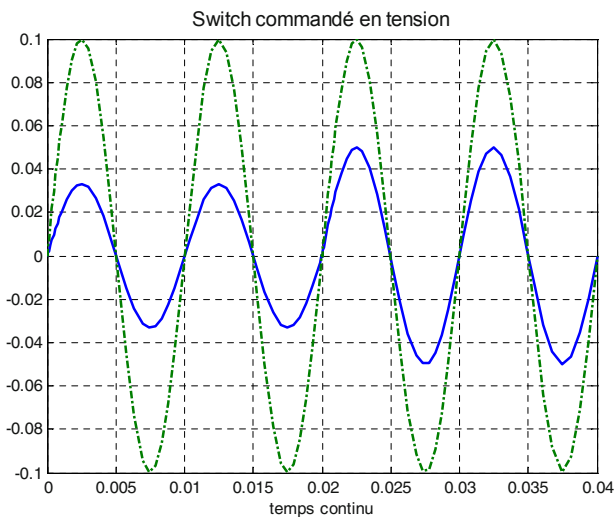
$$V_s = \frac{(R_2 // R_3)}{(R_2 // R_3) + R_1} = \frac{1}{4} V_e$$

Après l'instant 0.02, la résistance  $R_2$  n'est plus mise en parallèle à  $R_3$ , la tension  $V_s$  n'est plus le 1/4 mais la moitié de la tension d'entrée  $V_e$ .



Dans la fonction Callback `InitFcn`, nous spécifions les valeurs des différentes résistances et dans `StopFcn` nous traçons les tensions d'entrée et de sortie.

Nous remarquons bien, dans la figure suivante, que le signal de sortie est bien la moitié et non le quart du signal d'entrée, selon la mise en parallèle de  $R_2$  à  $R_3$  par le switch Voltage Controlled Switch.

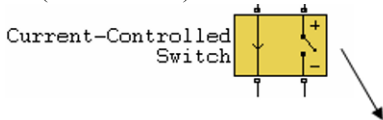




### III.3.2. Switch commandé en courant

Ce switch se ferme lorsque l'intensité du courant qui le traverse dépasse la valeur du seuil (threshold).

Il se comporte alors comme une résistance  $R_{on}$  dont on a spécifié la valeur ainsi que celle du seuil (Threshold).



Parameters		
Threshold current, IT:	0.5	mA
Hysteresis current, IH:	0	A
On resistance, RON:	0.001	Ohm
Off resistance, ROFF:	1e+12	Ohm
Initial switch state:	Off	

On choisit un seuil de 0.5 mA.

La résistance  $R$ , choisie égale à  $1\text{ k}\Omega$ , sera parcourue par un courant de 1 mA grâce à la tension de 1 V fournie par le bloc Signal Builder.

Dans le modèle suivant, nous avons un circuit dérivateur de résistance  $R=10\text{ k}\Omega$  et de capacité variable grâce au switch commandé en courant. Le switch commandé en courant permet de court-circuiter ou non la capacité  $C_1$ .

- à  $0 \leq t < 0.2$ , le switch est ouvert, les 2 capacités  $C_1$  et  $C_2$  sont en série.

Elles forment, alors, une capacité de valeur  $\frac{C_1 C_2}{C_1 + C_2} = 50\text{ nF}$ . La sortie du dérivateur est donnée par :

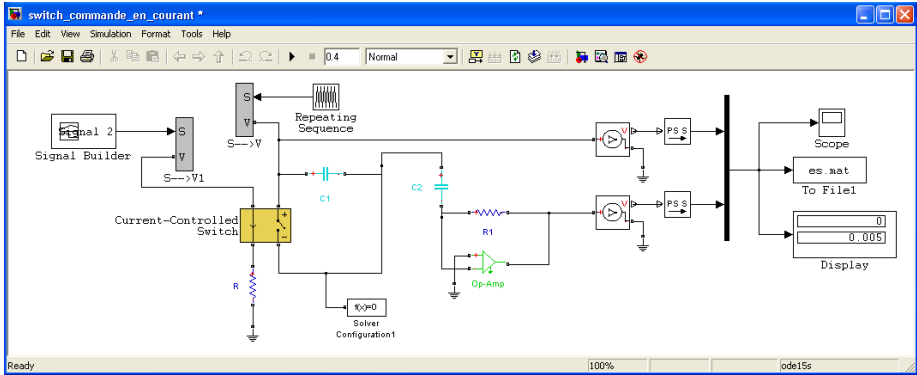
$$V_s = R \frac{C}{2} \frac{dV_e}{dt} = 10 * 10^3 * 50 * 10^{-9} \frac{dV_e}{dt} = 0.0005 \frac{dV_e}{dt}$$

- à  $t \geq 0.2$ , le switch est fermé, la capacité  $C_1$  est court-circuitée, il ne reste que  $C_2$ .

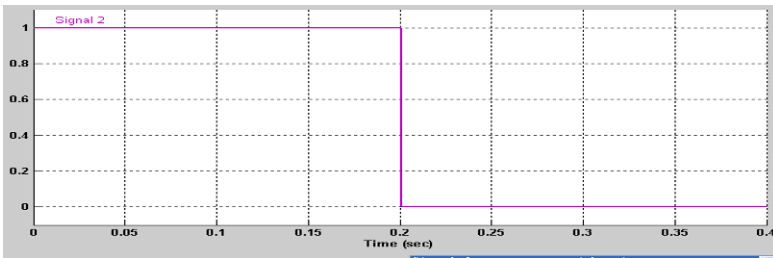
$$V_s = R C \frac{dV_e}{dt} = 10 * 10^3 * 100 * 10^{-9} \frac{dV_e}{dt} = 0.001 \frac{dV_e}{dt}$$

Le signal d'entrée est triangulaire de période 0.2s, sa dérivée est égale à 10 en valeur absolue.

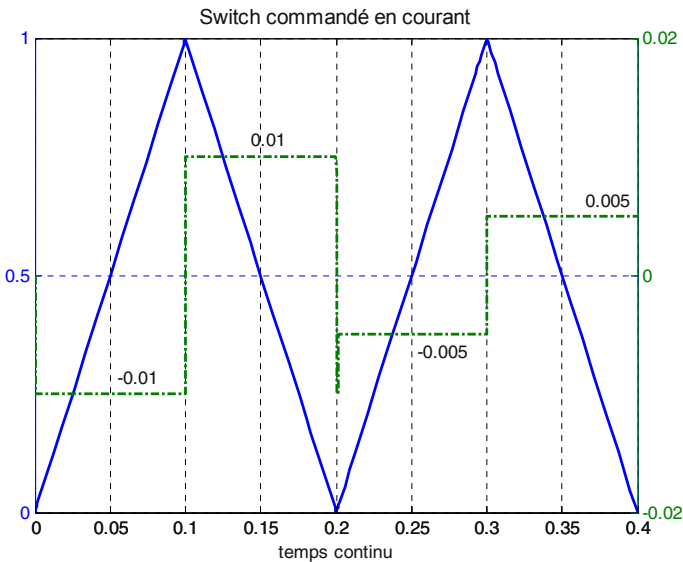
Ainsi, nous pouvons bien vérifier les valeurs du signal  $V_s$  que nous avons affichées sur les paliers des créneaux obtenus.



Dans cet exemple, le switch est commandé par le signal suivant généré par le bloc Signal Builder:

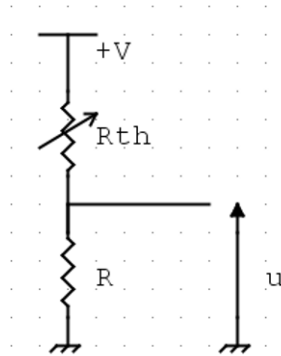


Entre les instants 0 et 0.2, on applique une tension de 1 V aux bornes de la résistance R de  $1k\Omega$ , soit 1 mA, le double de la valeur fixée pour le seuil ; ce qui a pour effet de le fermer.



### III.3.3. Mesure de température par une thermistance

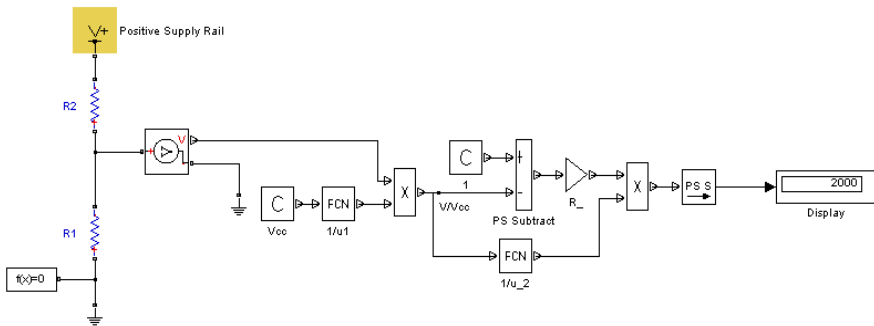
Dans beaucoup de cas de mesure de température, dans le monde de la mécatronique, on utilise la thermistance dans un pont diviseur de tension.



La mesure de la température consiste à mesurer la tension  $u = \frac{R}{R + R_{th}}$ .

Pour valider le calcul, on commencera à utiliser des résistances fixes.

Le but est de calculer la résistance  $R_2$  (qu'on remplacera par une thermistance pour laquelle, on calculera la résistance) à partir de la mesure de la tension aux bornes de  $R_1$ .



La tension aux bornes de  $R_1$  est donnée par :

$$V = \frac{R_1}{R_1 + R_2} V_{cc}$$

A partir de cette relation, la résistance  $R_2$  est donnée par :

$$R_2 = R_1 \left(1 - \frac{V}{V_{cc}}\right) \frac{V_{cc}}{V}$$

$V$  : tension mesurée aux bornes de  $R_1$

Dans la fonction Callback `InitFcn`, nous avons spécifié les résistances  $R_1$ ,  $R_2$  et  $V_{cc}$ .

```
R1=1e3; R2=2*R1;
Vcc=12; % tension d'alimentation
```



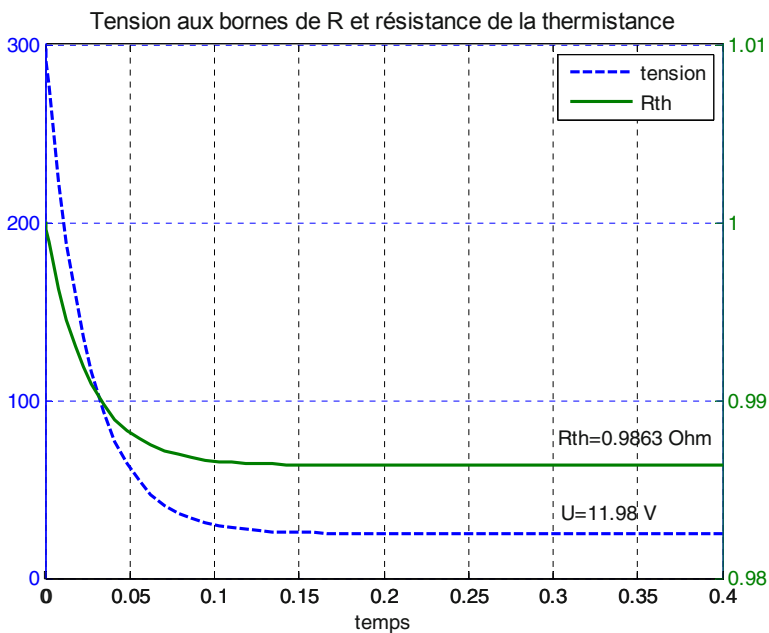
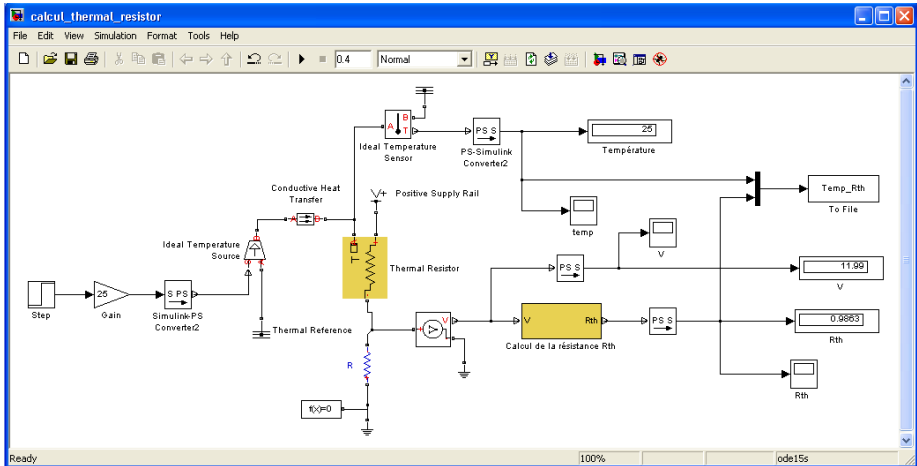
avec :

$T_0$  : température de référence, 25°C,

$\alpha$  : coefficient de température ( $5 \cdot 10^{-5} \text{ K}^{-1}$ ).

La thermistance constitue un pont diviseur avec la résistance R de  $10\text{ k}\Omega$ , et on obtient une tension de 11.99 V à ses bornes.

Cette tension atteint cette valeur avec une certaine dynamique due au délai du transfert par conduction et du temps de réponse de la résistance thermique.



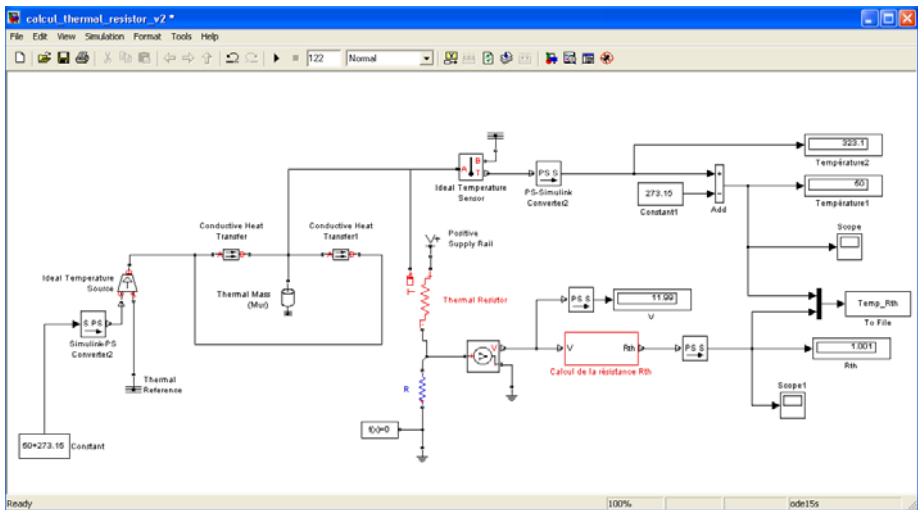
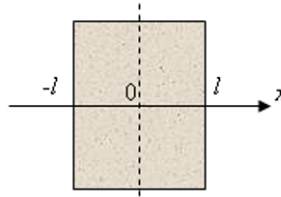
Nous avons utilisé le sous-système physique « Calcul de la résistance  $R_{th}$  » que nous avons créé précédemment pour un pont diviseur à résistances fixes.

### III.3.4. Choc thermique

Dans le modèle suivant, nous avons considéré le problème du choc thermique à température imposée en surface sur une barre métallique d'épaisseur  $2l$ .

Nous appliquons la même température de  $50^{\circ}\text{C}$  de chaque côté de la barre (abscisses  $-l$  et  $+l$ ). On mesure la température en  $x=0$ .

Le transfert se fait par conduction.



Nous obtenons une valeur quasiment identique ( $1.001\ \Omega$ ) à la température de  $50^{\circ}\text{C}$ , comme c'est spécifié précédemment dans l'expression de la résistance de la thermistance en fonction de la température :

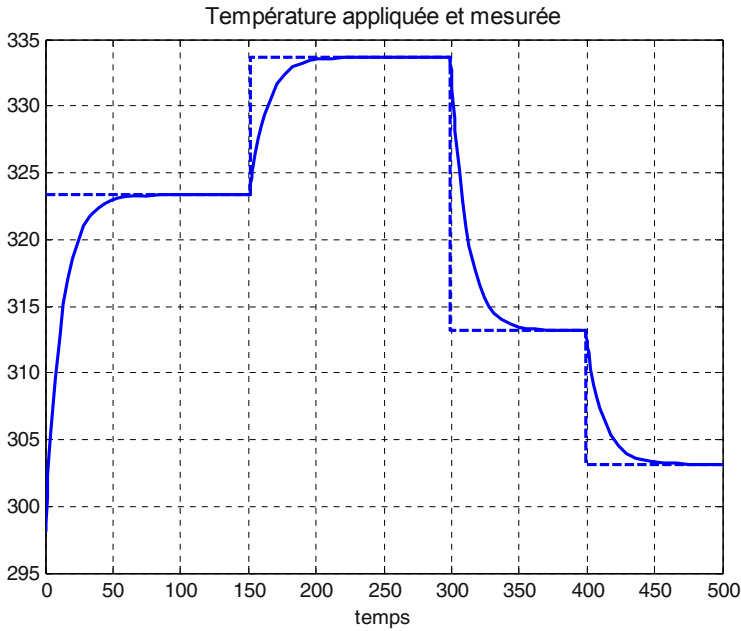
```
>> R50=1*(1+5e-5*(50-25))
```

```
R50 =  
1.0013
```

Nous retrouvons bien la valeur de la résistance à cette température de  $50^{\circ}\text{C}$ .



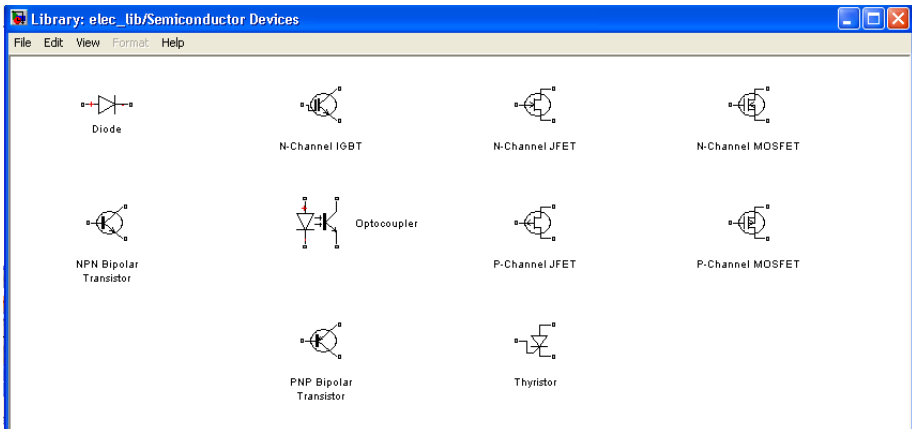
Dans la fonction Callback `StopFcn`, nous traçons les 2 courbes, températures, appliquée et mesurée :



Pour chaque échelon, nous observons une convergence de la température mesurée vers la valeur appliquée.

### III.4. Semiconductor Devices

Cette librairie contient les composants actifs semi-conducteurs tels les transistors bipolaires, NPN et PNP, les MOSFET à canal N et canal P, les transistors JFET, et passifs comme la diode et le thyristor.





### III.4.1. Le transistor JFET (Junction Field Effect Transistor)

Le JFET est un transistor à effet de champ à jonction PN, comme les transistors bipolaires. Il existe des JFET à canal N ou à canal P, comme pour les transistors bipolaires NPN ou PNP. Les transistors à effet de champ sont commandés par la tension  $V_{GS}$  (tension grille-source) alors que les transistors bipolaires le sont par leur courant de base  $i_b$ .

Les caractéristiques des JFET sont semblables à celles des transistors bipolaires, à part que le courant de base  $i_b$  est remplacé par la tension grille-source  $V_{GS}$ .

Ainsi, leur courant de drain  $I_D$  dépend de la tension  $V_{GS}$ , à comparer avec le courant de collecteur  $I_C$  dépendant du courant de base  $i_b$ .

La caractéristique  $I_D(V_{GS})$  est donnée par la relation suivante :  $I_D = I_{DSS} (1 - \frac{V_{GS}}{V_p})^2$

avec :

$I_{DSS}$  : courant de saturation ou de court-circuit ( $V_{GS}=0$ ),

$V_p$  : tension de pincement, tension  $V_{GS}$  qui annule le courant de drain  $I_D$ .

#### • JFET comme source de courant d'une LED d'un optocoupleur

Dans l'application qui suit, nous utilisons le transistor JFET (Junction Field Effect Transistor ou Transistor à effet de champ à jonction) comme source de courant afin d'alimenter la LED d'un optocoupleur à courant constant de 10 mA.

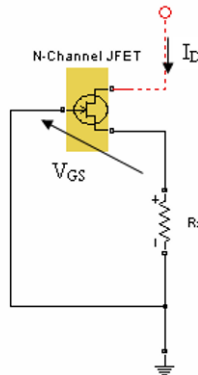
On choisit le transistor dans un catalogue et dans son datasheet, on s'intéresse à la caractéristique  $I_D(V_{GS})$ . Dans celui qu'on a choisi, ce courant est limité par la valeur maximale de 20 mA et minimale de 2 mA.

D'après le datasheet du transistor, pour un courant  $I_D = 10$  mA, correspond la tension  $V_{GS}$  de -1,8 V.

On obtient alors la valeur de la résistance  $R_s$  :  $R_s = -\frac{V_{GS}}{I_D} = -\frac{-1,8}{0,01} = 180 \Omega$

Dans cette application, nous alimentons le circuit à une valeur d'alimentation  $V_{CC}$  variable. Le but étant que la LED soit alimentée par un courant constant.

Si on observe le schéma du JFET, nous avons :



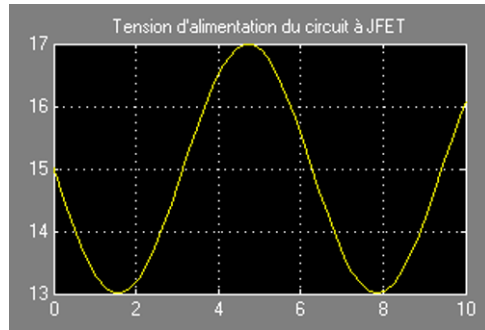
Avec  $V_{GS} = -R_s I_D$ , la polarisation du JFET est assurée par la résistance  $R_s$  et le courant de drain.

Cette polarisation est dite automatique. En effet, si le courant de drain augmente, ceci a pour effet d'augmenter négativement  $V_{GS}$ , ce qui diminue le courant  $I_D$ .

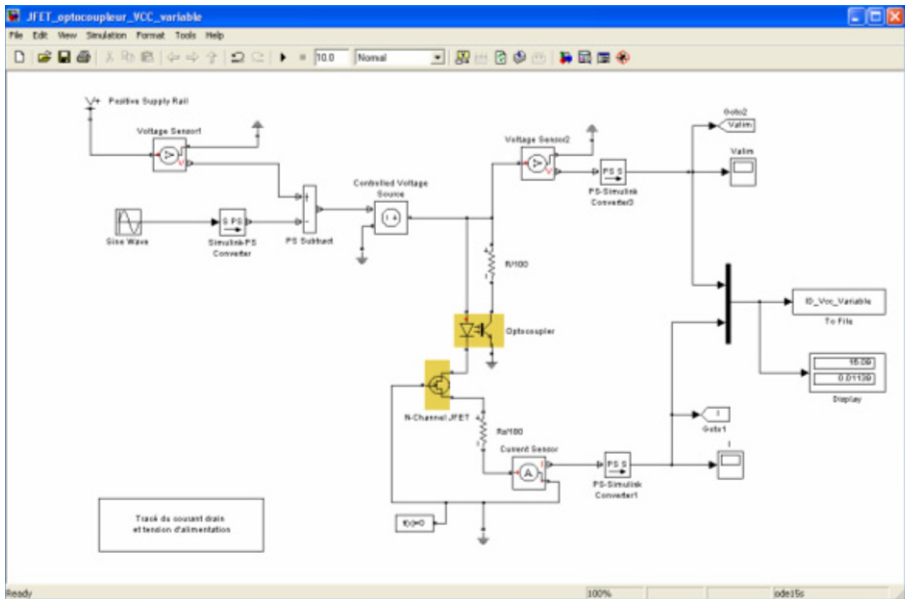
On obtient donc un effet de régulation du courant de drain.

Dans le modèle suivant, le montage est alimenté par une tension continue de 15 V Positive Supply Rail à laquelle on soustrait une sinusoïde d'amplitude 2V.

Le circuit est alors alimenté par la tension variable (13 à 17 V). Cette tension est affichée dans l'oscilloscope Valim.

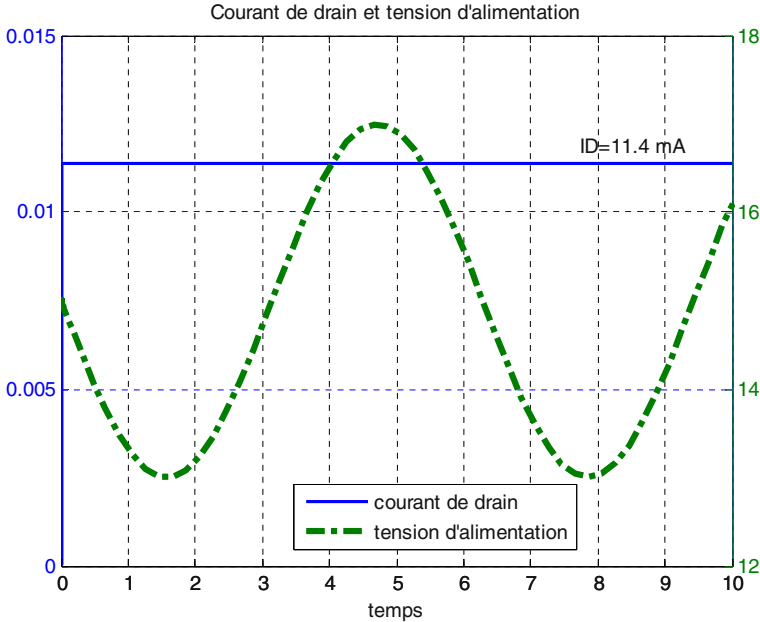


Le courant de drain,  $I_D$  traverse la diode de l'optocoupleur.



Le courant  $I_D$  est mesuré par un capteur de courant *Current Sensor* au niveau de la résistance  $R_S$ .

La figure suivante représente les variations de la tension d'alimentation du circuit ainsi que le courant de drain.



On remarque que malgré les variations sinusoïdales de la tension d'alimentation, le courant de drain reste constant, égal à 11.4 mA.

▲ *JFET comme source de courant d'induit, tension d'alimentation fixe de 15V*

Dans cette application, la source de courant réalisée par le transistor JFET attaque l'induit d'un moteur à courant continu.

La tension d'alimentation est d'abord choisie constante et égale à 15 V.

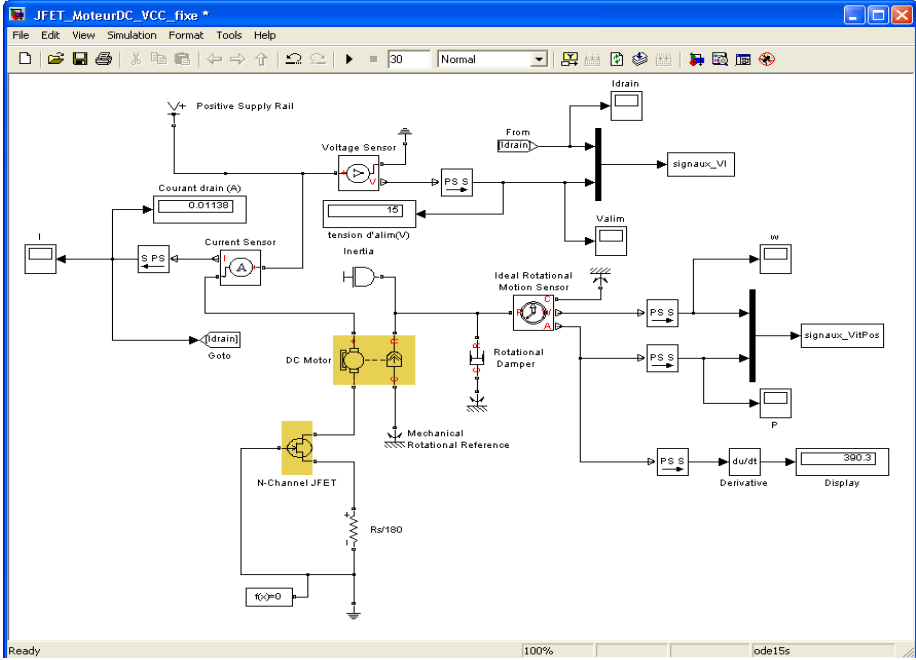
Dans l'afficheur `Courant drain`, on observe la valeur du courant de drain, égale à 11.38 mA, le même ordre de grandeur que dans l'exemple de l'optocoupleur. Le moteur est chargé mécaniquement par une inertie (`Inertia`,  $1e-7$  kg\*m<sup>2</sup>) et un amortisseur (`Rotational Damper`,  $1e-8$  N\*m/(rad/s)).

Dans le modèle Simulink ci-dessous, nous avons calculé la dérivée de la position que nous affichons dans `Display` dont la valeur est 390.3 rad/s et que nous pouvons vérifier avec la valeur en régime permanent de la courbe de vitesse, laquelle est égale à 391.2281 rad/s.

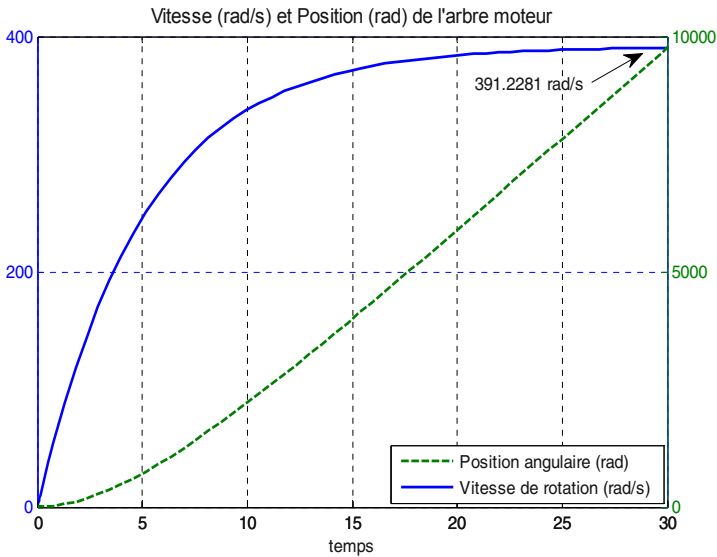
La position angulaire et la vitesse de rotation sont enregistrées dans le fichier binaire `signaux_VitPos.mat`.

Les deux valeurs, calculée et obtenue graphiquement, sont quasiment identiques.

La vitesse et la position sont enregistrées dans le fichier binaire `signaux_VitPos.mat`. Dans la fonction Callback `StopFcn`, on lit ce fichier et on trace les courbes de la position angulaire et la vitesse de rotation.



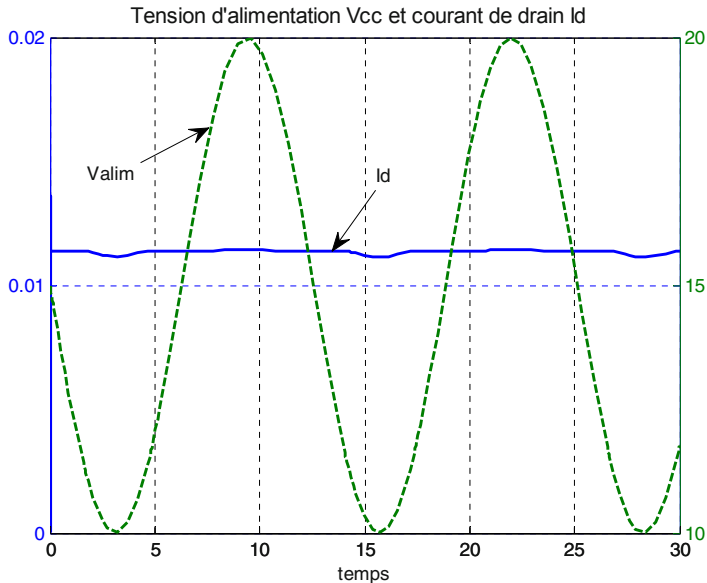
Ci-après, nous traçons les courbes de vitesse et de position angulaires.



#### • Tension d'alimentation sinusoïdale

Dans cette application, on ajoute à la tension fixe de 15 V, une sinusoïde d'amplitude 5V et de pulsation de 1 rad/s.

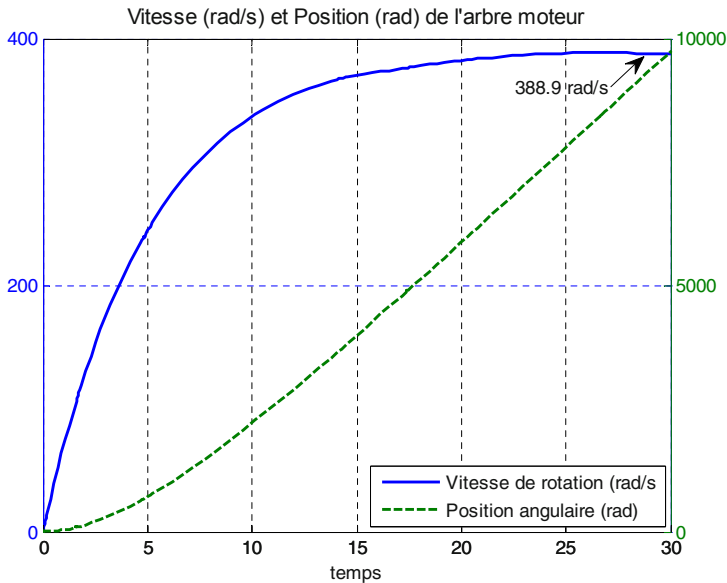




Pour ces mêmes variations de 30% de la tension d'alimentation, la variation relative de la vitesse de rotation est égale à :

$$\frac{\Delta w}{w} = \frac{391.23 - 388.9}{391.23} = 0.6\%$$

La figure suivante représente la vitesse de rotation et la position angulaire de l'arbre moteur :



### III.4.2. Transistor MOSFET, contrôle d'un moteur DC

Les inconvénients des transistors bipolaires sont :

- Courant de base non nul d'où consommation notable,
- Risque d'emballement thermique,
- Pilotage en courant (pas en tension).

Le transistor MOSFET possède les avantages suivants:

- Consommation d'énergie très réduite en commutation,
- Pilotable en tension,
- Pas de risque d'emballement.

Les transistors MOSFET (Métal-Oxyde-Semiconducteur Field Effect Transistor) possèdent des vitesses élevées de commutation et peuvent contrôler des moteurs à courant continu ou des moteurs pas à pas.

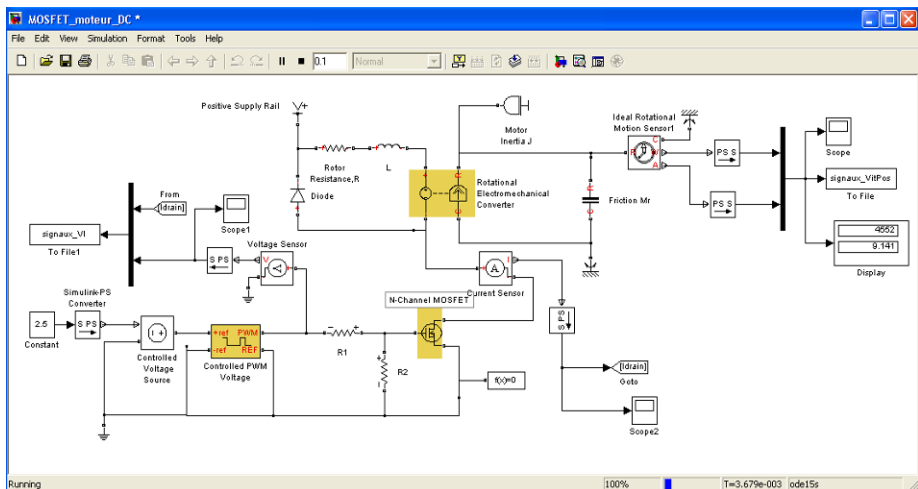
Ils sont utilisés comme de parfaits switches (commutateurs) dans le cas de commande d'un moteur par signal PWM.

Dans le modèle suivant, l'induit du moteur DC est directement branché sur le drain du MOSFET.

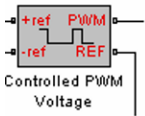
Nous avons placé en parallèle de cet inductif une diode de roue libre pour dévier la f.e.m. générée par le moteur lorsque le MOSFET l'arrête.

Nous avons placé un ampèremètre pour mesurer et tracer le courant d'induit (courant du drain) en même temps que la vitesse et l'angle de rotation.

Le moteur est soumis à un couple d'inertie et de frottements.



La constante de valeur 2.5 est transformée en tension pour imposer un rapport cyclique de 50% du fait que nous avons spécifié dans le générateur PWM que 5 V et 0V correspondent, respectivement, à 100 et 0 % de rapport cyclique.

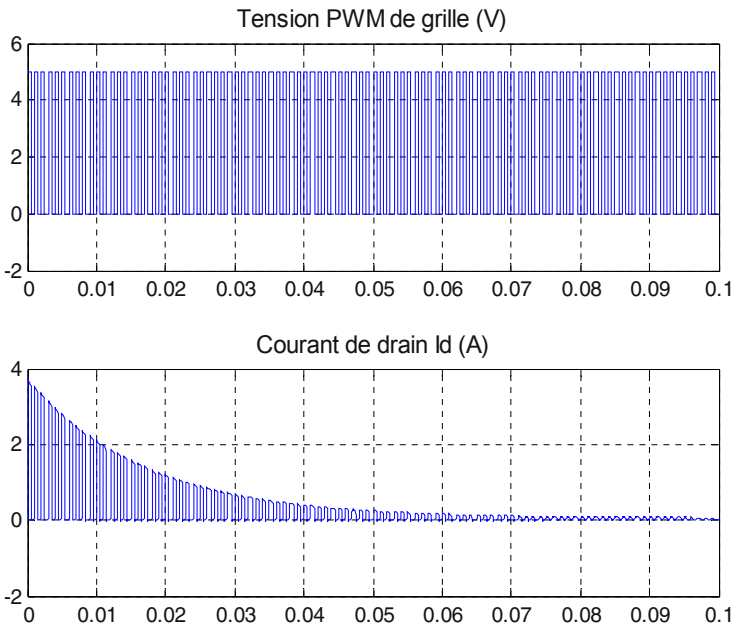


**Parameters**

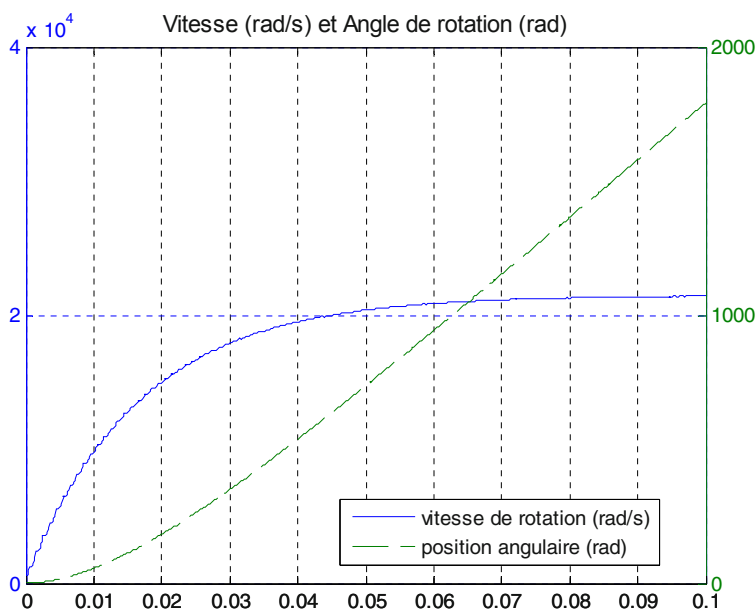
PWM frequency:	1000	Hz
Input value Vmin for 0% duty cycle:	0	V
Input value Vmax for 100% duty cycle:	5	V
Output voltage amplitude:	5	V
Simulation mode:	PWM	

Le signal de commande est du type PWM, de fréquence 1 kHz, avec 5 V pour 100% et 0 V pour 0% de rapport cyclique.

Les courbes suivantes, de la vitesse et de l'angle de rotation ainsi que le courant d'induit, sont tracées dans la fonction Callback `StopFcn` (fin de simulation).

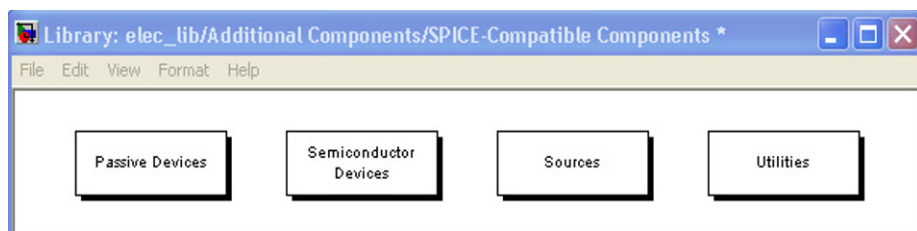






### III.5. Additional Components

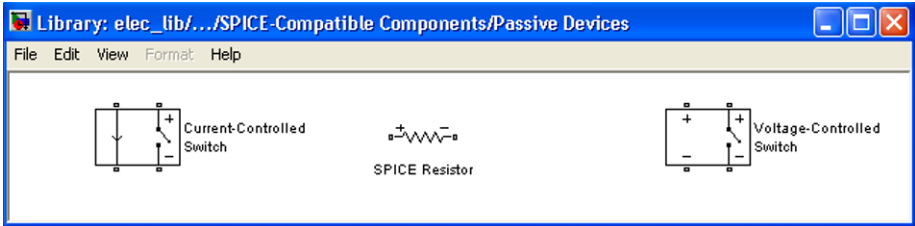
Cette librairie contient des composants additionnels, notamment des composants compatibles SPICE.



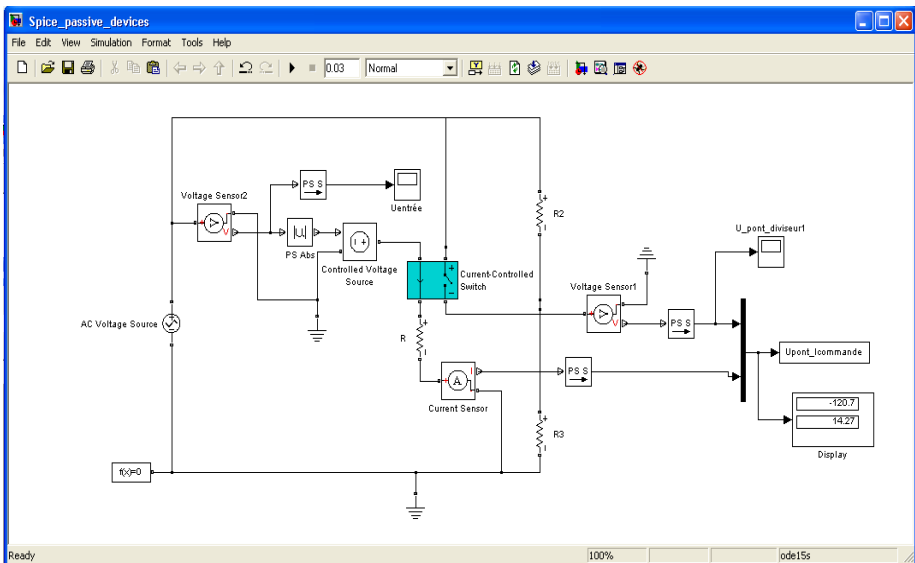
SPICE (Simulation Program with Integrated Circuit Emphasis) est un programme "open source", un simulateur puissant de circuits électroniques.

### III.5.1. Passive Devices

Les composants passifs compatibles SPICE de cette librairie sont la résistance, 2 switches commandés, respectivement en courant et en tension avec hystérésis.



Dans l'application qui suit, nous utilisons le switch commandé en courant et une résistance, compatibles SPICE.

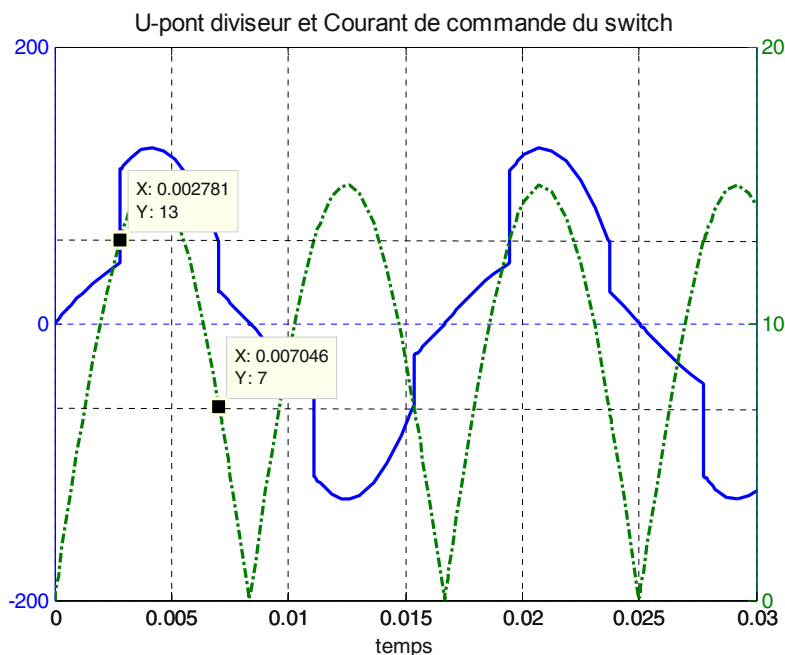


Le courant de commande du switch est celui qui passe dans la résistance R lorsqu'on lui applique la valeur absolue de la tension sinusoïdale du générateur AC Voltage Source (amplitude 150 V, fréquence 60 Hz).

La résistance R, valant  $10\Omega$ , le courant est alors une sinusoïde redressée en double alternance d'amplitude 15 A, comme on l'observe dans la figure suivante.

La figure suivante représente le courant de commande (en Ampères) du switch ainsi que la tension au niveau du pont diviseur.

Le switch est défini par un seuil (Threshold current,  $I_T$ ) et une valeur d'hystérésis (Hysteresis current,  $I_H$ ).



La boîte de dialogue de ce switch commandé est la suivante.

Parameters		
Threshold current, IT:	<input type="text" value="10"/>	A <input type="button" value="v"/>
Hysteresis current, IH:	<input type="text" value="3"/>	A <input type="button" value="v"/>
On resistance, RON:	<input type="text" value="1"/>	Ohm <input type="button" value="v"/>
Off resistance, ROFF:	<input type="text" value="1e+12"/>	Ohm <input type="button" value="v"/>
Initial switch state:	<input type="button" value="Off"/>	

Le switch se comporte comme une résistance très faible ( $R_{on} = 1 \Omega$ ) lorsque le contact est fermé et une résistance très élevée ( $R_{off} = 10^{12} \Omega$ ) lorsque le contact est ouvert.

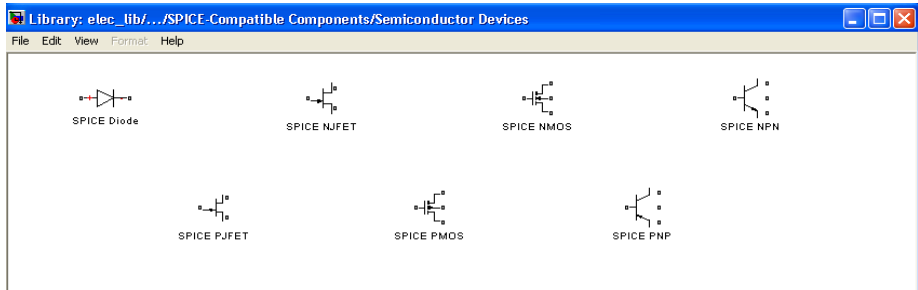
Le contact se ferme si l'intensité du courant de commande (ici la sinusoïde de 15 A d'amplitude, redressée) devient plus grande que la valeur du seuil et de l'hystérésis, respectivement de 10 et 3 A, soit 13 A.

Si le courant de contrôle devient plus faible que la différence du courant de seuil et du courant d'hystérésis alors le commutateur s'ouvre et sa valeur de résistance est égale à  $R_{off}$ .

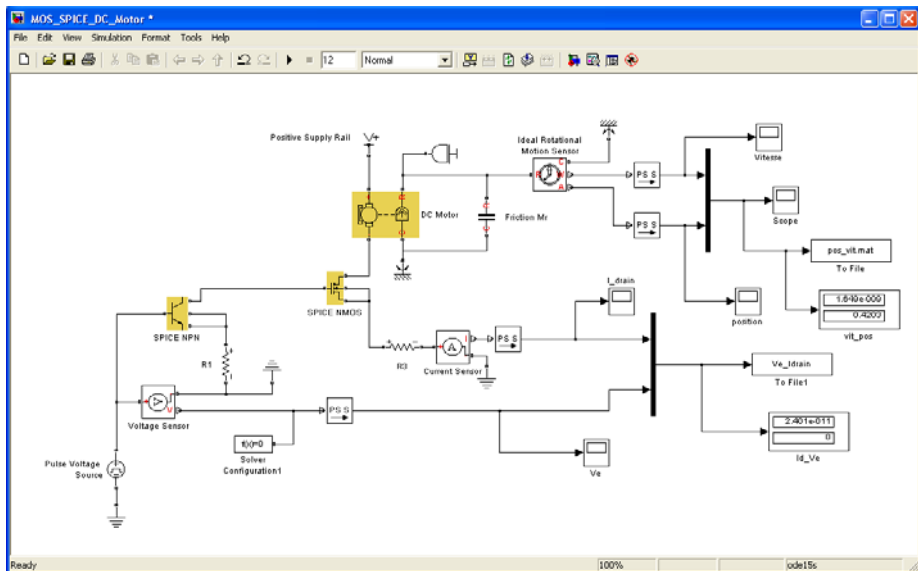
Entre ces deux valeurs, la position du commutateur est inchangée.

### III.5.2. Semiconductor Devices

Cette librairie contient des composants semi-conducteurs de type SPICE, tels des transistors bipolaires (SPICE NPN, SPICE PNP), des JFET canal N (SPICE NJFET) et canal P (SPICE PJFET) et des transistors MOS (SPICE NMOS), canal et canal P (SPICE PMOS)



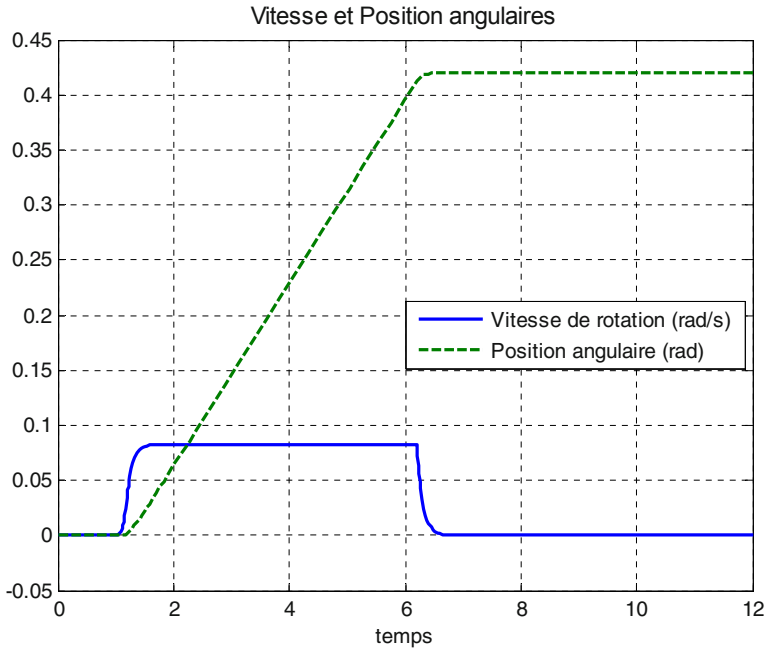
Dans l'application qui suit, nous associons un transistor bipolaire NPN et un NMOS pour commander l'induit d'un moteur à courant continu.



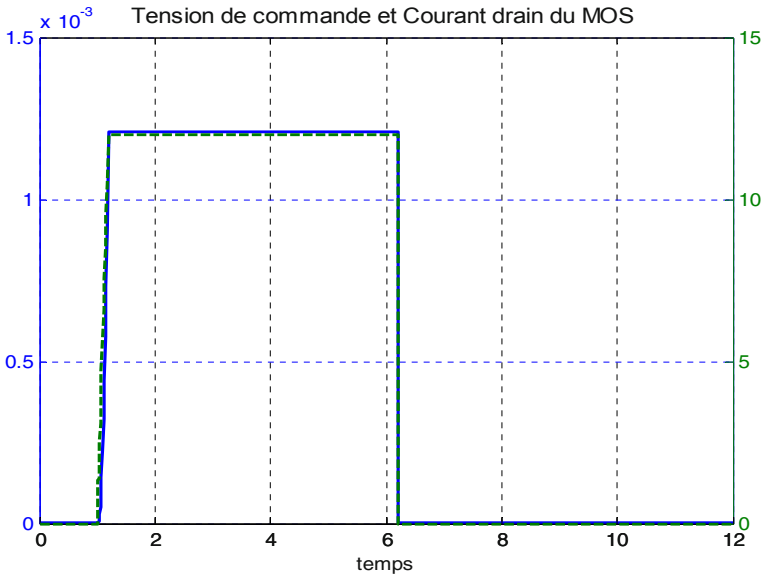
La tension de commande est un créneau d'amplitude 12 V, de largeur 5s (Pulse width, PW), un retard de 1s (Pulse delay time, TD) et un temps de montée de 0.2s (Pulse rise time, TR).

Tous ces paramètres sont fixés dans la boîte de dialogue du générateur Pulse Voltage Source).

La courbe qui suit représente l'évolution de la vitesse de rotation et de la position angulaire de l'arbre du moteur DC.



La figure suivante représente les courbes, dans deux axes d'ordonnées différents, du courant du drain du transistor NMOS et de la tension d'entrée.



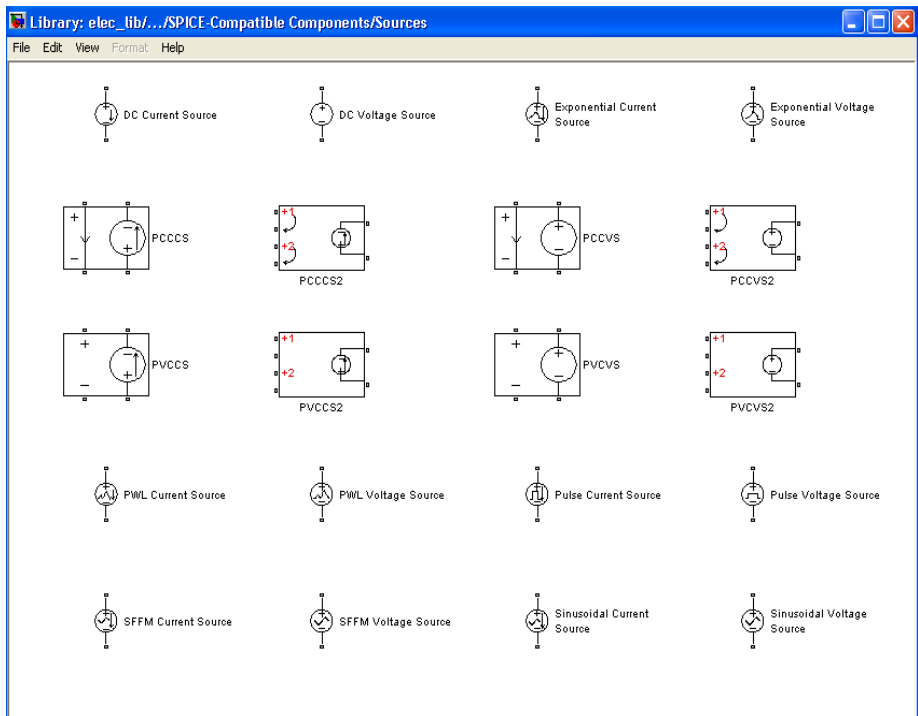
### III.5.3. Sources

Cette librairie comprend différentes sources compatibles au langage SPICE.

Nous trouvons :

- des sources de tension et de courant continues,
- des sources exponentielles de courant et de tension,
- des sources contrôlées de courant par un courant,
- des sources contrôlées de courant par une tension,
- des sources de courant et de tension à rapport cyclique variable,
- des sources, de tension et courant, sinusoïdales,
- etc.

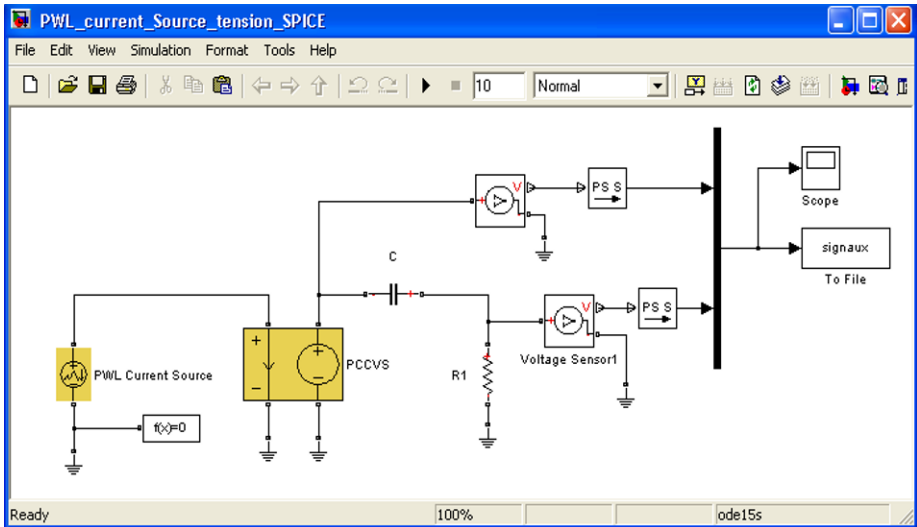
Cette librairie contient des sources de tension, de courant, contrôlées par une tension ou un courant, un générateur d'impulsion, une source de tension et de courant de forme exponentielle et bien d'autres types de sources propres à SPICE.



#### • Source de courant PWL et source de tension contrôlée PCCVS

Dans cette application, nous utilisons une source de courant PWL (Piecewise Wave Lookup Source) dont on spécifie le temps, les valeurs de courant avec des méthodes d'interpolation, linéaire, cubique ou spline.

Dans notre cas, nous avons choisi une interpolation linéaire comme les tables d'interpolation de Simulink.



Dans la fonction Callback `StopFcn`, nous avons programmé les commandes suivantes qui permettent de lire le fichier binaire `signaux.mat` et de tracer dans un même graphique à 2 axes d'ordonnées différents.

La tension contrôlée par la source de courant PWL, contrôle elle-même une source de tension PCCVS avec un polynôme du premier degré et la tension aux bornes de la résistance qui correspond, à un terme près, à la dérivée de la tension fournie par la source PCCVS.

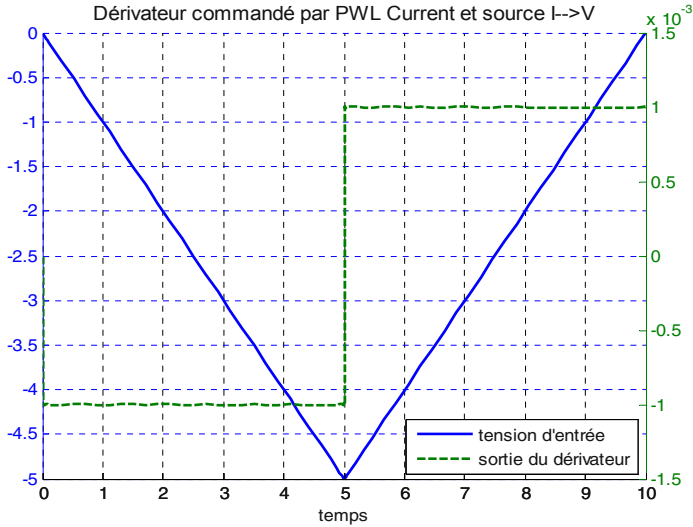
```
load signaux.mat
plotyy(x(1,:),x(2,:),x(1,:),x(3,:))
title('Dérivateur commandé par PWL Current et source I-->V')
xlabel('temps')
grid
gtext('Tension d'entrée')
gtext('Sortie du dérivateur')
```

La tension aux bornes de la résistance est donnée par l'expression :

$$V_R = \frac{1}{RC} \frac{dV_e}{dt}$$

Le signal  $V_e$ , sous forme de rampes, devient un échelon en sortie du dérivateur.

Avec  $R=1k\Omega$ ,  $C=1\mu F$ , nous obtenons des paliers de valeur  $10^{-3}$  V en valeur absolue. C'est ce que nous vérifions dans le graphique suivant.



• **Source de courant contrôlée PVCCS2**

Dans cette application, nous étudions la source de courant PVCCS2, contrôlée par 2 autres sources de tension. Le bloc PVCCS2 (Two-Input Polynomial Voltage-Controlled Current Source) représente une source de courant dont la sortie est une fonction polynomiale des tensions appliquées aux paires de ports de contrôle (entre chaque numéro 1 et 2, le suivant représentant la masse).

Les équations décrivant ce courant en fonction du temps, sont les suivantes :

$$I_s = p_1 + p_2 V_1 + p_3 V_2 + p_4 V_1^2 + p_5 V_1 V_2 + p_6 V_2^2 + \dots$$

où:

- $V_1$  : tension à la 1<sup>ère</sup> paire des ports d'entrée,
- $V_2$  : tension à la 2<sup>ème</sup> paire des ports d'entrée,
- $p$  : vecteur des coefficients.

Lorsqu'on double-clique sur ce bloc, on choisit les valeurs des coefficients du vecteur  $p$ .

**PVCCS2**

The Polynomial Voltage-Controlled Current Source (PVCCS2) block generates a current waveform,  $I_{out}$ , by evaluating a polynomial function for two controlling input voltages,  $V_{in1}$  and  $V_{in2}$ .  $V_{in1}$  and  $V_{in2}$  are the time-dependent voltages across the input terminals. If you specify a vector of polynomial coefficients,  $p$ , in ascending order, the output is:  $I_{out} = p(1) + p(2)*V_{in1} + p(3)*V_{in2} + p(4)*V_{in1}^2 + p(5)*V_{in1}*V_{in2} + p(6)*V_{in2}^2 + \dots$

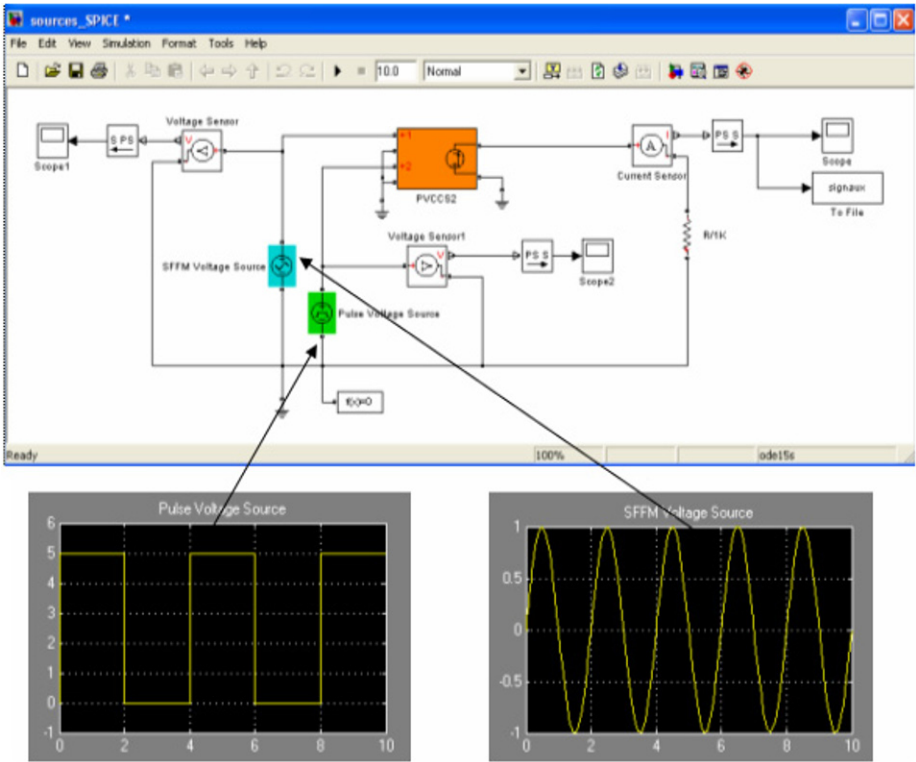
**Parameters**

Polynomial coefficients:

[ 0 0.5 1 ]



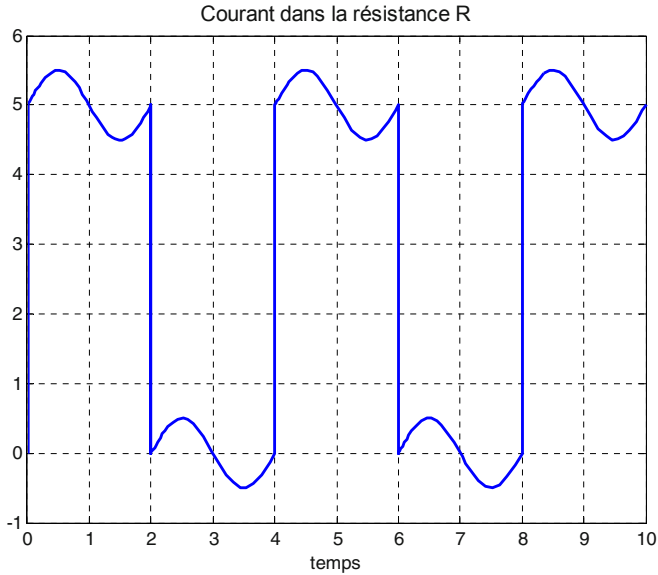
Le courant parcourant la résistance R est donné ainsi par :  $I_R = 0.5 V_1 + V_2$



$V_1$ , (SFFM Voltage Source), est une sinusoïde d'amplitude 1V, de fréquence 0.5 Hz.  
 $V_2$ , (Pulse Voltage Source), est un signal carré d'amplitude 5V et de largeur 2s (Pulse width, PW).

Parameters		
Initial value, V1:	0	V
Pulse value, V2:	5	V
Pulse delay time, TD:	0	s
Pulse rise time, TR:	1e-09	s
Pulse fall time, TF:	1e-09	s
Pulse width, PW:	2	s
Pulse period, PER:	4	s

Par le coefficient 0.5 du 2<sup>ème</sup> coefficient du vecteur p, l'amplitude de la sinusoïde est divisée par 2 et elle se superpose au signal carré, comme on l'observe dans la figure suivante.

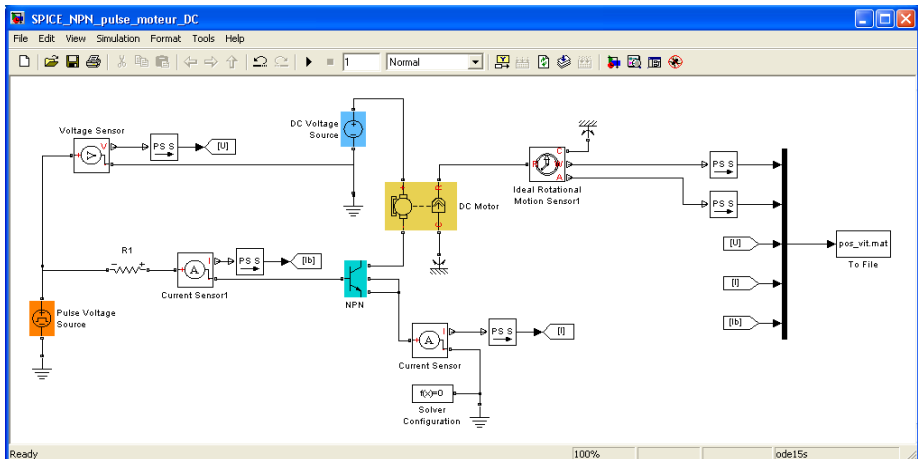


• *Commande d'un moteur DC par transistor NPN compatible SPICE*

Nous utilisons dans le modèle suivant un transistor NPN compatible SPICE pour commander sur son collecteur, un moteur à courant continu sans couples externes sur son arbre. A la base du transistor, nous appliquons un signal carré de largeur 5 s et de période 20 s par la source Pulse Voltage Source.

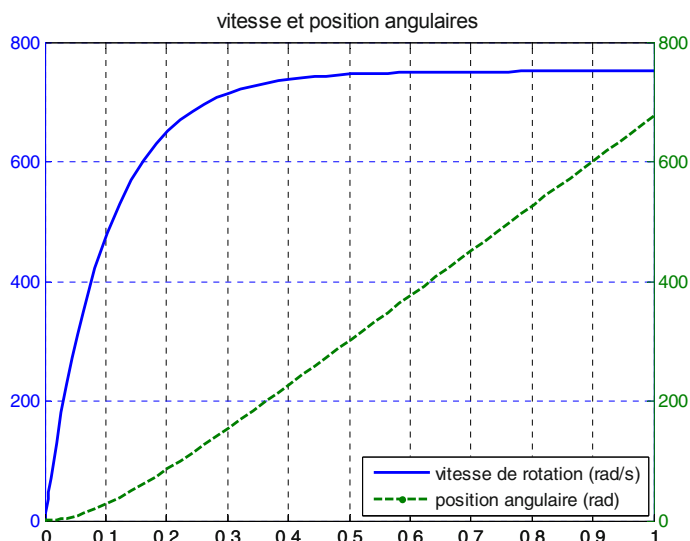
Le transistor est alimenté par la source continue DC Voltage Source.

Nous sauvegardons dans le fichier binaire `pos_vit.mat`, la vitesse, la position angulaire de l'arbre moteur, le signal carré, le courant à l'émetteur du transistor ainsi que le courant de base du transistor.



Grâce aux commandes suivantes, dans la fonction Callback `StopFcn`, nous lisons le fichier binaire et nous traçons les différentes courbes.

La dynamique de la vitesse est assez rapide car il n'y a pas de couples résistants supplémentaires, à part ceux propres au moteur, qui sont de  $0.01 \text{ g.cm}^2$  pour l'inertie et  $10^{-8} \text{ N.m/(rad/s)}$  pour l'amortisseur.



La figure suivante représente les courants de base et d'émetteur du transistor liés par un rapport de 10.

C'est ce que nous avons spécifié dans l'onglet `Forward Gain` de la boîte de dialogue qu'on obtient lorsqu'on double-clique sur le bloc NPN du transistor.

Parameters

Main Forward Gain Reverse Gain Resistors Capacitance Temperature

Transport saturation current, IS:  A/m<sup>2</sup> ▼

Forward beta, BF:

Forward emission coefficient, NF:

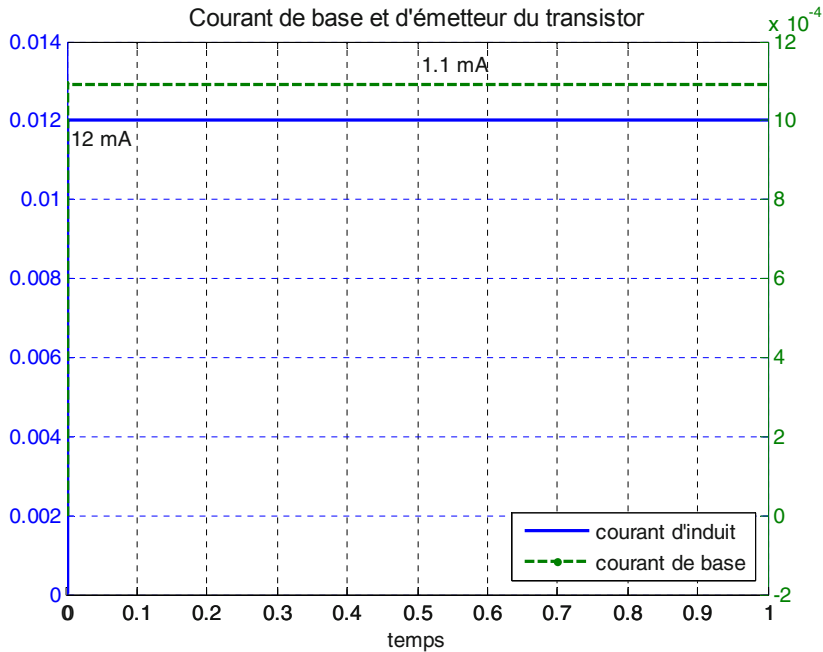
B-E leakage current, ISE:  A/m<sup>2</sup> ▼

B-E emission coefficient, NE:

Forward knee current, IKF:  A/m<sup>2</sup> ▼

Forward Early voltage, VAF:  V ▼

La figure suivante représente le courant de base du transistor ainsi que celui de son émetteur ou d'induit du moteur.



• *Source contrôlée de courant PVCCS et de tension PCCCS*

Dans l'exemple suivant nous utilisons la source de courant contrôlée par 2 sources de tension (sinusoïde et carré).

Le courant fourni par cette source alimente la résistance  $R_3$  de  $10\text{ k}\Omega$ .

Le bloc PCCCS (Polynomial Current-Controlled Current Source) représente une source de courant dont la sortie est une fonction polynomiale.

Les équations suivantes décrivent ce courant en fonction du temps.

$$I_s = p(0) + p(1)I_e + \dots + p(n-1)I_e^{n-1} + p(n)I_e^n$$

Si on spécifie uniquement un scalaire  $p$  au lieu d'un vecteur, le courant de sortie est alors donné par :  $I_s = p I_e$

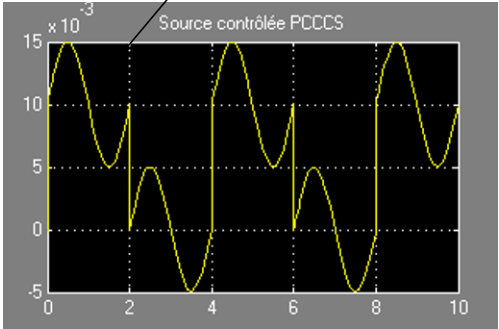
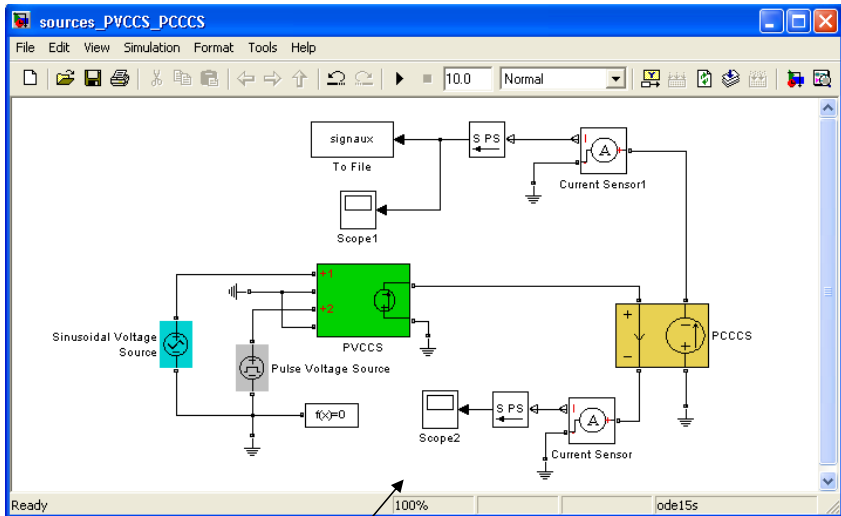
Dans notre cas, nous avons spécifié  $p(0) = 0$  et  $p(1) = 0.1$ , ce qui donne un courant de sortie par :

$$I_s = 0.1 I_e$$

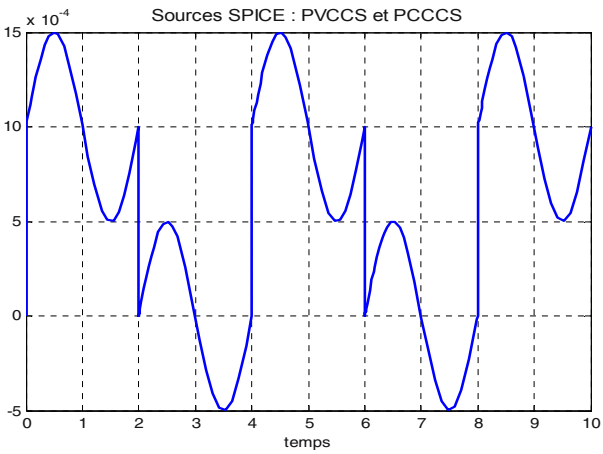
Parameters

Polynomial coefficients:

[ 0 0.1 ]



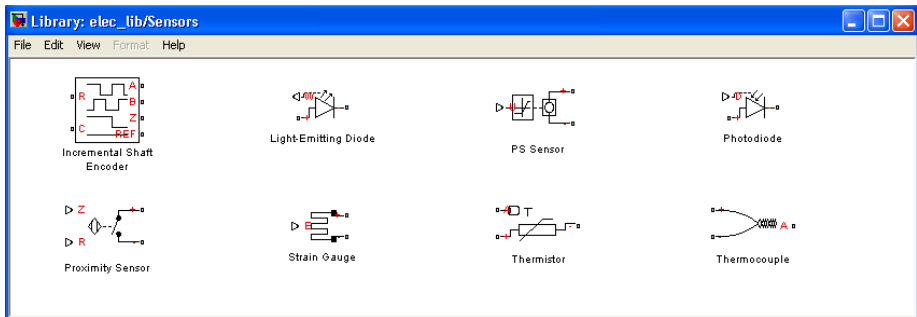
Le courant de sortie de la source PVCCS est 10 fois supérieur à celui de la source PCCCS du fait qu'on a spécifié un gain de 0.1 dans sa boîte de dialogue.




### III.6. Sensors

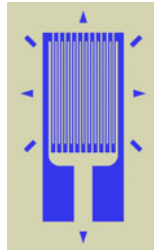
Nous trouvons dans cette bibliothèque des capteurs tels que :

- le codeur incrémental,
- la photodiode,
- la jauge de contraintes,
- la thermistance,
- le thermocouple,
- le capteur de proximité,
- le codeur incrémental,
- etc.



#### III.6.1. Jauges de contrainte

Les jauges de contrainte,  **Strain Gauge**, sont constituées par un fil très fin, collé sur un support isolant comme suit :



Elles consistent en des spires rapprochées et sont le plus souvent fabriquées à partir d'une mince feuille métallique (quelques  $\mu\text{m}$  d'épaisseur) et d'un isolant, qu'on traite comme un circuit imprimé (par lithographie ou par attaque à l'acide).


Le fonctionnement de ce dispositif repose sur le principe physique selon lequel la résistance électrique d'un fil se modifie proportionnellement à sa déformation lorsque ce fil est étiré ou comprimé par une force exercée.

La modification de cette résistance est ainsi utilisée comme indicateur de la force exercée.

Il est ainsi possible de procéder à des mesures de traction et de pression de ce fil.

La résistance d'un fil électrique est donnée par :  $R = \rho \frac{l}{s}$ . Elle dépend donc de la longueur du matériau qui la constitue. Pour un allongement relatif  $\frac{\Delta l}{l}$ , la variation relative de la résistance est donnée par :

$$\frac{\Delta R}{R} = K \frac{\Delta l}{l}$$

Le bloc  Strain Gauge est défini par :  $\frac{\Delta R}{R} = K \varepsilon$  où  $\varepsilon$  est la contrainte à son port B et K

le facteur de la jauge ( $K = \frac{\Delta R / R}{\Delta l / l}$ , sans dimensions).

En double-cliquant sur ce bloc, nous pouvons définir la résistance R de la jauge et son facteur K.

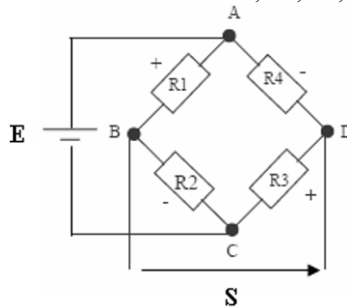
**Parameters**

Gauge resistance:  Ohm

Gauge factor:

La jauge étant un capteur passif, elle est nécessairement associée à un conditionneur actif, généralement le pont de Wheatstone.

Le pont de Wheatstone est constitué de 4 résistances R1, R2, R3, et R4 montées comme suit :



Il est alimenté par une tension continue entre les points A et C et la tension de sortie est mesurée entre les points B et D.

Si on branche une résistance  $R_{BD}$  entre les points B et D, à l'équilibre, il ne circule aucun courant dans cette dernière.

Alimenté par une source de courant, le pont présente à l'équilibre une tension nulle entre les points B et D.

La variation de l'une de ces résistances, fait apparaître, entre B et D, une tension non nulle.

$$E = (R_1 + R_2) i_1 = (R_3 + R_4) i_2$$

A l'équilibre,  $R_1 i_1 = R_4 i_2$ . Ainsi, nous avons alors :

$$\frac{R_1}{R_1 + R_2} = \frac{R_4}{R_4 + R_3}$$

soit :

$$R_4 R_2 = R_1 R_3$$

L'équilibre du pont est réalisé quand les produits en croix des résistances sont égaux.

Si on met la jauge à la place de  $R_4$ , par exemple, avec sa résistance  $R_x$ , l'équilibre du pont est obtenu lorsque cette valeur vaut :

$$R_x = \frac{R_1 R_3}{R_2}$$

Les applications des jauges de contraintes sont la mesure d'une déformation sur une pièce, d'une force ou d'une pression,

En utilisant le théorème de Millman, l'expression de la tension  $S$  est :

$$S = E \frac{R_1 R_3 - R_2 R_4}{(R_1 + R_2)(R_3 + R_4)}$$

Nous retrouvons bien, la même condition d'équilibre du pont,  $R_1 R_3 - R_2 R = 0$ .

Dans le modèle suivant, nous disposons d'un pont de Wheatstone avec les valeurs des résistances :

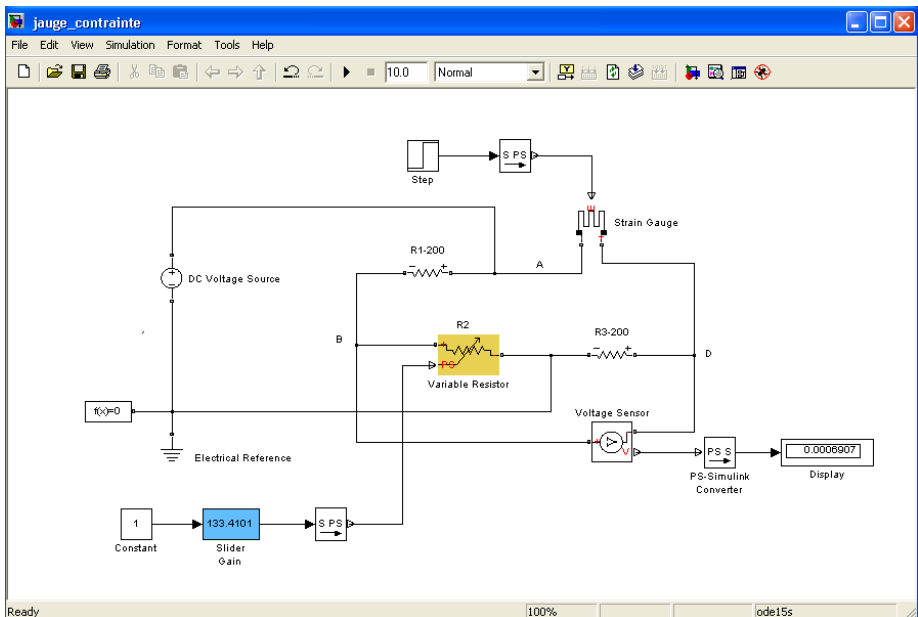
$$R_1 = 200 \Omega,$$

$$R_2 : \text{résistance variable},$$

$$R_3 = 200 \Omega,$$

$$R_4 = R_x : \text{jauge de contrainte}.$$

Nous utilisons une résistance variable que l'on fait varier jusqu'à obtenir l'équilibre, soit une tension nulle entre les points B et D.



La tension entre B et D devient nulle pour une résistance de  $133.26 \Omega$  donnée par la valeur Slider Gain.



### III.6.2. Thermocouple

Le modèle du thermocouple permet de convertir une différence de potentiel thermique en différence de potentiel électrique.

La mesure de température par un thermocouple est basée sur l'effet Seebeck.

Le bloc `Thermocouple` représente un thermocouple en utilisant des tables et des fonctions standard définies dans la base de données NIST ITS-90 Thermocouple Database.

Ces tables et fonctions permettent d'obtenir la force électromotrice (f.e.m) en fonction de la température.

Elles ont été adoptées par American Society for Testing and Materials (ASTM) et International Electrotechnical Commission (IEC).

Des lettres ont été désignées pour les thermocouples selon la plage de températures à mesurer.

Le thermocouple de type K permet une mesure dans une gamme de températures très large (250°C à 1 372°C).

Il est constitué de Chromel (alliage nickel + chrome) et Alumel (alliage nickel + aluminium).

Dans le modèle suivant, nous appliquons un échelon de température grâce à la source de température (`Ideal Temperature Source`) contrôlée par un échelon d'amplitude 300, lequel est ensuite transformé en signal physique de Simscape par le convertisseur `Simulink PS Converter (S→PS)`.

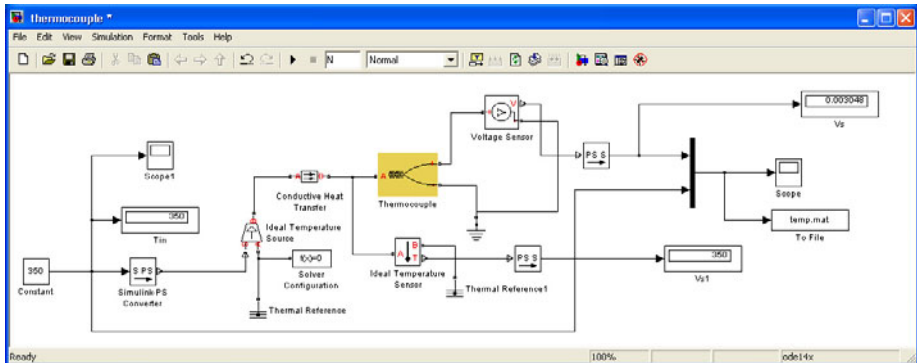
Après un échange de chaleur de type conductif (`Conductive Heat Transfert`), la température est mesurée grâce au thermocouple `Thermocouple`, à partir de la force électromotrice que nous sauvegardons dans le fichier binaire `temp.mat` en même que l'échelon d'entrée.

La température est aussi mesurée par le capteur `Ideal Temperature Sensor` qui mesure la température entre les A et B. Le point B étant relié à la masse thermique, nous mesurons ainsi la température au point A.

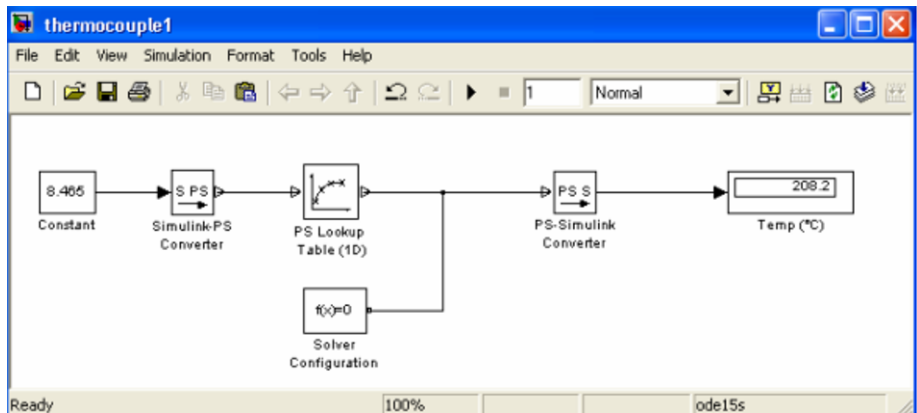
Les valeurs, par défaut, des coefficients du polynôme sont définies pour un thermocouple de type S dont la gamme de températures est : -50 à 1064°C.

La f.e.m en sortie du thermocouple est donnée par le polynôme défini dans la base de données NIST ITS-90.

(<http://srdata.nist.gov/its90/main/>) .



Pour une température d'entrée de 350°C, celle mesurée par le thermocouple l'est sous forme d'une tension de 0.0004801 V. Pour passer de la valeur de cette f.e.m à la température, on peut utiliser des tables ITS-90.



Le thermocouple de type K présente une courbe de conversion, f.e.m-Température, presque linéaire entre 0 et 1000 °C avec un coefficient de Seebeck,  $\alpha \approx 40 \mu V / ^\circ C$ .

Electronique Appliquée, Electromécanique sous  
Simscape & SimPowerSystems (Matlab/Simulink)  
Mokhtari, M.; Martaj, N.  
2012, XIV, 664 p. 500 ill. Avec online files/update.,  
Softcover  
ISBN: 978-3-642-24200-7