

# Modélisation et Simulation des Systèmes Multi-Physiques avec MATLAB – Simulink (R2020b)

pour l'étudiant et l'ingénieur

Quatrième édition

Introduction au Model Based Design



- **MATLAB**
- **Simulink**
- **Simscape**
  - domaine électrique
  - domaine mécanique
  - domaine thermique
  - domaine hydraulique
- **Stateflow**
  - diagramme d'états
- **MATLAB Online**
- **MATLAB Mobile**



*Contrôle commande des systèmes asservis, réglage des correcteurs*  
*Ingénierie numérique en langage MATLAB, les algorithmes fondamentaux*



# **Modélisation et Simulation des Systèmes Multi-Physiques avec MATLAB/Simulink R2020b pour l'étudiant et l'ingénieur Introduction au Model-Based-Design**

**Auteur : Ivan LIEBGOTT**

*Professeur de Chaire Supérieure en Classes Préparatoires aux Grandes Ecoles  
Lycée des Eucalyptus, Nice*

*Ingénieur diplômé de l'Institut National des Sciences Appliquées (INSA) de Lyon  
MASTER en conception de structures Aéronautiques et Spatiales (ISAE-SUPAERO)  
Agrégé de Sciences de l'Ingénieur  
Ancien élève de l'Ecole Normale Supérieure de Cachan*

[ivan.liebgott@gmail.com](mailto:ivan.liebgott@gmail.com)

Rejoignez-moi sur 

Ce livre a été créé pour être librement partagé avec la communauté des utilisateurs de MATLAB/Simulink.

Vos remarques et vos suggestions seront les bienvenues et me permettront de faire évoluer et d'améliorer cet ouvrage. Cette version est la quatrième édition du livre et fait suite aux éditions de 2013, 2015 et 2018.

Toute utilisation, même partielle, du contenu de ce document devra obligatoirement faire référence à l'ouvrage et à son auteur. L'hébergement sur un site internet des fichiers contenant le livre numérique et les modèles associés est soumise à l'autorisation de l'auteur.



**Ivan LIEBGOTT @ 2020**



## Préface :

Les ingénieurs sont au cœur du processus de conception des systèmes complexes et doivent chaque jour relever les défis de la compétitivité, de l'innovation et de la performance. Cela ne peut se faire sans l'intégration de processus industriels structurés, ni sans la maîtrise des outils modernes de modélisation et de simulation. A chaque étape du cycle de conception, les méthodes mises en œuvre doivent permettre de baisser les coûts, de réduire le risque d'erreur et d'en minimiser les impacts. Au cœur de ce processus, la modélisation et la simulation numérique jouent un rôle majeur et permettent aux ingénieurs d'anticiper, de comprendre et de vérifier les analyses qu'ils mènent tout au long du projet.

Les démarches industrielles standards, comme le cycle en V intègrent pleinement la simulation numérique au travers de méthodes associées comme le « Model Based Design » (conception basée sur le modèle). Les outils modernes de simulation permettent de créer des modèles globaux complexes intégrant tous les comportements du système et prenant en compte l'ensemble des interactions, cette démarche est appelée modélisation multi-physique. Le système réel peut avantageusement être remplacé par son modèle numérique pour réaliser des tests qui auparavant mobilisaient des moyens matériels et humains importants. Cette démarche impose de disposer de modèles validés qui reproduisent fidèlement le comportement des systèmes réels.

Cet ouvrage vous présente une approche de la modélisation multi-physique qui exploite les fonctionnalités et les innovations des logiciels de simulation afin de rendre le processus de modélisation plus rapide et plus efficace. La plate-forme de simulation utilisée est le logiciel MATLAB/Simulink version 2020b.

L'ouvrage propose de donner les clés permettant d'aborder la modélisation globale d'un système en créant le lien entre les méthodes industrielles et les méthodes utilisées dans le cycle de formation des ingénieurs. Il est illustré par de très nombreux exemples dans différents domaines de la technologie (électrique, hydraulique, mécanique, thermique...) et met en évidence l'interconnexion des domaines physiques.

Les fondamentaux de tous les outils nécessaires pour mener cette démarche sont présentés :

- MATLAB
- Simulink
- Simscape (Foundation Library)
- Simscape Fluids
- Simscape Multibody
- Simscape Electrical
- Stateflow

Dans cette quatrième édition, un tout nouveau chapitre est consacré à l'ingénierie numérique en langage MATLAB. Vous pourrez trouver les algorithmes fondamentaux de l'analyse numérique (dérivation numérique, intégration numérique, résolution d'équations stationnaires, filtrage d'un signal, résolution d'équations différentielles, résolution de systèmes d'équations...). Pour chaque chapitre des exemples concrets d'utilisation de ces algorithmes sont présentés. Tous les scripts et les fonctions définis dans ce chapitre peuvent être facilement réutilisés et adaptés afin de gagner un temps précieux dans vos projets. La programmation en langage MATLAB s'avèrera très utile en complément de l'utilisation des outils graphiques présentés également dans l'ouvrage.

Deux nouveaux chapitres sont consacrés à l'utilisation de MATLAB/Simulink Online et de MATLAB Mobile, vous pourrez découvrir les nouvelles méthodes de travail et de partage que proposent ces outils innovants. Prenez le temps de les découvrir.

L'ouvrage explore le potentiel des outils de modélisation et de programmation indispensables au déroulement d'un projet industriel de conception de système. Vous pourrez en percevoir toute la pertinence et l'exploiter en fonction des besoins spécifiques que vous rencontrerez dans votre démarche de conception et de modélisation.

Bonne lecture,

Ivan LIEBGOTT

**Modélisation et Simulation  
des Systèmes Multi-Physiques  
avec MATLAB/Simulink R2020b  
pour l'étudiant et l'ingénieur  
Introduction au Model-Based-Design**

**Ivan LIEBGOTT**

## Table des matières

### Chapitre 1 : Concepts et stratégies en modélisation

---

I. Introduction.....	19
II. Industrialisation et cycle de conception d'un système .....	20
A. Les compétences de l'ingénieur .....	21
B. Le triptyque des performances (*) .....	22
III. Mise en œuvre de la démarche- Introduction au « Model Based Design ».....	23
A. Architecture matérielle du projet .....	23
B. La phase d'Expression et de Spécification du Besoin.....	24
C. La phase de Conception-Modélisation-Simulation .....	25
1. La modélisation « white box » .....	26
2. La modélisation « Multi-Physique » .....	28
3. La simulation des modèles.....	29
4. La comparaison des performances simulées et mesurées .....	30
5. La modélisation « gray box ».....	30
6. Le Model-in-the-loop (MIL).....	34
D. La phase de Codage Implémentation .....	34
1. Le Software-in-the-loop (SIL).....	35
2. Le Processor-in-the-loop (PIL) .....	36
E. La phase d'Intégration Vérification .....	37
1. Le Hardware-In-the-Loop (HIL).....	37
F. La phase de Validation Recette.....	40

### Chapitre 2: Introduction et présentation des outils de modélisation

---

I. Le logiciel MATLAB-Simulink.....	41
A. Description et hiérarchie des outils utilisés.....	41
1. MATLAB.....	41
2. Simulink.....	43
3. Simscape.....	44
4. Stateflow .....	47
5. Utilisation des outils de modélisation.....	47
II. Présentation de l'environnement MATLAB – Simulink.....	49
A. Lancement du logiciel .....	49
B. La fenêtre de l'environnement MATLAB .....	49
1. La barre de commande MATLAB .....	50
C. La fenêtre de l'environnement Simulink.....	50
D. Configuration de MATLAB – Simulink.....	55
1. Nommer un fichier dans MATLAB/Simulink.....	55
2. Le « path » de MATLAB.....	55
3. Ajout de dossiers dans le « path » pour toutes les sessions .....	55
4. Ajout de dossiers dans le « path » pour la session courante.....	56

## Chapitre 3: Stratégie de conception d'un modèle multi-physique

---

I. Stratégie de conception d'un modèle multi-physique .....	57
A. Lien avec le diagramme Chaîne d'énergie/Chaîne d'information .....	57
II. Application à un pilote hydraulique de bateau .....	59
A. Diagramme présentant la chaîne d'énergie et d'information du pilote hydraulique de bateau .....	60
B. Modèle multi-physique du pilote hydraulique de bateau réalisé avec MATLAB - Simulink .....	61
C. Chargement et simulation du modèle .....	62
D. Visualisation des résultats issus du modèle multi-physique.....	62
E. Exploration du modèle.....	68
1. Exploration du modèle de la chaîne d'information : Simulink et Stateflow.....	68
2. Exploration du modèle de la chaîne d'énergie : Simscape Electric Library .....	69
3. Exploration de la chaîne d'énergie : Simscape Fluids .....	70
4. Exploration de la chaîne d'énergie : Simscape Multibody .....	71
5. Exploration de la chaîne d'énergie : Simulink.....	72
F. Pilotage interactif du modèle .....	72
III. Exemples de modèles multi-physique et exploitations possibles.....	75
A. Le robot Maxpid.....	75
B. L'axe linéaire Control'X.....	79
C. Comment faire un modèle multi-physique avec MATLAB-Simulink ?.....	84

## Chapitre 4: Prise en main de Simscape

---

I. Introduction à la modélisation acausale avec Simscape.....	85
A. Choix des composants.....	86
B. Placement et assemblage des composants .....	86
C. Les différents types de ports et de connexions .....	88
D. Paramétrage des composants.....	90
E. Lancement de la simulation et analyse des résultats .....	93
F. Utilisation du Data Inspector.....	96
II. Comparaison avec l'approche causale .....	103
A. Equation de comportement du système .....	103
B. Choix des composants.....	103
C. Placement et assemblage des composants.....	103
D. Paramétrage des composants.....	104
E. Lancement de la simulation et analyse des résultats .....	105
F. Avantage et inconvénients des approches causale et acausale.....	107
III. Les fondamentaux de la modélisation avec Simscape .....	108
A. Notions de domaines physiques.....	108
B. Les blocs importants de Simscape.....	109
C. Variables de type « Across » et « Throught » et positionnement des capteurs.....	110
D. L'orientation des composants .....	110
1. Utilisation de composants actifs.....	111
2. Implantation et orientation des capteurs.....	114
3. Utilisation de composants dont la dynamique est orientée .....	117

4. Utilisation de composants passifs.....	117
5. Choix du solveur .....	118
6. Les problèmes que peut rencontrer le solveur .....	119
IV. Exemples de modélisation multi-domaine .....	120
A. Domaine électromécanique – Axe linéaire .....	120
1. Choix des composants.....	121
2. Placement et assemblage des composants .....	123
3. Paramétrage des composants.....	124
4. Simulation du modèle en boucle ouverte .....	131
5. Utilisation du Simscape Results Explorer .....	133
6. Création de sous-systèmes.....	137
7. Modélisation de l’asservissement en position de l’axe.....	144
B. Domaines hydraulique– vérin hydraulique simple effet .....	149
1. Choix des composants.....	149
1. Placement et assemblage des composants .....	152
2. Paramétrage des composants.....	152
3. Simulation.....	159
4. Utilisation des fonctionnalités de routage des signaux.....	161
5. Remplacement de la source de pression par une source de débit.....	164
C. Domaine électrique –Commande PWM d’un moteur à courant continu .....	168
1. Utilisation du composant « Controlled PWM Voltage » .....	169
2. Commande PWM d’un moteur à courant continu .....	173
3. Utilisation du composant « H-Bridge » (pont en H).....	175
D. Rendre un modèle interactif, utilisation de la bibliothèque « dashboard » .....	183
1. Exemple de modèle interactif.....	183
2. Utilisation des blocs de la bibliothèque.....	185
E. Le domaine thermique .....	190
1. Rappels théoriques sur les échanges thermiques .....	190
2. Le phénomène de conduction.....	190
3. Le phénomène de convection .....	192
4. Le phénomène de rayonnement.....	193
5. La masse thermique .....	195
6. La résistance thermique.....	196
7. Présentation de la bibliothèque Thermal de Simscape .....	198
8. Chauffage d’une barre métallique par conduction .....	200
9. Choix des composants du domaine thermique .....	200
10. Les ports transmetteurs de puissance (PCP) du domaine thermique .....	204
11. Paramétrage des composants du modèle .....	204
12. Mise en place des capteurs du domaine thermique.....	207
13. Définition des conditions initiales du modèle .....	213
14. Réglage du temps de simulation.....	214
15. Lancement de la simulation et exportation des résultats .....	214
16. Interprétation des résultats de la simulation .....	215
17. Rappels théoriques sur les échanges thermiques à travers une paroi d’un bâtiment.....	222
18. Modélisation avec Simscape des échanges par conduction et convection dans une paroi monocouche .....	223

19. Utilisation du bloc Thermal Résistance.....	227
20. Modélisation avec Simscape d'une paroi multicouche d'un bâtiment.....	230
21. Modélisation de l'efficacité d'un double vitrage avec couche d'argon.....	241
22. Modélisation thermique d'une maison.....	253
V. Application pédagogique.....	257
A. Présentation du hacheur série.....	257
B. Objectifs pédagogiques.....	259
C. La construction du modèle.....	261
D. La didactisation du modèle.....	268
1. Création d'un sous-système et ajout d'une image.....	268
2. L'instrumentation du modèle.....	268
3. Conclusion sur la didactisation du modèle.....	272
4. Optimiser la didactisation du modèle en fonction de l'objectif d'apprentissage visé.....	274
E. Exploitation des résultats issus de la simulation du modèle.....	278
1. Objectif 1 : Comprendre la circulation du courant dans le circuit en phase active et en phase de roue libre.....	278
2. Objectif 2 : Visualiser et évaluer l'influence du rapport cyclique sur le courant moteur.....	280
3. Objectif 3 : Visualiser et évaluer l'influence de la fréquence de hachage sur l'ondulation du courant.....	282
4. Objectif 4 : Visualiser et évaluer l'influence de l'inductance de la charge sur l'ondulation du courant.....	284
F. Conclusion.....	287

## Chapitre 5: Prise en main de MATLAB

---

I. Introduction.....	288
A. Création de variable.....	288
B. Création de vecteur.....	289
C. Indexation des composantes d'un vecteur.....	289
D. Tracés de courbes.....	290
E. Mise en forme élémentaires des courbes.....	292
F. Annotation des graphiques.....	294
G. Création de matrices.....	296
H. Opération éléments par éléments.....	301
I. Créer un script élémentaire.....	302
J. Les opérateurs de comparaison de MATLAB.....	305
K. Les structure de boucles usuelles.....	306
1. Syntaxe de la boucle if – elseif – else.....	306
2. Syntaxe de la boucle for.....	306
3. Syntaxe de la boucle while.....	306
II. Exemple d'exploitations.....	307
A. Interpolation d'une série de données.....	307
B. Le calcul symbolique avec MATLAB.....	309
1. Résolution d'une équation algébrique.....	309
2. Développer ou factoriser une expression.....	311
3. Dériver une fonction.....	312
4. Intégrer une fonction.....	312

5. Utiliser la transformée de Laplace.....	313
6. Utiliser la transformée inverse de Laplace .....	314
7. Décomposition en éléments simples.....	314
1. Résolution d'une équation différentielle d'ordre 1.....	315
1. Résolution d'une équation différentielle d'ordre 2.....	316
C. Manipulation des fonctions de transfert.....	318
1. Création d'une fonction de transfert .....	318
2. Opérations sur les fonctions de transfert.....	319
3. Tracer les réponses temporelles d'un système.....	322
4. Tracer les réponses fréquentielles d'un système.....	323
5. Evaluer les marges de gain et de phase.....	325
6. Tableau récapitulatif des commandes utiles sur les fonctions de transfert .....	326

## Chapitre 6: Prise en main de Simulink

---

I. Introduction.....	328
II. Régulation en température d'un four.....	328
A. Ouverture du modèle.....	329
B. Ouverture du script contenant la définition des variables.....	330
C. Lancement de la simulation.....	331
D. Tracer un diagramme de Bode avec Simulink .....	331
1. Tracer un digramme de Bode en boucle ouverte.....	332
2. Tracer un diagramme de Bode en boucle fermée.....	336
E. Tracer d'un diagramme de Black-Nichols .....	339
F. Ajout et paramétrage d'une saturation.....	340
G. Exportation des variables de la simulation vers le Workspace.....	342
1. Ecriture d'un script pour tracer une série de courbes .....	345
H. Linéarisation d'un modèle Simulink à partir d'un script .....	346

## Chapitre 7: Prise en main de Stateflow

---

I. Introduction à Stateflow .....	349
A. Modélisation d'une machine à état avec Stateflow .....	349
B. Construction du diagramme d'état.....	349
1. Ouverture du modèle.....	349
2. Insertion d'un « chart ».....	350
C. Création d'un diagramme d'état élémentaire .....	351
1. Création des états.....	352
2. Création d'une transition par défaut .....	353
3. Création des transitions .....	353
4. Création des actions dans les états.....	354
5. Création des étiquettes de transitions.....	354
6. Définitions des variables d'entrée et de sortie du diagramme d'état .....	355
7. Simulation du diagramme d'états.....	360
D. Architecture des machines à états.....	360

1. La hiérarchie des états .....	360
2. Les priorités de test des transitions.....	361
3. Etats parallèles .....	361
E. Ajout de niveaux hiérarchique et d'états parallèles dans un diagramme d'état .....	362
F. Récapitulatif et complément des commandes utiles de Stateflow .....	368

## Chapitre 8: Prise en main de Multibody

---

I. Introduction à Multibody .....	370
A. Analyse d'un modèle Multibody .....	370
B. Paramétrage de la gravité.....	372
II. Intégration d'un modèle Multibody dans un modèle multi-physique .....	373
A. Connexions du modèle.....	375
B. Interfaçage entre Simscape et Multibody .....	376
1. Interfaçage entre Simscape et Multibody pour la translation.....	376
2. Interfaçage entre Simscape et Multibody pour la rotation.....	377
3. Ajout de ports sur une liaison.....	378
4. Modélisation d'un effort extérieur variable .....	382
C. Résultat de la simulation.....	385
III. Importation d'un modèle SolidWorks dans Multibody.....	385
A. Les principes .....	385
B. Installation de « Multibody Link » .....	385
C. Conversion d'un fichier assemblage de Solidworks en fichier xml.....	387

## Chapitre 9: L'identification d'un modèle

---

I. La modélisation black-box, l'identification.....	391
A. Présentation de la méthode.....	391
B. Mise en œuvre de la méthode en utilisant la toolbox Identification .....	392
1. Analyse des données utilisées pour l'identification .....	392
2. Ouverture et présentation de la toolbox « System Identification » .....	393
3. Importation des données.....	394
C. Utilisation de la méthode en utilisant les lignes de commandes.....	400

## Chapitre 10: Le contrôle commande avec MATLAB/Simulink

---

I. Introduction.....	402
II. Réglage automatique d'un PID .....	402
A. Modélisation .....	402
B. Ouverture du modèle.....	403
1. Analyse de la réponse temporelle.....	406
2. Importation dans Simulink.....	410
III. Réglage manuel d'un PID avec l'outil « Control System Designer » .....	412
A. Ouverture du modèle.....	412
B. Réglage du PID.....	413

1. Lancement de Control System Designer .....	413
2. Diagrammes de Bode et de Black de la FTBO .....	416
3. Visualisation des réponses temporelles.....	418
4. Réglage du PID.....	423
5. Définition et visualisation des critères de performance.....	423
6. Réglage du PID à l'aide des curseurs .....	426
7. Exportation du réglage dans le modèle Simulink .....	428
IV. Conception et réglage d'un correcteur de forme quelconque .....	430
A. Ouverture du modèle.....	430
B. Conception du correcteur .....	431
1. Diagrammes fréquentiels de la FTBO.....	433
2. Visualisation des réponses temporelle et fréquentielle de la FTBF .....	436
3. Synthèse du correcteur.....	441
4. Visualisation de l'influence du gain de la FTBO.....	442
5. Ajout d'un intégrateur.....	444
6. Ajout d'un correcteur à avance de phase (Lead).....	446
7. Ajout d'un filtre rejecteur (Notch).....	449
8. Réglage d'un filtre rejecteur .....	450
9. Exportation de la fonction de transfert du correcteur vers le modèle Simulink .....	455

## Chapitre 11: Ingénierie numérique en langage MATLAB

---

I. Introduction.....	457
II. Les fonctions.....	457
A. Organisation et structure.....	457
B. Portée des variables.....	459
C. Création et définition d'une fonction.....	459
D. Utilisation d'une fonction par un script .....	463
E. Création d'une fonction prenant en argument une autre fonction .....	468
F. Appel d'une fonction à plusieurs arguments.....	473
III. Dérivation numérique.....	475
A. Aspects théoriques .....	475
1. Différence finie progressive .....	476
2. Différence finie rétrograde .....	476
3. Différence finie centrée .....	477
B. Codage en langage MATLAB.....	478
1. Différence finie progressive .....	478
2. Différence finie rétrograde .....	479
3. Différence finie centrée .....	481
4. Calcul exact de la fonction dérivée en utilisant le calcul symbolique .....	482
C. Exploitation de l'algorithme .....	483
1. Comparaison des différentes méthodes de dérivation numérique .....	483
2. Quantification et visualisation de l'erreur induite par les différentes méthodes.....	487
3. Détermination de la loi d'évolution de l'erreur en fonction du pas.....	489
4. Robustesse du processus.....	494

D. Application de la dérivation numérique .....	497
1. Dérivation du signal issu d'un codeur incrémental .....	497
2. Mise en évidence des problèmes rencontrés .....	498
3. Dérivation numérique à pas variable .....	499
IV. Résolution de l'équation $f(x)=0$ .....	503
A. Dichotomie .....	503
B. Méthode de Newton .....	507
V. Intégration numérique .....	511
A. Principe .....	511
B. Codage en langage MATLAB .....	513
1. Méthode des rectangles .....	513
2. Méthode des trapèzes .....	514
3. Calcul exact de l'intégrale d'une fonction en utilisant le calcul symbolique .....	515
C. Exploitation de l'algorithme .....	517
1. Comparaison des différentes méthodes d'intégration numérique .....	517
2. Quantification et visualisation de l'erreur induite par les différentes méthodes .....	520
3. Détermination de la loi d'évolution de l'erreur en fonction du pas .....	522
4. Robustesse du processus .....	526
D. Applications .....	529
1. Analyse d'une loi en trapèze de vitesse .....	529
VI. Filtrage d'un signal .....	533
A. Intérêt du filtrage d'un signal .....	533
B. Le filtre à moyenne glissante .....	535
C. Le filtre passe-bas du premier ordre .....	538
D. Utilisation de Simulink pour le filtrage d'un signal .....	542
1. Utilisation d'une fonction de transfert de la bibliothèque Continuous de Simulink .....	543
2. Utilisation du bloc Analog Filter Design de la DSP Toolbox de Simulink .....	545
3. Utilisation du bloc Moving Average de la DSP System Toolbox .....	547
VII. Résolution numérique des équations différentielles .....	550
A. Résolution d'une équation différentielle d'ordre 1 – Méthode d'Euler .....	550
1. Aspects théoriques .....	550
2. Codage en langage MATLAB .....	552
B. Résolution d'une équation différentielle d'ordre n – Méthode d'Euler .....	555
1. Aspects théoriques .....	555
2. Codage en langage MATLAB .....	558
3. Robustesse du processus. Résolution d'une équation différentielle d'ordre 3 .....	565
4. Robustesse du processus. Application à la résolution de l'équation de Van der Pol .....	567
C. Applications .....	570
1. Modélisation d'un oscillateur mécanique .....	570
2. Les différentes méthodes de résolution d'une équation différentielle avec MATLAB .....	570
3. Résolution avec my_ode, ode 23 et en calcul symbolique .....	571
4. Résolution de l'équation différentielle avec Simulink .....	573
5. Résolution de l'équation différentielle avec Simscape .....	575
VIII. Résolution de systèmes linéaires – Méthode de Gauss .....	577
A. Présentation de la méthode .....	577

B. Exemple .....	578
1. Recherche du pivot.....	578
2. Echange de lignes.....	578
3. Transvection - Triangularisation .....	579
4. Résolution.....	579
C. Codage en langage MATLAB .....	580
1. Recherche du pivot.....	580
2. Echange de lignes.....	583
3. Transvection.....	584
4. Triangularisation.....	586
5. Résolution.....	587
6. Pivot de Gauss.....	589
D. Robustesse .....	591

## Chapitre 12: MATLAB/Simulink Online

---

I. Présentation .....	593
II. Installation de l'environnement de travail à distance.....	594
III. Utilisation de MATLAB/Simulink online .....	595
A. Création d'un script élémentaire avec MATLAB online .....	596
B. Création d'un modèle élémentaire avec Simulink online.....	597
C. Exploitation d'un modèle multi-physique avec MATLAB/Simulink on line .....	599
D. Partager des dossiers contenant ses modèles avec d'autres utilisateurs.....	602
1. Partage d'un dossier à l'aide de la version de MATLAB installée sur son ordinateur .....	602
2. Partage d'un dossier en utilisant MATLAB online.....	604
3. Partage d'un dossier en utilisant MATLAB Drive online .....	605

## Chapitre 13: MATLAB Mobile

---

I. Présentation .....	607
II. Les fonctionnalités de MATLAB Mobile.....	608
1. Taper des lignes de commandes .....	608
2. Configurer le clavier MATLAB Keyboard.....	609
3. Accès à MATLAB Drive.....	609
4. Exécuter un script depuis MATLAB Mobile .....	609
5. Acquisition des données des capteurs du smartphone avec MATLAB Mobile .....	610
6. Exploitation des données des capteurs du smartphone avec MATLAB Mobile .....	612

## ANNEXES

---

Paramétrage des scopes.....	615
Utilisation du signal builder.....	622
Table des figures.....	628





# Chapitre 1 : Concepts et stratégies en modélisation

---

## I. Introduction

---

L'évolution rapide des technologies amène l'homme à concevoir des systèmes de plus en plus complexes dont le comportement ne peut plus être modélisé sans l'aide des logiciels de simulation.

Les interactions entre les composants étant nombreuses, il est primordial d'être en mesure de modéliser le système dans sa globalité en prenant en compte sa structure, son dispositif de commande et son environnement. Les chaînes d'énergie et d'information ne peuvent plus être modélisées séparément mais doivent être intégrées dans un modèle unique. A partir de ce modèle il sera possible d'anticiper le comportement du système réel en phase de conception afin de permettre au système réel de remplir sa fonction.

La modélisation des systèmes complexes est au cœur de la formation des ingénieurs. Cette approche, qui s'appuie sur la décomposition de la complexité fait apparaître des sous-ensembles qui mobilisent chacun des technologies différentes. Le décloisonnement des univers technologiques est donc nécessaire à la mise en place d'une démarche de modélisation pertinente. L'évolution très rapide des logiciels de simulation nous permet maintenant de modéliser toutes les parties d'un système à l'aide d'un logiciel unique. Cette approche multidisciplinaire, permet de gérer les échanges entre les sous-systèmes et de disposer de résultats qui prennent en compte la totalité des phénomènes physiques qui entrent en compte dans le fonctionnement du système. La pertinence des résultats est accrue et la mise en évidence des interactions devient possible. En contrepartie, la maîtrise des outils de simulation mobilise un champ de compétences nouveau, plus large et qui impose un processus d'apprentissage structuré. Cette nouvelle approche requiert un outil de modélisation et de simulation à environnement unique, visuel et interactif, validé industriellement et proposant tous les outils de modélisation de la chaîne d'information (régulateur classique, diagramme d'états...) et de la chaîne d'énergie (bibliothèques de composants avancées, maquette CAO intégrée...).

Afin de rendre efficace la démarche de formation, les méthodes et les concepts mis en œuvre doivent être les mêmes dans le cycle de formation des futurs ingénieurs que dans le monde industriel. Nous allons dans un premier temps voir comment il est possible de relier les standards de conception industrielle avec les concepts en vigueur dans les cursus de formation des ingénieurs.

La démarche mise en œuvre industriellement par les équipes d'ingénieurs pour concevoir et développer un système est un processus complexe qui respecte le plus souvent des standards industriels comme le « Cycle en V » et des méthodes associées comme le « Model Based Design ». Ces démarches imposent de disposer de modèles validés que les équipes d'ingénieurs utiliseront pour remplacer le matériel réel dans les phases de test et de réglage. La compétence « modéliser » est donc au cœur du processus d'apprentissage et reste l'une des compétences de base de l'ingénieur. L'évolution des logiciels de modélisation et de simulation a fortement modifié les méthodes d'acquisition de cette compétence pour les futurs ingénieurs. Le processus de modélisation ne se limite plus à écrire les équations différentielles qui régissent le comportement d'un système et à tenter de les résoudre analytiquement dans les rares cas où cela est possible. L'exploitation pertinente des logiciels de modélisation et de simulation permet de mettre en œuvre différentes approches afin de modéliser et d'obtenir rapidement des résultats valides :

- *La modélisation « white box » :*  
Cette approche consiste à écrire les équations et à les résoudre à l'aide de l'outil numérique. Cela permet d'obtenir une première approximation de la structure du modèle. Dans le cas général, les paramètres qui caractérisent le comportement du système ne sont jamais totalement connus. Il faut alors les négliger ou les estimer pour mener à bien la simulation. Les résultats obtenus doivent alors être confrontés à l'expérimentation afin d'optimiser et de valider le modèle. L'outil de modélisation privilégié est le schéma bloc qui décrit les équations mises en place dans la phase de modélisation.

- *La modélisation « multi-physique » :*  
 Cette approche consiste à assembler des composants dont le comportement est déjà modélisé dans le logiciel. Cette méthode ne nécessite pas l'écriture des équations et permet un gain de temps considérable. Les bibliothèques de composants permettent de modéliser dans tous les domaines physiques (mécanique, électrique, hydraulique, thermique...). Il est alors très simple de relier les différents domaines pour obtenir le modèle global d'un système. Il est également possible d'intégrer la maquette 3D de la structure modélisée ce qui permet de prendre en compte très facilement le comportement dynamique et les non-linéarités liées à la géométrie.
- *La modélisation « gray box » :*  
 Cette approche vient compléter le modèle « white box » ou « multi-physique » en estimant les paramètres inconnus par comparaison entre la réponse du modèle et la réponse du système réel. Des algorithmes permettent d'automatiser cette tâche et de donner des valeurs aux paramètres inconnus afin d'annuler les écarts entre les performances du modèle et les performances du système réel.
- *La modélisation « black box » :*  
 Cette approche, appelée également identification, consiste à associer un modèle mathématique à un comportement mesuré expérimentalement. L'outil numérique permet d'automatiser cette tâche en proposant des modèles mathématiques d'identification complexes. Cette méthode donne uniquement un modèle du comportement global du système sans se préoccuper de l'influence séparée des différents paramètres sur les performances.

La combinaison de ces approches permet de construire un modèle rapidement et de le valider vis-à-vis de l'expérimentation. Cette partie propose la mise en œuvre sur un cas concret d'une approche de type « Model Based Design ». En partant du cahier des charges et des exigences liées à la conception, les différentes étapes du cycle en V seront parcourues en utilisant les outils modernes de modélisation, de simulation et de validation. Les différentes approches de modélisation sont comparées et les avantages et inconvénients des différentes méthodes mis en évidence. La démarche permet de créer un lien entre le processus de formation et la réalité industrielle. Les étapes importantes de la démarche « Model Based Design » et de la détection d'erreur dans la génération de code C sont illustrées (Model in the loop, Software in the loop, Processor in the loop et Hardware in the loop).

## II. Industrialisation et cycle de conception d'un système

La démarche mise en œuvre industriellement par les équipes d'ingénieurs pour concevoir un système est un processus complexe qui respecte le plus souvent des standards industriels comme le « Cycle en V » et des méthodes associées comme le « Model Based Design ». La Figure 1 propose une représentation simplifiée du « Cycle en V », adaptée à la présentation des concepts de ce processus.

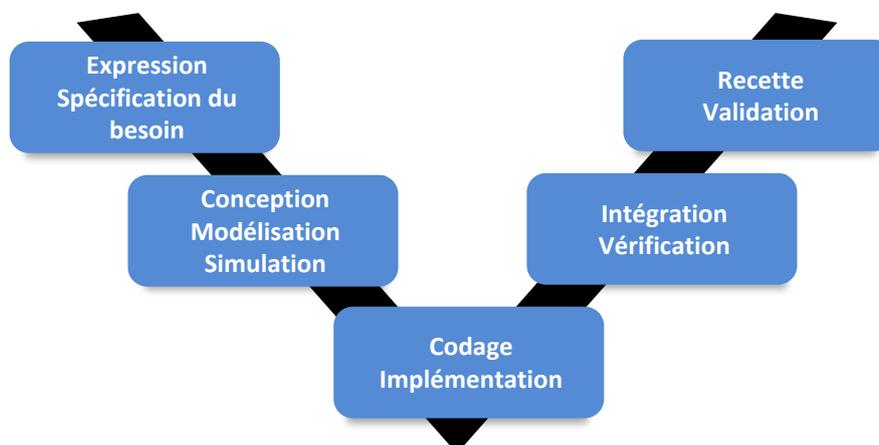


Figure 1 : cycle en V simplifié

## A. Les compétences de l'ingénieur

---

Pour mettre en œuvre cette démarche l'étudiant et l'ingénieur vont mobiliser 6 grandes compétences. Ces compétences sont à la base de la formation des ingénieurs et sont détaillées sur la Figure 2.

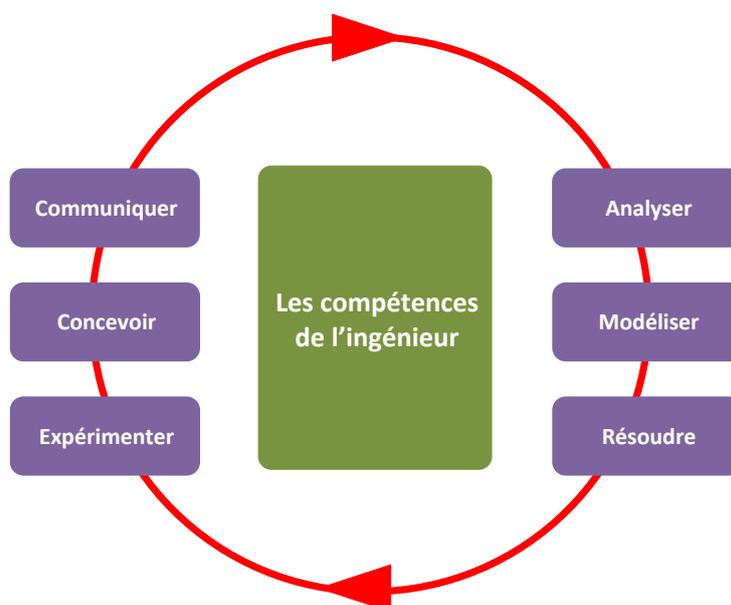


Figure 2 : les compétences de l'ingénieur

## B. Le triptyque des performances (\*)

La Figure 3 décrit les trois types de performances qui seront considérés et comparés dans la démarche :

- Les performances issues des spécifications du cahier des charges
- Les performances issues de la simulation numérique des modèles
- Les performances issues de la mesure directe sur le système réel à l'aide de capteurs

Trois types d'écart pourront être ainsi formés et analysés :

- L'écart entre les performances spécifiées par le cahier des charges et celles mesurées sur le système réel
- L'écart entre les performances mesurées sur le système réel et les performances issues de la simulation des modèles numériques
- L'écart entre les performances spécifiées par le cahier des charges et les performances issues de la simulation des modèles numériques

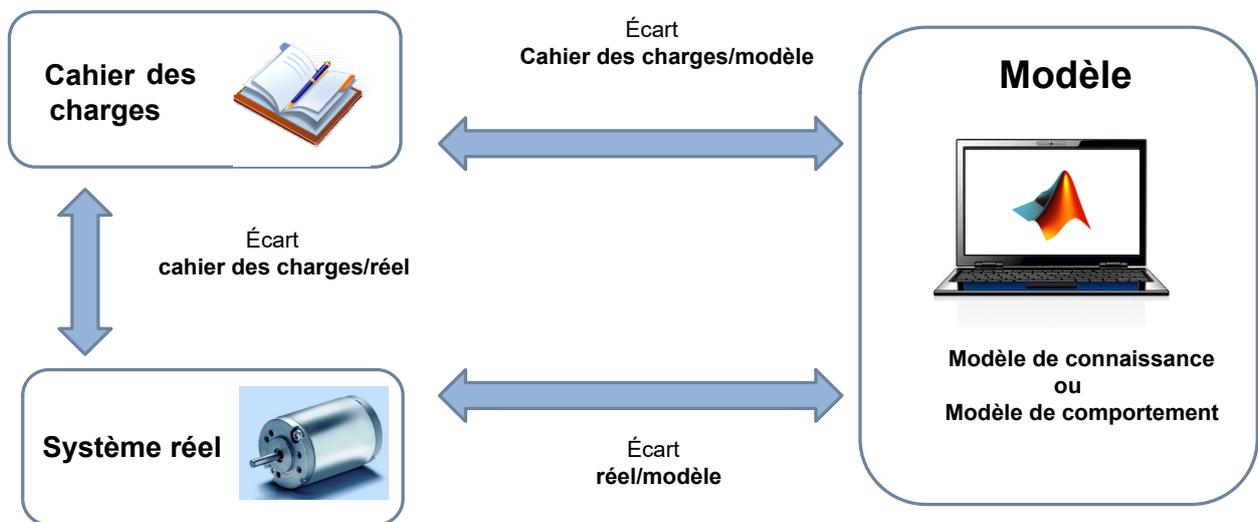


Figure 3 : triptyque Cahier des charges/Système réel/Modèle

(\*) Vous pouvez assister au webinar que nous avons réalisé avec Nadia Bedjaoui, qui présente la mise en œuvre de la démarche de l'ingénieur en exploitant les fonctionnalités du logiciel MATLAB/Simulink :

<http://fr.mathworks.com/videos/matlab-and-simulink-for-teaching-98093.html>

### III. Mise en œuvre de la démarche- Introduction au « Model Based Design »

Afin de comprendre les actions à mener dans les différentes phases du cycle en V, nous allons appliquer cette démarche à une problématique simple. Il est conseillé avant d'aborder la partie de l'ouvrage consacrée à la manipulation des outils de simulation et de modélisation, de prendre connaissance de cette partie. Cela permettra au lecteur de prendre du recul sur la démarche de modélisation, de comprendre la place du modèle dans le processus industriel et de percevoir tout l'intérêt de la conception basée sur le modèle appelée plus communément « Model Based Design »

L'exemple choisi propose de reproduire la démarche de conception de la logique de commande de l'asservissement de position angulaire d'un moteur à courant continu. L'objectif est de positionner angulairement l'axe du moteur conformément à une consigne de position imposée en respectant des critères de performances courants comme la précision, la rapidité ou la stabilité.

#### A. Architecture matérielle du projet

Afin de mettre en œuvre la démarche, la Figure 4 présente l'environnement matériel du projet. Il n'est pas nécessaire de disposer de ce matériel pour comprendre les concepts abordés dans cette partie.

- *Un ordinateur équipé d'un logiciel de modélisation, de simulation et de pilotage :*

Le rôle de l'ordinateur est de servir de support à la conception des modèles numériques, à leur simulation et à la compilation des résultats. L'ordinateur va également servir à créer des interfaces de mesures et de pilotage afin de commander le moteur via la carte de commande du moteur.

- *La carte de commande du moteur :*

Le rôle de la carte de commande du moteur est de piloter le moteur. Elle possède des entrées, pour récupérer les informations issues des capteurs (position angulaire du moteur...) et prendre en compte la consigne issue de l'ordinateur, et des sorties pour fournir une tension aux bornes du moteur et permettre sa mise en rotation. La tension fournie est modulable en fonction de la logique de commande programmée. Dans notre exemple elle fera également office de carte de puissance.

- *Le moteur à courant continu :*

Il est alimenté par une tension continue et délivre une vitesse de rotation. Un capteur de position intégré renvoie à la carte de commande la position angulaire du moteur.

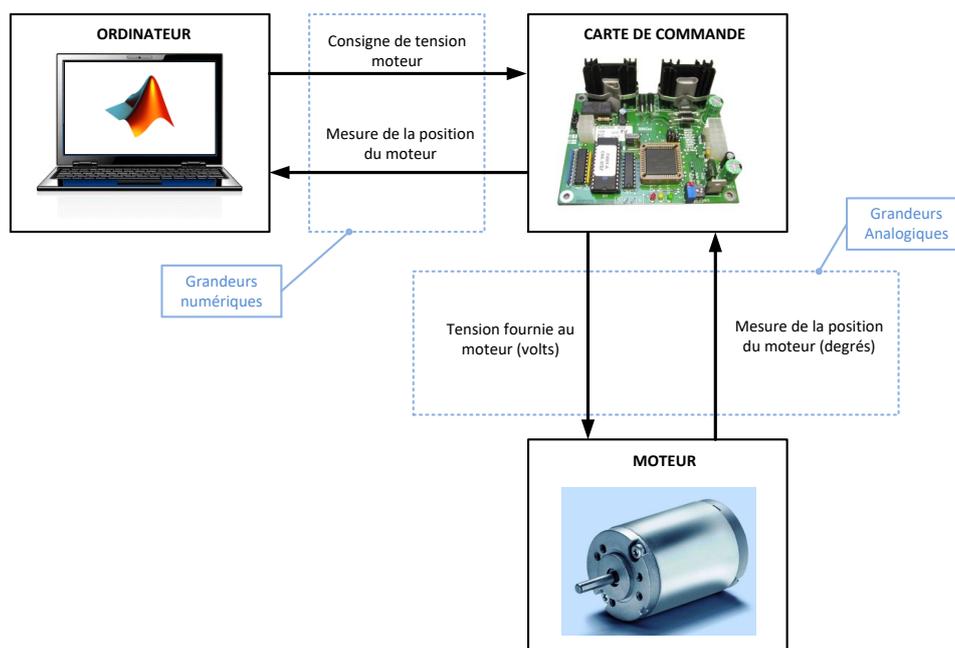


Figure 4 : architecture matérielle nécessaire à la mise en place de la démarche

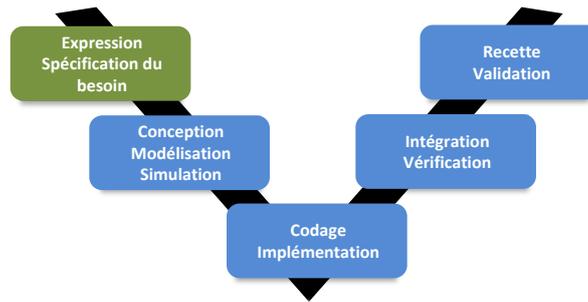


Figure 5 : la phase d'expression et de spécification du besoin

Le cycle démarre par **l'Expression et la Spécification** des besoins du système. Dans cette phase l'ensemble des exigences que doit remplir le système est formalisé. Les critères de performances sont définis et quantifiés. L'ensemble de ces informations est regroupé dans le cahier des charges fonctionnel. Il est analysé par les équipes de concepteur qui devront produire les solutions technologiques adéquates. Le diagramme des exigences de la Figure 6 regroupe les exigences de l'asservissement de position à concevoir.

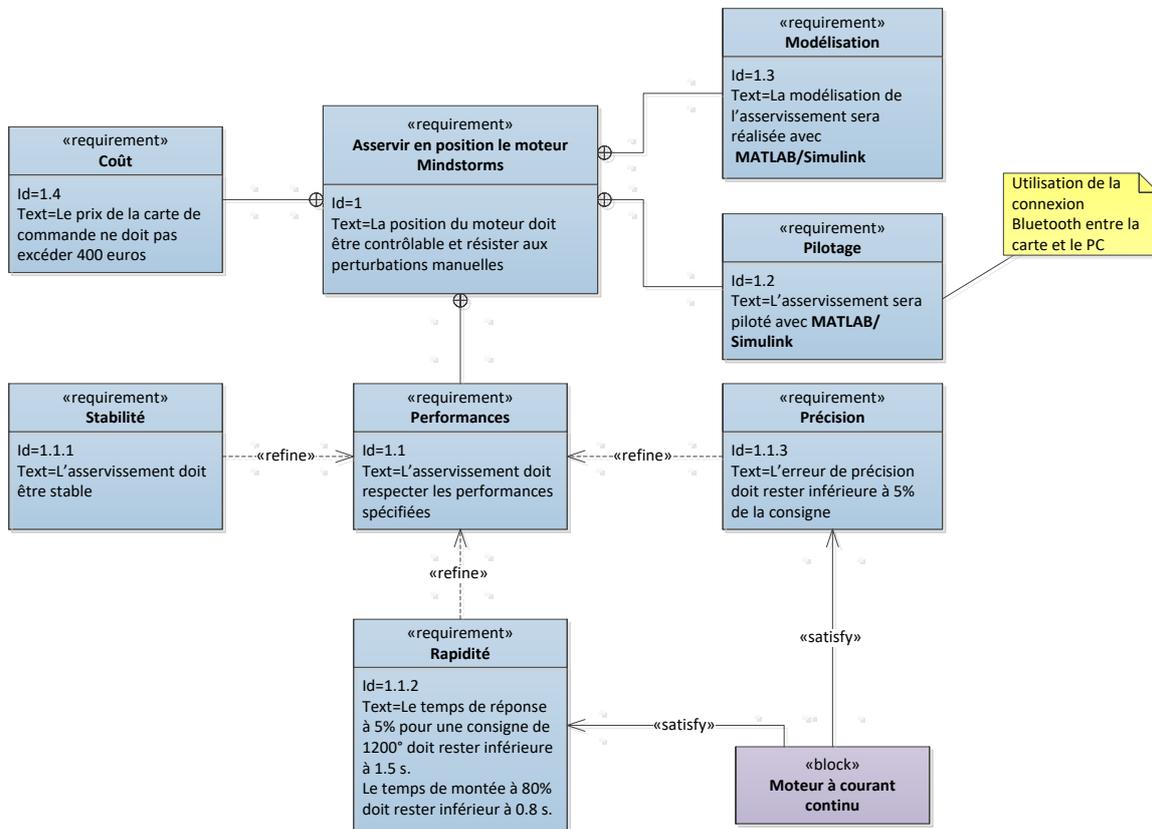


Figure 6 : diagramme des exigences de l'asservissement de position

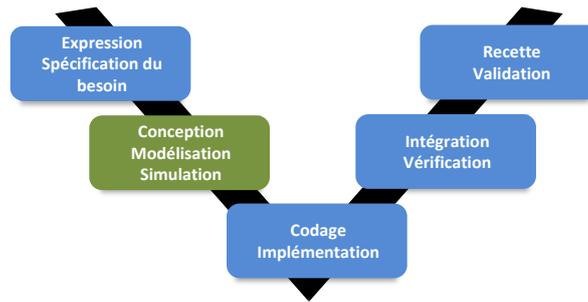


Figure 7 : la phase de Conception, Modélisation, Simulation

La phase de **Conception-Modélisation-Simulation** consiste à inventer, innover et proposer des solutions technologiques pour répondre aux exigences spécifiées. Dans cette phase sont définis, la structure du système, l'interconnexion des domaines physiques (mécanique, électrique, hydraulique, thermique...), les lois de commandes... Cette phase très complexe nécessite de pouvoir anticiper les comportements avant même que le système réel ne soit produit. C'est alors que le travail de modélisation va commencer. Les ingénieurs vont construire des modèles pour tenter de comprendre et d'anticiper les choix technologiques effectués. Cette étape requiert des connaissances théoriques et technologiques de haut niveau dans tous les domaines qui sont abordés. Afin de pouvoir exploiter les modèles et obtenir des résultats, la simulation de ces modèles à l'aide d'outils numériques adaptés doit alors être mise en œuvre.

Industriellement, mener des campagnes d'essais sur du matériel réel est très coûteux. Cette activité mobilise des moyens matériels et humains importants et impose des contraintes très fortes en termes de délais et de coût. L'un des objectifs majeurs est alors de limiter la durée de ces tests en optimisant la préparation de cette phase. Une solution consiste à remplacer les tests sur le matériel réel par des essais sur des modèles numériques reproduisant le comportement du système réel. Cette méthode appelée « Model Based Design » (conception basée sur le modèle) tend à devenir un standard de conception qui vient enrichir la démarche du cycle en V.

Cette méthode impose de disposer de modèles validés. Les techniques de modélisation et leur maîtrise par les équipes d'ingénieurs se retrouvent au cœur de l'activité de conception. L'évolution rapide des fonctionnalités des logiciels de simulation numérique ouvre de nouvelles possibilités de modélisation qui accélèrent le processus de création et de validation des modèles. La résolution numérique est indispensable au processus et le logiciel de simulation devient le partenaire incontournable de cette activité. Sa performance et ses fonctionnalités détermineront souvent la rapidité et l'efficacité du processus de modélisation. Différentes méthodes peuvent être mises en œuvre pour optimiser cette phase.

- La modélisation « white box » :
- La modélisation « multi-physique » :
- La modélisation « gray box » :
- La modélisation « black box » :

Ces différentes approches ne sont pas exclusives dans le processus de modélisation. Elles sont au contraire complémentaires et il est fréquent de les combiner pour mener la démarche avec la plus grande pertinence. La connaissance et la maîtrise de ces concepts par les étudiants et les ingénieurs leur permettent d'être compétents dans la démarche de modélisation. Dans tous les cas la confrontation à l'expérimentation et à la mesure permettra de valider le comportement du modèle.

Dans notre exemple, nous aurons besoin de créer le modèle numérique d'un moteur à courant continu afin de réaliser toute la phase de contrôle commande sur un modèle lorsqu'il sera validé.

La modélisation du moteur à courant continu consiste à établir la relation entre la tension aux bornes de l'induit du moteur et sa vitesse de rotation.

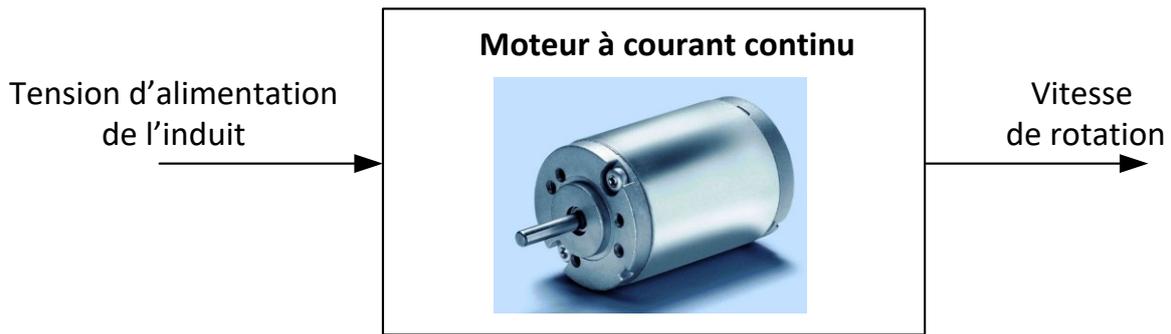


Figure 8 : relation entrée/sortie pour le moteur à courant continu

### 1. La modélisation « white box »

Les équations électrique, mécanique et de couplage d'un moteur représentent le modèle de connaissance du moteur à courant continu et sont détaillées sur la Figure 9 :

$$\left\{ \begin{array}{l} u(t) = e(t) + R i(t) + L \frac{d i(t)}{dt} \\ e(t) = K_e \omega_m(t) \\ J \frac{d \omega_m(t)}{dt} = C_m(t) - C_r(t) - f \omega_m(t) \\ C_m(t) = K_t i(t) \end{array} \right.$$

$u(t)$ : tension de commande du moteur (V)  
 $e(t)$ : force contre électromotrice du moteur (V)  
 $i(t)$ : intensité dans le moteur (A)  
 $C_m(t)$ : couple exercé par le moteur (N.m)  
 $C_r(t)$ : couple résistant ramené sur l'arbre moteur (N.m)  
 $\omega_m(t)$ : vitesse de rotation du moteur (rad/s)  
 $R$  : résistance de l'induit ( $\Omega$ )  
 $L$  : inductance de l'induit (H)  
 $J$  : inertie équivalente ramenée sur l'arbre moteur ( $\text{kg.m}^2$ )  
 $f$  : paramètre de frottement visqueux (N.m.s)  
 $K_t$  : constante de couple (N.m/A)  
 $K_e$  : coefficient de force contre électromotrice (V.s/rad)

Figure 9 : équations de comportements du moteur à courant continu

Parmi les paramètres qui caractérisent le comportement du moteur, certains sont parfaitement connus et d'autres restent indéterminés. L'inductance  $L$  et la résistance  $R$  de l'induit sont données dans la documentation technique du moteur, il en va de même pour les coefficients  $K_e$  et  $K_t$ . Par contre le paramètre de frottement visqueux  $f$  n'est pas forcément connu, en particulier si le moteur est ensuite couplé à un mécanisme. Il faudra soit négliger ce paramètre  $f$  ou alors en faire une estimation pour espérer obtenir des résultats valides avec le modèle.

Après avoir appliqué la transformée de Laplace à ces équations, il est facile d'obtenir un modèle numérique sous forme de schéma bloc que l'on pourra modéliser et résoudre à l'aide du logiciel de simulation numérique (Figure 10). Ce principe de résolution est basé sur le principe de causalité. Le comportement dynamique d'un système est caractérisé par un bloc contenant, par exemple, la fonction de transfert du système avec une entrée et une sortie. L'information circulant dans les connexions entre deux blocs est un signal numérique orienté. La sortie du bloc est calculée numériquement en déterminant pour chaque pas de calcul la transformation du signal d'entrée imposé par le contenu du bloc. Cette approche largement utilisée industriellement nécessite une parfaite connaissance des lois physiques qui caractérisent le comportement des systèmes.

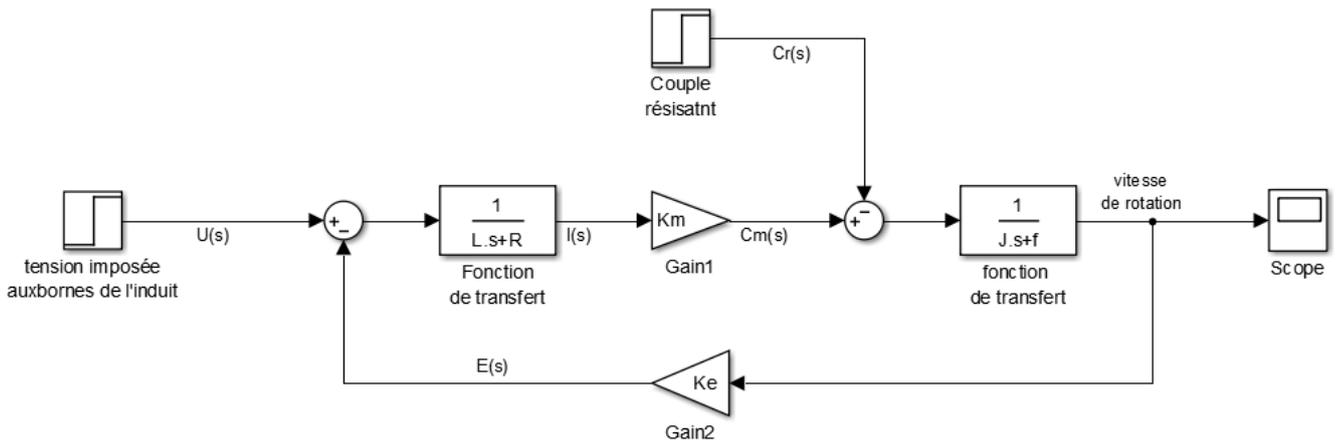


Figure 10 : modélisation sous forme de schéma bloc du moteur à courant continu

Il est maintenant possible de remplacer le système par son modèle comme représenté sur la Figure 11.

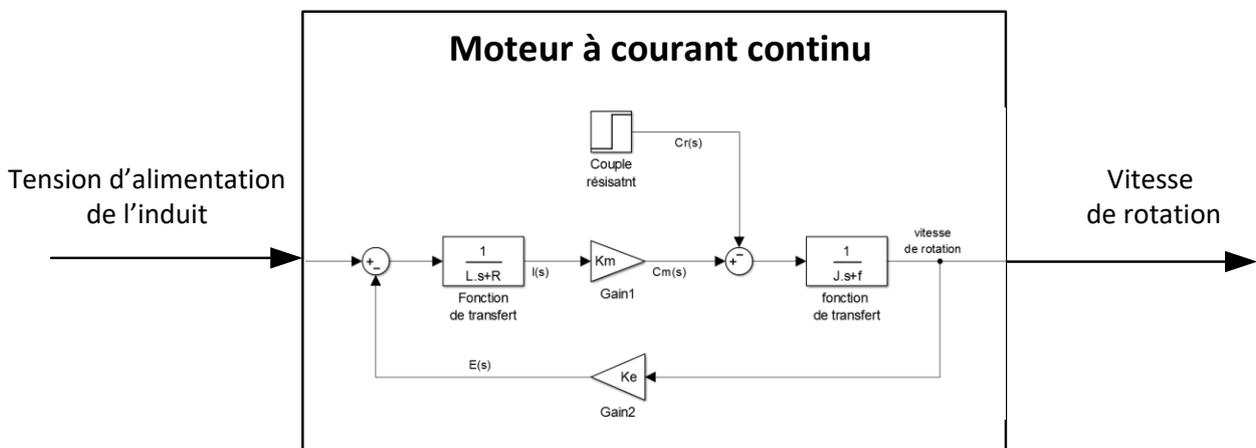


Figure 11 : modélisation « white box » du moteur à courant continu

## 2. La modélisation « Multi-Physique »

La modélisation multi-physique propose une approche acausale de la modélisation et permet de faire de la modélisation par assemblage de composants issus de bibliothèques prédéfinies. Le comportement physique des composants est pris en compte directement par le logiciel, il est donc possible de modéliser un système sans avoir à écrire les équations différentielles qui caractérise son comportement.

Les connexions entre deux composants ne sont pas orientées, ont une signification physique et transmettent un niveau d'information supérieure aux connexions en modélisation causale. Ces connexions peuvent être un fil électrique (transfert d'information de type courant et tension), un arbre moteur (transfert d'information de type couple et vitesse angulaire), l'extrémité de la tige d'un vérin (transfert d'information de type force et vitesse linéaire)... Le principe de calcul s'appuie sur un bilan de puissance à chaque nœud du modèle et ne repose pas sur le principe de causalité, d'où le nom de modélisation acausale.

La modélisation multi-physique du moteur à courant continu est représentée sur la Figure 12.

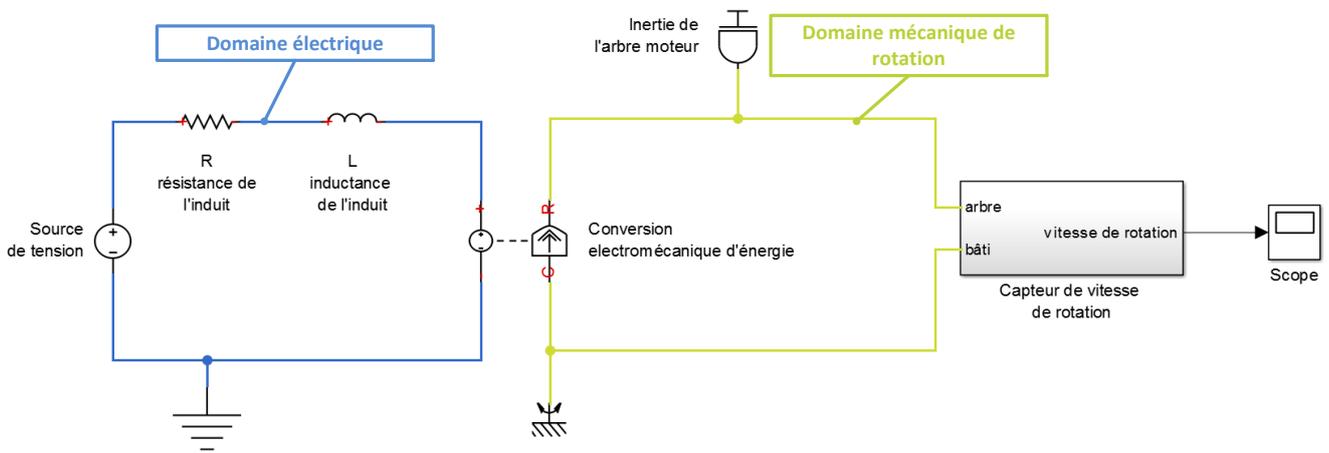


Figure 12 : modélisation par assemblage de composants du moteur à courant continu

Il est maintenant possible de remplacer le système par son modèle comme représenté sur la Figure 13.

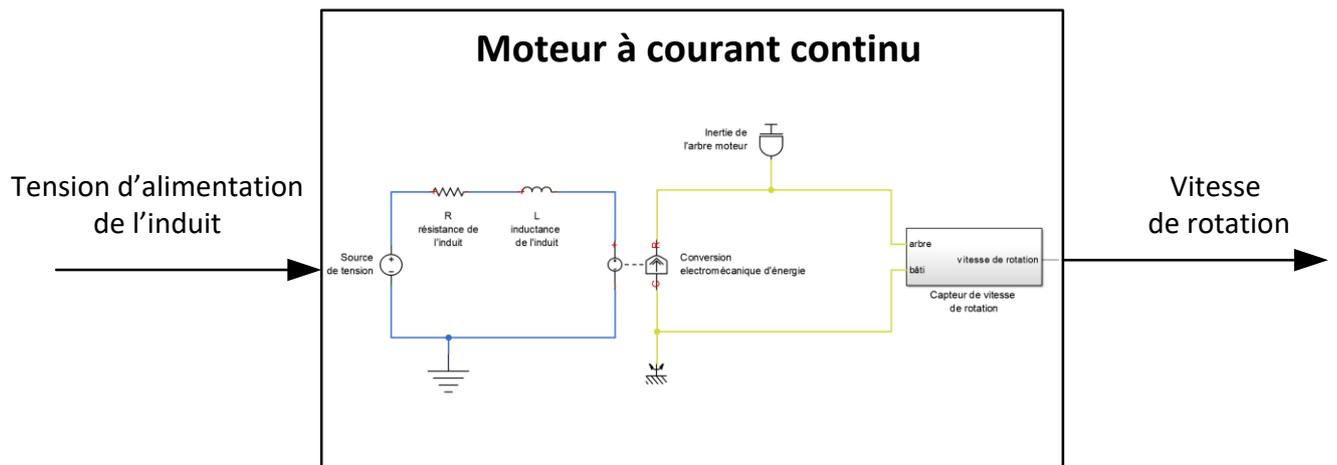


Figure 13 : modélisation multi-physique acausal du moteur à courant continu

### 3. La simulation des modèles

La simulation et la résolution des modèles par le logiciel de simulation imposent de connaître les valeurs de tous les paramètres numériques. Dans notre exemple tous les paramètres sont connus sauf le paramètre  $f$  caractérisant les frottements. Il sera donc considéré comme nulle en première approximation ce qui revient à négliger l'influence des frottements.

Quelle que soit l'approche choisie, de type « white box » ou « multi-physique », les résultats obtenus par simulation numériques sont identiques (Figure 14). Dans les deux cas les phénomènes physiques pris en compte sont les mêmes, la différence réside dans la méthode utilisée pour construire le modèle et la méthode de résolution numérique utilisée qui diffère.

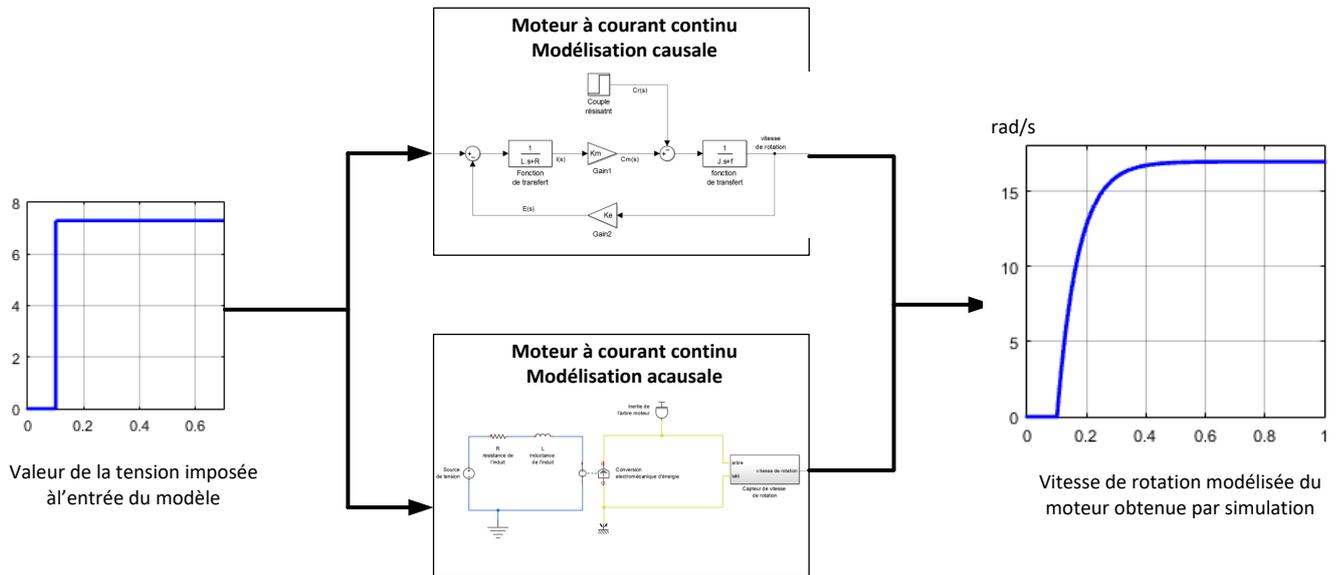


Figure 14 : résolution des modèles et visualisation des résultats

Les résultats obtenus permettent d'avoir une première approximation de l'évolution de la vitesse de rotation du moteur en fonction du temps. Dans le cas général cette approche est très souvent insuffisante et le seul moyen de vérifier la validité du modèle est d'évaluer les écarts entre les performances modélisées et les performances mesurées sur le système réel. En imposant une tension identique aux bornes de l'induit du moteur réel, il est possible de relever l'évolution de la vitesse de rotation réelle du moteur comme le montre la Figure 15.

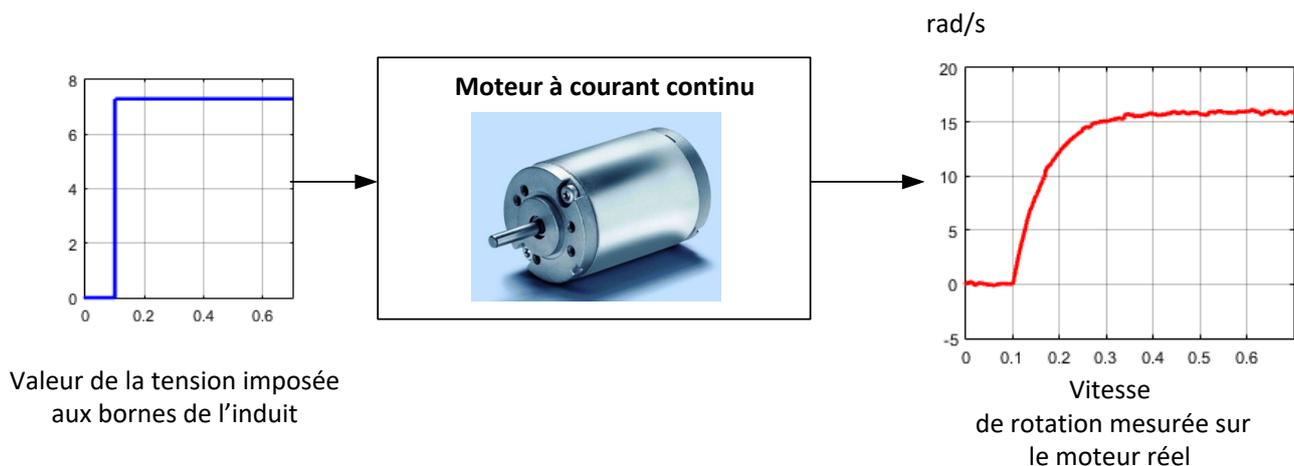


Figure 15 : mesure de l'évolution de la vitesse de rotation du moteur en fonction du temps

#### 4. La comparaison des performances simulées et mesurées

Les performances simulées et mesurées sont alors comparées sur la Figure 16.

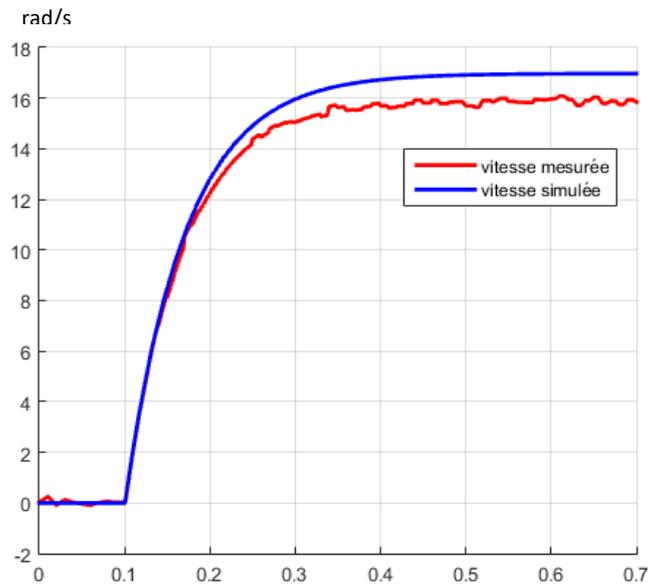


Figure 16 : évaluation des écarts entre les performances simulée et les performances mesurées

Dans le cas général et à ce stade, la mise en évidence d'écarts de comportement entre le modèle et le réel est très fréquent. Il faut alors évaluer ces écarts et trouver des pistes pour améliorer la modélisation en prenant en compte un phénomène nouveau ou en adaptant la valeur de certains paramètres.

Dans notre exemple l'analyse des écarts montre que le modèle n'est pas validé. L'absence de prise en compte des frottements est à l'origine de cet écart. Il faut alors évaluer les paramètres inconnus, ici le paramètre de frottement  $f$ . L'approche de type « gray box » va permettre de compléter le processus de modélisation en proposant une méthode de détermination des paramètres inconnus.

#### 5. La modélisation « gray box »

Le principe de cette modélisation est d'estimer les paramètres inconnus du modèle par comparaison entre les performances simulées et les performances mesurées. Un algorithme intégré au logiciel de simulation va effectuer le travail d'analyse des écarts et proposer des valeurs possibles pour les paramètres inconnus permettant l'annulation des écarts. La méthode consiste à effectuer un essai sur le système réel en relevant le signal imposé à l'entrée du système et le signal obtenu à la sortie (Figure 17).

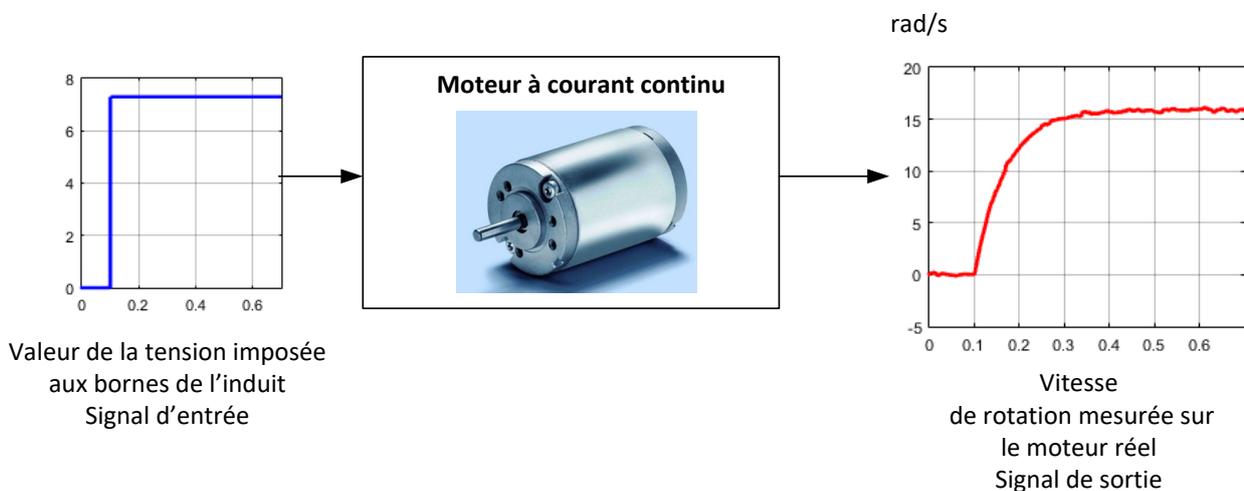


Figure 17 : essai réalisé en vue de l'estimation des paramètres inconnus

Un signal d'entrée identique est alors imposé à l'entrée du modèle et le logiciel va donner des valeurs possibles pour les paramètres inconnus de manière à annuler les écarts entre le réel et le modèle. Le modèle va chercher à atteindre la réponse cible, celle du système réel (Figure 18). Il est possible d'utiliser aussi bien le modèle causal que le modèle acausal pour réaliser cette optimisation.

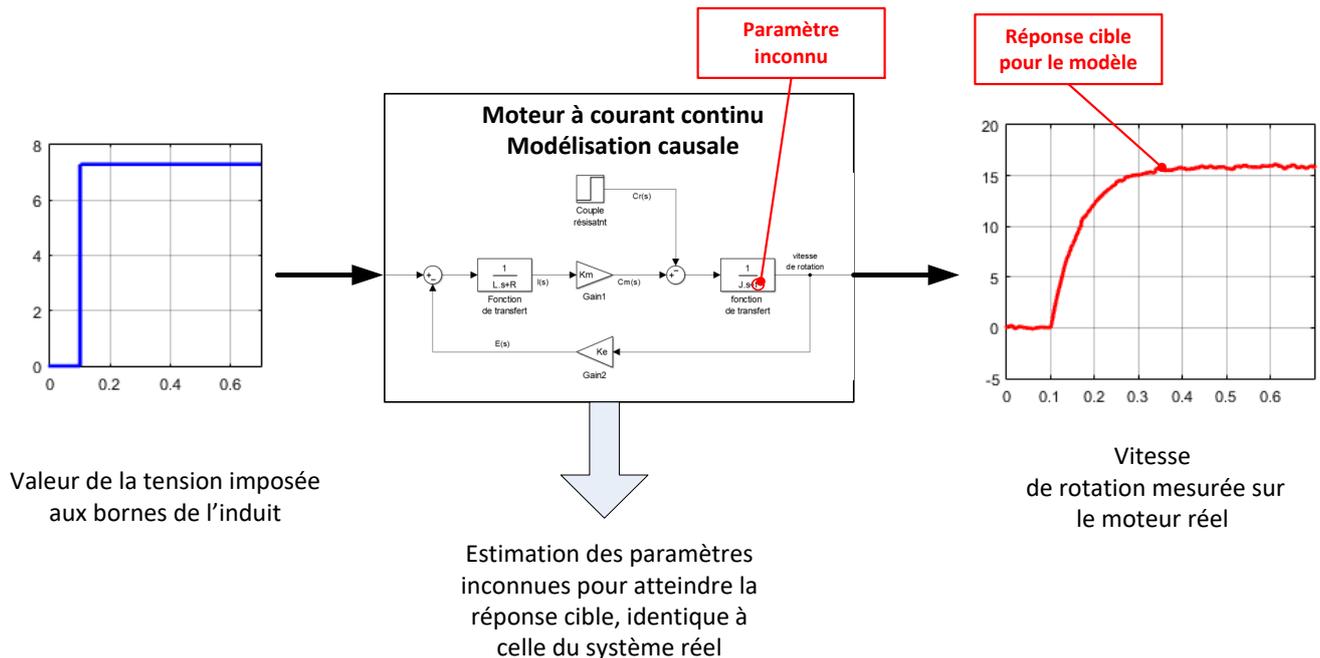


Figure 18 : illustration du principe de l'estimation de paramètre

La Figure 19 montre la réponse cible en rouge que va chercher à atteindre l'algorithme d'estimation de paramètres. La réponse du modèle avant estimation des paramètres est représentée sur la courbe bleue et la réponse du modèle après estimation des paramètres sur la courbe verte. L'intérêt est évident et permet d'obtenir un modèle validé dont le comportement est très proche de celui du système réel. Les écarts entre les performances mesurées et les performances simulées ont été annulés.

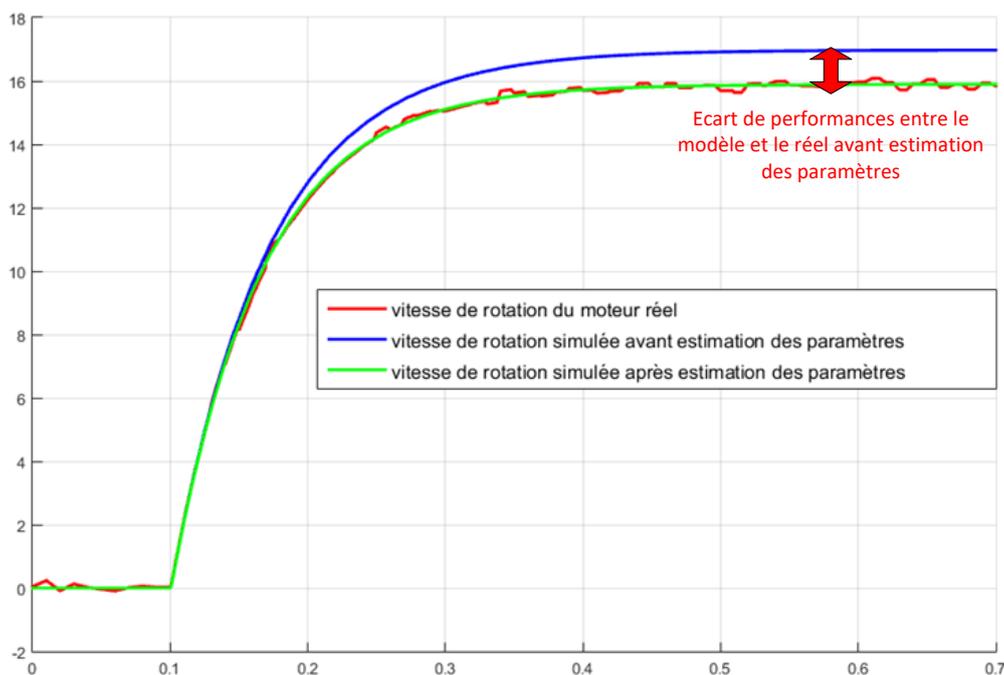


Figure 19 : les réponses du modèle avant et après estimation des paramètres inconnus

A l'issue de cette phase le projet dispose de modèles validés. Il est alors possible de remplacer le matériel (ici le moteur) par son modèle. La mise en œuvre de la phase de conception et de réglage de la commande

se fera uniquement à l'aide du logiciel de simulation en travaillant sur le modèle. Les tests sur le matériel réel se feront après que les performances du modèle atteignent les performances souhaitées par le cahier des charges.

Nous pouvons alors utiliser le modèle validé du moteur pour construire l'asservissement en position du moteur à courant continu représenté sur la Figure 20.

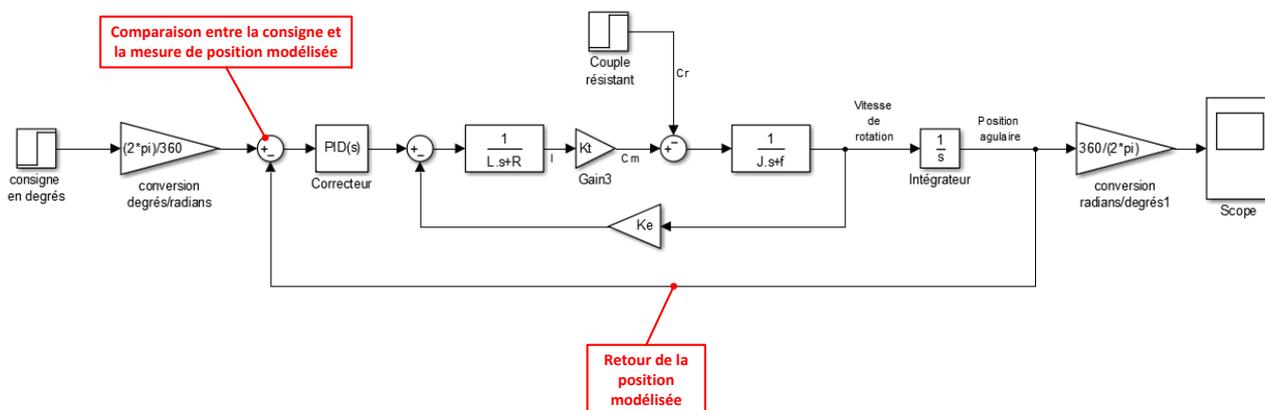


Figure 20 : modélisation de l'asservissement de position du moteur

Dans un premier temps le correcteur n'est pas actif et la réponse de l'asservissement de position est relevée sur la Figure 21.

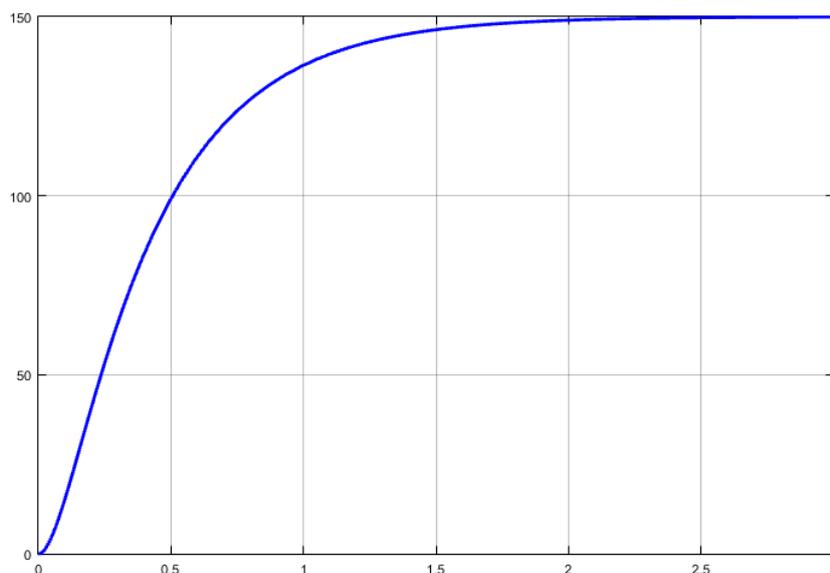


Figure 21 : réponse de l'asservissement de position non corrigé

Le diagramme des exigences impose les critères de performance suivants :

- Temps de réponse à 5% inférieur à 0.8 s
- Temps de montée à 80% inférieur à 0.4 s
- Ecart statique inférieur à 5% de la consigne

Il apparaît que les performances modélisées de l'asservissement ne sont pas conformes aux performances souhaitées. Le temps de réponse à 5% est de l'ordre de 1.3 s et le temps de montée à 80 % de l'ordre de 0.7s.

Il faut donc optimiser la commande du moteur en procédant au réglage du correcteur PID. Le logiciel de simulation propose alors de nombreuses fonctionnalités permettant de mettre en œuvre la démarche.

La Figure 22 illustre l'utilisation de la démarche de réglage d'un correcteur dans le domaine fréquentiel à l'aide d'outils spécifiques.

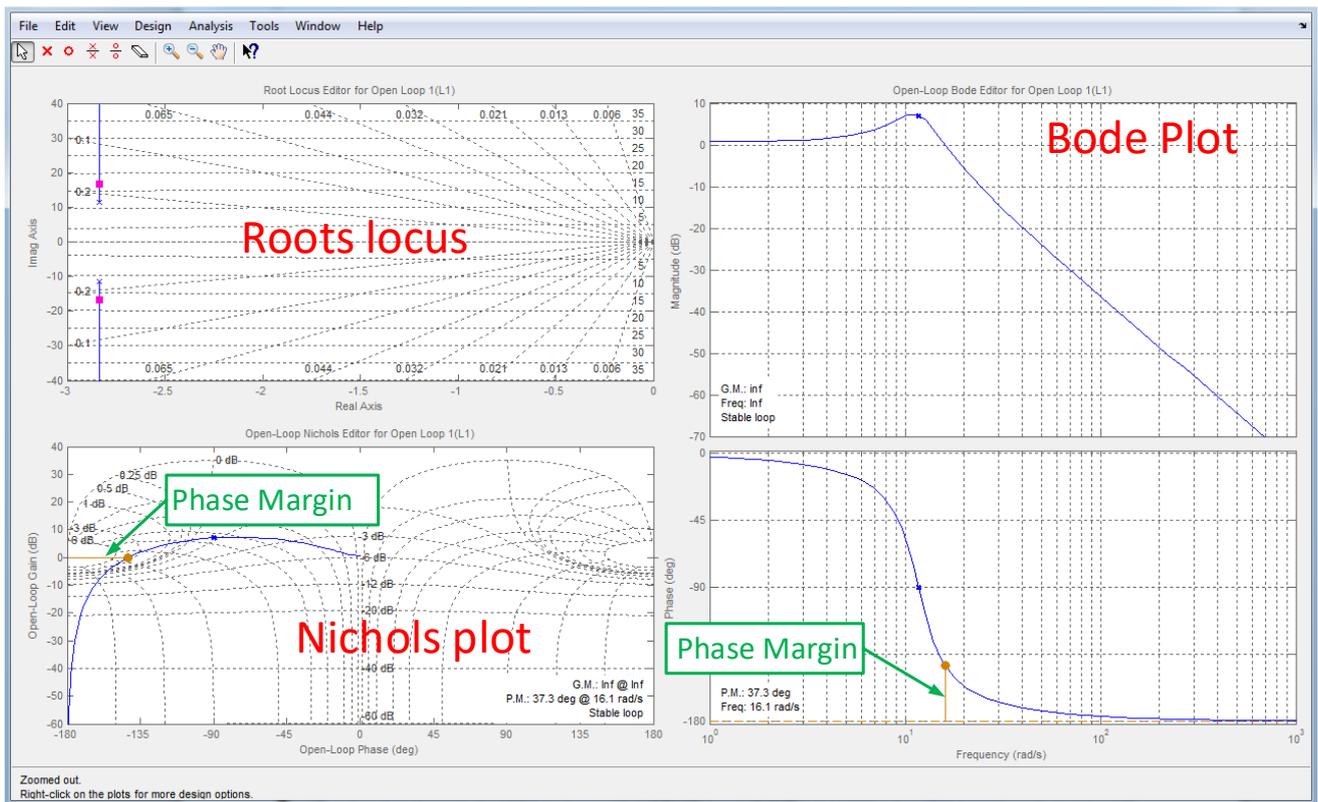


Figure 22 : illustration de l'utilisation d'un outil de contrôle commande pour régler un correcteur

Ces méthodes seront largement développées dans la suite de l'ouvrage et demandent de réaliser de nombreux essais sur le modèle. Le modèle de l'asservissement étant construit sur la base du modèle validé du moteur les résultats obtenus par simulation seront très proches des résultats mesurés sur le système réel.

A l'issue de cette phase de réglage, le correcteur est déterminé pour obtenir la réponse représentée sur la Figure 23.

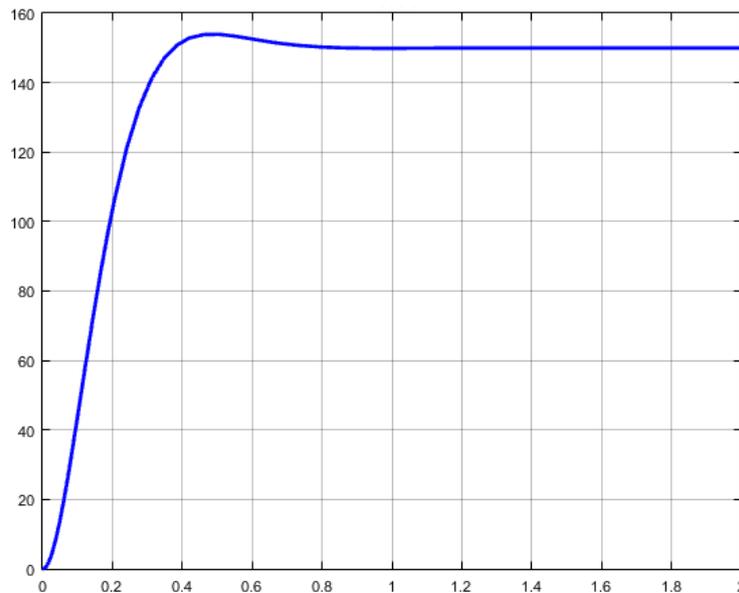


Figure 23 : réponse de l'asservissement de position corrigée

Le temps de réponse à 5% est maintenant de l'ordre de 0.3 s et le temps de montée à 80 % de l'ordre de 0.25s. Les performances modélisées sont conformes aux performances souhaitées et la phase de **Codage Implémentation** va pouvoir commencer.

## 6. Le Model-in-the-loop (MIL)

Cette étape se situe entièrement dans le domaine de la simulation et correspond à la simulation du modèle complet précédemment décrit. Le modèle est décrit entièrement en code MATLAB, il s'exécute entièrement sur l'ordinateur avec lequel a été construit le modèle.

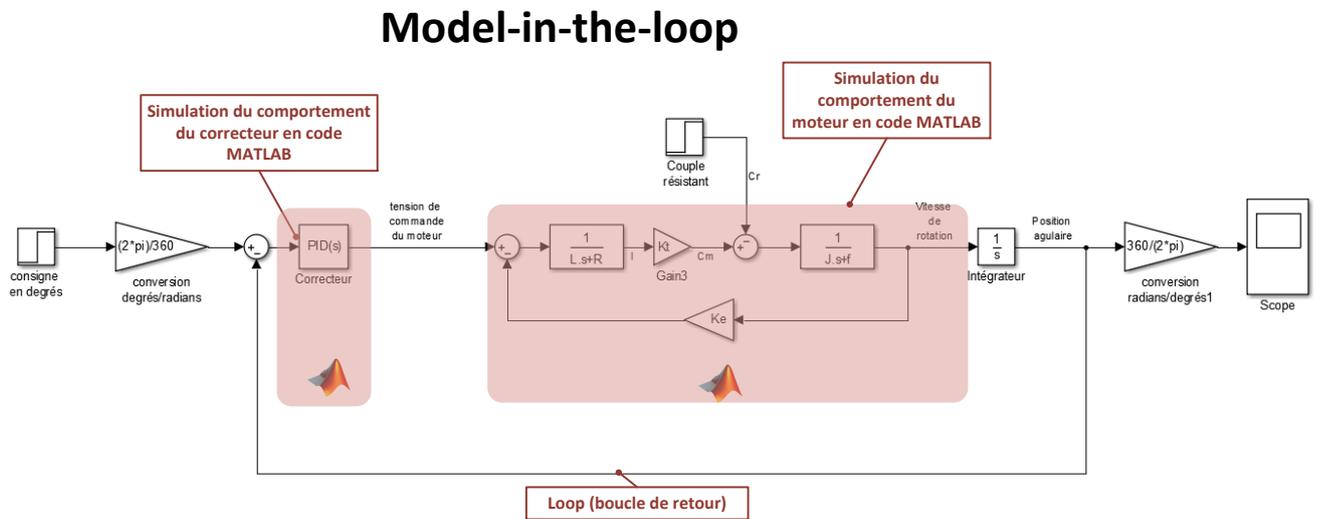


Figure 24 : le Model-in-the-loop (MIL)

## D. La phase de Codage Implémentation

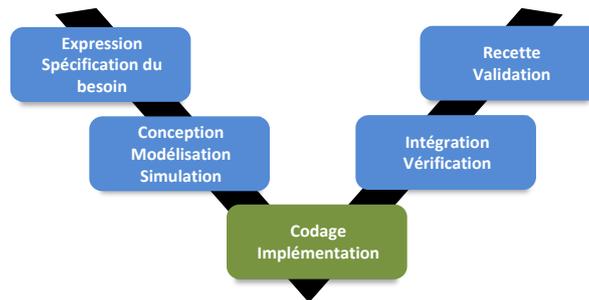


Figure 25 : la phase de Codage Implémentation

Cette phase permet de migrer progressivement du domaine de la simulation vers la phase de test sur le matériel réel.

A ce stade il est important de revenir sur la structure du modèle validé qui vient d'être finalisée à l'issue de la phase de **Conception Modélisation Simulation**. La Figure 26 montre les deux parties du modèle qu'il faudra bien distinguer pour comprendre les étapes qui vont suivre.

- la partie du modèle correspondant au correcteur qui était à la base des objectifs de conception du projet.
- la partie du modèle correspondant au moteur qui permet de disposer d'une modélisation validée des actionneurs.

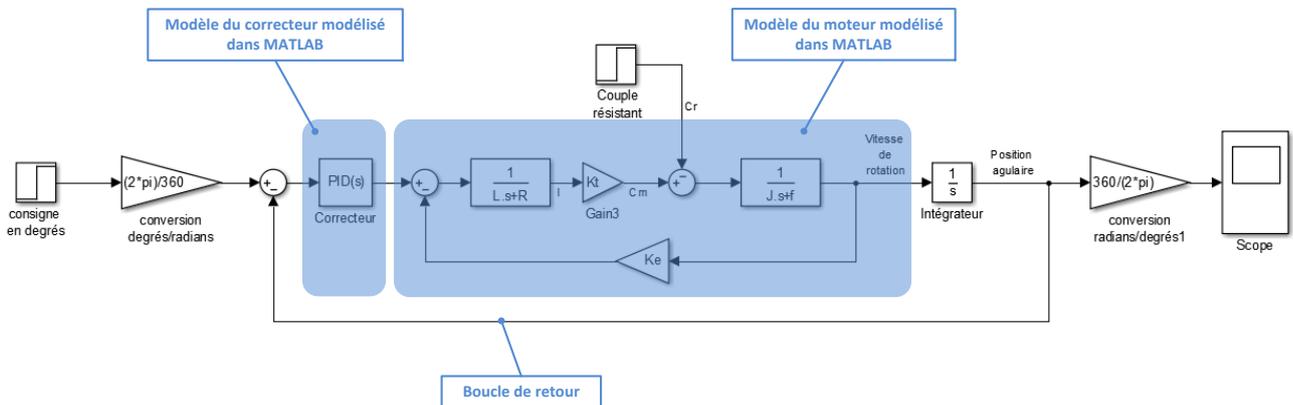


Figure 26 : structure du modèle à l'issue de la phase de Conception Modélisation Simulation

Cette phase permet de traduire des parties du modèle en Code C. Le code C est le langage de programmation utilisé par la carte de commande qui ne peut pas interpréter directement le code MATLAB. Une première méthode consiste alors à isoler la partie du modèle que l'on veut remplacer par du code C et à faire le bilan des entrées et des sorties de cette partie du modèle. Il est possible de réaliser ce codage manuellement. Dans ce cas une équipe de codeurs spécialiste du langage C vont réaliser les programmes qui généreront la même relation entre les entrées et les sorties que celle obtenue avec du code MATLAB. Cette approche est longue et mobilise des compétences de haut niveau en codage et en programmation.

Les logiciels de simulation permettent également de générer du code C automatiquement et MATLAB propose des interfaces permettant d'automatiser ce processus afin de gagner un temps précieux.

Qu'il soit automatique ou manuel, le processus de codage en langage C, peut générer des erreurs et le comportement du modèle codé en langage C peut différer de son comportement codé en MATLAB. Afin de détecter et de corriger ces erreurs avant qu'elles ne puissent impacter la phase de test sur du matériel réel, il faut mettre en place deux configurations de fonctionnement :

- le software-in-the-loop (SIL)
- le processor-in-the-loop (PIL)

### 1. Le Software-in-the-loop (SIL)

Dans cette configuration, une partie du modèle est remplacé par du code C qui va s'exécuter sur l'ordinateur. Cela permet de vérifier que la génération de code C n'a pas engendré d'erreur en termes de traduction de code C vers le code MATLAB.

Dans notre exemple nous allons remplacer le correcteur codé dans MATLAB par une fonction C qui va générer du code C en remplacement du code MATLAB et exécuter ce code C sur l'ordinateur (Figure 27).

## Software-in-the-loop

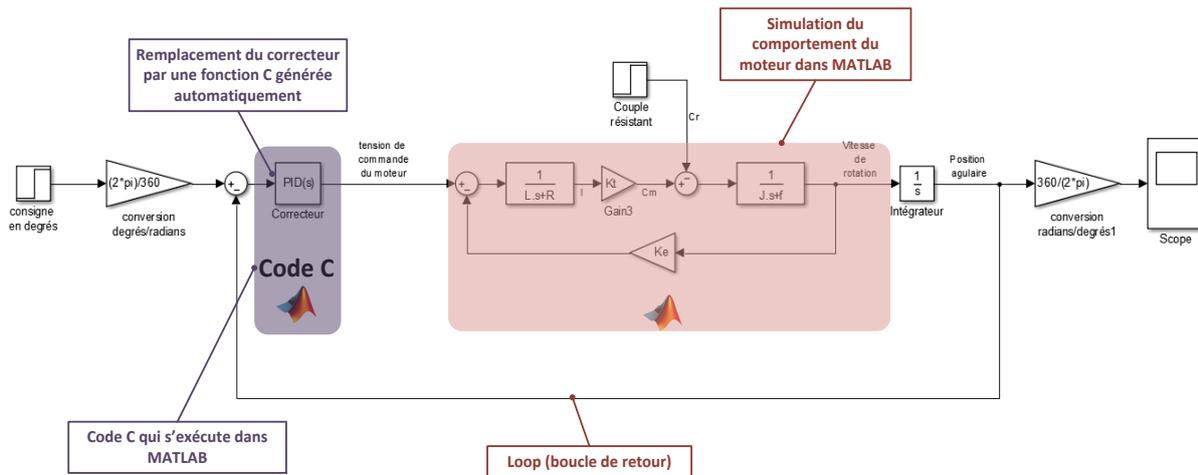


Figure 27 : le Software-in-the-loop(SIL)

Il faut alors vérifier que le comportement du modèle est le même, qu'il soit piloté avec le correcteur codé en code MATLAB ou avec le correcteur codé en code C. Une fois cette vérification effectuée, les concepteurs peuvent exclure toute erreur due à la génération de code C.

## 2. Le Processor-in-the-loop (PIL)

Cette configuration permet d'éviter les erreurs d'exécution du code C. En effet les conditions d'exécution du code C dépendent largement des capacités et des performances de la cible matérielle sur laquelle il s'exécute. En mode software-in-the-loop, le code s'exécute sur le processeur de l'ordinateur, en mode processor-in-the-loop, le code s'exécute sur la cible matérielle qui embarquera le code dans les conditions réelles de fonctionnement. Ces deux processeurs peuvent gérer différemment l'exécution d'un même programme. Le correcteur modélisé est alors remplacé par du code C qui s'exécutera directement sur la carte de commande. Dans cette phase, le processeur de la carte pilote le modèle du moteur.

## Processor-in-the-loop

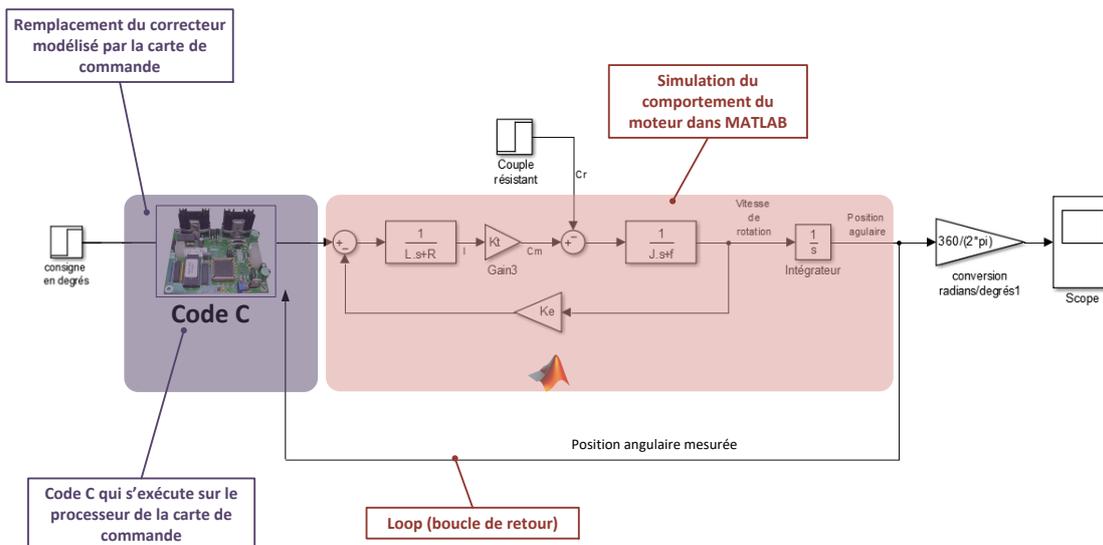


Figure 28 : le Processor-in-the-loop

A l'issue de ces deux phases, les erreurs de génération et d'exécution de code C ont été corrigées et la phase de test sur les actionneurs réels va pouvoir commencer.

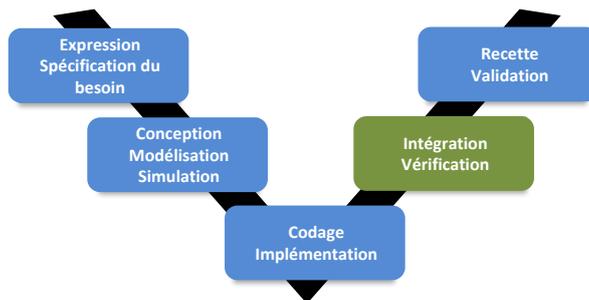


Figure 29 : la phase d'Intégration Vérification

### 1. Le Hardware-In-the-Loop (HIL)

Cette étape consiste à remplacer le modèle du moteur par le moteur réel. Le correcteur et les calculs associés peuvent être effectués soit par l'ordinateur (mode externe), soit par la carte de commande qui fonctionne de manière autonome (mode embarqué).

*HIL - mode externe (Figure 30 et Figure 31)*

Dans cette phase tous les calculs sont réalisés sur l'ordinateur et la carte de commande fonctionne uniquement en interface de puissance. Les fonctions de calcul du processeur de la carte de commande ne sont pas sollicitées.

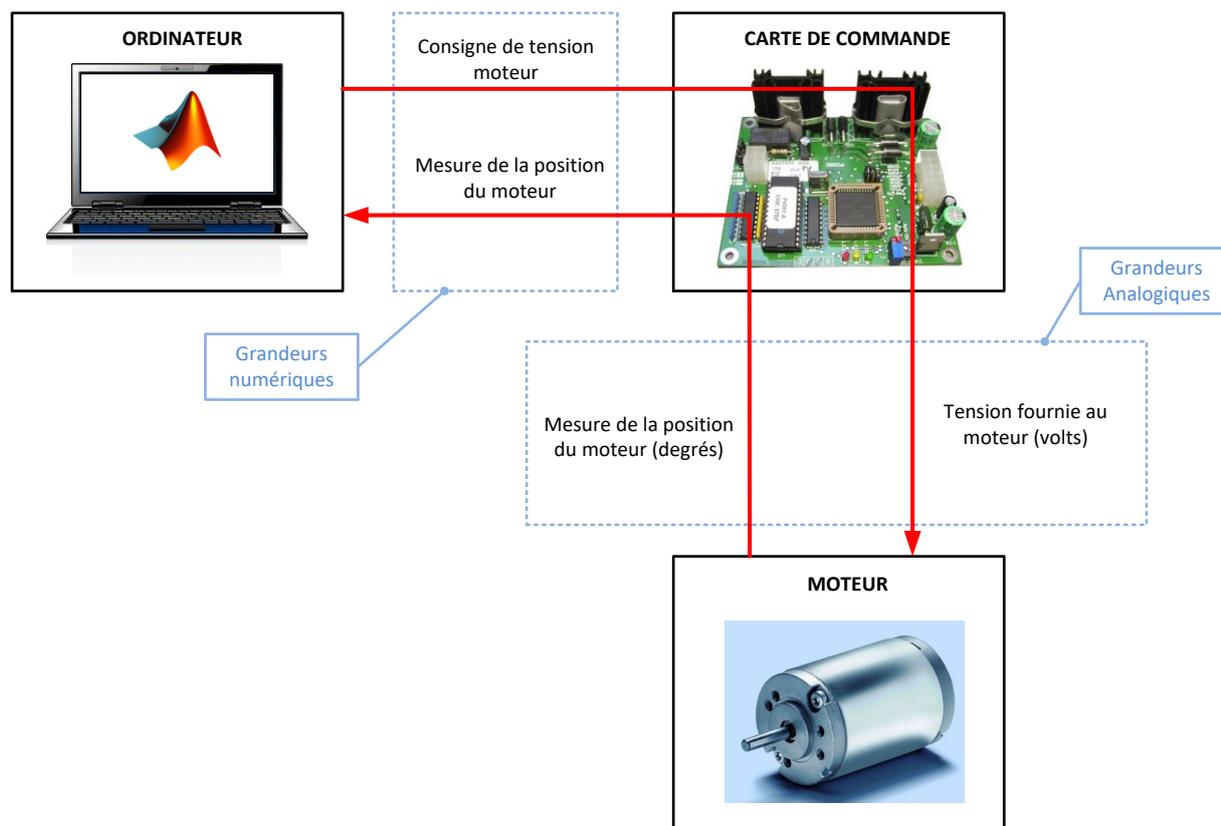


Figure 30 : architecture matérielle du hardware-in-the-loop en mode externe

# Hardware-in-the-loop Mode externe

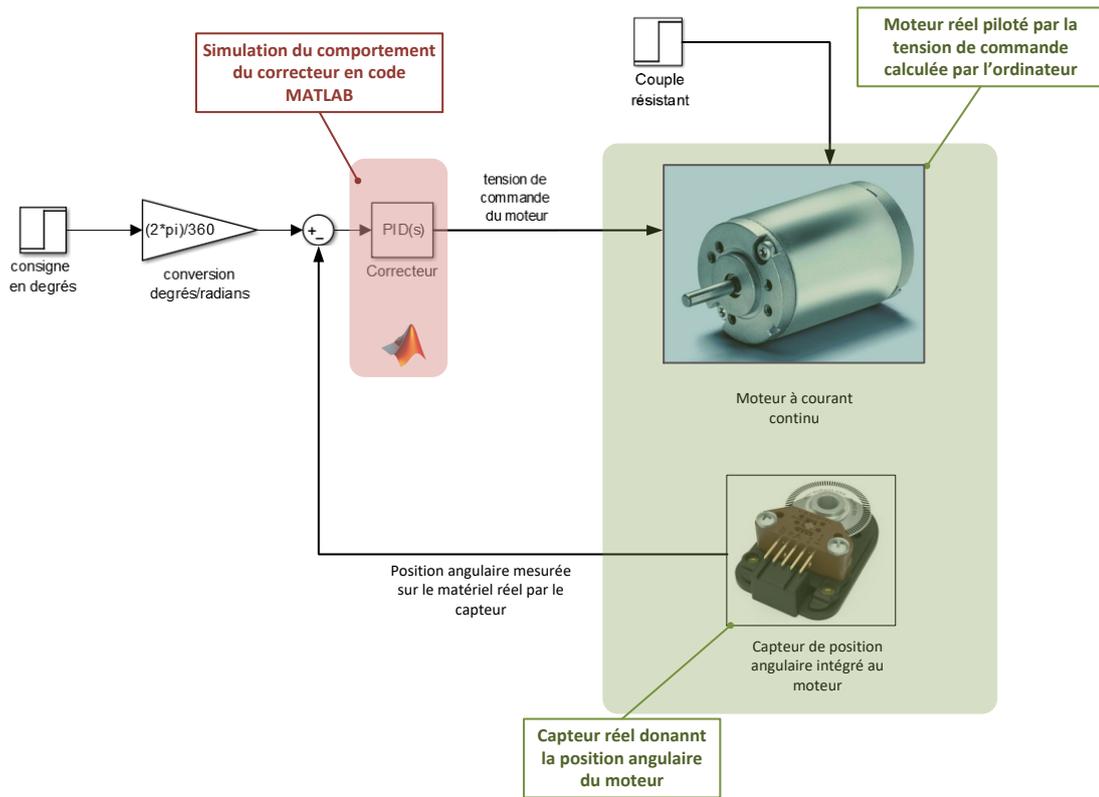


Figure 31 : le hardware-in-the-loop en mode externe: intégration du matériel dans la boucle

## HIL - mode embarqué (Figure 32 et Figure 33)

Dans cette phase tous les calculs sont réalisés par la carte de commande. La logique de commande s'exécute sur le processeur de la carte de commande.

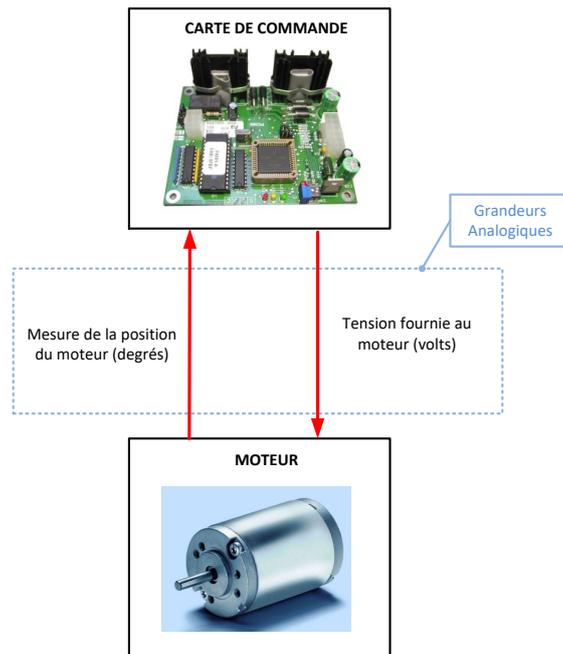


Figure 32 : architecture matérielle du Hardware-In-the-Loop en mode embarqué

# Hardware-in-the-loop Mode embarqué

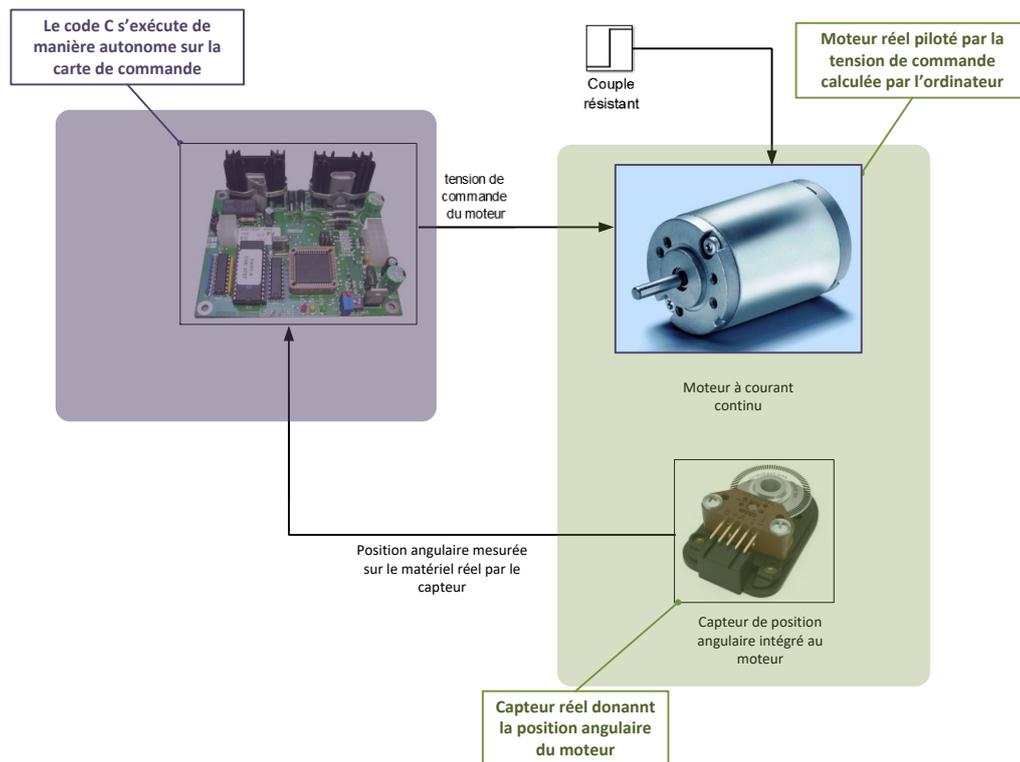


Figure 33 : le Hardware-In-the-Loop mode embarqué: fonctionnement autonome du matériel

L'une ou l'autre de ces configurations permettent de vérifier que le correcteur mis au point permet d'atteindre les performances spécifiées sur le matériel réel. Il est alors possible de relever la vitesse effective du moteur sur la Figure 34. Il est alors possible de passer à la phase de **Validation**.

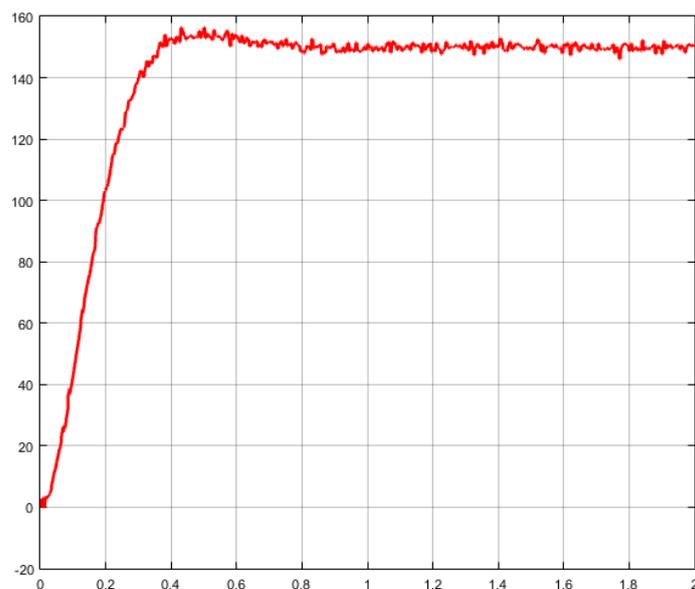
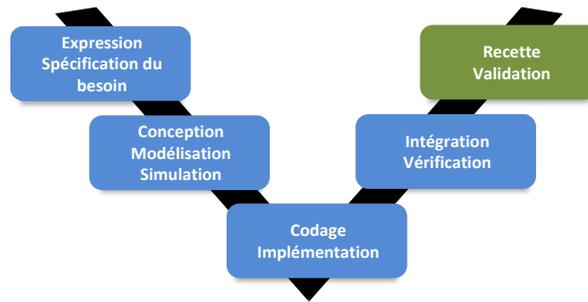


Figure 34 : visualisation de la vitesse de sortie du moteur à l'issu de la phase Hardware-in-the-loop



**Figure 35 : la phase de Validation Recette**

Dans cette phase, il reste à vérifier que les performances mesurées correspondent aux performances spécifiées dans le cahier des charges. L'analyse de la Figure 34 permet de voir que le temps de réponse à 5% est de l'ordre de 0.3 s et le temps de montée à 80 % de l'ordre de 0.25 s. Les performances réelles sont conformes aux performances modélisées et aux performances souhaitées.

La phase de test a été réduite au strict minimum et les spécifications imposées par le cahier des charges sont respectées.

# Chapitre 2 : Introduction et présentation des outils de modélisation

---

## I. Le logiciel MATLAB-Simulink

---

Le logiciel **MATLAB-Simulink** propose de nombreux outils permettant de mettre en œuvre une démarche de modélisation globale multi-physique. La première étape est de sélectionner une palette d'outils adaptés permettant de modéliser les systèmes dans le cadre de la formation des ingénieurs.

Il est préférable, avant d'aborder la phase de prise en main de ces outils, d'avoir une vision globale de leurs fonctionnalités et de leur approche causale ou acausale. La pertinence de la démarche de modélisation réside avant tout dans le choix de l'outil adapté pour modéliser les différentes parties du système. La maîtrise structurelle de ces outils et de leur potentiel garantira la réussite du processus de modélisation.

Dans un premier temps, nous allons décrire les fonctionnalités de ces outils. Ensuite nous ferons le lien entre ces outils et le descripteur « Chaîne d'énergie /Chaîne d'information » afin de mettre en place une stratégie de modélisation globale des systèmes.

### A. Description et hiérarchie des outils utilisés

---

#### 1. MATLAB

**MATLAB** est un puissant outil de calcul permettant de saisir des instructions sous la forme de lignes de commandes. **MATLAB** possède des bibliothèques de fonctions regroupées dans des « Toolbox » relatives à un domaine de connaissance. **Control System Toolbox** pour le contrôle commande des systèmes, **Signal Process Toolbox** pour le traitement du signal...

La Figure 36 présente un exemple de programme écrit en MATLAB (script) et la Figure 37 présente les résultats obtenus.

```

1
2   % Ce script permet de tracer sur une même fenêtre graphique les courbes
3   % f(t) = sin (a.t) avec un paramètre a variant de 1 à 4 et la variable
4   % theta variant de 0 à 2*pi
5
6   % création du vecteur theta
7   theta = [0:0.01:2*pi];
8
9   % création d'une nouvelle fenêtre graphique
10  figure;
11
12  % la fonction hold all permet de garder tous les tracés sur la même fenêtre
13  % chaque courbe sera tracée avec une couleur différente
14  hold all;
15
16  % pour a prenant les valeurs entre 1 et 4 avec un pas de 1
17  for a = 1:4
18      % tracé de la fonction f(theta) = sin(a*theta) pour theta
19      % compris entre 0 et 2*pi
20      % LineWidth permet de spécifier une épaisseur de ligne
21      plot(theta,sin(a*theta), 'LineWidth',2);
22  end
23
24  grid on;
25  % spécification du titre du graphique et des grandeurs physiques des axes
26  title('Fonction Sinus')
27  % MATLAB intègre un interpréteur TeX
28  xlabel('angle \theta en radians');
29  ylabel('f(\theta)=sin(a.\theta)');
30
31  % Spécification des limites des axes
32  axis([-1 7 -1.5 1.5]);

```

Figure 36 : exemple de script écrit en langage MATLAB

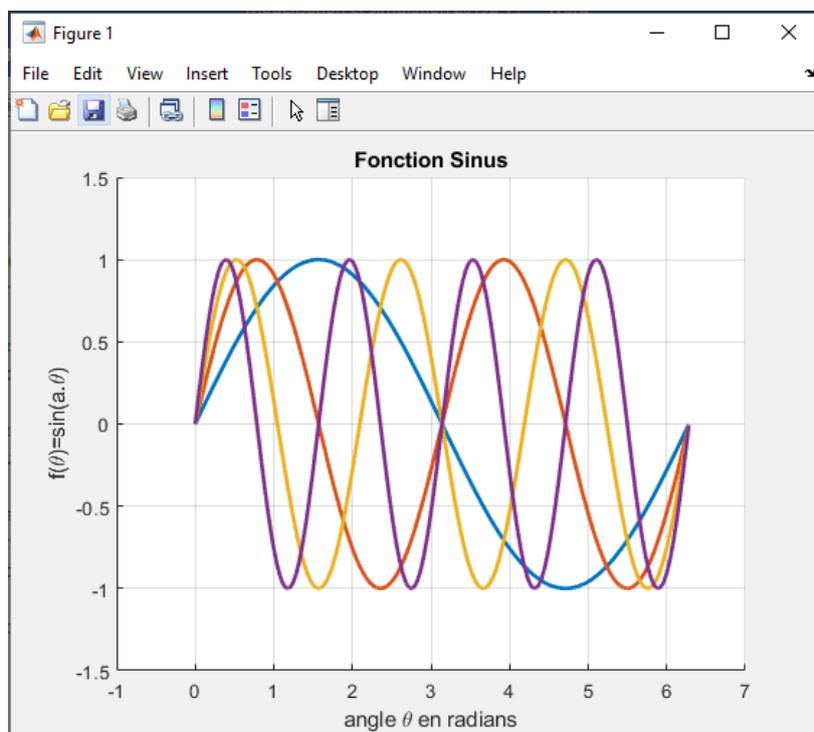


Figure 37 : le résultat obtenu après exécution du script

## 2. Simulink

**Simulink** est l'interface graphique de **MATLAB** qui permet de s'affranchir du code et de la syntaxe indispensable à la saisie des lignes de commandes. **Simulink** possède des bibliothèques de blocs, regroupés dans des Blocksets (Simulink Control Design pour le contrôle commande des systèmes par exemple). **Simulink** propose une approche causale de la modélisation. Le comportement dynamique d'un système est caractérisé par un bloc contenant, par exemple, la fonction de transfert du système avec une entrée et une sortie. L'information circulant dans les connexions entre deux blocs est un signal numérique orienté. La sortie du bloc est calculée numériquement en déterminant pour chaque pas de calcul la transformation du signal d'entrée imposé par le contenu du bloc. Simulink propose également toute une palette d'outils très évolués pour le contrôle commande des systèmes asservis.

Cette approche largement utilisée, nécessite une parfaite connaissance des lois physiques qui caractérisent le comportement des systèmes. Toute phase de modélisation commence par l'écriture des équations différentielles caractéristiques du phénomène physique étudié et l'obtention des fonctions de transfert de tous les sous-systèmes composant le système étudié.

La Figure 38 présente un **modèle Simulink** et la Figure 39 montre les résultats obtenus :

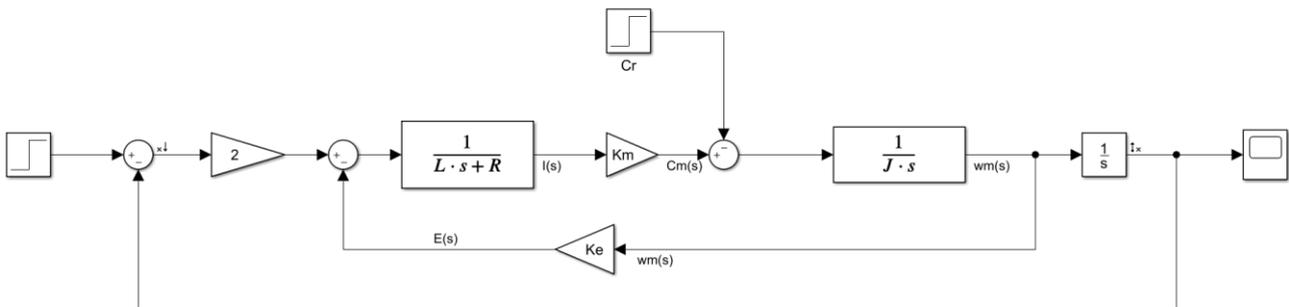


Figure 38 : exemple de modélisation Simulink

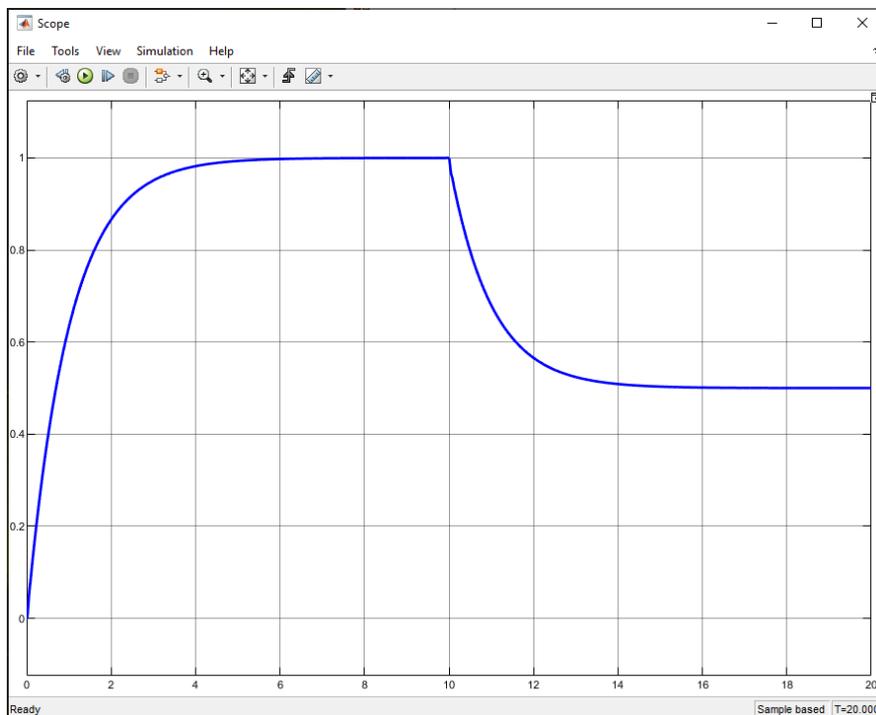


Figure 39 : la visualisation dans un scope des résultats obtenus

### 3. Simscape

**Simscape** propose une approche acausale de la modélisation et permet de faire de la modélisation par assemblage de composants. Le comportement physique des composants est pris en compte directement par le logiciel, il est donc possible de modéliser un système sans avoir à écrire l'équation différentielle qui caractérise son comportement.

Chaque domaine physique est représenté par une couleur. Les connexions entre deux composants d'un même domaine ne sont pas orientées, ont une signification physique et transmettent un niveau d'information supérieur aux connexions en modélisation causale. Ces connexions peuvent être un fil électrique (transfert d'information de type courant et tension), un arbre moteur (transfert d'information de type couple et vitesse angulaire), l'extrémité de la tige d'un vérin (transfert d'information de type force et vitesse linéaire)... Le principe de calcul s'appuie sur un bilan de puissance à chaque nœud du modèle et ne repose pas sur le principe de causalité, d'où le nom de modélisation acausale.

La Figure 40 présente un exemple de modèle réalisé avec **Simscape** et la Figure 41 montre les résultats obtenus.

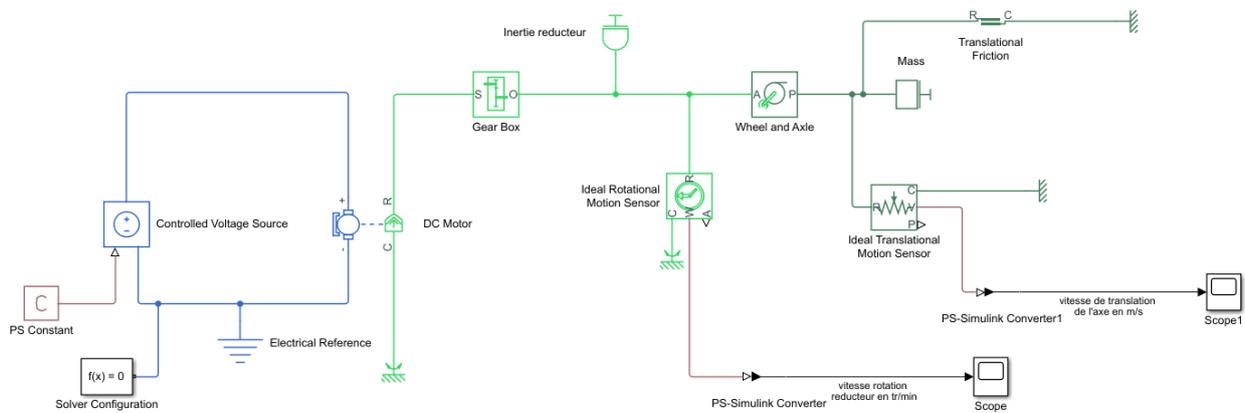


Figure 40 : exemple de modélisation Simscape

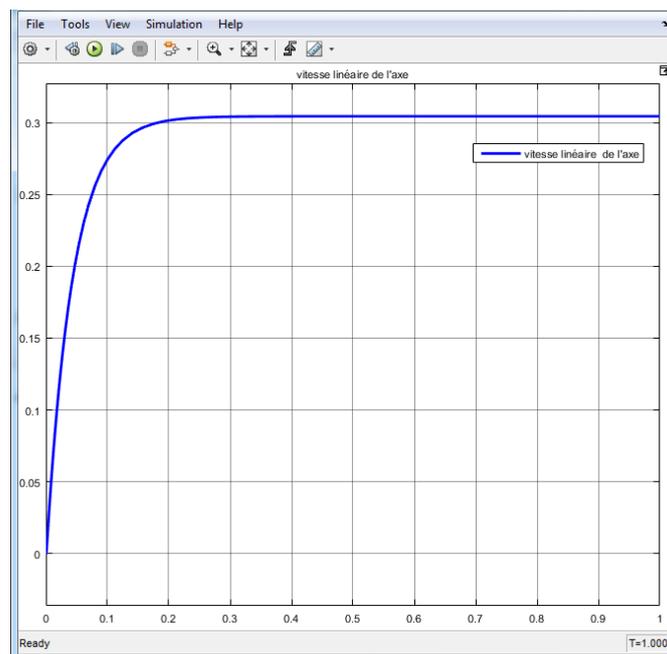


Figure 41 : la visualisation dans un scope des résultats obtenus

**Simscape** inclue une bibliothèque de composants élémentaires dans toutes les technologies (mécanique, électrique, hydraulique, pneumatique, thermique, magnétique...) et comprend également différents modules :

- **Drivelines** : composants de transmission de puissance (réducteurs, freins, embrayages, différentiels, accouplements...)
- **Electrical** : composants électriques (toutes les technologies de moteurs, alimentation, filtres, préactionneurs électriques), composants pour l'électronique et la mécatronique (servomoteurs, commande PWM, pont en H, convertisseurs, portes logiques...)
- **Fluids** : composants permettant de réaliser des circuits hydrauliques (toutes les technologies de pompes, distributeurs, vérins, clapets, soupapes...)
- **Multibody** : ce module permet d'importer des modèles CAO 3D et de les intégrer directement dans une modélisation multi-physique, un asservissement... Les propriétés cinétiques des pièces sont prises en compte et le comportement dynamique automatiquement intégré.
- 

Chacun de ces modules explore un champ technologique spécifique et propose une modélisation de niveau supérieur aux bibliothèques de base de **Simscape**.

# Simscape et ses bibliothèques

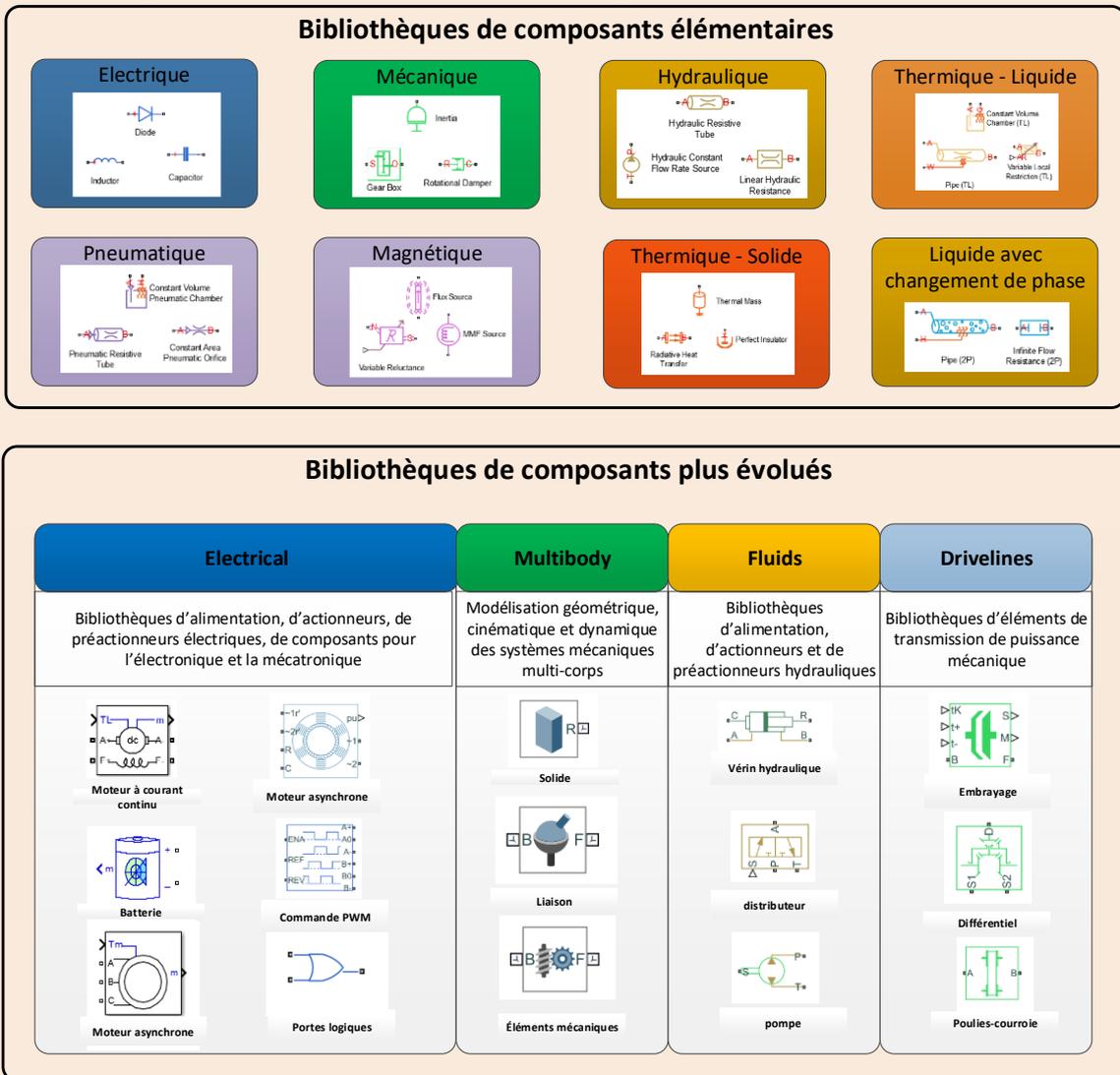


Figure 42 : Simscape et ses bibliothèques

## 4. Stateflow

**Stateflow** prend en charge la modélisation du comportement combinatoire et séquentiel des systèmes à partir d'une description sous la forme de diagrammes d'états, de flux logiques ou de tables de vérité. La Figure 43 présente un exemple de modélisation de machine à états modélisée avec **Stateflow**.

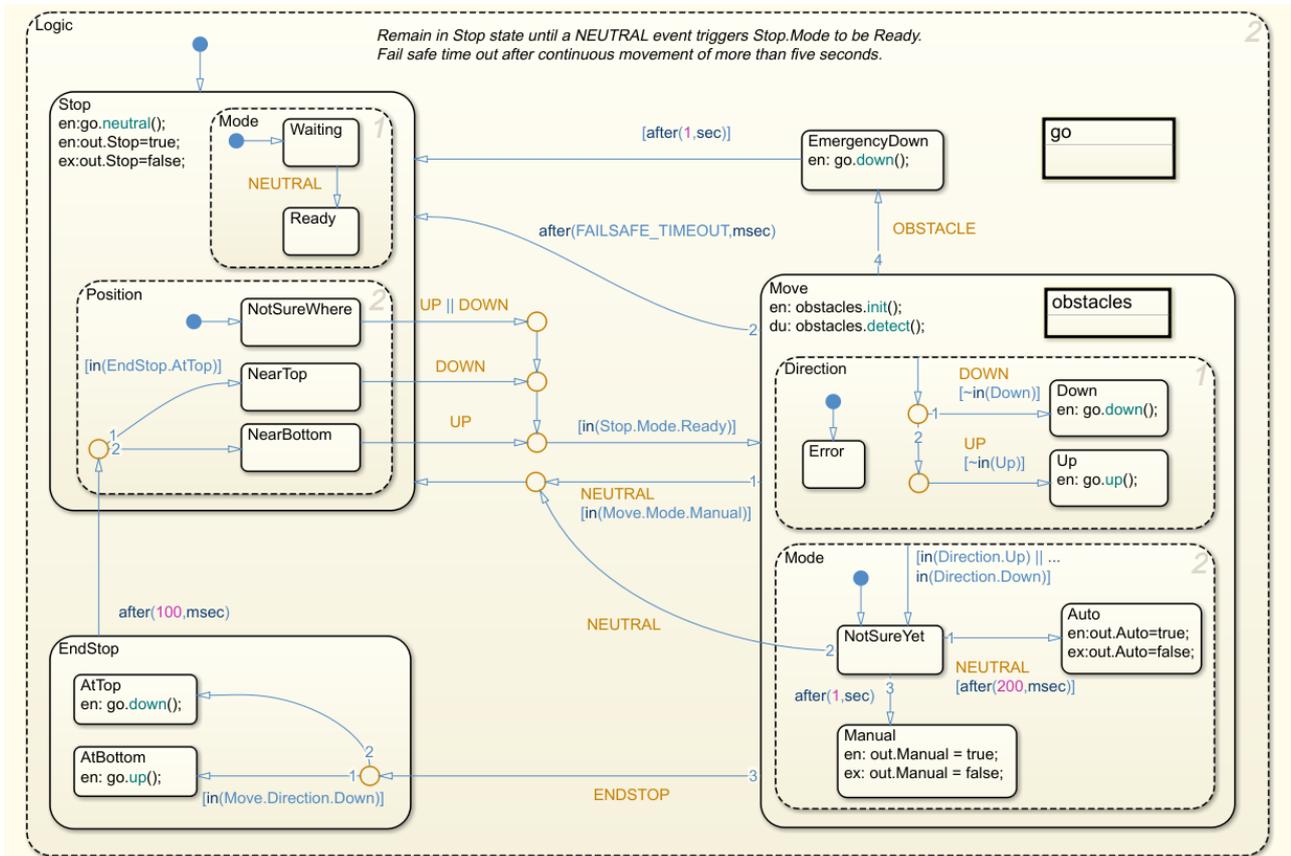


Figure 43 : modélisation du comportement d'une boîte de vitesse automatique avec Stateflow

## 5. Utilisation des outils de modélisation

Tous ces outils de modélisation communiquent entre eux et permettent de mener une démarche de modélisation globale d'un système. Il est possible dans un même modèle de combiner modélisation causale et acausale, d'intégrer une maquette CAO 3D avec **Multibody**, de prendre en compte un comportement séquentiel avec **Stateflow** tout en utilisant les outils de contrôle commande de **Simulink**. Il est possible de faire communiquer le modèle avec **MATLAB** et d'utiliser des fonctions d'exportation de résultats, de lancer une simulation à partir d'un programme **MATLAB** en faisant varier automatiquement les paramètres souhaités...

La Figure 42 et la Figure 44 décrivent l'organisation hiérarchique de la palette d'outils sélectionnée dans l'environnement **MATLAB-Simulink**.

Dans un premier temps, nous allons présenter l'environnement du logiciel afin de se familiariser avec les procédures de lancement des différents outils. Ensuite nous aborderons la prise en main des outils sur la base d'exemples qui peuvent être abordés avec des étudiants dans le cadre de la formation des ingénieurs. La performance dans la démarche de modélisation repose avant tout sur la connaissance des fondements de chaque outil. Il sera ensuite aisé pour l'utilisateur d'approfondir l'exploitation de ces outils en fonction des besoins de la démarche de modélisation menée.

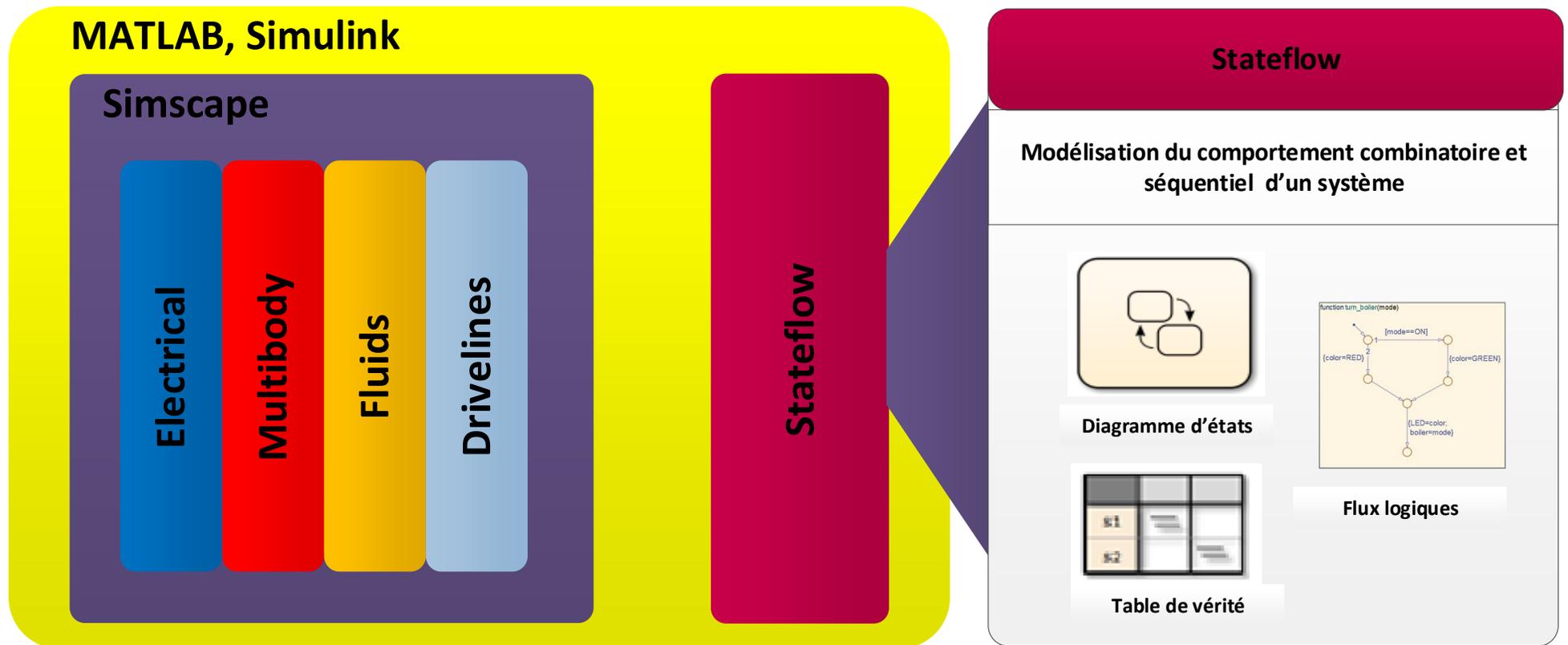


Figure 44 : les outils MATLAB pour la modélisation multi-physique

## II. Présentation de l'environnement MATLAB – Simulink

### A. Lancement du logiciel

A partir de l'icône  situé sur le bureau, lancer le logiciel MATLAB.

### B. La fenêtre de l'environnement MATLAB

A l'ouverture de MATLAB, plusieurs fenêtres sont accessibles.

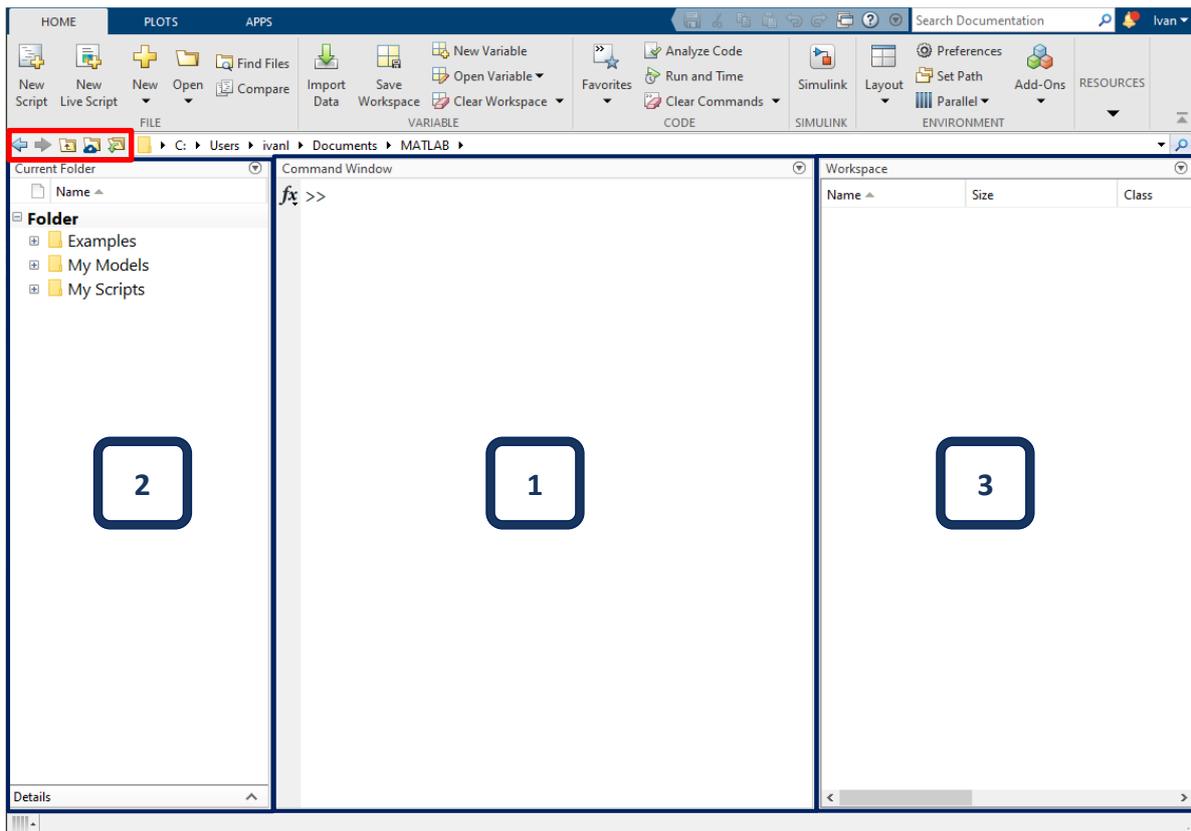


Figure 45 : la fenêtre de l'environnement MATLAB

#### 1:Command Windows

Fenêtre de commande MATLAB. C'est la fenêtre avec laquelle, l'utilisateur dialogue avec MATLAB (saisie de ligne de commande, d'instructions...)

#### 2:Current Folder

Cette fenêtre indique le dossier courant dans lequel travaille le logiciel

La barre d'outils  située dans la partie supérieure de cette fenêtre permet de naviguer dans les dossiers.

#### 3:Workspace

Cette fenêtre indique le nom et le format de toutes les variables créées et utilisées lors de la session courante. Il est possible de sauvegarder le Workspace dans un fichier.

## 1. La barre de commande MATLAB

La Figure 46 présente les principales commandes de la barre de commandes MATLAB.

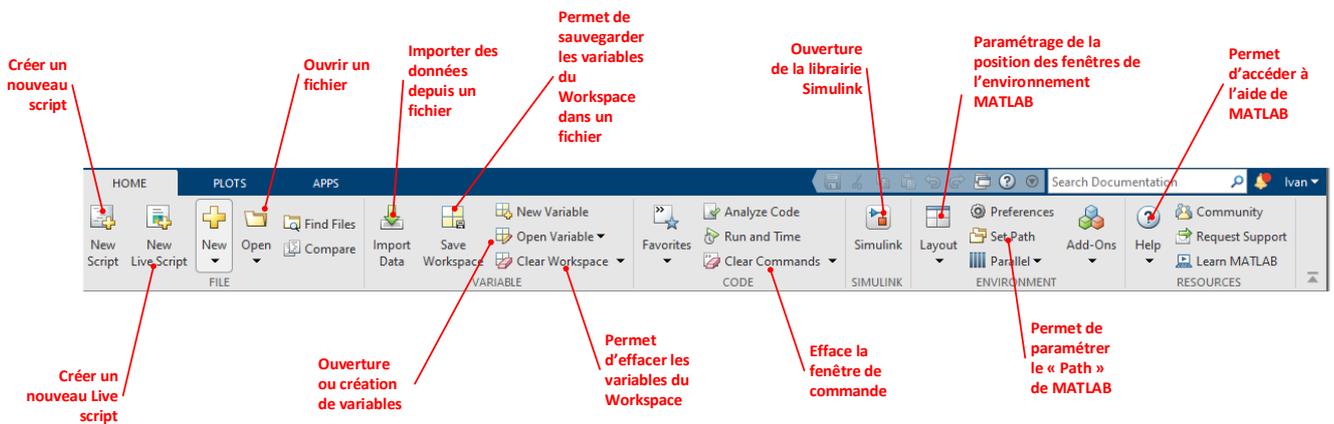


Figure 46 : la barre de commande de l'environnement MATLAB

La barre de commande possède également deux onglets complémentaires :

- L'onglet « PLOT » qui permet de tracer des courbes à partir de variables stockées dans le Workspace sans avoir à utiliser du code.



- L'onglet « APPS » qui permet d'accéder à des applications en fonction des besoins de l'utilisateur (interpolation d'une série de données, réglage d'un correcteur...)

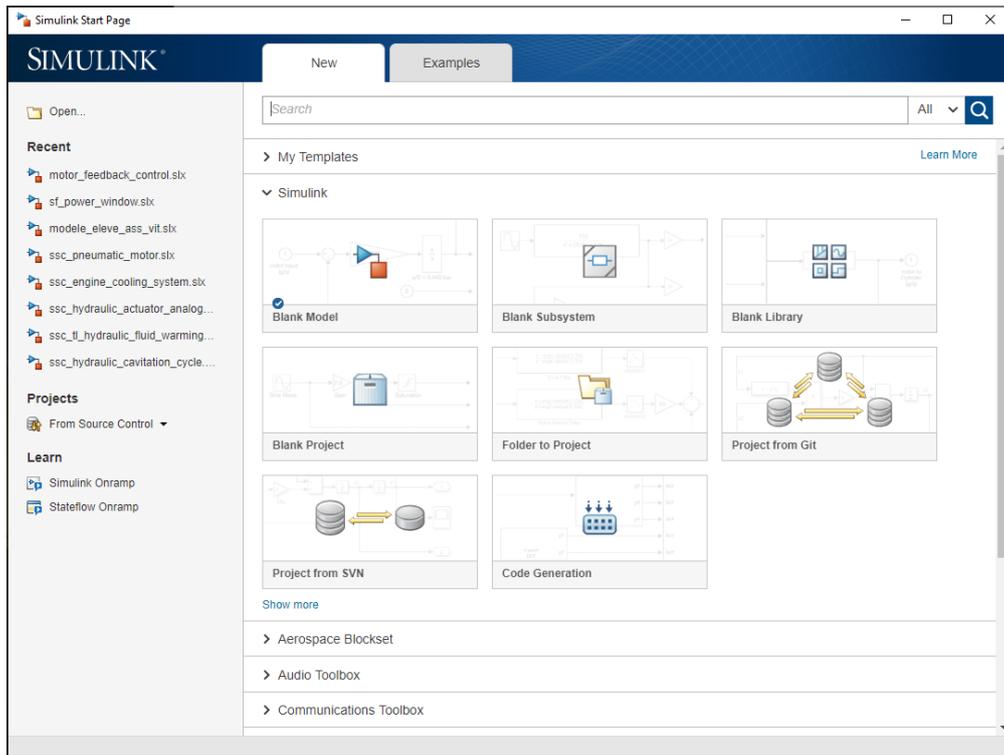


## C. La fenêtre de l'environnement Simulink.

Pour lancer Simulink cliquer sur l'icône  
permet d'ouvrir un fichier Simulink.



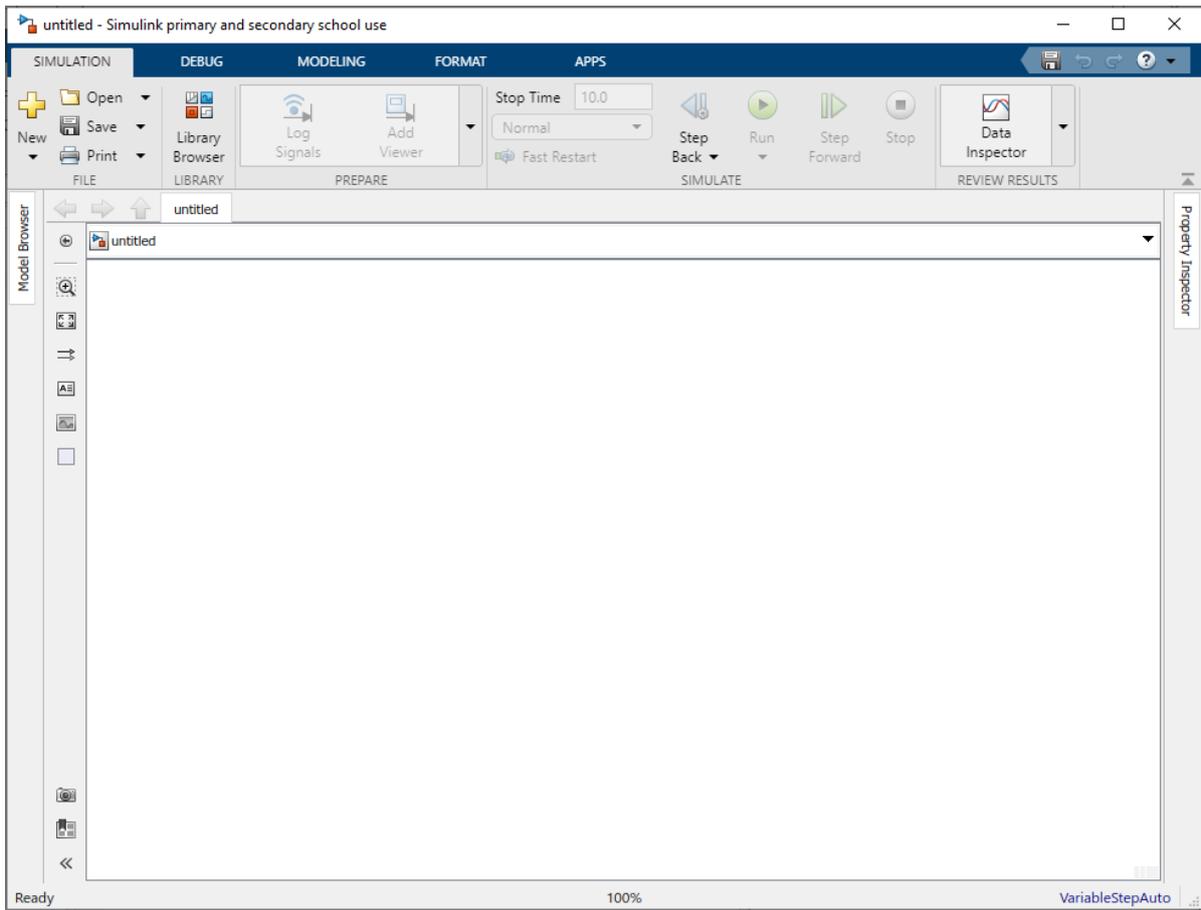
afin d'ouvrir la Simulink Start Page (Figure 47) qui



**Figure 47 : Simulink Start Page**

A partir de cette page, il est possible d'ouvrir un nouveau modèle. Afin d'accélérer le processus de conception Simulink propose différents modèles de fichiers en fonction des besoins spécifiques de l'utilisateur et des outils de modélisation qui seront utilisés. Ces modèles proposent donc des environnements de modélisation adaptés en termes de choix de solveur, de mise en place par défaut des blocs les plus utiles, de choix du format d'exportation des variables de la modélisation... Dans un premier temps nous allons découvrir l'environnement de modélisation et nous partirons d'un modèle vierge « Blank Model » afin de paramétrer manuellement les fonctionnalités qui nous intéressent au fur et à mesure. Plus tard avec de l'expérience, il sera plus efficace de partir d'un modèle adapté à l'objectif de la modélisation.

A partir de la Simulink Start Page sélectionner un « Blank Model » (modèle vierge).



**Figure 48 : Blank Model Simulink**

Ouvrir la librairie Simulink en cliquant sur



La librairie de Simulink s'ouvre (Figure 49) et propose toute une palette de composants pour la modélisation causale ou acausale.

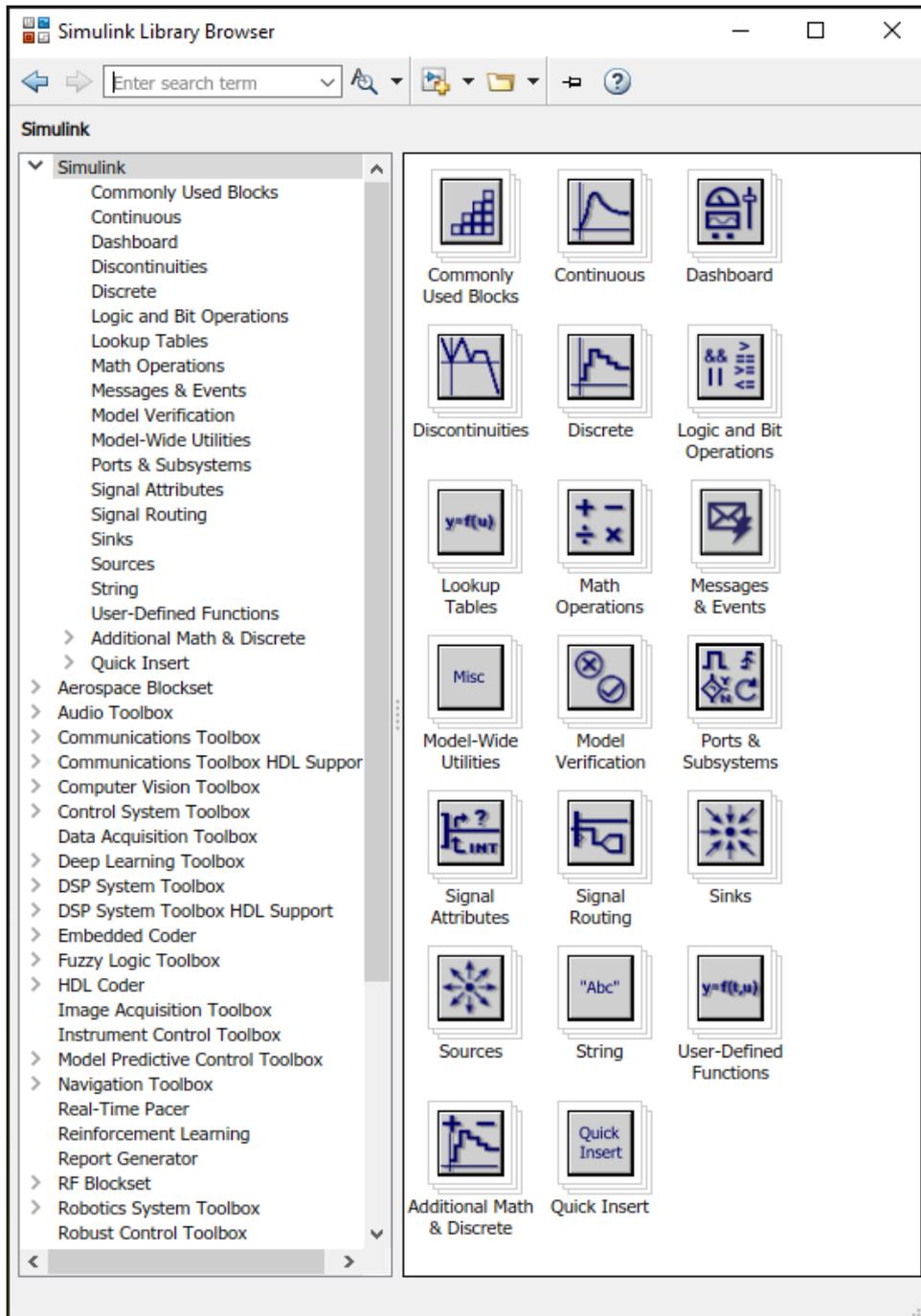


Figure 49 : la bibliothèque de Simulink

La Figure 50 décrit les fonctionnalités de la fenêtre de l'environnement Simulink.

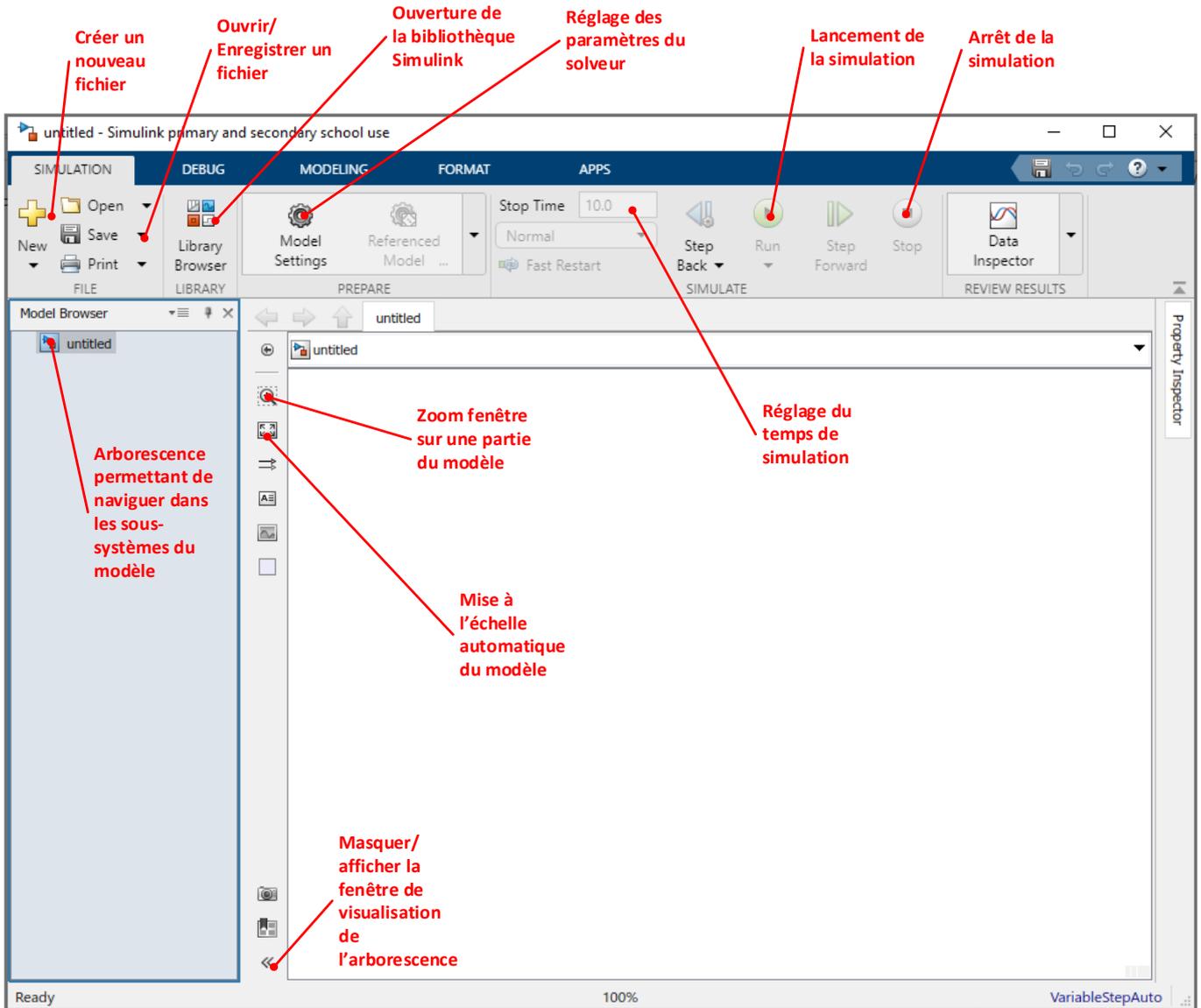


Figure 50 : la fenêtre de l'environnement Simulink

Le placement des composants se fera par une simple action de glisser-déposer des composants depuis la bibliothèque **Simulink** vers l'intérieur de cette fenêtre.

## D. Configuration de MATLAB – Simulink

### 1. Nommer un fichier dans MATLAB/Simulink

Vous pouvez donner le nom de votre choix à un fichier dans MATLAB/Simulink en respectant cependant ces règles :

- Ne pas utiliser d'espace
- Ne pas utiliser de caractères accentués

### 2. Le « path » de MATLAB

Le « path » de **MATLAB** est constitué de l'ensemble des dossiers que **MATLAB** peut lire lors d'une session courante. Tout fichier appartenant à l'un des dossiers du « path » de **MATLAB** peut être accessible sans en indiquer le chemin. Il convient donc d'inclure dans le « path » tous les dossiers qui contiennent nos fichiers de travail. Cette stratégie d'accès aux fichiers permet de ne pas avoir à spécifier le chemin d'accès aux fonctions et aux fichiers. Il suffit que ces derniers se trouvent dans le « path » spécifié par l'utilisateur pour qu'ils soient automatiquement retrouvés. L'utilisateur tapera le nom d'un script ou fera appel à une fonction dans la fenêtre de commande pour que l'exécution soit immédiate. Cela implique par exemple que deux scripts ou deux fonctions ne doivent pas porter le même nom. Il est donc important de maîtriser la structure du « path » de **MATLAB** et de n'inclure que les dossiers utiles à la session de travail. Il existe deux procédures pour ajouter des dossiers au « Path ». L'une ajoutera les dossiers sélectionnés pour la session de travail courante et pour toutes les sessions futures. L'autre ajoutera les dossiers sélectionnés uniquement pour la session de travail en cours. Il est important de bien comprendre cette différence et de n'inclure définitivement dans le « path » que les dossiers vraiment utiles pour toutes les sessions afin d'éviter les conflits éventuels entre des fichiers du même nom.

### 3. Ajout de dossiers dans le « path » pour toutes les sessions

Pour ajouter des dossiers au « path » de **MATLAB** pour la session courante et pour toutes les sessions futures,

cliquer sur  dans la barre de commande de **MATLAB** pour ouvrir la fenêtre de définition du « path » (Figure 51).

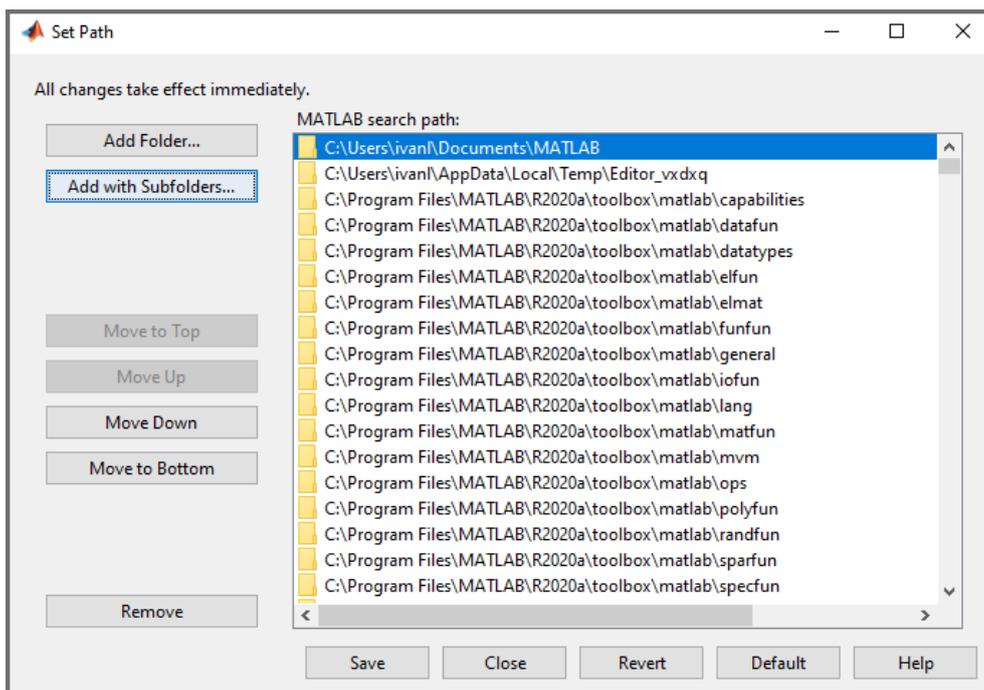


Figure 51 : le path de MATLAB

Sélectionner **Add with Subfolders**, puis choisir les dossiers que vous voulez ajouter dans le « path » de MATLAB (les sous-dossiers seront automatiquement ajoutés) et tous les dossiers et sous-dossiers ajoutés par cette procédure seront dans le « Path » pour toute les sessions futures.

Sélectionner **Save**, puis **Close**.

A la suite de cette opération, le « Path » est sauvegardé et sera le même lors de l'ouverture des prochaines sessions.

#### 4. Ajout de dossiers dans le « path » pour la session courante

Il est également possible de modifier le « Path » pour la session en cours uniquement. Pour cela, cliquer avec le bouton droit de la souris sur le dossier que vous voulez ajouter au « Path » dans la fenêtre « Current Folder » (Figure 52). pour faire apparaître le menu contextuel. Sélectionner ensuite **Add to Path** puis **Selected Folders and Subfolders**.

Il est également possible d'utiliser cette procédure pour supprimer des dossiers du « Path » en sélectionnant **Remove from Path** puis **Selected Folders and Subfolders**.

Afin de pouvoir utiliser l'ensemble des fichiers de simulation utilisés dans l'ouvrage, ajouter le dossier contenant tous les modèles du livre dans le « Path » de MATLAB.

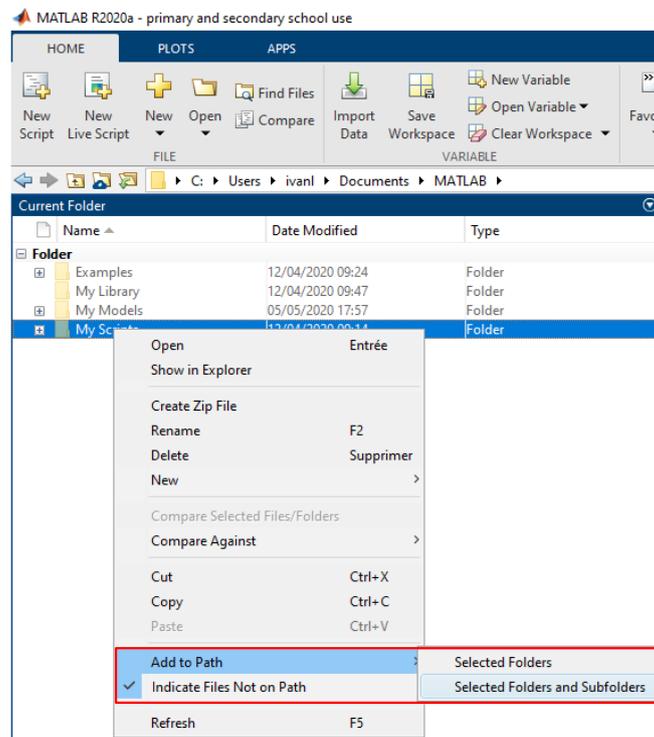


Figure 52 : ajout de dossier au « Path » pour la session courante

Les dossiers qui n'appartiennent pas au « path » apparaissent éclaircis dans la fenêtre **Current Folder** de l'environnement **MATLAB**.

# Chapitre 3 : Stratégie de conception d'un modèle multi-physique

## I. Stratégie de conception d'un modèle multi-physique

### A. Lien avec le diagramme Chaîne d'énergie/Chaîne d'information

Le descripteur chaîne d'énergie/chaîne d'information est particulièrement adapté à une approche de modélisation multi-physique. Il permet de mettre en évidence la structure de la transformation de l'énergie dans le système et ses interactions avec le dispositif de commande. De plus il est possible de réaliser une analogie complète entre ce descripteur et le modèle réalisé avec **MATLAB-Simulink**.

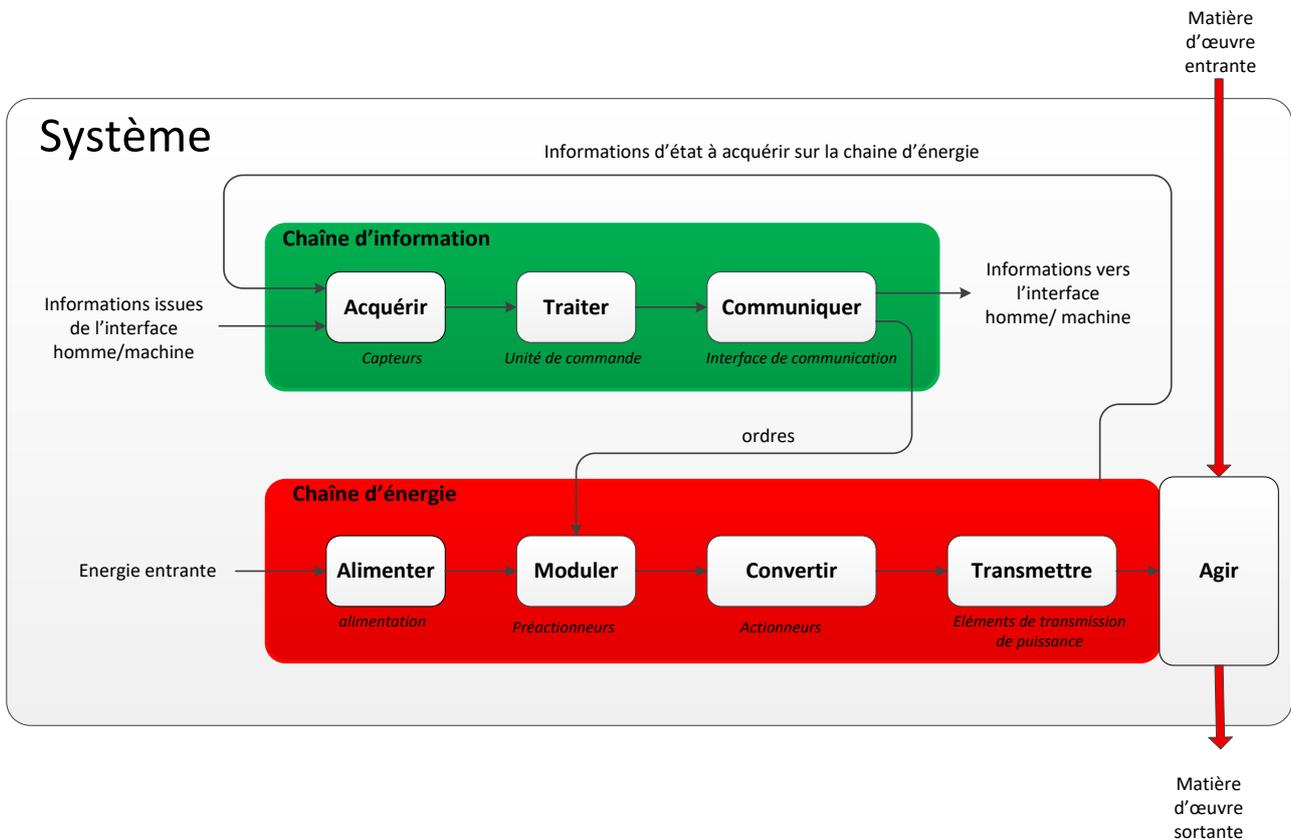


Figure 53 : le diagramme Chaîne d'énergie / Chaîne d'information

Pour mettre en place une stratégie de modélisation multi-physique, il est possible de s'appuyer sur ce descripteur en reliant les parties de ce diagramme à l'outil de modélisation à privilégier. Il s'agit de dégager une tendance, chaque problématique de modélisation est unique et de nombreuses stratégies sont envisageables.

Il aurait été également possible d'utiliser un descripteur issu du langage **SysML** comme le diagramme de blocs internes qui permet d'avoir le même type d'approche et de niveau de description que le diagramme chaîne d'énergie / chaîne d'information. La toolbox **System Composer** permet de construire un diagramme de blocs internes et d'y inclure les modèles.

Le diagramme de la Figure 54 illustre cette démarche.

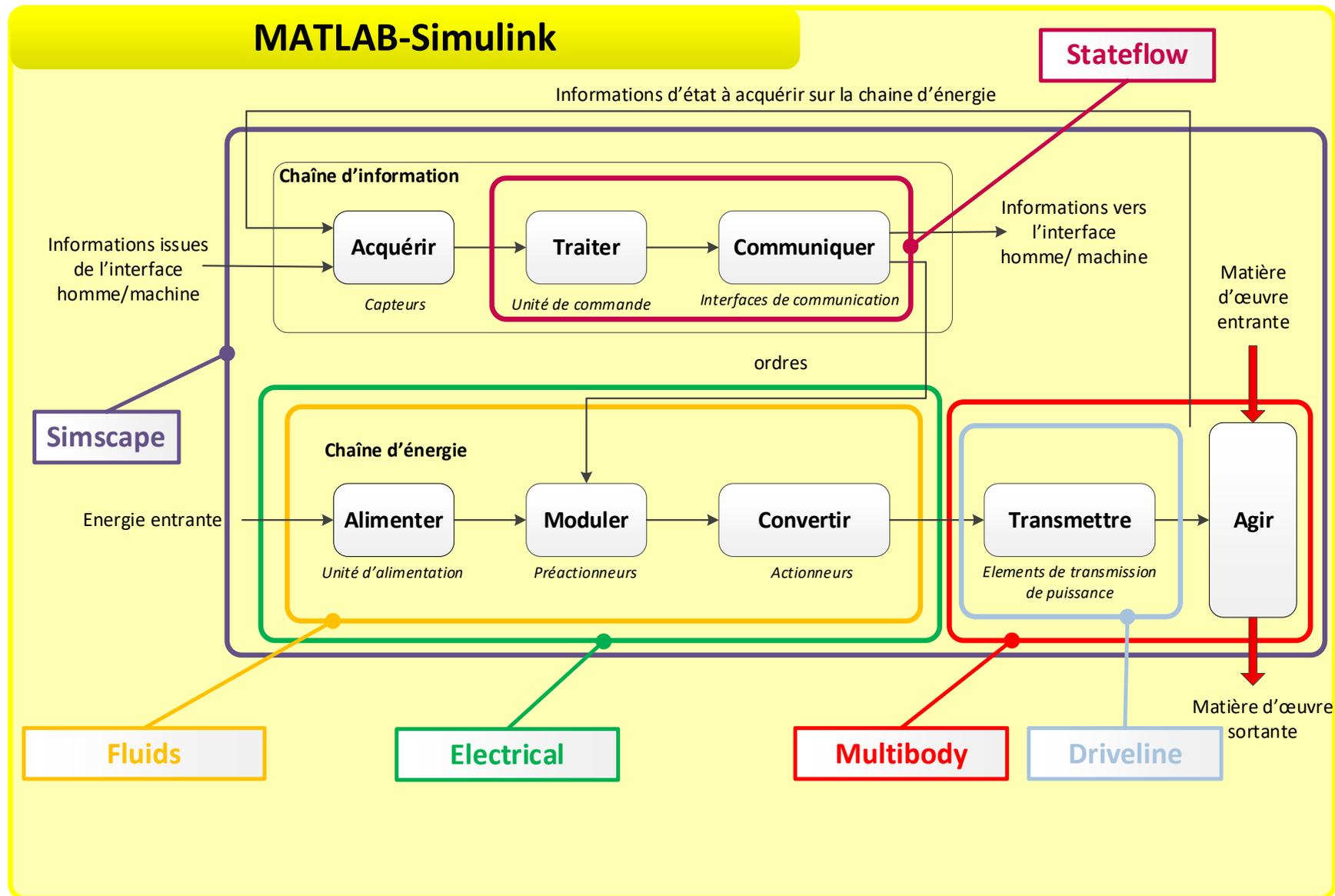


Figure 54 : correspondance entre le diagramme chaîne d'énergie / Chaîne d'information et les outils MATLAB

## II. Application à un pilote hydraulique de bateau

Afin d'illustrer cette approche, nous allons appliquer la démarche proposée à un pilote hydraulique automatique de bateau. Le système fonctionne de la manière suivante (Figure 55).

Un moteur à courant continu actionne une pompe hydraulique à double sens de flux qui débite dans un vérin par l'intermédiaire d'un circuit de distribution hydraulique. La translation du vérin entraîne la rotation du bras de mèche sur lequel est fixée la barre du bateau. La rotation de la barre entraîne la modification du cap du bateau. La chaîne d'information assure la commande du moteur en garantissant que le cap réel du bateau suive la consigne de cap.

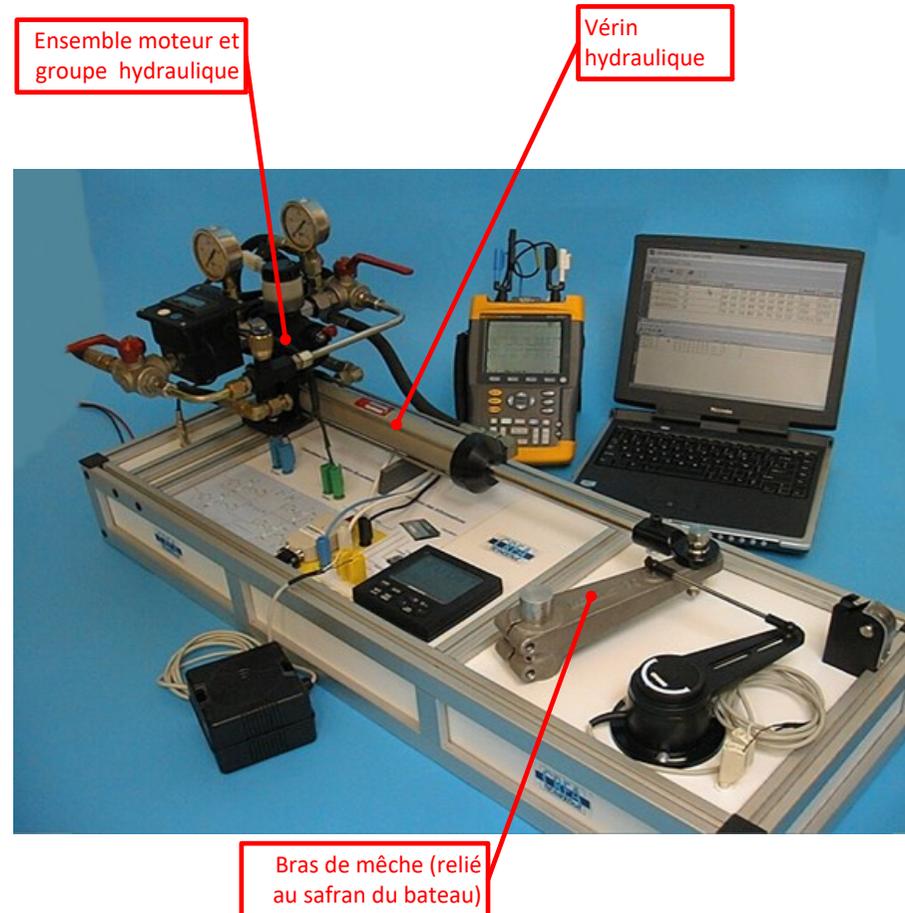


Figure 55 : photo du pilote automatique de bateau

## A. Diagramme présentant la chaîne d'énergie et d'information du pilote hydraulique de bateau

La Figure 56 présente le diagramme chaîne d'énergie/chaîne d'information du pilote de bateau et les outils choisis pour modéliser les différentes parties du système.

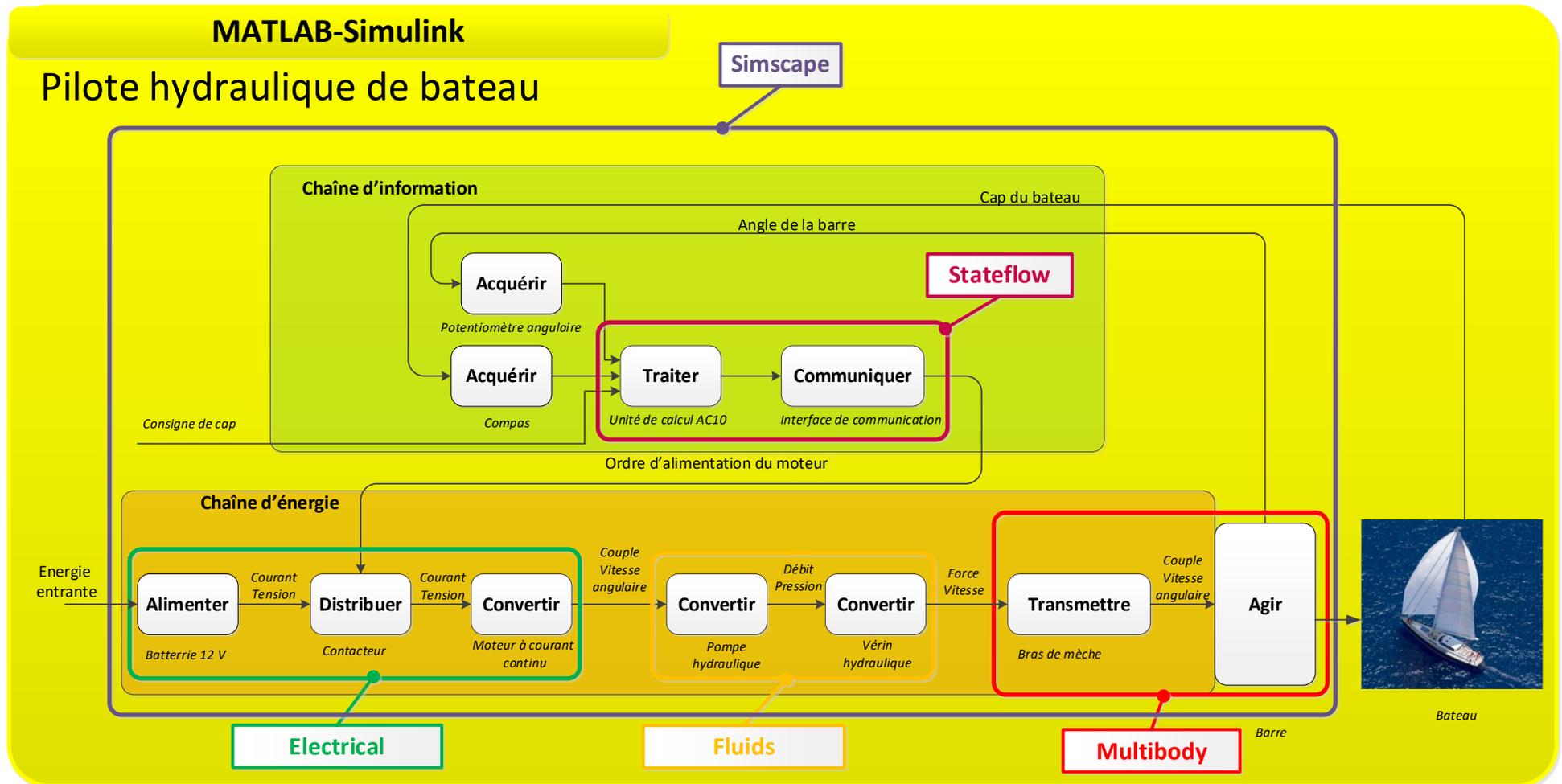


Figure 56 : correspondance entre le diagramme chaîne d'énergie / Chaîne d'information et les outils MATLAB

## B. Modèle multi-physique du pilote hydraulique de bateau réalisé avec MATLAB - Simulink

Le modèle multi-physique est structuré en suivant exactement le même formalisme que le descripteur. Les chaînes d'information et d'énergie apparaissent sous la forme de sous-systèmes. Les échanges d'énergie et d'information entre les blocs du descripteur sont les mêmes que les connexions entre les sous-systèmes du modèle multi-physique.

En double cliquant sur un sous-système, il est possible d'accéder à la modélisation. Le modèle multi-physique devient également un outil d'aide à la description qui permet d'explorer de manière dynamique les composants qui réalisent les fonctions.

La Figure 57 présente une copie d'écran du modèle multi-physique réalisé avec MATLAB – Simulink.

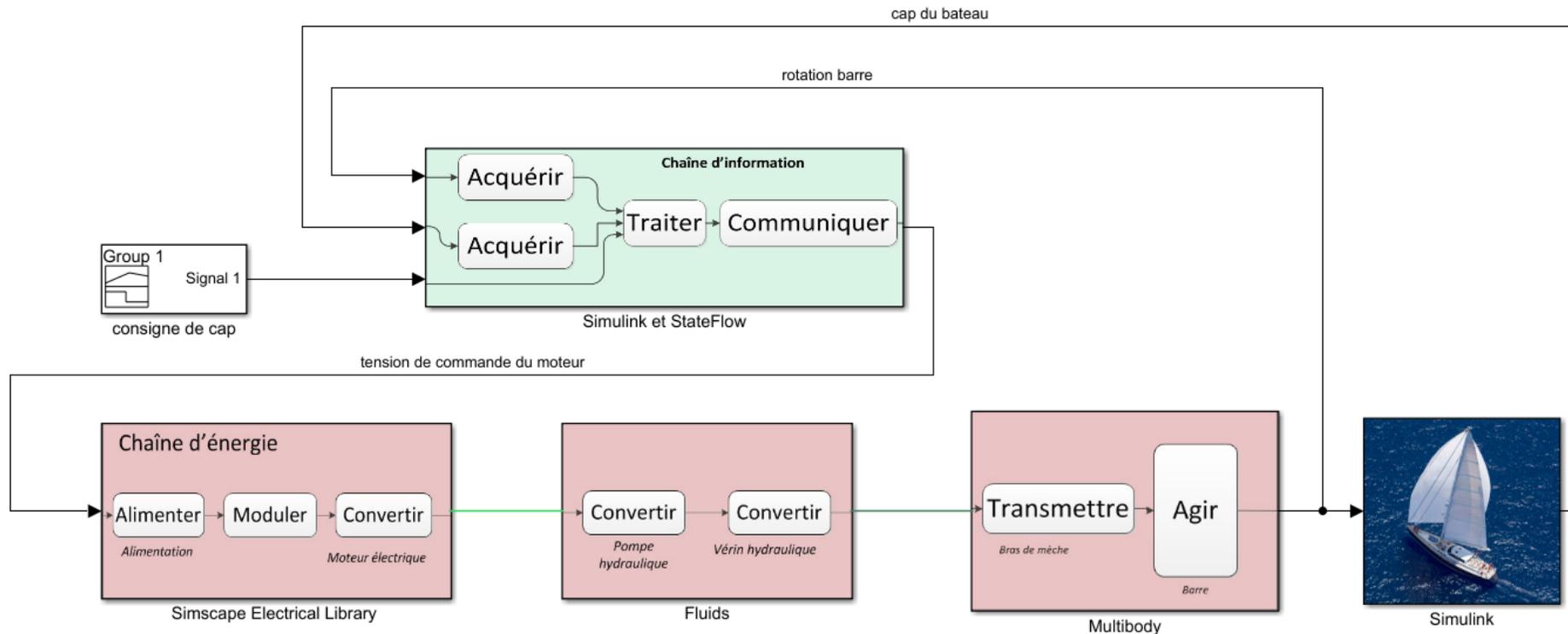


Figure 57 : copie d'écran du modèle multi-physique du pilote automatique de bateau réalisé avec MATLAB

## C. Chargement et simulation du modèle

Afin de pouvoir utiliser l'ensemble des fichiers de simulation utilisés dans l'ouvrage, le dossier contenant tous les modèles du livre doit impérativement être inclus dans le « Path » de MATLAB (procédure décrite page 55).

## D. Visualisation des résultats issus du modèle multi-physique

**Ouvrir** le fichier **pilote\_hydraulique.slx**. (dossier *Chapitre\_1\_Introduction/Pilote\_hydraulique*)

**Double-cliquez** sur le bloc « consigne de cap » pour visualiser le signal de consigne.

La consigne de cap impose au bateau un cap de  $20^\circ$  pendant 300 s puis un cap de  $-40^\circ$  pendant 700 s.

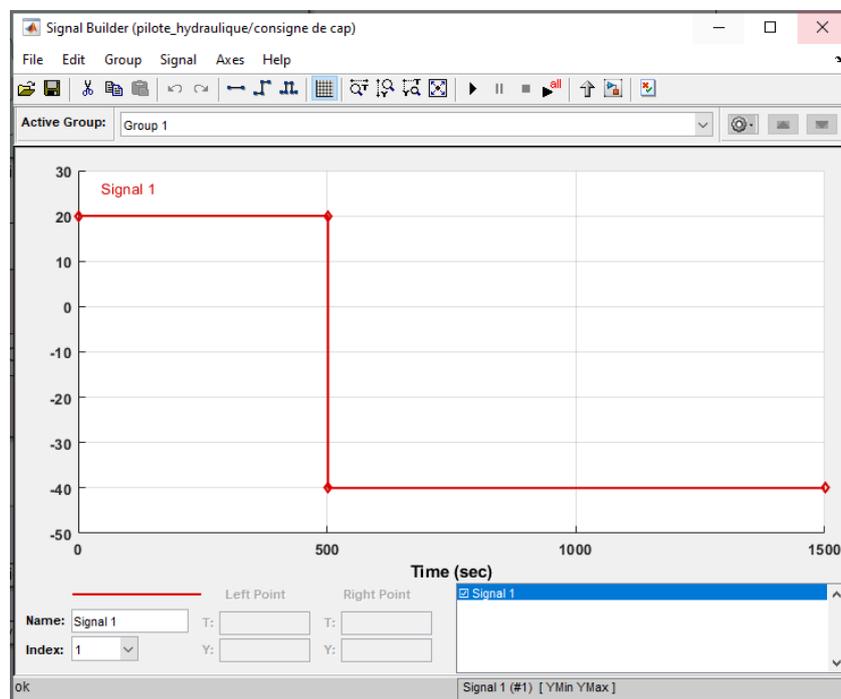


Figure 58 : visualisation de la consigne de cap



Lancer la simulation en cliquant sur .

Lors du lancement de la simulation une fenêtre **Mechanics Explorers** s'ouvre permettant de visualiser dynamiquement les mouvements de la partie mécanique modélisée à l'aide de **Multibody** (Figure 59).

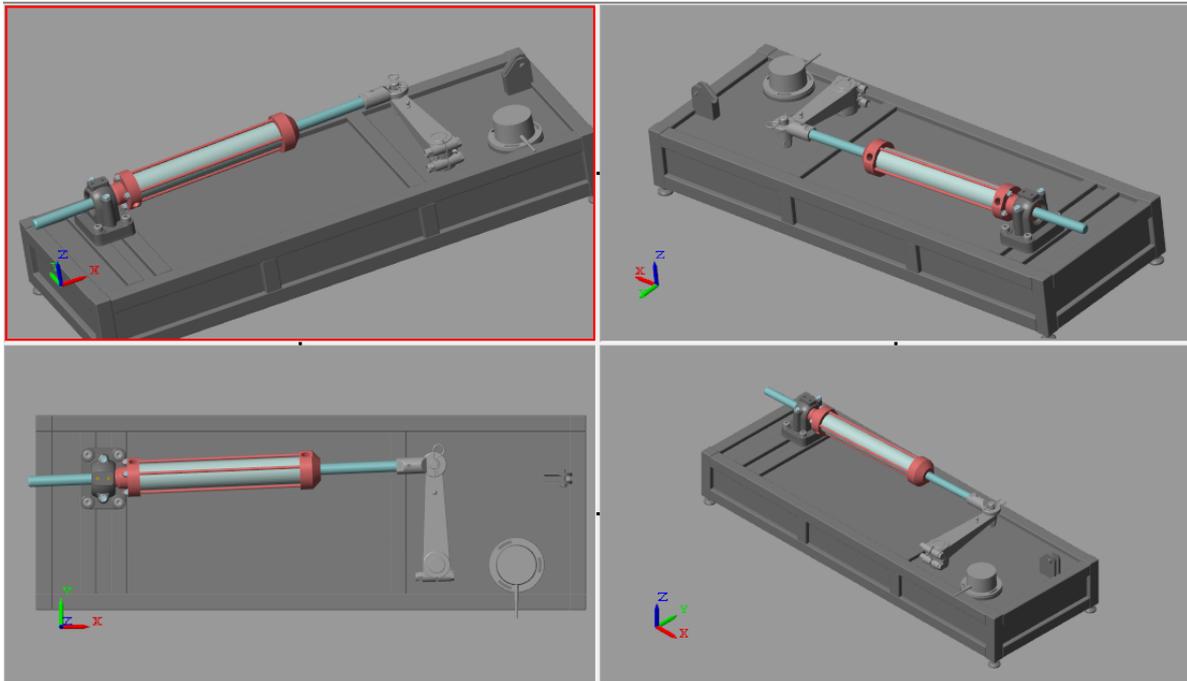


Figure 59 : fenêtre de visualisation de Multibody (Mechanics Explorer)

Une barre de visualisation permet de choisir le mode d'observation du modèle.

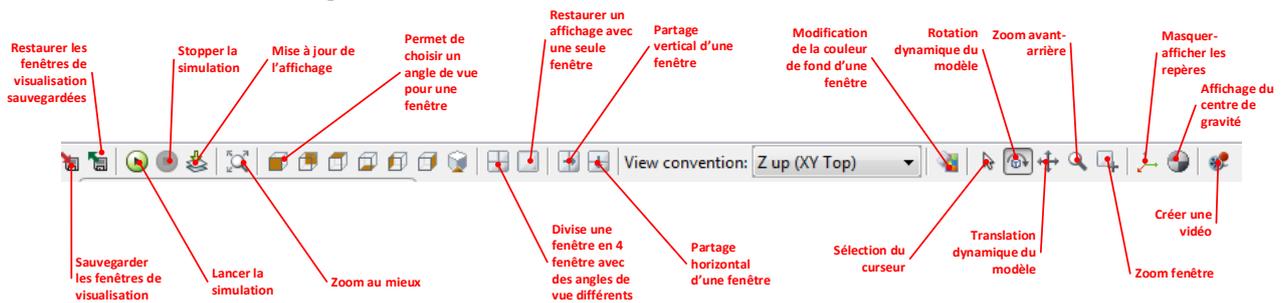


Figure 60 : barre de visualisation de Multibody

Cliquez sur l'icône  dans la partie supérieure de la fenêtre **Mechanics Explorers** (Figure 60) pour revenir à un affichage avec une seule fenêtre (Figure 61).

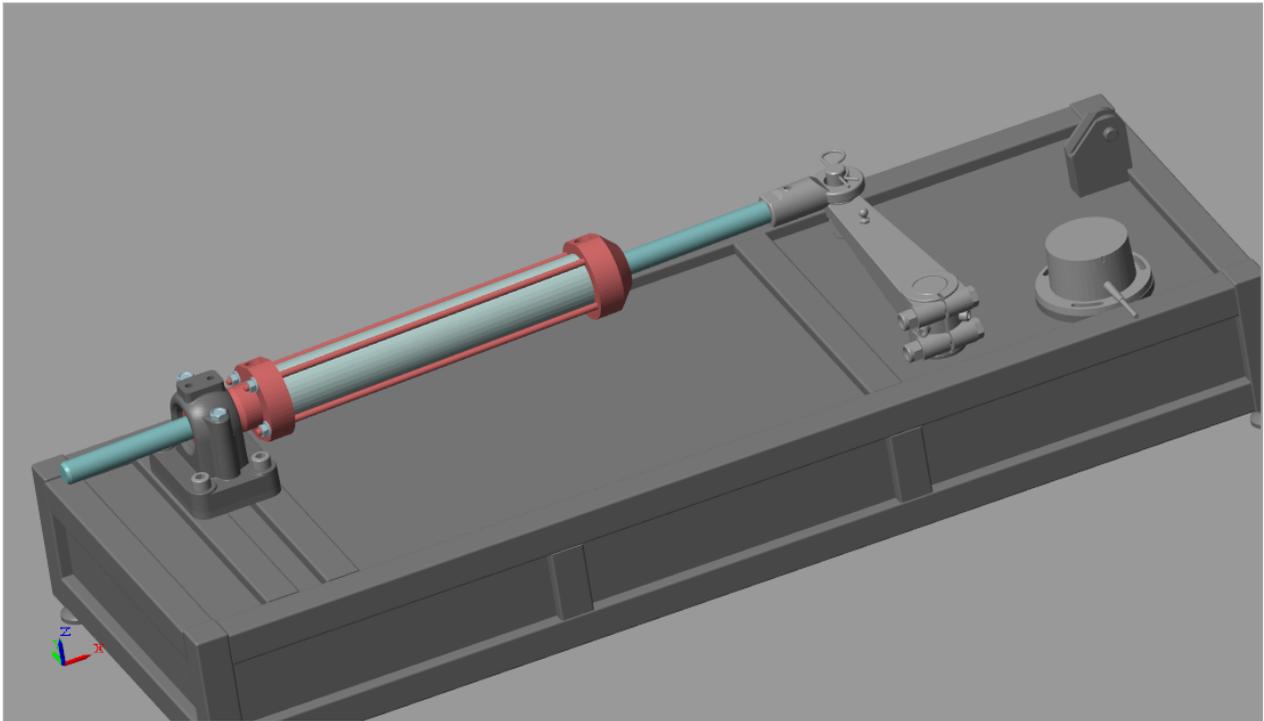


Figure 61 : fenêtre de visualisation de Multibody (Mechanics Explorer)

Dans la partie inférieure de la fenêtre une barre temporelle permet de visualiser l'évolution temporelle de la simulation. Il est possible d'accélérer la visualisation à l'aide d'un curseur ou de revenir à un instant quelconque de la simulation (Figure 62).

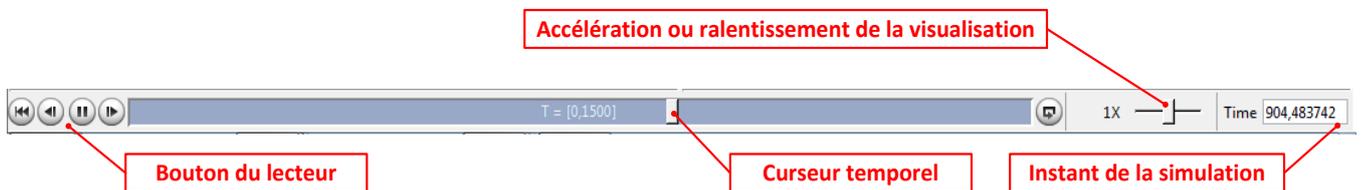


Figure 62 : barre temporelle de Multibody

A la fin de la simulation, les résultats sont facilement exploitables en double-cliquant sur les scopes. Il est possible de relever toutes les grandeurs physiques du modèle.

On peut visualiser la consigne de cap et le cap effectif suivi par le bateau :

**Cliquer** sur la mise à l'échelle automatique pour visualiser toute la courbe



Il faudra prendre l'habitude d'utiliser la commande de mise à l'échelle automatique à l'ouverture de chaque scope pour bien visualiser le signal.

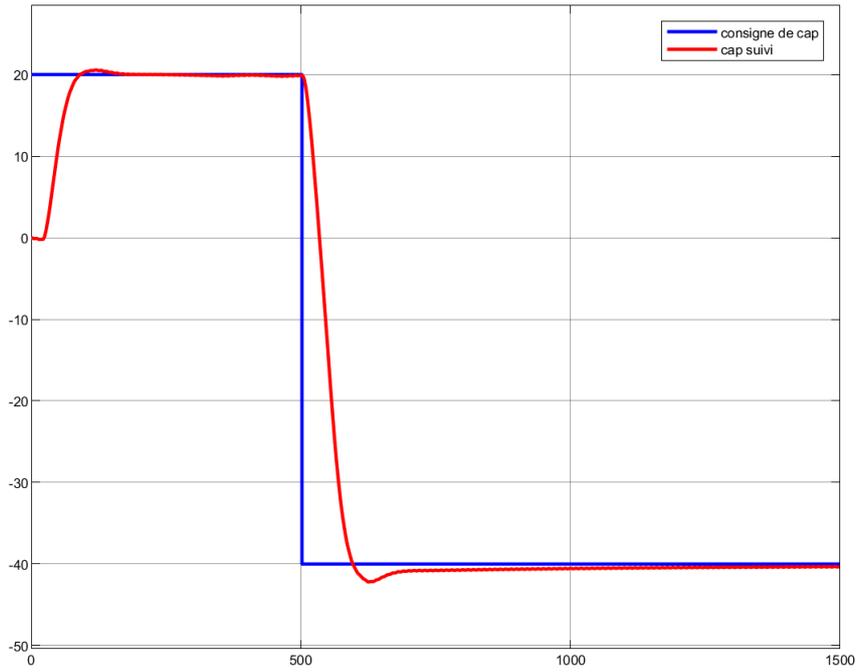


Figure 63 : visualisation de la consigne de cap et du cap effectif suivi par le bateau

Le courant, la tension et le couple du moteur :

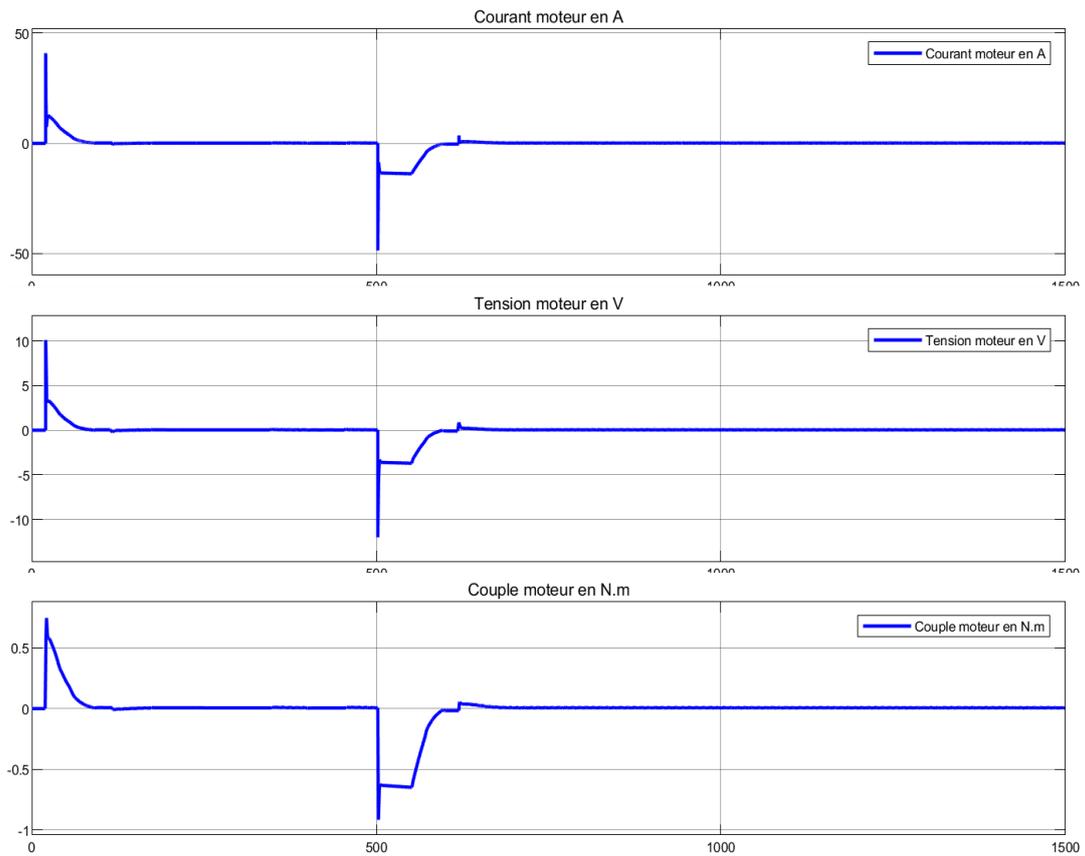


Figure 64 : visualisation des grandeurs relatives au moteur

L'angle de rotation de la barre :

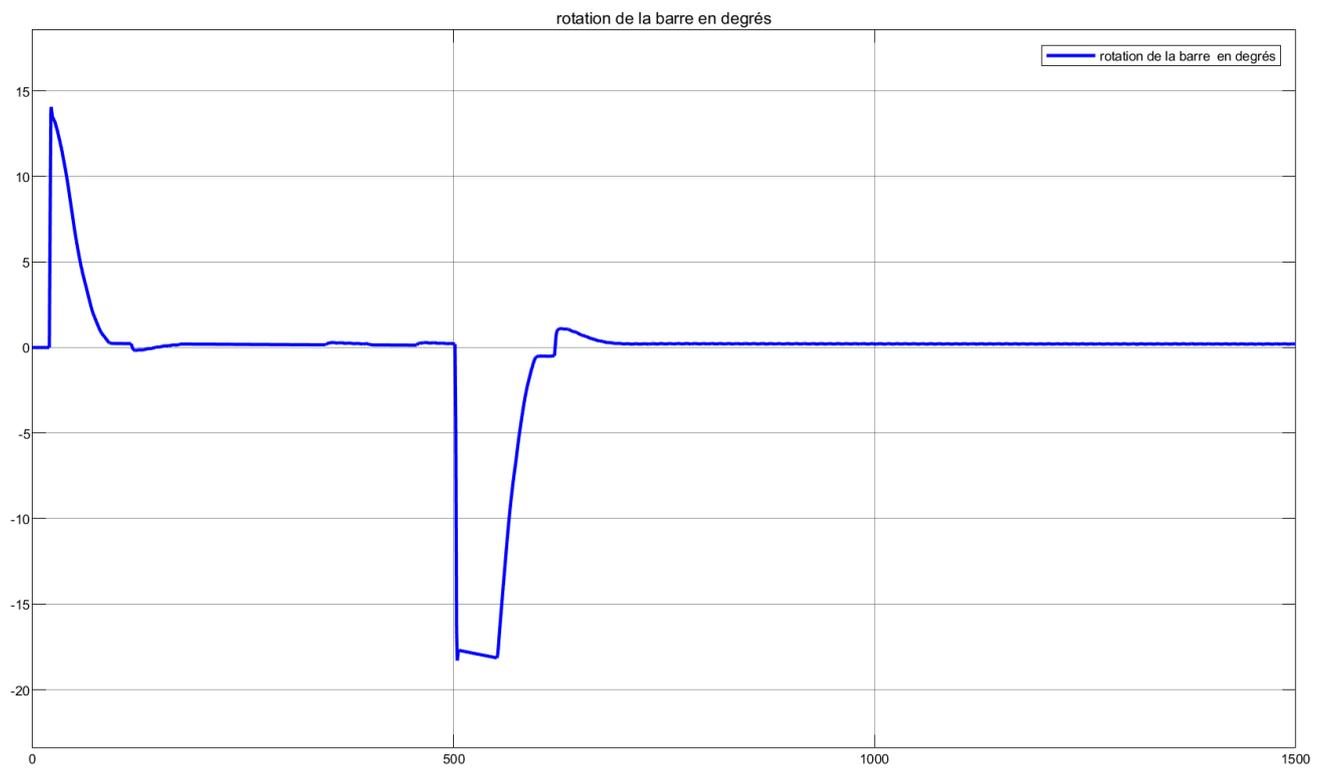


Figure 65 : visualisation de l'angle de rotation de la barre

Les pressions dans les chambres avant et arrière du vérin :

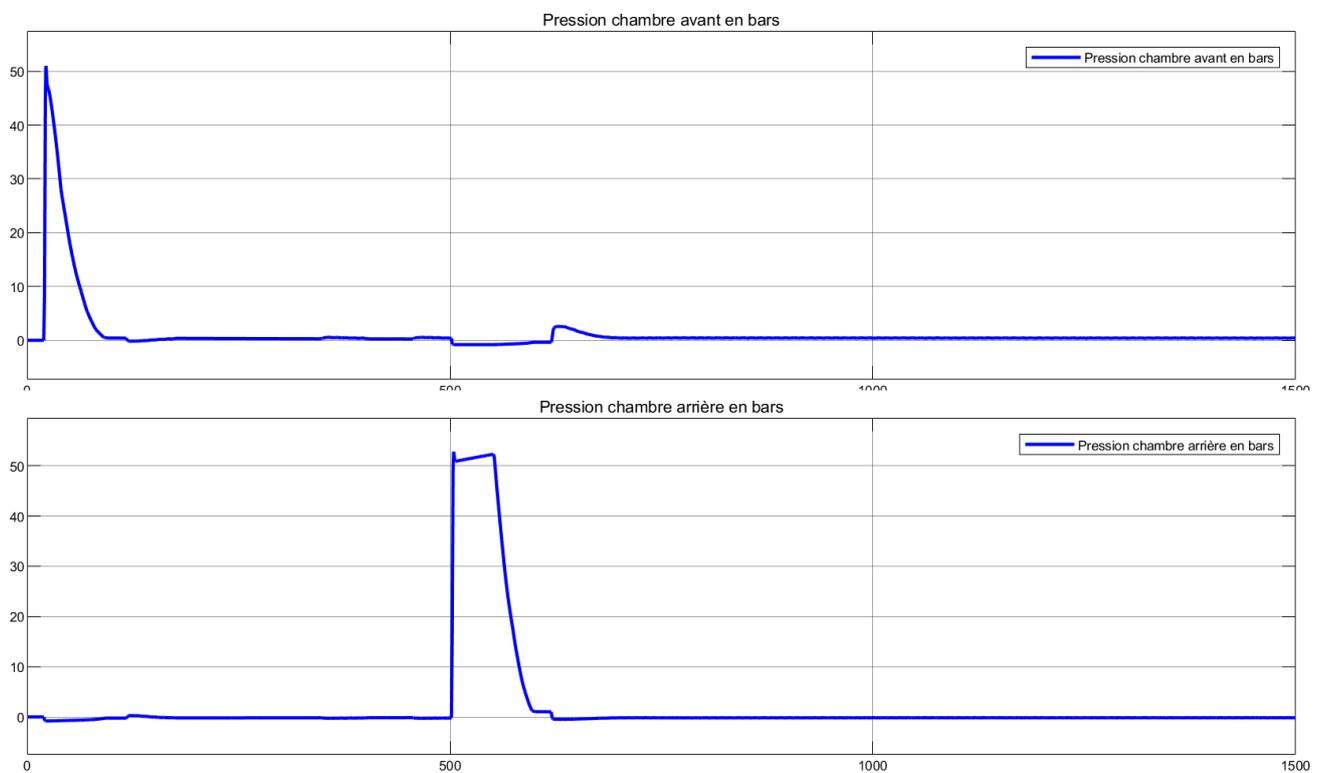


Figure 66 : visualisation des pressions dans les chambres avant et arrière du vérin

Les débits dans les chambres avant et arrière du vérin :

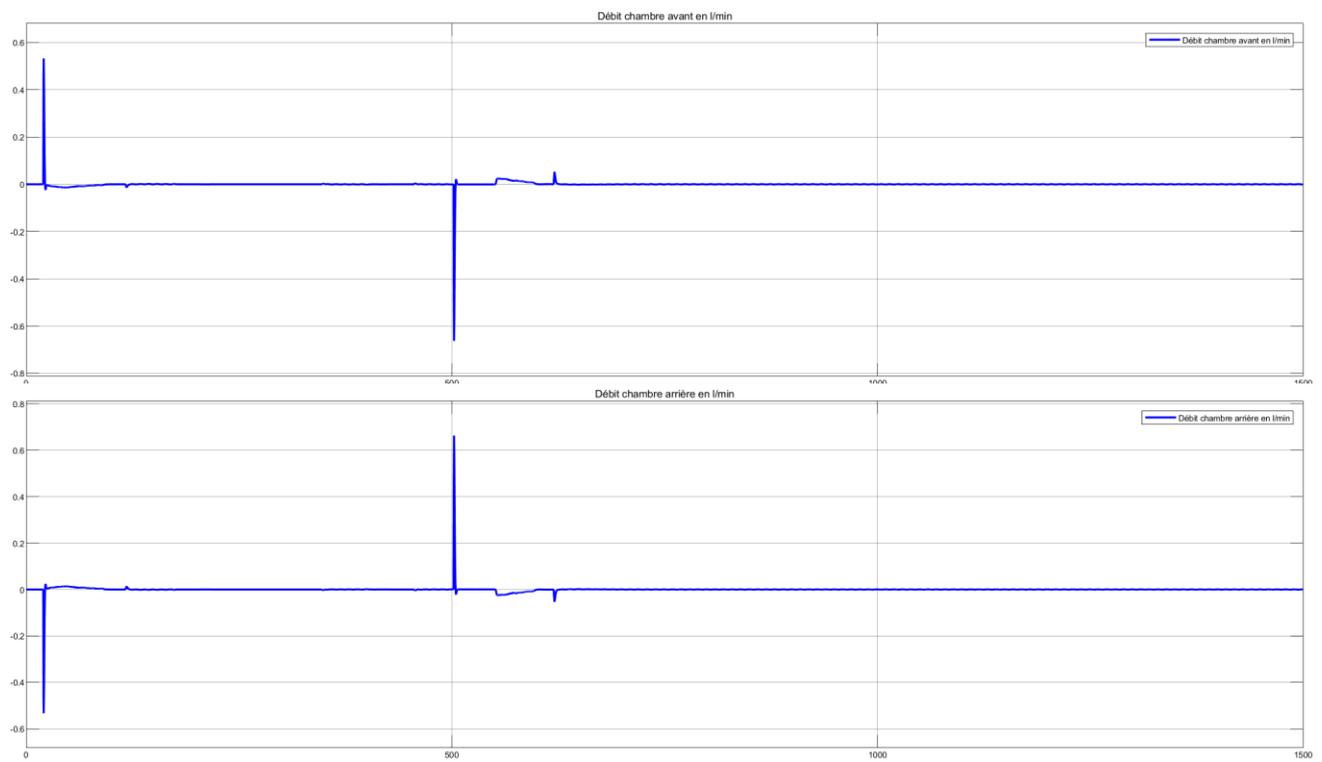


Figure 67 : visualisation des débits dans les chambres avant et arrière du vérin

## E. Exploration du modèle

### 1. Exploration du modèle de la chaîne d'information : Simulink et Stateflow

La chaîne d'information est modélisée à l'aide de **Simulink** et de **Stateflow** (Figure 68). Le fonctionnement du système nécessite une boucle pour asservir la barre en position. Cette boucle est régulée par un correcteur PID. La seconde boucle permet de respecter la consigne de cap. La logique est programmée sous forme de diagrammes d'états avec **Stateflow** et permet au système de régulation de se mettre en action uniquement si le bateau quitte son cap durant plus de 20 s.

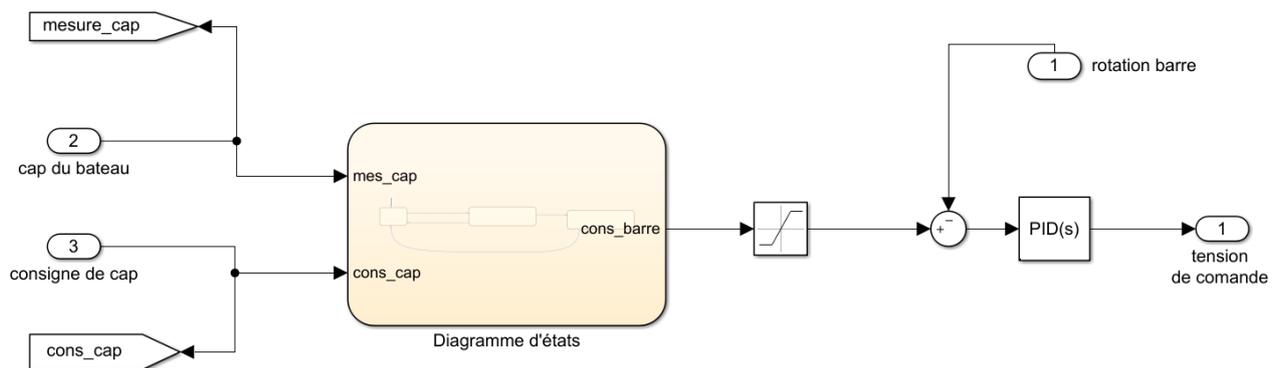


Figure 68 : modèle MATLAB - Simulink de la chaîne d'information du pilote automatique

Diagramme d'état de la boucle de cap :

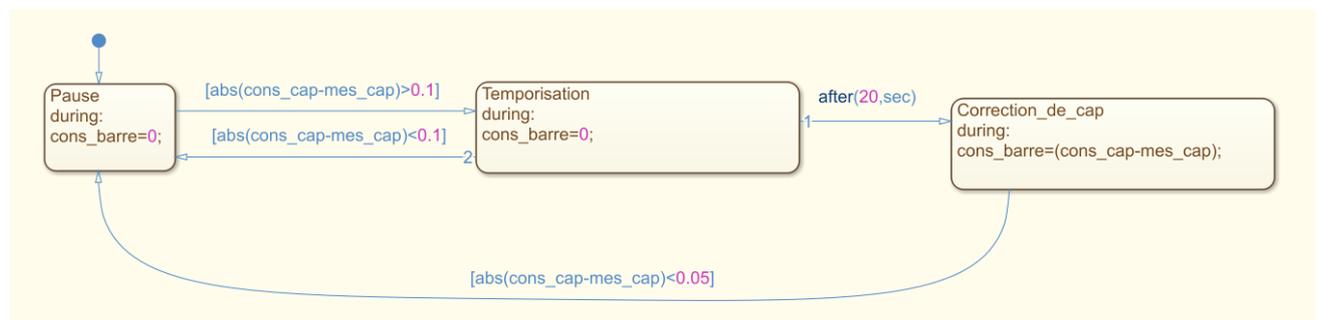


Figure 69 : modélisation de la commande de cap avec des diagrammes d'états.

## 2. Exploration du modèle de la chaîne d'énergie : Simscape Electric Library

La partie électrique de la chaîne d'énergie est modélisée à l'aide des éléments de base de la bibliothèque de composants électriques de **Simscape**. Le moteur à courant continu est modélisé à l'aide d'une inductance, d'une résistance et d'un composant permettant de convertir la puissance électrique en puissance mécanique. Des mesures de courant et de tension sont réalisées dans le circuit électrique, une mesure de couple est également effectuée sur l'arbre moteur.

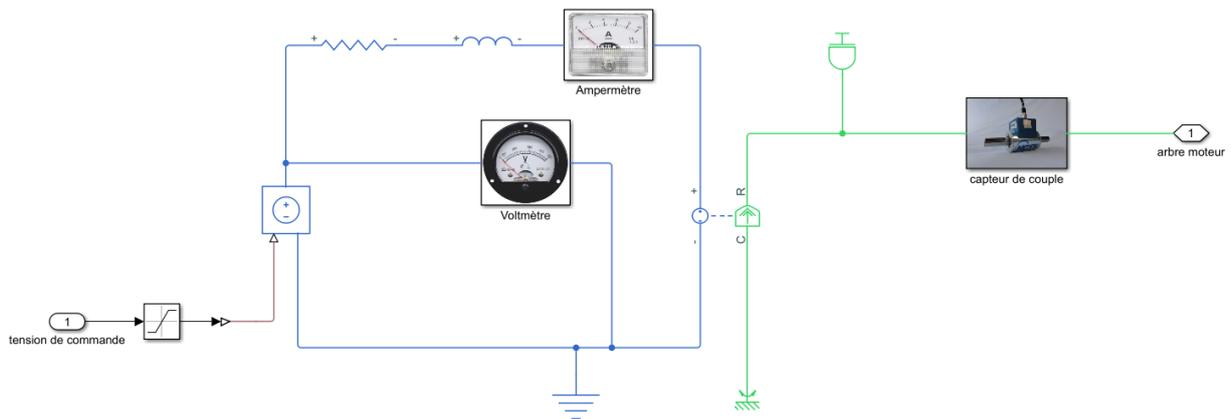


Figure 70 : modèle Simscape de la partie électrique de la chaîne d'énergie du pilote automatique

### 3. Exploration de la chaîne d'énergie : Simscape Fluids

La partie hydraulique de la chaîne d'énergie est réalisée à l'aide de composants extraits de la bibliothèque **Fluids**. La pompe à double sens de flux à cylindrée variable directement dans un vérin double effet. Des limiteurs de pression permettent l'évacuation du fluide lorsque la tige du vérin est en butée. Des mesures de pression et de débit sont réalisées en différents points du circuit hydraulique.

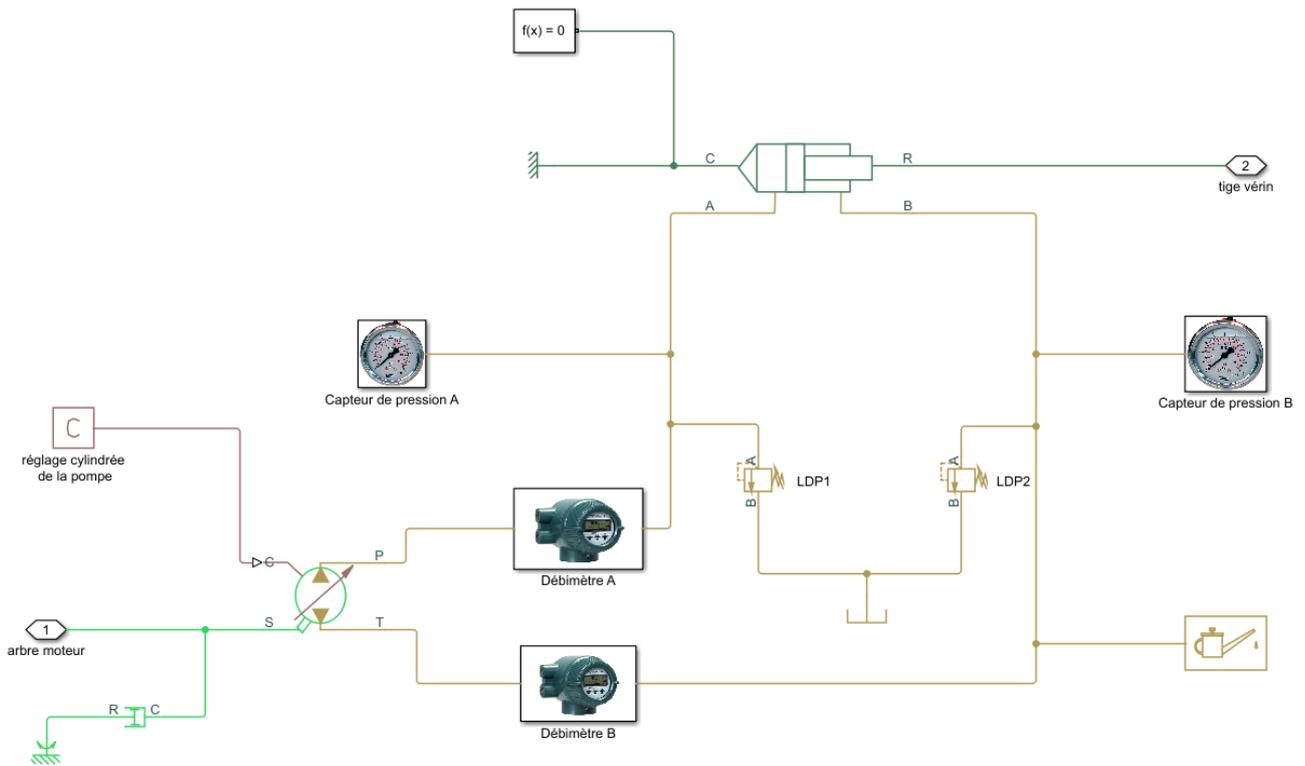


Figure 71 : modèle réalisé avec Simscape Fluids de la partie hydraulique de la chaîne d'énergie du pilote automatique



## 5. Exploration de la chaîne d'énergie : Simulink

Le modèle qui permet de prendre en compte la dynamique du bateau et les perturbations qu'il subit est réalisé avec **Simulink**.

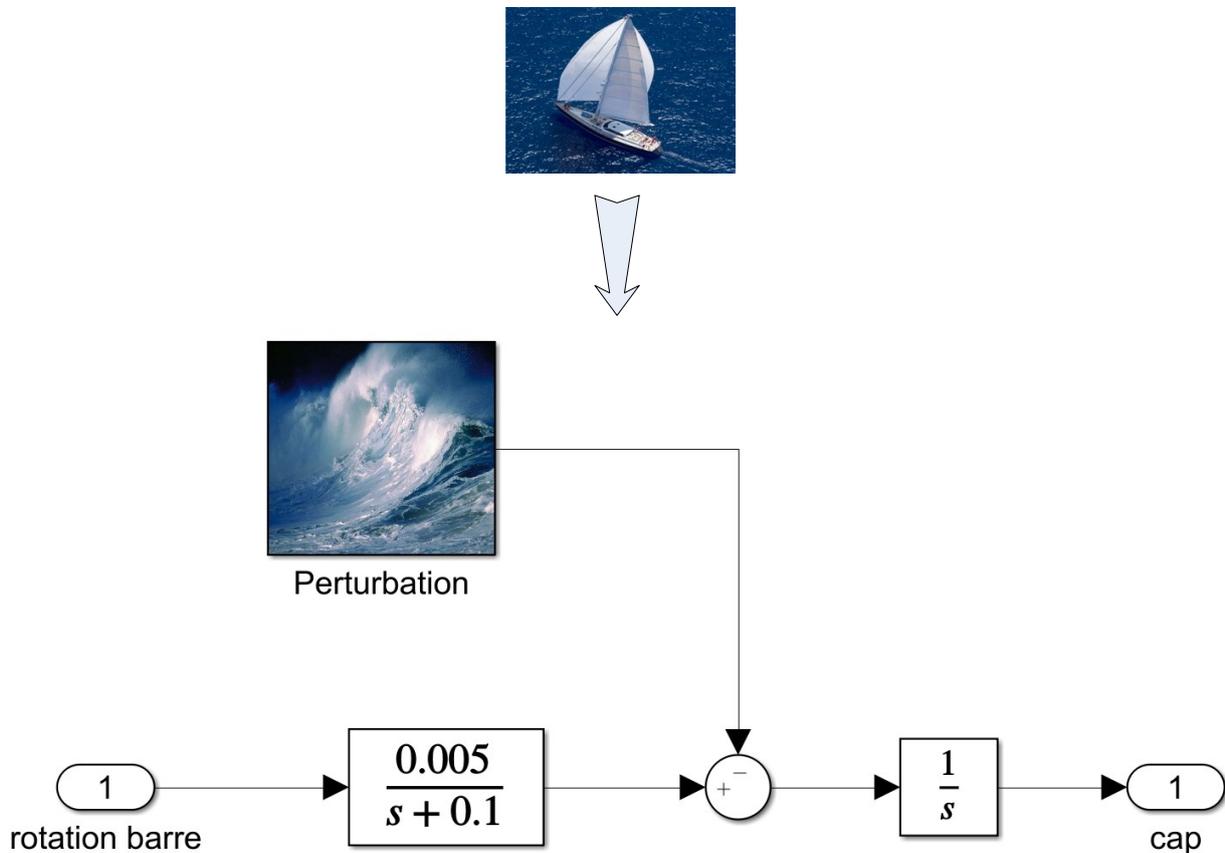


Figure 73 : Modélisation Simulink du comportement dynamique du bateau

## F. Pilotage interactif du modèle

Il est également possible d'illustrer différemment le modèle en associant chaque sous-système à son équivalent visuel sur le système réel et de le piloter de manière interactive. Pour cela la bibliothèque de Simulink **Dashboard** sera utilisée. Son utilisation et le paramétrage des blocs est abordée plus tard dans l'ouvrage (page 183).

**Ouvrir** le fichier « **pilote\_hydraulique\_dashboard.slx** ». Ce modèle est le même que le précédent, un autre point de vue a été choisi pour l'illustration des sous-systèmes et des éléments de paramétrage et de visualisation interactifs ont été ajoutés. Nous verrons l'importance qu'il faut accorder à la présentation du modèle afin de le rendre le plus accessible possible pour l'analyse et la modification.

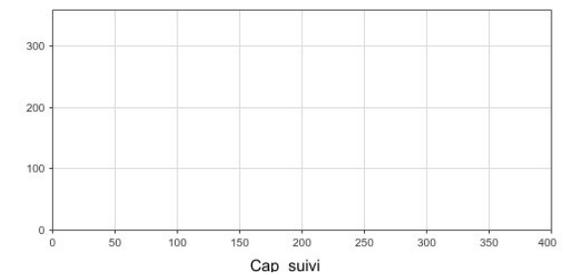
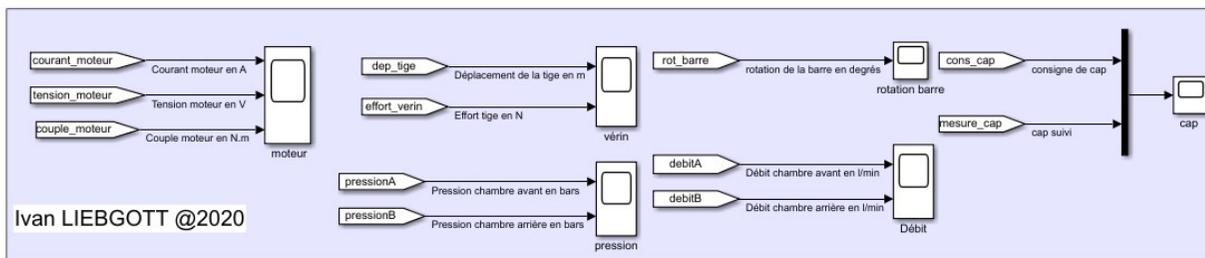
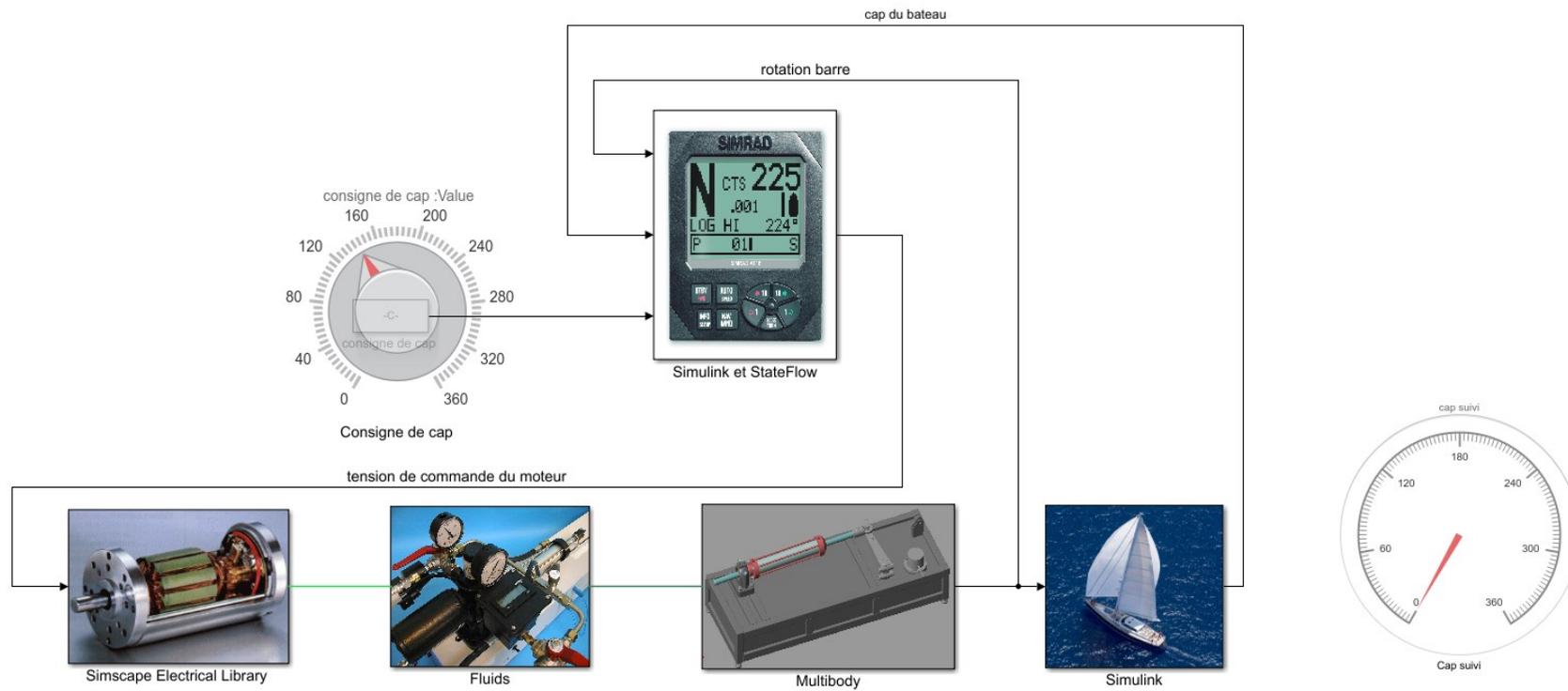
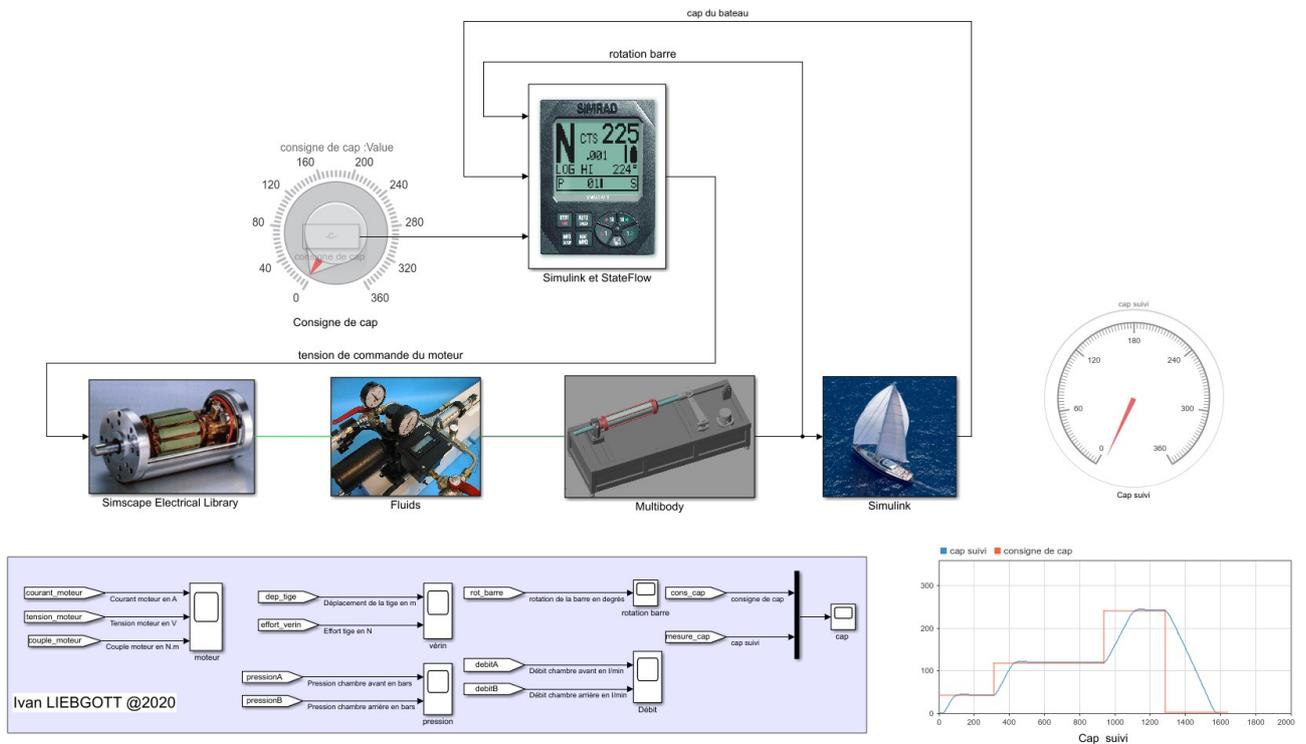


Figure 74 : modèle du pilote hydraulique avec pilotage interactif

**Lancer** la simulation et **Modifier** la consigne de cap en agissant sur le bouton rotatif durant la simulation.

**Observer** le cap suivi dans le scope interactif et sur le quadrant de visualisation.



**Figure 75 : utilisation du pilotage interactif d'un modèle**

### III. Exemples de modèles multi-physique et exploitations possibles

#### A. Le robot Maxpid

Le robot Maxpid est un sous-système d'un robot initialement utilisé pour la cueillette de fruit et le tri de déchets. Maxpid est asservi et peut orienter angulairement son bras conformément à une consigne. Un système de transformation de mouvement de type vis-écrou est utilisé pour transformer le mouvement de rotation de l'axe de sortie du moteur (vis) en mouvement de rotation du bras.

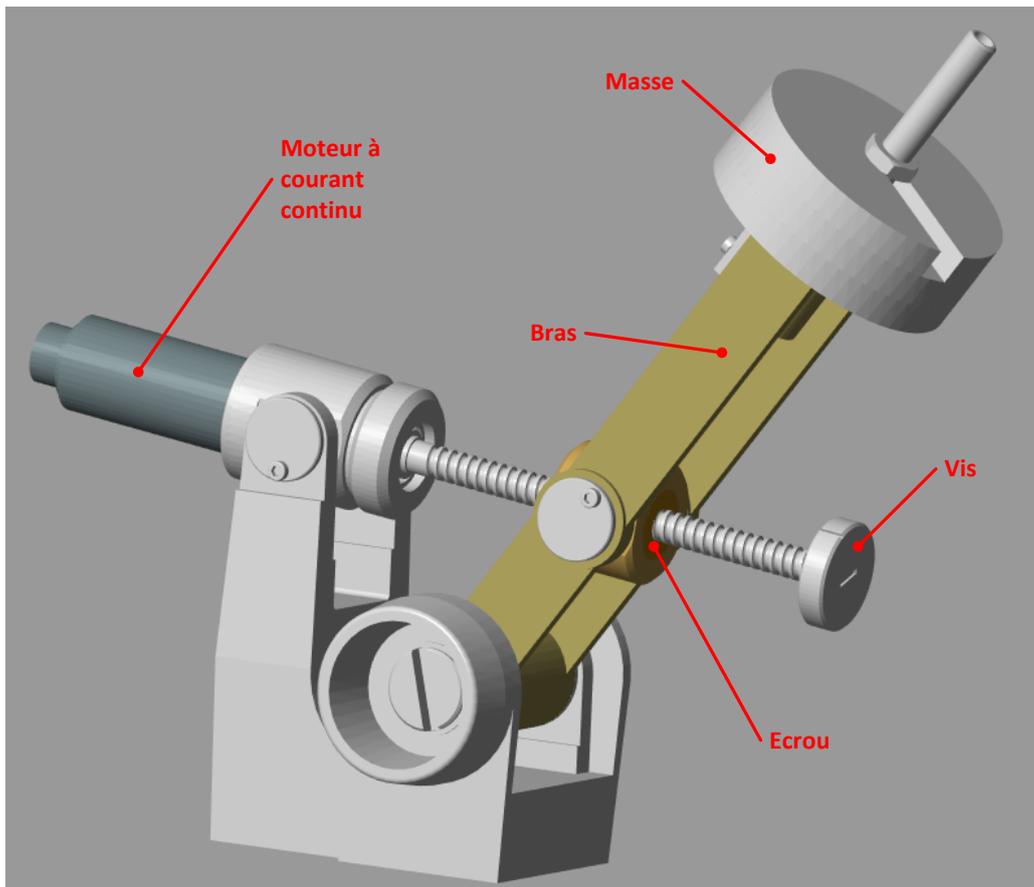


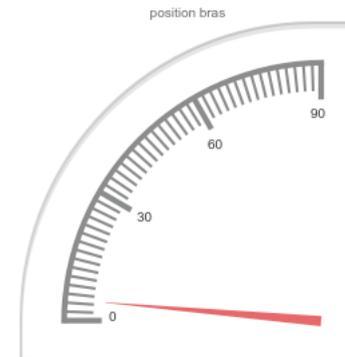
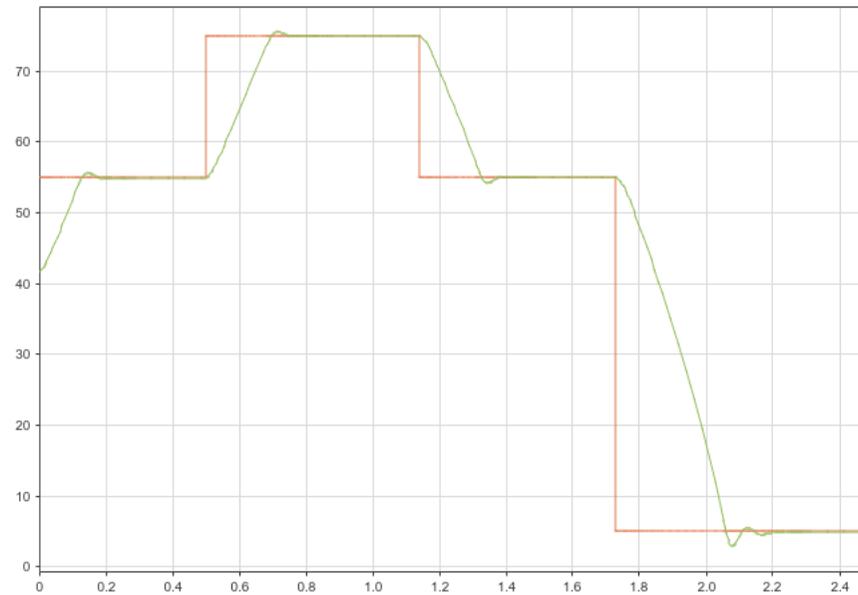
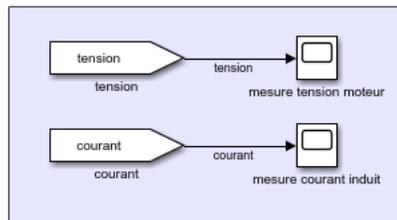
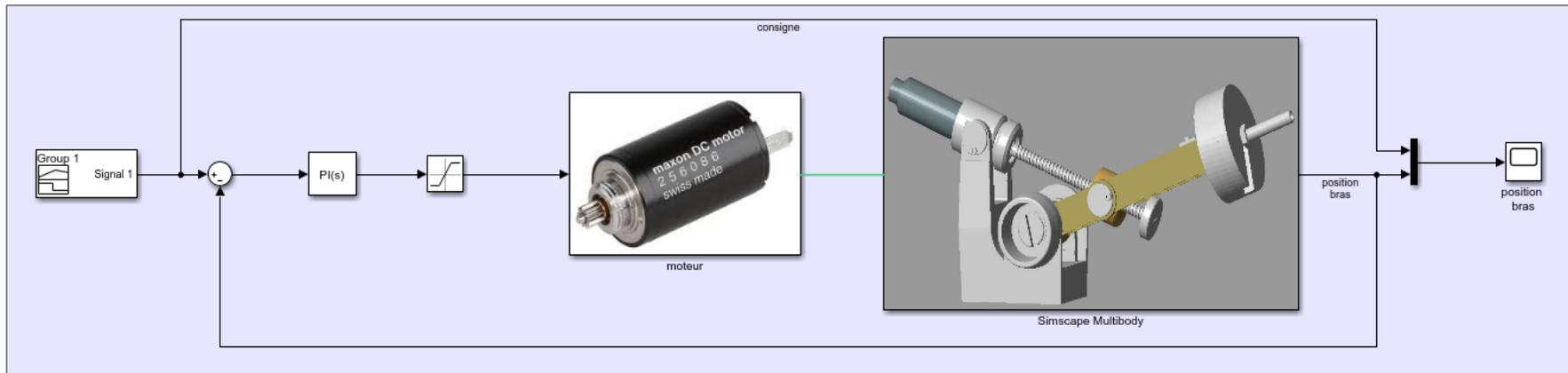
Figure 76 : présentation du robot Maxpid

La nécessité d'étudier l'asservissement de position du bras de Maxpid impose de disposer d'un modèle validé. Dans ce cas précis ce modèle est très difficile à mettre en place analytiquement car la partie mécanique est complexe et les équations différentielles qui caractérisent son comportement sont non linéaires. La résolution de ces équations n'est possible qu'en considérant des petites variations autour d'une position donnée ce qui est très contraignant dans la démarche d'analyse.

Le recours à la modélisation multi-physique présente dans ce type de cas un avantage évident. L'ajout de la maquette 3D dans le modèle va prendre en compte tous les effets dynamiques sans que le moindre calcul ne soit nécessaire. Le modèle sera valable pour toute la plage de variation angulaire de la position bras.

**Ouvrir** le fichier « **Maxpid.slx** », **Explorer** les sous-systèmes et **Lancer** la simulation.

La fenêtre **Mechanics Explorer** permet de visualiser les mouvements du bras du robot.



@ Ivan LIEBGOTT 2020

Figure 77 : modélisation multi-physique du robot MAXPID

Visualisation de la position du bras et de la consigne :

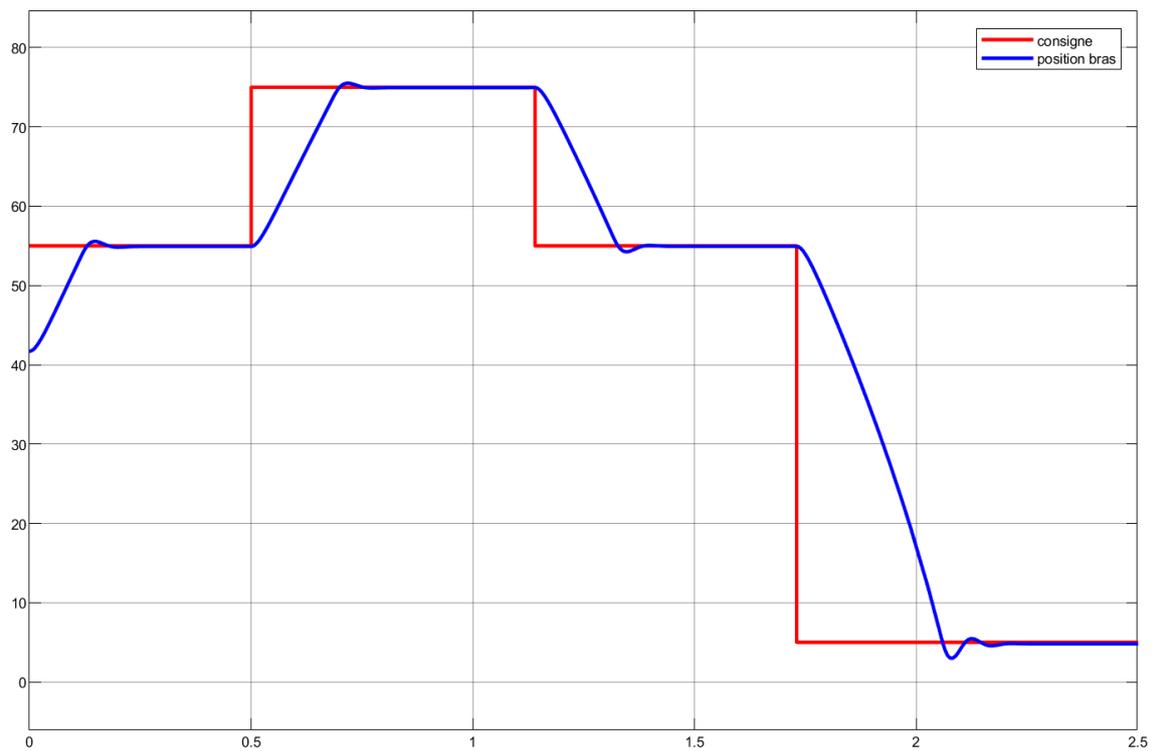


Figure 78 : position du bras du robot MAXPID

Visualisation de la tension de commande :

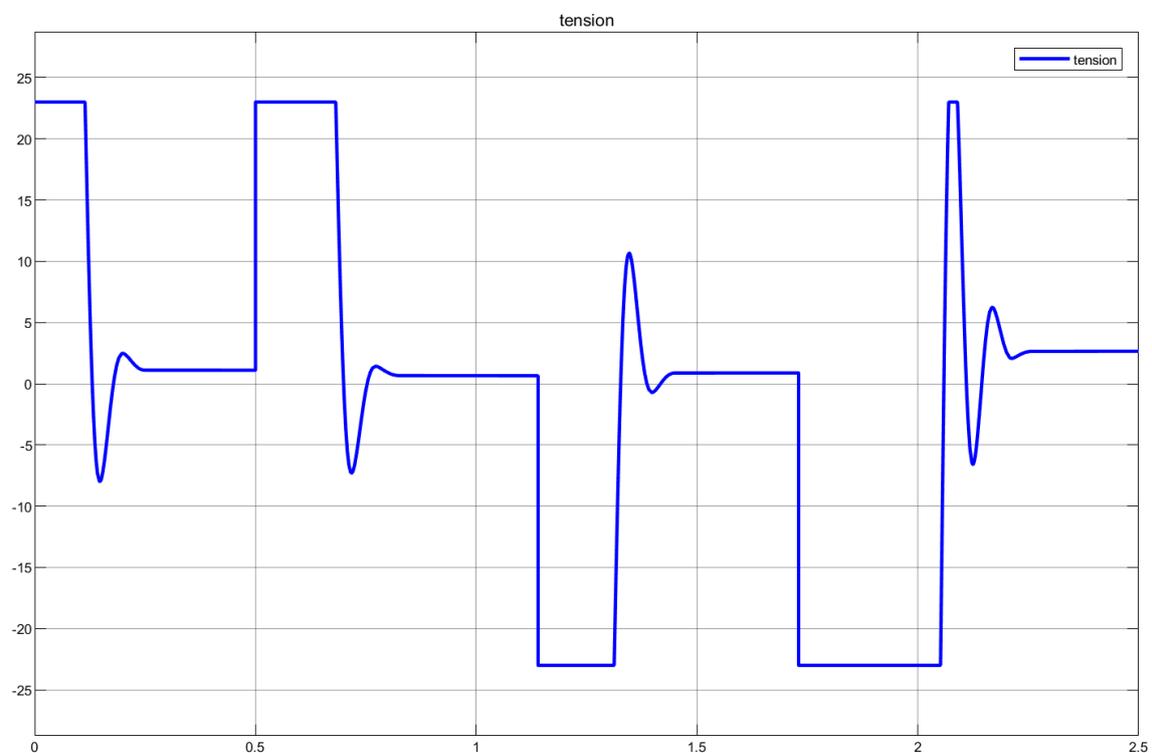


Figure 79 : visualisation de la tension de commande du moteur

Visualisation du courant dans l'induit du moteur :

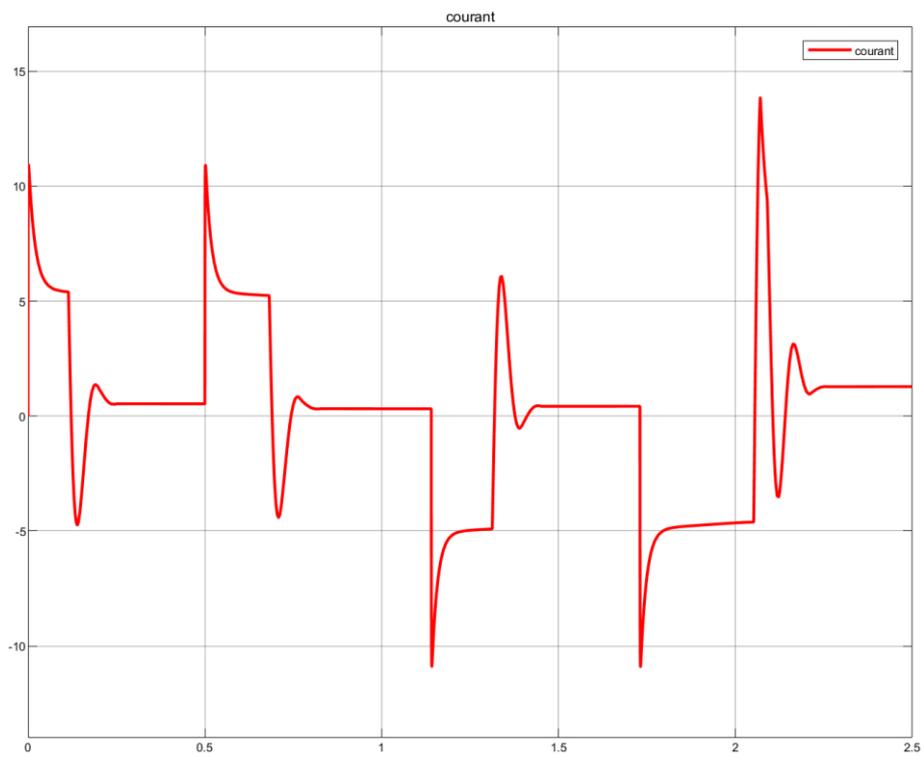


Figure 80 : visualisation du courant dans l'induit du moteur

Il est possible de modifier les paramètres de masse directement dans **Multibody** afin de voir l'influence de l'ajout de masse en bout de bras, et agir sur tout type de paramètre.

## B. L'axe linéaire Control'X

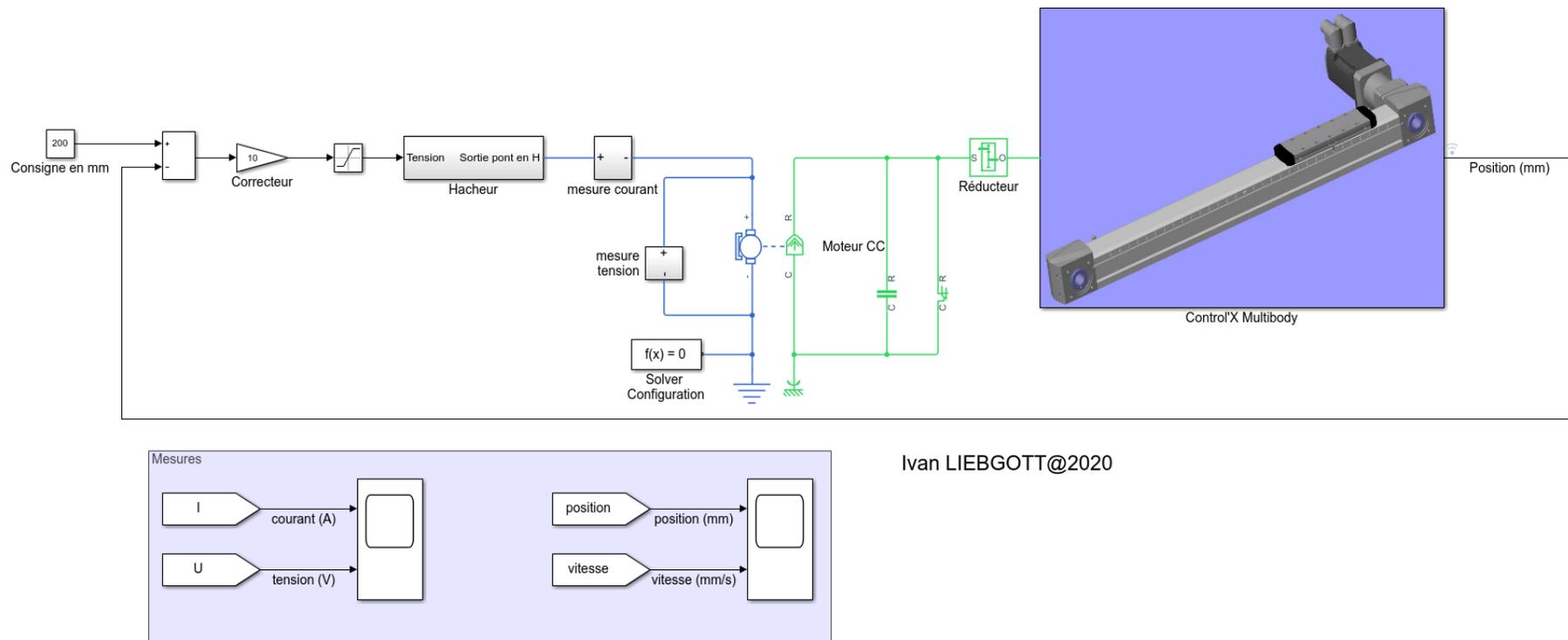
---

Le système Control'X est un axe linéaire asservi. Un motoréducteur à courant continu est accouplé un système de transformation de mouvement de type poulies/courroie. Le chariot de l'axe est solidaire de la courroie et se déplace en translation.



Figure 81 : l'axe linéaire Control'X

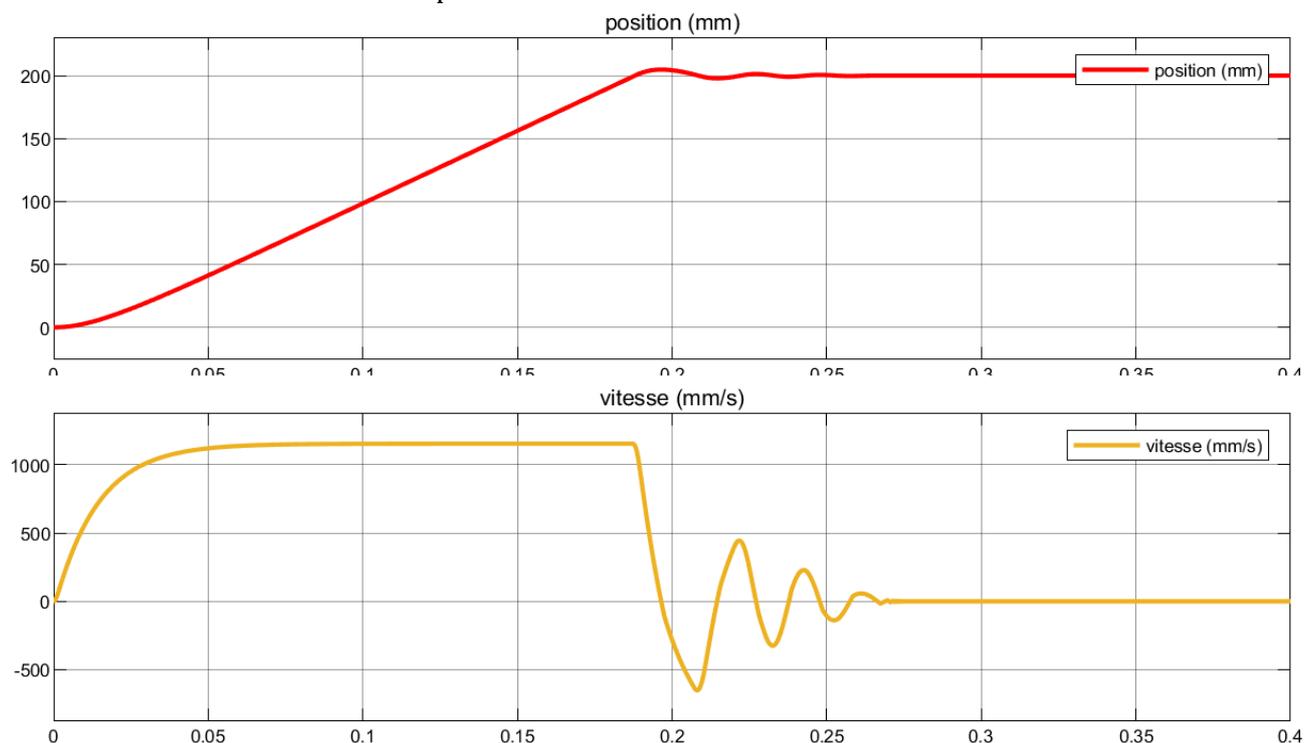
**Ouvrir** le fichier « **ControlX.slx** » et **Explorer** les sous-systèmes.



**Figure 82 : modélisation multi-physique du système Control'X**

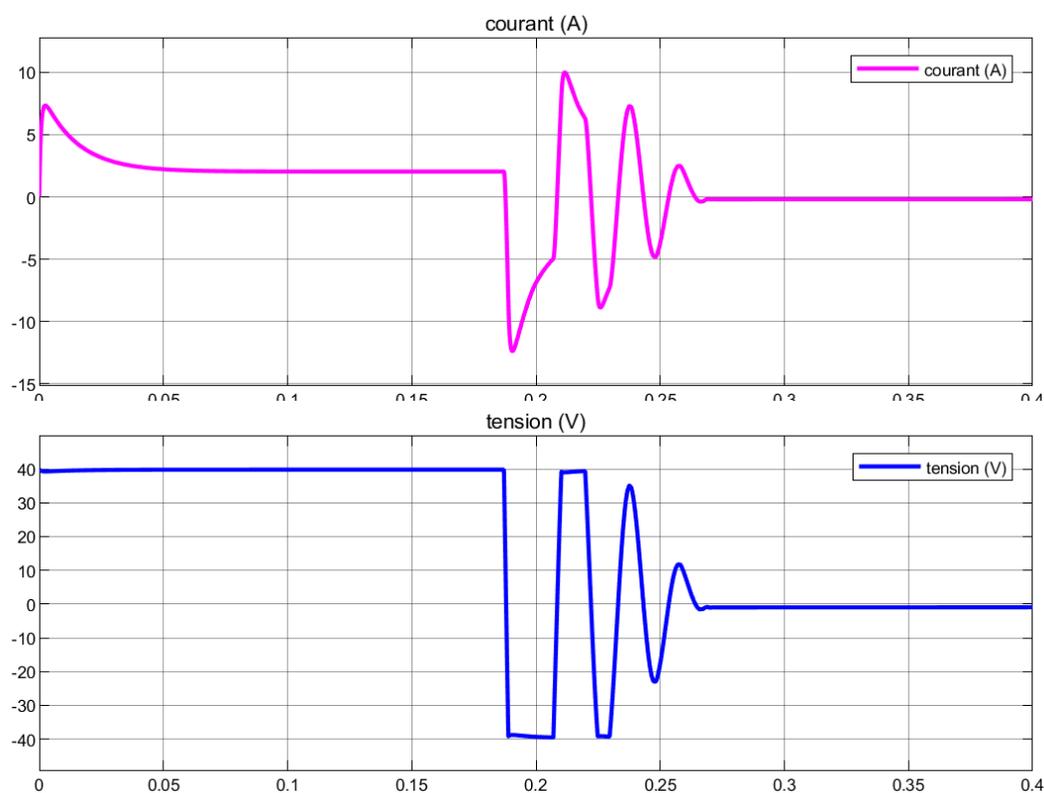
**Lancer** la simulation et observer les résultats obtenus.

Visualisation de la vitesse et de la position de l'axe :



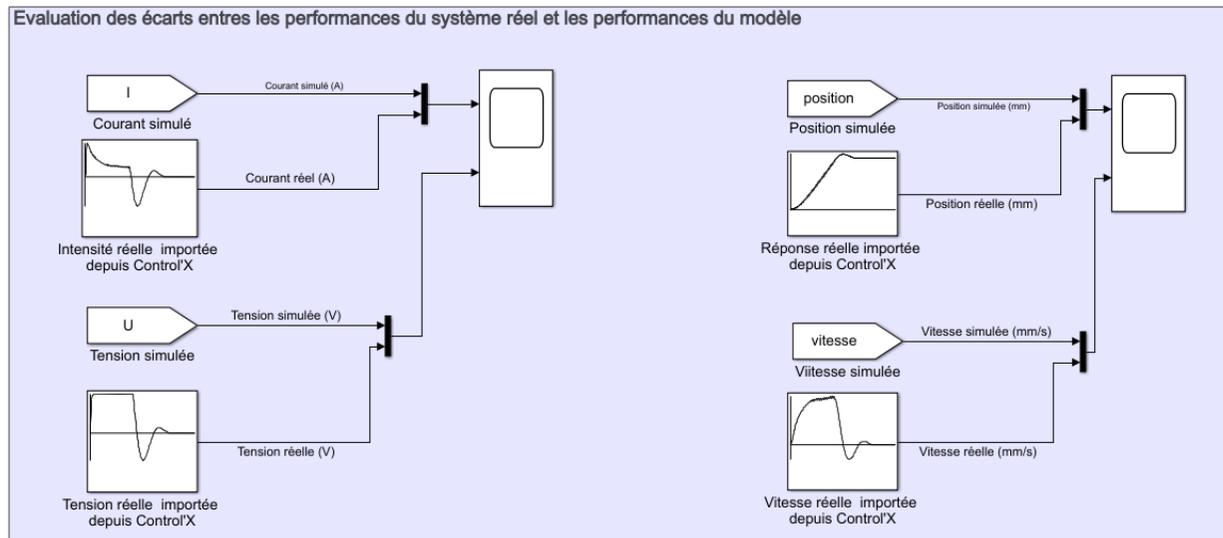
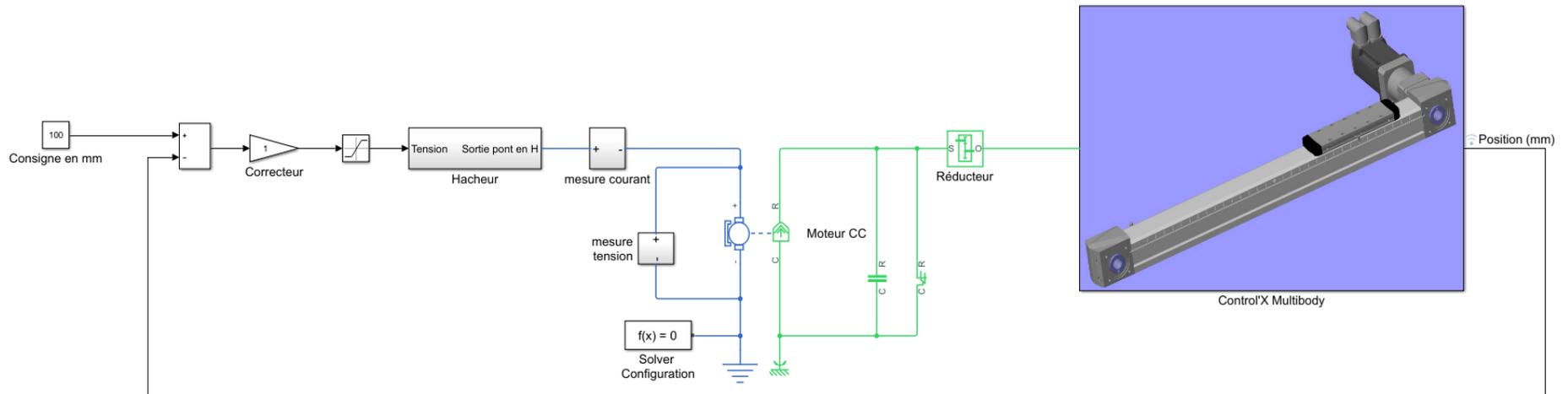
**Figure 83 : visualisation de la vitesse et de la position de l'axe linéaire**

Visualisation de la tension de commande du moteur et du courant dans l'induit :



**Figure 84 : visualisation de la tension de commande du moteur et du courant dans l'induit**

La validation du modèle ne peut se faire que par comparaison avec les résultats expérimentaux. Le fichier « **ControlX\_compar\_reel\_modele.slx** » contient une comparaison entre les performances issues de la simulation et les performances relevées sur le système réel.  
**Lancer** la simulation et observer les résultats obtenus.



Ivan LIEBGOTT@2020

**Figure 85 : évaluation des écarts entre les performances du modèle et celles de l'axe réel**

Evaluation des écarts modèle/réel constatés sur la position et la vitesse :

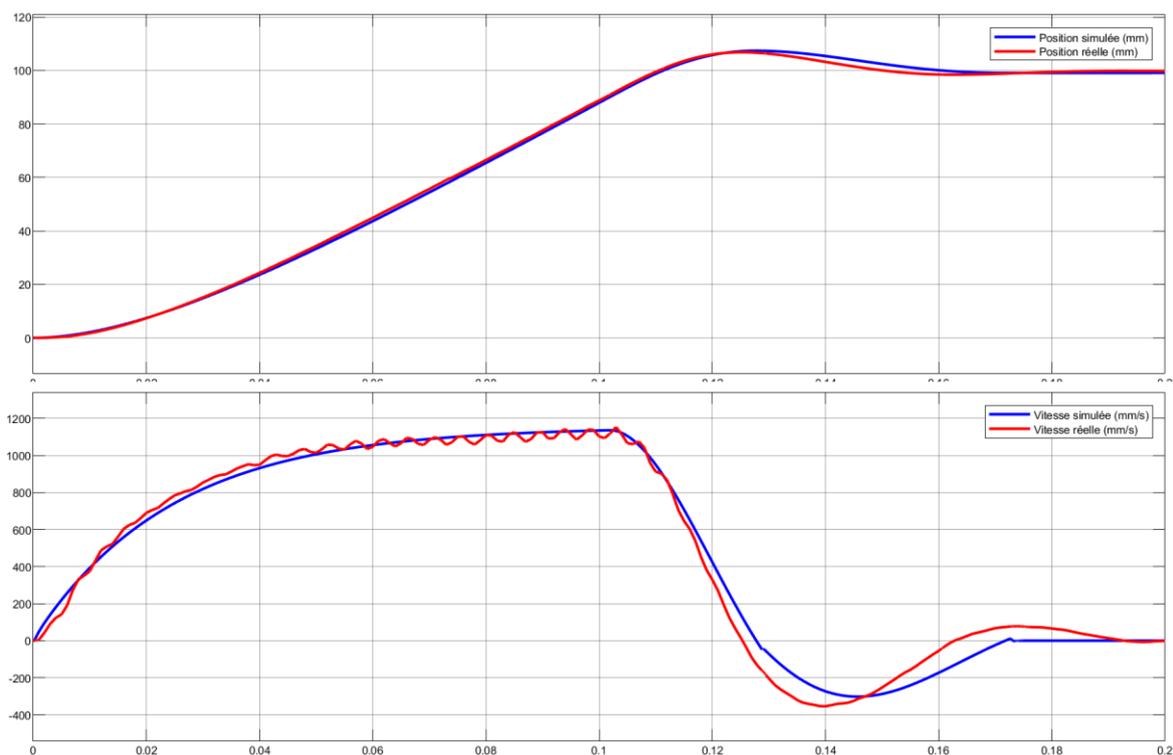


Figure 86 : évaluation des écarts réel/modèle pour la vitesse et la position de l'axe

Evaluation des écarts modèle/réel constatés sur la position et la vitesse :

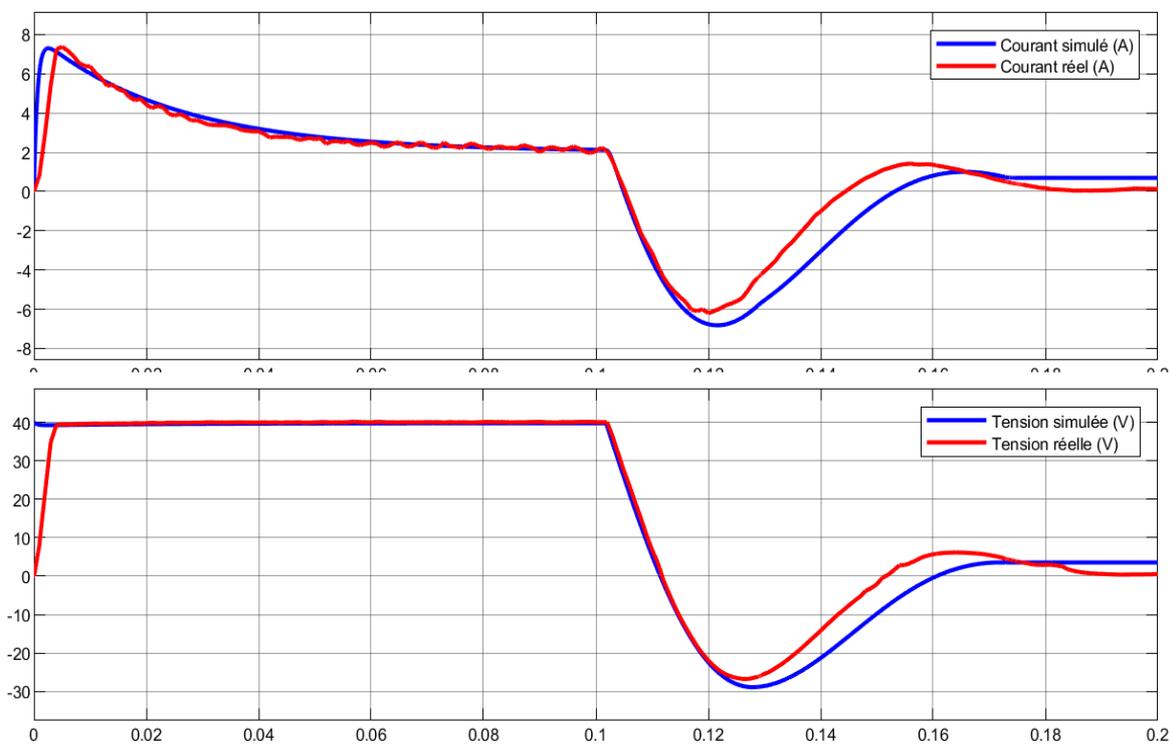


Figure 87 : évaluation des écarts réel/modèle pour le courant d'induit et la tension de commande

Les écarts constatés sont très faibles ce qui permet de mettre en évidence la pertinence de la démarche de modélisation multi-physique. Il sera possible de construire des modèles simples qui auront un comportement très proche des systèmes réels.

### C. Comment faire un modèle multi-physique avec MATLAB-Simulink ?

Le modèle multi-physique fait intervenir de nombreux outils de modélisation. Pour construire un modèle multi-physique, il faudra dans un premier temps prendre en main les bases des outils suivants :

- MATLAB
- Simulink
- Simscape et ses libraires fondamentales (mécanique, électrique, hydraulique...)
- Stateflow

Il faudra également être capable d'utiliser les composants des bibliothèques spécifiques en fonction des besoins de la modélisation :

- Multibody
- Fluids
- Electrical
- Drivelines

Cet ouvrage propose un apprentissage progressif de ces différents outils au travers d'exemples qui seront exploitables dans le cadre de la formation des ingénieurs. La première partie permet de se familiariser avec **Simscape** et la modélisation par composants (acausale). Ensuite, nous verrons les bases de la programmation en **MATLAB** et comment **MATLAB** et **Simulink** peuvent communiquer. La suite présentera une introduction à l'utilisation de **Stateflow** et de **Multibody**. La fin de l'ouvrage présente des outils d'analyse intéressants en lien direct avec la démarche de conception proposée dans l'ouvrage, comme la « **System Identification Toolbox** » ou « **Simulink Control Design** » qui permet de mener les études de **contrôle commande** des systèmes asservis. Des méthodes de conception et de réglage des correcteurs sont présentées en utilisant des outils graphiques interactifs particulièrement adaptés à la formation et à la compréhension des phénomènes.

La maîtrise de ces différents outils permettra d'aborder une grande partie des problèmes de modélisation qui pourront être rencontrés dans le cadre de la formation des ingénieurs.

# Chapitre 4 : Prise en main de Simscape

Ce chapitre a pour objectif de présenter les bases théoriques de la modélisation multi-physique acausale en utilisant **Simscape** et ses bibliothèques.

## I. Introduction à la modélisation acausale avec Simscape

Au sein de l'environnement **MATLAB-Simulink**, l'outil **Simscape** propose une approche acausale de la modélisation et possède une vaste bibliothèque de composants élémentaires dont le comportement physique est déjà modélisé. La démarche de modélisation acausale consiste à réaliser un assemblage de composants pour modéliser le comportement d'un système. Le principe de calcul s'appuie sur un bilan de puissance à chaque nœud du modèle et ne repose pas sur le principe de causalité, d'où le nom de modélisation acausale.

Prenons l'exemple d'un circuit R-L.

A  $t=0$  s, une source de tension alimente un circuit RL. Le modèle doit permettre d'analyser l'évolution du courant  $i(t)$  qui traverse le circuit.

$$R=20 \Omega$$

$$L=0.01 \text{ mH}$$

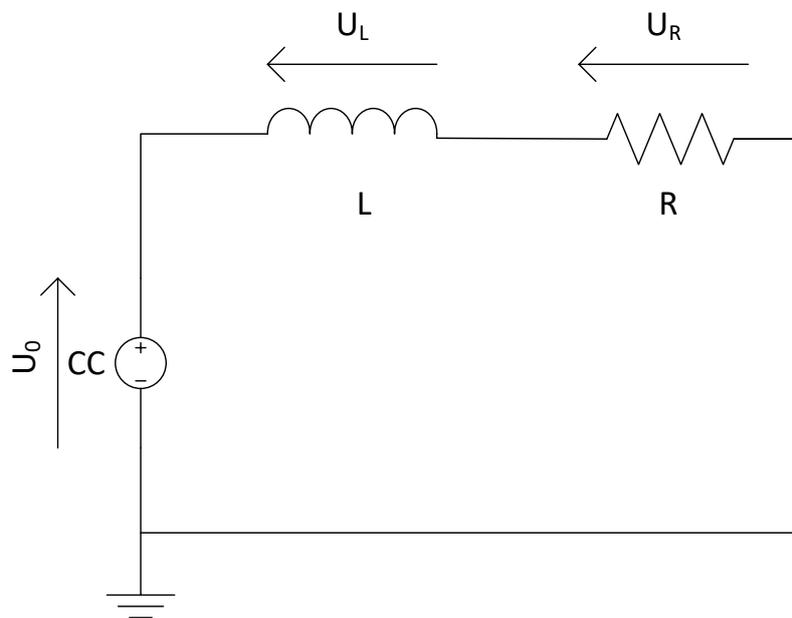


Figure 88 : circuit R-L

## A. Choix des composants

La modélisation acausale du circuit RL est obtenue en utilisant les composants suivants :

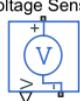
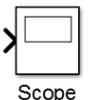
Fonction du composant	Représentation	Bibliothèque
Source de tension continue	 DC Voltage Source	Simscape/Foundation Library/Electrical/Electrical Sources
Résistance	 Resistor	Simscape/Foundation Library/Electrical/Electrical Element
Inductance	 Inductor	Simscape/Foundation Library/Electrical/Electrical Element
Référence électrique	 Electrical Reference	Simscape/Foundation Library/Electrical/Electrical Element
Capteur de courant	 Current Sensor	Simscape/Foundation Library/Electrical/Electrical Sensors
Capteur de tension	 Voltage Sensor	Simscape/Foundation Library/Electrical/Electrical Sensors
Conversion d'un signal physique en signal Simulink	 PS-Simulink Converter	Simscape/Utilities
Moniteur	 Scope	Simulink/Sinks
Solveur	 Solver Configuration	Simscape/Utilities

Figure 89 : les composants nécessaires à la modélisation du circuit R-L avec Simscape

## B. Placement et assemblage des composants

Dans la fenêtre **Simulink Library Browser** (bibliothèque de Simulink), sélectionner la commande

**File/New/Model**  pour créer un nouveau fichier.

**Glisser/Déposer** les différents composants à partir des bibliothèques et les disposer dans la fenêtre de travail comme indiqué sur la Figure 90 ou ouvrir directement le fichier **circuit\_RL\_start.slx**. Vous pouvez vous reporter à la Figure 91 pour prendre connaissance des commandes utiles à l'orientation des composants.

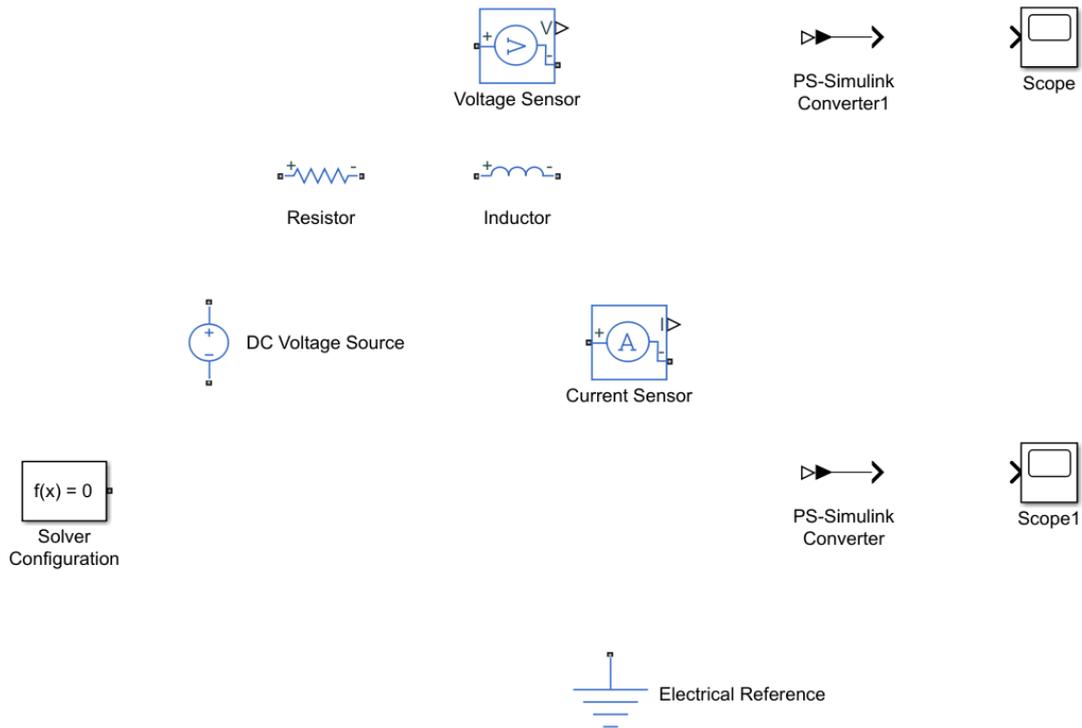


Figure 90 : composants nécessaires à la modélisation du circuit RL

Par défaut le nom des composants n'est pas visible. Pour afficher le nom de tous les composants, sélectionner tous les composant avec la souris (ou Ctrl+A). Cliquer ensuite sur un composant avec le bouton droit de la souris pour faire apparaître le menu contextuel et sélectionner **Format/Show Bloc Name/on**.

Commandes utiles		
Fonctions	Actions	Raccourcis clavier
Rotation d'un composant sens horaire	<b>Clic droit</b> sur le composant, <b>Rotate&amp;Flip/Clockwise</b>	Ctrl+R
Rotation d'un composant sens antihoraire	<b>Clic droit</b> sur le composant, <b>Rotate&amp;Flip/CounterClockwise</b>	Shift+Ctrl+R
Inversion gauche/droite d'un composant	<b>Clic droit</b> sur le composant, <b>Rotate&amp;Flip/Flip Bloc/Left Right</b>	Ctrl+I
Inversion haut/bas d'un composant	<b>Clic droit</b> sur le composant, <b>Rotate&amp;Flip/Flip Bloc/Up Down</b>	
Copier un composant	<b>Clic droit</b> sur le composant à dupliquer, puis <b>glisser/déposer</b> en gardant le bouton droit appuyé	
Supprimer un composant	<b>Clic droit</b> sur le composant puis sélectionner <b>Delete</b>	Suppr

Figure 91 : commandes utiles

Relier les composants pour obtenir le modèle ci-dessous :

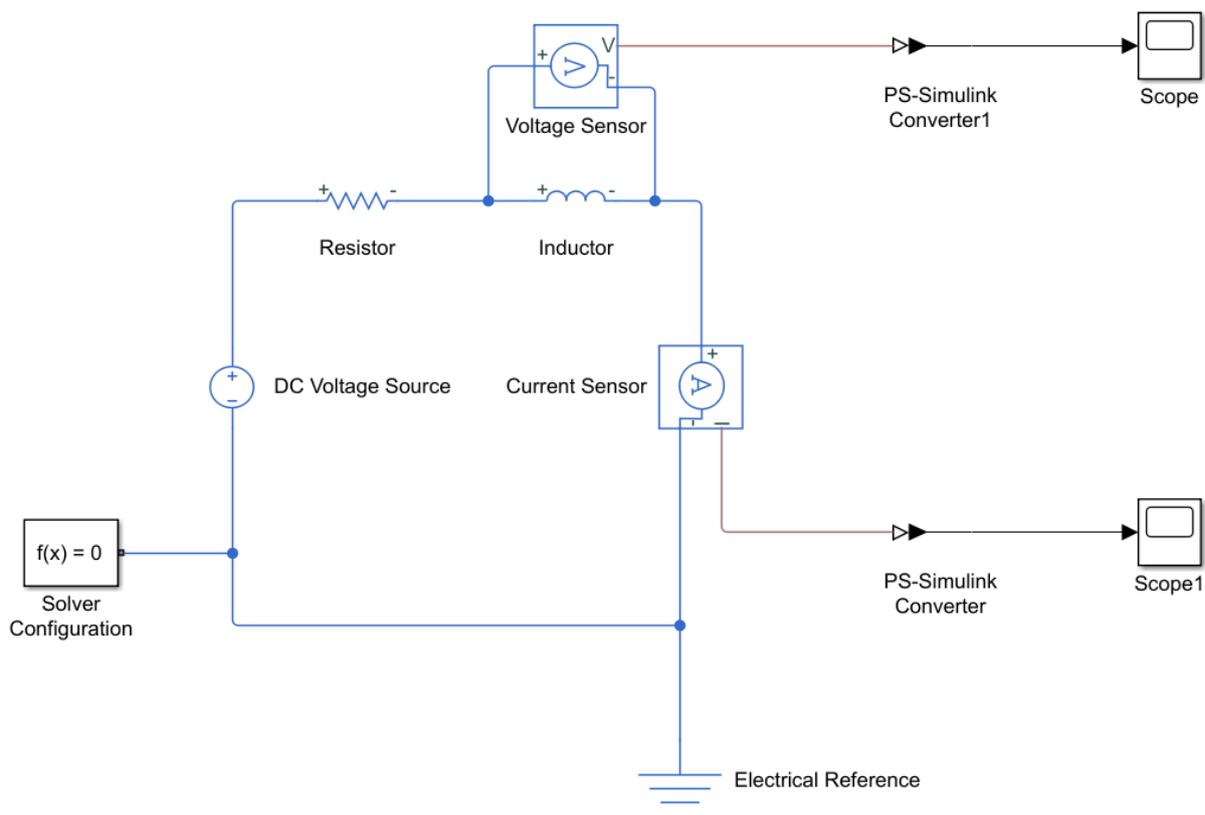


Figure 92 : Modèle Simscape du circuit R-L

Commandes utiles		
Fonctions	Actions	Raccourcis clavier
Relier deux composants	Clic gauche sur le port du premier composant puis se déplacer en maintenant le bouton gauche enfoncé jusqu'au port du second composants	

Figure 93 : commandes utiles

### C. Les différents types de ports et de connexions

Avant de paramétrer les blocs et de lancer la simulation, analysons la nature des différentes connexions entre les blocs du modèle. Le modèle présenté utilise différents types de ports et de connexions. Les ports peuvent être classés en trois catégories (Figure 94).

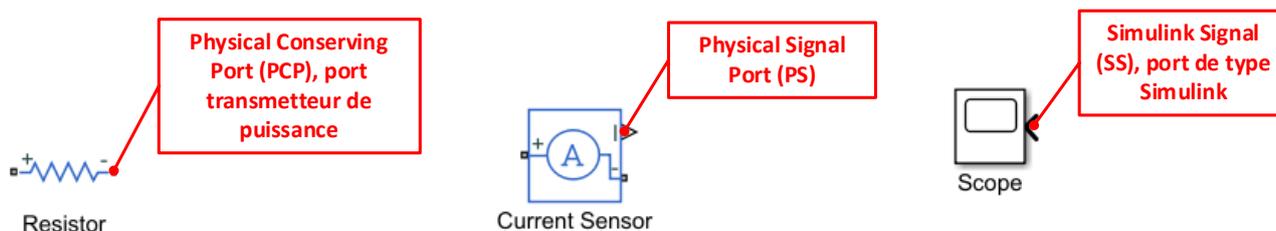


Figure 94 : les différents types de ports de Simscape

- Les ports de type « **Physical Conserving Port** » (**PCP**) qui transmettent la puissance entre deux composants d'un même domaine. Les connexions relatives à ces ports ne sont pas orientées et sont analogues aux liens qui relient les composants dans la réalité et se situent dans le domaine de la modélisation acausale. Dans le cas de notre modèle qui ne fait intervenir que le domaine électrique, ces connexions sont des fils électriques. Ils sont donc traversés par un courant et il est possible de mesurer une différence de potentiel entre deux points appartenant à ces connexions. La mesure du courant dans le circuit se fait par un capteur de courant placé en série dans le circuit. La mesure de la tension aux bornes de la bobine se fait par un capteur de tension placé en parallèle, aux bornes de la bobine.
- Des ports de type « **Physical Signal Port** » (**PSP**) qui transmettent des signaux physiques prélevés à l'aide de capteurs sur le modèle. Ces signaux sont orientés et sont l'image de la grandeur physique prélevée. Ces ports et ces connexions fonctionnent selon le principe de causalité.
- Des ports de type « **Simulink Signal** » (**SS**), qui transmettent des signaux numériques orientés exploitables par les blocs de la bibliothèque Simulink. Ces ports et ces connexions fonctionnent selon le principe de causalité.

Le type de connexion peut être identifié par le format du port qui lui est associé.

La construction des modèles nécessite une parfaite compréhension de la nature des informations et des signaux qui parcourent les différentes connexions ainsi que l'identification de la nature des ports des blocs qui interviennent dans la modélisation. Des ports de natures différentes ne peuvent pas être reliés. De plus deux ports de types **Physical Conserving Port** (transmetteur de puissance) ne peuvent être reliés que s'ils appartiennent au même domaine physique. Afin d'apporter plus de lisibilité aux modèles réalisés avec Simscape, chaque domaine physique est représenté avec une couleur différente (Figure 95).

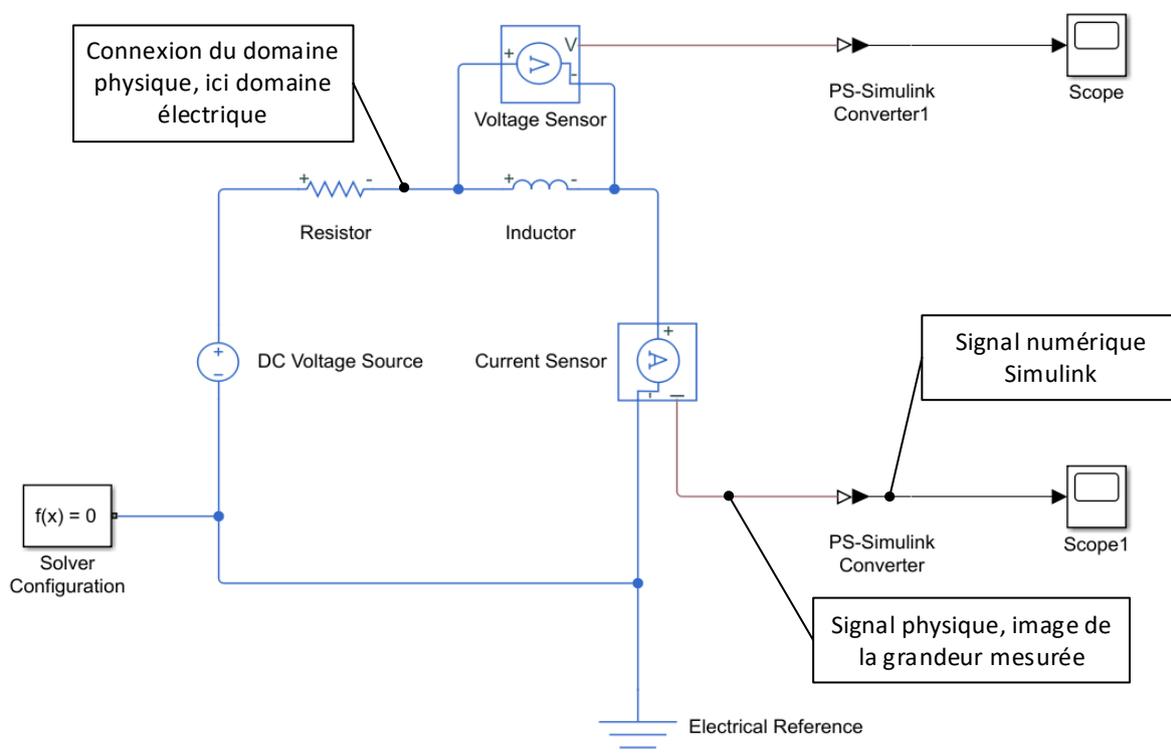


Figure 95 : identification des connexions d'un modèle Simscape

## D. Paramétrage des composants

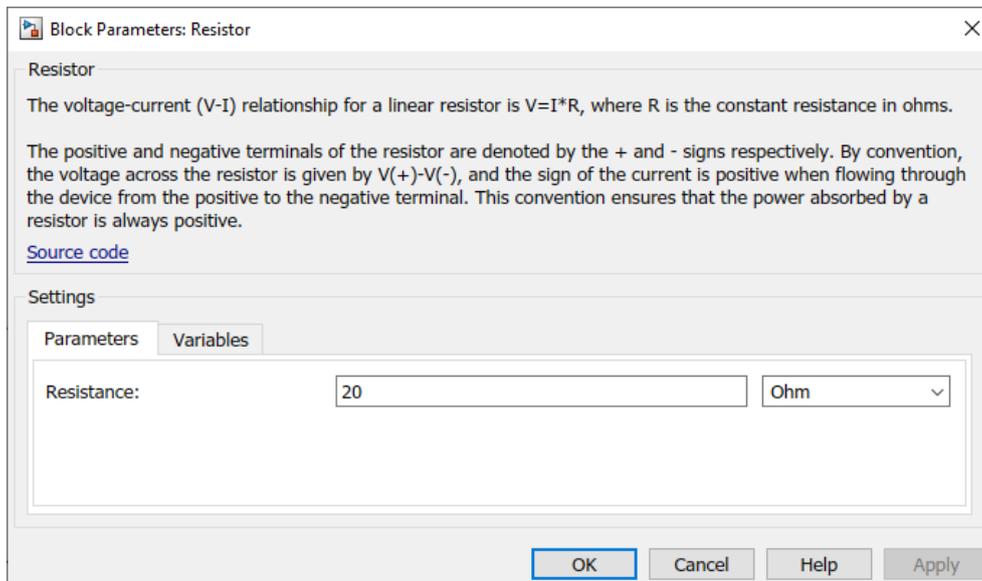
Le paramétrage des composants est accessible par un double-clic sur le bloc.

RESISTOR



Simscape/Foundation Library/Electrical/Electrical Element

Le paramétrage de ce composant consiste à saisir la valeur de la résistance

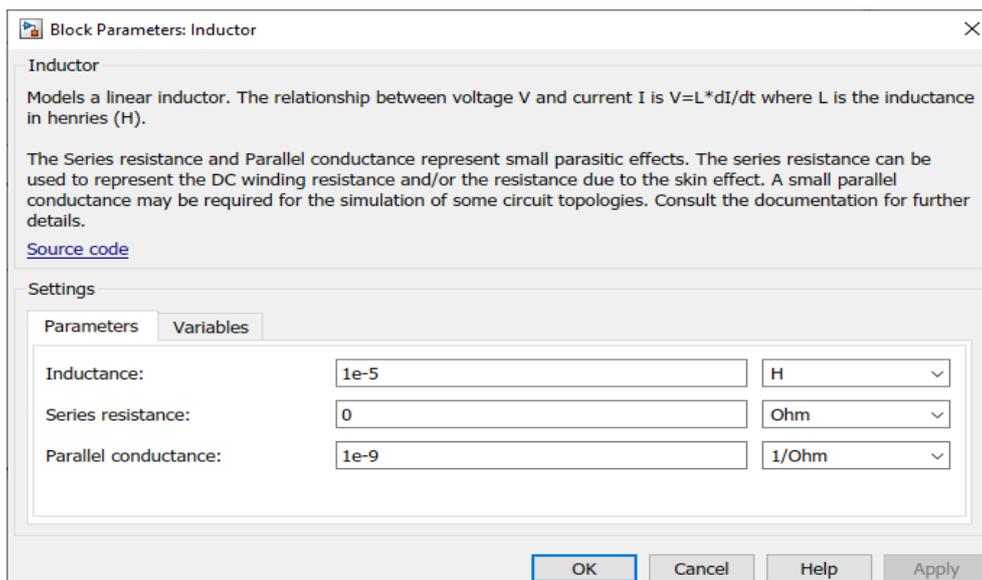


INDUCTOR



Simscape/Foundation Library/Electrical/Electrical Element

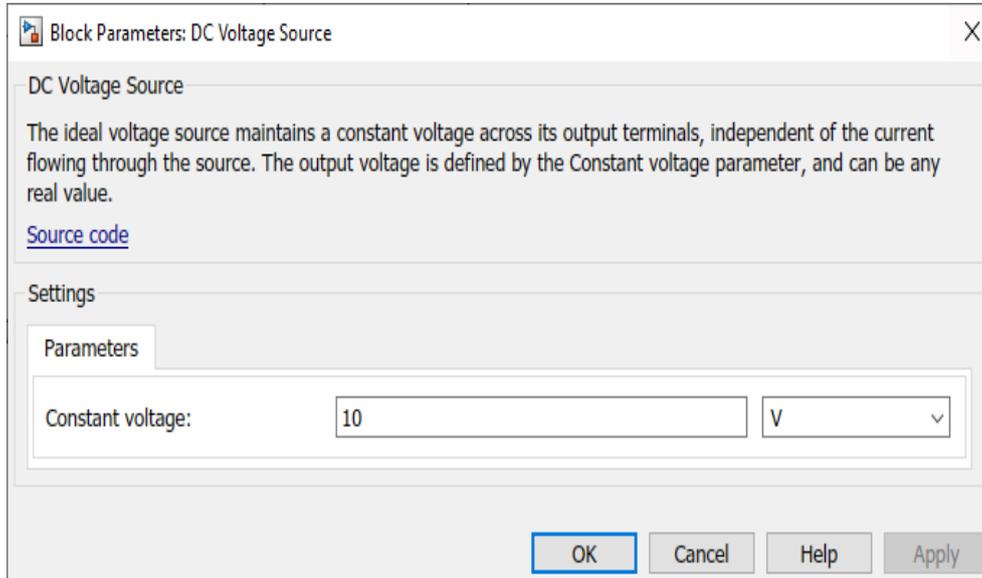
Le paramétrage de ce composant consiste à saisir la valeur de l'inductance.



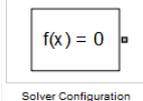
## Paramétrage

DC VOLTAGE SOURCE		Simscape/Foundation Library/Electrical/Electrical Sources
-------------------	---	---

Le paramétrage de ce composant consiste à saisir la valeur de la tension d'alimentation.



## Paramétrage

SOLVER CONFIGURATION		Simscape/Utilities
----------------------	---	--------------------

Aucune modification n'est à apporter au paramétrage de ce bloc pour ce modèle. Un bloc **Solver Configuration** doit obligatoirement être relié au modèle afin que le solveur puisse effectuer les calculs.

## Paramétrage

ELECTRICAL REFERENCE		Simscape/Foundation Library/Electrical/Electrical Element
----------------------	---	---

Aucun paramétrage n'est nécessaire pour ce composant. Un bloc référence du domaine physique utilisé doit obligatoirement être présent dans le modèle.

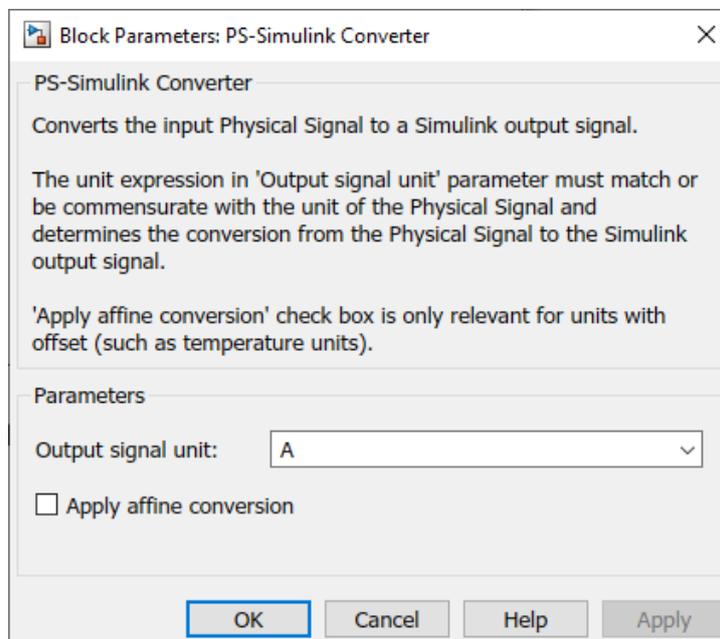
PS-SIMULINK CONVERTER



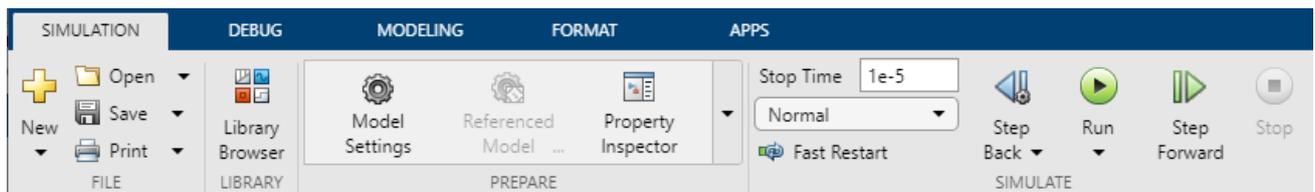
Simscape/Utilities

Ce composant permet de convertir un signal physique image d'une grandeur mesurée en signal **Simulink** afin de pouvoir être affiché dans un Scope. Il est souhaitable de préciser l'unité de la grandeur physique souhaitée pour l'affichage dans Simulink. Indiquer dans les deux blocs correspondants que l'on souhaite relever le courant en ampère (A) et la tension en volts (V). Le menu déroulant propose un choix d'unités, nous verrons qu'il est possible d'étendre les choix possibles d'unités. Si l'unité n'est pas spécifiée par l'utilisateur, l'unité SI du domaine sera choisie par défaut.

La fonctionnalité de ce bloc est très utile et permet de maîtriser les unités des grandeurs physiques relevées. Simscape gère automatiquement la cohérence des unités pour la résolution du modèle et l'utilisateur n'a pas à s'en préoccuper.



## E. Lancement de la simulation et analyse des résultats



Si nécessaire, le modèle complet est disponible dans le fichier *circuit\_RL.slx*

Spécifier un temps de simulation de **1e-5 seconde** et **lancer** la simulation en cliquant sur 

**Double cliquer** sur le Scope qui mesure le courant pour obtenir la variation du courant en fonction du temps.

**Cliquer** sur la mise à l'échelle automatique  si nécessaire pour obtenir la courbe suivante :

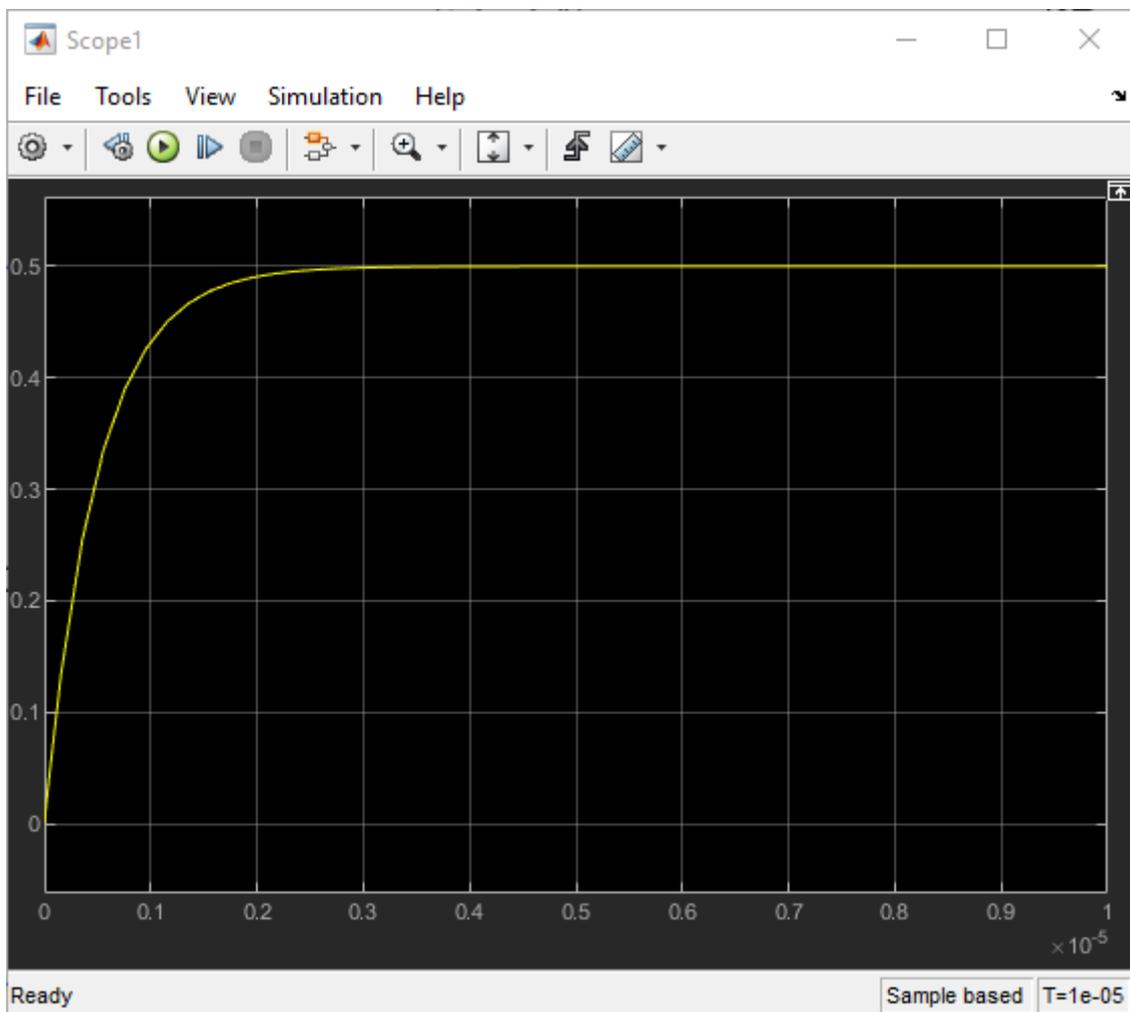


Figure 96 : évolution de l'intensité du courant dans le circuit RL

Les couleurs de mise en forme des scopes peuvent être modifiées. Cliquer avec le bouton droit de la souris

sur l'icône  dans le scope pour faire apparaître le menu déroulant de la Figure 97.

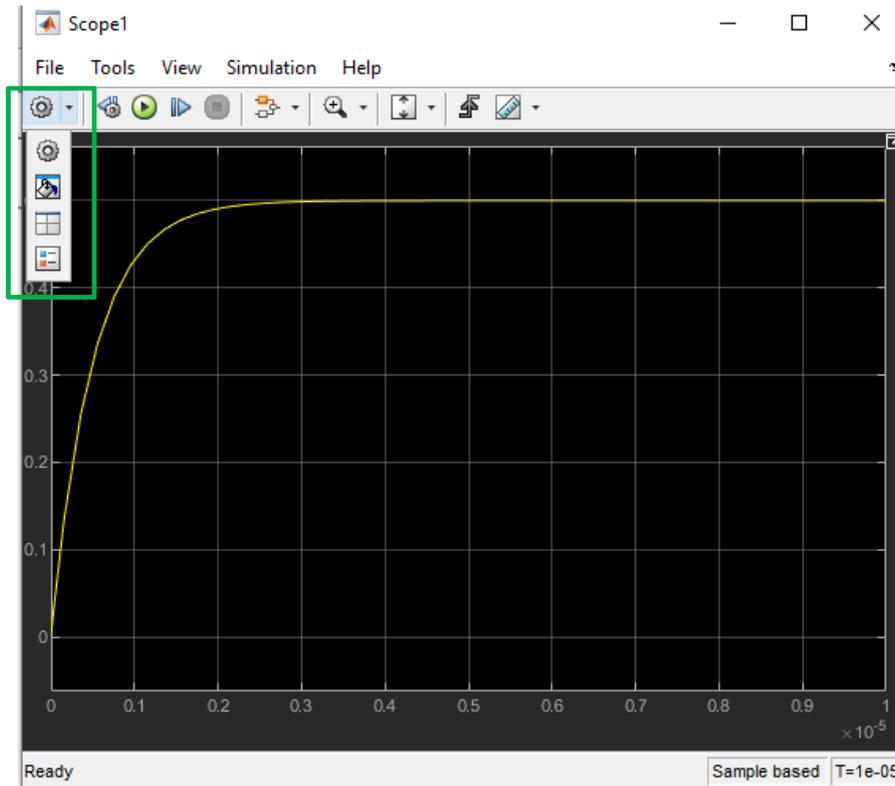


Figure 97 : mise en forme des scopes



Sélectionner l'icône et paramétrer la mise en forme conformément à la Figure 98.

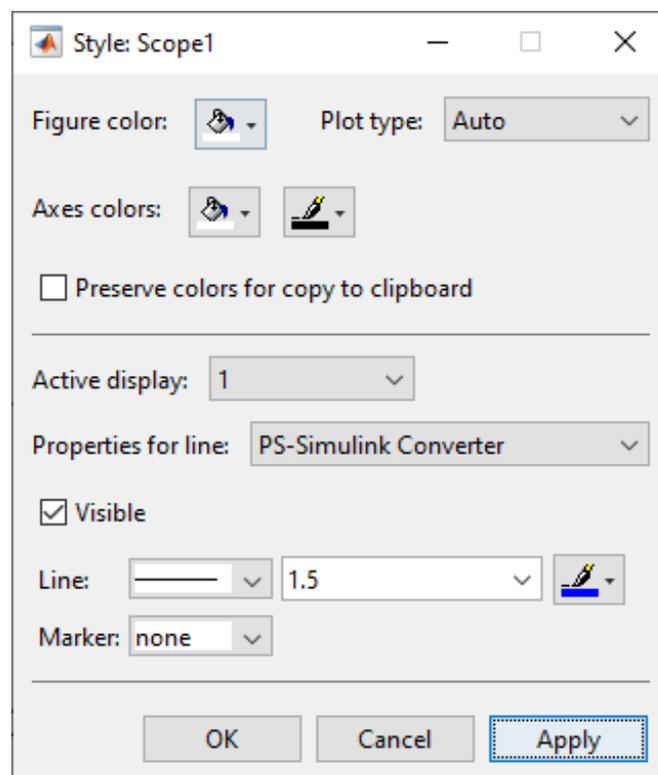
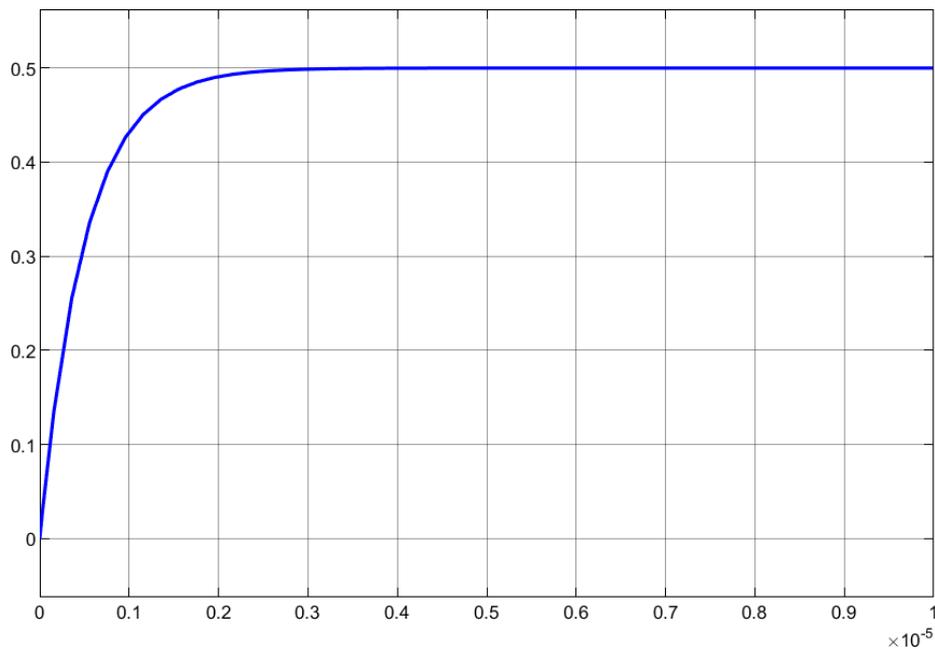


Figure 98 : modification de la mise en forme d'un scope

Vous obtenez alors le résultat de la Figure 99.

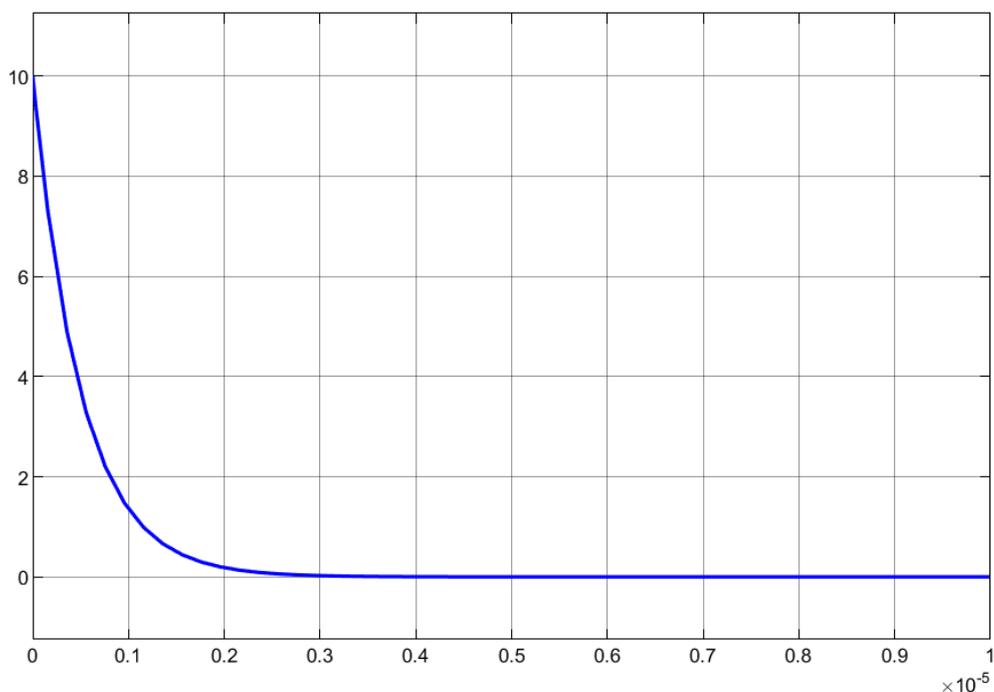


**Figure 99 : modification de l'aspect des courbes d'un scope**

Il est possible de choisir les options de mise en forme plus avancées. Pour cela vous pouvez vous reporter à l'annexe « Paramétrage des scopes » en fin d'ouvrage pour la mise en forme des courbes.

Nous procédons de manière identique pour visualiser l'évolution temporelle de la tension aux bornes de la bobine.

**Cliquer** sur la mise à l'échelle automatique  si nécessaire pour obtenir la courbe suivante :



**Figure 100 : évolution de la tension aux bornes de la bobine pour le circuit RL**

Il est possible de modifier la valeur de l'inductance de la bobine pour voir son influence sur l'établissement du courant dans le circuit en modifiant le paramètre du bloc **inductor** et en relançant la simulation.

La courbe suivante représente l'évolution du courant avec une valeur de l'inductance de  $5 \cdot 10^{-5}$  H.

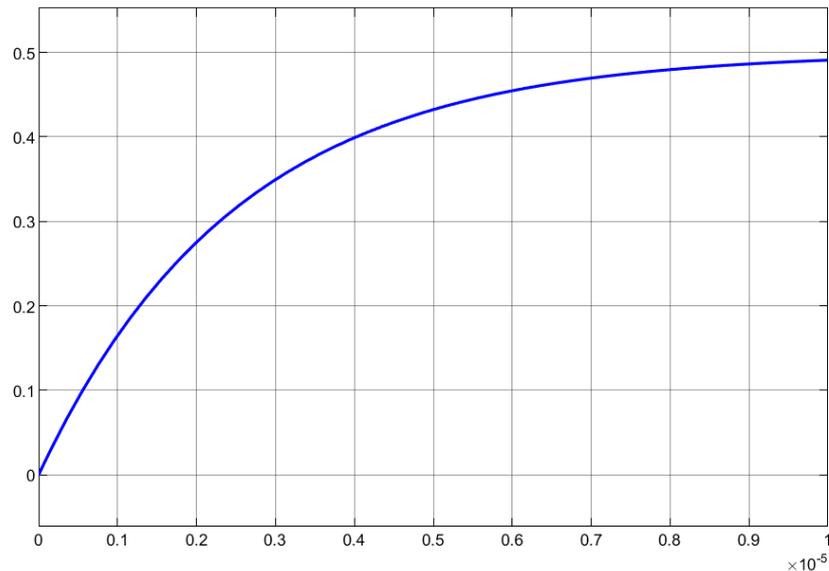


Figure 101 : évolution du courant avec une autre valeur de l'inductance

Nous constatons que cette approche de modélisation est très intuitive et permet de construire et de modifier très rapidement un modèle. Le modèle ainsi construit possède une structure très proche de celle que l'on peut observer sur le système réel ce qui lui donne une excellente lisibilité.

## F. Utilisation du Data Inspector

Le **Data Inspector** est un outil qui permet de garder en mémoire le résultat de plusieurs simulations et de comparer les résultats. Il est très simple à utiliser et nous allons voir dans cet exemple une partie de ses fonctionnalités. Pour bien utiliser cet outil, il est nécessaire de réaliser quelques opérations sur le modèle.

Dans un premier temps, il est conseillé de nommer les signaux qui arrivent dans les scopes afin de pouvoir mieux les identifier par la suite. Pour cela, il suffit de double cliquer directement sur la connexion Simulink qui arrive dans le scope et de taper le nom du signal. Pour revenir à la ligne, vous devez utiliser la combinaison de touche Maj+Entrée.

**Renommer** les signaux comme indiqué sur la Figure 102.

**Paramétrer** la valeur de l'inductance à  $1 \cdot 10^{-5}$  H comme initialement.

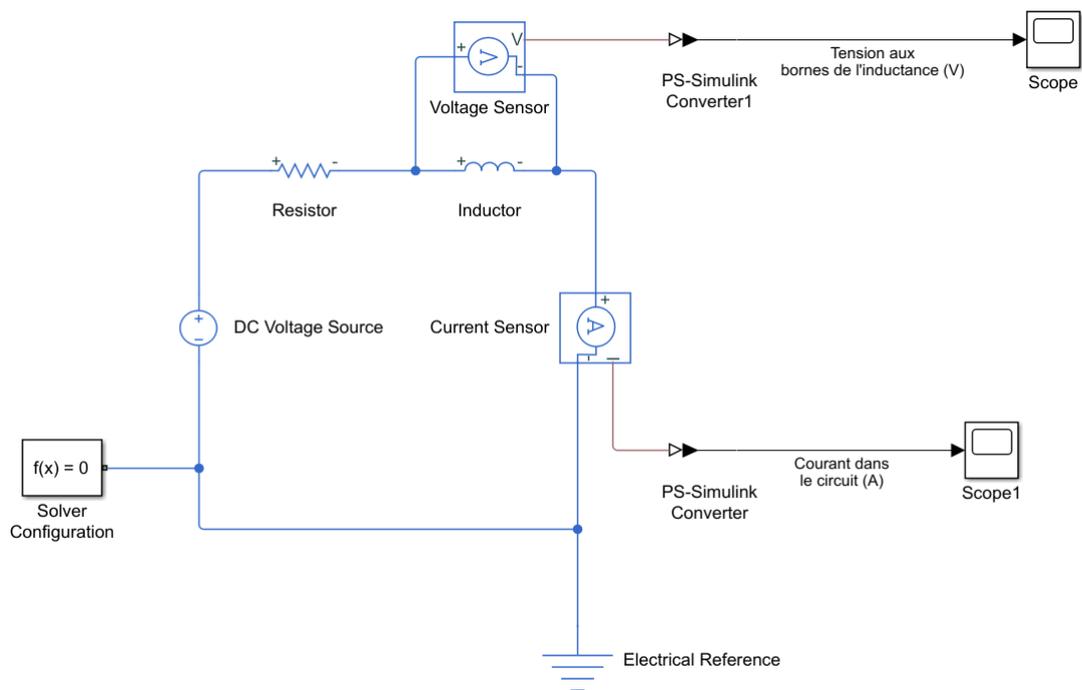


Figure 102 : définition du nom des signaux à analyser

Il faut maintenant indiquer au modèle les signaux que l'on veut visualiser à l'aide du **Data Inspector**. Dans notre exemple nous suivons les signaux « Tension aux bornes de l'inductance » et « Courant dans le circuit ».

**Ouvrir** le menu contextuel de la connexion correspondant au signal « Tension aux bornes de l'inductance » en cliquant avec le bouton droit de la souris sur cette connexion et sélectionner **Log Selected Signals** dans ce menu (Figure 103).

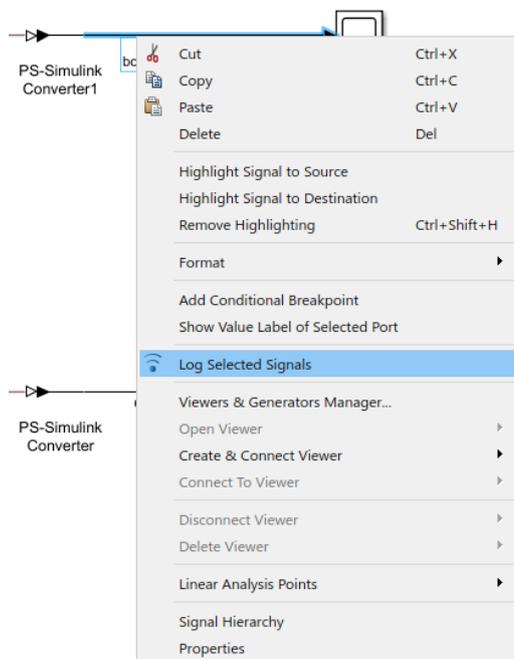


Figure 103 : sélection des signaux à suivre avec le Data Inspector

Reproduire la procédure avec le signal « Courant dans le circuit ». Vous devez obtenir le modèle de la Figure 104.

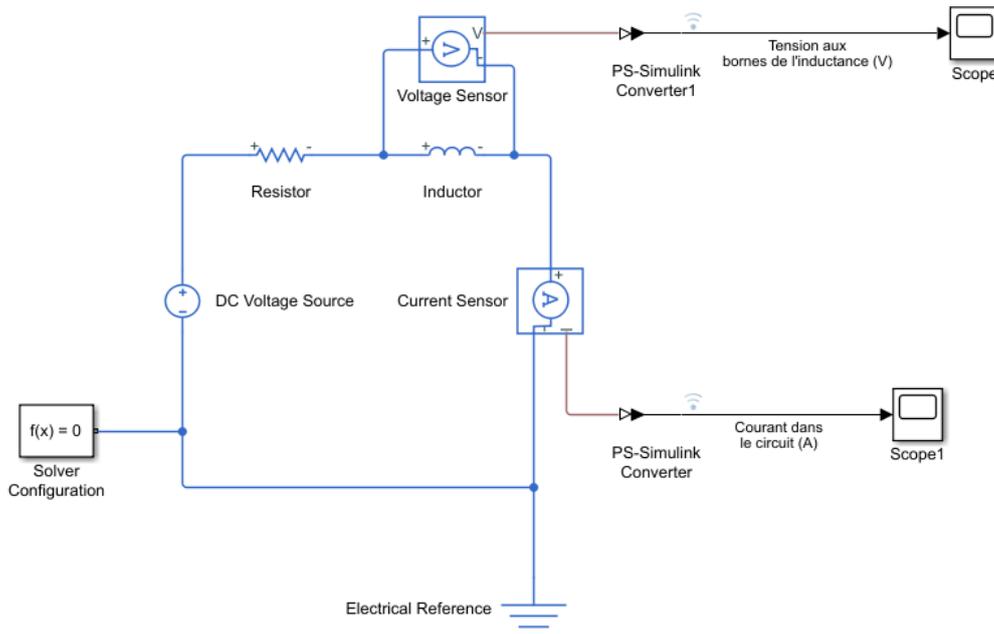


Figure 104 : visualisation des signaux à suivre avec le Data Inspector



Lancer la simulation et ouvrir le Data Inspector en cliquant sur l'icône Data Inspector dans la fenêtre de commande de Simulink. Fermer la fenêtre d'aide qui va s'ouvrir automatiquement. Vous devez voir la fenêtre du Data Inspector

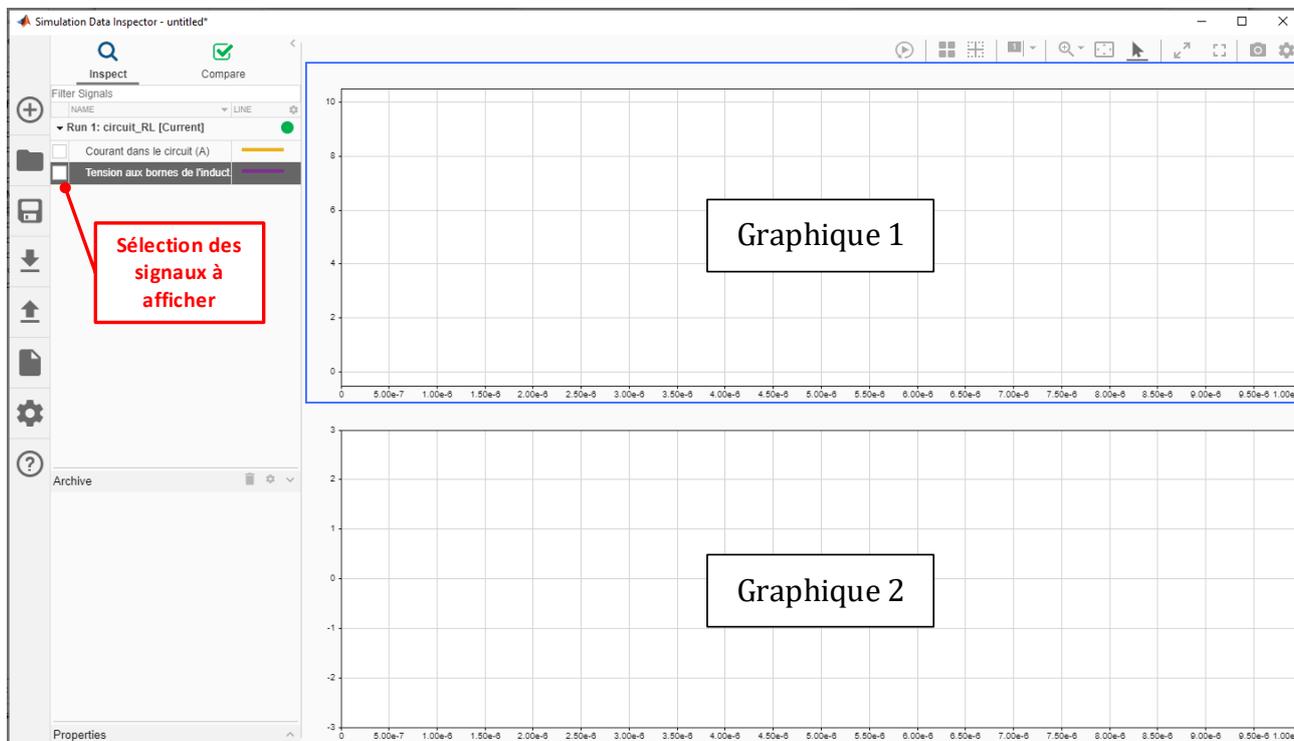


Figure 105 : fenêtre du Data Inspector

La fenêtre du Data Inspector dispose de deux fenêtres graphiques que vous pouvez sélectionner en cliquant dessus. Sur la Figure 105, la fenêtre « Graphique 1 » est sélectionnée (contour bleu). Afin de visualiser la courbe « Courant dans le circuit » dans la fenêtre « Graphique 1 », il suffit de cocher la case « Courant dans le circuit » (Figure 106).

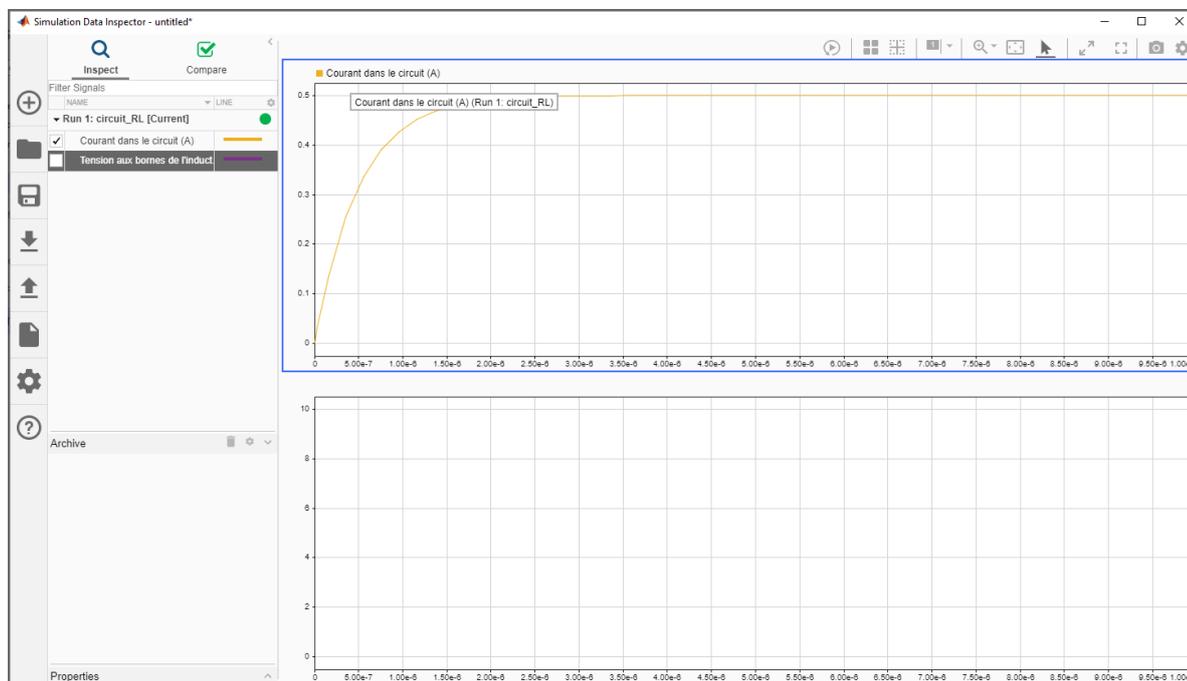


Figure 106 : visualisation d'une courbe à l'aide du Data Inspector

Sélectionner maintenant la fenêtre « Graphique 2 » et afficher le signal « Tension aux bornes de l'inductance » dans cette fenêtre (Figure 107).

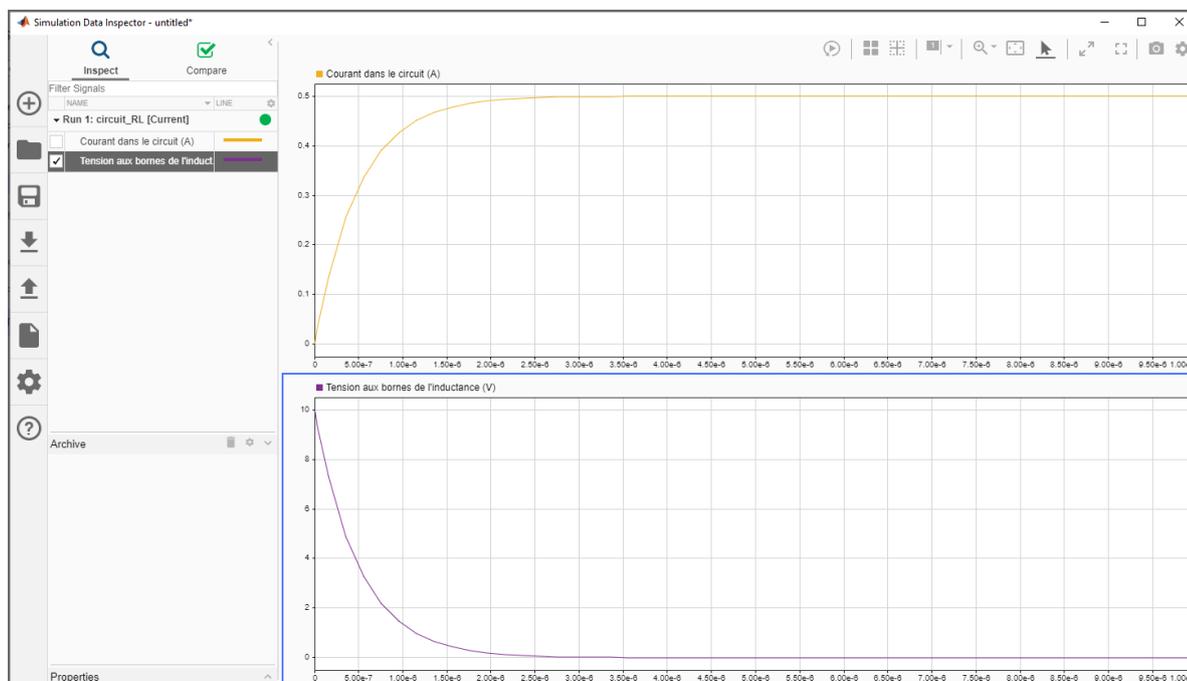


Figure 107 : visualisation de plusieurs courbes à l'aide du Data Inspector

Revenir dans le modèle et **paramétrer** la valeur de l'inductance à  $5.10^{-5}$  H. Lancer la simulation et revenir dans la fenêtre du **Data Inspector**. Le dernier Run de simulation correspondant à la dernière valeur de l'inductance s'affiche par défaut dans la fenêtre graphique (Figure 108). Le Data Inspector enregistrera les signaux marqués à chaque nouvelle simulation.

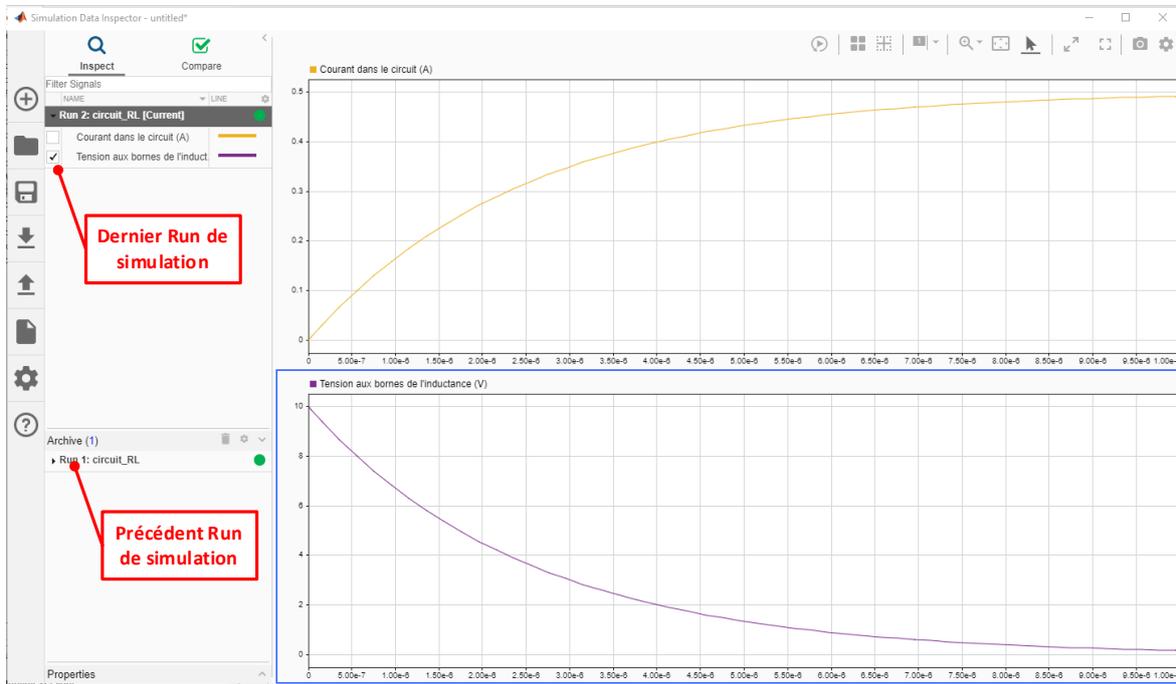


Figure 108 : visualisation des résultats de simulations différentes à l'aide du Data Inspector

Pour visualiser les courbes des deux simulations sur un même fenêtre graphique, il faut sélectionner une fenêtre graphique et cocher toutes les cases qui correspondent aux signaux que l'on veut afficher. La Figure 109 permet de comparer les courants et les tensions pour les deux simulations.

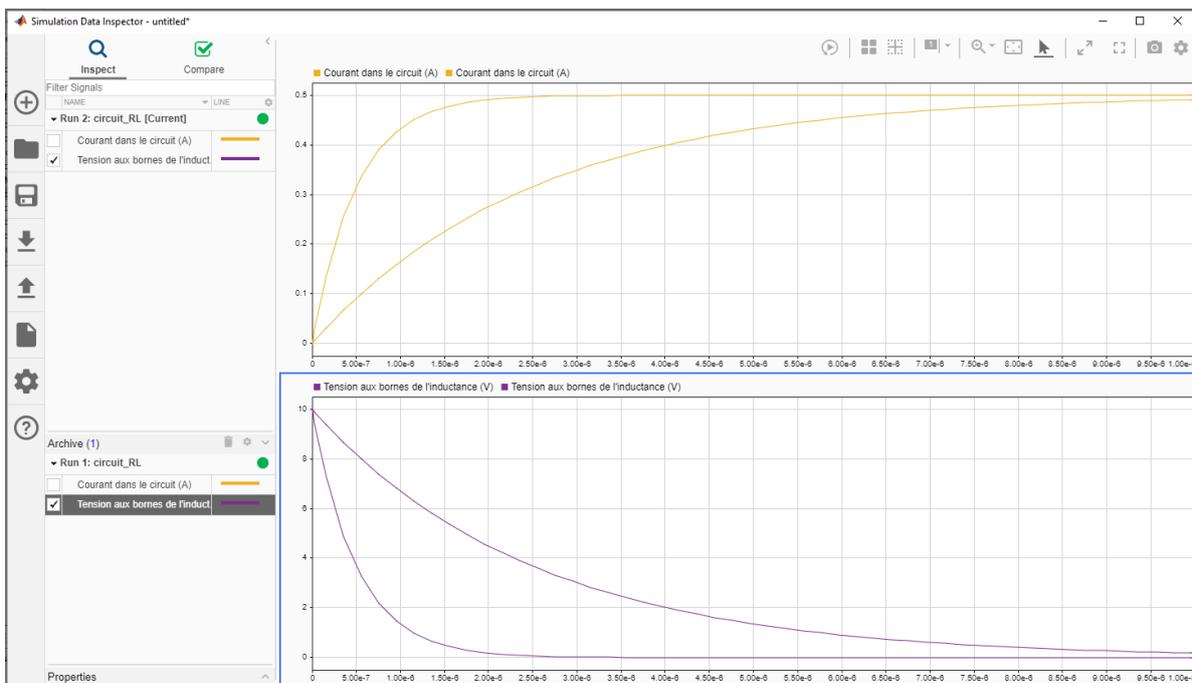


Figure 109 : affichage de plusieurs courbes de résultats à l'aide du Data Inspector

Il est possible de changer la couleur des courbes en cliquant sur la ligne de couleur à droite du signal (Figure 110 et Figure 111).

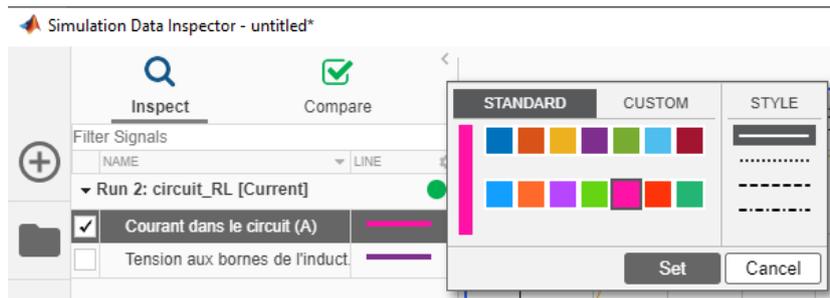


Figure 110 : modification de la couleur des courbes à l'aide du Data Inspector

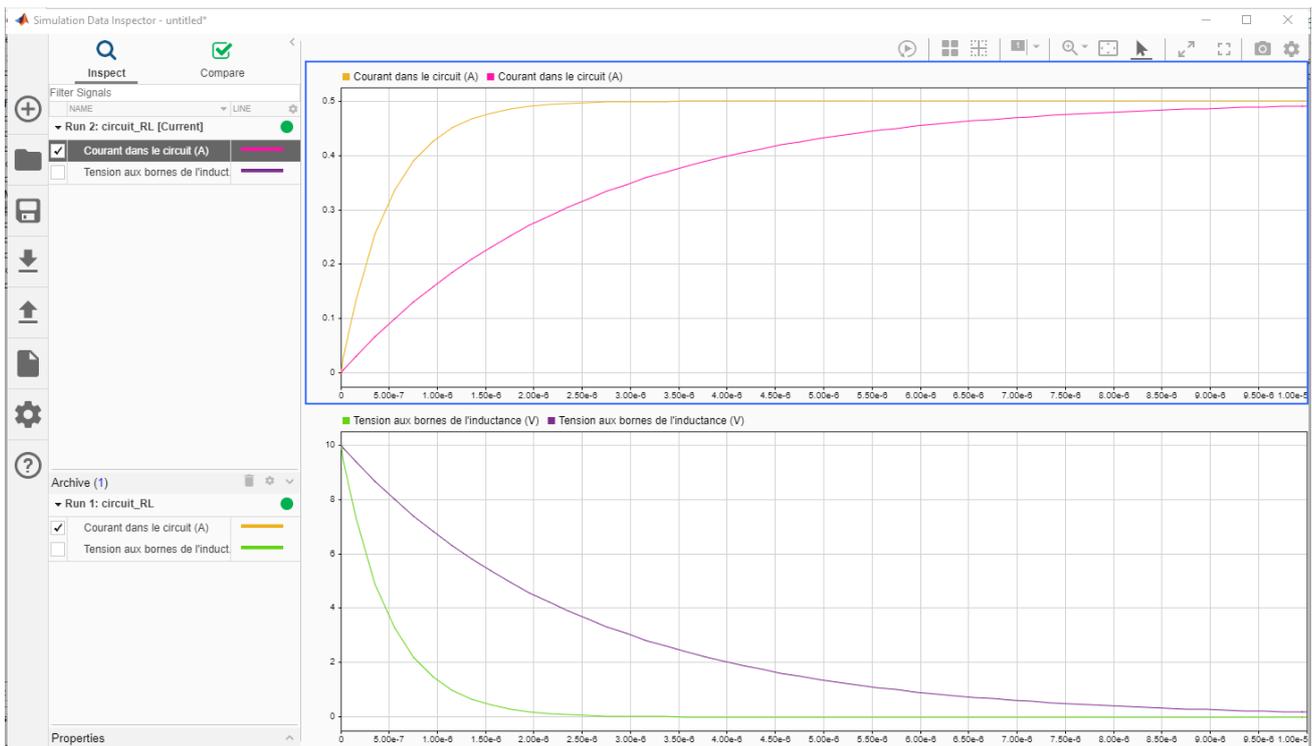


Figure 111 : visualisation de la modification de la couleur des courbes à l'aide du Data Inspector

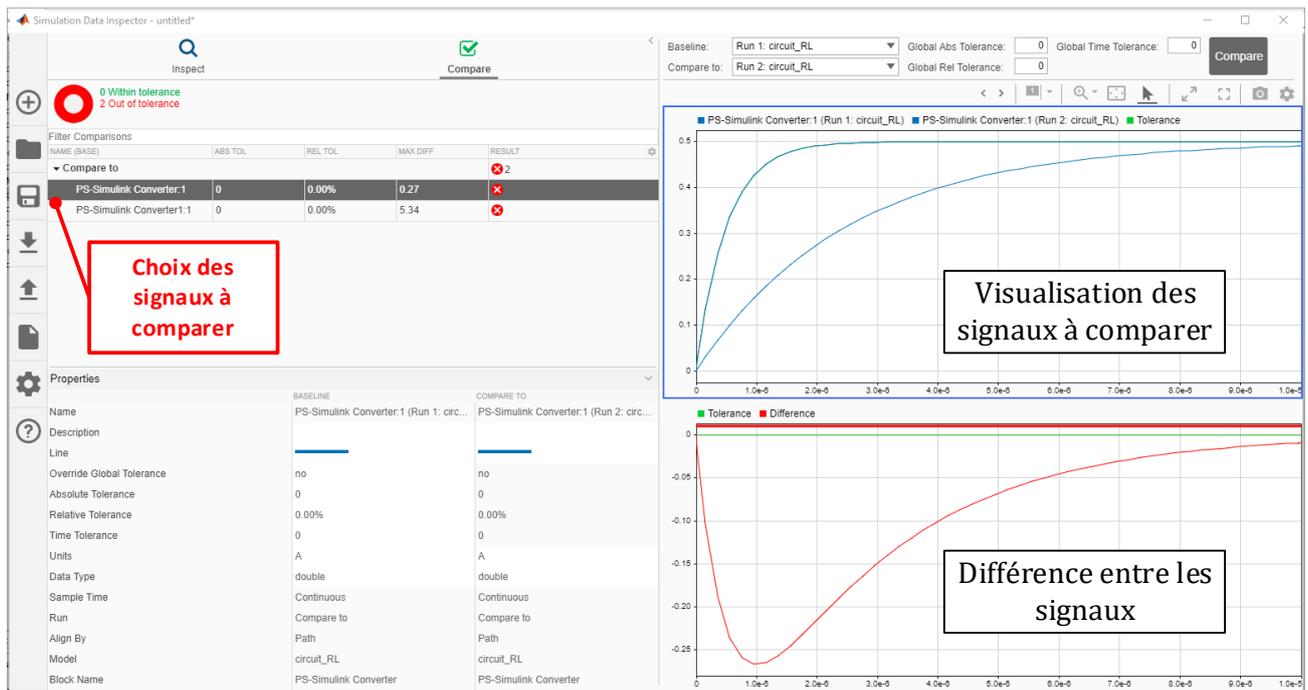
L'une des fonctionnalités intéressantes du Data Inspector est la comparaison automatique des courbes.

Dans la partie supérieure de la fenêtre du Data Inspector cliquer sur **Compare**, puis à l'aide des menus déroulants, indiquer que vous voulez comparer les résultats du Run1 avec les résultats du Run2 (Figure 112) et cliquer sur **Compare**.



Figure 112 : sélection des simulations à comparer à l'aide du Data Inspector

Vous devez obtenir le résultat de la Figure 113.



**Figure 113 : visualisation de la comparaison entre deux simulations à l'aide du Data Inspector**

Il est possible de comparer les signaux correspondant à la tension ou au courant en sélectionnant le signal dans la partie gauche de la fenêtre (Figure 113).

Il est possible de sauvegarder les résultats en cliquant sur  dans la partie gauche de la fenêtre pour les consulter ultérieurement.

Pour utiliser plus en profondeur les fonctionnalités du Data Inspector, vous pouvez consulter l'aide en cliquant sur .

## II. Comparaison avec l'approche causale

### A. Equation de comportement du système

Les tensions dans le circuit sont liées par la relation temporelle suivante :

$$U(t) = U_R(t) + U_L(t) = R i(t) + L \frac{d i(t)}{dt}$$

En appliquant la transformée de Laplace à cette équation en admettant les conditions de Heaviside :

$$U(p) = (R + Lp) I(p)$$

$$I(p) = \frac{1}{(R + Lp)} U(p)$$

L'évolution du courant sera donnée par la réponse temporelle à un échelon de tension d'une fonction de transfert du premier ordre.

### B. Choix des composants

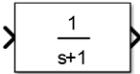
Fonction du composant	Représentation	Bibliothèque
Source échelon	 Step	Simulink/Sources
Fonction de transfert	 Transfer Fcn	Simulink/Continuous
Moniteur	 Scope	Simulink/Sinks

Figure 114 : les composants nécessaires à la modélisation du circuit R-L avec Simulink

### C. Placement et assemblage des composants

Dans la fenêtre **Simulink Library Browser**, cliquer sur  pour créer un nouveau fichier.

**Glisser/Déposer** les différents composants à partir des bibliothèques et les disposer dans la fenêtre de travail comme indiqué ci-dessous.

**Relier** les composants. Vous pouvez vous reporter à la Figure 116 pour prendre connaissances des commandes utiles pour relier et ordonner les blocs dans **Simulink**.

Le modèle obtenu est le suivant :

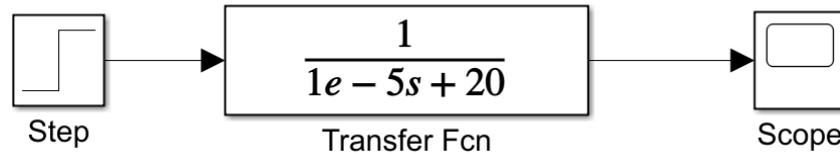


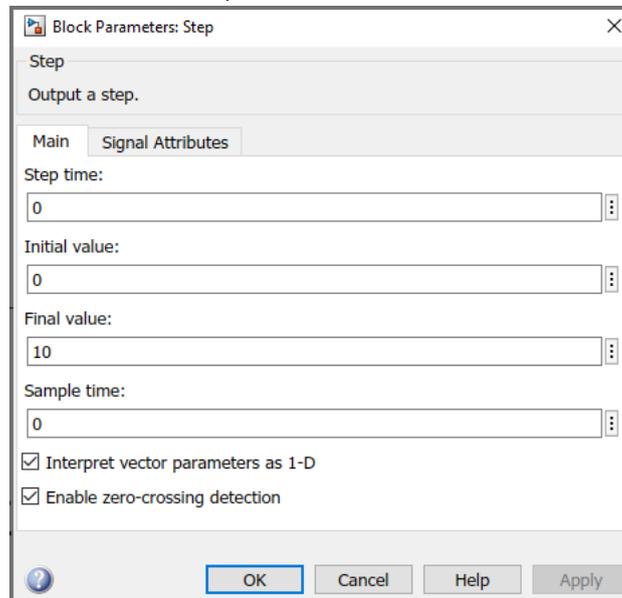
Figure 115 : modèle du circuit R-L réalisé avec Simulink

Commandes utiles	
Fonctions	Actions
Connecter deux composants	Sélectionner le premier composant. En maintenant la touche contrôle enfoncée, sélectionner le deuxième bloc. Les blocs sont reliés automatiquement
Aligner les composants	Sélectionner les composants concernés. Puis Clic droit, <b>Arrange/ Align Middle</b>
Ordonner le positionnement des composants	Sélectionner les composants concernés. Puis Clic droit, <b>Arrange</b> puis choisir l'ordonnancement désiré

Figure 116 : Commandes utiles

## D. Paramétrage des composants

Paramétrage



**Step time** : instant à partir duquel l'échelon passe à la valeur initiale, ici à  $t=0$  s.

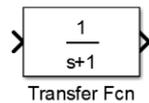
**Initial value** : valeur initiale de l'échelon, ici 0

**Final value** : valeur finale de l'échelon qui correspond ici à la tension de 10 V que l'on souhaite imposer.

**Sample time** : toujours laisser cette valeur à 0 pour l'étude des systèmes à temps continu.

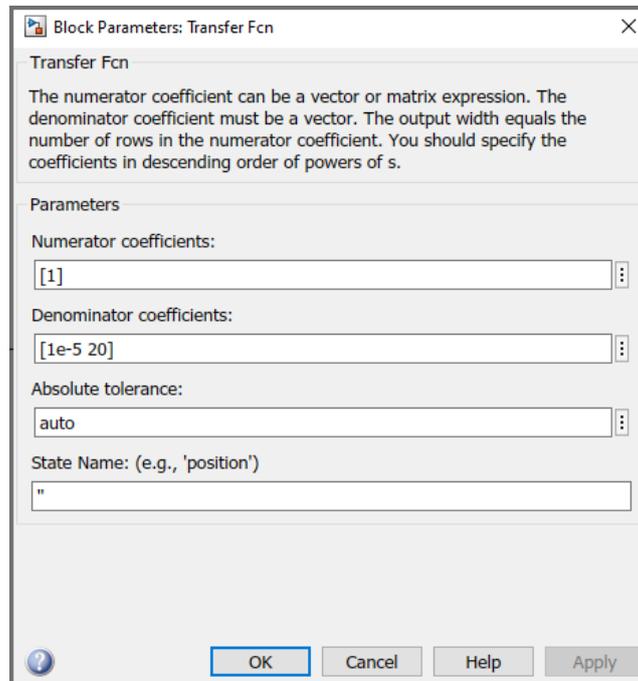
## Paramétrage

TRANSFERT FUNCTION



Simulink/Continuous

La fonction de transfert à saisir est  $H(p) = \frac{1}{R + Lp} = \frac{1}{1e^{-5}p + 20}$  avec  $R=20 \Omega$  et  $L=0.01 \text{ mH}$



Le paramétrage des fonctions de transfert se fait par la saisie des coefficients du numérateur et du dénominateur en commençant par le coefficient du monôme de plus haut degré.

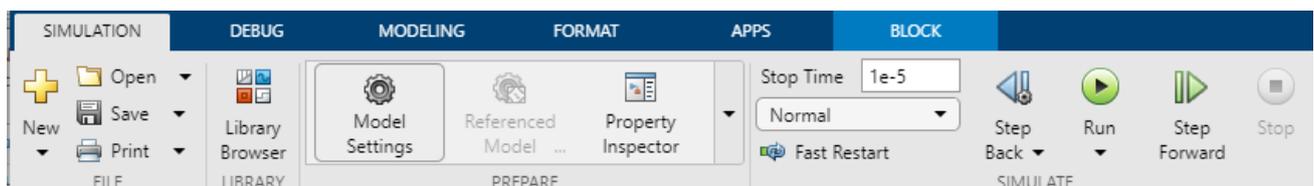
**Exemple :**

$$G(p) = \frac{3p + 1}{12p^2 + 5p + 4}$$

Numerator coefficients : [3 1]

Denominator coefficients : [12 5 4]

## E. Lancement de la simulation et analyse des résultats



**Spécifier** un temps de simulation de **1e-5 seconde** et **lancer** la simulation en cliquant sur

Faire un double-clic sur le Scope qui mesure le courant pour obtenir la variation du courant en fonction du temps.

Cliquer sur la mise à l'échelle automatique pour obtenir la courbe

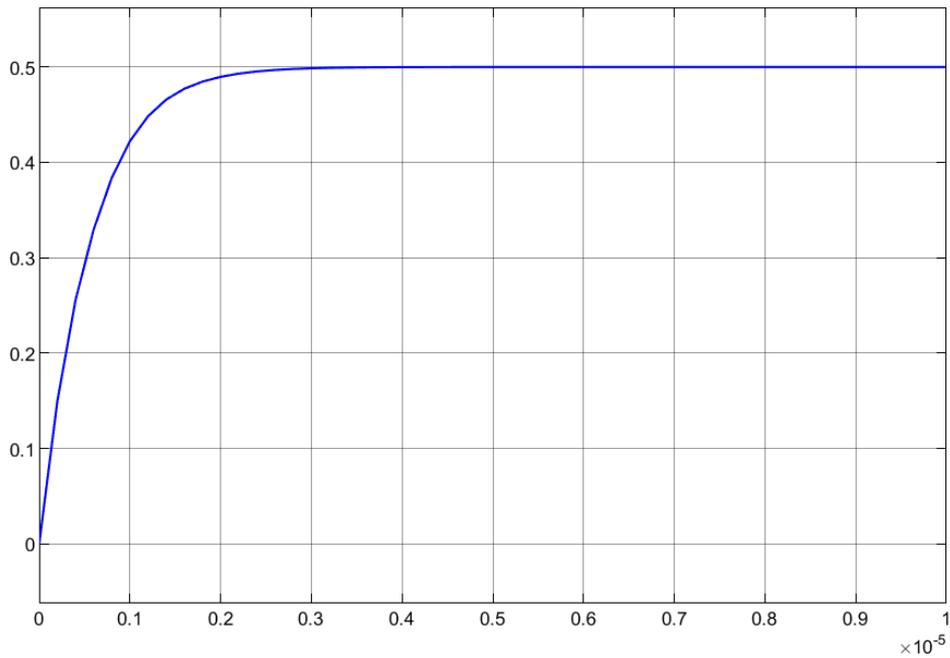


Figure 117 : évolution de l'intensité du courant dans le circuit RL

La courbe donnant l'évolution du courant dans le circuit est la même que pour le modèle acausale (heureusement !). Nous pouvons remarquer que l'évolution de la tension aux bornes de la bobine n'est pas directement accessible, une modification du modèle permettrait cependant d'y avoir accès.

## F. Avantage et inconvénients des approches causale et acausale

	Modélisation acausale	Modélisation causale
Avantages	<ul style="list-style-type: none"> <li>• Très proche de la structure du système réel</li> <li>• Grande facilité à prendre en compte des phénomènes physiques complexes</li> <li>• Facile à modifier, facile à lire</li> <li>• Plus rapide que les autres méthodes de modélisation</li> <li>• Les conditions initiales peuvent être choisies quelconques</li> <li>• Toutes les grandeurs physiques sont accessibles et mesurables au sein du modèle</li> <li>• Pas d'équation à écrire</li> <li>• Les connexions entre les composants transmettent un niveau d'information plus riche</li> <li>• Aucun outil mathématique nécessaire</li> <li>• Reconnaissance immédiate du domaine</li> <li>• Méthode de résolution numérique plus adaptée à la résolution des modèles complexes</li> </ul>	<ul style="list-style-type: none"> <li>• Parfaite maîtrise du niveau de modélisation</li> <li>• Structure du modèle plus adaptée à la mise en place d'une démarche de contrôle commande</li> <li>• Maîtrise de l'introduction des non linéarités</li> </ul>
Inconvénients	<ul style="list-style-type: none"> <li>• Peu de maîtrise concernant le niveau de modélisation des composants utilisés<sup>(1)</sup></li> <li>• Les modèles obtenus sont fortement non linéaires ce qui rend la démarche de contrôle commande délicate<sup>(2)</sup></li> <li>• Nécessite des connaissances de base dans les domaines physiques abordés</li> </ul>	<ul style="list-style-type: none"> <li>• Les grandeurs physiques sont considérées nulles à <math>t=0</math> (conditions de Heaviside)</li> <li>• Difficile à lire et à modifier</li> <li>• Nécessite une parfaite connaissance théorique des phénomènes étudiés</li> <li>• Impose l'utilisation d'outils mathématiques avancés (transformée de Laplace)</li> <li>• La modélisation des systèmes complexes entraîne de nombreux bouclages dans le modèle ce qui pose des problèmes de résolution numérique pour les solveurs.</li> </ul>

Figure 118 : avantages et inconvénients des approches causales et acausales

(1) Le niveau de modélisation du composant est imposé par le logiciel. La compréhension et l'interprétation des équations utilisées nécessitent une bonne connaissance du domaine pour être interprétées et permettre un paramétrage correct du composant. Il est important de noter que **Simscape** et ses bibliothèques proposent souvent plusieurs niveaux de modélisation pour le même type de composant.

(2) La modélisation des composants intègre par défaut de nombreuses non-linéarités qui reflètent le comportement réel du composant (seuil, saturation, hystérésis...).

Ces deux approches de modélisation coexistent dans la construction d'un modèle multi-physique. Le choix de l'une ou de l'autre des approches se fera en fonction de l'utilisation que l'on souhaite faire du modèle et du niveau de connaissance du domaine physique abordé.

### III. Les fondamentaux de la modélisation avec Simscape

#### A. Notions de domaines physiques

Au sein de l'environnement **Simscape**, plusieurs domaines physiques ou technologiques coexistent. Il est très important d'analyser et d'identifier la nature des flux d'énergies qui transitent au travers des ports d'un composant et d'identifier le ou les domaine(s) physique(s) qui interviennent dans la modélisation. Le nombre de ports de type **Physical Conserving Port (PCP)** que possède un composant est déterminé par le nombre de flux d'énergie qu'il échange avec l'extérieur. Il peut échanger des flux d'énergie dans un même domaine physique ou assurer une conversion de puissance d'un domaine physique à un autre. La conversion de puissance peut se faire avec un rendement de 100% ou tenir compte de paramètres caractérisant les pertes. Cela dépendra du niveau de modélisation du composant et des phénomènes pris en compte. Le tableau de la Figure 119 présente quelques exemples de composants et détaille les types de ports associés.

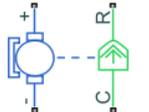
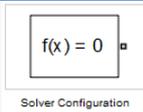
Fonction du composant	Représentation	Types de ports
Résistance	 Resistor	Ce composant possède 2 ports de type <b>PCP</b> du domaine <b>électrique</b> <u>Paramètre</u> : résistance
Réducteur	 Gear Box	Ce composant possède 2 ports de type <b>PCP</b> du domaine <b>mécanique de rotation</b> . <u>Paramètre</u> : rapport de réduction
Conversion mécanique rotation/translation	 Wheel and Axle	Ce composant possède 2 ports de type <b>PCP</b> . Port A du domaine <b>mécanique de rotation</b> et Port B du domaine <b>mécanique de translation</b> . <u>Paramètre</u> : rayon de la roue
Conversion de puissance électrique en puissance mécanique	 DC Motor	Ce composant possède 4 ports de type <b>PCP</b> . 2 port (+ et -) du domaine <b>électrique</b> et 2 ports (C et R) du domaine <b>mécanique de rotation</b> . <u>Paramètre</u> : $k_t$ en N.m/A
Source de débit hydraulique	 Hydraulic Constant Mass Flow Rate Source	Ce composant possède 2 ports de type <b>PCP</b> du domaine hydraulique. <u>Paramètre</u> : débit
Conversion de puissance mécanique en puissance hydraulique	 Variable-Displacement Pump	Ce composant possède 3 ports de type <b>PCP</b> . 1 port (S) du domaine <b>mécanique</b> et 2 ports (T et R) du domaine <b>hydraulique</b> . Le port mécanique (S) transmet un couple qui fait tourner le barillet de la pompe générant un transfert de fluide du port T au port P. <u>Paramètre</u> : cylindrée de la pompe (et d'autres paramètres annexes caractérisant le rendement de la conversion de puissance)

Figure 119 : les ports PCP des composants Simscape

## B. Les blocs importants de Simscape

Les modèles conçus avec **Simscape** doivent obligatoirement contenir certains blocs indispensables au déroulement de la simulation.

### Le bloc Solveur :

Désignation	Représentation	Bibliothèque
Solveur		Simscape/Utilities

Un bloc Solveur doit être relié à une partie du modèle **Simscape**. Si ce bloc n'est pas présent un message d'erreur s'affiche au lancement de la simulation.

### Les blocs « Référence » des domaines physiques utilisés :

Si un domaine physique est utilisé par un modèle **Simscape**, le modèle doit obligatoirement contenir une référence du domaine utilisé. Si un modèle exploite plusieurs domaines physiques, une référence de chaque domaine doit être utilisée. Si une référence est absente, un message d'erreur s'affiche au lancement de la simulation.

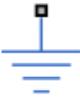
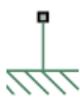
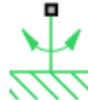
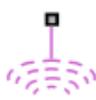
	Electrical Reference		Hydraulic Reference
	Mechanical Translational Reference		Mechanical Rotational Reference
	Pneumatic Absolute Reference		Magnetic Reference
	Pneumatic Atmospheric Reference		Thermal Reference

Figure 120 : les références des principaux domaines physiques de Simscape

## C. Variables de type « Across » et « Through » et positionnement des capteurs

Dans **Simscape**, les flux d'énergie sont définis à l'aide de deux variables, l'une de type « **Across** » (parallèle) et l'une de type « **Through** » (série). Ces variables sont appelées variables conjuguées et leur produit caractérise la puissance dans le domaine physique considéré.

- Les variables de type **Through** sont mesurées à l'aide d'un capteur placé en **série** dans le modèle (courant électrique, débit hydraulique...).
- Les variables de type **Across** sont mesurées à l'aide d'un capteur placé en **parallèle** entre deux points du modèle (différence de potentiel entre deux points d'un circuit électrique, différence de pression entre deux points d'un circuit hydraulique...).

Il est donc fondamental de connaître la nature (Across ou Through) de la grandeur physique que l'on souhaite mesurer sur une connexion du modèle dans la mesure où cette information déterminera l'implantation du capteur. Une erreur pourra entraîner un message d'erreur lors du lancement de la simulation ou un comportement erroné du modèle.

Domaine Physique	Variable(s) Across	Variable(s) Through
Electrique	Tension	Courant
Hydraulique	Pression	Débit
Mécanique de translation	Vitesse linéaire	Force
Mécanique de rotation	Vitesse angulaire	Moment
Pneumatique	Pression et temperature	Débit massique et flux d'entropie
Thermique	Température	Flux thermique et flux d'entropie

Figure 121 : les variables Across et through de Simscape

## D. L'orientation des composants

Dans **Simscape**, les composants ont une convention d'orientation. Le sens positif d'un composant est toujours défini d'un port vers l'autre. Cette information est donnée dans la fenêtre de paramétrage du composant.

Il faudra prendre en compte la polarité des composants dans les cas suivants :

- Utilisation de composants actifs
- Implantation des capteurs
- Utilisation de composants dont la dynamique est orientée

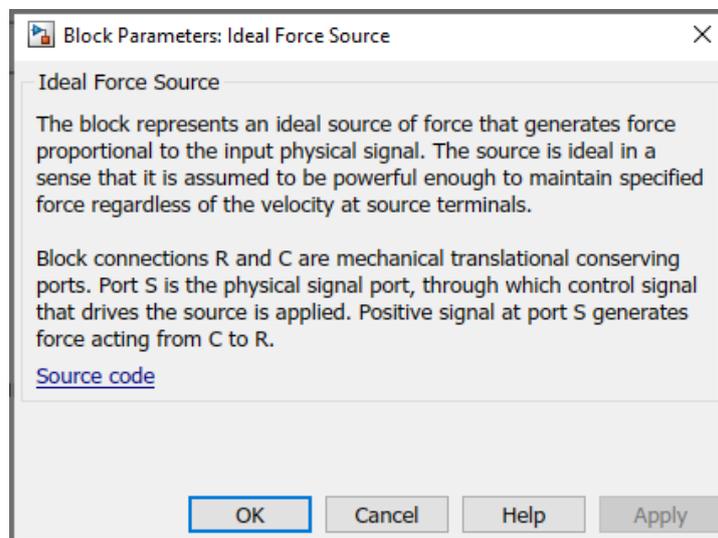
## 1. Utilisation de composants actifs

Lors de l'utilisation des composants actifs (sources de vitesse, de force, de débit, de pression...), il faut tenir compte de l'orientation du composant.

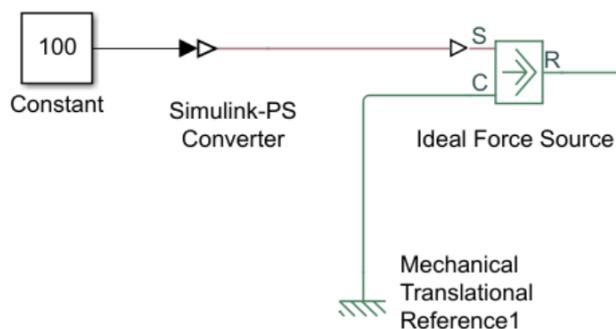
### Exemple de l'utilisation d'une source d'effort.

Désignation	Représentation	Bibliothèque
Source d'effort	 Ideal Force Source	Simscape/Foundation Library/Mechanical/Mechanical Sources

Fenêtre de paramétrage:

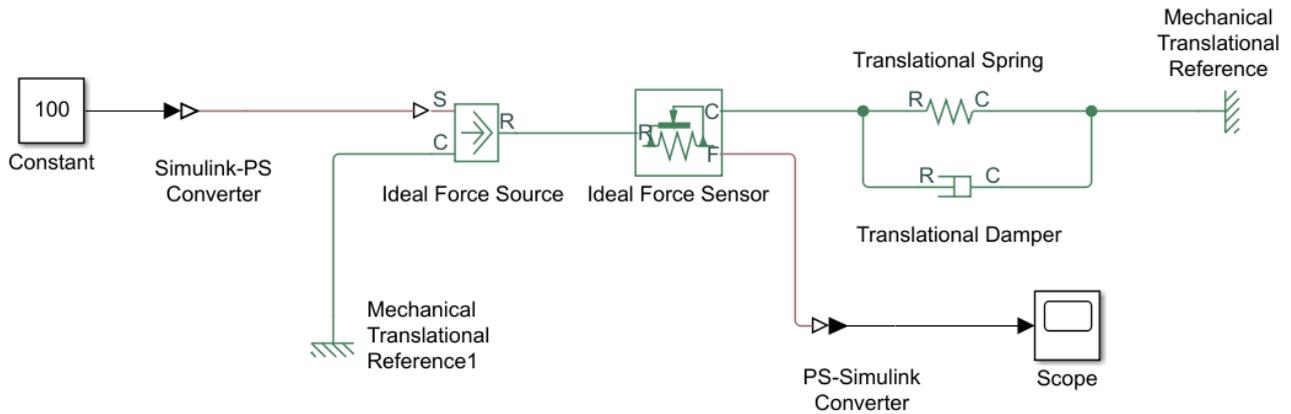


### Source d'effort



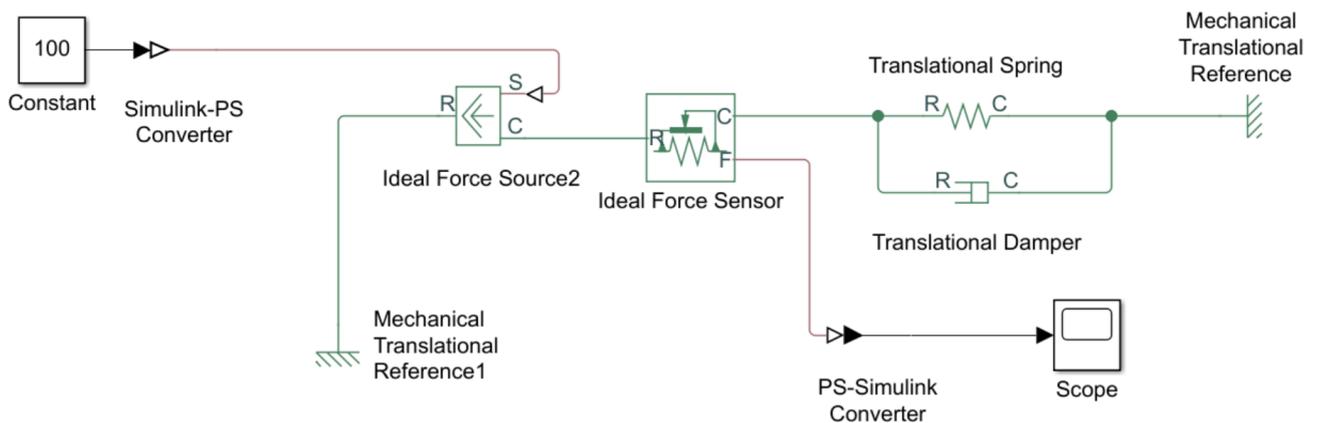
Ce modèle permet de simuler une source de force de 100 N.  
Le paramétrage du bloc nous indique qu'une valeur positive du signal qui arrive sur le port S génère un effort qui s'exerce positivement de C vers R.

Sur le modèle de la Figure 122, une force de 100 N, dirigée de C vers R va agir sur l'ensemble ressort amortisseur.



**Figure 122 : source de force orientée positivement**

Sur le modèle de la Figure 123, les ports C et R ont été inversés, et la force va s'appliquer à l'opposé de la configuration de la Figure 122.



**Figure 123 : source de force orientée négativement**

Ce raisonnement peut être généralisé à tout type de source d'énergie.

Ouvrir le fichier « orientation\_source\_force.slx ».

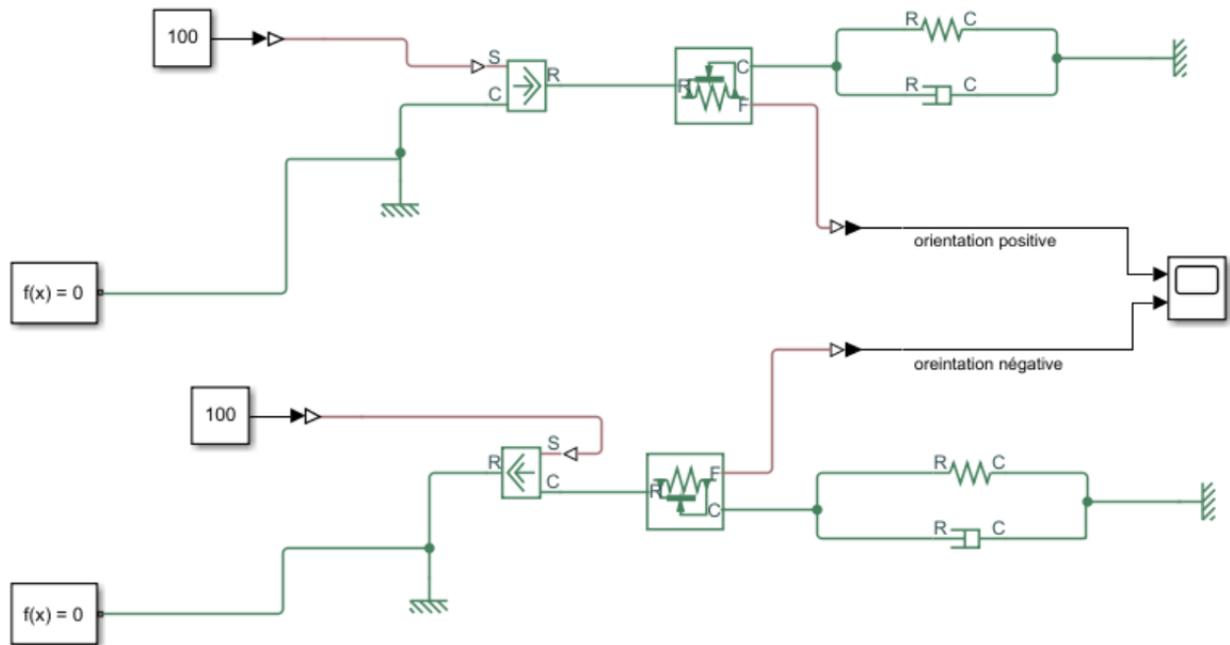


Figure 124 : orientation d'une source de force

Ce fichier contient deux modélisations identiques d'une source de force qui agit sur un ensemble ressort+amortisseur. Un capteur d'effort placé en série mesure l'effort (variable de type Through) exercé par la source d'effort sur l'ensemble ressort+amortisseur. Dans le modèle correspondant au bas de la Figure 124, l'orientation de la source de force a été inversée.

**Lancer la simulation** et observer la mesure de l'effort à l'aide du scope pour les deux modélisations.

Cliquer sur la mise à l'échelle automatique du scope si nécessaire 

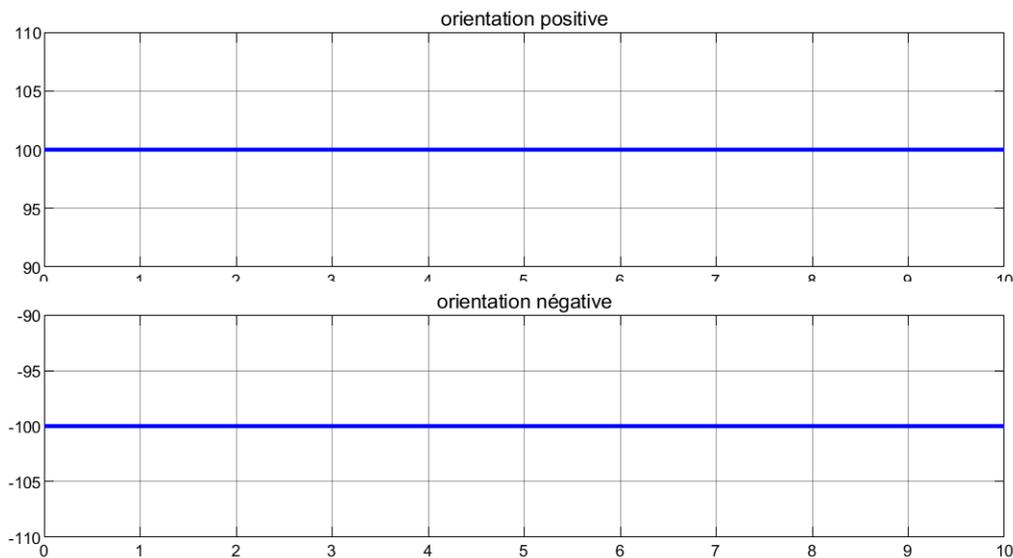


Figure 125 : visualisation de l'influence de l'orientation de la source de force

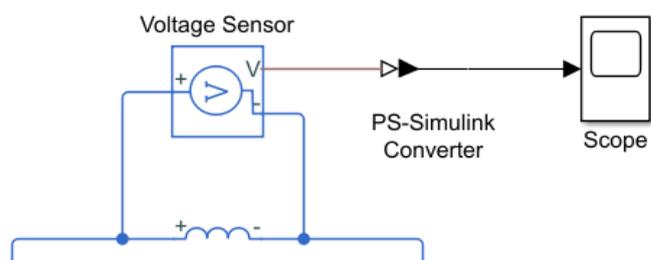
On observe que l'orientation de la source a une influence directe sur le sens de l'effort exercé sur l'ensemble ressort+amortisseur.

## 2. Implantation et orientation des capteurs

Lors de l'implantation d'un capteur. Le signe de la mesure dépend de l'orientation du capteur.

**Exemple de mesure de grandeurs électriques :**

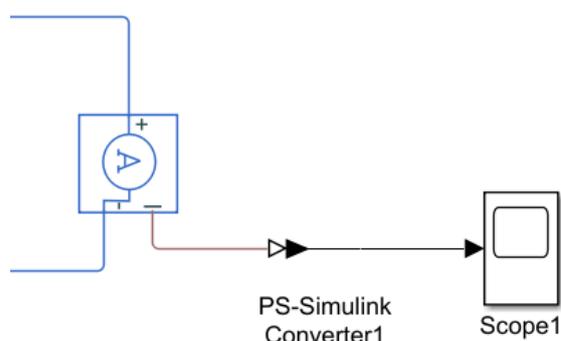
### Mesure d'une tension (variable Across)



Le capteur est placé en parallèle.

Le résultat de la mesure est  $V^+ - V^-$  et sera positif si le potentiel relié à la borne + du capteur est supérieur au potentiel relié à la borne - du capteur. Si on inverse les bornes du capteur, le résultat de la mesure sera de signe contraire.

### Mesure d'un courant (variable Through)



Le capteur est placé en série.

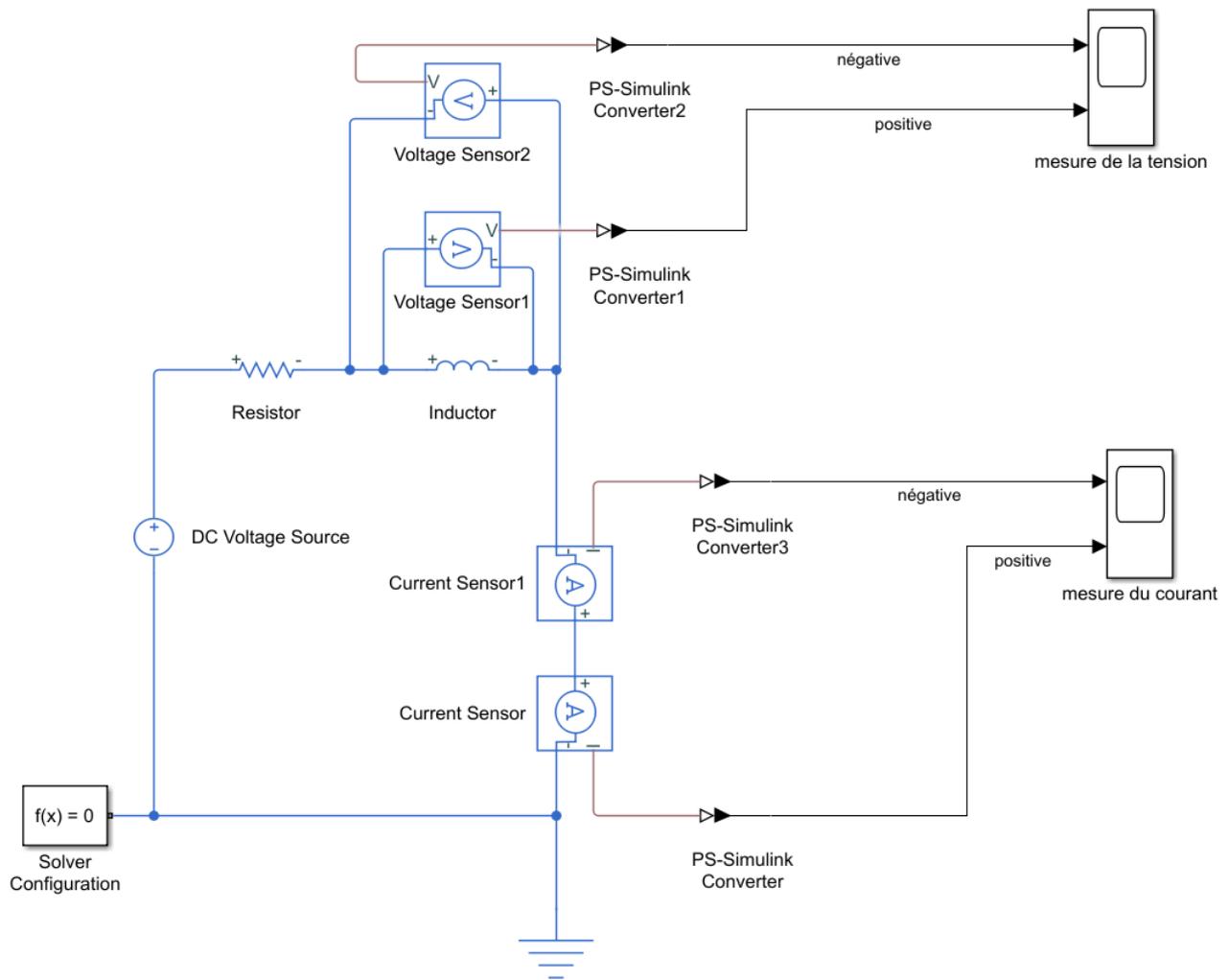
Le résultat de la mesure sera positif si le courant circule de la borne + vers la borne - du capteur. Si on inverse les bornes du capteur, le résultat de la mesure sera de signe contraire.

Figure 126 : exemple de mesure de grandeurs électriques

Ce raisonnement peut être généralisé à tout type de capteur et tout type de variable de type « Across » et « Through ».

**Ouvrir** le fichier « **implantation\_des\_capteurs.slx** »

Ce fichier contient la modélisation d'un circuit RL. La tension aux bornes de la bobine et le courant qui traverse le circuit sont mesurés à l'aide de deux capteurs dont la polarité a été inversée.



**Figure 127 : implantation des capteurs**

**Lancer** la simulation et observer à l'aide des scopes l'influence de l'orientation des capteurs sur le signe de la mesure obtenue.

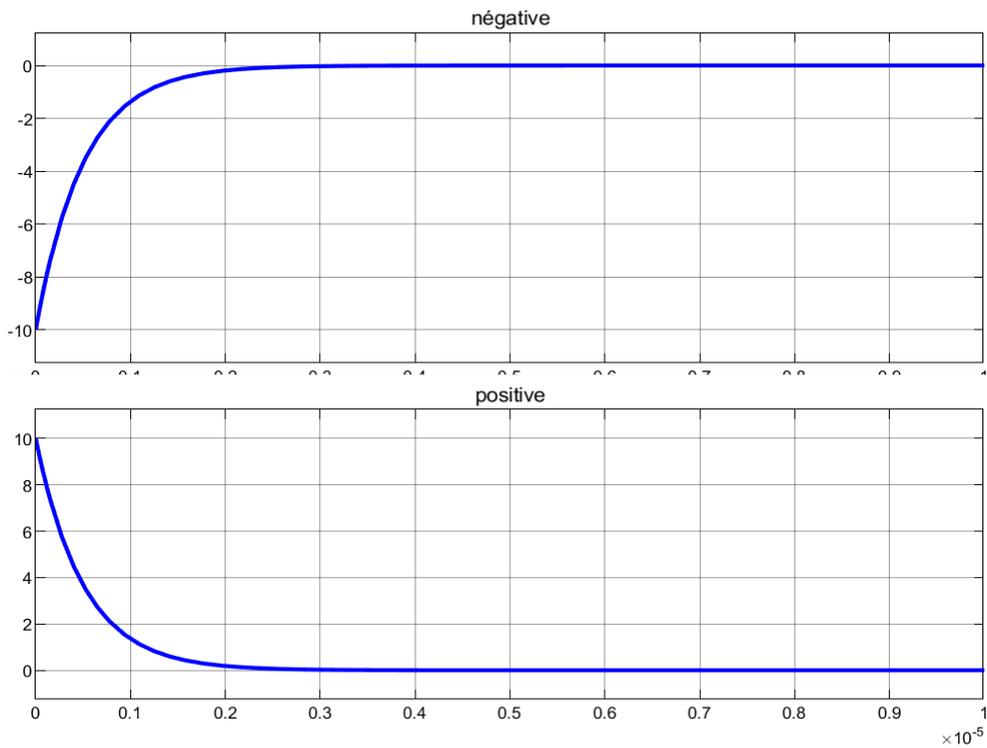


Figure 128 : visualisation de la tension aux bornes de la bobine pour les deux orientations des capteurs

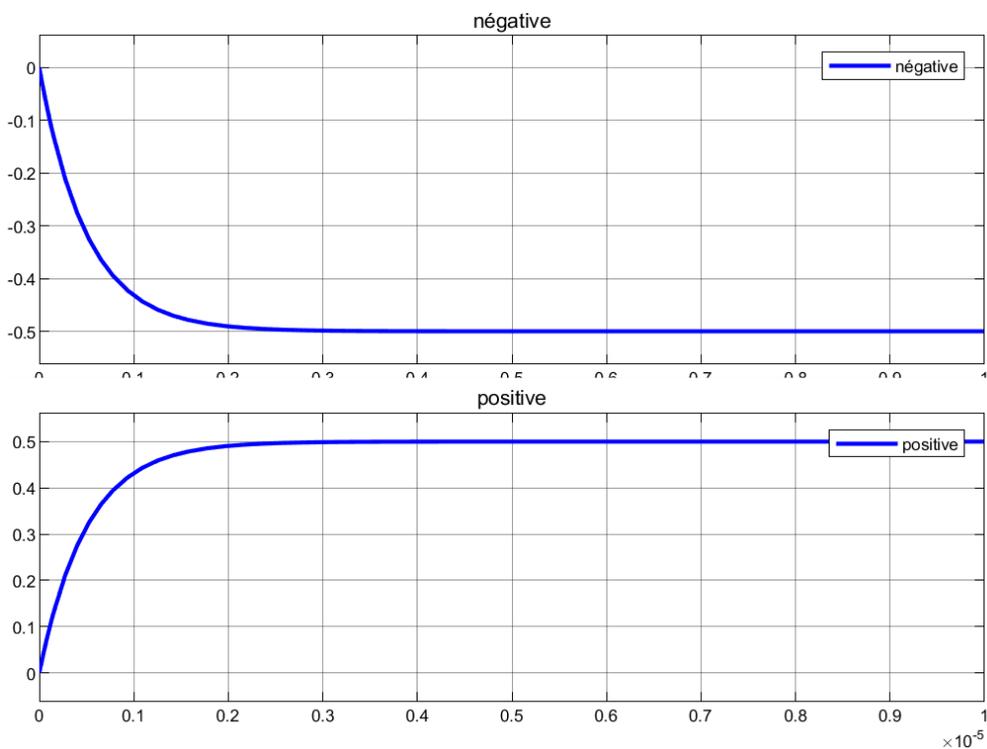


Figure 129 : visualisation de l'intensité du courant dans le circuit en fonction des deux orientations des capteurs

Les résultats obtenus mettent en évidence l'influence de l'orientation des capteurs sur le signe de la mesure obtenue.

### 3. Utilisation de composants dont la dynamique est orientée

Si on utilise un composant dont la dynamique est orientée (diode, réducteur de vitesse...), le composant doit être orienté de la même manière que sur le système réel afin d'obtenir un résultat cohérent avec le système que l'on souhaite modéliser. Si l'orientation est mauvaise, aucun message d'erreur ne permettra de détecter cette erreur et le système modélisé ne sera plus conforme au système réel.

### 4. Utilisation de composants passifs

Lors de l'utilisation de composants passifs (résistance, ressort, amortisseurs, canalisation...), dont la dynamique n'est pas orientée, l'orientation du composant n'a pas d'influence sur le comportement du modèle.

## 5. Choix du solveur

Le choix du solveur permet de définir les méthodes de résolution numériques qui seront mise en œuvre pour que le logiciel puisse mener la résolution numérique du modèle dans les meilleures conditions. Lorsque les modèles **Simscape** deviennent complexes, il est préférable de choisir le solveur le mieux adapté à la résolution de ce type de problème.

Pour cela dans la fenêtre de votre modèle **cliquer** sur **Simulation/Model Configuration Parameters**

ou sur l'icône  pour afficher la fenêtre de paramétrage du solveur (Figure 130). Développer la fenêtre **Solver details**.

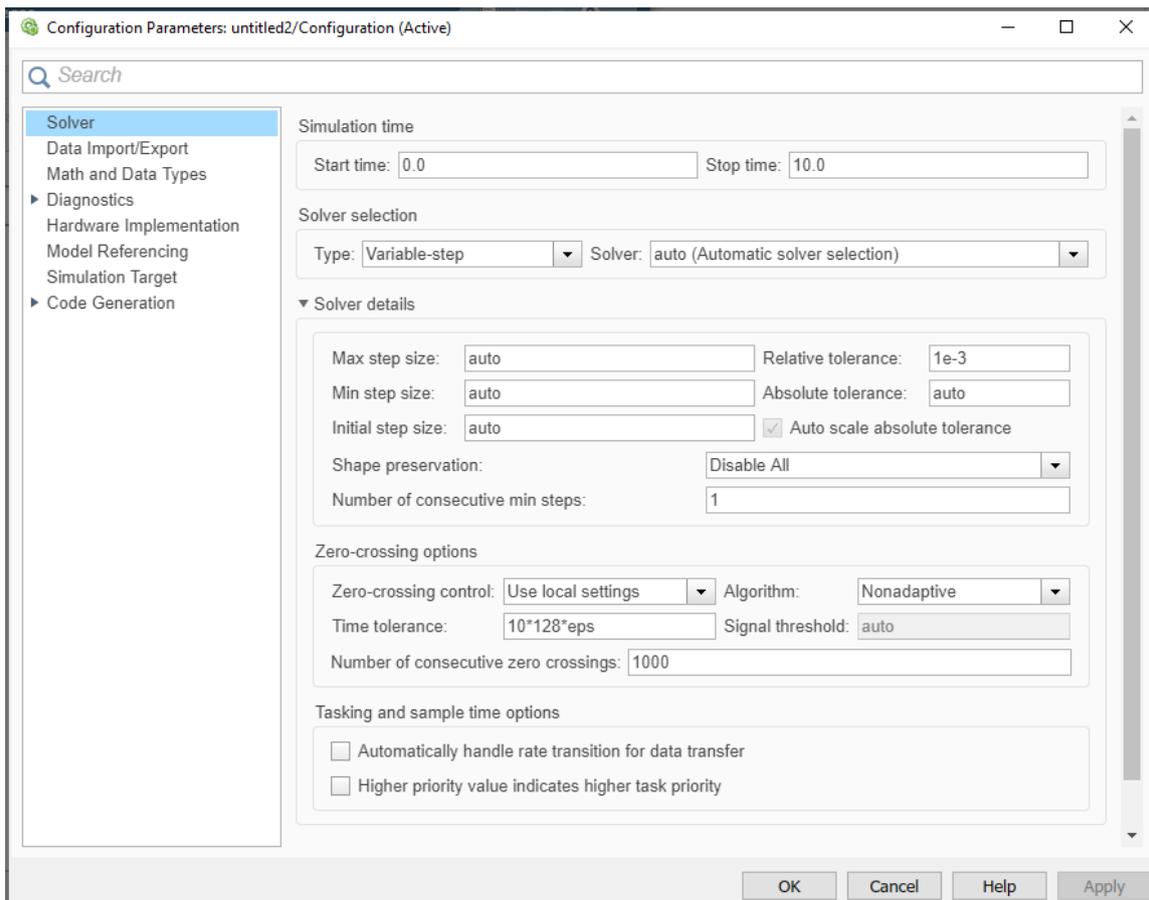


Figure 130 : fenêtre de paramétrage du solveur

Par défaut le choix du solveur est automatique. Dans le cas général cette option permet de choisir le solveur le plus adapté.

Les deux solveurs **ode15s** ou **ode23t** sont les mieux adaptés à la résolution des modèles **Simscape** et multiphysiques. Il est fortement conseillé de ne travailler qu'avec ces deux solveurs pour les modèles **Simscape**. Dans le cas général, si le choix automatique du solveur ne donne pas de résultats satisfaisants, on pourra choisir le solveur **ode15s**. Si ce solveur ne parvient toujours pas à résoudre le modèle ou nécessite un temps de calcul trop long, on utilisera **ode23t**.

Il est également possible dans cette fenêtre de spécifier un pas de calcul minimum et/ou maximum pour choisir le nombre de points et la résolution du calcul. On renseignera alors les champs **Max step size** et **Min step size**.

## 6. Les problèmes que peut rencontrer le solveur

Lors du lancement de la simulation, le solveur va résoudre numériquement le modèle afin de déterminer l'évolution en fonction du temps de toutes les variables. Ce processus mathématique très complexe sera géré automatiquement par le logiciel. Cependant des problèmes peuvent entraîner un arrêt du processus de calcul entraînant un message d'erreur indiquant que le solveur ne parvient pas au terme de la résolution.

Les solveurs utilisés par MATLAB étant extrêmement robustes, dans la majorité des cas les problèmes viennent d'une erreur dans la modélisation. Il faudra reprendre le modèle et corriger les éventuelles erreurs. Lors de la réalisation d'un modèle complexe, il faudra simuler séparément chaque partie du modèle pour identifier les parties du modèle qui provoquent les erreurs.

Voici quelques pistes pour permettre de résoudre les problèmes les plus courants :

- Il manque le bloc « Solveur » ;
- Il manque une référence d'un domaine physique utilisé ;
- Il manque des composants ou les composants ne sont pas paramétrés avec des valeurs cohérentes. Les grandeurs calculées par le solveur n'ont plus de sens physique (vitesse infinie, pression infinie, température inférieure au zéro absolu...). Dans ce cas le solveur arrêtera le calcul en indiquant les problèmes qu'il rencontre. Il faudra revoir le paramétrage ou le positionnement des composants.

## IV. Exemples de modélisation multi-domaine

---

Cette partie propose, au travers d'exemples, un aperçu des domaines physiques les plus courants qui peuvent être abordés avec **Simscape**.

Chaque exemple propose de construire un modèle et d'exploiter les résultats de la simulation. Au fur et à mesure de la construction des modèles, des notions importantes sont introduites permettant d'acquérir les compétences nécessaires à la conception de modèle multi-physiques avec **Simscape**. Il est donc conseillé de parcourir l'ensemble des exemples proposés dans cette partie.

### A. Domaine électromécanique – Axe linéaire

---

Le système modélisé est un axe linéaire. L'actionneur est un motoréducteur à courant continu commandé par une source de tension variable. Le mouvement de rotation est transformé en mouvement de translation par un système poulies/courroie. L'objectif est d'évaluer les réponses en vitesse et en position de l'axe pour estimer le temps de réponse et la rapidité ou tout autre type de performances. Il sera également possible de mesurer des paramètres électriques comme le courant moteur.

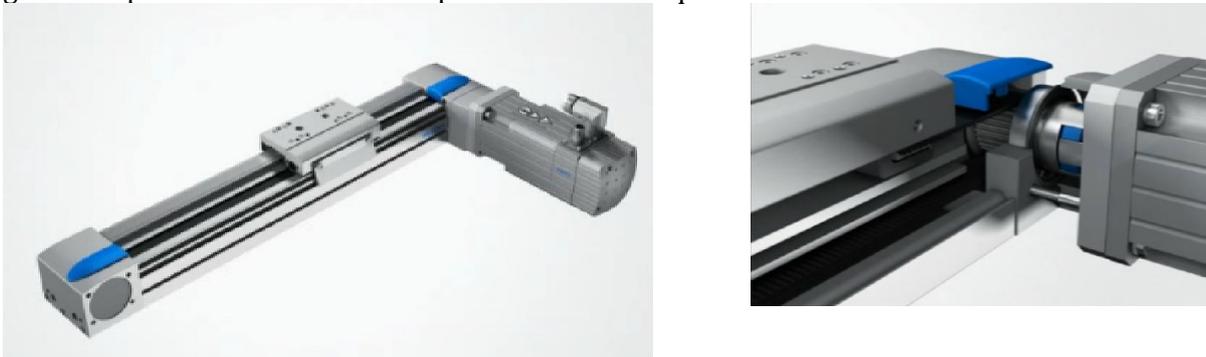


Figure 131 : photo de l'axe linéaire

Le système étudié fait intervenir 3 domaines physiques :

- Domaine électrique
- Domaine mécanique de rotation
- Domaine mécanique de translation

Certains composants appartiennent à un domaine physique particulier et d'autres composants permettront de faire la conversion de l'un à l'autre des domaines physiques. Ce sera le cas du composant « **DC Motor** » qui fera la conversion du domaine électrique au domaine mécanique de rotation et du composant « **Wheel and Axle** » qui fera la conversion du domaine mécanique de rotation au domaine mécanique de translation.

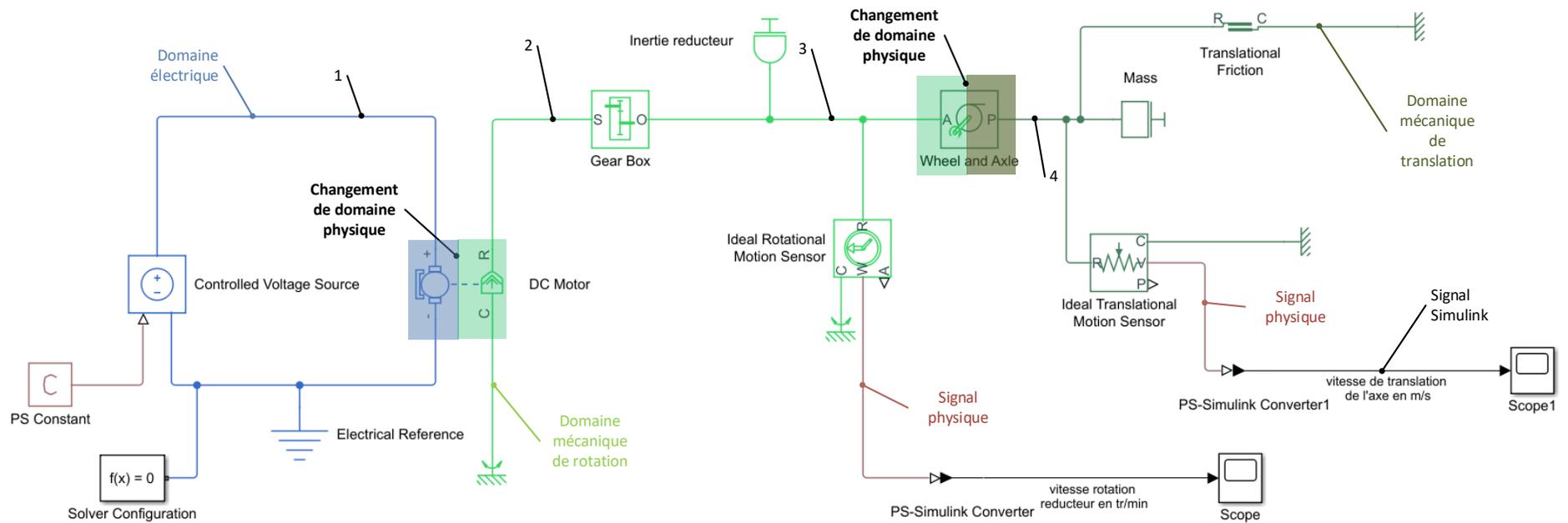
## 1. Choix des composants

Seuls sont référencés dans le tableau de la Figure 132, les composants qui n'ont pas encore été utilisés auparavant dans le document.

Fonction du composant	Représentation	Bibliothèque
Source de tension variable	 Controlled Voltage Source	Simscape/Foundation Library/Electrical/Electrical Sources
Source de signal physique	 PS Constant	Simscape/Foundation Library/Physical Signals/Sources
Moteur à courant continu	 DC Motor	Simscape/Electrical/Electromechanical/Brushed Motors
Réducteur	 Gear Box	Simscape/Foundation Library/Mechanical/Mechanisms
Inertie	 Inertia	Simscape/Foundation Library/Mechanical/Rotational Elements
Conversion mouvement de rotation en mouvement de translation	 Wheel and Axle	Simscape/Foundation Library/Mechanical/Mechanisms
Masse	 Mass	Simscape/Foundation Library/Mechanical/Translational Elements
Frottement sec et visqueux en translation	 Translational Friction	Simscape/Foundation Library/Mechanical/Translational Elements
Capteur de position ou de vitesse de translation	 Ideal Translational Motion Sensor	Simscape/Foundation Library/Mechanical/Mechanical Sensors
Capteur d'angle ou de vitesse de rotation	 Ideal Rotational Motion Sensor	Simscape/Foundation Library/Mechanical/ Mechanical Sensors

**Figure 132 : les composants nécessaires à la modélisation de l'axe linéaire avec Simscape**

Le modèle **Simscape** de l'axe linéaire est présenté Figure 133 et permet de visualiser les différents domaines physiques qui interviennent dans la modélisation ainsi que les composants qui réalisent les conversions entre les domaines physiques. Une correspondance est donnée entre les connexions des différents domaines physiques et les éléments réels associés.



**Correspondance physique des connexions:**  
 1: fil électrique (domaine électrique)  
 2: arbre sortie moteur (domaine mécanique de rotation)  
 3: arbre sortie réducteur (domaine mécanique de rotation)  
 4: courroie (domaine mécanique de translation)

**Figure 133 : visualisation des domaines physiques intervenant dans la modélisation de l'axe linéaire avec Simscape**

## 2. Placement et assemblage des composants

Dans la fenêtre **Simulink Library Browser**, cliquer sur l'icône  pour créer un nouveau fichier.

**Glisser/Déposer** les différents blocs à partir des bibliothèques, les disposer dans la fenêtre de travail et les relier pour obtenir la configuration de la Figure 134.

La Figure 135 donne quelques commandes utiles à la mise en forme du modèle.

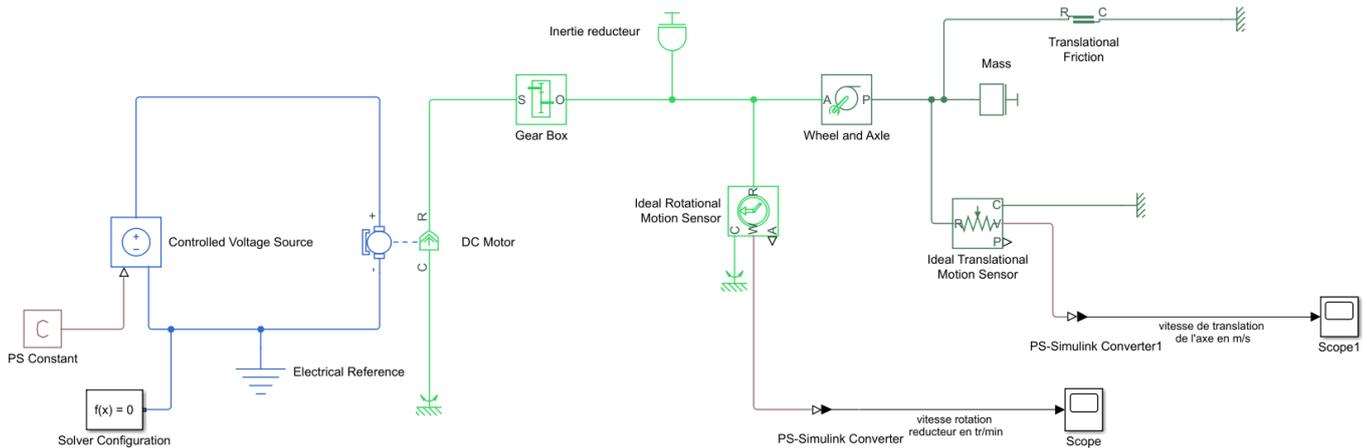


Figure 134 : modèle Simscape de l'axe linéaire

Commandes utiles	
Fonctions	Actions
Donner un nom à un signal Simulink	Clic droit directement sur la connexion du signal Simulink, sélectionner <b>Properties</b> , puis compléter le champ <b>Signal Name</b> dans la fenêtre. Le nom du signal va apparaître dans la légende du scope.
Renommer un bloc	Double-clic directement sur le nom du composant puis taper le nouveau nom directement (attention deux composants ne peuvent pas avoir le même nom)
Supprimer le nom d'un composant	Clic droit sur le composant, sélectionner <b>Format</b> puis désélectionner <b>Show Block Name</b>

Figure 135 : commandes utiles

### 3. Paramétrage des composants

**Effectuer** le paramétrage des différents blocs conformément aux informations données ci-dessous.

#### Paramétrage

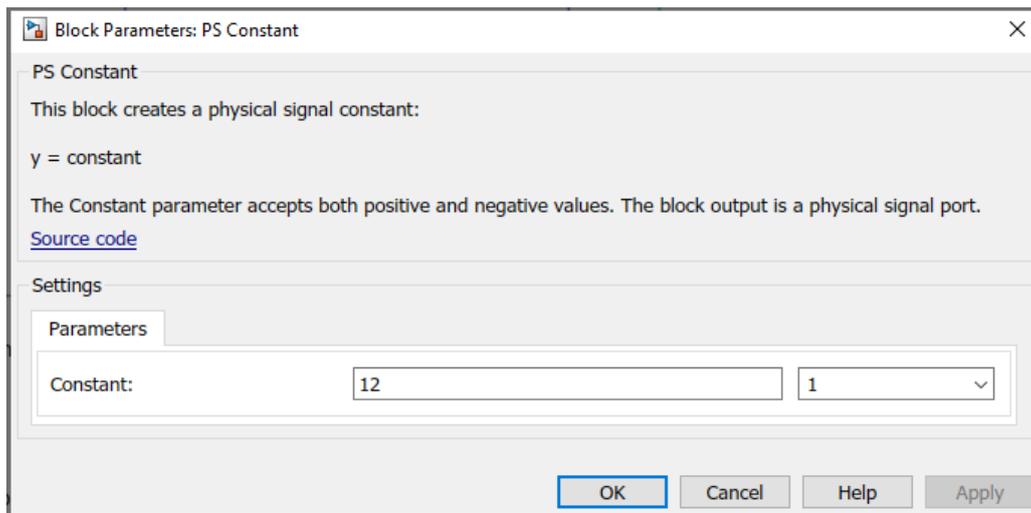
PS Constant



Simscape/Foundation Library/Physical  
Signales/Sources

La valeur indiquée dans ce composant permet de définir la valeur de la tension de commande de la source de tension variable.

Le paramétrage de ce bloc consiste à saisir la valeur du signal physique, ici 12 V.



#### Paramétrage

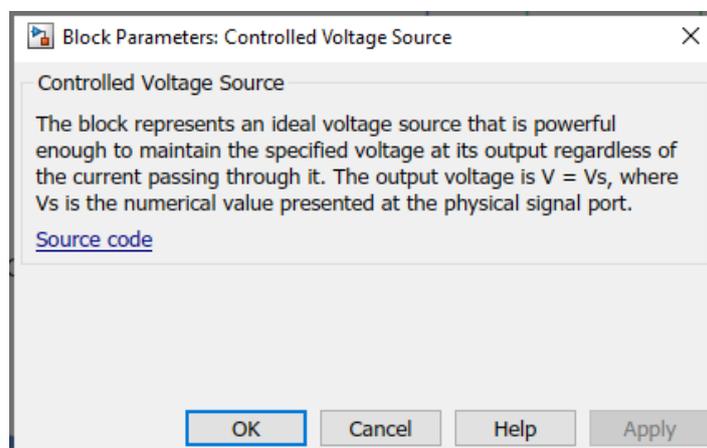
CONTROL VOLTAGE SOURCE



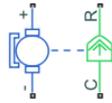
Simscape/Foundation  
Library/Electrical/Electrical Element

Ce composant ne nécessite aucun paramétrage.

Le port correspondant au signal physique, indique à la source de tension variable la valeur de la tension imposée aux bornes de la source.



DC MOTOR

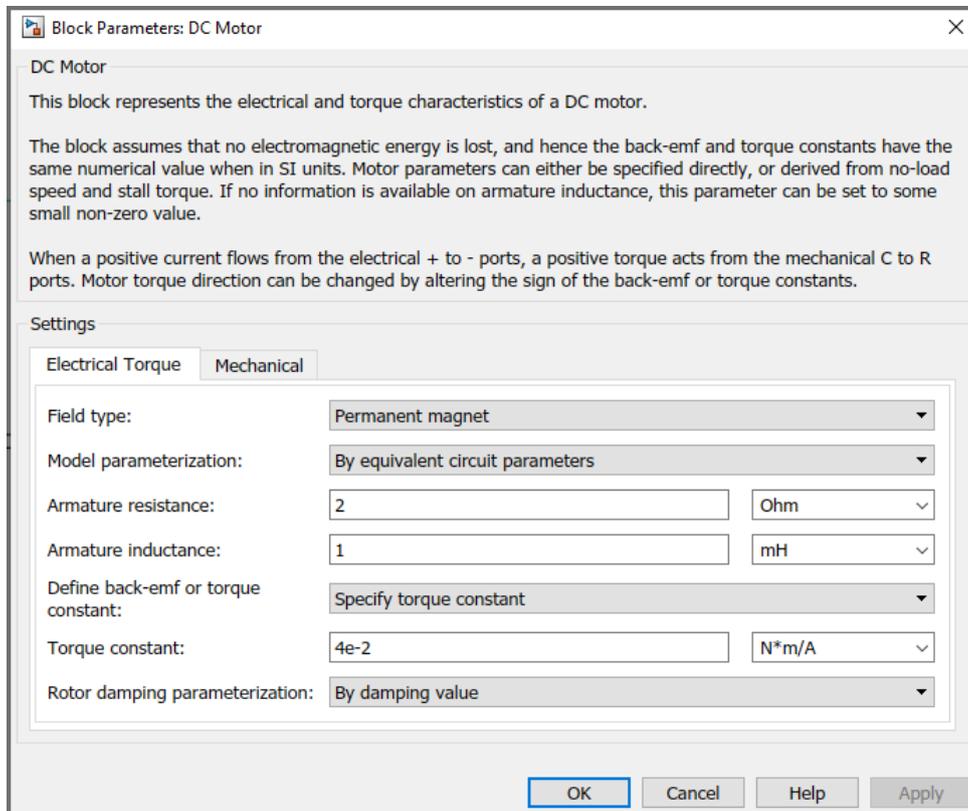


Simscape/Electrical/Electromechanical/Brushed Motors

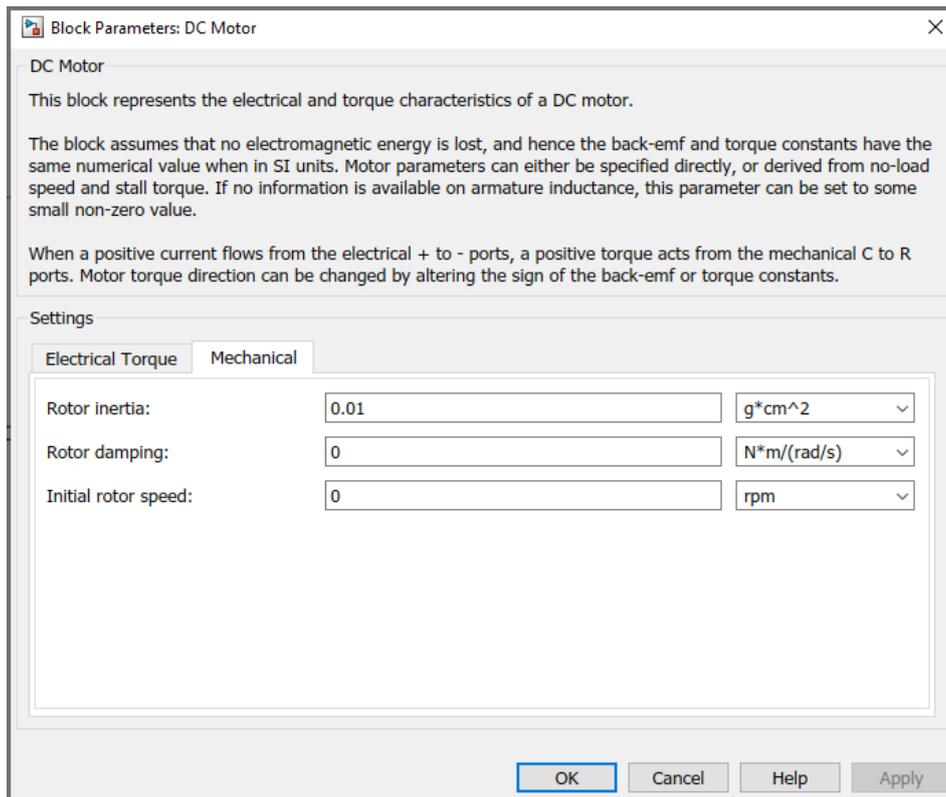
Ce composant modélise un moteur à courant continu. Il possède 4 ports de type **PCP**, 2 du domaine électrique (+ et -) et 2 du domaine mécanique de rotation (**C** et **R**). Le PCP du domaine mécanique **C** (**C**age, partie fixe) sera relié à une référence mécanique de rotation que l'on pourra associer au bâti. L'autre port mécanique de type **R** (**R**otor, partie mobile) sera associé à l'arbre de sortie du moteur.

Ce bloc permet de disposer d'un moteur à courant continu déjà modélisé et de permettre un paramétrage à partir des données constructeur. Deux onglets permettent de saisir les paramètres électriques et les paramètres mécaniques.

### Onglet **Electrical Torque**



## Onglet **Mechanical**



Par défaut le **Model Parameterization** de l'onglet **Electrical Torque** est réglé sur **By equivalent circuit parameters**. En fonction des informations disponibles dans la fiche constructeur du moteur, il sera possible de modifier le type de paramètres à saisir en choisissant **By stall Torque & and no-load speed** ou **By rated power, rated speed & noload speed**.

## Paramétrage

### GEAR BOX

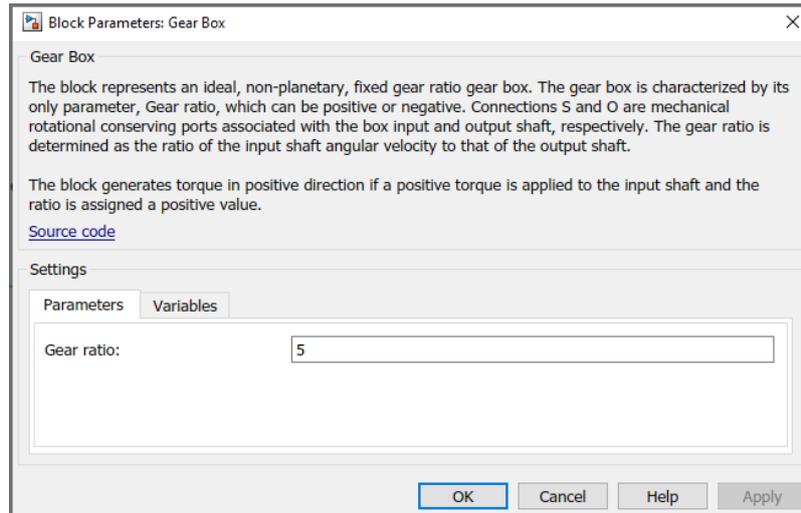


Simscape/Foundation  
Library/Mechanical/Mechanisms

Ce composant permet de modéliser le comportement d'un réducteur parfait.

Par convention le Gear ratio est défini tel que  $Gear\ ratio = \frac{\omega_{entrée}}{\omega_{sortie}}$  ce qui signifie que pour un réducteur

de rapport de réduction  $r = \frac{1}{5} = \frac{\omega_{sortie}}{\omega_{entrée}}$  le Gear ratio à spécifier sera égal à 5.



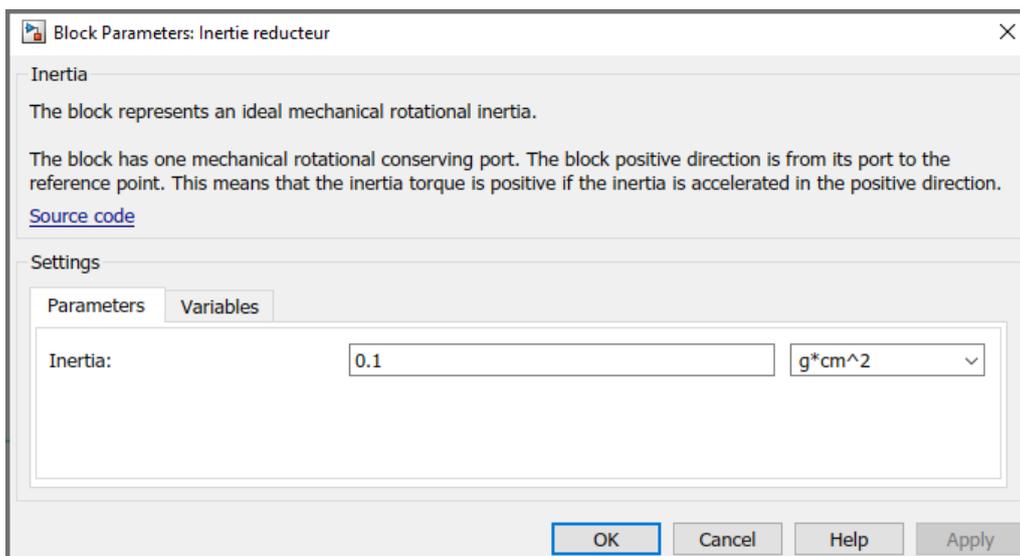
## Paramétrage

### INERTIA



Simscape/Foundation  
Library/Mechanical/Rotational Elements

Ce composant permet de modéliser une inertie associée à une connexion du domaine mécanique de rotation.



Il est également possible de spécifier une vitesse initiale pour cette inertie. La modélisation acausale permet de choisir des conditions initiales non nulles (par opposition à l'approche causale par fonction de transfert ou les grandeurs physiques doivent être nulles à  $t=0$ ).

### Paramétrage

WHEEL AND AXLE

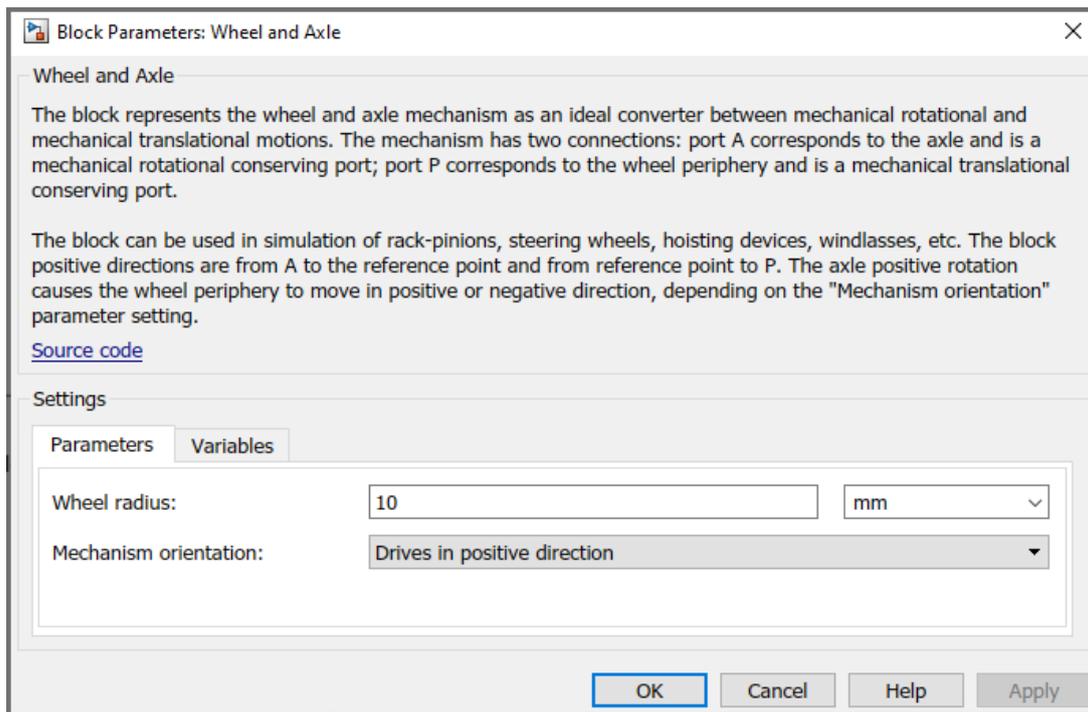


Simscape/Foundation  
Library/Mechanical/Mechanisms

Ce composant permet de modéliser une conversion de puissance parfaite entre un mouvement de rotation et un mouvement de translation.

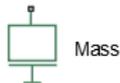
Le paramètre à renseigner est le **rayon** de la poulie.

Il est également possible de spécifier une convention de signe pour réaliser la conversion.



### Paramétrage

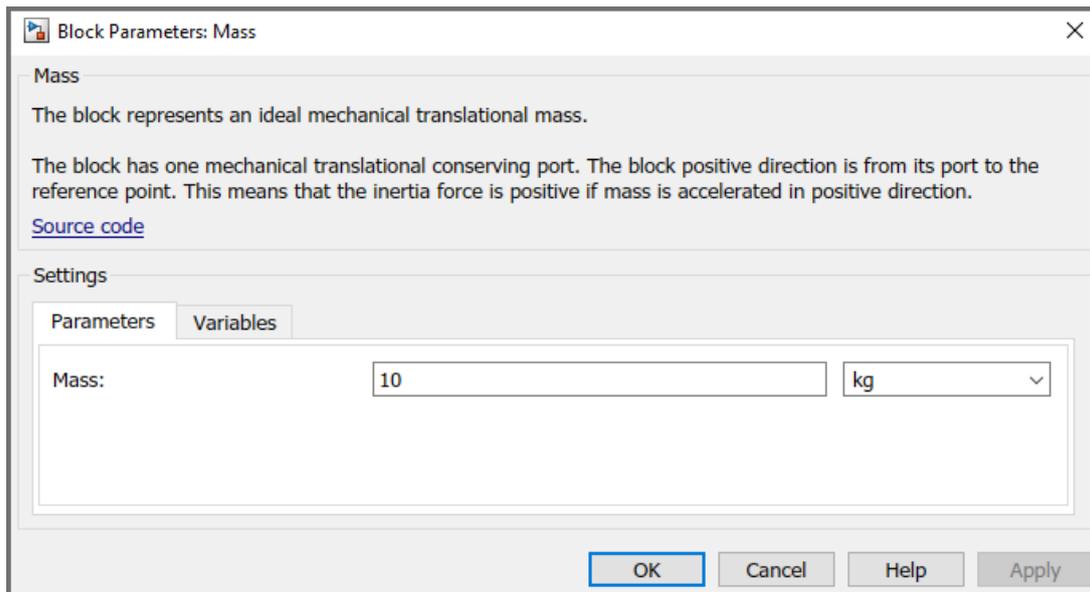
MASS



Simscape/Foundation  
Library/Mechanical/Translational Elements

Ce composant permet de modéliser une masse associée à une connexion du domaine mécanique de translation.

Ici, la masse correspond à la masse du chariot entraîné en translation par la poulie.



Il est également possible de spécifier une vitesse initiale pour cette masse.

## Paramétrage

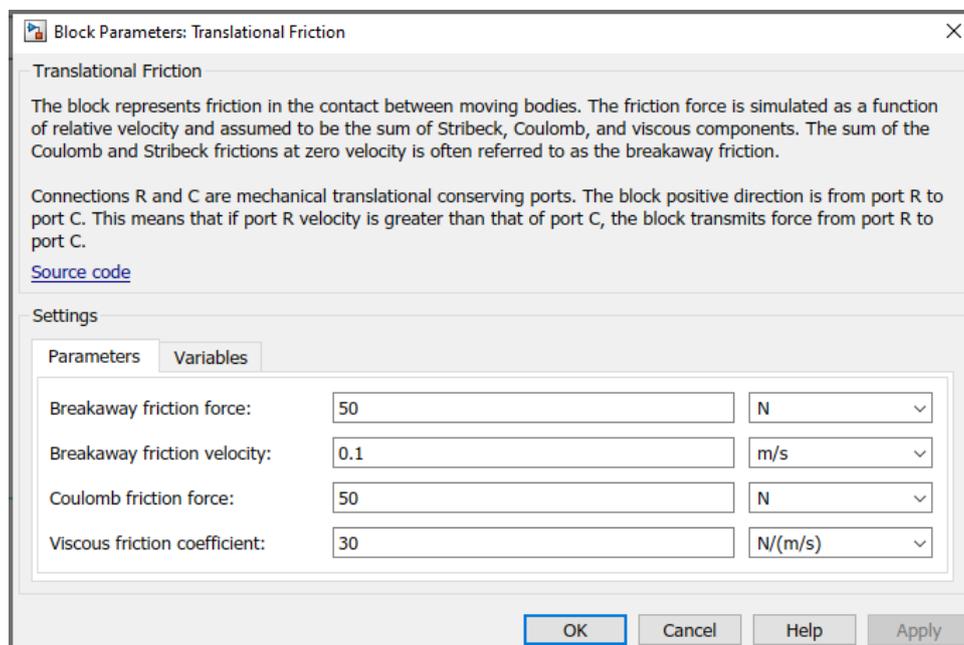
TRANSLATIONAL FRICTION



Simscape/Foundation

Library/Mechanical/Translational Elements

Ce composant permet de modéliser une force de frottement (sec et visqueux) qui s'exerce sur un élément du domaine mécanique de translation. Le port **C** sera relié au bâti et le port **R** à la connexion qui subit l'effort de frottement. L'effort sera exercé par le bâti sur l'élément du domaine mécanique de translation auquel, le port **R** est relié. L'effort s'exercera toujours en s'opposant au déplacement. Ici, il s'agira d'un effort exercé par le bâti sur le chariot qui jouera le rôle de perturbation pour le système.



**Breakaway friction force** : force limite d'adhérence (force à partir de laquelle le mouvement commence)

**Breakaway friction velocity** : ce terme permet de caractériser le passage du frottement d'adhérence au frottement de glissement. Plus ce paramètre est faible et plus le passage se fera brutalement.

**Coulomb friction force** : force de frottement de glissement (qui s'exerce pendant le mouvement)  
**Viscous friction coefficient** : coefficient de frottement visqueux

Paramétrage

IDEAL TRANSLATIONAL  
 MOTION SENSOR

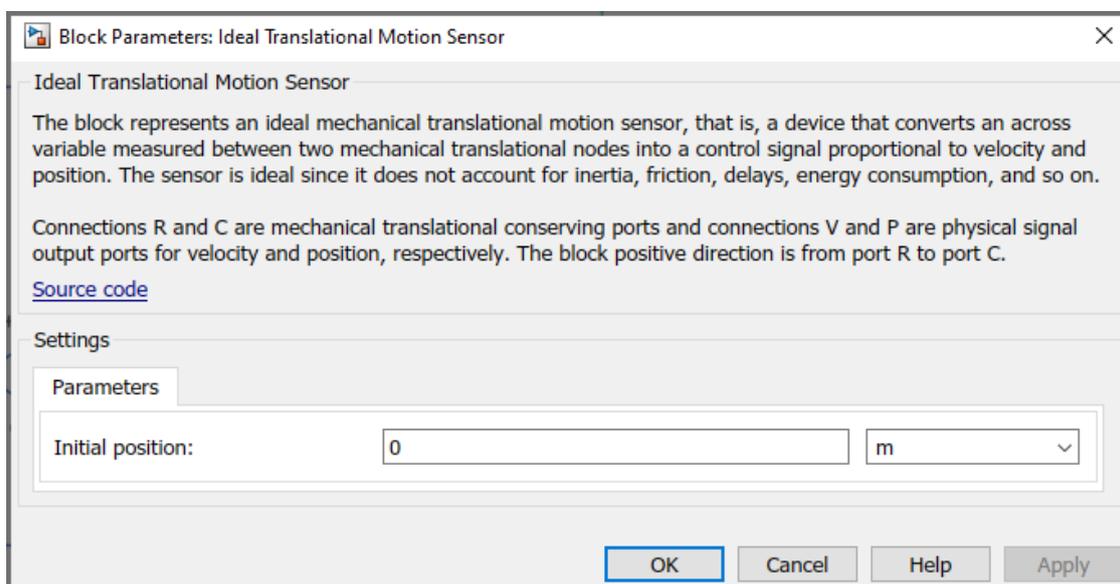


Simscape/Foundation  
 Library/Mechanical/Mechanical Sensors

Ce capteur permet de mesurer une vitesse de translation (variable de type « Across ») ou une position en translation. Il se monte donc en parallèle. Le port **R** est relié à la connexion dont on souhaite connaître la vitesse par rapport au port **C** que l'on relie à la référence choisie. Le signal physique que l'on peut prélever est soit la vitesse de translation (port **V**) soit la position (port **P**).

Ici, on souhaite connaître la vitesse de déplacement du chariot par rapport au bâti.

Il est également possible de spécifier la valeur initiale de la mesure en début de simulation (0 par défaut).



Paramétrage

IDEAL ROTATIONAL  
 MOTION SENSOR

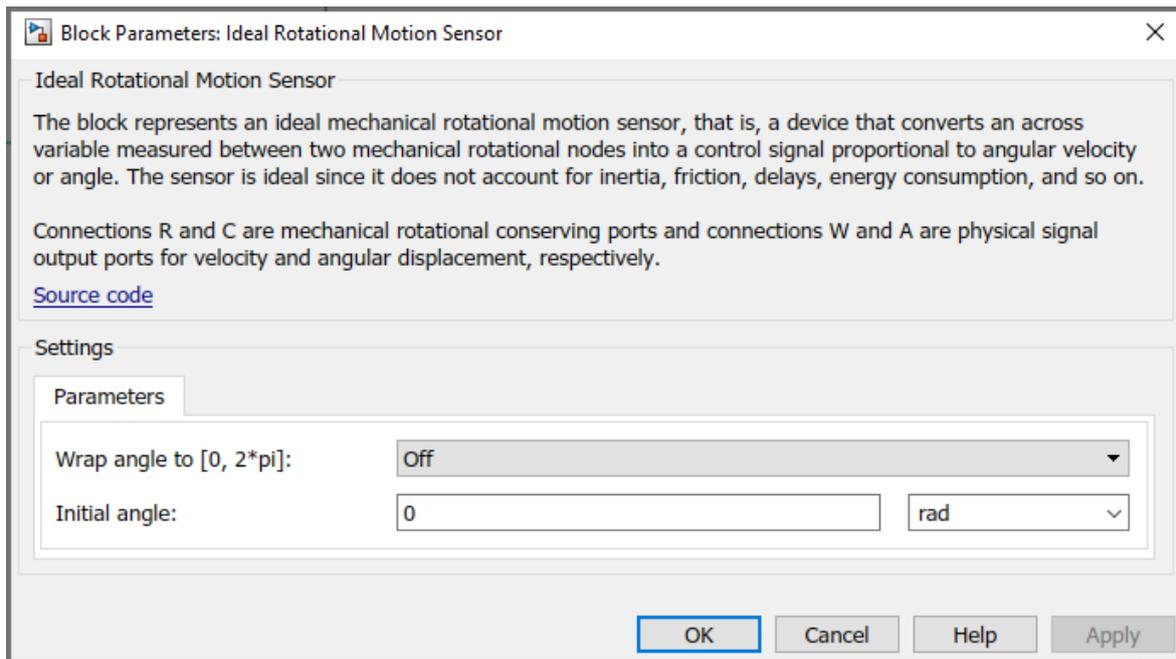


Simscape/Foundation  
 Library/Mechanical/Rotational Sensors

Ce capteur permet de mesurer une vitesse de rotation (variable de type « Across ») ou un angle. Il se monte donc en parallèle. Le port **R** est relié à la connexion dont on souhaite connaître la vitesse par rapport au port **C** que l'on relie à la référence choisie. Le signal physique que l'on peut prélever est soit la vitesse de rotation (port **W**) soit l'angle (port **A**).

Ici, on souhaite connaître la vitesse de déplacement du chariot par rapport au bâti.

Il est également possible de spécifier la valeur initiale de la mesure en début de simulation (0 par défaut).



## Paramétrage

PS-SIMULINK CONVERTER



Simscape/Utilities

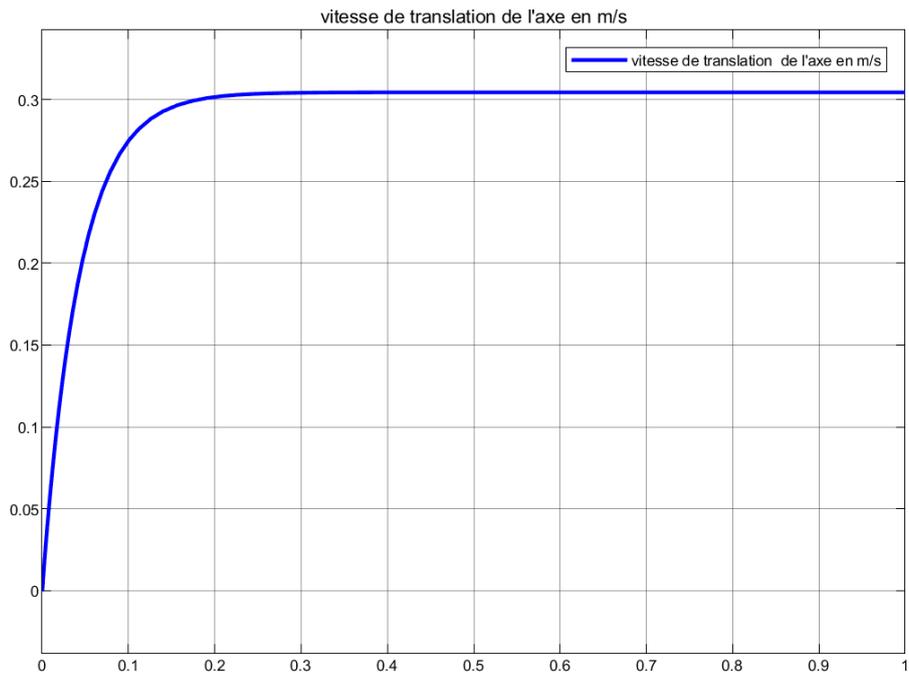
Ce composant permet de convertir un signal physique image d'une grandeur mesurée en signal Simulink afin de pouvoir être affiché dans un Scope. Spécifier que la vitesse de translation du chariot est relevée en m/s et la vitesse de rotation de l'arbre en tr/min (rpm pour Simscape)

Le fichier contenant le modèle paramétré est disponible sous le nom **axe\_lineaire\_0.slx**

### 4. Simulation du modèle en boucle ouverte

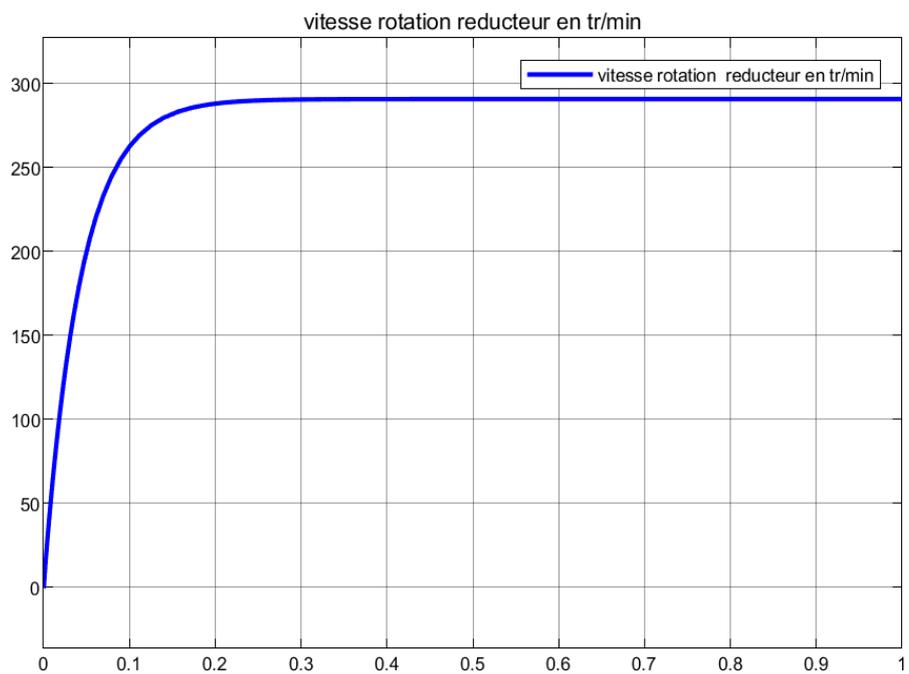
**Lancer** la simulation en spécifiant un temps de simulation de **1s**.

**Observer** la vitesse de translation en double-cliquant sur le scope.



La vitesse de translation du chariot se stabilise autour de 0.3 m/s.

**Observer** la vitesse de rotation de l'arbre de sortie du réducteur en double-cliquant sur le scope.



La vitesse de rotation de l'arbre de sortie de réducteur se stabilise autour de 290 tr/min.

## 5. Utilisation du Simscape Results Explorer

Lors de la conception d'un modèle, il est très utile d'avoir accès à l'ensemble des variables qui interviennent dans le modèle. La première solution peut consister à implanter des capteurs partout où il est nécessaire d'observer des variables. Cette méthode se montrera vite longue et fastidieuse et rendra le modèle très peu lisible. Pour répondre à ce besoin de conception, **Simscape** offre la possibilité de stocker l'évolution de toutes les variables qui évoluent au sein du modèle. Il sera alors possible à l'aide du **Simscape Results Explorer** de visualiser l'évolution de toutes les variables.

Il faut dans un premier temps spécifier le nom de la variable qui stockera toutes les données de la simulation.

Cliquer sur l'icône Model Setting , puis choisir l'onglet **Simscape** dans la partie gauche de la fenêtre.

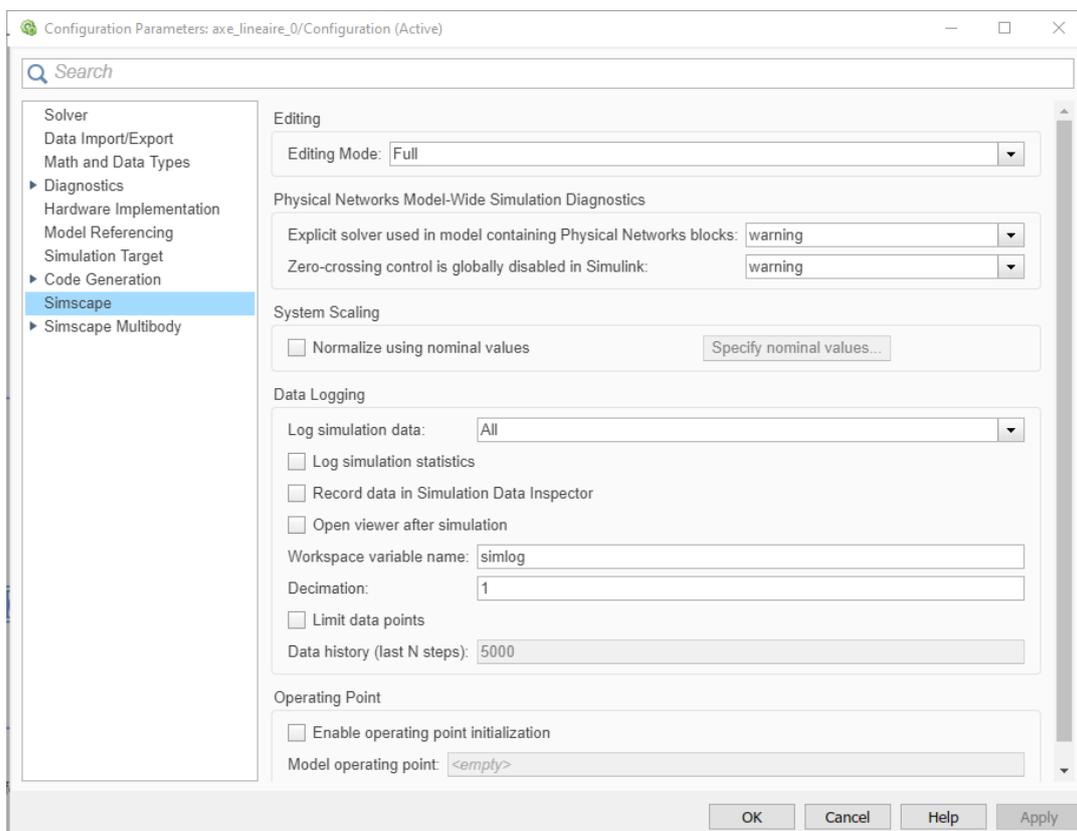


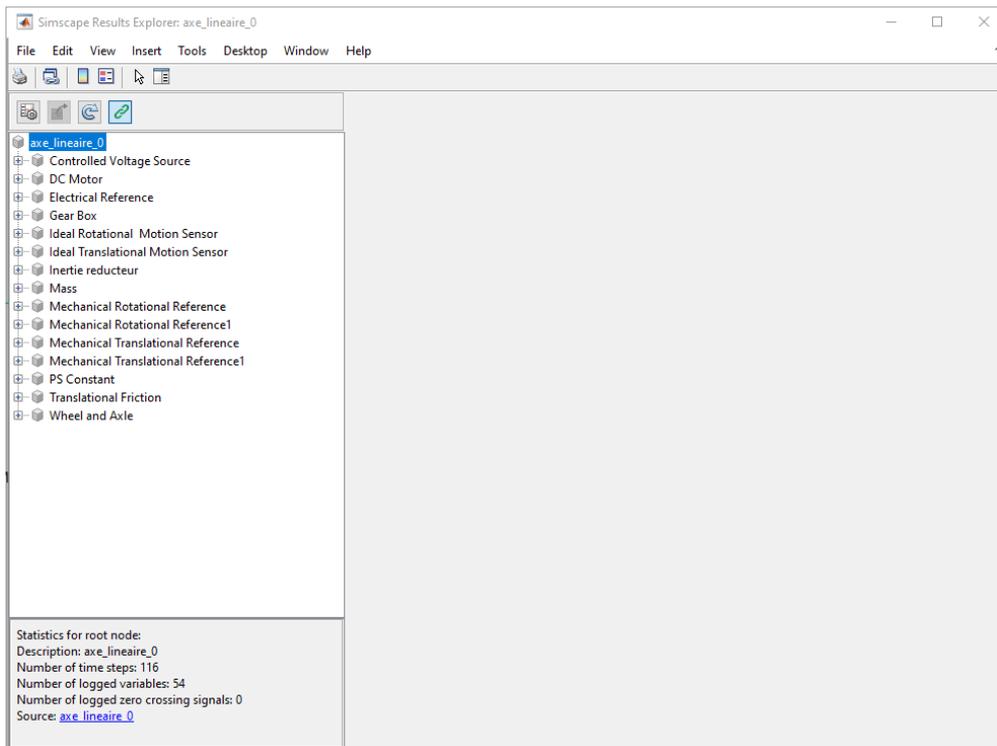
Figure 136 : spécification des variables Simscape à prendre en compte dans le logger

Choisir **all** dans le menu **Log simulation data**.

Choisir le nom de la variable qui stockera les données de la simulation dans le champ **Workspace Variable Name** (ici **simlog**, mais ce nom peut être modifié)

Désélectionner **Limit data points** afin d'être certain de récupérer toutes les données de la simulation.

**Relancer** la simulation et cliquer sur l'icône Simscape Results  pour ouvrir la fenêtre Simscape Results Explorer (Figure 137).



**Figure 137 : fenêtre du Simscape Result Explorer**

Dans la partie gauche de la fenêtre apparaît le nom des composants du modèle.

Choisir par exemple le composant **Gear\_Box** et développer l'arborescence pour voir apparaître toutes les variables du composant. L'arborescence fait apparaître les deux ports **O** et **S** du composant.

Deux courbes apparaissent montrant l'évolution du **Moment** en N.m (variable de type « Through ») à l'entrée **S** et à la sortie **O** du composant (Figure 138).

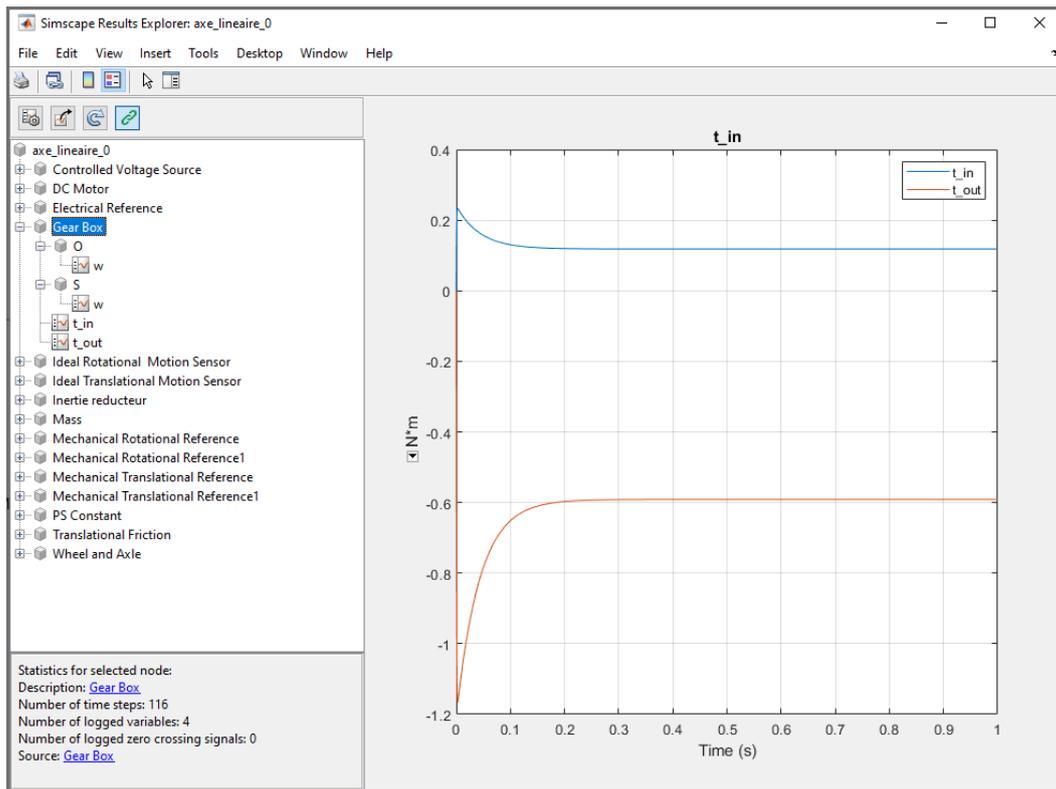
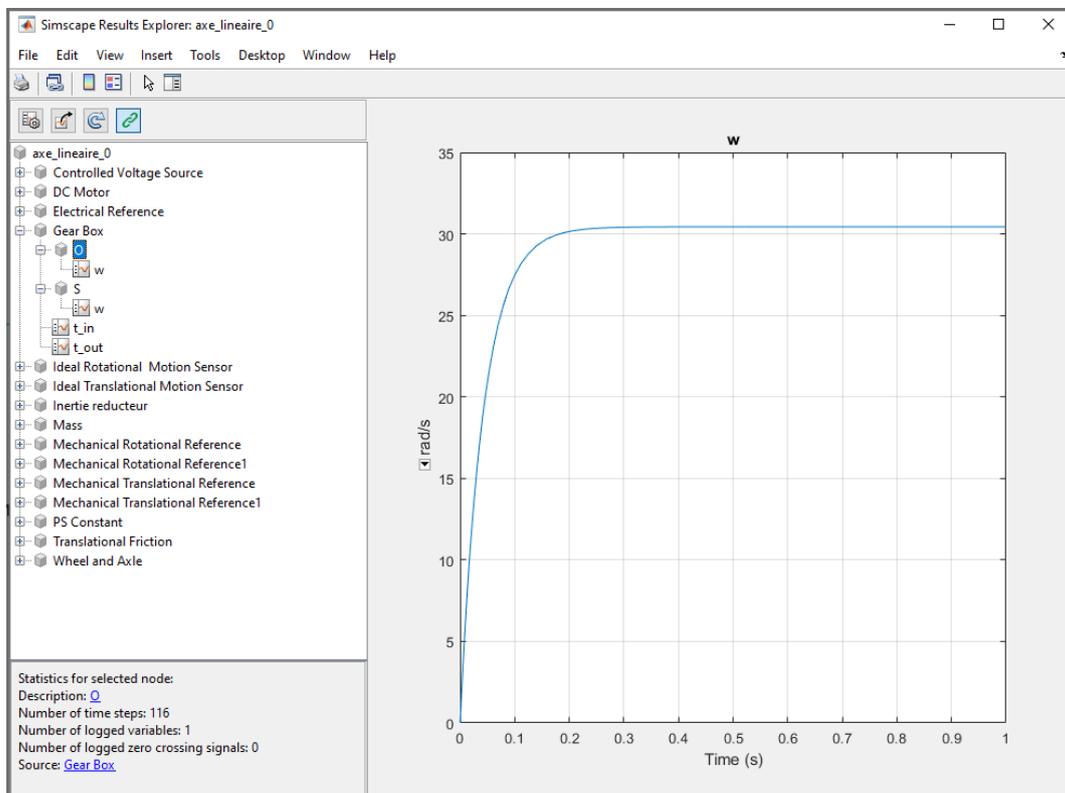


Figure 138 : visualisation des résultats de la simulation

Il est également possible d'accéder à la vitesse de rotation (variable de type « Across ») au niveau de l'entrée **O** et de la sortie **S** du composant.



Le principe du Simscape Results Explorer est de pouvoir observer les variables de types « Across » et « Through » pour tous les composants du modèle. Pour chaque composant, les ports de type **PCP** apparaissent dans l'arborescence. Il suffit de cliquer sur la variable que l'on souhaite observer pour visualiser la courbe.

Le Simscape Result Explorer de **Simscape** devient vite indispensable et permet un gain de temps considérable lors de la conception des modèles.

## 6. Création de sous-systèmes

Afin de simplifier la compréhension et l'exploration du modèle, il est possible de créer des sous-systèmes. Cette démarche deviendra vite indispensable lors de la conception de modèles complexes. Le principe consiste à sélectionner un groupe de composants pour créer un sous-système.

Nous allons créer un sous-système comprenant le moteur et son alimentation.

**Sélectionner** comme indiqué sur la Figure 139 les composants qui composeront le sous-système puis faire glisser la souris sur l'icône **Create Subsystem** en bas à droite de la sélection.

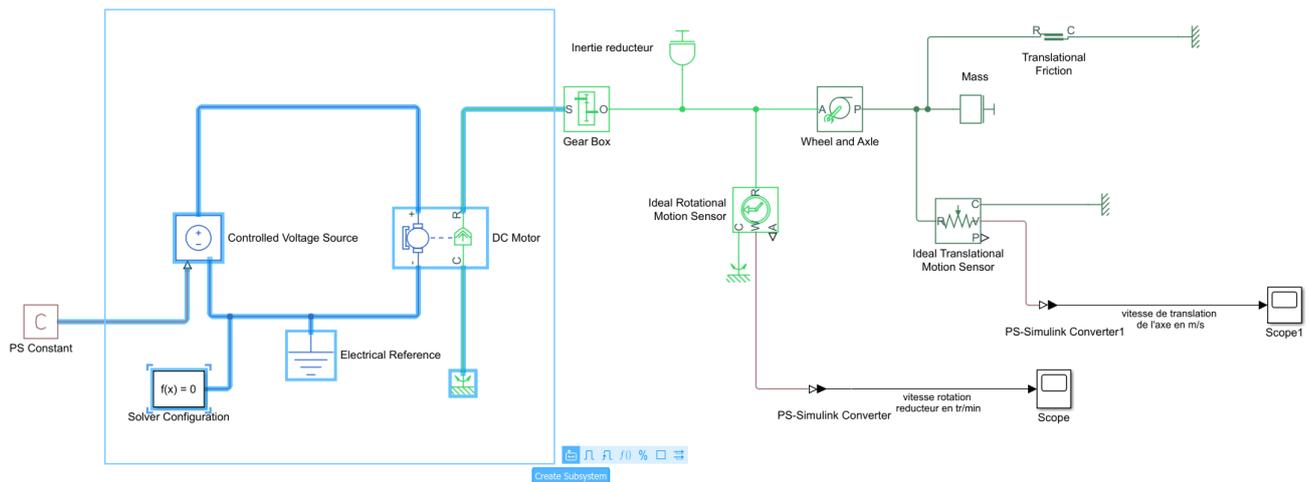


Figure 139 : création d'un sous-système

Il est également possible de faire un **Clic-droit** sur la sélection et de choisir **Create Subsystem from Selection** dans le menu contextuel ou d'utiliser le raccourci clavier **Ctrl+G**.

**Redimensionner** ou pivoter éventuellement le sous-système pour voir apparaître les ports **Controlled Voltage Source1** et **DC Motor1** du sous-système (Figure 141).

### Commandes utiles

Fonctions	Actions
Redimensionner un composant	<b>Sélectionner</b> le composant avec le bouton gauche de la souris, puis <b>tirer</b> sur les poignées qui apparaissent sur le contour du composant pour le redimensionner

Figure 140 : commandes utiles

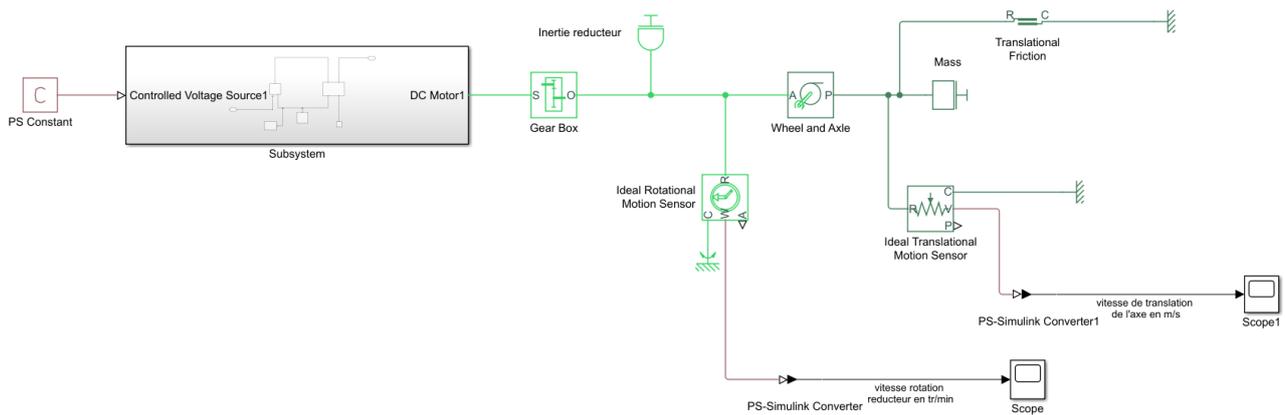


Figure 141 : création d'un sous-système

Pour information, il est également possible de revenir à la forme précédente du modèle en « étendant » le sous-système. Pour cela faire un Clic droit sur le sous-système et choisir la commande **Subsystem & Model Reference/Expand Subsystem**.

**Double-cliquer** sur le nom du sous-système situé sous le cadre du sous-système pour le remplacer par «Moteur »

**Double-cliquer** ensuite sur le sous-système pour voir les composants qui le constituent.

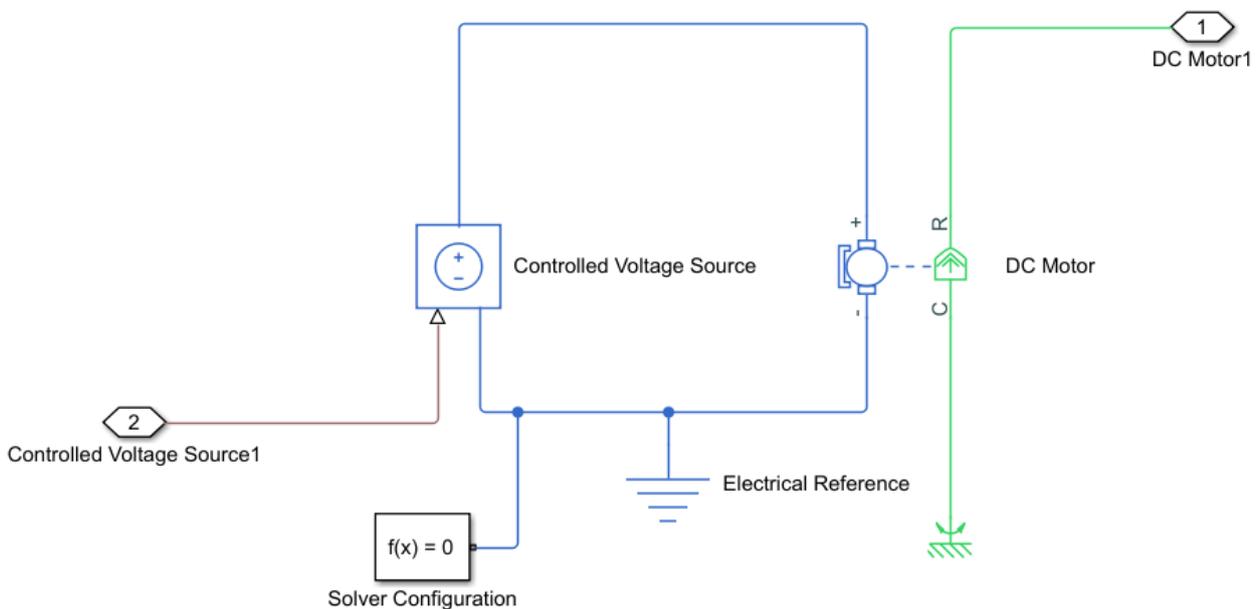


Figure 142: visualisation du contenu du sous-système

Nous pouvons observer que les connexions avec l'extérieur apparaissent sous la forme de ports **Controlled Voltage Source1** et **DC Motor1**. Il est également possible de renommer ces ports.

**Double-cliquer** directement sur le nom du port **Controlled Voltage Source1** et le remplacer par « **Tension de commande** ».

Faire de même en remplaçant le nom **DC Motor1** par « **Arbre moteur** » (Figure 143)

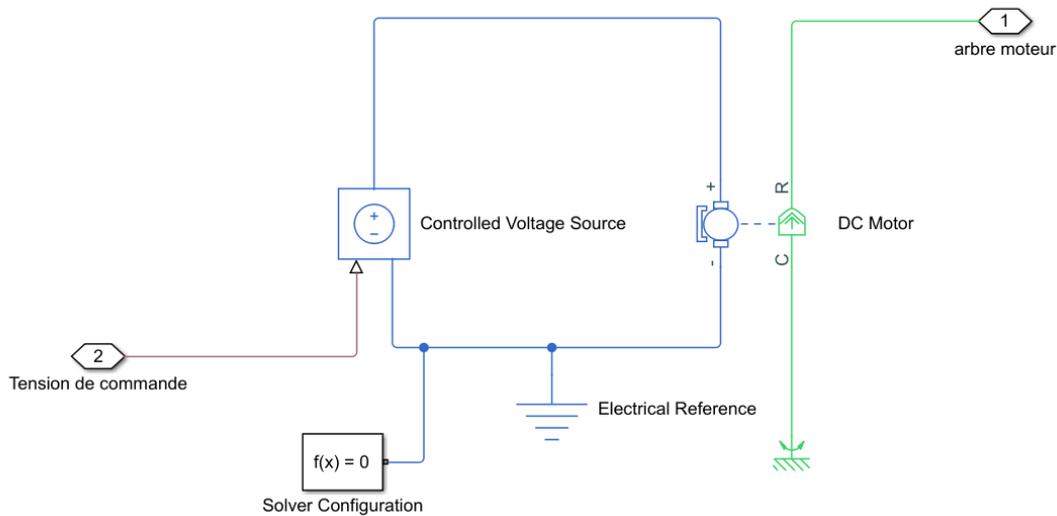


Figure 143 : renommer les ports d'un sous-système

Revenir au modèle global en cliquant sur le nom du modèle global dans la barre de navigation ou en utilisant le Model Browser. Le Model Browser (situé à gauche de la fenêtre) permet de naviguer facilement d'un sous-système au système global (Figure 144). Il est également possible d'utiliser les flèches de navigation dans les différents niveaux du modèle.

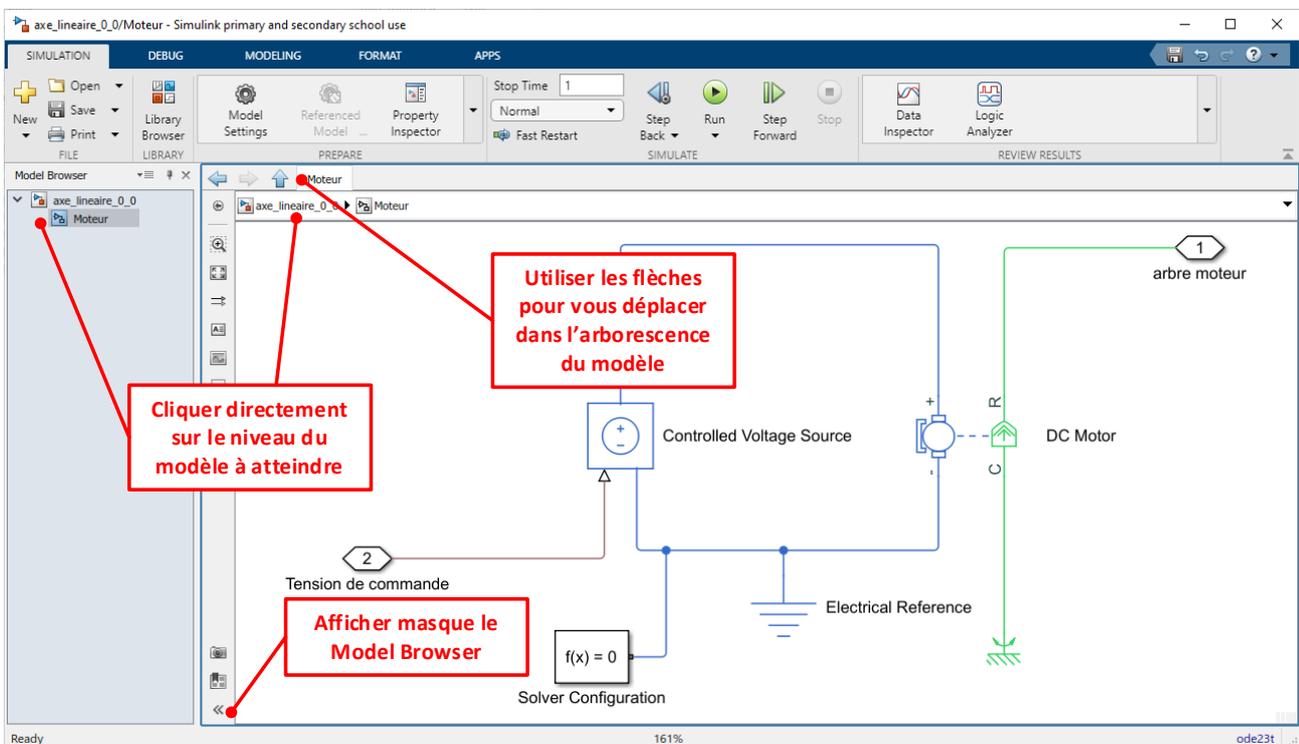
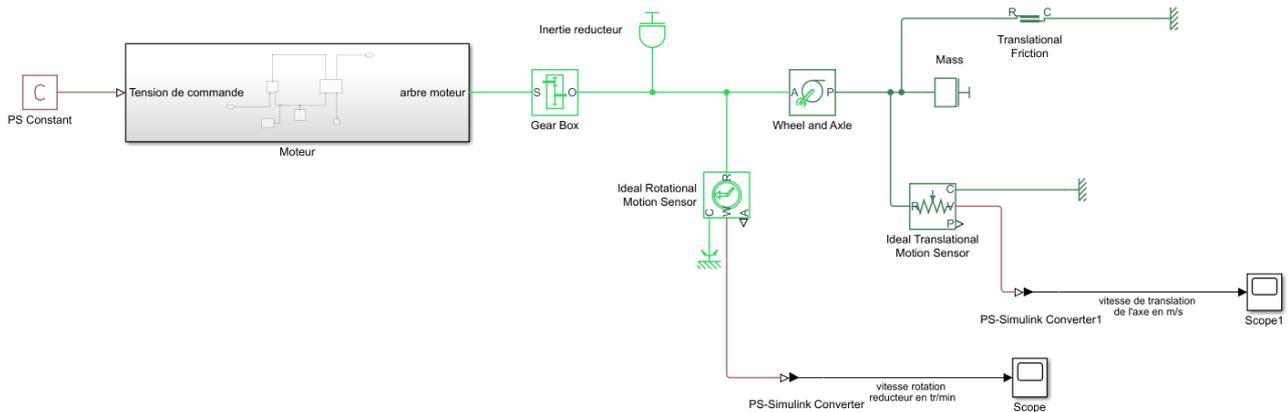


Figure 144 : naviguer dans les sous-systèmes d'un modèle

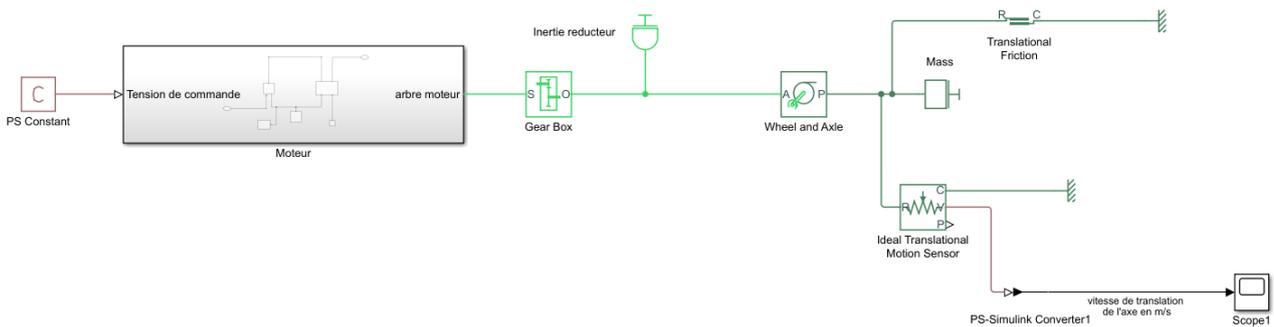
Vous devez obtenir la configuration de la Figure 145.



**Figure 145 : modèle Simscape de l'axe linéaire avec sous-système « moteur »**

Dans l'objectif de créer un asservissement de position, nous allons supprimer le capteur de vitesse de rotation de l'arbre moteur et transformer la mesure de la vitesse linéaire de l'axe en mesure de position.

**Supprimez** le capteur de vitesse de rotation de l'arbre moteur pour obtenir le schéma de la Figure 146.



**Figure 146 : modèle Simscape de l'axe linéaire sans le capteur de vitesse de rotation**

**Modifier** la grandeur physique prélevée sur l'**Ideal Translational Motion Sensor** afin de relever la position linéaire du chariot (Figure 147).

Il faudra changer l'unité dans le bloc **PS-Simulink Converter** et choisir mètre (m).

On modifiera également le nom du signal pour obtenir la configuration de la Figure 147.

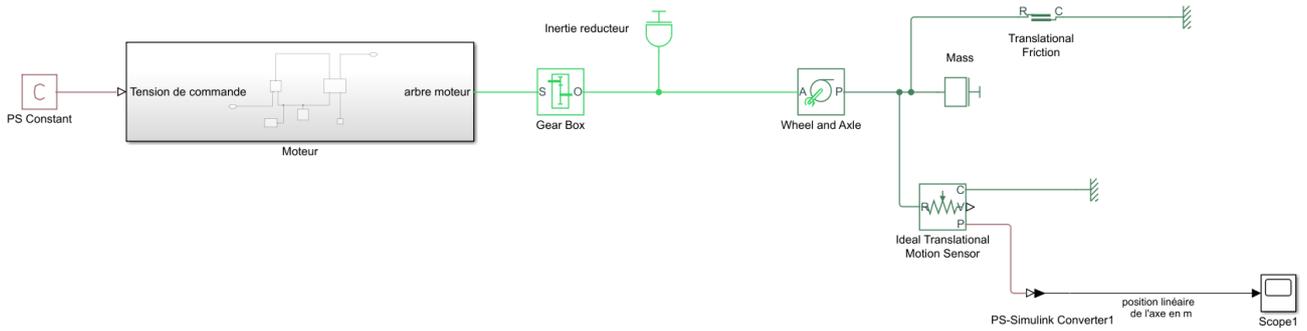


Figure 147 : modèle Simscape de l'axe linéaire avec le nom des signaux

Créer un sous-système « Axe » en sélectionnant les composants conformément à la Figure 148.

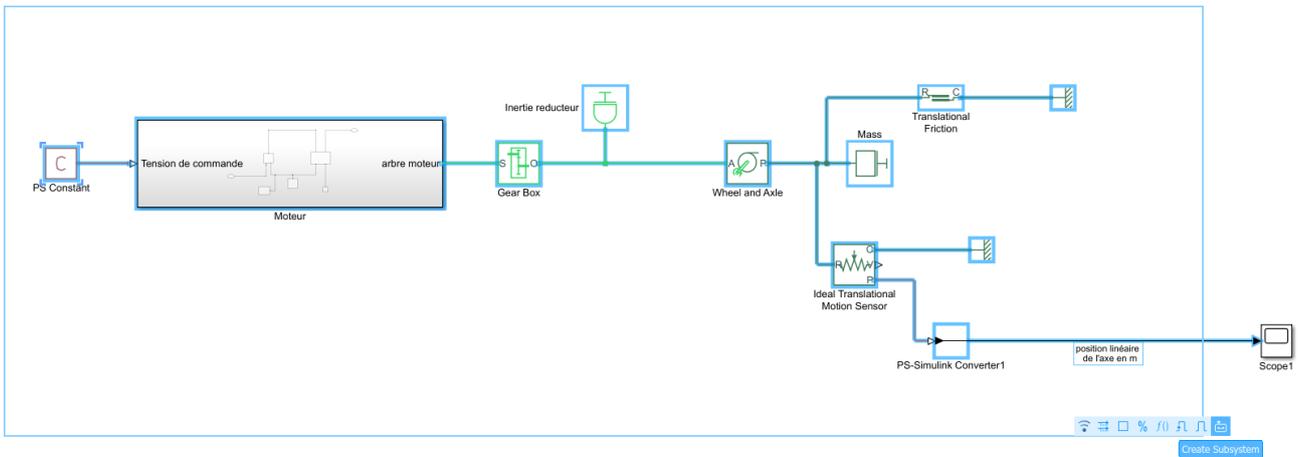


Figure 148 : modèle Simscape de l'axe linéaire, création du sous-système « Axe »

Vous devez obtenir le modèle suivant après redimensionnement du sous-système :

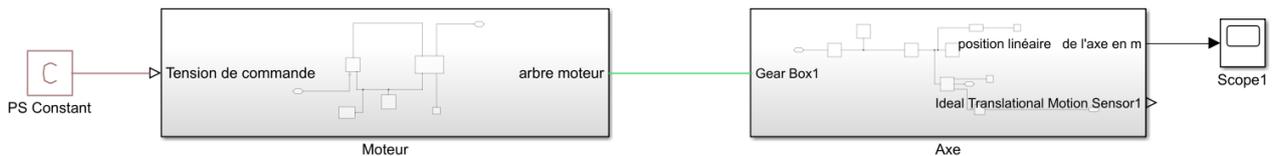


Figure 149 : modèle Simscape de l'axe linéaire avec sous-système « moteur » et « Axe »

On remarque qu'un port supplémentaire **Ideal Translational Motion Sensor1** a été créé.

**Double cliquer** sur le sous-système pour l'explorer.

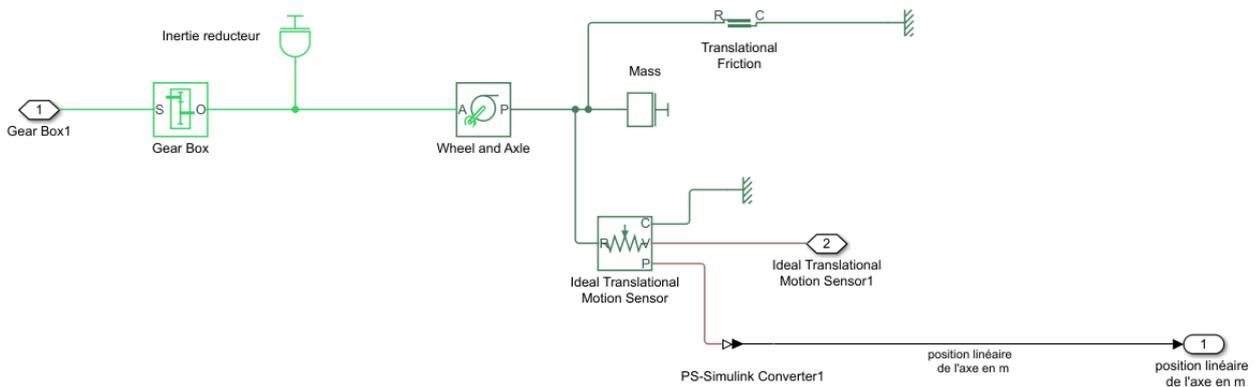


Figure 150 : suppression du port d'un sous-système

Le port **Ideal Translational Motion Sensor1** a été créé par défaut sur la sortie en vitesse du capteur qui n'était pas utilisée. Lors de la création d'un sous-système, tout port qui n'est pas connecté à l'intérieur du sous-système apparaîtra sous la forme d'un port de communication avec l'extérieur.

**Supprimer** ce port.

On remarque que le format du port **Ideal Translational Motion Sensor1** et le format du port **position linéaire de l'axe en m** sont différents. Le port **Ideal Translational Motion Sensor1** est un port de type **Simscape** (transmetteur de puissance, transmetteur de signal physique) et le port **position linéaire de l'axe en m** est un signal **Simulink** (signal numérique orienté, modélisation causale). Le logiciel permet de faire visuellement la distinction entre ces deux types de ports.

Vous devez obtenir le modèle de l'axe linéaire de la Figure 151.

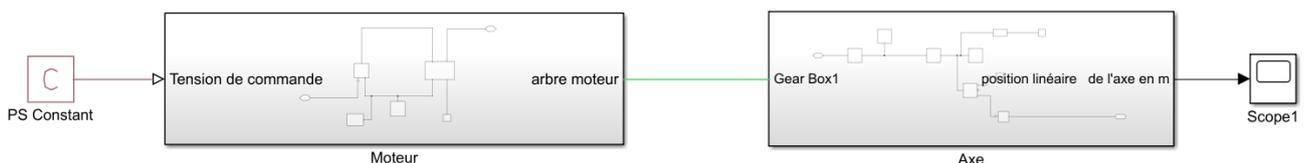


Figure 151 : modèle Simscape terminé de l'axe linéaire

Si nécessaire le modèle correspondant est disponible dans le fichier **axe\_lineaire\_1.slx**.

Il est possible d'ajouter une image sur un sous-système afin d'améliorer la lisibilité du modèle. Pour cela cliquer droit sur le sous-système et sélectionner **Mask/Add Icon Image** (Figure 152).

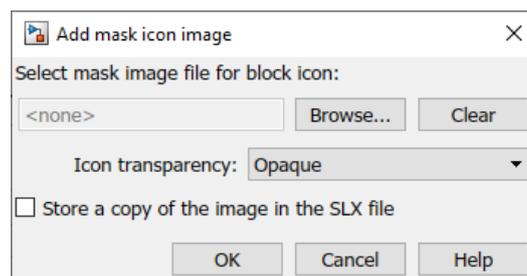


Figure 152 : mettre une image sur un sous-système

Cliquer sur **Browse** afin de chercher l'image **axe.jpg**, puis cliquer sur **Store a copy of the image in the SLX file** qui permettra de sauvegarder l'image dans le modèle. Cette image restera donc toujours reliée au Mask du système. Cliquer ensuite sur **OK**.

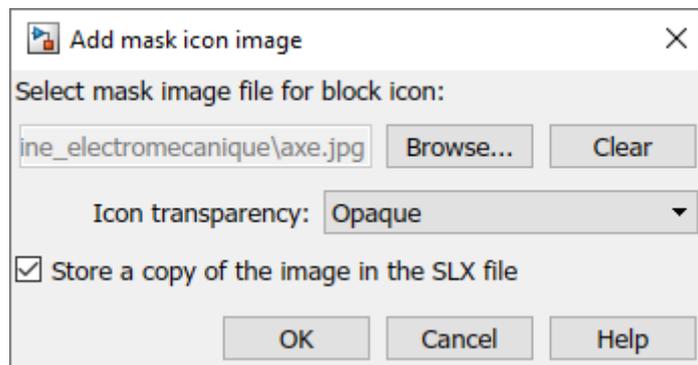


Figure 153 : configuration de la fenêtre « Add mask icon image »

Redimensionnez éventuellement la taille du sous-système pour que l'image retrouve ses proportions initiales (Figure 154).

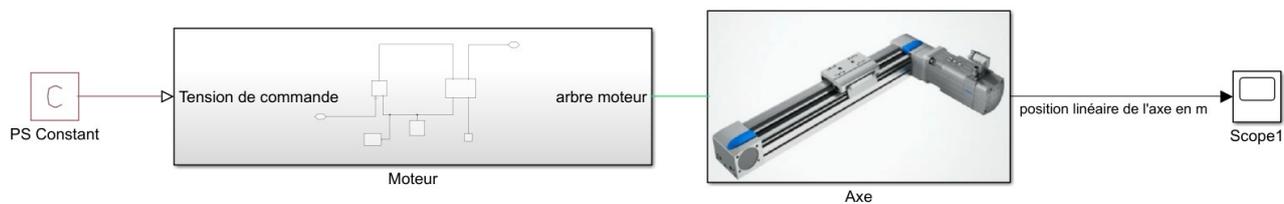


Figure 154 : affichage d'une image sur un sous-système

## 7. Modélisation de l'asservissement en position de l'axe

Le modèle ainsi obtenu permet de visualiser la réponse en boucle ouverte de la position du chariot pour une valeur donnée de la tension moteur.

Pour une tension de 12V, la réponse en position de l'axe est la suivante :

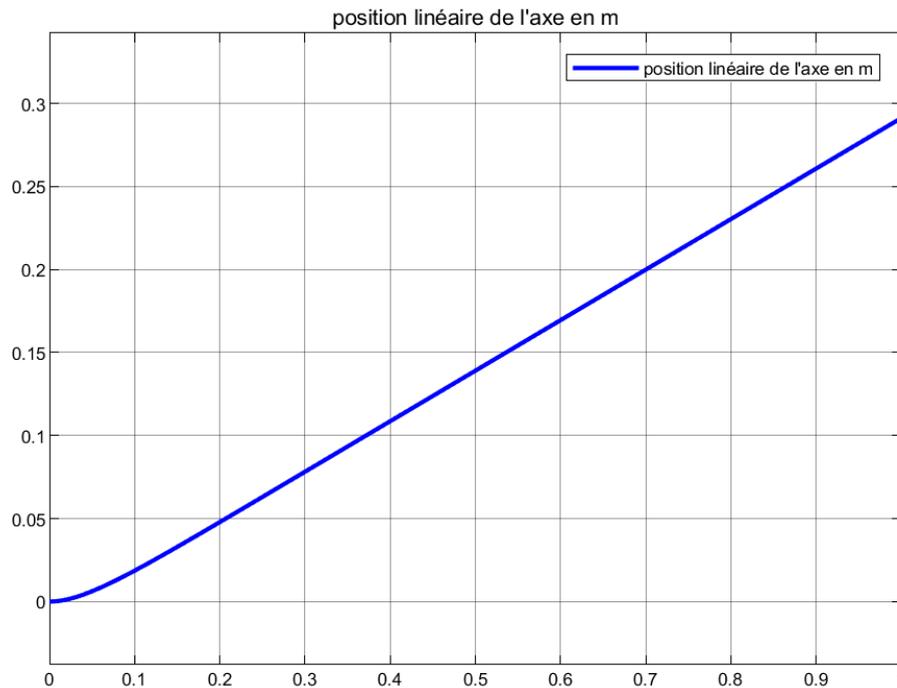


Figure 155 : réponse en position de l'axe linéaire non asservi

Nous pouvons maintenant réaliser le modèle de l'asservissement en position de l'axe représenté sur la Figure 156. Les nouveaux composants ajoutés sont décrits Figure 157.

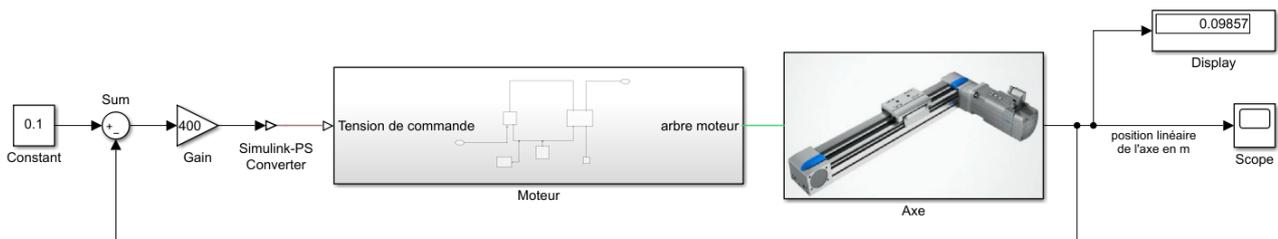


Figure 156 : modèle de l'asservissement en position linéaire de l'axe

Nous imposons une consigne de 0.1 m. Cette consigne est comparée à la mesure de la position de l'axe. Un correcteur proportionnel, transforme l'écart de position en consigne de tension pour le moteur.

Seuls sont référencés dans le tableau de la Figure 157, les composants qui n'ont pas encore été utilisés auparavant dans le document.

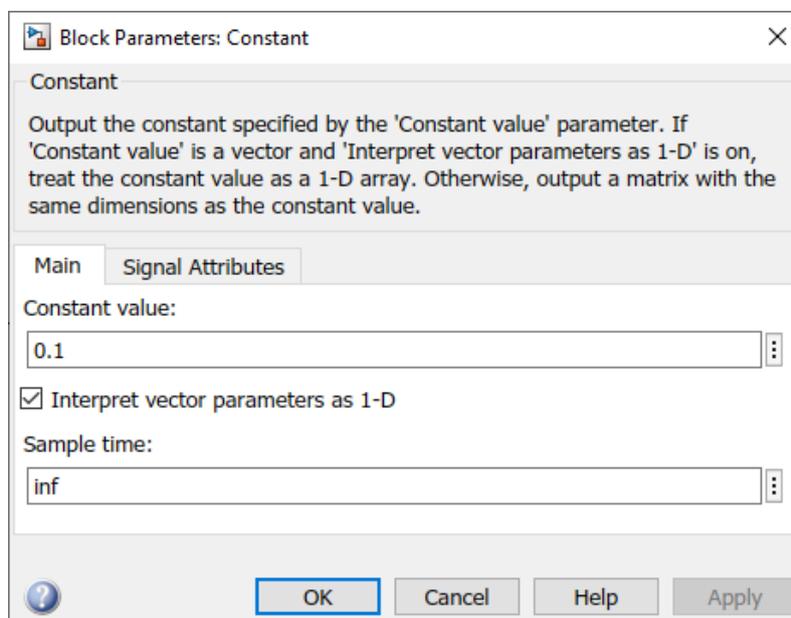
Fonction du composant	Représentation	Bibliothèque
Constante	 Constant	Simulink/Sources ou Simulink/Commonly used Blocks
Gain	 Gain	Simulink/Math Operations ou Simulink/Commonly used Blocks
Sommateur	 +	Simulink/Math Operations ou Simulink/Commonly used Blocks
Afficheur	 Display	Simulink/Sinks

Figure 157 : composants à ajouter au modèle

### Paramétrage

CONSTANT	 Constant	Simulink/Sources ou Simulink/Commonly used Blocks
----------	--	---

Ce bloc génère en sortie un signal numérique constant qui correspond ici à la consigne de position de 0.1 m.



## Paramétrage

SUM



Simulink/Math Operations ou Simulink/Commonly used Blocks

Ce composant permet de réaliser un somme ou une différence de deux signaux. Le paramétrage correspond à la succession de signes que l'on souhaite obtenir lors de l'opération. Indiquer dans le champs **List of Signs** la disposition des signes.

- **| - +** pour un sommateur de type
- **- + |** pour un sommateur de type

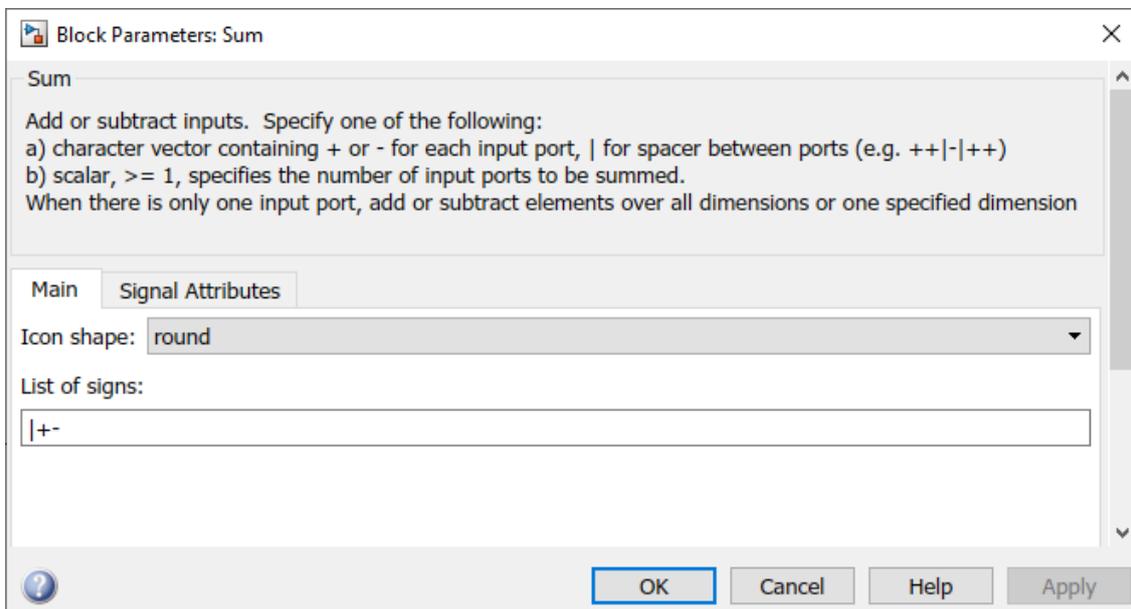


Figure 158 : paramétrage du bloc sommateur

## Paramétrage

DISPLAY



Simulink/Sinks

Ce bloc permet d'afficher une valeur en régime permanent pour visualiser ici la précision de la réponse en position.

**Modifier** le modèle ou ouvrir le fichier « **axe\_lineaire\_asservi.slx** »

**Lancer** la simulation en spécifiant un temps de simulation de **1 s**.

**Visualiser** la réponse en position à l'aide du scope.

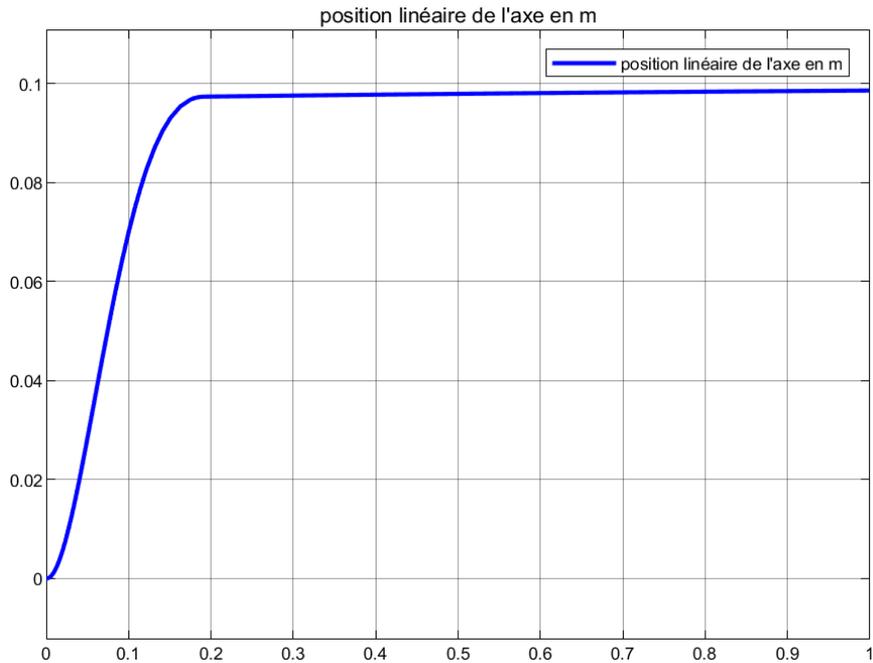


Figure 159 : réponse en position de l'axe linéaire asservi

On remarque que la position finale atteinte par l'axe est de 0.09857 m, ce qui nous permet d'évaluer que l'écart statique est de l'ordre de 2%. Il est possible d'améliorer les performances de l'asservissement en utilisant un correcteur PID.

Remplacer le bloc gain qui représente le correcteur proportionnel par le bloc PID.

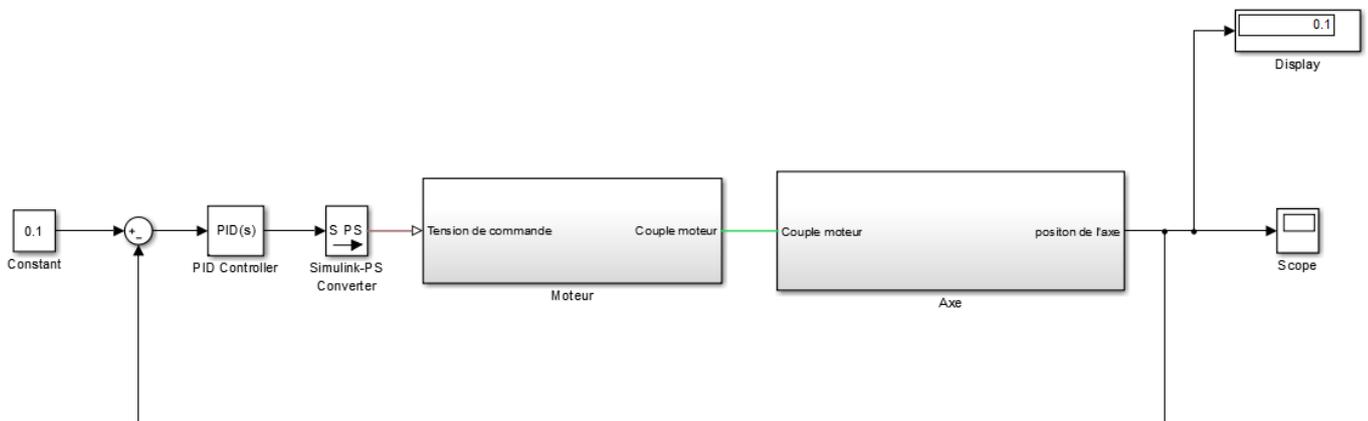
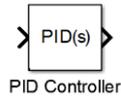


Figure 160 : modèle de l'asservissement en position de l'axe linéaire

## Paramétrage

PID Controller



Simulink/Continues

Ce bloc permet de paramétrer un correcteur PID. Nous nous contenterons ici de choisir un correcteur de type **PI** (proportionnel et intégral) avec un gain proportionnel de 400 et un gain intégral de 50 pour annuler l'erreur statique. Nous reviendrons sur l'utilisation de ce bloc qui possède de nombreuses fonctionnalités très intéressantes pour le contrôle commande des systèmes. Des méthodes permettant de choisir la valeur de ces coefficients sont présentées en fin d'ouvrage.

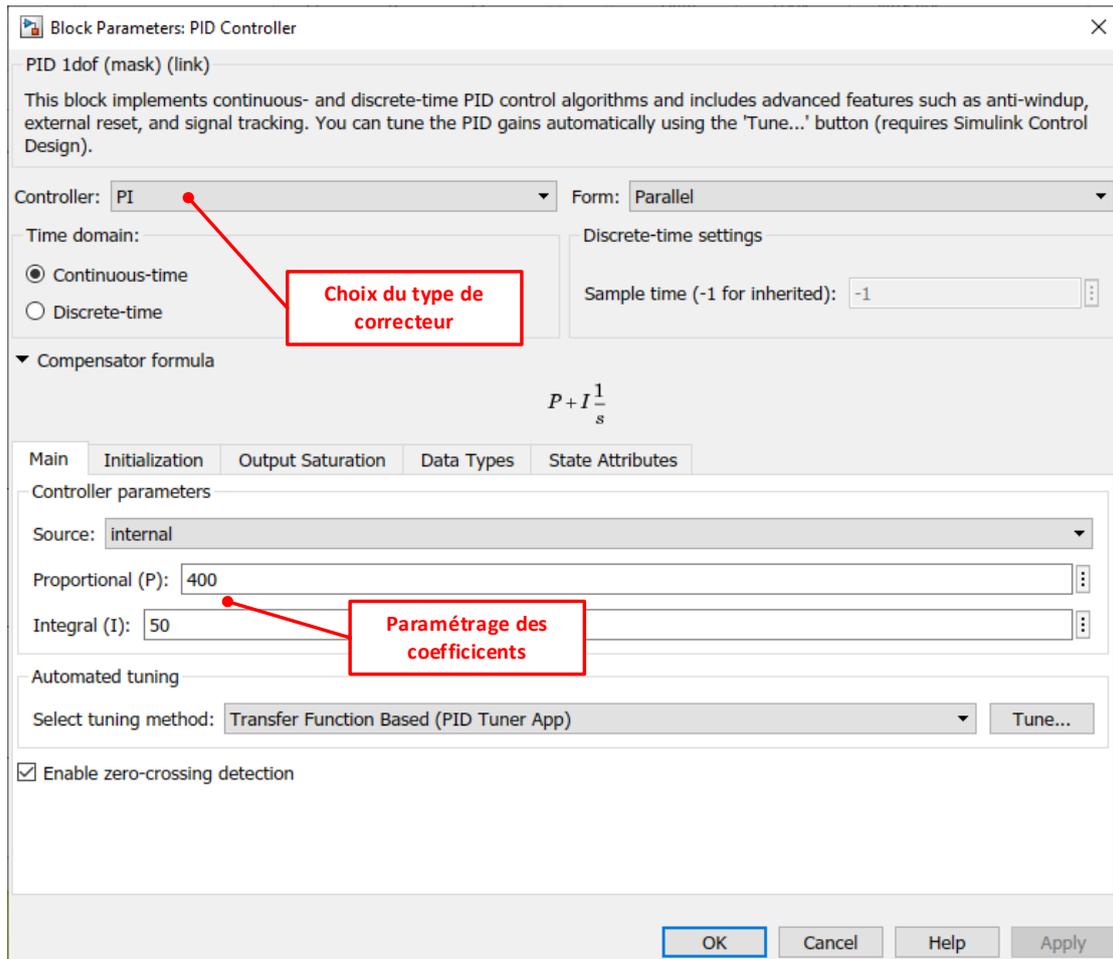
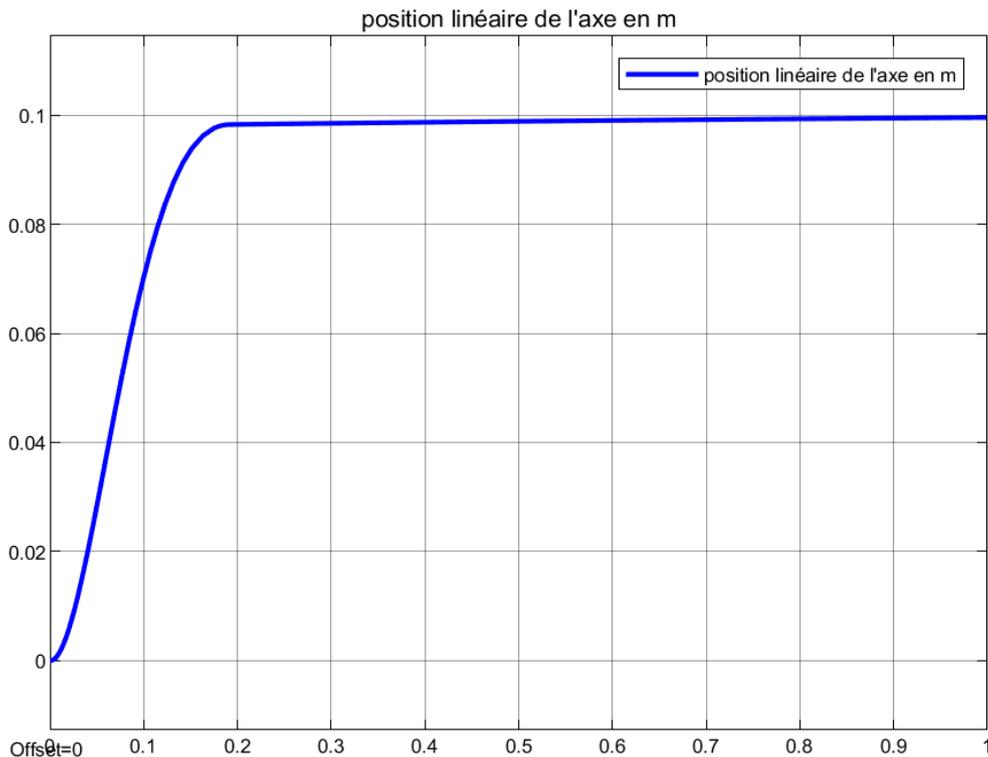


Figure 161 : paramétrage du bloc PID

**Lancer** la simulation.

**Visualiser** la réponse en position avec correction proportionnelle et intégrale.



**Figure 162 : réponse corrigée de l'asservissement en position de l'axe linéaire**

Nous constatons l'annulation de l'erreur statique en raison de la présence d'un terme intégrateur dans le PID.

La perturbation engendrée par le frottement sec est compensée par le correcteur PI pour une consigne de 0.1 m.

## B. Domaines hydraulique- vérin hydraulique simple effet

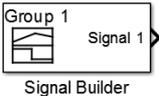
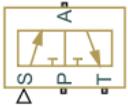
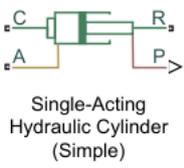
Le système modélisé est un vérin hydraulique simple effet commandé par un distributeur proportionnel 3/2 alimenté par une source de pression constante. Le déplacement de la tige entraîne le déplacement d'une masse qui agit sur un ensemble ressort+amortisseur. L'objectif est d'estimer l'évolution de la vitesse et de la position de la tige en fonction du temps.

Le système étudié fait intervenir 2 domaines physiques :

- Domaine hydraulique
- Domaine mécanique

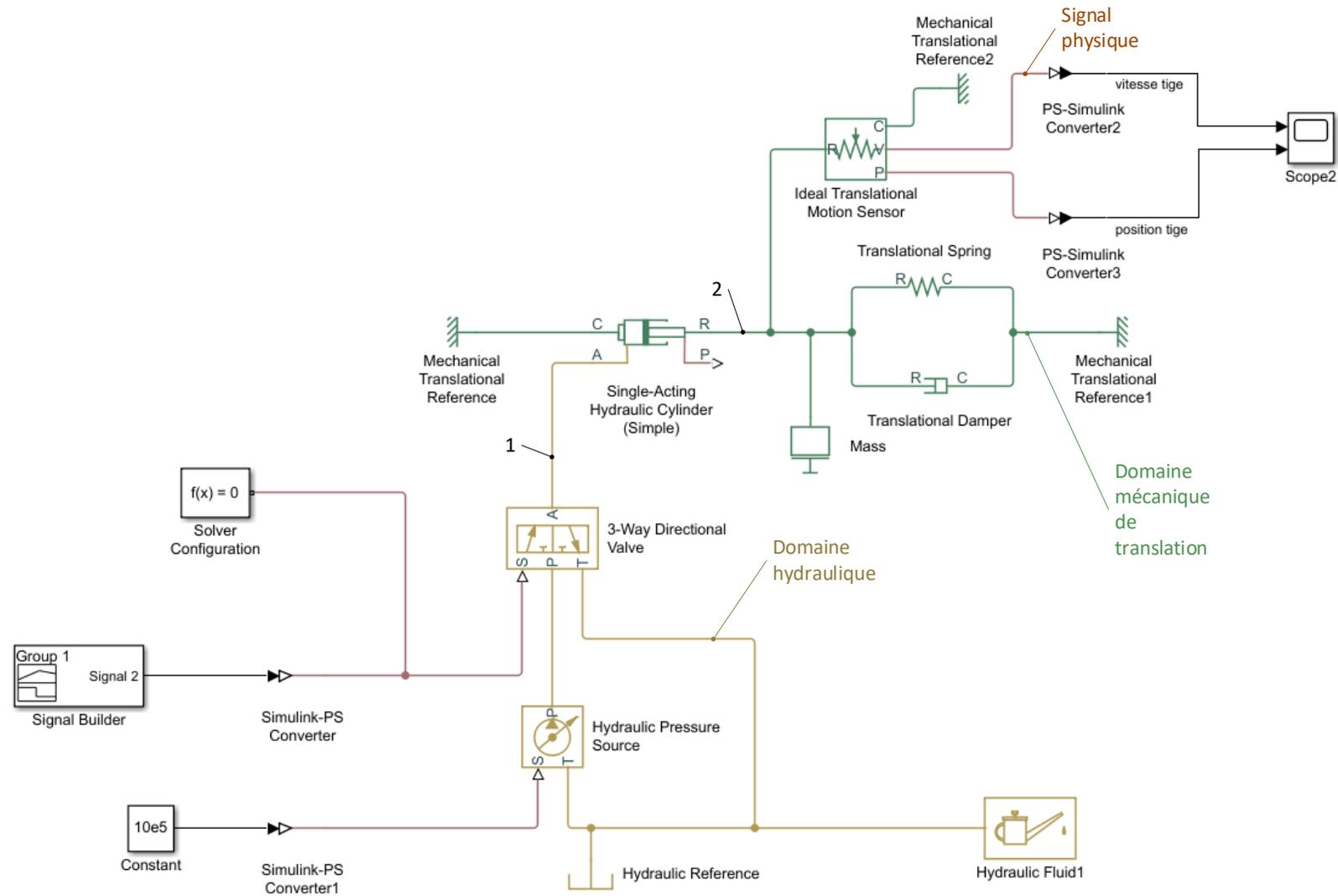
### 1. Choix des composants

Seuls sont référencés dans le tableau de la Figure 163 les composants qui n'ont pas encore été utilisés auparavant dans le document.

Fonction du composant	Représentation	Bibliothèque
Source de signal paramétrable	 Signal Builder	Simulink/Sources
Source de pression hydraulique	 Hydraulic Pressure Source	Simscape/Foundation Library/Hydraulic/Sources
Distributeur proportionnel 3/2	 3-Way Directional Valve	Simscape/Fluids/Hydraulics (Isothermal)/Valves/Directional Valves
Référence hydraulique	 Hydraulic Reference	Simscape/Foundation Library/Hydraulic/Hydraulic elements
Propriétés du fluide	 Hydraulic Fluid	Simscape/Fluids/ Hydraulics (Isothermal)/Hydraulic Utilities
Vérin hydraulique simple effet	 Single-Acting Hydraulic Cylinder (Simple)	Simscape/Fluids/ Hydraulics (Isothermal)/Hydraulic Cylinders
Ressort en translation	 Translational Spring	Simscape/Foundation Library/ Mechanical/Translational Elements
amortisseur en translation	 Translational Damper	Simscape/Foundation Library/ Mechanical/Translational Elements
masse	 Mass	Simscape/Foundation Library/ Mechanical/Translational Elements

**Figure 163 : les composants nécessaires à la modélisation de la commande d'un vérin hydraulique**

Le modèle Simscape du vérin hydraulique simple effet est représenté sur la Figure 164 et permet de visualiser les différents domaines physiques qui interviennent dans le modèle. La correspondance entre les connexions du domaine physique et les éléments réels est donnée.



**Correspondance physique des connexions:**

- 1: canalisation hydraulique (domaine hydraulique)
- 2: tige du vérin (domaine mécanique de translation)

**Figure 164 : visualisation des domaines physiques intervenant dans la modélisation de la commande du vérin hydraulique**

## 1. Placement et assemblage des composants

Dans la fenêtre **Simulink Library Browser**, cliquer sur  pour créer un nouveau fichier.

**Glisser/Déposer** les différents blocs à partir des bibliothèques, les disposer dans la fenêtre de travail et les relier comme pour obtenir la configuration de la Figure 165.

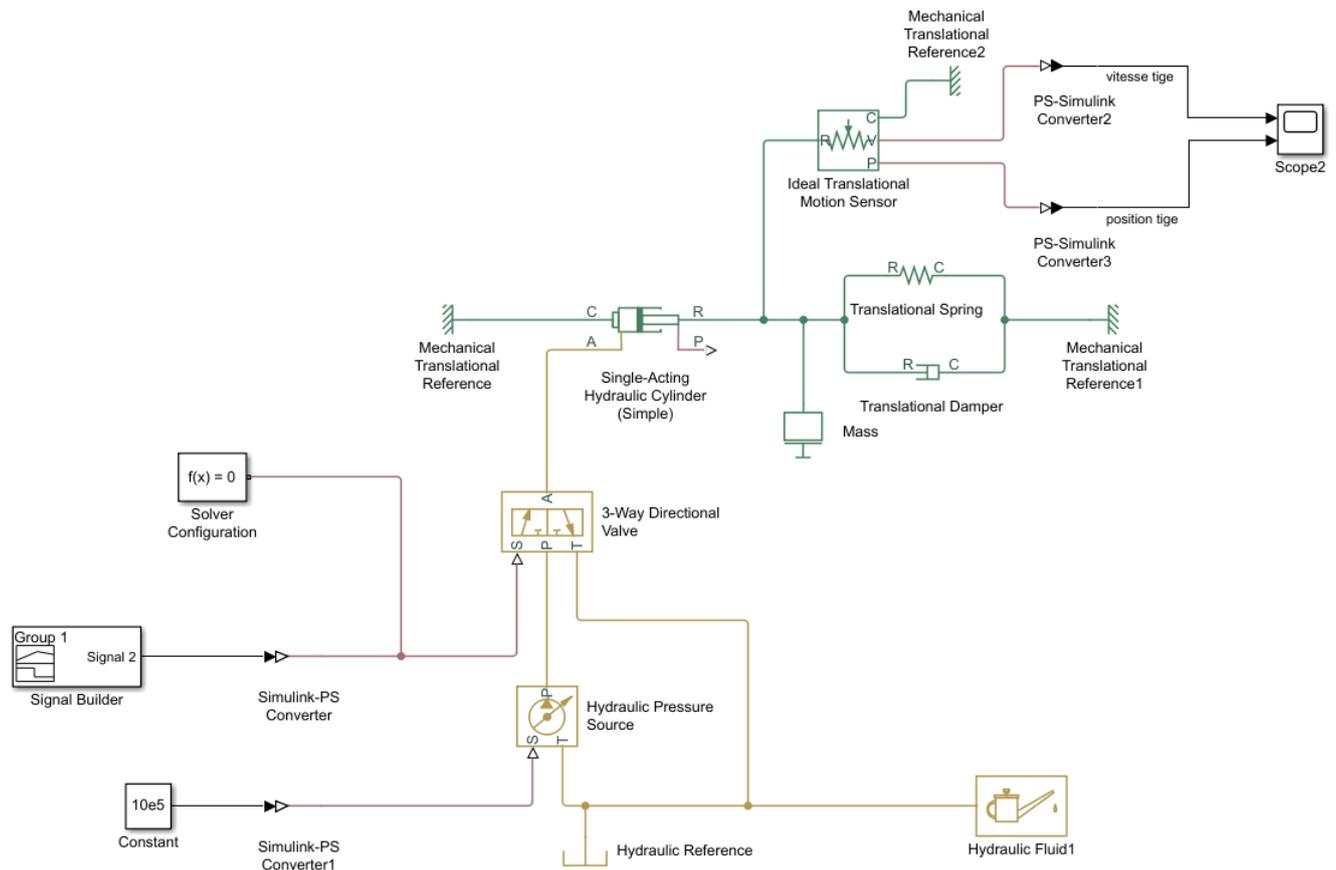


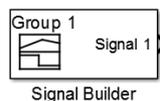
Figure 165 : modèle Simscape de la commande du vérin hydraulique

## 2. Paramétrage des composants

Effectuer le paramétrage des différents blocs conformément aux informations données ci-dessous.

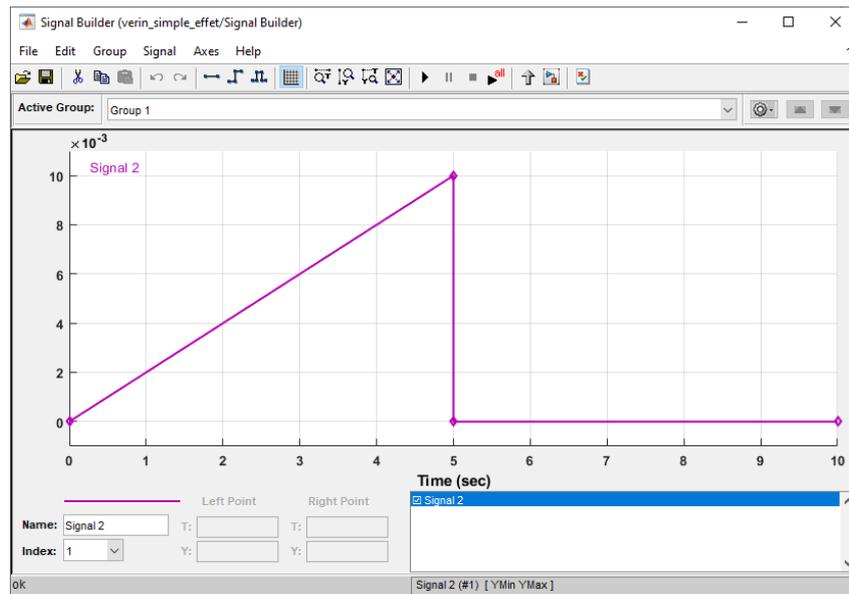
### Paramétrage

Signal Builder



Simulink/Sources

Ce bloc permet de créer un signal de commande quelconque. Nous souhaitons ici commander le distributeur de manière linéaire de la valeur 0 à la valeur 0.01 correspondant à l'ouverture maximale de l'orifice d'alimentation de l'actionneur. A partir de  $t=5$  s, le signal de commande est ramené à 0. Se reporter à l'annexe « **Utilisation du signal builder** » pour paramétrer le signal.



## Paramétrage

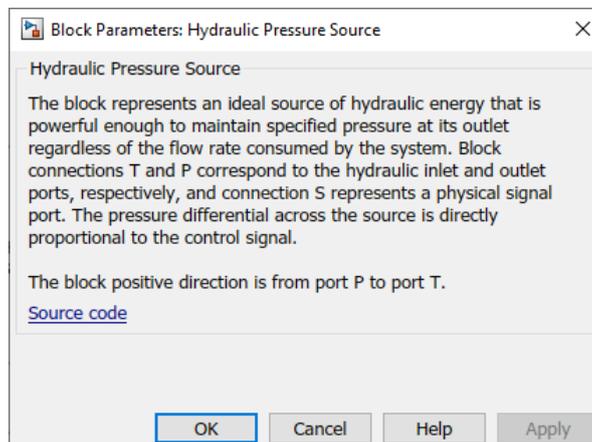
Hydraulic Pressure Source



Hydraulic Pressure Source

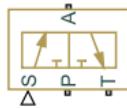
Simscape/Foundation Library/Hydraulic/Sources

Ce composant permet de modéliser une source de pression parfaite. La pression sera constante quelles que soient les conditions de fonctionnement. Il possède 2 ports (**T** et **P**) de type **PCP** du domaine hydraulique et un port de type signal physique qui indique la valeur de la pression. Ici la pression sera réglée à  $10^6$  Pa. Le bloc « **Constant** » alimente cette source de pression par l'intermédiaire d'un bloc **S-PS** dont l'unité sera réglée sur **Pa**.



## Paramétrage

### 3 - Way Directional Valve



3-Way Directional Valve

Simscape/Fluids/Hydraulics  
(Isothermal)/Valves/Directional Valves

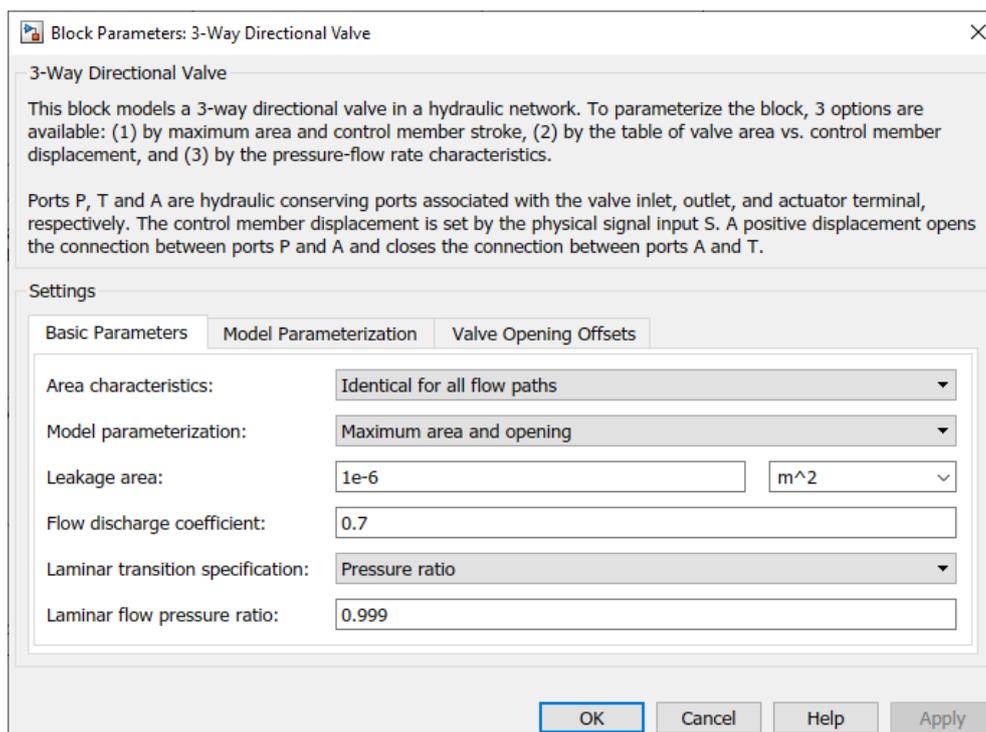
Ce composant permet de modéliser un distributeur hydraulique 3/2 à commande proportionnelle. Il possède 3 ports de type **PCP (T, P et A)** du domaine hydraulique et un port **S** de type signal physique qui permet de commander le distributeur.

- Port A : sortie vers actionneur
- Port P : alimentation
- Port T : échappement

La fenêtre de paramétrage de ce composant possède 3 onglets :

- Basic Parameters
- Model Parameterization
- Valve Opening Offsets

L'onglet **Basics Parameters** permet de spécifier Le paramétrage peut se faire de différentes manières en fonction des données constructeur disponibles. Ici le paramétrage est fait à partir de **By maximum area and opening**.

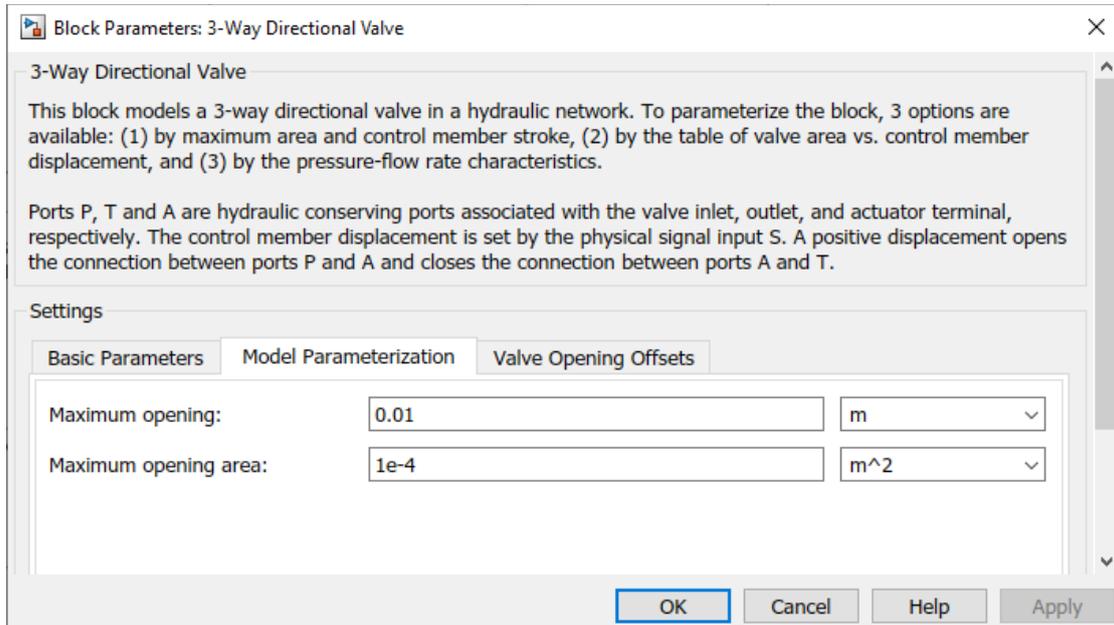


**Leakage area** : surface correspondant à un léger débit de fuite si la vanne est complètement fermée.

**Flow discharge coefficient** : coefficient empirique qui dépend des formes de la servo-valve et qui est donné dans les données constructeur (valeur par défaut 0.7)

**Laminar transition Pressure Ratio** : rapport de la pression de sortie sur la pression d'entrée auquel l'écoulement passe de laminaire à turbulent

L'onglet **Model Parametrization** permet de spécifier les paramètres d'ouverture et fermeture de la valve.



**Maximum opening** : déplacement maximal de l'axe de commande de la valve

**Maximum opening area**: diamètre d'ouverture maximal des orifices de la valve

L'onglet **Valves opening Offsets** permet de spécifier les ouvertures initiales des différents orifices (en début de simulation)

**Between P-A** ( $h_{PA0}$ ) : ouverture initiale de l'orifice P-A

**Between A-T** ( $h_{AT0}$ ) : ouverture initiale de l'orifice A-T

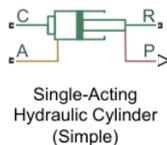
Si on appelle  $x$  la commande qui arrive sur le port S, les ouvertures respectives des orifices seront :

$$h_{PA} = h_{PA0} + x$$

$$h_{AT} = h_{AT0} - x$$

## Paramétrage

Single-Acting Hydraulic Cylinder (Simple)

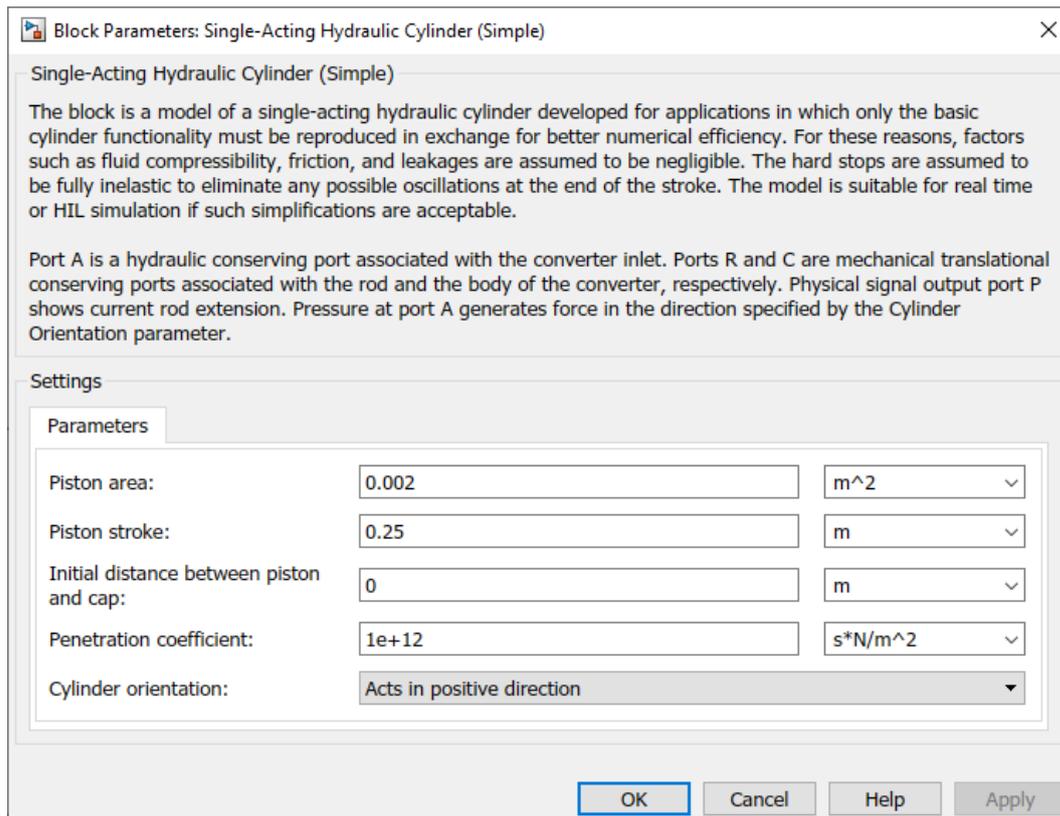


Simscape/Fluids/ Hydraulics (Isothermal)/Valves/Directional Valves

Ce composant permet de modéliser un vérin hydraulique simple effet. Il possède 3 ports de type **PCP**, un du domaine hydraulique (**A**) et deux du domaine mécanique de translation (**C** et **R**).

- Port A : orifice d'alimentation hydraulique du vérin
- Port C : associé au corps du vérin
- Port R : associé à la tige du vérin

Le port **P** est un signal physique qui représente le déplacement de la tige (dans notre exemple nous utiliserons un capteur et ce port ne sera pas utilisé)



**Piston area** : surface du piston

**Piston stroke** : course du piston

**Initial distance between piston and cap** : position initiale de la tige du vérin (0 si le vérin est complètement rentré au début de la simulation).

**Cylinder orientation** : permet de choisir l'orientation de l'effort généré en bout de tige.

## Paramétrage

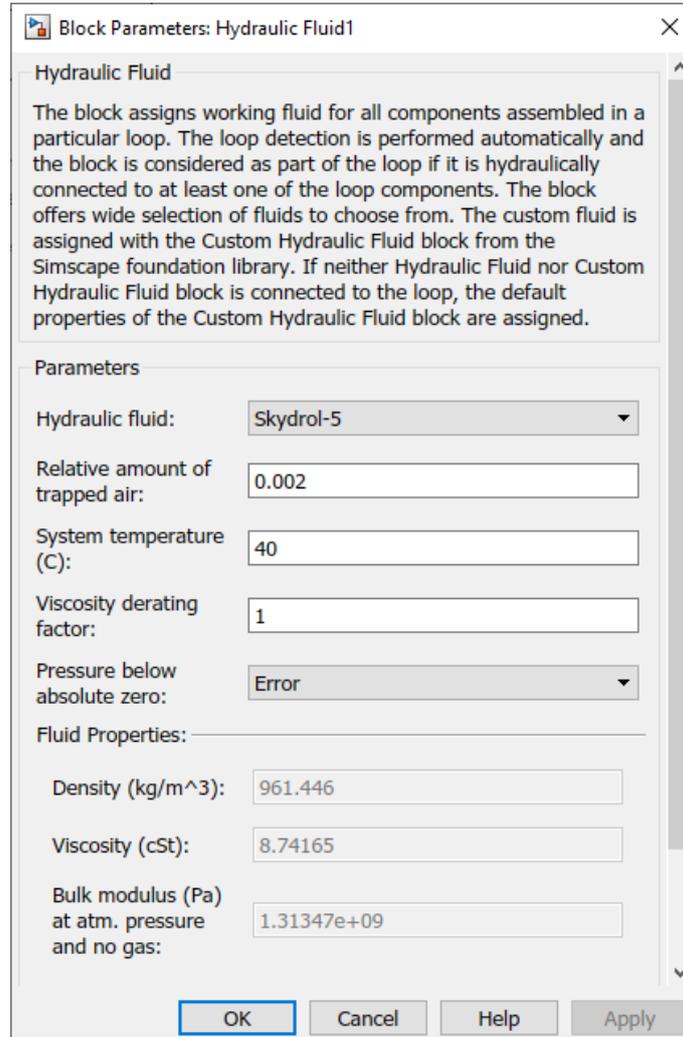
Hydraulic fluid



Simscape/Fluids/ Hydraulics  
(Isothermal)/Hydraulic Utilities

Ce composant permet de modéliser les propriétés du fluide utilisé.

Un bloc du même type existe dans la bibliothèque Simscape/Foundation Library/Hydraulics/Hydraulics Utilities qui permet de rentrer manuellement les caractéristiques du fluide si cela est nécessaire.



**Block Parameters: Hydraulic Fluid1**

**Hydraulic Fluid**

The block assigns working fluid for all components assembled in a particular loop. The loop detection is performed automatically and the block is considered as part of the loop if it is hydraulically connected to at least one of the loop components. The block offers wide selection of fluids to choose from. The custom fluid is assigned with the Custom Hydraulic Fluid block from the Simscape foundation library. If neither Hydraulic Fluid nor Custom Hydraulic Fluid block is connected to the loop, the default properties of the Custom Hydraulic Fluid block are assigned.

**Parameters**

Hydraulic fluid: Skydrol-5

Relative amount of trapped air: 0.002

System temperature (C): 40

Viscosity derating factor: 1

Pressure below absolute zero: Error

**Fluid Properties:**

Density (kg/m<sup>3</sup>): 961.446

Viscosity (cSt): 8.74165

Bulk modulus (Pa) at atm. pressure and no gas: 1.31347e+09

OK Cancel Help Apply

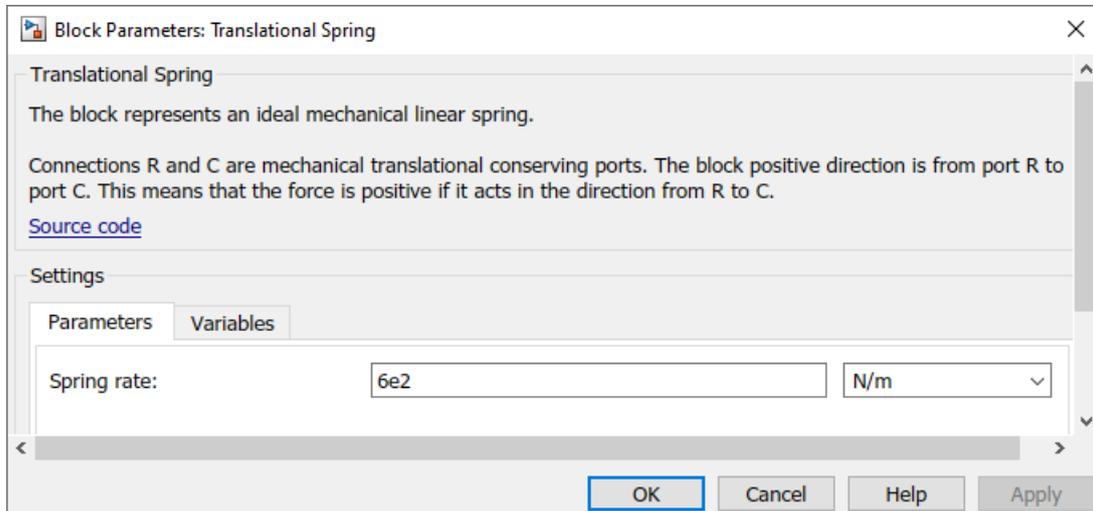
## Paramétrage

Translational Spring



Simscape/Foundation Library/  
Mechanical/Translational Elements

Ce composant modélise un ressort linéaire, il faut spécifier la raideur et l'allongement initial.



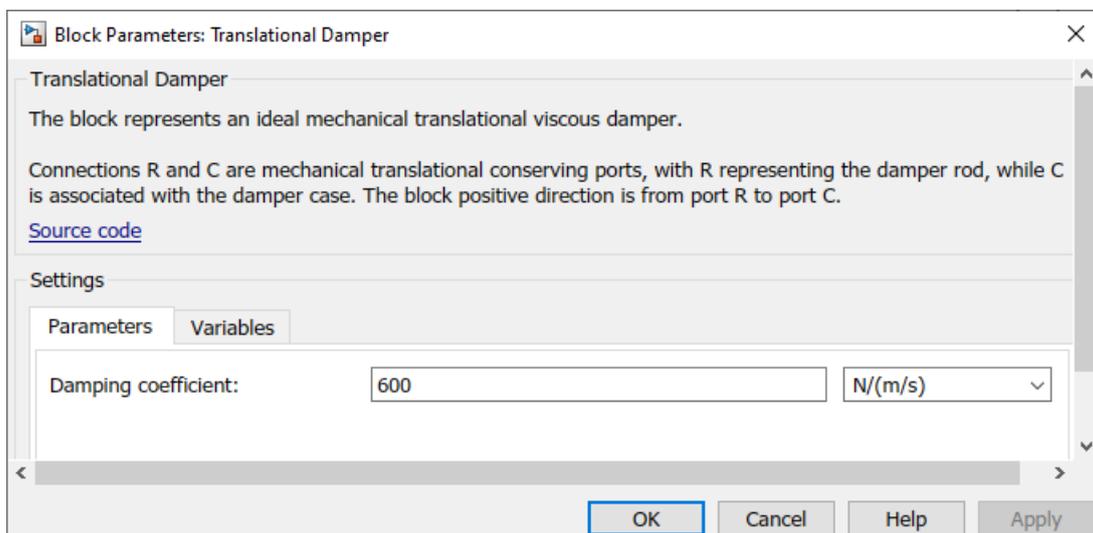
## Paramétrage

amortisseur en  
translation



Simscape/Foundation Library/  
Mechanical/Translational Elements

Ce composant modélise un amortisseur, il faut spécifier le coefficient d'amortissement.



## Paramétrage

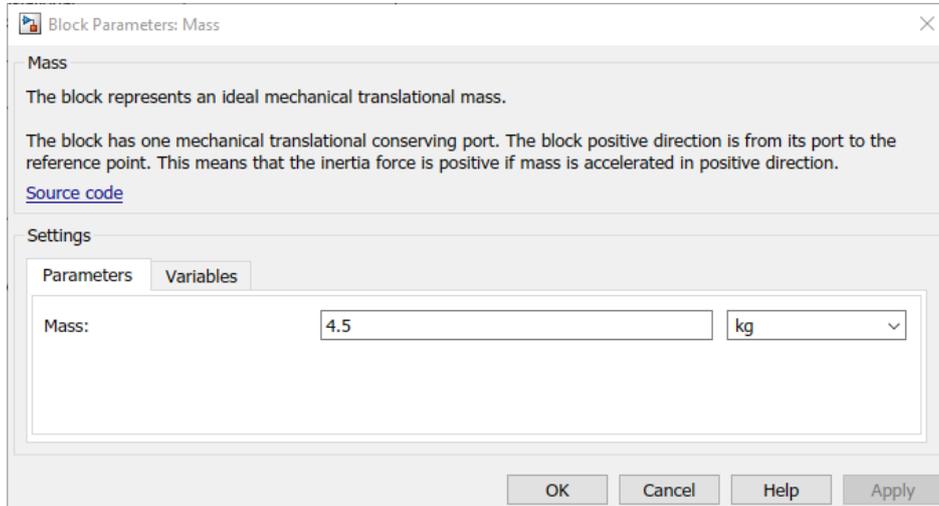
---

Mass



Simscape/Foundation Library/  
Mechanical/Translational Elements

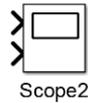
Le paramétrage de ce bloc se limite à indiquer la valeur de la masse.



## Paramétrage

---

Scope



Simulink/Sinks

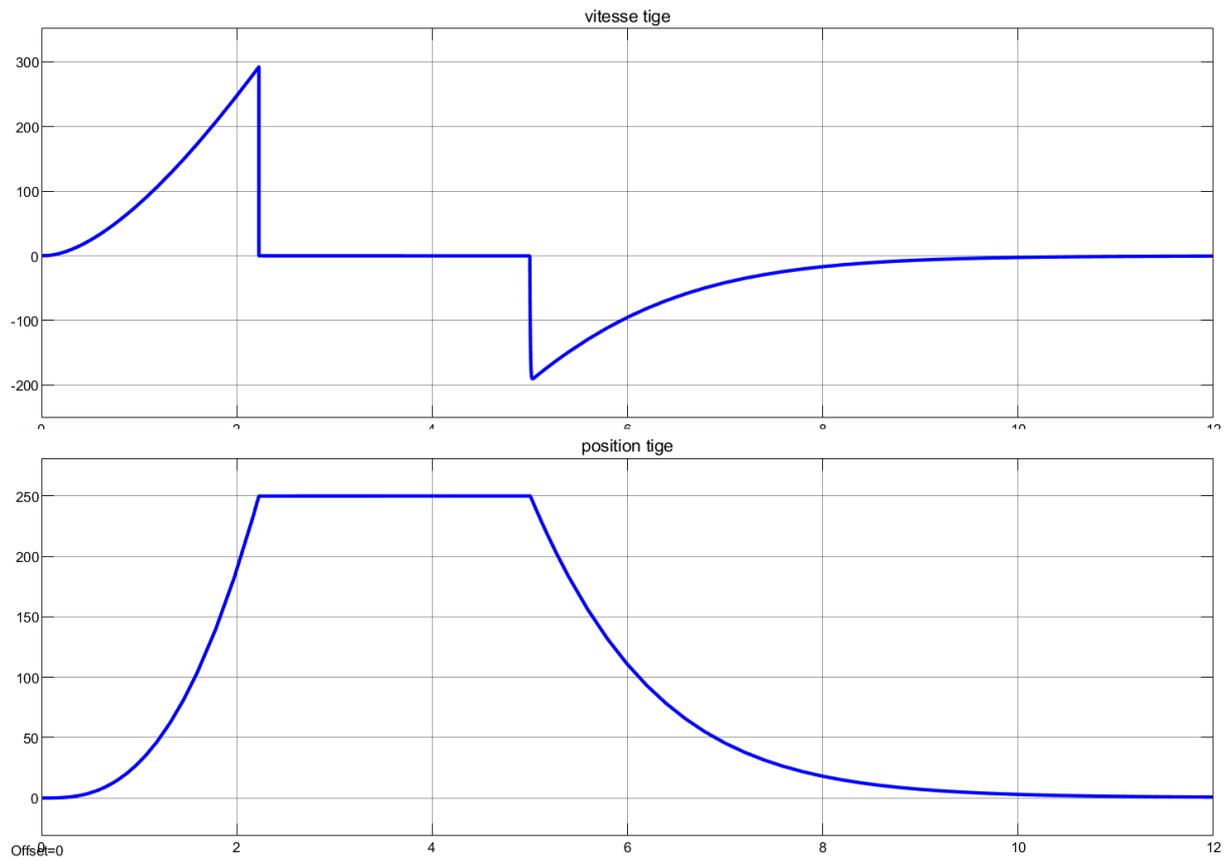
Pour paramétrer un Scope à deux entrées, se reporter à l'annexe paramétrage des **Scopes**.

Le fichier contenant le modèle paramétré est disponible sous le nom **verin\_simple\_effet\_pression.slx**.

### 3. Simulation

**Choisir** le solveur **ode23t**.

**Lancer** la simulation en spécifiant un temps de simulation de **10 s**.



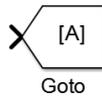
**Figure 166 : évolution de la vitesse et de la position de la tige du vérin**

Nous observons sur la Figure 166 que la tige du vérin sort, jusqu'à atteindre sa course maximale à  $t=2.3s$ . La tige est à l'arrêt jusqu'à la disparition de la commande à  $t=5s$ . La tige revient ensuite en position initiale sous l'action du ressort.



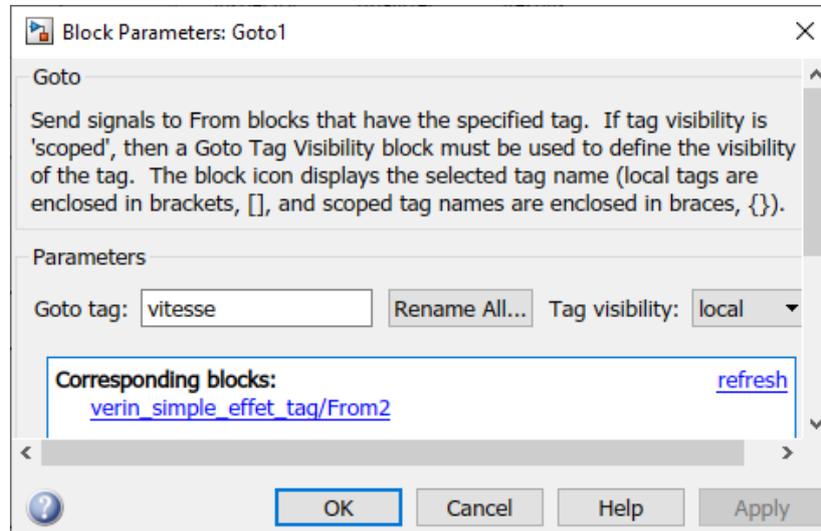
## Paramétrage

Goto



Simulink/Signal Routing

Ce bloc permet de mettre un **Tag** sur un signal et de récupérer ce signal à l'aide d'un bloc **From**.

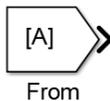


**Goto tag** : indique le nom que l'on donne au signal

**Tag visibility** : indique si le **Tag** est **Local** (uniquement exploitable pour le sous-système dans lequel il est créé) ou **Global** (exploitable dans tout le système)

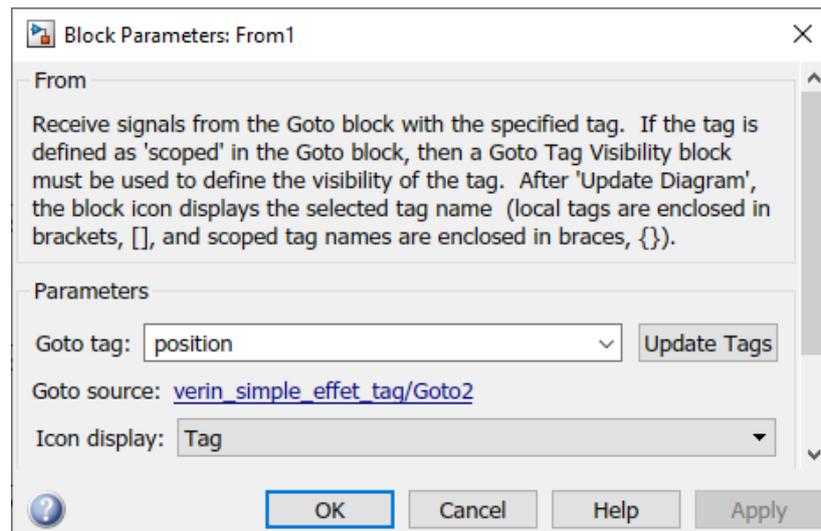
## Paramétrage

From



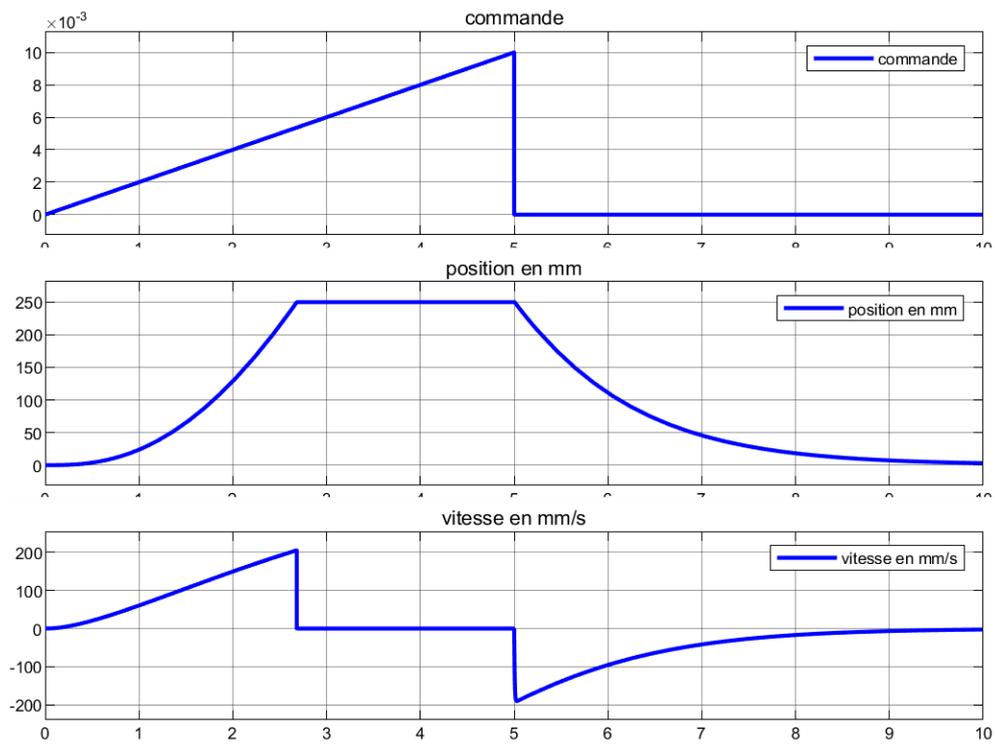
Simulink/Signal Routing

Ce bloc permet de récupérer le Tag d'un signal créé à l'aide d'un bloc **Goto**.



Le choix du Tag se fait directement à partir du menu déroulant. La commande **Update Tags** permet de rafraîchir les Tags afin de faire apparaître tous les Tags dans le menu déroulant.

En double-cliquant sur le scope, il est possible de visualiser l'ensemble des signaux.

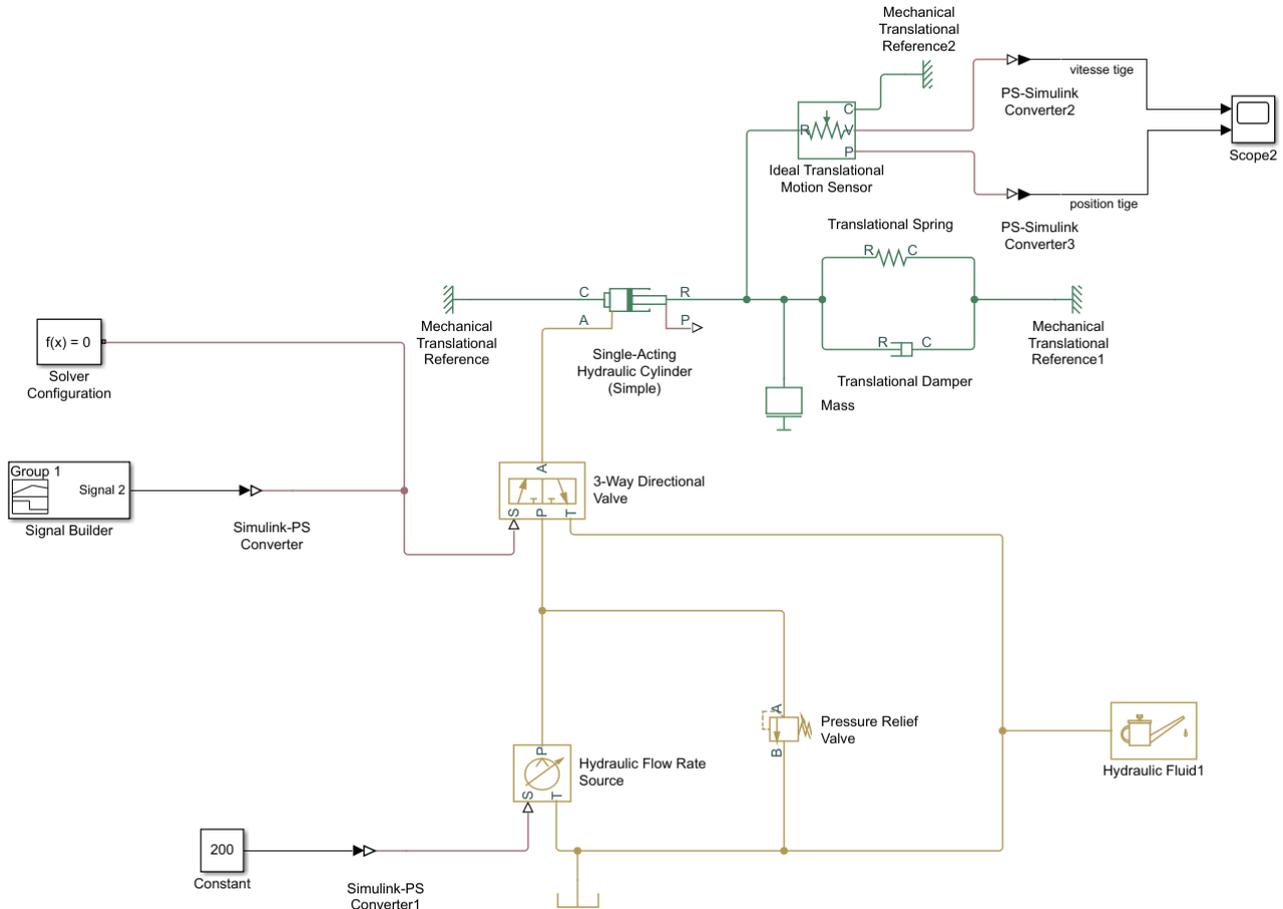


**Figure 168 : visualisation des résultats de la simulation**

## 5. Remplacement de la source de pression par une source de débit

Il est possible de remplacer la source de pression par une source de débit. Dans ce cas, il faudra prévoir impérativement une soupape de décharge afin que le fluide puisse s'écouler vers le réservoir lorsque la tige du vérin est en butée (Figure 169).

Ouvrir le fichier **verin\_simple\_effet\_debit.slx**.



**Figure 169 : modèle Simscape de la commande d'un vérin hydraulique avec source de débit**

Le modèle est le même que précédemment. La source de pression a été remplacée par une source de débit de 200 l/min. Une soupape de décharge a été placée sur la canalisation d'alimentation afin de limiter la pression lorsque le vérin arrive en butée.

### Paramétrage

Hydraulic Flow Rate Source



Simscape/Foundation Library/ Hydraulic/ Sources

Ce composant permet de modéliser une source de débit parfaite. Le débit sera constant quelles que soient les conditions de fonctionnement. Il possède 2 ports (**T** et **P**) de type **PCP** du domaine hydraulique et un port de type signal physique qui indique la valeur du débit. Ici le débit sera réglé à 200 l/min. Le bloc « **Constant** » alimente cette source de débit par l'intermédiaire d'un bloc **S-PS** dont l'unité sera réglée sur **l/min**.



Ce composant modélise un limiteur de pression.

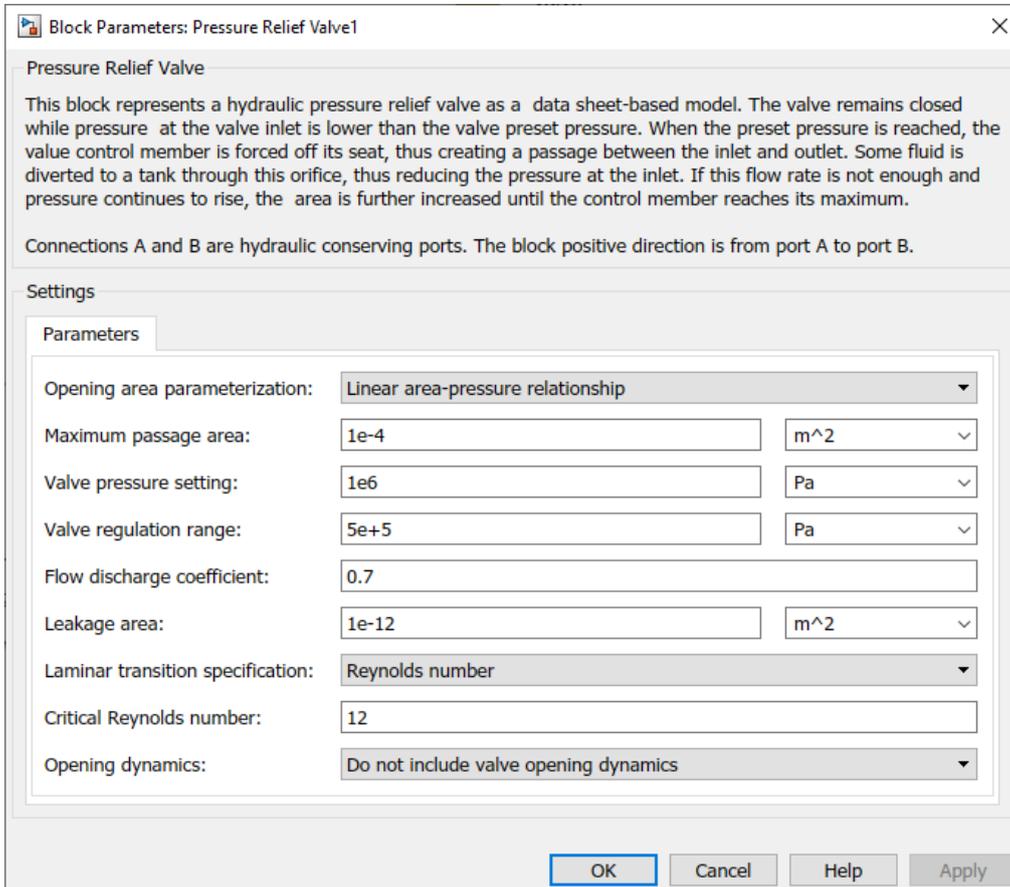


Figure 170 : paramétrage d'un bloc limiteur de pression

**Maximum passage area** : indique la surface maximale de passage du fluide

**Valve pressure setting** ( $p_{set}$ ) : indique la valeur de la pression à partir de laquelle la valve commence à s'ouvrir.

**Valve regulation range** ( $p_{reg}$ ): indique la variation de pression dans la valve entre la position ouverte et la position fermée.

La courbe de la Figure 171 montre la variation de l'ouverture de la valve en fonction de la pression.

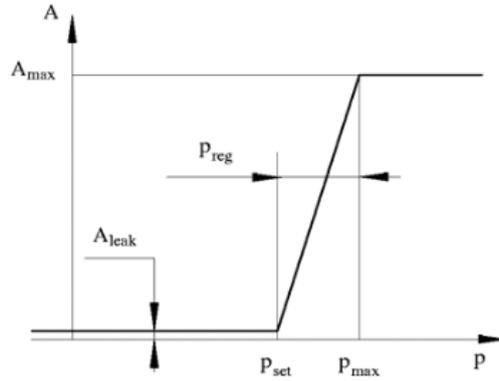


Figure 171 : variation de l'ouverture de soupape en fonction de pression

$$P_{max} = P_{set} + P_{reg}$$

Lancer la simulation et observer l'évolution de la vitesse et de la position de la tige du vérin.

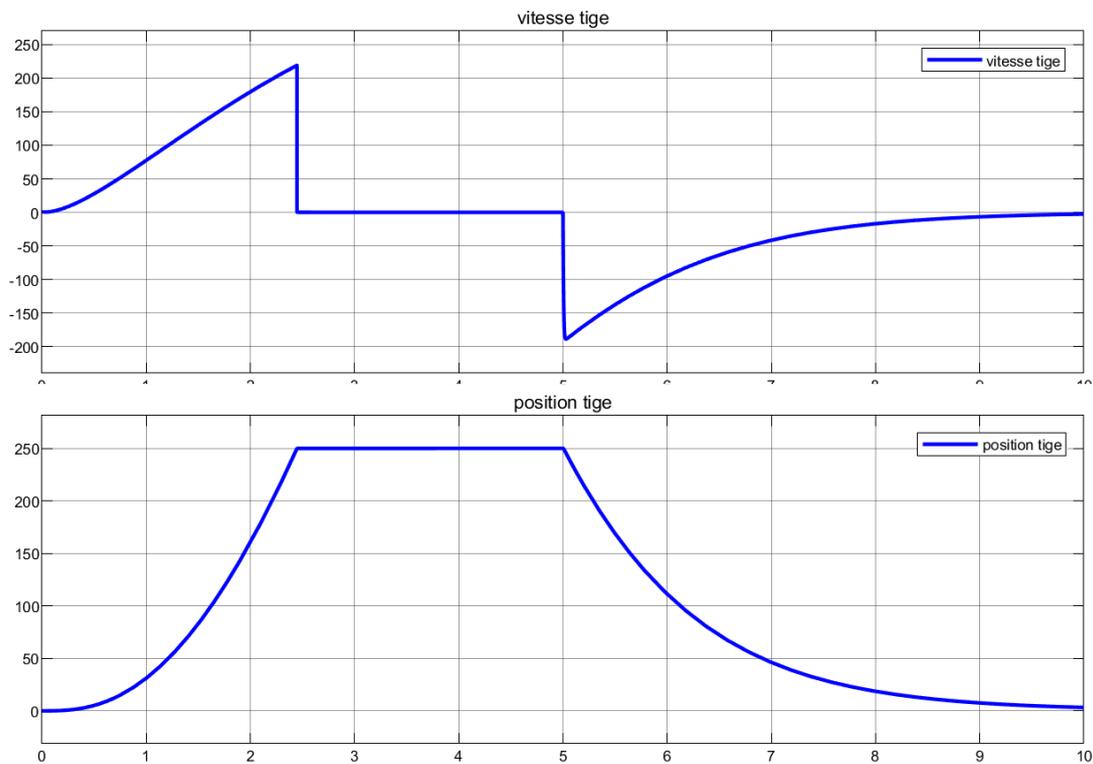
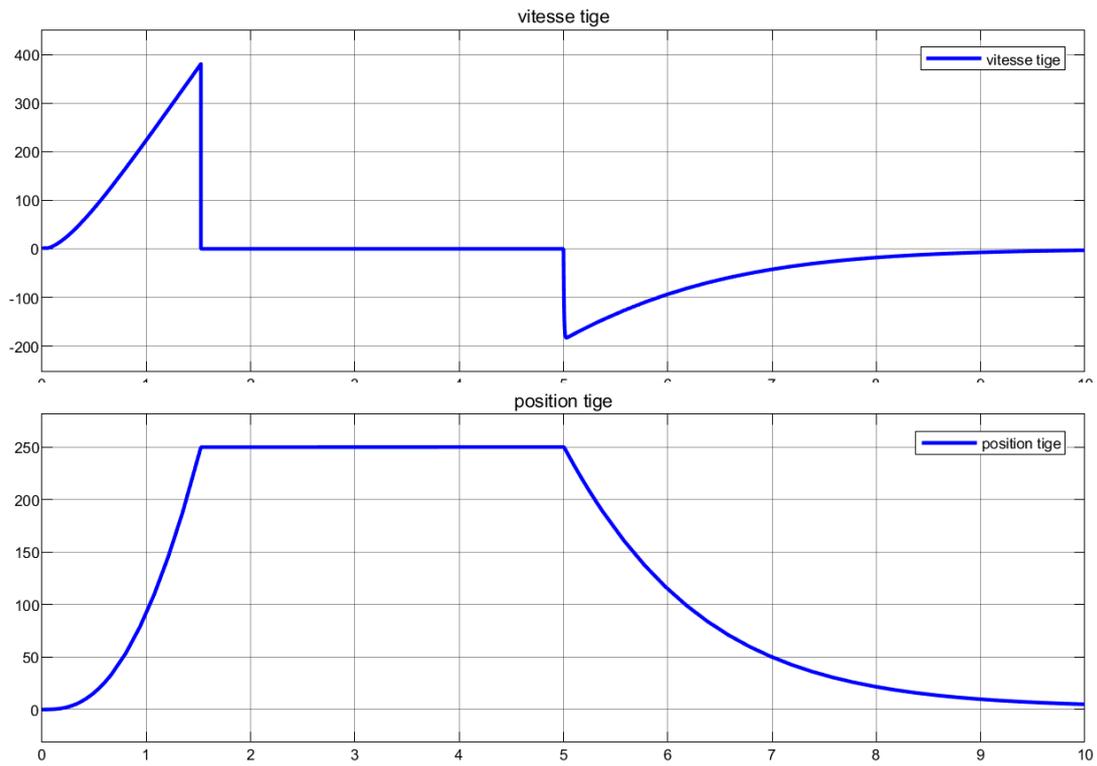


Figure 172 : évolution de la vitesse et de la position de la tige du vérin

On observe que la tige met environ 2.3 s pour atteindre la butée.

Augmenter la valeur de la **Valve pressure setting** dans le limiteur de pression **5e6 Pa** et relancer la simulation.



**Figure 173 : évolution de la vitesse et de la position de la tige du vérin**

On observe bien que l'augmentation de la pression admissible dans la canalisation d'alimentation du vérin permet une sortie de tige plus rapide. Le vérin arrive en butée au bout de 1.5 s.

## C. Domaine électrique –Commande PWM d’un moteur à courant continu

Dans cette partie nous apprendrons à utiliser les composants nécessaires à la conception d’une commande PWM d’un moteur à courant continu.

Une tension de commande PWM est caractérisée par sa fréquence, son rapport cyclique (duty cycle) et son amplitude.

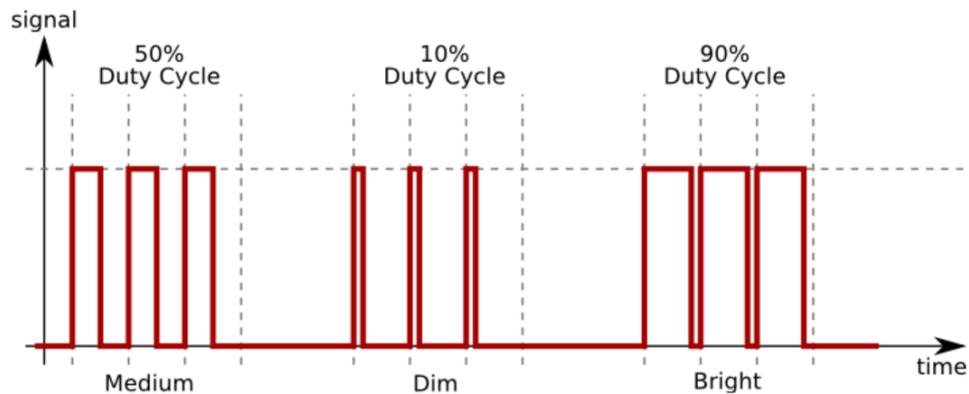
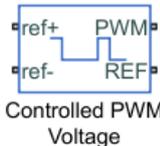
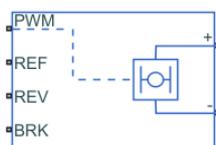


Figure 174 : principe de commande PWM

Pour générer cette tension nous utiliserons le composant « **Controlled PWM Voltage** ».

Pour inverser le sens de rotation du moteur ou le stopper nous utiliseront le composant « **H-Bridge** » (pont en H).

Ces composants sont issus de la bibliothèque **Electrical** de Simscape :

Fonction du composant	Représentation	Bibliothèque
Génération d’une tension de commande de type PWM	 <p>Controlled PWM Voltage</p>	Simscape/Electrical/Integrated Circuits
Pont en H		Simscape/Electrical/Semiconductors&Converters/Converters

## 1. Utilisation du composant « Controlled PWM Voltage »

Ouvrir le fichier **bloc\_PWM.slx**.

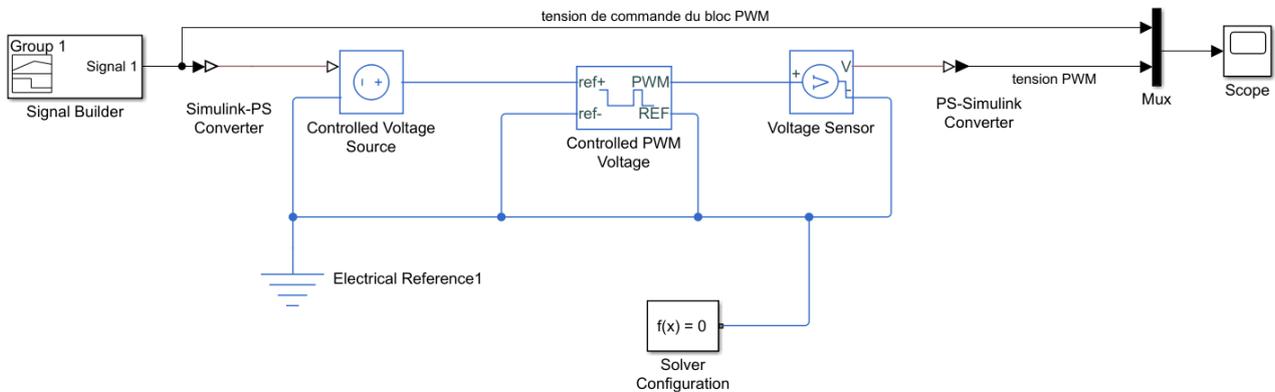


Figure 175 : Illustration du fonctionnement du bloc « Controlled PWM Voltage »

Double-cliquer sur le **Signal Builder** pour observer l'allure de la tension imposée au port +ref (Figure 176).

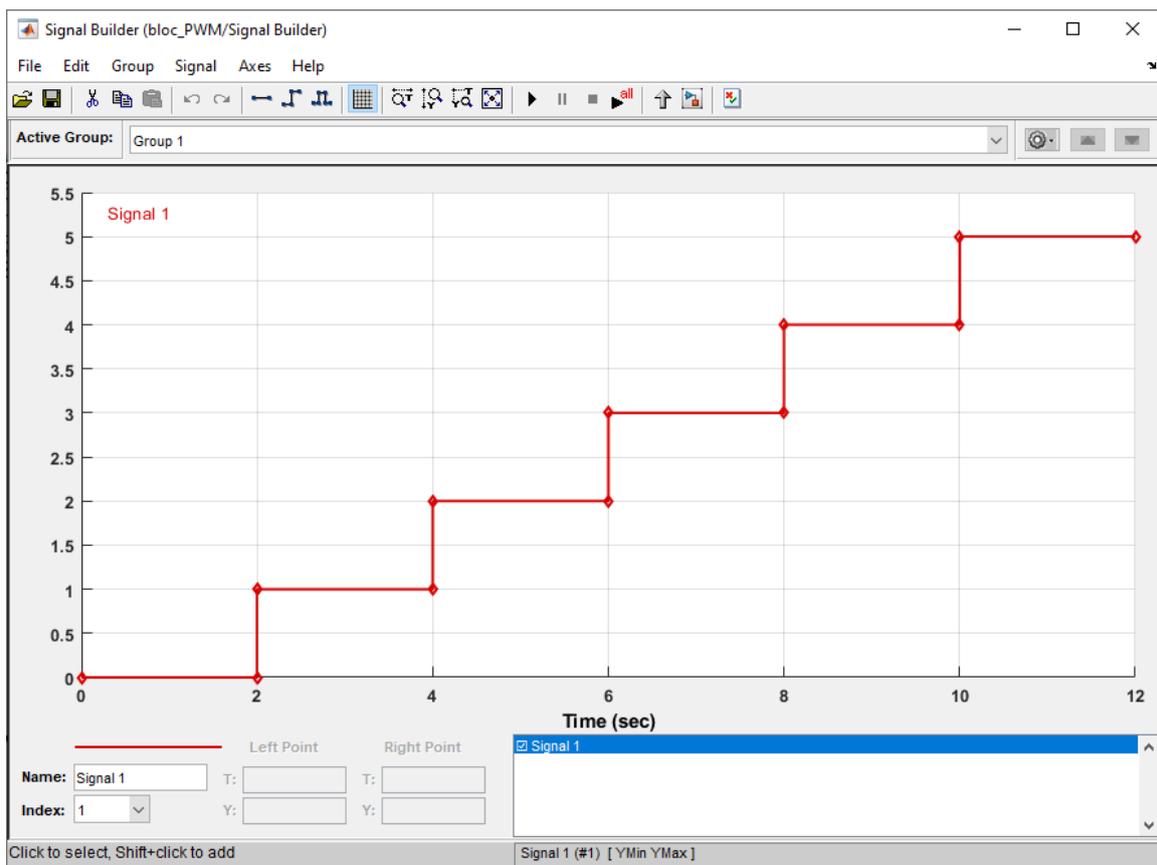
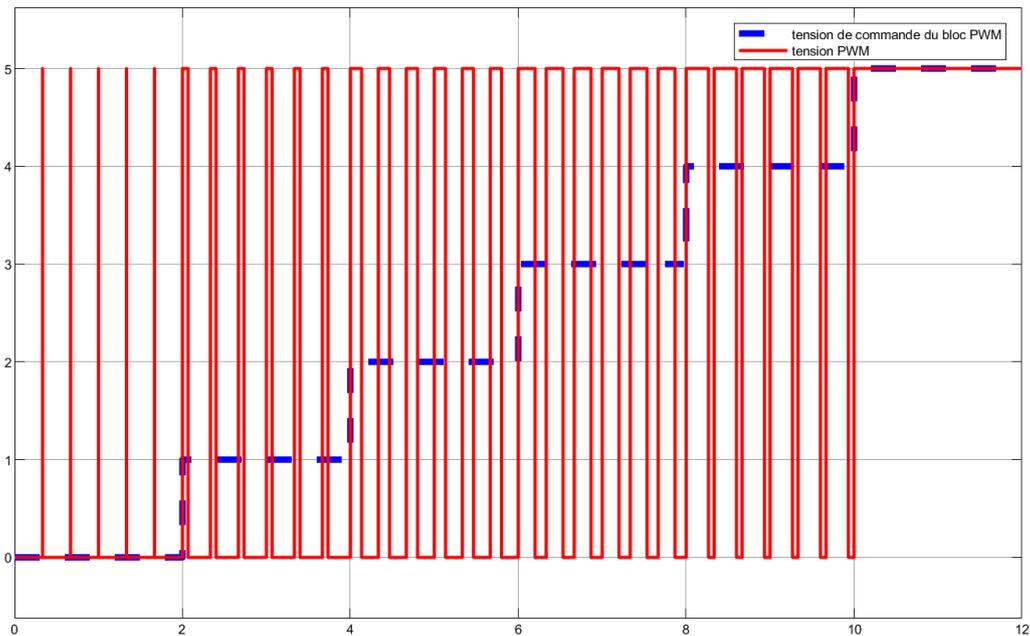


Figure 176 : tension de commande du bloc Controlled PWM Voltage

Cette tension varie de 0 V à 5 V et imposera le rapport cyclique du signal PWM.

**Lancer** la simulation et observer le résultat sur le scope.



**Figure 177 : signal PWM en fonction de la tension de commande**

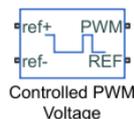
En bleu apparaît le signal de commande imposé sur le port **+ref**, en rouge la tension PWM qui sort sur le port **PWM**.

La fréquence du signal PWM dépend de l'amplitude de la tension **+ref**.

La fréquence et l'amplitude sont imposées dans le paramétrage du composant.

### Paramétrage

Controlled PWM Voltage  
Source



Simscape/Electronics/Actuators and  
Drivers/Drivers

Ce composant modélise un générateur de tension PWM. Il possède 4 ports de type **PCP** du domaine électrique. Il permet de générer une tension PWM à partir des informations prélevées sur les ports d'entrée et du paramétrage du composant.

- **+ref** : borne + du bloc PWM
- **-ref** : borne - du bloc PWM
- **PWM** : signal de tension PWM dont le rapport cyclique est donné par  $\alpha = 100 \times \left( \frac{V_{ref} - V_{min}}{V_{max} - V_{min}} \right)$  avec  $V_{ref} = ref_+ - ref_-$ . L'amplitude et la fréquence sont spécifiées dans les paramètres du composant.
- **REF** : référence de la tension du signal PWM (souvent relié à la masse)

Le paramétrage de ce composant nécessite de renseigner plusieurs onglets :

- PWM
- Input Scaling
- Output Voltage

## Paramétrage de l'onglet **PWM** :

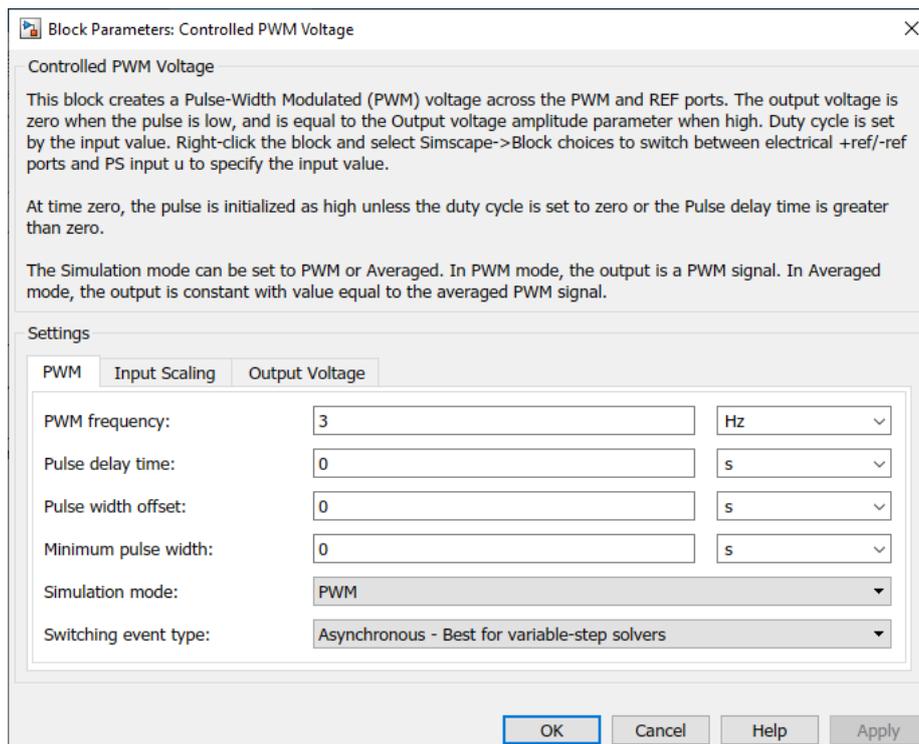


Figure 178 : paramétrage de l'onglet PWM du bloc Controlled PWM Voltage

**PWM frequency** : fréquence du signal PWM

**Pulse delay time** : permet de décaler dans le temps le démarrage de la génération du signal PWM

**Pulse width offset** : permet d'imposer un offset sur la fréquence du signal

**Minimum Pulse Width** : largeur d'impulsion minimale

**Simulation mode** : il est possible de choisir un signal en sortie de type **PWM** ou **Averaged** qui moyenne le signal PWM et permet de limiter le temps de calcul du solveur.

## Paramétrage de l'onglet **Input Scaling** :

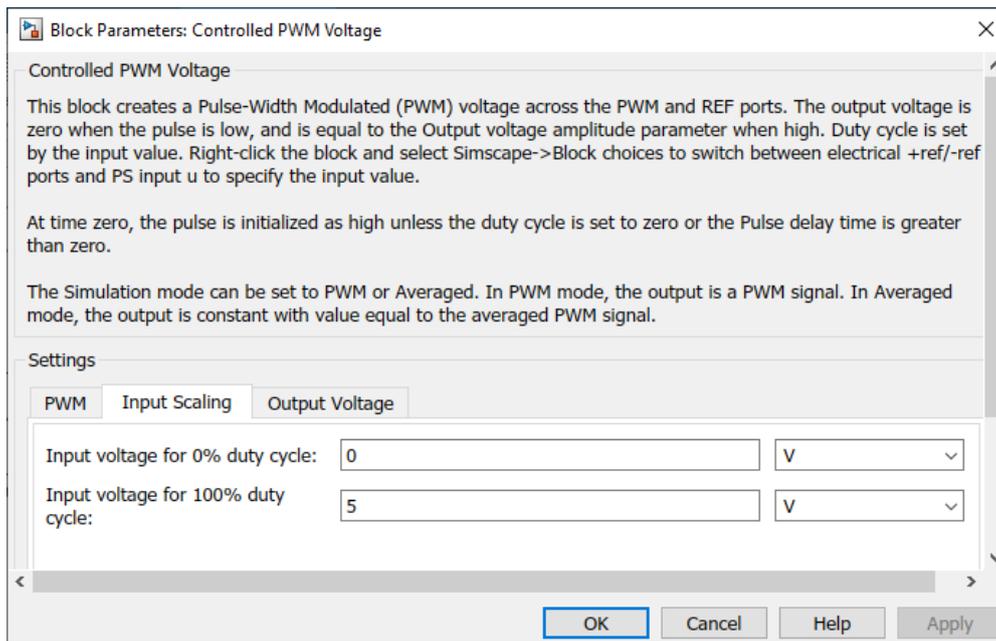


Figure 179 : paramétrage de l'onglet **Input Scaling** du bloc **Controlled PWM Voltage**

**Input value  $V_{min}$  for 0% duty cycle** : valeur de la tension  $V_{ref}$  qui donne un rapport cyclique de 0%.

**Input value  $V_{max}$  for 100% duty cycle** : valeur de la tension  $V_{ref}$  qui donne un rapport cyclique de 100%.

## Paramétrage de l'onglet **Output Voltage**

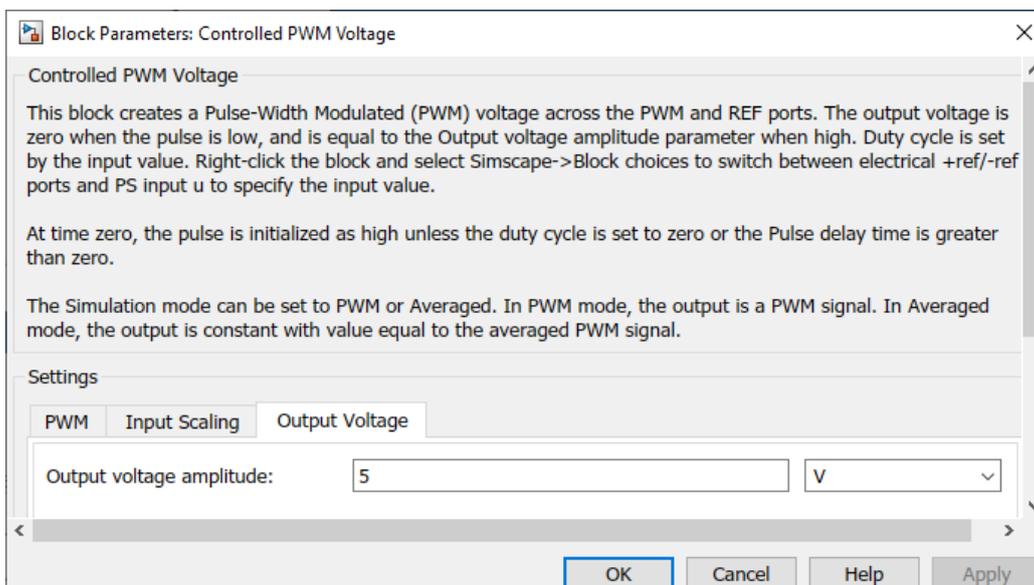


Figure 180 : paramétrage de l'onglet **Output Voltage** du bloc **Controlled PWM Voltage**

**Output voltage amplitude** : amplitude maximale de la tension de sortie

## 2. Commande PWM d'un moteur à courant continu

Il est possible de commander un moteur à courant continu à l'aide du bloc « Controlled PWM Voltage Source ».

Ouvrir le fichier **commande\_moteur\_PWM.slx**.

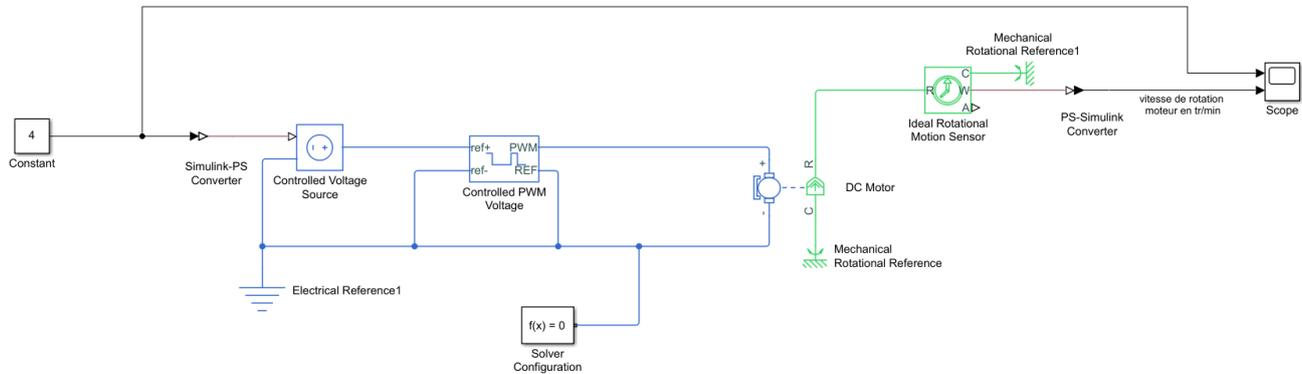


Figure 181 : commande PWM d'un moteur à courant continu

Ce modèle représente un moteur à courant continu commandé par un signal de tension PWM.

Lancer la simulation et observer la vitesse de rotation du moteur à l'aide du scope (Figure 182).

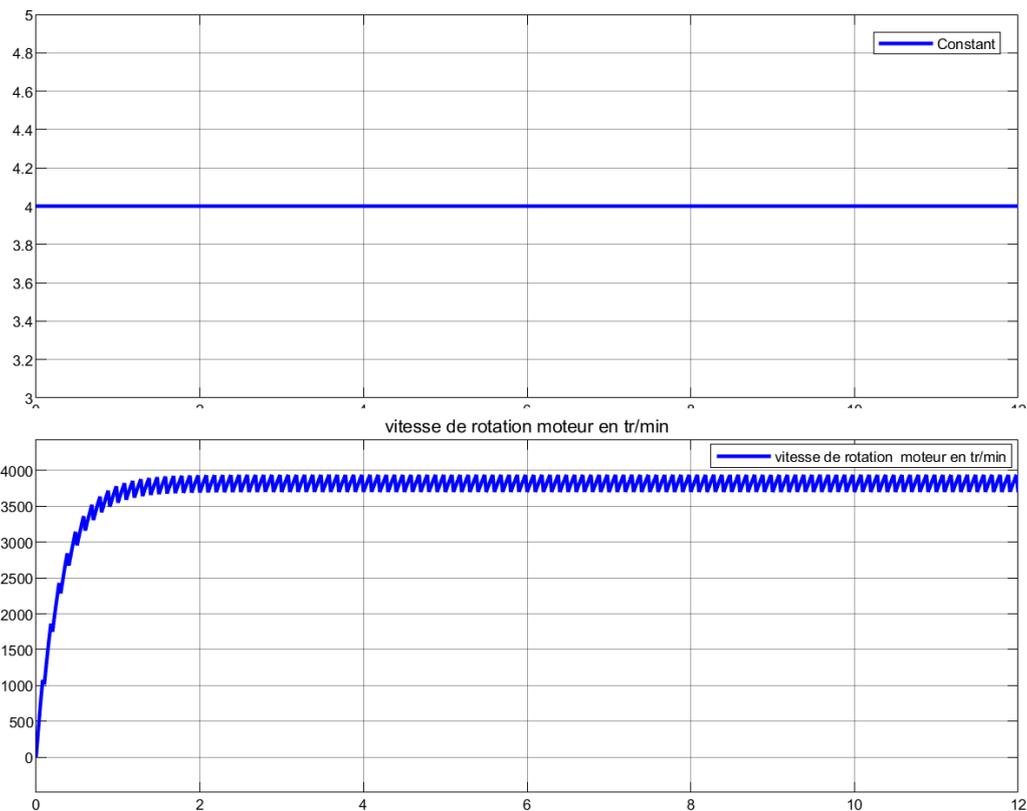


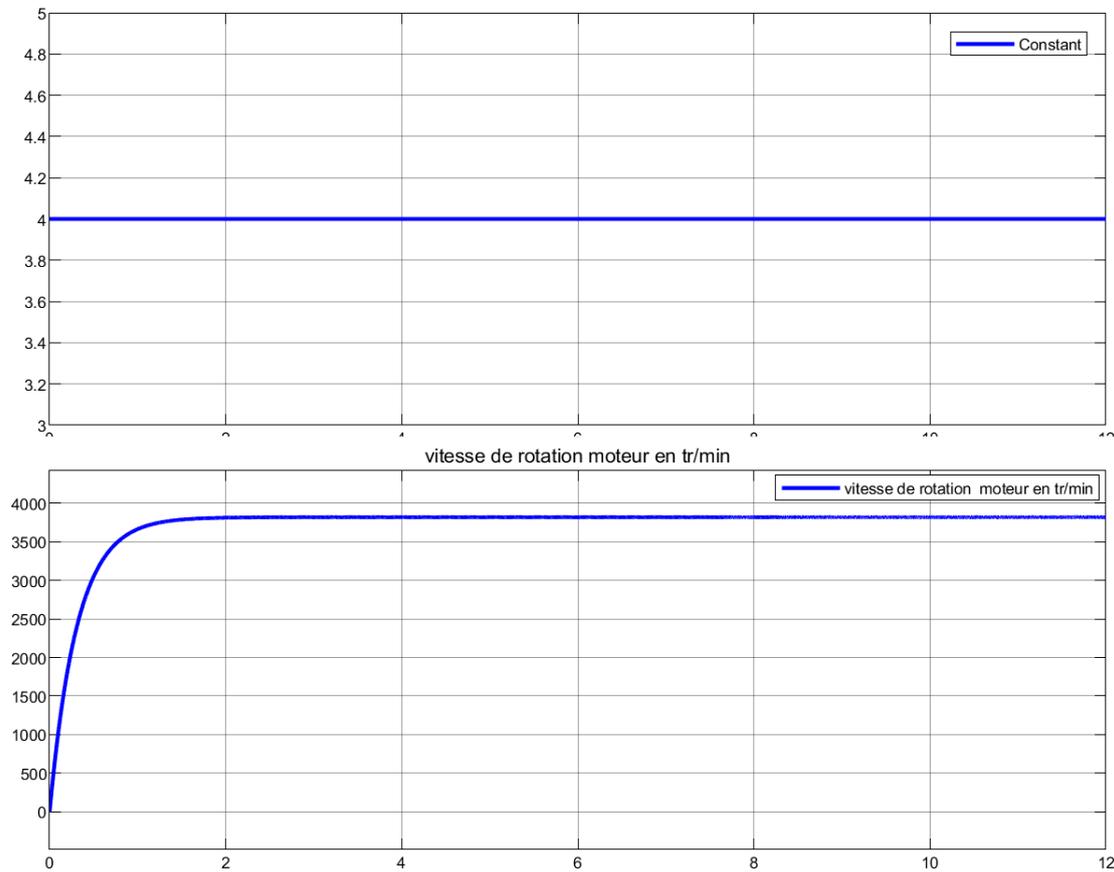
Figure 182 : vitesse de rotation du moteur

La fréquence de la tension PWM est très faible (10 hz). La période du signal est trop proche de la constante de temps du moteur ce qui explique l'allure « hachée » de la vitesse. Pour obtenir une réponse en vitesse

du moteur plus satisfaisante, il faut augmenter la fréquence du signal PWM en modifiant le paramétrage du bloc « Controlled PWM Voltage Source ».

Régler la **PWM Frequency** du bloc sur **1000 hz** et relancer la simulation. Le calcul va prendre plus de temps dans la mesure où la fréquence élevée impose un petit pas de calcul pour le solveur.

**Observer** la vitesse de rotation du moteur (Figure 183).



**Figure 183 : vitesse de rotation du moteur après modification de la fréquence PWM**

Nous remarquons sur la Figure 183 que la réponse en vitesse est maintenant plus lisse et l'influence de la commande PWM est moins visible.

Nous constatons également que le temps de calcul du solveur a considérablement augmenté. Si on règle une fréquence PWM de l'ordre de 20 kHz le temps de simulation devient très long puisque le solveur doit s'imposer au moins 20000 pas de calcul par seconde. Afin de simplifier le travail du solveur on peut régler le **Simulation mode** sur **Averaged** afin de commander le moteur avec la valeur moyenne de la tension. Pour des fréquences élevées, les résultats seront identiques et le temps de calcul largement diminué.

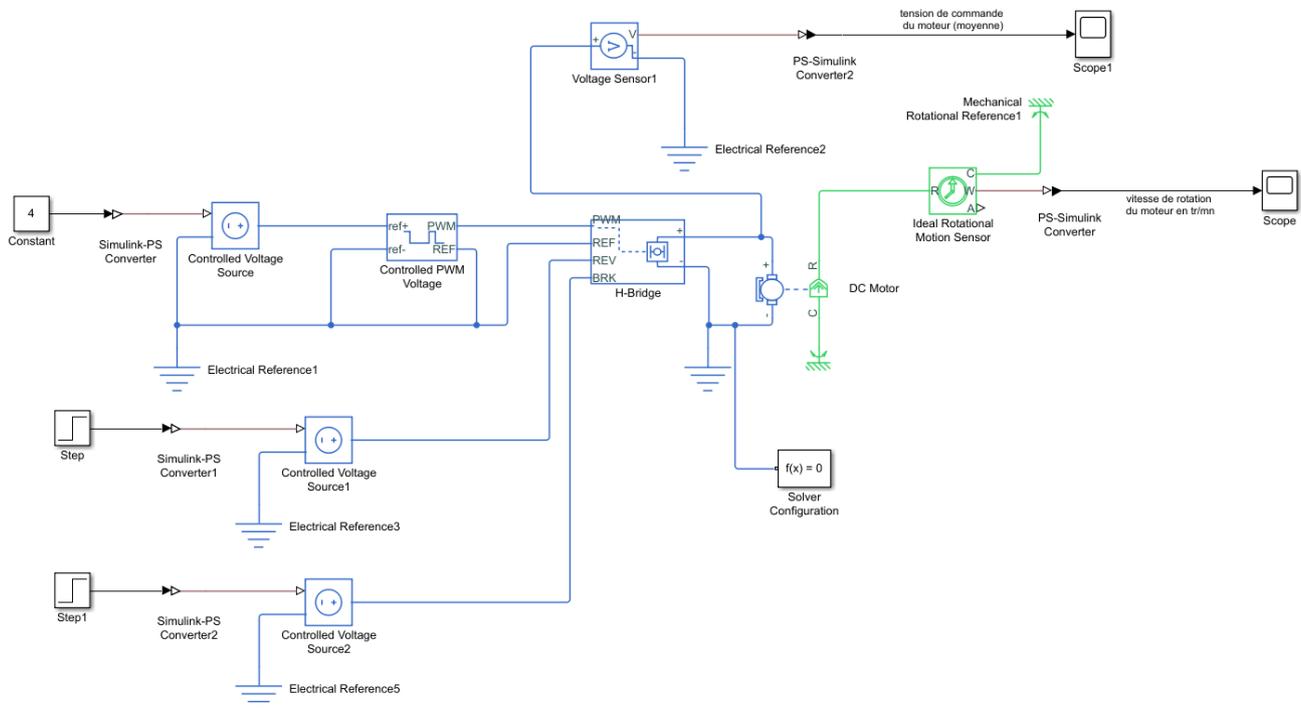
Régler le **Simulation mode** sur **Averaged** et relancer la simulation pour observer le résultat.

Le résultat sera strictement identique à celui de la Figure 183. Nous avons pu constater le gain de temps dans la résolution du modèle par le solveur.

### 3. Utilisation du composant « H-Bridge » (pont en H).

Afin de pouvoir contrôler le sens de rotation du moteur, il est nécessaire d'utiliser un pont en H. Le composant « **H-Bridge** » remplit cette fonction.

**Ouvrir** le fichier `pont_en_H.slx`.



**Figure 184 : commande PWM d'un moteur à courant continu avec pont en H**

Ce modèle permet de commander un moteur à courant continu à partir d'une commande PWM et d'un pont en H. Il est possible de visualiser la vitesse de rotation du moteur ainsi que la tension moyenne d'alimentation du moteur

**Lancer** la simulation et **observer** la vitesse de rotation du moteur (Figure 185).

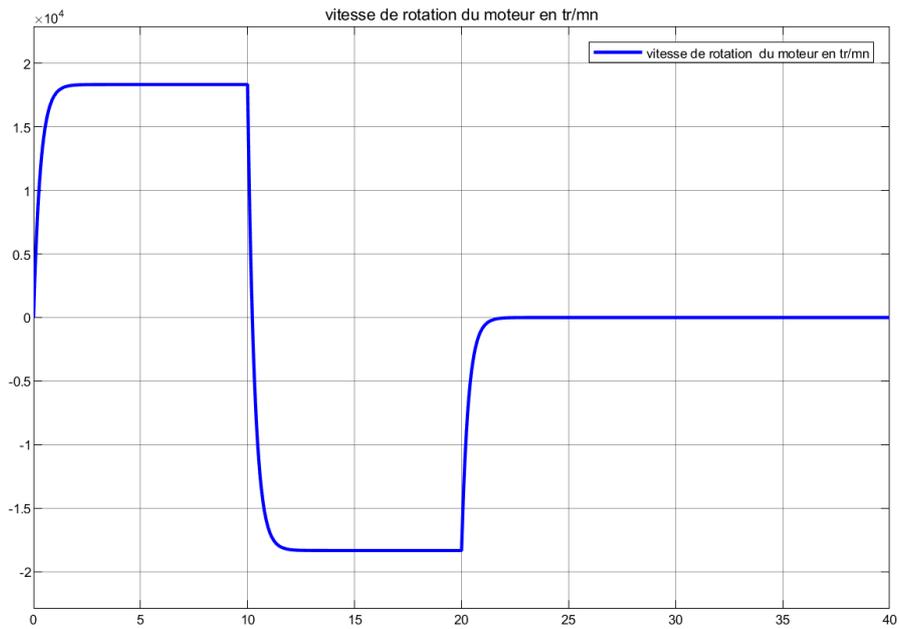


Figure 185 : vitesse de rotation du moteur commandé par le pont en H

Le moteur tourne dans le sens positif jusqu'à  $t=10$  s, puis dans le sens négatif jusqu'à  $t=20$  s, puis s'arrête à partir de  $t=20$  s.

**Ouvrir** le scope correspondant à la tension de commande du moteur (Figure 186).

Nous remarquons qu'à l'instant  $t=10$  s un échelon de tension de 5 V est imposé sur l'entrée **REV** du pont en H provoquant l'inversion du sens de rotation. A l'instant  $t=20$  s, un échelon de tension de 5 V est imposé sur l'entrée **BRK** du pont en H provoquant l'arrêt du moteur.

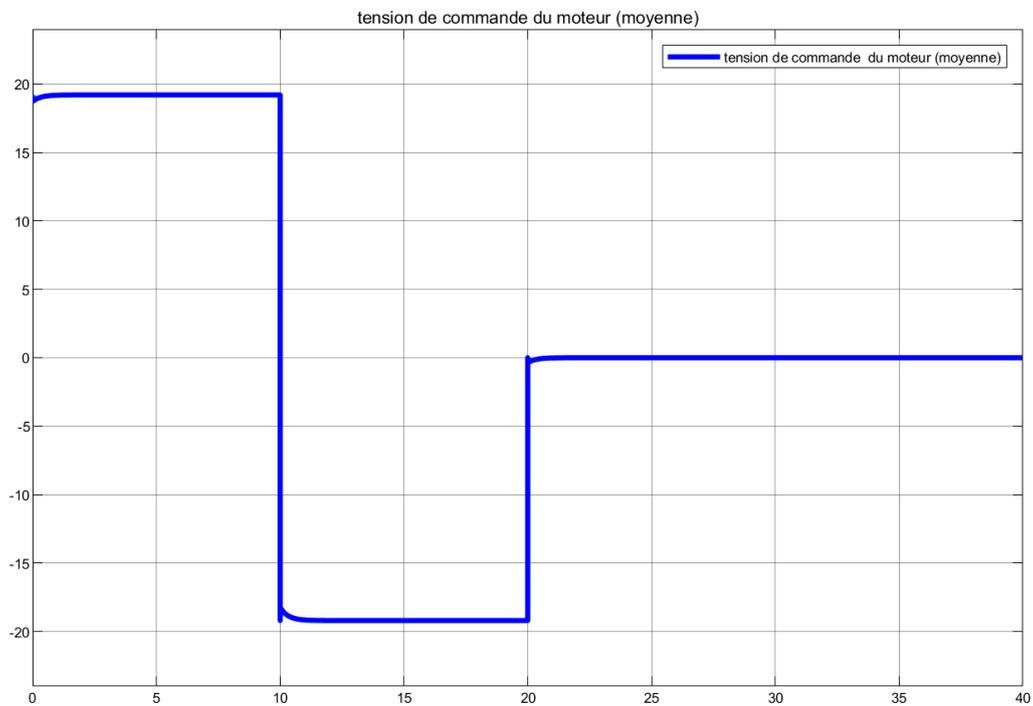
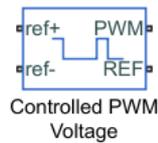


Figure 186 : tension moyenne d'alimentation du moteur

## Paramétrage

Controlled PWM Voltage



Simscape/Electronics/Actuators and Drivers/Drivers

## Paramétrage de l'onglet **PWM** :

La fréquence PWM est réglée sur 1000 hz et le **Simulation mode** sur **Averaged**.

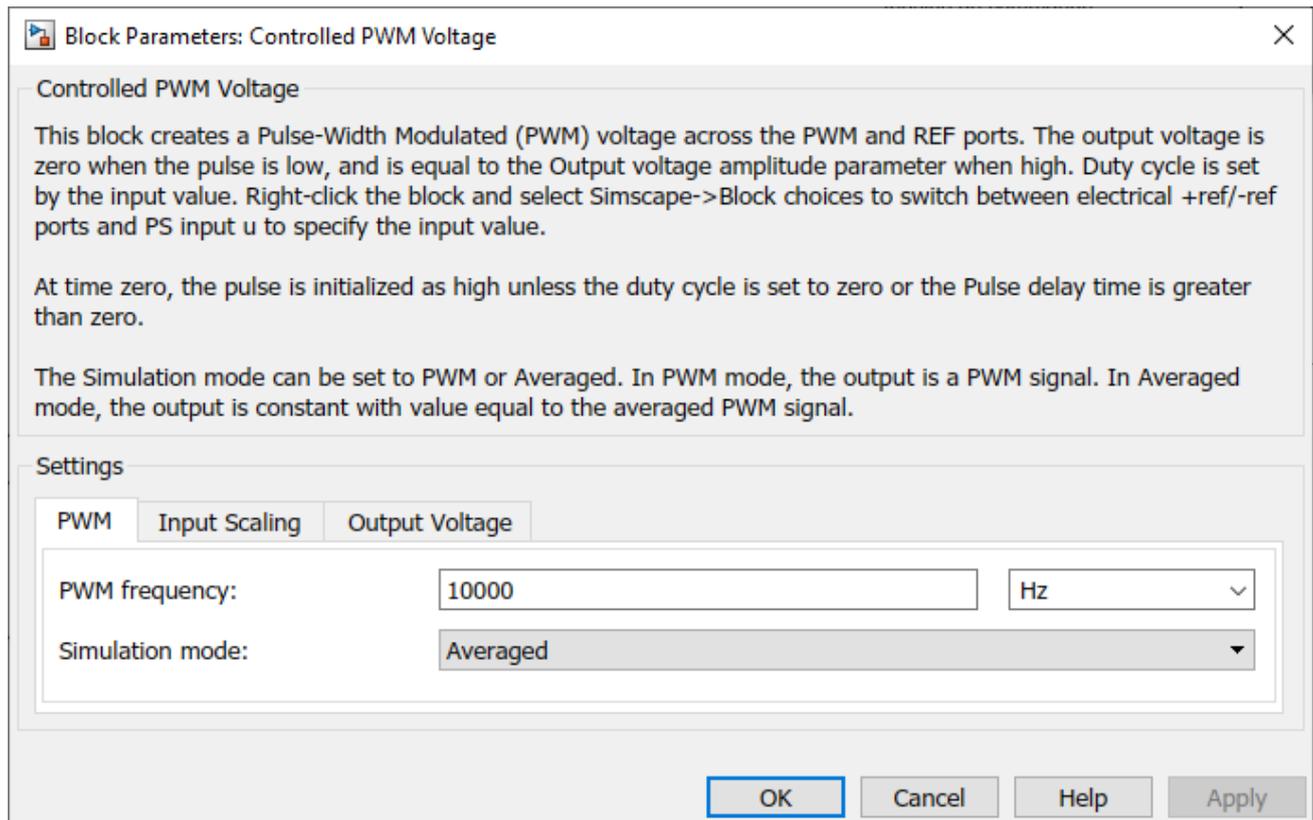


Figure 187 : paramétrage de l'onglet PWM du bloc Controlled PWM Voltage

## Paramétrage de l'onglet **Input Scaling** :

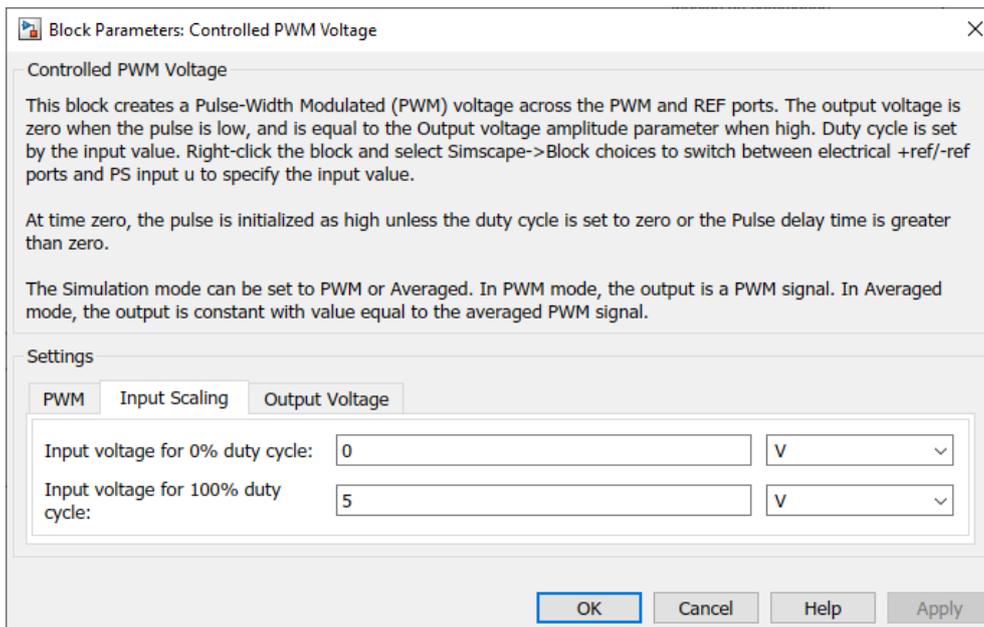


Figure 188 : paramétrage de l'onglet **Input Scaling** du bloc **Controlled PWM Voltage**

## Paramétrage de l'onglet **Output Voltage**

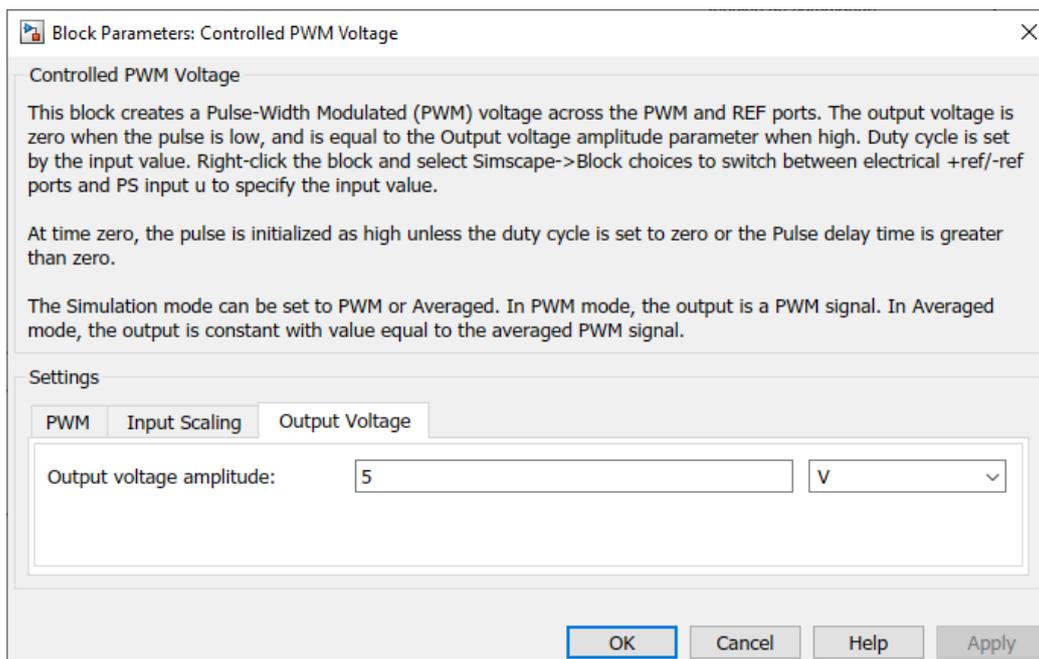
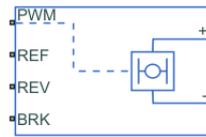


Figure 189 : paramétrage de l'onglet **Output Voltage** du bloc **Controlled PWM Voltage**

## Paramétrage

### H-Bridge



Simscape/Electronics/Actuators and Drivers/Drivers

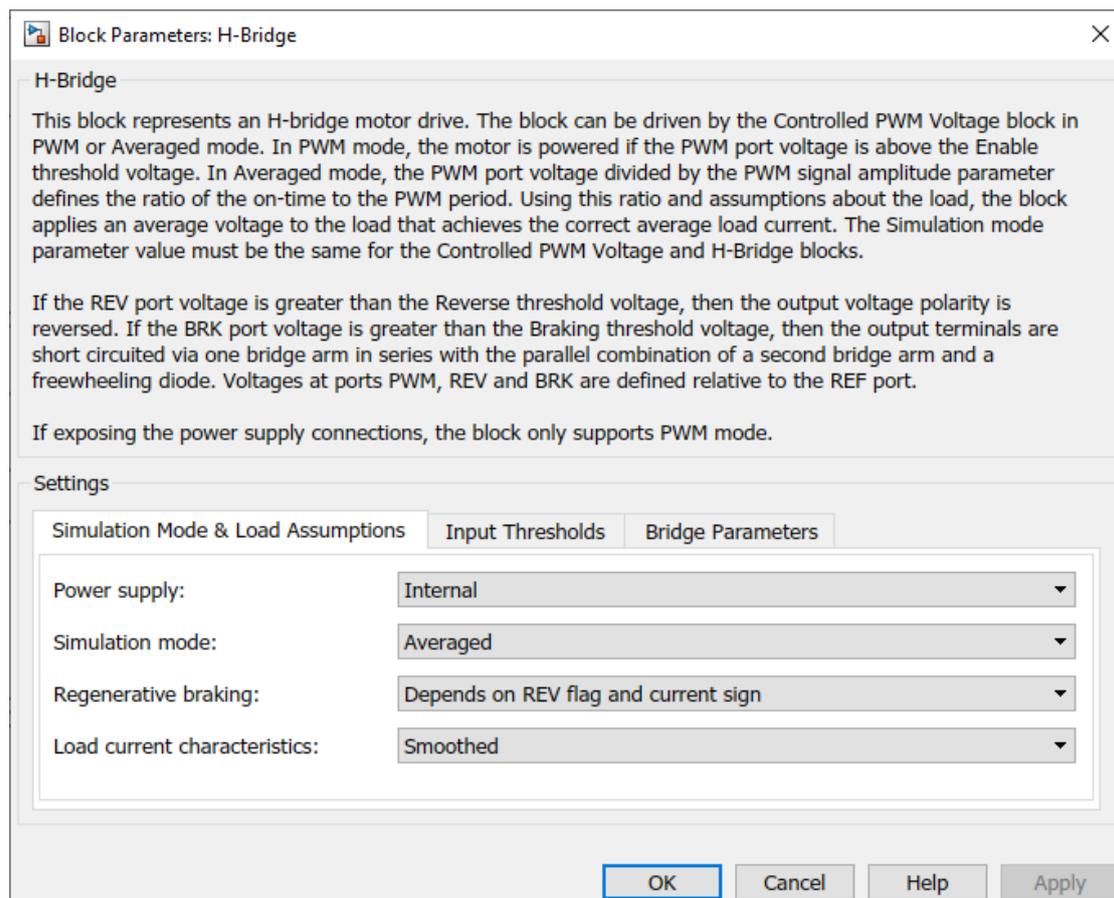
Ce composant modélise un pont en H. Il possède 6 ports de type **PCP** du domaine électrique.

- **PWM** : signal d'entrée PWM en provenance du bloc « Controlled PWM Voltage Source »
- **REF** : référence de la tension du signal PWM (souvent relié à la masse)
- **REV** : commande de l'inversion du sens de rotation
- **BRK** : commande de l'arrêt du moteur
- **+** et **-** : ces deux ports sont reliés aux bornes du moteur

Le pont en H reçoit, la tension PWM sur le port d'entrée **PWM** et alimente le moteur directement avec cette tension. Si la tension de l'entrée **REV** dépasse un seuil fixé dans les paramètres du composant, le sens de rotation s'inverse. Si la tension de l'entrée **BRK** dépasse un seuil fixé dans les paramètres du composant, l'alimentation du moteur est coupée.

### Onglet **Simulation Mode & Load assumptions**

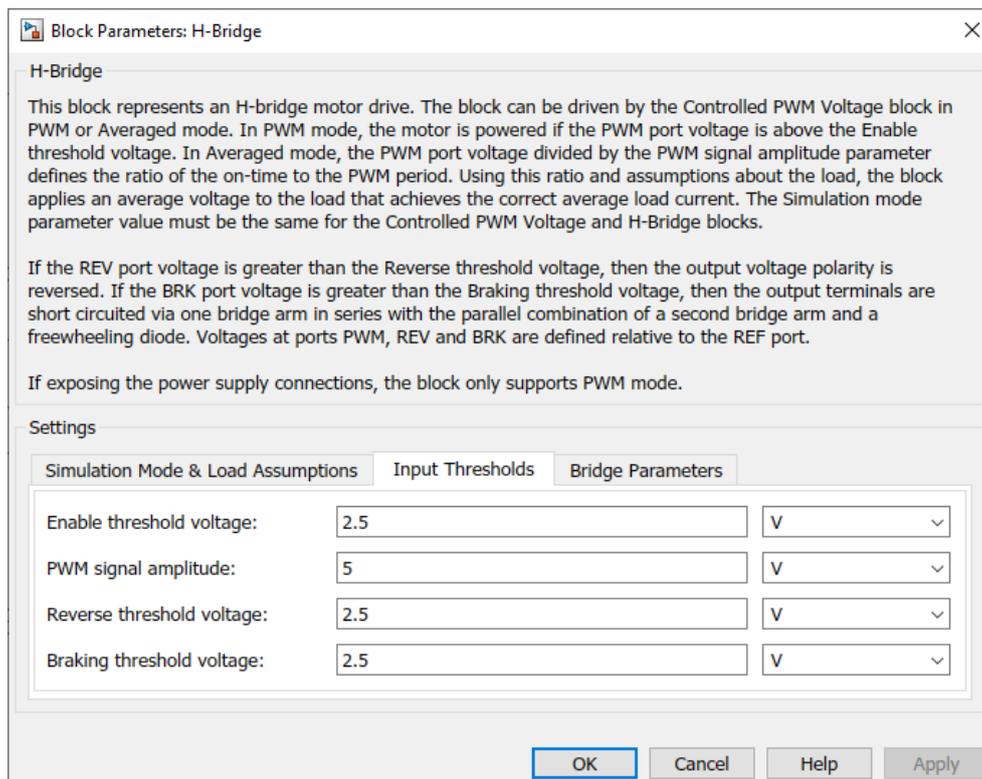
Le **Simulation Mode** est réglé sur **Average**.



**Power supply**: permet de spécifier si les ports d'alimentation du pont en H sont internes ou externes.

**Simulation mode** : choix entre **Average** et **PWM**

## Onglet **Input Thresholds**



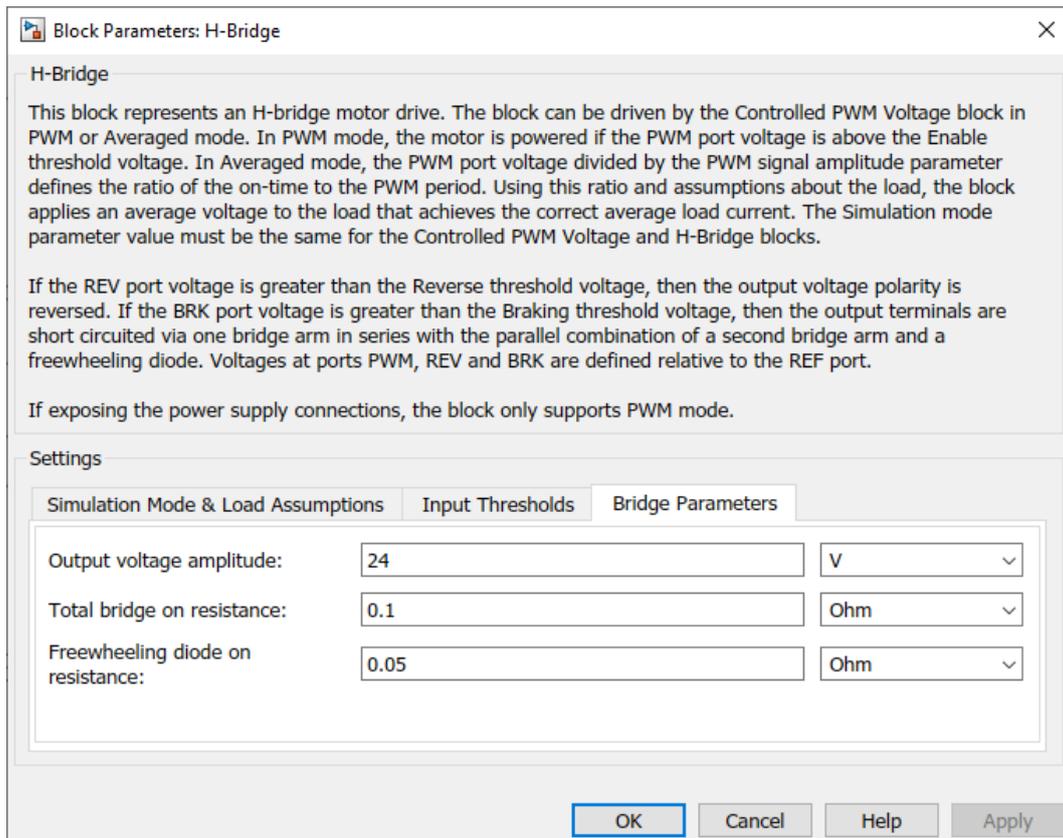
C'est ici que sont entrés les différents seuils qui caractérisent le fonctionnement du pont en H.

**Enable thresholds voltage** : niveau de tension que doit atteindre l'amplitude du signal PWM pour déclencher l'alimentation du moteur (uniquement valable si le **Simulation mode** est sur **PWM**)

**PWM signal amplitude** : amplitude du signal PWM qui doit correspondre avec l'amplitude du signal PWM généré par le bloc « Controlled PWM voltage source »

**Reverse thresholds voltage** : niveau de tension relevée sur le port **REV** qui provoque l'inversion du sens de rotation du moteur

**Braking thresholds voltage** : niveau de tension relevée sur le port **BRK** qui provoque l'arrêt du moteur.



**Figure 190 : paramétrage du bloc H-Bridge**

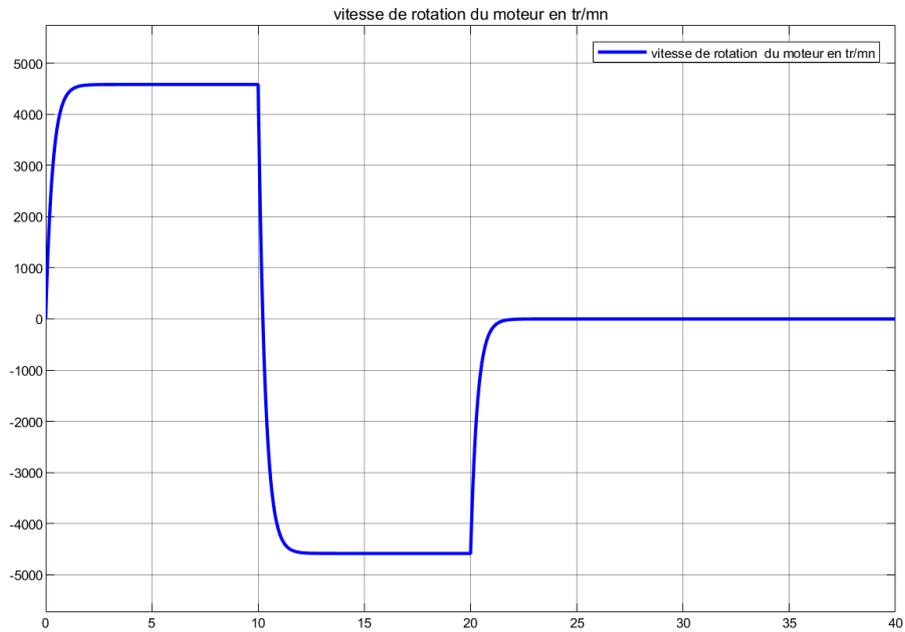
**Output voltage amplitude :** permet de fixer l'amplitude maximale de la tension d'alimentation du moteur.

**Total bridge on resistance :** résistance des deux transistors qui jouent le rôle d'interrupteur pour le pont en H.

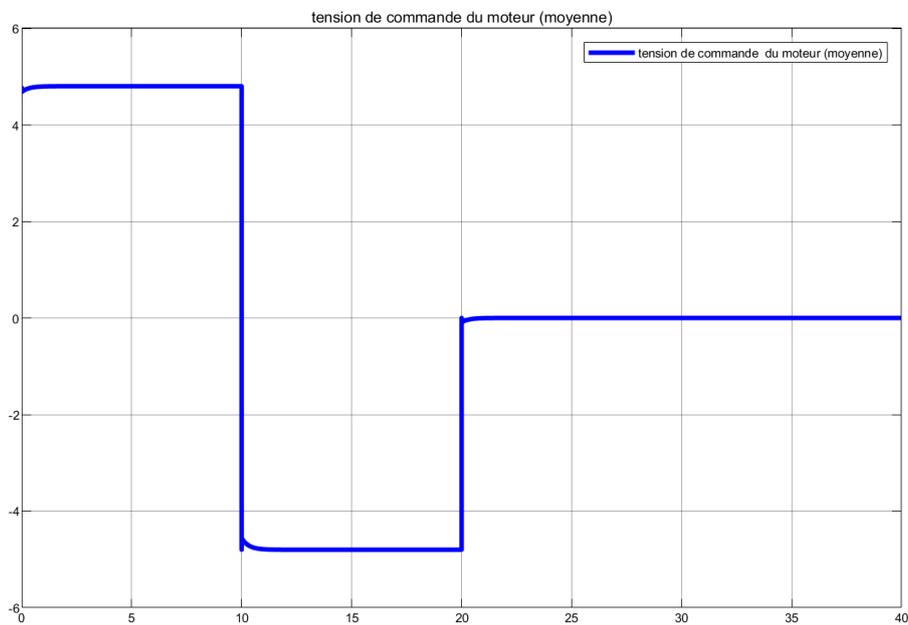
**Freewheeling diode resistance :** résistance des diodes de roue libre du pont qui permettent le passage transitoire du courant lors de la fermeture des interrupteurs (uniquement pour le **Simulation Mode PWM**)

**Modifier** le bloc Constant permettant d'alimenter le composant « Controlled PWM Voltage» et fixer sa valeur à **1** au lieu de **4** afin de modifier l'amplitude de la tension d'alimentation

**Lancer** la simulation et **visualiser** l'influence sur la tension d'alimentation et la vitesse de rotation du moteur (Figure 191 et Figure 192).



**Figure 191 : vitesse de rotation du moteur commandé par le pont en H**



**Figure 192 : tension moyenne d'alimentation du moteur**

## D. Rendre un modèle interactif, utilisation de la bibliothèque « dashboard »

Il est possible d'interagir avec un modèle durant la simulation. Pour cela, la bibliothèque de Simulink intitulée « **dashboard** » propose une série de boutons et d'interfaces de visualisation qui vont nous permettre de créer des modèles interactifs (Figure 193).

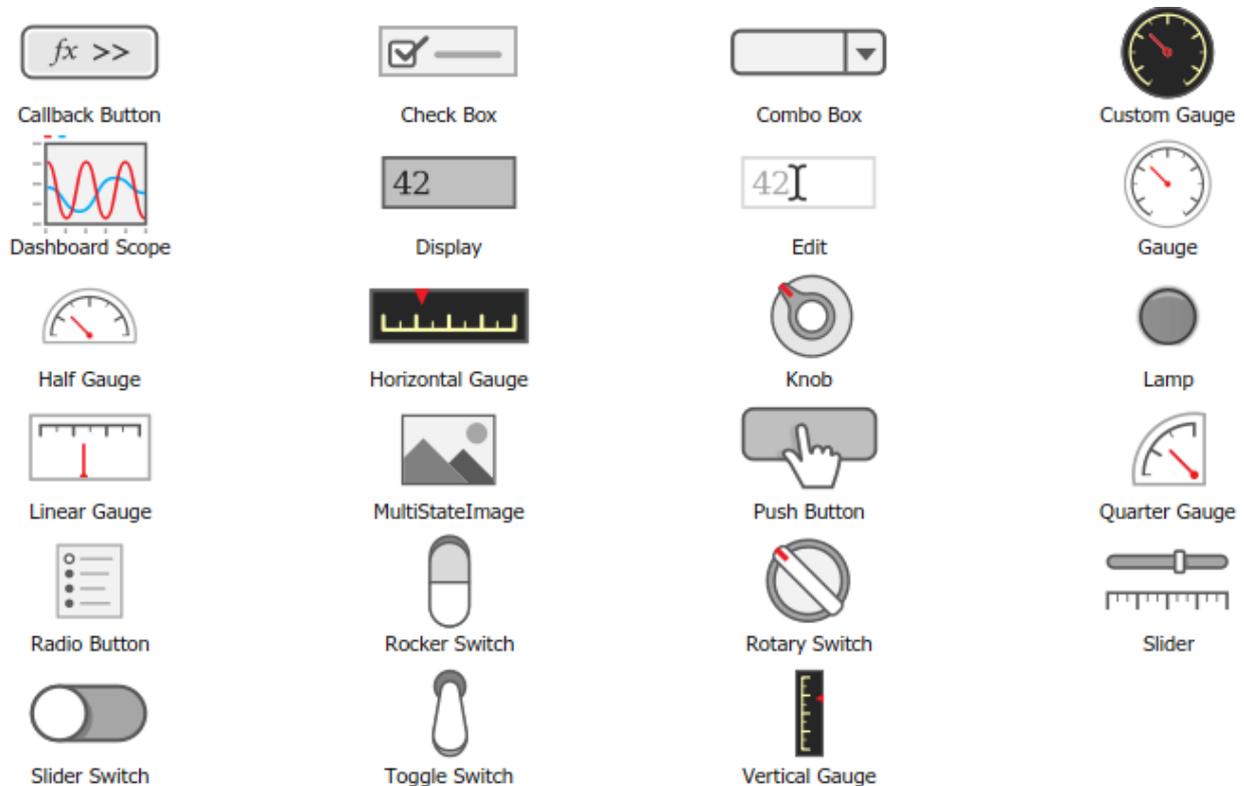


Figure 193 : les éléments de la bibliothèque de Simulink « Dashboard »

Il est ainsi possible d'agir sur des boutons pour modifier des paramètres du modèle et d'observer en temps réel l'influence de cette modification sur le comportement du modèle.

### 1. Exemple de modèle interactif

Ouvrir le fichier « **axe\_lineaire\_dashboard.slx** » (Figure 194).

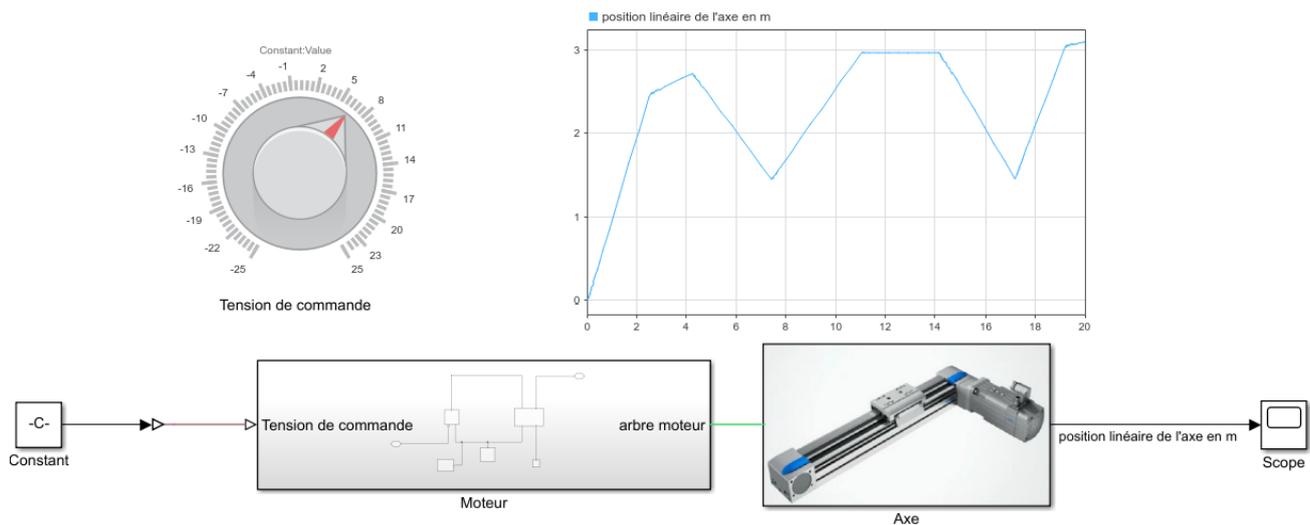


Figure 194 : pilotage interactif d'un modèle

Ce fichier contient le modèle de l'axe linéaire commandé en boucle ouverte. Deux composants de la bibliothèque Dashboard ont été ajoutés afin de rendre ce modèle interactif :

- Un **Knob**, bouton qui a été connecté avec le bloc constant qui permet de régler la tension de commande de l'axe.
- Un **Dashboard Scope** qui permet de visualiser de manière dynamique l'évolution d'un paramètre durant la simulation. Ce **Dashboard Scope** est relié au signal qui donne la position de l'axe linéaire en m.

Le **temps de simulation** est réglé sur **20 s** et la simulation a été paramétrée pour se dérouler en temps réel.

**Lancer** la simulation et agir sur le bouton « **Tension de commande** » par simple glisser déposer et observer la variation de la position linéaire de l'axe directement dans le scope interactif (Figure 195).

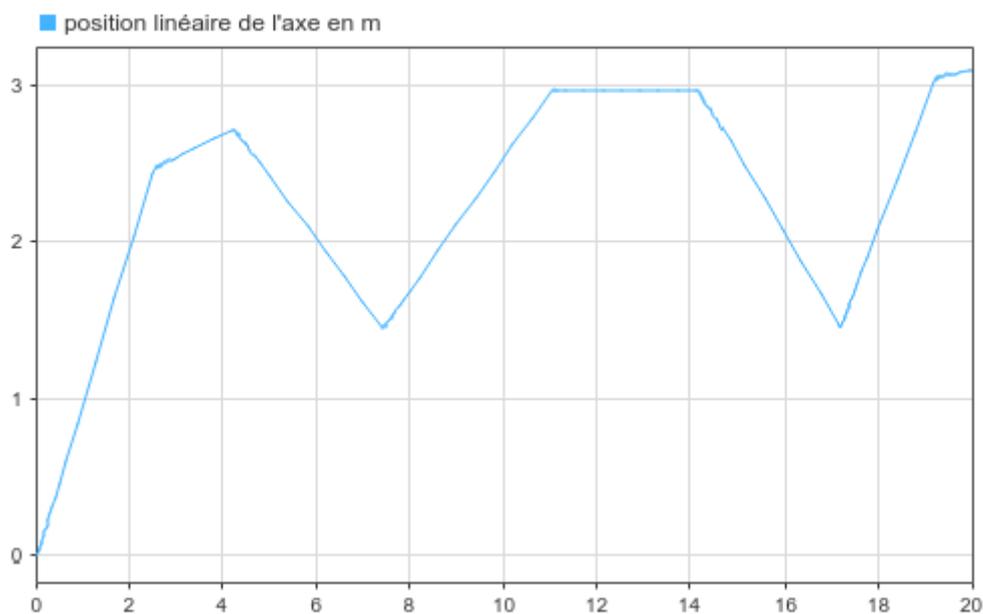


Figure 195 : visualisation en temps réel de la position de l'axe linéaire

## 2. Utilisation des blocs de la bibliothèque

Dans cet exemple nous allons paramétrer les blocs de la bibliothèque « Dashboard » du modèle utilisé précédemment.

**Ouvrir** le modèle « `axe_lineaire_dashboard_start.slx` ».

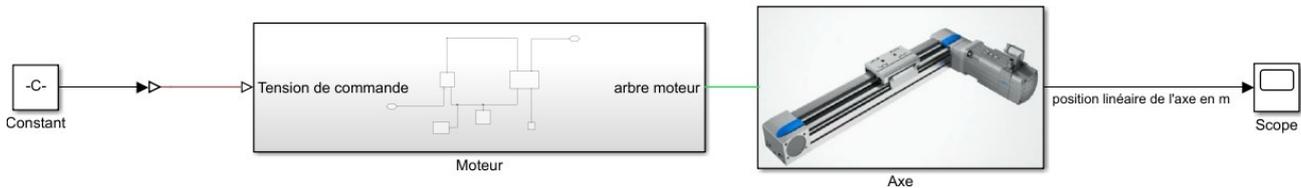


Figure 196 : modèle « `axe_lineaire_dashboard_start.slx` »

Ce fichier contient le modèle de l'axe linéaire précédemment réalisé. Nous allons ajouter des composants de la bibliothèque Dashboard afin de le rendre interactif.

Afin de rendre la visualisation plus réaliste, il est préférable que le calcul s'effectue en temps réel, c'est-à-dire que 1 s dans le déroulement de la simulation représentera 1 s dans la réalité.

Pour cela cliquer sur la petite flèche sous le bouton  afin d'ouvrir le menu déroulant, puis choisir **Simulation Pacing** (Figure 197).

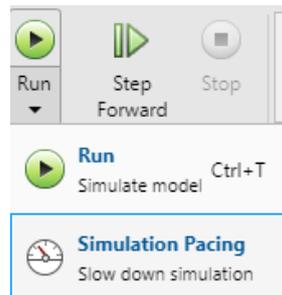


Figure 197 : sélection de la fonction Simulation Pacing

La fenêtre **Simulation Pacing Options** (Figure 198) permet d'indiquer le coefficient de déroulement temporel de la simulation.

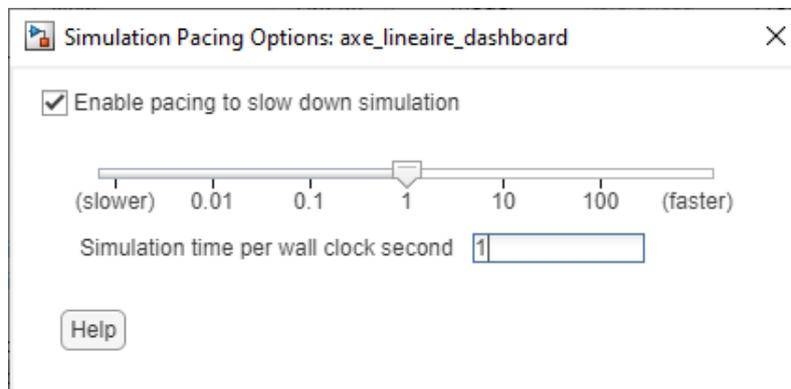
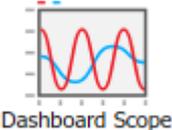


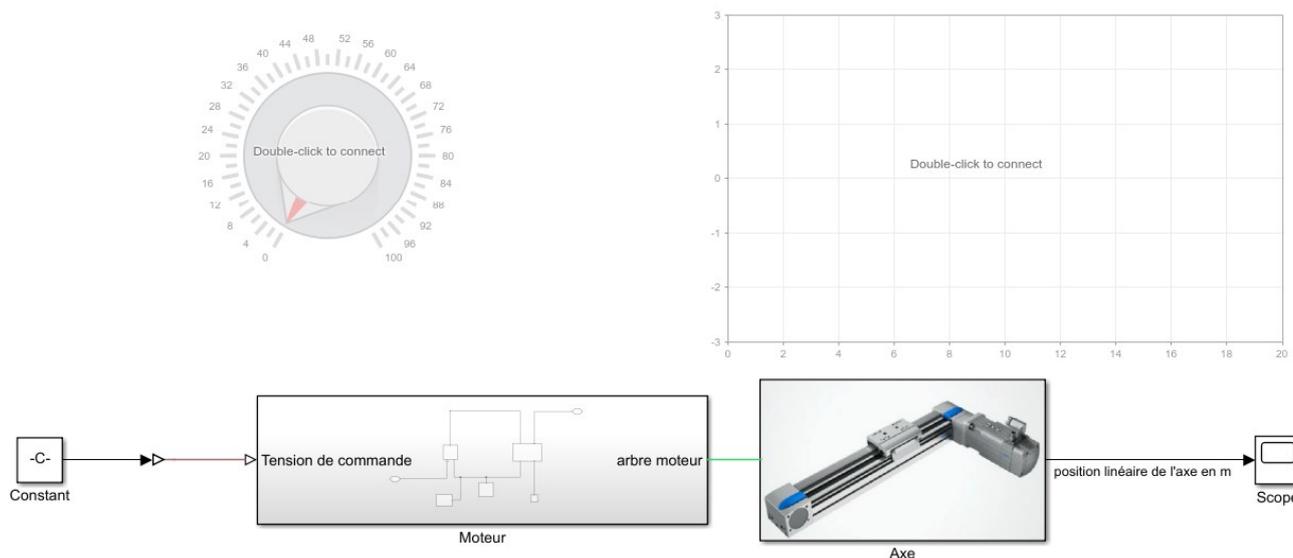
Figure 198 : paramétrage de la fenêtre du Simulation Pacing

Cocher la case **Enable pacing to slow down simulation**, puis choisir le coefficient de déroulement temporel de la simulation (**Simulation time per wall clock second**). Si ce coefficient est à 1, la simulation se déroulera en temps réel. Une seconde dans le temps de simulation durera exactement une seconde durant le processus de résolution du modèle. Cela permettra d'observer les phénomènes à la même vitesse que dans la réalité. Ce coefficient peut être modifié pour accélérer ou diminuer le temps de simulation du modèle.

Nous allons maintenant ajouter les composants de la bibliothèque Dashboard.

Fonction du composant	Représentation	Bibliothèque
Bouton interactif (Knob)	 Knob	Simulink/Dashboard
Scope interactif	 Dashboard Scope	Simulink/Dashboard

Faire **glisser, positionner** et **redimensionner** le bloc **knob** et le bloc **Dashboard Scope** pour obtenir la configuration de la Figure 199.



**Figure 199 : insertion des blocs de la bibliothèque Dashboard**

A ce stade les blocs de la bibliothèque Dashboard ne sont pas encore connectés au modèle et ne permettent pas d'interagir avec lui.

Bouton interactif (Knob)



Simulink/Dashboard

**Double cliquer** sur le bloc **Knob**.

**Cliquer** ensuite sur le bloc qui sera piloté par le bouton. Ici, il faudra sélectionner le bloc « **Constant** » qui impose la tension à l'entrée de l'axe linéaire puis valider la connexion en cliquant sur la case à cocher **connect** (Figure 200).

Le paramétrage du composant consiste également à préciser la plage de variation du paramètre.

**Minimum** : limite inférieure de variation du paramètre

**Maximum** : limite supérieure de variation du paramètre

**Ticks** : intervalle entre deux graduations

La fenêtre de paramétrage doit être identique à la Figure 200.

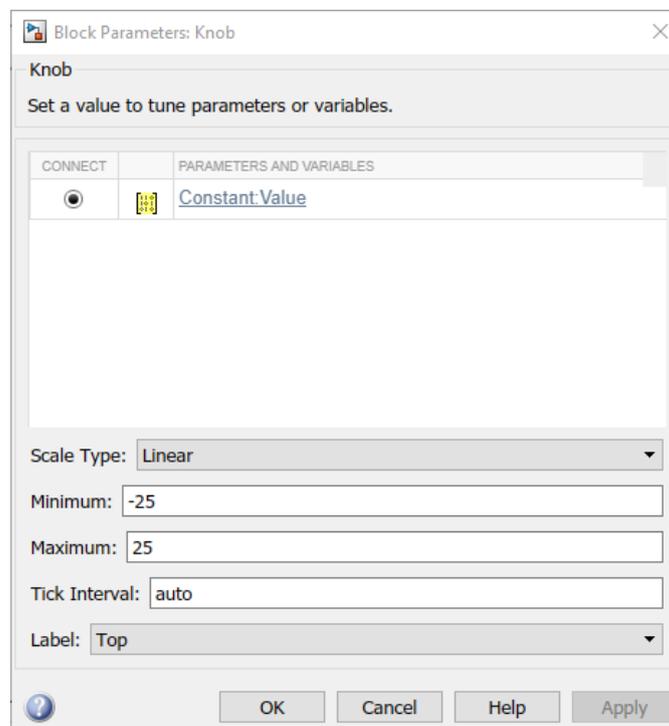


Figure 200 : fenêtre de paramétrage du bloc knob

Le bouton change alors d'aspect pour prendre la forme représentée sur la Figure 201.

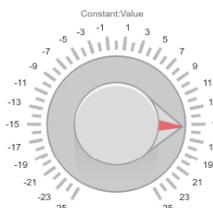


Figure 201 : allure du bouton knob après paramétrage et connexion avec le modèle

Scope interactif



Simulink/Dashboard

**Double cliquer** sur le bloc **Scope Interactif**.

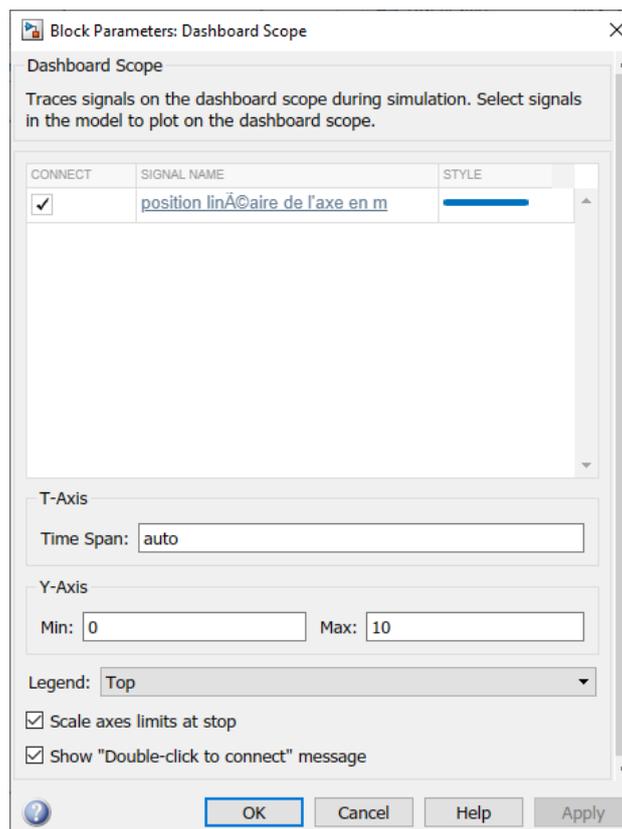
**Cliquer** ensuite sur le lien qui contient le signal que l'on souhaite afficher. Ici, il faudra sélectionner le signal de sortie du modèle (celui qui entre dans le scope et qui donne la position linéaire de l'axe) puis valider la connexion en cliquant sur la case à cocher **connect** (Figure 202).

Le paramétrage du composant consiste également à préciser la plage de temps d'affichage de la courbe et les limites supérieures et inférieures de l'axe Y.

**Time Span** : plage de temps spécifiée pour l'observation du phénomène. Si ce paramètre est sur **auto**, le Time Span sera égal au temps de simulation.

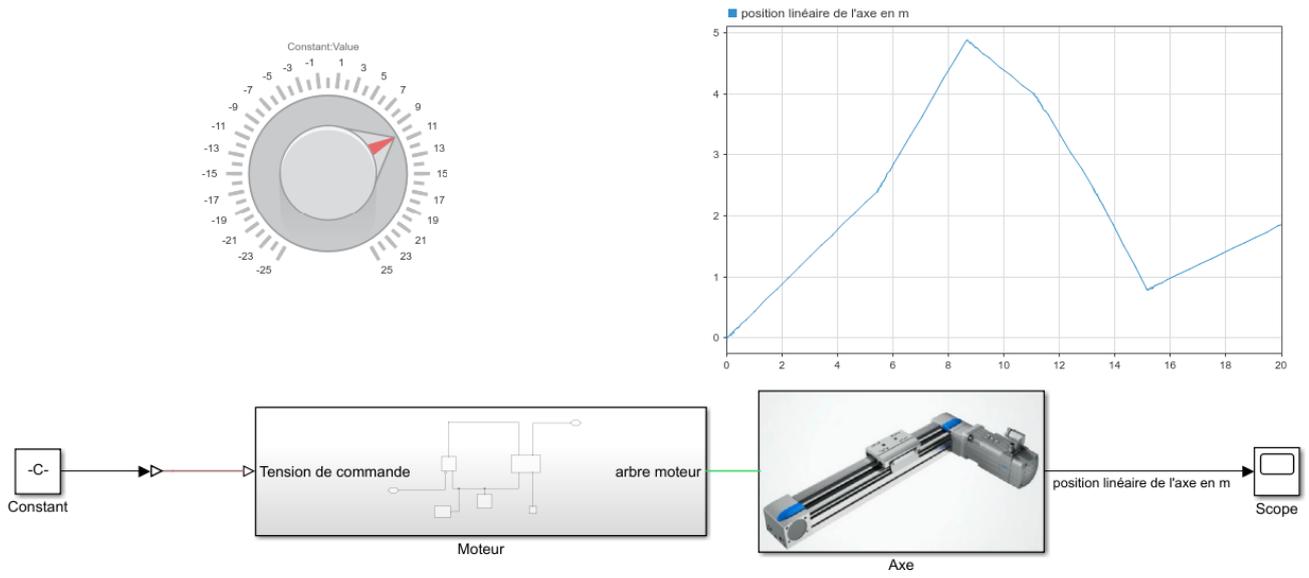
**Y-Axis Limits** : limites inférieure et supérieure de l'axe Y

La fenêtre de paramétrage doit être identique à la Figure 202.



**Figure 202 : fenêtre de paramétrage du bloc Dashboard Scope**

**Lancer** la simulation et observer en temps réel l'évolution de la position de l'axe.



**Figure 203 : simulation d'un modèle interactif avec les composants de la bibliothèque dashboard**

## E. Le domaine thermique

### 1. Rappels théoriques sur les échanges thermiques

Le domaine thermique de Simscape permet de modéliser les trois principaux phénomènes qui donnent lieu à des échanges thermiques :

- Le phénomène de conduction
- Le phénomène de convection
- Le phénomène de rayonnement

Comme pour chaque domaine physique de Simscape une connaissance théorique minimale du domaine physique est requise afin de pouvoir modéliser des systèmes.

### 2. Le phénomène de conduction

La conduction (Figure 204) est une transmission de chaleur provoquée par la différence de température entre deux régions d'un même matériau qui sont en contact physique. Un flux de chaleur s'établit naturellement de la partie chaude vers la partie froide du solide. Le flux thermique qui s'établit dépend des caractéristiques et de la géométrie du matériau. Plus le matériau est isolant et moins il y a de conduction.

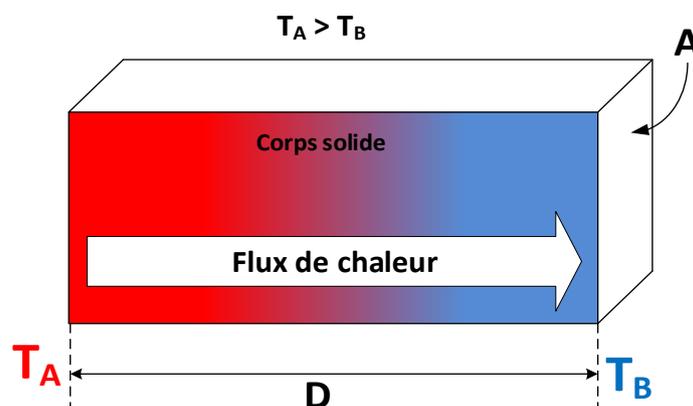


Figure 204 : transfert de chaleur par conduction

Ce type de transfert thermique peut être modélisé par la loi de Fourier définie de la manière suivante :

$$Q = \lambda \frac{A}{D} (T_A - T_B)$$

$Q$  : flux thermique en W ou J.s

$\lambda$  : conductivité thermique du matériau en W/(m.K)

$A$  : aire de la surface d'échange, normale à la direction du flux thermique en  $m^2$

$D$  : distance entre les parois chaude et froide du matériau en m

$T_A$  et  $T_B$  : températures des parois

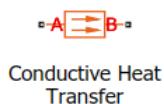
Dans la bibliothèque thermique de Simscape ce phénomène est modélisé à l'aide du bloc **Conductive Heat Transfer** (Figure 205).



Figure 205 : bloc Conductive Heat Transfer de Simscape

### Paramétrage du bloc Conductive Heat Transfer

Conductive Heat  
Transfert



Simscape/Foundation  
Library/Thermal/Thermal Elements

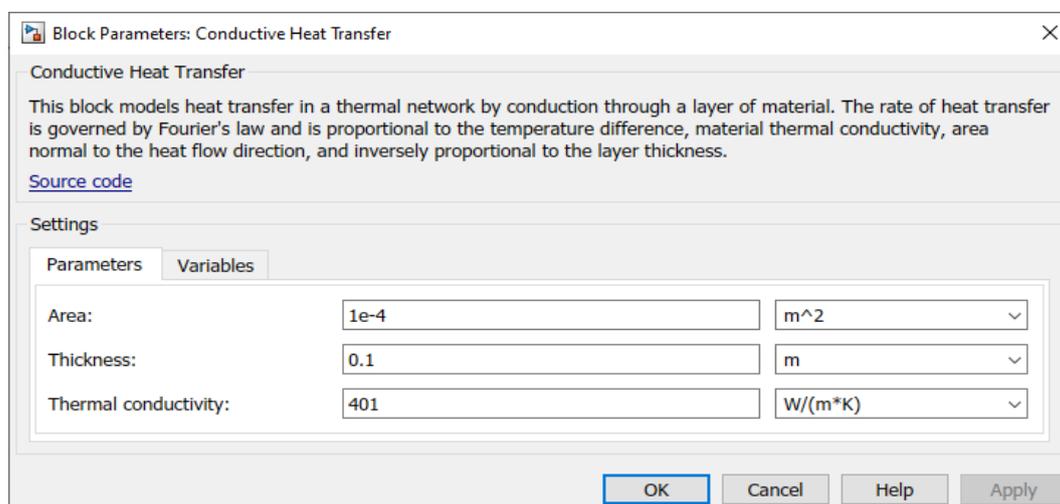


Figure 206 : Paramétrage du bloc Conductive Heat Transfer

**Area** : aire de la surface d'échange

**Thickness** : distance entre les parois chaude et froide du matériau

**Thermal Conductivity** : conductivité thermique du matériau au travers duquel s'effectue l'échange de chaleur par conduction.

### 3. Le phénomène de convection

La convection (Figure 207) est une transmission de chaleur au sein d'un fluide en mouvement ou entre un fluide en mouvement et une paroi solide. Plus le fluide est mobile et plus il y a de convection.

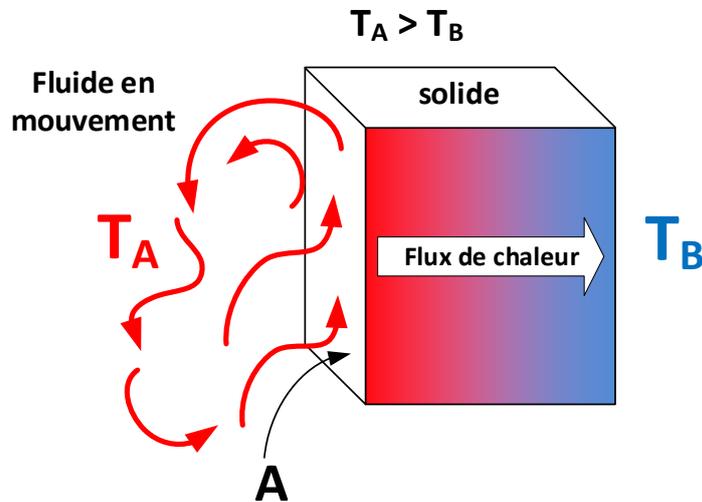


Figure 207 : transfert de chaleur par convection

Ce type de transfert thermique peut être modélisé par la loi de Newton définie de la manière suivante :

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

$Q$  : flux thermique en W ou J.s

$k$  : coefficient de transfert de chaleur par convection  $W/(m^2 \cdot K)$

$A$  : aire de la surface d'échange, normale à la direction du flux thermique  $m^2$

$T_A$  et  $T_B$  : températures des deux corps

Dans la bibliothèque thermique de Simscape ce phénomène est modélisé à l'aide du bloc **Convective Heat Transfert** (Figure 208).



Convective Heat  
Transfer

Figure 208 : bloc Convective Heat Transfer de Simscape

## Paramétrage du bloc Convective Heat Transfer

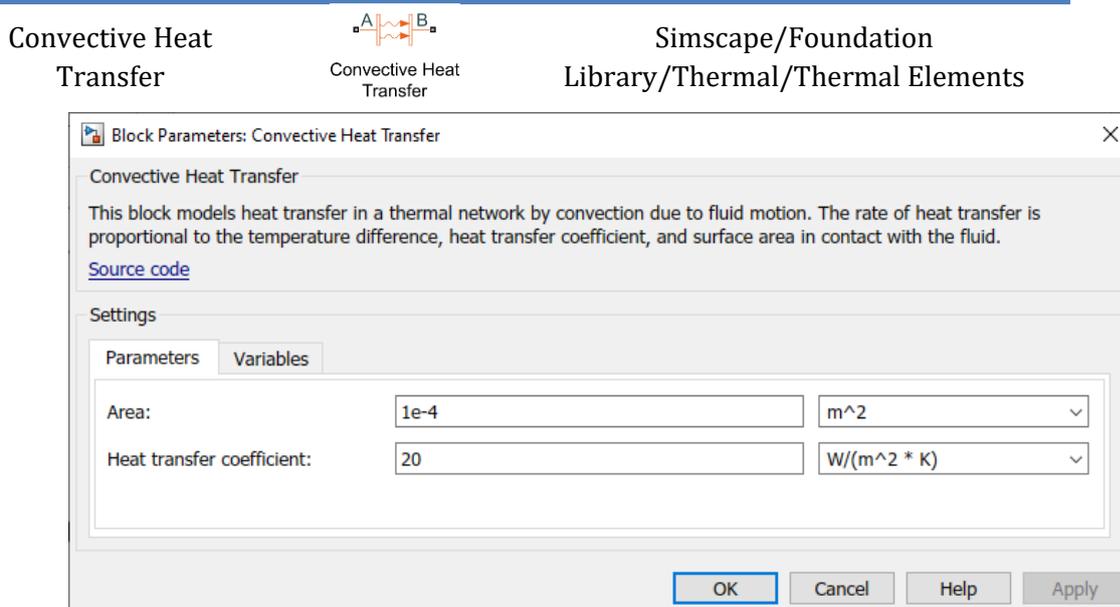


Figure 209 : Paramétrage du bloc Convective Heat Transfer

**Area** : aire de la surface d'échange

**Heat Transfer coefficient** : coefficient de transfert thermique par convection

### 4. Le phénomène de rayonnement

Le rayonnement thermique est un transfert de chaleur d'un corps à un autre par onde électromagnétique sans contact direct. Ce type de transfert peut se faire dans le vide.

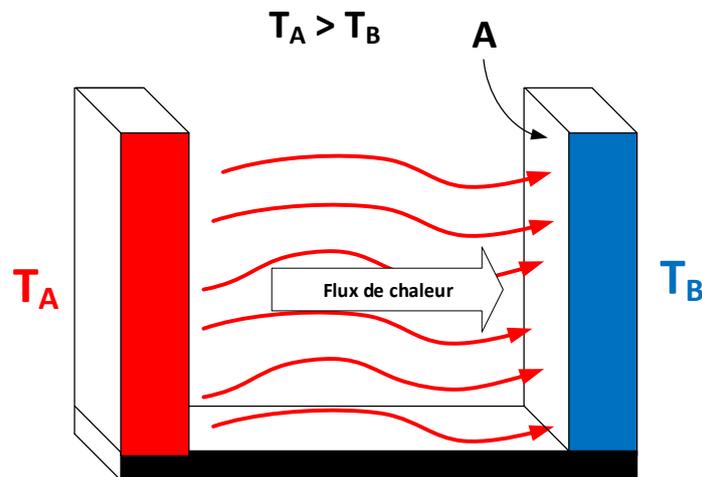


Figure 210 : transfert de chaleur par rayonnement

Ce type de transfert thermique peut être modélisé par la loi de Stefan-Boltzman définie de la manière suivante :

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

$Q$  : flux thermique en W ou J.s

$k$  : coefficient de radiation  $W/(m^2 \cdot K^4)$

$A$  : aire de la surface d'échange, normale à la direction du flux thermique en  $m^2$

$T_A$  et  $T_B$  : températures des deux corps

Dans la bibliothèque thermique de Simscape ce phénomène est modélisé à l'aide du bloc **Radiative Heat Transfert** (Figure 211).

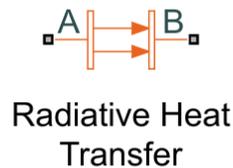


Figure 211 : bloc Radiative Heat Transfer de Simscape

### Paramétrage du bloc Radiative Heat Transfer

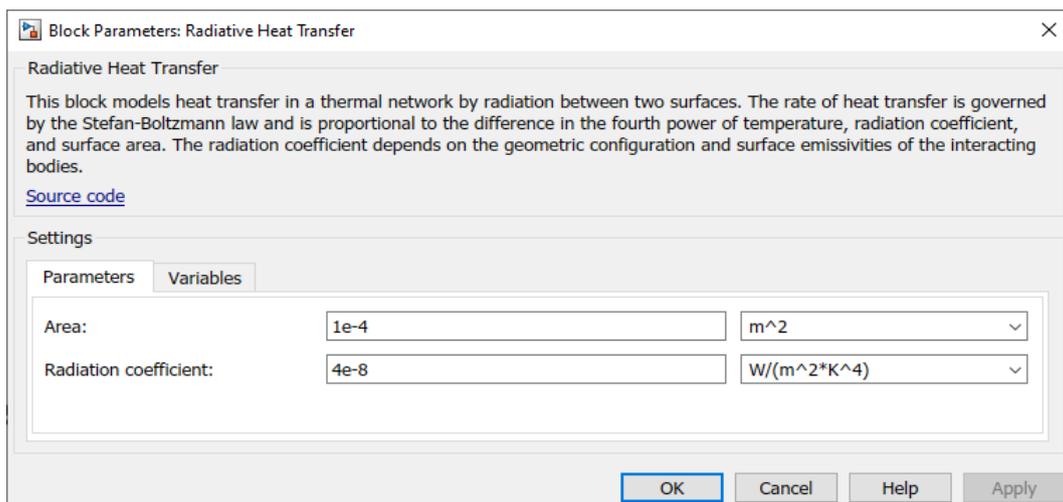
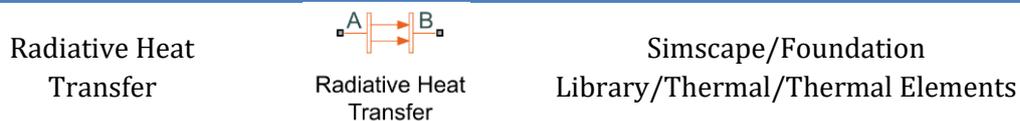


Figure 212 : Paramétrage du bloc Radiative Heat Transfer

**Area** : aire de la surface d'échange

**Radiation coefficient** : coefficient de transfert thermique par rayonnement

## 5. La masse thermique

La masse thermique permet de modéliser l'inertie thermique d'un élément et sa capacité à stocker et restituer de la chaleur. Plus le matériau aura une masse thermique importante et plus il sera long à monter en température. De plus cette montée en température dépendra de la nature du matériau. Un matériau métallique montera plus facilement en température que du béton ou que du bois. La masse thermique d'un matériau dépend donc de sa masse et de sa chaleur spécifique.

La masse thermique est caractérisée par l'équation suivante

$$Q = c \cdot m \frac{dT}{dt}$$

$Q$  : flux thermique en W ou J.s

$c$  : chaleur spécifique du matériau J/kg/K

$m$  : masse du matériau en kg

$\frac{dT}{dt}$  : variation de température de la masse en K/s

Dans la bibliothèque thermique de Simscape une masse thermique se modélise à l'aide du bloc **Thermal Mass** (Figure 213).



Figure 213 : bloc Thermal Mass de Simscape

### Paramétrage du bloc Thermal Mass

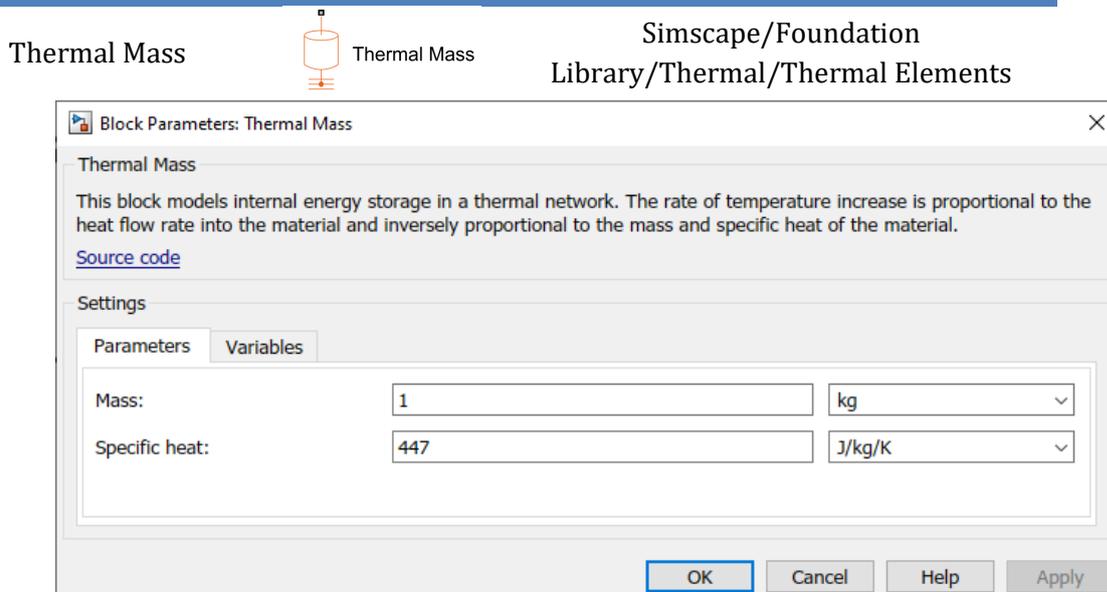


Figure 214 : Paramétrage du bloc Thermal Mass

**Mass** : masse du matériau

**Specific Heat** : chaleur spécifique

## 6. La résistance thermique

La résistance thermique est un concept permettant d'appréhender de manière globale le comportement thermique d'un élément. Ce comportement peut représenter des échanges thermiques par conduction, convection ou rayonnement. Cette représentation est très utilisée puisqu'elle permet de modéliser de manière globale la résistance que pourra présenter une paroi à un flux thermique qui s'établit entre deux isothermes. Plus la résistance thermique est élevée et plus le matériau sera isolant. En fonction du point de vue de modélisation choisi et des données à disposition pour la modélisation, les échanges thermiques pourront donc être modélisés soit à l'aide de résistances thermiques soit à l'aide des blocs, conduction, convection et rayonnement.

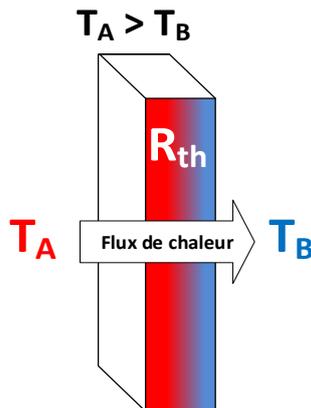


Figure 215 : résistance thermique  $R_{th}$

Une résistance thermique est caractérisée par l'équation suivante qui s'apparente à la loi d'Ohms du domaine électrique :

$$(T_A - T_B) = R_{TH} Q$$

$Q$  : flux thermique en W ou J.s

$R_{TH}$  : résistance thermique en K/W

$T_A - T_B$  : différence de température entre les parois

Il est possible de définir la résistance thermique de conduction ou de convection par identification en utilisant les lois de Fourier ou de Newton et la relation  $Q = \frac{(T_A - T_B)}{R_{TH}}$

### Résistance thermique de conduction

$$Q = \lambda \frac{A}{D} (T_A - T_B) = \frac{1}{R_{cond}} (T_A - T_B) \Rightarrow R_{cond} = \frac{D}{\lambda \cdot A}$$

$D$  : distance entre les parois chaude et froide du matériau en m

$\lambda$  : conductivité thermique du matériau en W/(m.K)

$A$  : aire de la surface d'échange, normale à la direction du flux thermique  $m^2$

$R_{cond}$  : résistance thermique de conduction en K/W

On notera qu'une augmentation de la surface d'échange  $A$  diminue la résistance thermique de conduction. Une augmentation de l'épaisseur du matériau augmente la résistance thermique de conduction.

### Résistance thermique de convection

$$Q = kA(T_A - T_B) = \frac{1}{R_{conv}}(T_A - T_B) \Rightarrow \boxed{R_{conv} = \frac{1}{k.A}}$$

$k$  : coefficient de transfert de chaleur par convection  $W/(m^2 \cdot K)$

$A$  : aire de la surface d'échange, normale à la direction du flux thermique  $m^2$

$R_{conv}$  : résistance thermique de convection en  $K/W$

On notera qu'une augmentation de la surface d'échange  $A$  diminue la résistance thermique de convection. L'épaisseur du matériau n'intervient pas dans la résistance thermique de convection puisque les échanges de chaleur se produisent à la surface du matériau.

### Résistance thermique surfacique

Dans la documentation qui donne les caractéristiques des matériaux, le terme de résistance thermique est souvent évoqué. La résistance thermique permet de comparer la performance énergétique des matériaux. Plus la résistance thermique est élevée et plus le matériau est isolant.

Il faut être très vigilant dans la mesure ou la résistance thermique est souvent donnée en  $m^2 \cdot K/W$ . Il s'agit en fait de la **résistance thermique surfacique** qui représente la résistance thermique d'un mètre carré de matériau. La résistance thermique surfacique est donc une grandeur usuelle qui diffère de la résistance thermique définie théoriquement.

Il existe une relation très simple entre ces deux grandeurs :

$$R_{thermique} (K / W) = \frac{R_{thermique\ surfacique} (m^2 \cdot K / W)}{A (m^2)}$$

La résistance thermique surfacique de conduction est donnée par :

$$\boxed{R_{thermique\ surfacique\ conduction} = \frac{D}{\lambda} \quad (m^2 \cdot K / W)}$$

La résistance thermique surfacique de convection est donnée par :

$$\boxed{R_{thermique\ surfacique\ convection} = \frac{1}{k} \quad (m^2 \cdot K / W)}$$

Dans la bibliothèque thermique de Simscape une résistance thermique se modélise à l'aide du bloc **Thermal Resistance** (Figure 216).



## Thermal Resistance

Figure 216 : bloc Thermal Resistance de Simscape

## Paramétrage du bloc Thermal Resistance

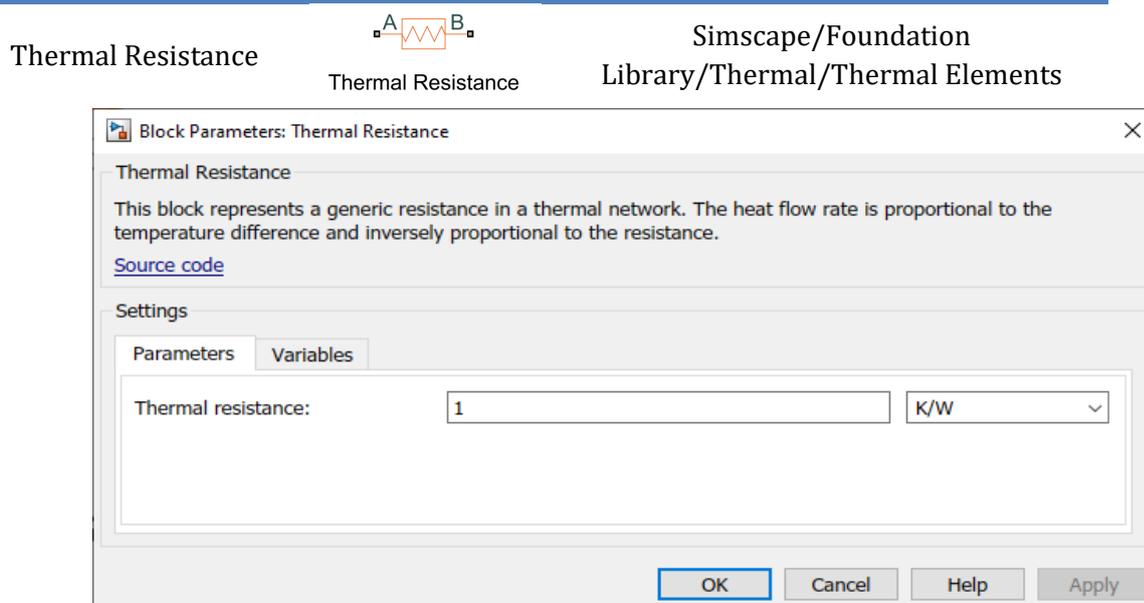


Figure 217 : Paramétrage du bloc Thermal Resistance

**Thermal Resistance** : résistance thermique de l'élément en K/W

### 7. Présentation de la bibliothèque Thermal de Simscape

La bibliothèque Thermal de Simscape comprend des éléments, des capteurs (Sensors) et des sources.

- Les éléments (Figure 218) permettent de modéliser les échanges thermiques comme décrit dans le paragraphe précédent
- Les sources (Figure 219) permettent de modéliser soit un de flux thermique constant « **Heat Flow Rate Source** » soit une zone du modèle qui restera à température constante « **Temperature Source** ». Ces sources peuvent également être pilotées par des grandeurs variables « **Controlled Heat Flow Rate Source** » et « **Controlled Temperature Source** »
- Les capteurs (Figure 220) permettent de prélever les grandeurs physiques sur le modèle. Il sera possible de prélever la température en un point de notre modèle « **Temperature Sensor** » (Across variable) ou le flux thermique à travers une connexion « **Heat Flow Rate Sensor** » (Through variable)

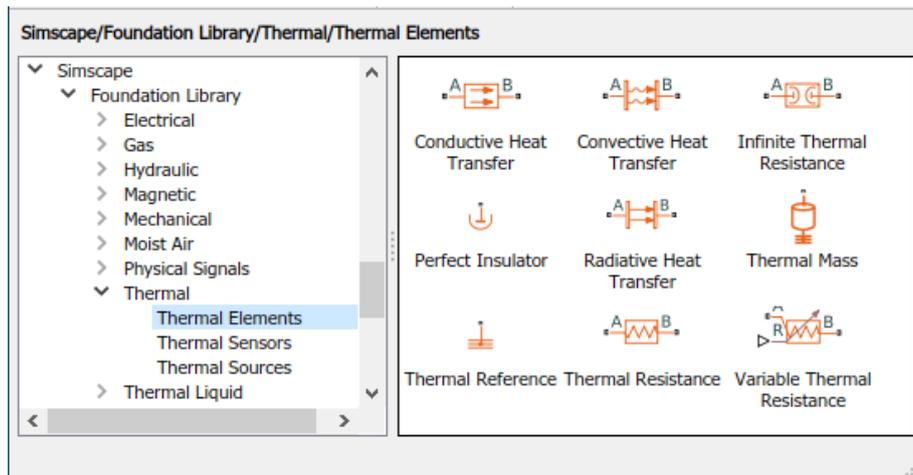


Figure 218 : les éléments de la bibliothèque Simscape Thermal

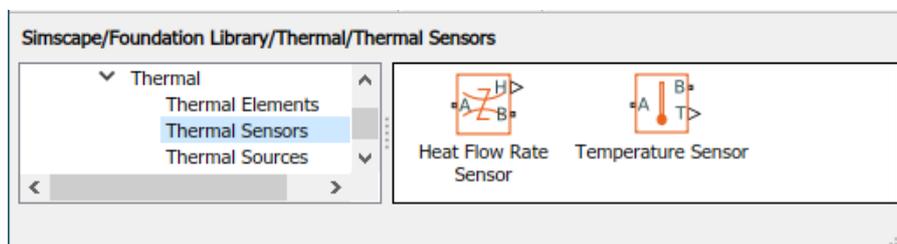


Figure 219 : les capteurs de la bibliothèque Simscape Thermal

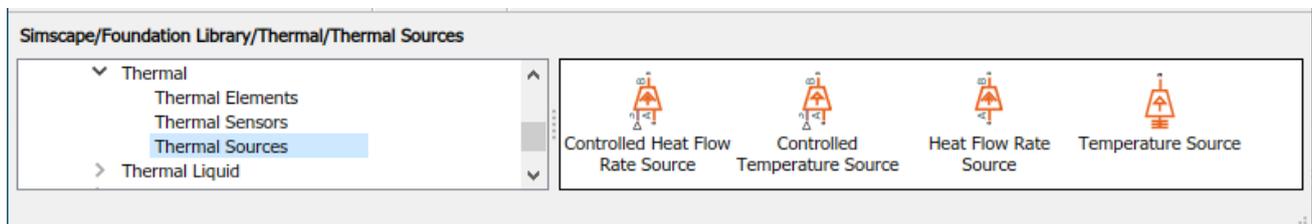


Figure 220 : les sources de la bibliothèque Simscape Thermal

## 8. Chauffage d'une barre métallique par conduction

La Figure 221 représente une barre d'acier à une température initiale de  $T_B=20^\circ\text{C}$  placée dans un milieu parfaitement isolant. Ses deux extrémités sont reliées à un solide chaud à une température  $T_H=300^\circ\text{C}$  et un solide froid à une température de  $T_C=0^\circ\text{C}$ .

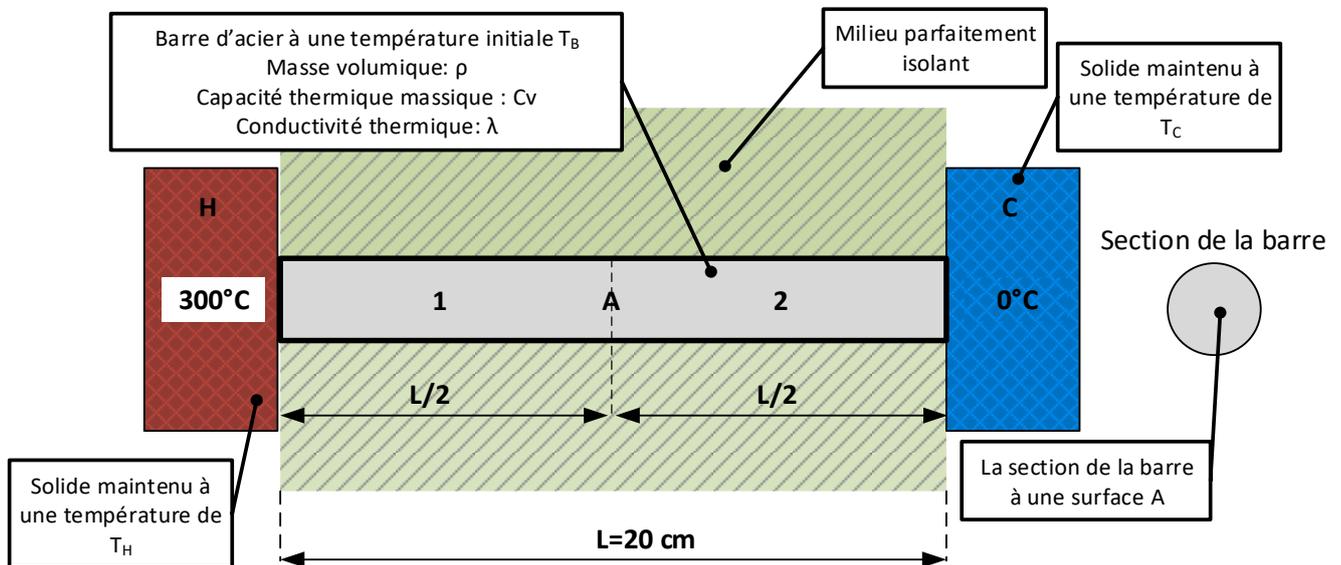


Figure 221 : chauffage d'une barre par conduction

Données pour la modélisation :

Longueur de la barre:  $L = 20\text{ cm}$

Diamètre de la barre :  $D = 3.2\text{ cm}$

Section de la barre :  $A = 8\text{ cm}^2$

Masse volumique de l'acier:  $\rho = 7800\text{ kg/m}^3$

Capacité thermique massique de l'acier :  $C_v = 444\text{ J.kg}^{-1}.\text{K}^{-1}$

Conductivité thermique de l'acier:  $\lambda = 50\text{ W.m}^{-1}.\text{K}^{-1}$

L'objectif de la modélisation est de déterminer :

- le temps nécessaire pour atteindre le régime permanent
- la température au milieu de la barre (point A, distance  $L/2$  des sources de température)
- le flux thermique qui traverse la barre en régime permanent

## 9. Choix des composants du domaine thermique

Ouvrir un « Blank Model » à partir de l'environnement Simulink.



Figure 222 : ouverture d'un «Blank Model » Simulink

Cliquer sur  pour ouvrir la bibliothèque de Simulink.

Pour modéliser l'ensemble nous aurons besoin de la bibliothèque de composants du domaine thermique.

Développer la bibliothèque Simscape pour faire apparaître la bibliothèque **Foundation Library/Thermal** (Figure 223)

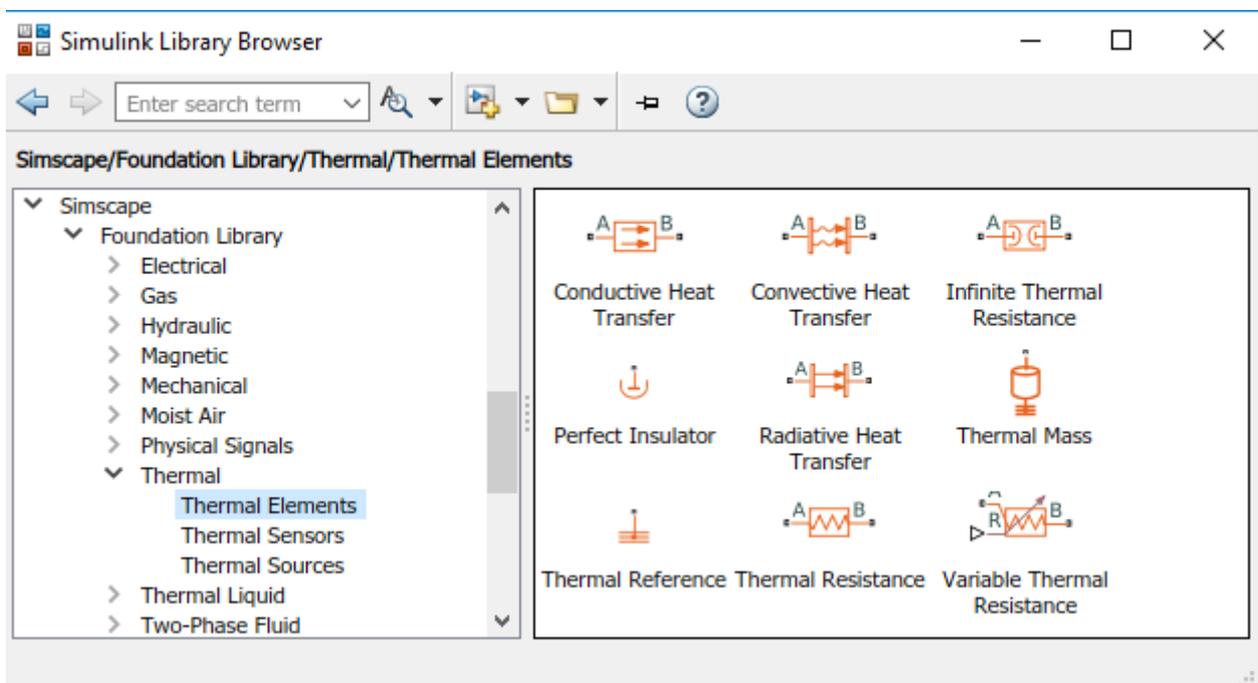


Figure 223 : bibliothèque Simscape Thermal

Pour commencer la démarche de modélisation il faut choisir les composants permettant de modéliser la barre.

La modélisation du chauffage d'une barre par conduction est obtenue en utilisant les composants présentés sur la Figure 224

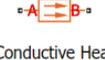
Fonction du composant	Icône	Emplacement dans la bibliothèque
Source de température		Simscape/Foundation Library/ Thermal/Thermal Sources
Transfert de chaleur par conduction	 Conductive Heat Transfer	Simscape/Foundation Library/ Thermal/Thermal Elements
Masse thermique	 Thermal Mass	Simscape/Foundation Library/ Thermal/Thermal Elements
Référence du domaine thermique	 Thermal Reference	Simscape/Foundation Library/ Thermal/Thermal Elements
Bloc solveur	 Solver Configuration	Simscape/Utilities

Figure 224 : les composants nécessaires à la modélisation du chauffage d'une barre par conduction

Ouvrir le fichier « **chauffage\_barre\_0.slx** » ou **Glisser/Déposer** les différents composants à partir des bibliothèques et les disposer dans la fenêtre de travail comme indiqué sur la Figure 225. Vous pouvez vous reporter à la Figure 226 pour prendre connaissance des commandes utiles à l'orientation des composants. Un simple double-clic sur le nom du composant permet de modifier son nom en ajoutant des informations qui améliorent la lisibilité.

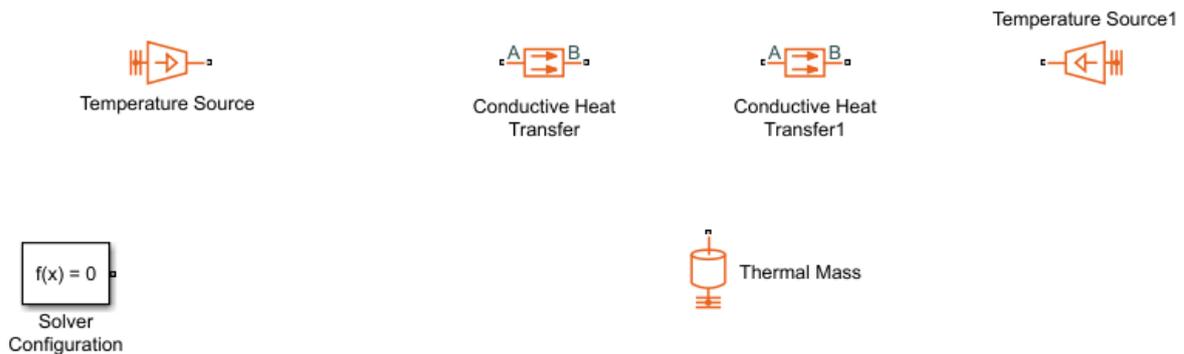


Figure 225 : positionnement des composants dans la fenêtre de travail

Commandes utiles		
Fonctions	Actions	Raccourcis clavier
Rotation d'un composant sens horaire	<b>Clic droit</b> sur le composant, <b>Rotate&amp;Flip/Clockwise</b>	Ctrl+R
Rotation d'un composant sens antihoraire	<b>Clic droit</b> sur le composant, <b>Rotate&amp;Flip/CounterClockwise</b>	Shift+Ctrl+R
Inversion gauche/droite d'un composant	<b>Clic droit</b> sur le composant, <b>Rotate&amp;Flip/Flip Bloc/Left Right</b>	Ctrl+I
Inversion haut/bas d'un composant	<b>Clic droit</b> sur le composant, <b>Rotate&amp;Flip/Flip Bloc/Up Down</b>	
Copier un composant	<b>Clic droit</b> sur le composant à dupliquer, puis <b>glisser/déposer</b> en gardant le bouton droit appuyé	
Supprimer un composant	<b>Clic droit</b> sur le composant puis sélectionner <b>Delete</b>	Suppr
Relier deux composants	<b>Clic gauche</b> sur le port du premier composant puis se déplacer en maintenant le bouton gauche enfoncé jusqu'au port du second composant	

Figure 226 : les commandes de rotation et d'inversion des composants

**Relier** les composants pour obtenir le modèle de la Figure 227 ou ouvrir le fichier « **chauffage\_barre\_1.slx** ».

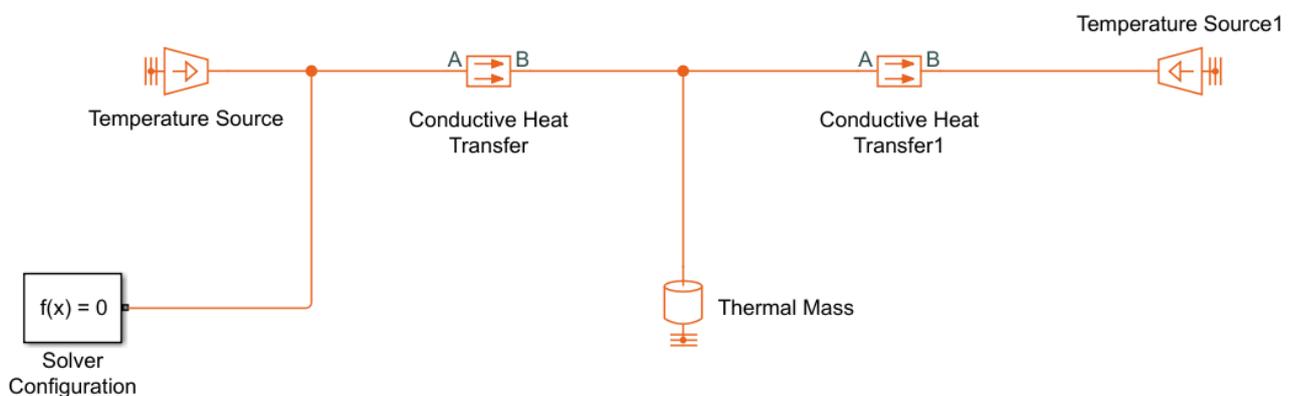


Figure 227 : modèle Simscape du chauffage d'une barre par conduction

## 10. Les ports transmetteurs de puissance (PCP) du domaine thermique

Chaque composant du domaine thermique possède des ports afin de pouvoir se connecter avec les autres composants. Ces ports qui permettent la transmission de puissance entre les composants d'un même domaine physique sont appelés « **Physical Conserving Port** » (PCP) (Figure 228).

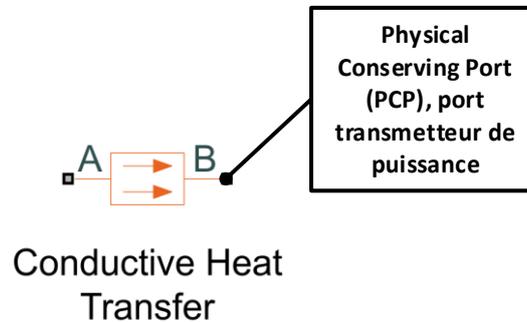


Figure 228 : les ports transmetteurs de puissance (PCP) du domaine thermique

Les connexions que nous venons de réaliser permettent de relier les ports de type PCP de notre modèle. Elles ne sont pas orientées et sont analogues aux liens qui relient les composants dans la réalité. Dans le cas de notre modèle qui ne fait intervenir que le domaine thermique, ces connexions peuvent être assimilées à des conducteurs de flux thermique parfaits, sans résistance thermique. Elles peuvent donc conduire un flux thermique et il est possible de mesurer la température d'un nœud d'une connexion par rapport à un nœud de référence.

## 11. Paramétrage des composants du modèle

Le paramétrage des composants est accessible par un double-clic sur le bloc correspondant.

### Paramétrage du bloc Ideal Temperature Source

Ideal Temperature  
Source



Simscape/Foundation  
Library/Thermal/Thermal Elements

Le modèle possède 2 blocs de ce type. Ces deux blocs permettent de modéliser le fait que les extrémités de la barre sont maintenues à des températures constantes de 300°C et 0°C.

Le paramétrage de ce bloc consiste à indiquer la température de la source qui restera constante tout au long de la modélisation. Il faudra donc indiquer 300 degrés Celsius pour la source de température de gauche et 0 degrés Celsius pour la source de température de droite. Le menu déroulant propose un choix d'unités, il est possible de taper l'unité de son choix si elle n'est pas proposée par défaut (ici degC pour degrés Celsius)

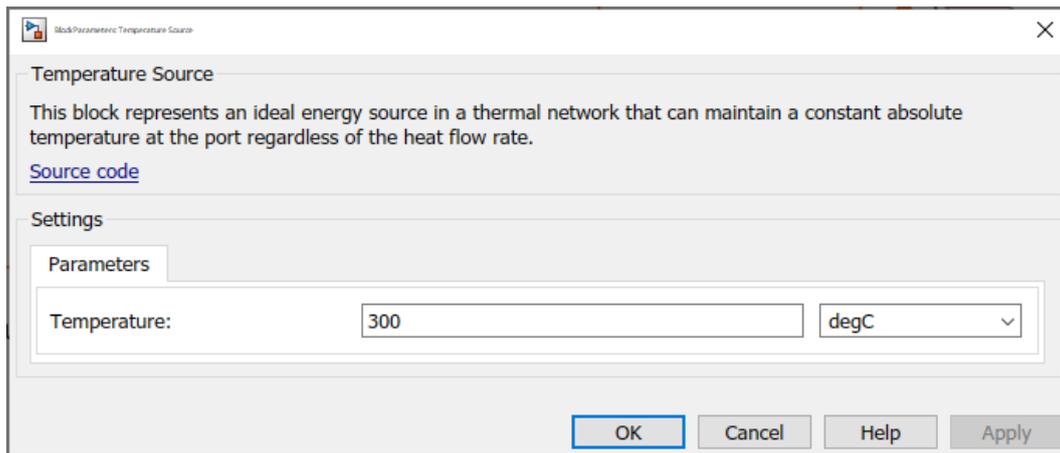


Figure 229 : paramétrage du bloc Temperature Source

## Paramétrage du bloc Conductive Heat Transfert

Conductive Heat  
Transfert



Simscape/Foundation  
Library/Thermal/Thermal Elements

Le modèle contient 2 blocs identiques de cette nature représentant chacun une moitié de la barre en acier.

Ce bloc permet de définir les paramètres de l'échange de chaleur par conduction. Les paramètres à renseigner sont :

**Area** : la surface correspondant à l'échange thermique. Dans notre exemple il s'agit de l'aire de la section de la barre soit  $A=8\text{cm}^2$

**Thickness** : la longueur de l'éléments traversé par le flux thermique. Dans notre exemple, il s'agit de la moitié de la longueur totale de la barre soit 10 cm.

**Thermal Conductivity** : il faut renseigner ici la conductivité thermique du matériau au travers duquel s'effectue l'échange de chaleur par conduction. Dans notre exemple, la conductivité thermique de l'acier est de  $50\text{ W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$

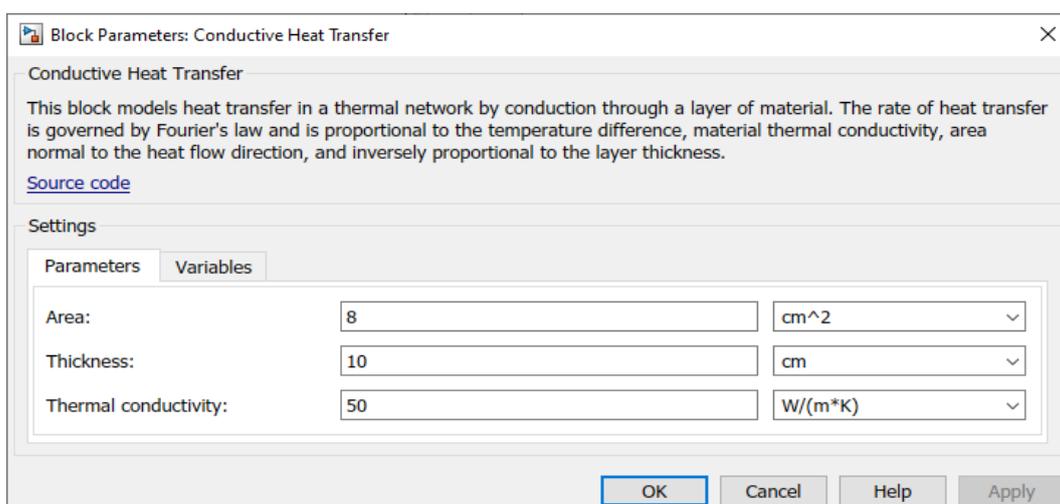


Figure 230 : paramétrage du bloc Conductive Heat Transfer

## Paramétrage du bloc Thermal Mass

Thermal Mass



Simscape/Foundation  
Library/Thermal/Thermal Elements

Ce bloc permet de modéliser le comportement d'une masse thermique. Les paramètres à renseigner sont :

**Mass** : la masse de l'élément considéré. Dans notre exemple la masse de la barre en acier est égale à son volume ( $A \times L$ ) multiplié par sa masse volumique ( $\rho$ ), soit 1,248 kg.

**Specific Heat** : il faut renseigner ici la capacité thermique massique du matériau considéré. Dans notre exemple, la capacité thermique massique de l'acier est de 444 J/Kg/K.

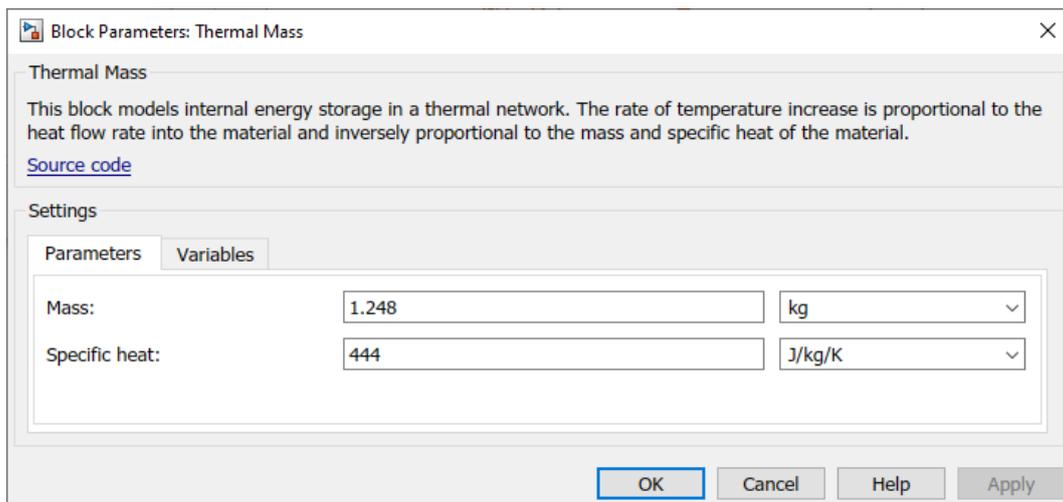
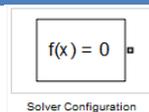


Figure 231 : paramétrage du bloc Thermal Mass

## Paramétrage du bloc Solver

Solver Configuration



Simscape/Utilities

Aucune modification n'est à apporter au paramétrage de ce bloc pour ce modèle. Un bloc **Solver Configuration** doit obligatoirement être relié au modèle afin que le solveur puisse résoudre le modèle. L'absence de ce bloc dans un modèle Simscape provoque un message d'erreur au lancement de la simulation.

## 12. Mise en place des capteurs du domaine thermique

Les capteurs du domaine thermique permettent de prélever :

- Une température
- Un flux thermique

Les grandeurs physiques à prélever sur le modèle sont les suivantes :

- $T_A$  : température au milieu de la barre
- Le flux thermique à l'entrée et à la sortie de la barre
- Le flux thermique absorbé ou restitué par la masse thermique

Afin de pouvoir relever ces grandeurs physiques, des capteurs doivent être placés sur le modèle. Nous allons implanter un capteur de température au milieu de la barre et 3 capteurs de flux thermique.

L'implantation des capteurs doit respecter des règles strictes correspondant au caractère Across ou Through de la variable à prélever.

### Le capteur de température

La température est une grandeur physique de type « Across » qui se mesure en se plaçant en parallèle dans le modèle.

Les ports de communication du capteur de température du domaine thermique sont décrits sur la Figure 232.

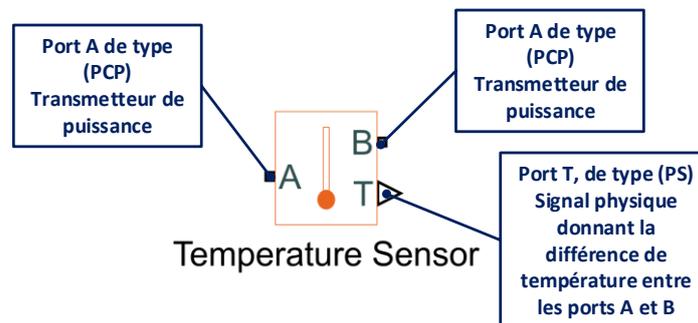


Figure 232 : les ports du capteur de température du domaine thermique

**Port A** : port de type (PCP) qui doit être relié au point du domaine physique dont on veut connaître la température par rapport au nœud relié au port B du capteur.

**Port B** : port de type (PCP) qui doit être relié au point du domaine physique qui sert de référence pour mesurer la température. Ce point peut être relié à un composant de type **Thermal Reference**.

**Port T** : port de type **Physical Signal (PS)** donnant la différence de température entre les ports A et B du capteur.

Les signaux de type Physical Signal (PS) contiennent une grandeur physique qui possède la dimension de la grandeur physique mesurée par le capteur. Elle pourra être affichée dans un scope en ayant préalablement fixé son unité à l'aide d'un bloc adapté. L'unité d'affichage devra obligatoirement être homogène à la grandeur physique mesurée.

Le capteur de température et de position **est orienté positivement de A vers B**.

La Figure 233 montre l'implantation du capteur de température.

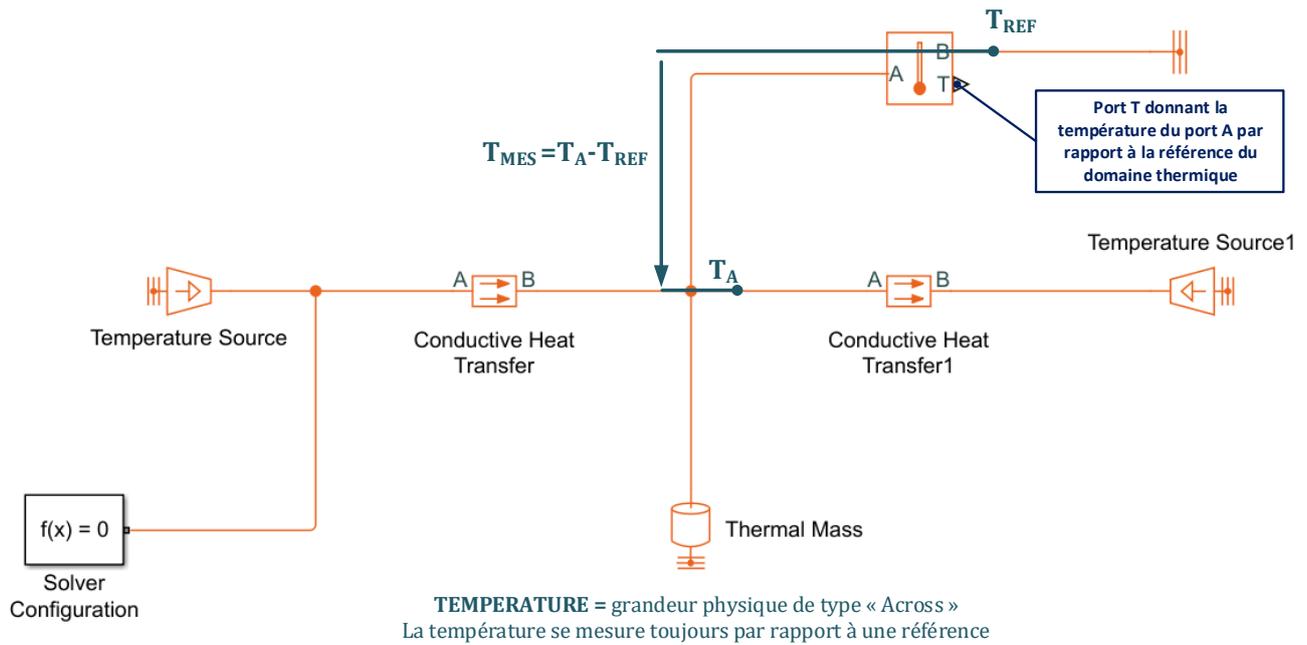
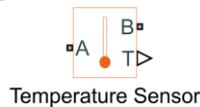


Figure 233 : implantation du capteur de température

### Paramétrage du bloc Ideal Temperature Sensor

Ideal Temperature  
Sensor



Simscape/Foundation  
Library/Thermal/Thermal Sensors

Ce composant permet de mesurer la température d'un nœud d'une connexion par rapport à une référence. Aucun paramétrage n'est nécessaire pour ce bloc.

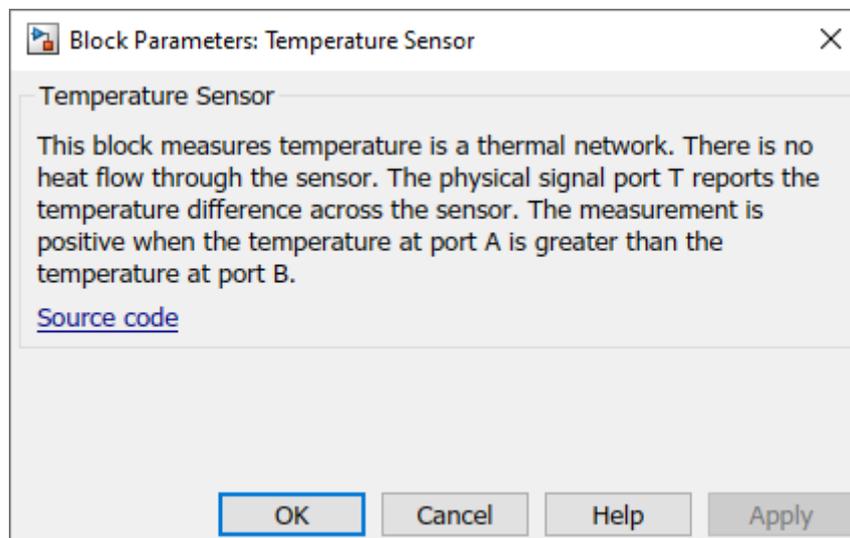


Figure 234 : paramétrage du bloc Ideal Temperature Source

Afin de pouvoir relever la température il faut ajouter un scope à notre modèle et convertir le signal physique dans la bonne unité pour l'affichage dans le scope.

**Glisser/Déposer** les composants de la Figure 235 dans le modèle du chauffage d'une barre par conduction et les relier afin d'obtenir la configuration de la Figure 236.

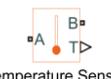
Capteur de température	 Temperature Sensor	Simscape/Foundation Library/ Thermal/Thermal Sensors
Conversion d'un signal physique en signal numérique	 PS-Simulink Converter	Simscape/Utilities
Afficheur de signaux numériques	 Scope	Simulink/Commonly Used Blocks

Figure 235 : les composants associés au capteur de température

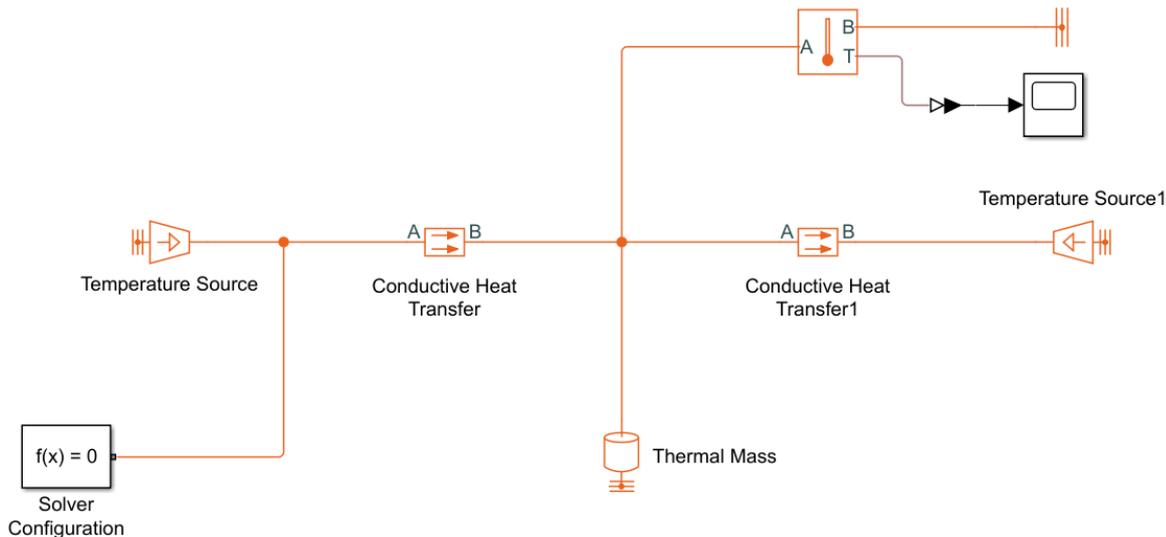


Figure 236 : implantation du capteur de température

### Paramétrage du bloc PS-Simulink Converter

PS – Simulink  
Converter



Simscape/Utilities

Ce composant permet de convertir le signal physique (PS) issu d'une grandeur mesurée en signal **Simulink** (S) afin de pouvoir être affiché dans un Scope. Il est souhaitable de préciser l'unité de la grandeur physique souhaitée pour l'affichage dans Simulink. Si l'unité n'est pas spécifiée par l'utilisateur (valeur par défaut 1 ou inherit), la grandeur sera affichée dans le scope en unités SI.

Le menu déroulant propose un choix d'unités, il est possible de taper l'unité de son choix si elle n'est pas proposée par défaut.

Ce bloc permet de spécifier l'unité de la grandeur physique qui sera affichée dans le scope. Ici, nous utilisons les degrés Celsius. Cela nécessite deux paramétrages :

**Output signal unit** : degC

Cocher la case **Apply affine conversion** pour assurer la conversion des Kelvin en degrés Celsius.

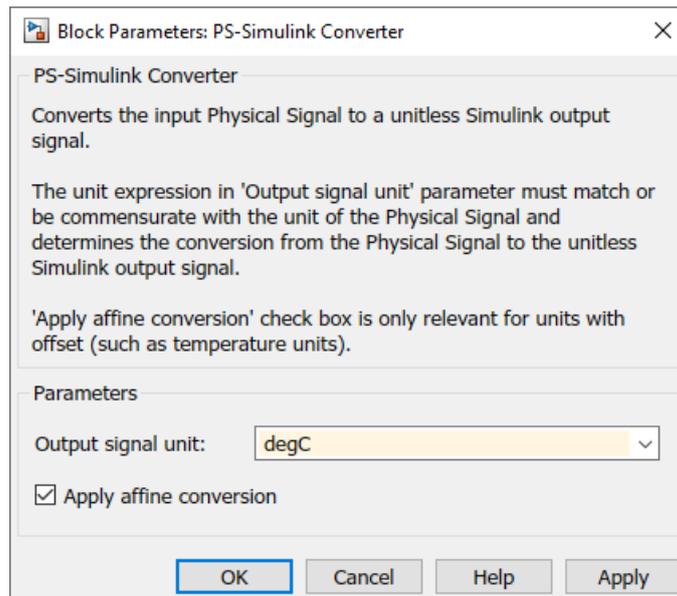


Figure 237 : paramétrage du bloc PS – Simulink Converter

### Le capteur de flux thermique

Le flux thermique est une grandeur physique traversante de type « Through » qui se mesure en se plaçant en série dans une connexion.

Les ports de communication de capteur de flux thermique sont décrits sur la Figure 238.

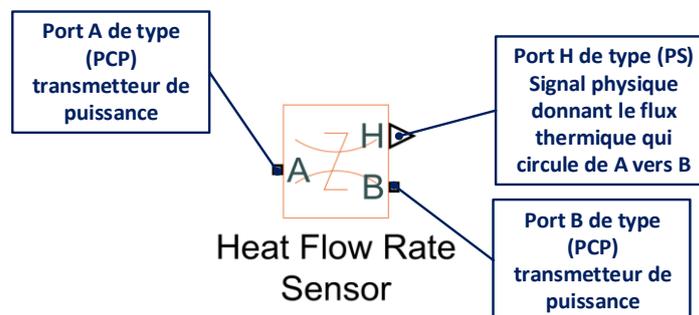


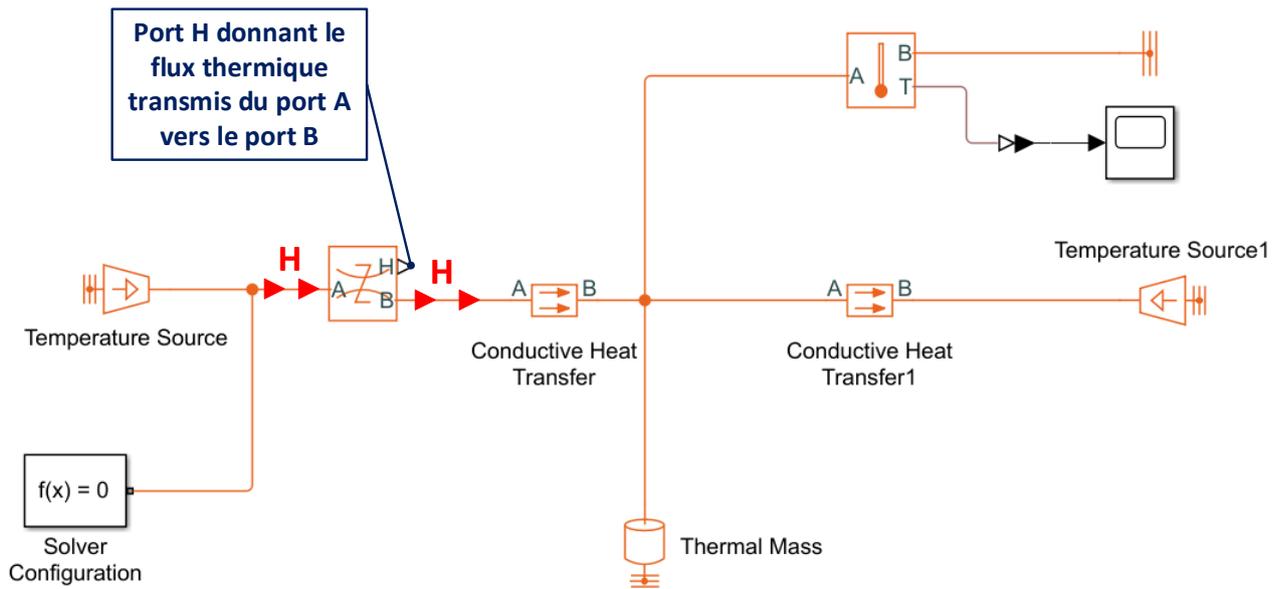
Figure 238 : les ports du capteur de flux thermique

Pour mettre en place le capteur de flux thermique, il faut donc ouvrir une connexion et placer le capteur en série.

**Port A** : port de type (PCP) qui doit être relié à l'un des points correspondant à l'ouverture de la connexion.

**Port B** : port de type (PCP) qui doit être relié à l'autre point correspondant à l'ouverture de la connexion.

**Port H** : port de type **Physical Signal (PS)** donnant le flux thermique qui traverse la connexion.



**FLUX THERMIQUE** = grandeur de type « Through »  
 Le flux thermique se mesure toujours en série dans une connexion

Figure 239 : mesure d'une grandeur physique de type flux thermique

Si le flux thermique est transmis du port A vers le port B, le capteur mesure un flux thermique positif. En cas d'inversion des ports A et B du capteur de flux thermique, le signe de la mesure est inversé.

### Paramétrage du bloc Ideal Heat Flow Sensor

Ideal Heat Flow Sensor	 Heat Flow Rate Sensor	Simscape/Foundation Library/Thermal/Thermal Sensors
------------------------	--	--

Ce composant permet de mesurer le flux thermique qui traverse une connexion du domaine thermique. Ce bloc ne nécessite aucun paramétrage.

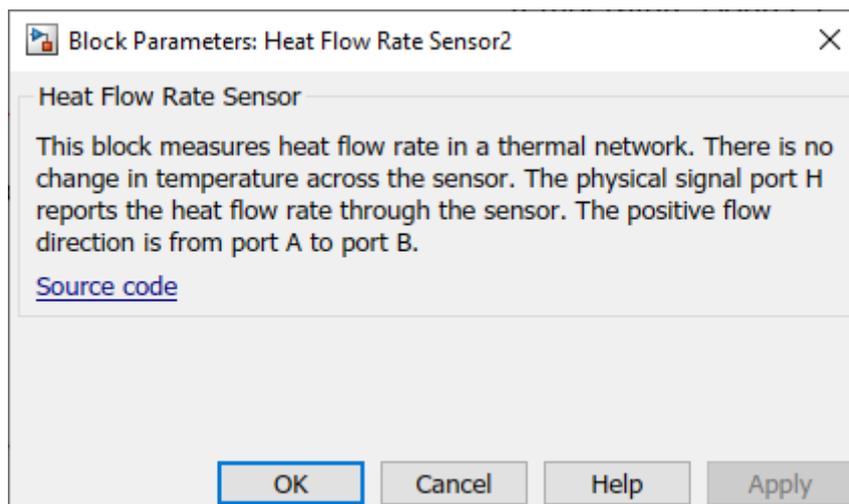


Figure 240 : paramétrage du bloc Ideal Heat Flow Sensor

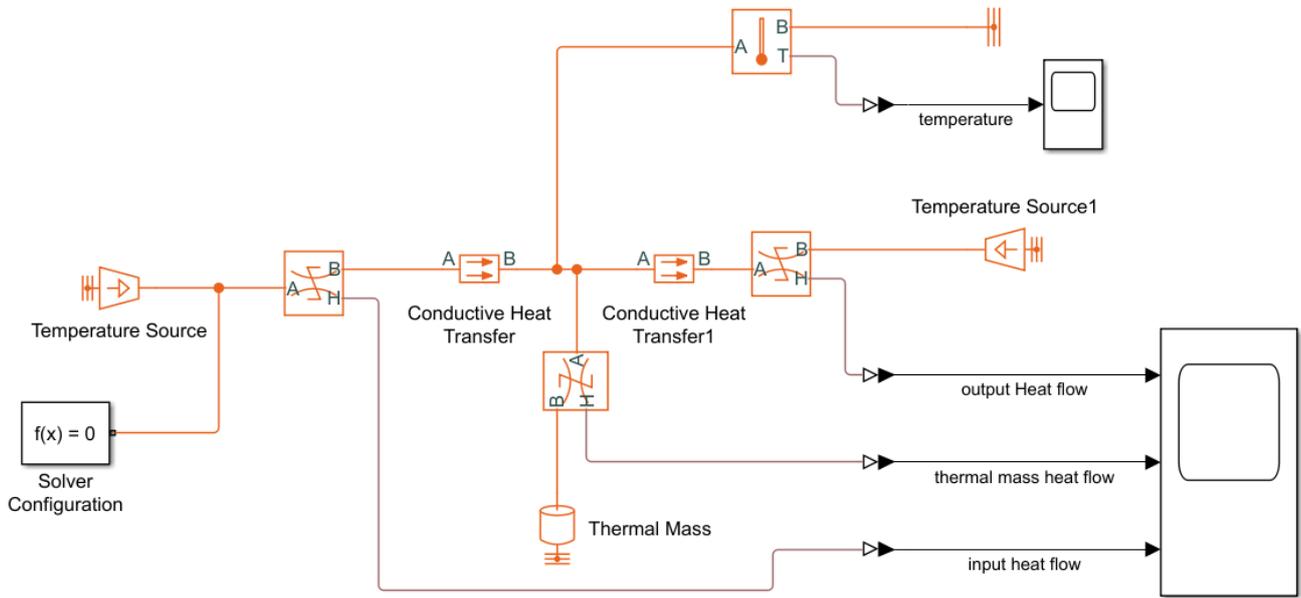
Nous allons implanter trois capteurs de flux thermique dans le modèle.  
Ouvrir le fichier «**chauffage\_barre\_3.slx**» ou **Glisser/Déposer** les composants de la Figure 241 dans le modèle afin d'obtenir la configuration de la Figure 242.

Capteur de température		Simscape/Foundation Library/ Thermal/Thermal Sensors
Conversion d'un signal physique en signal numérique		Simscape/Utilities
Afficheur de signaux numériques		Simulink/Commonly Used Blocks

**Figure 241 : les composants associés au capteur de flux thermique**

Dans notre exemple, nous souhaitons mesurer le flux thermique qui circule à chaque extrémité de la barre ainsi que celui absorbé ou restitué par la masse thermique. Les connexions ont été ouvertes afin de pouvoir implanter les trois capteurs de flux en série.

Le **port H** permet de transmettre le signal physique contenant le flux thermique qui traverse la connexion. Une inversion des ports A et B du capteur entraîne une inversion du signe de la grandeur physique mesurée.



**Figure 242 : implantation des capteurs de flux thermique dans le modèle**

### Paramétrage du bloc PS-Simulink Converter

PS – Simulink  
Converter

  
PS-Simulink  
Converter

Simscape/Utilities

Ce composant permet de convertir le signal physique (PS) issu d'une grandeur mesurée en signal **Simulink** (S) afin de pouvoir être affiché dans un Scope. Il est souhaitable de préciser l'unité de la grandeur physique souhaitée pour l'affichage dans Simulink. Si l'unité n'est pas spécifiée par l'utilisateur (valeur par défaut 1 ou inherit), la grandeur sera affichée dans le scope en unités SI.

Le menu déroulant propose un choix d'unités, il est possible de taper l'unités de son choix si elle n'est pas proposée par défaut.

Pour les blocs **PS-Simulink Converter** qui mesure des flux thermiques, choisir **J/s ou W** (Joule par seconde ou Watt) dans la case « Output Signal Unit »

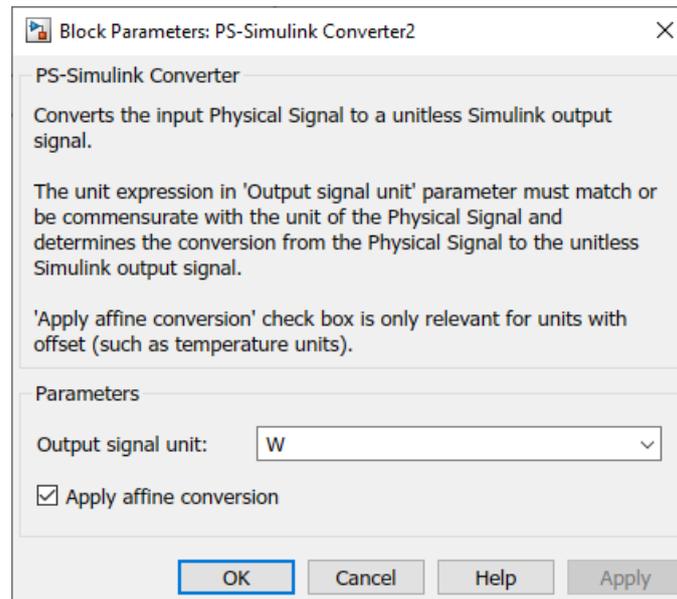


Figure 243 : paramétrage du bloc PS – Simulink Converter

### 13. Définition des conditions initiales du modèle

La phase de modélisation étant maintenant terminée, MATLAB doit résoudre le modèle. Afin de pouvoir mener les calculs, il faut préciser les conditions initiales du modèle, c'est-à-dire les valeurs que prendront les variables de notre modèle à l'instant  $t=0$ , instant initial de la simulation. Si rien n'est précisé par l'utilisateur, toutes les variables auront, par défaut, une valeur nulle en début de simulation. Si cette configuration correspond à la réalité de ce que l'on souhaite simuler, aucune opération n'est requise à ce niveau. Par contre si des variables ont des valeurs différentes de 0 en début de simulation, leurs valeurs devront être précisées.

Dans notre exemple, à  $t=0$ , la température de la barre est de  $20^{\circ}\text{C}$ . Cette information doit être indiquée dans les conditions initiales de la simulation.

Pour cela double cliquer sur le bloc **Thermal Mass**, et sélectionner l'onglet **Variables**.

**Cocher** la case température et indiquer une température initiale de **20 degC** comme indiqué sur la Figure 244.

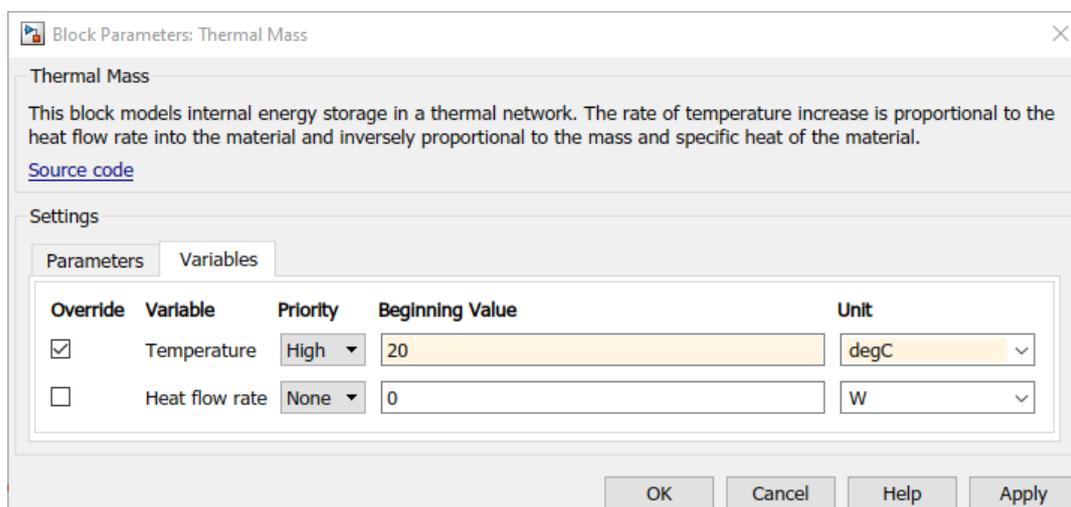


Figure 244 : spécification de la température initiale de la barre

## 14. Réglage du temps de simulation

La phase de modélisation étant terminée et les conditions initiales définies, MATLAB doit résoudre le modèle et une méthode de résolution numérique doit être mise en œuvre. L'utilisateur doit définir le temps de simulation et lancer le calcul (Figure 245).

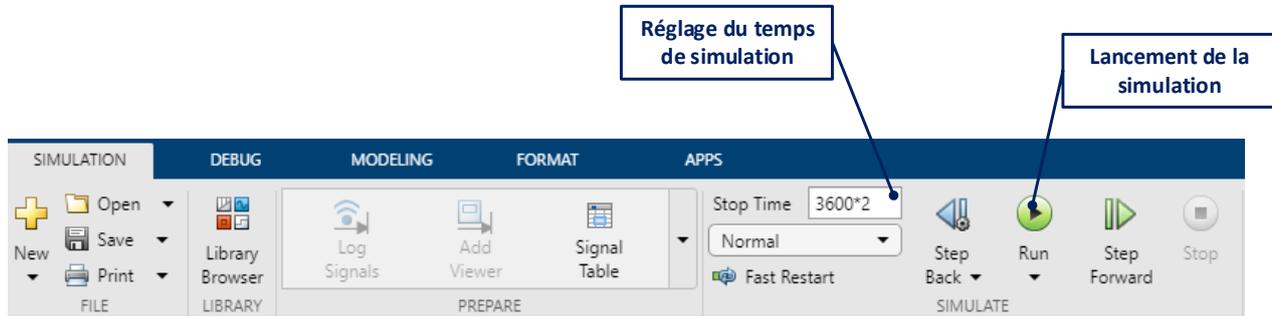


Figure 245 : configuration des paramètres du solveur

Nous choisirons un temps de simulation de **2 heures** soit **3600\*2 s**. Le temps de simulation doit être spécifié directement dans la barre d'outil (Figure 245).

## 15. Lancement de la simulation et exportation des résultats

Le fichier « **chauffage\_barre\_3.slx** » contient le modèle entièrement paramétré et prêt à être exécuté.

Le temps de simulation étant réglé sur 3600\*2 secondes, **lancer** la simulation en appuyant sur  dans la barre de commande Figure 245.

Ouvrir les scopes donnant la température au milieu de la barre et les flux thermiques. Pour mettre en forme les courbes dans les scopes, vous pouvez vous rendre à l'annexe paramétrage des scopes en fin d'ouvrage.

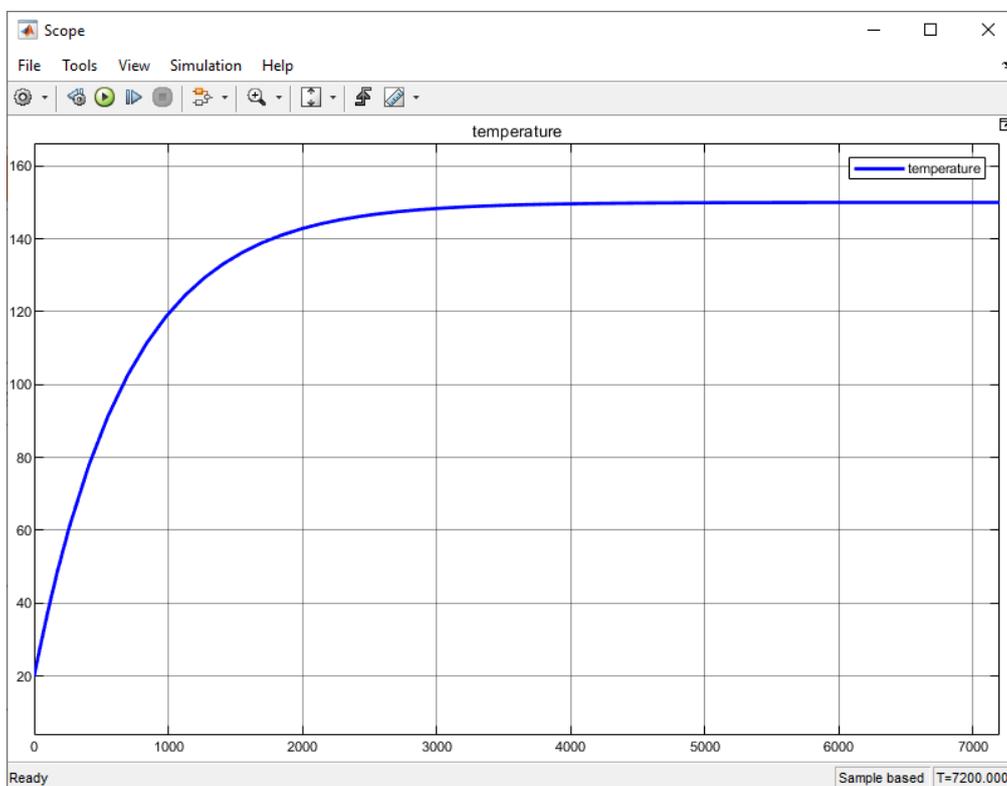


Figure 246 : température au milieu de la barre en fonction du temps

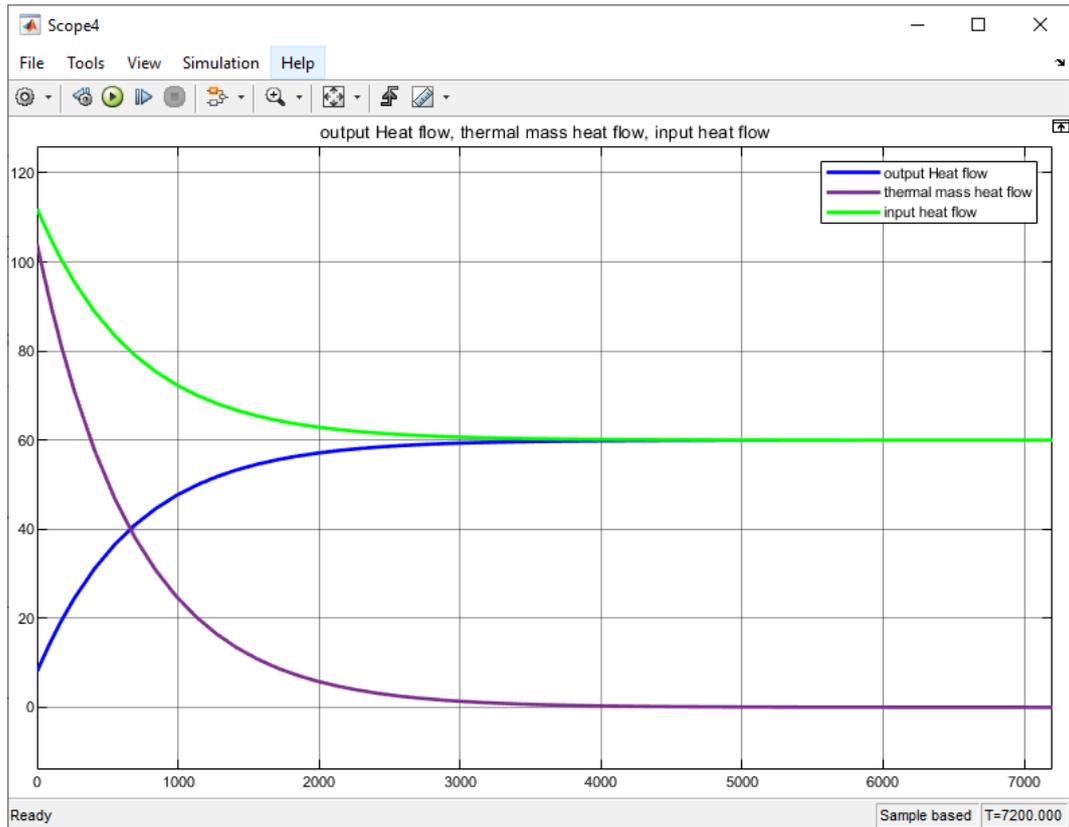


Figure 247 : flux thermiques mesurés sur le modèle

## 16. Interprétation des résultats de la simulation

La Figure 246 nous permet de voir que le régime permanent s'établit au bout de  $t_{5\%} = 2016 \text{ s}$  avec une température au centre de la barre  $T_A = 150^\circ\text{C}$ .

Analysons maintenant les mesures de flux thermique que l'on peut observer sur la Figure 247.

Le flux thermique  $H_i$  à l'extrémité chaude de la barre se divise en deux flux thermiques,  $H_{tm}$  vers la masse thermique et  $H_o$  vers l'extrémité froide de la barre. Il est aisé de vérifier la relation  $H_i = H_{tm} + H_o$

Le flux thermique  $H_{tm}$  tend logiquement vers 0 en régime permanent et l'inertie thermique de la masse n'a alors plus d'influence.

Le flux thermique  $H$  qui traverse la barre en régime permanent s'établit à  $H = 60 \text{ W}$ .

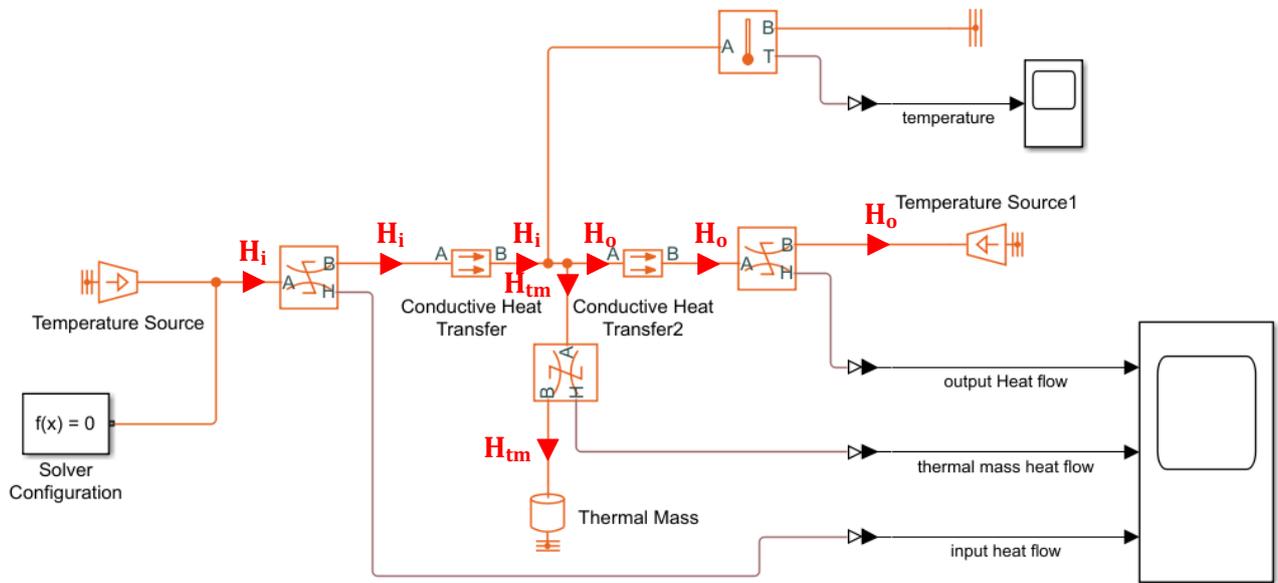


Figure 248 : circulation des flux thermiques dans les différentes parties du modèle

Il est très simple de modifier le modèle pour vérifier la loi des nœuds pour le flux thermique (variable Through) :  $H_i = H_{tm} + H_o$ .

Le fichier « **chauffage\_barre\_4.slx** » contient le modèle de la Figure 249.

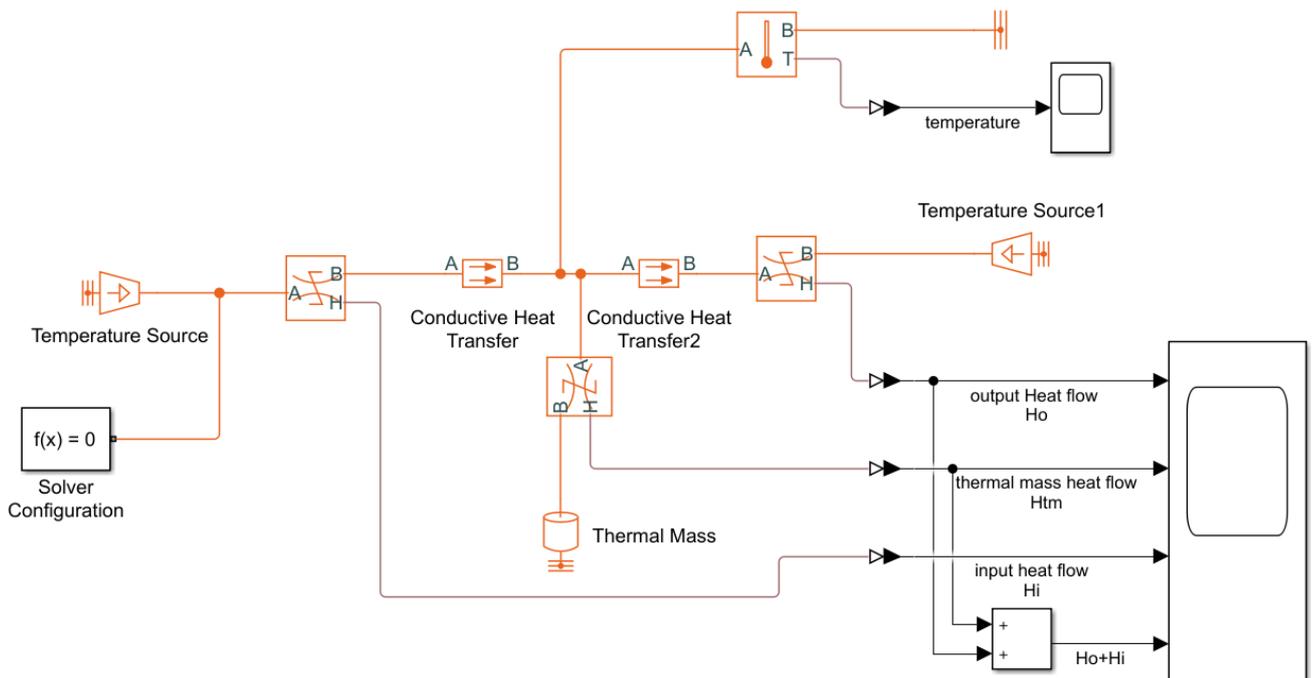


Figure 249 : vérification de la conservation du flux thermique dans le modèle

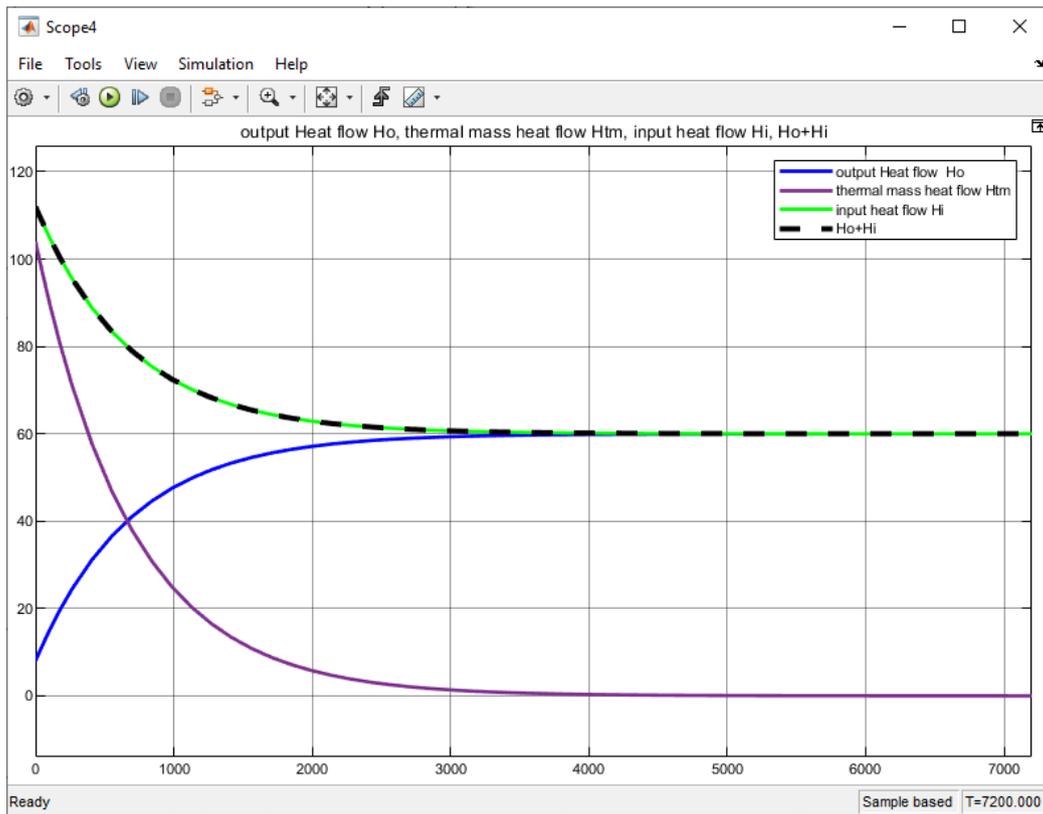


Figure 250 : visualisation de la conservation du flux thermique dans le scope

*Prise en compte des pertes*

On considère maintenant que la barre n'est plus isolée mais laissée à l'air libre. Il se produit alors des échanges de chaleur par convection sur toute la surface extérieure de la barre. Le coefficient d'échange de chaleur par convection entre l'air et la barre d'acier vaut  $k=32 \text{ W}/(\text{m}^2.\text{K})$  pour les condition de l'expérience. La surface d'échange correspond à la surface extérieure du cylindre  $S_{ext} = \pi DL = 201 \text{ cm}^2$

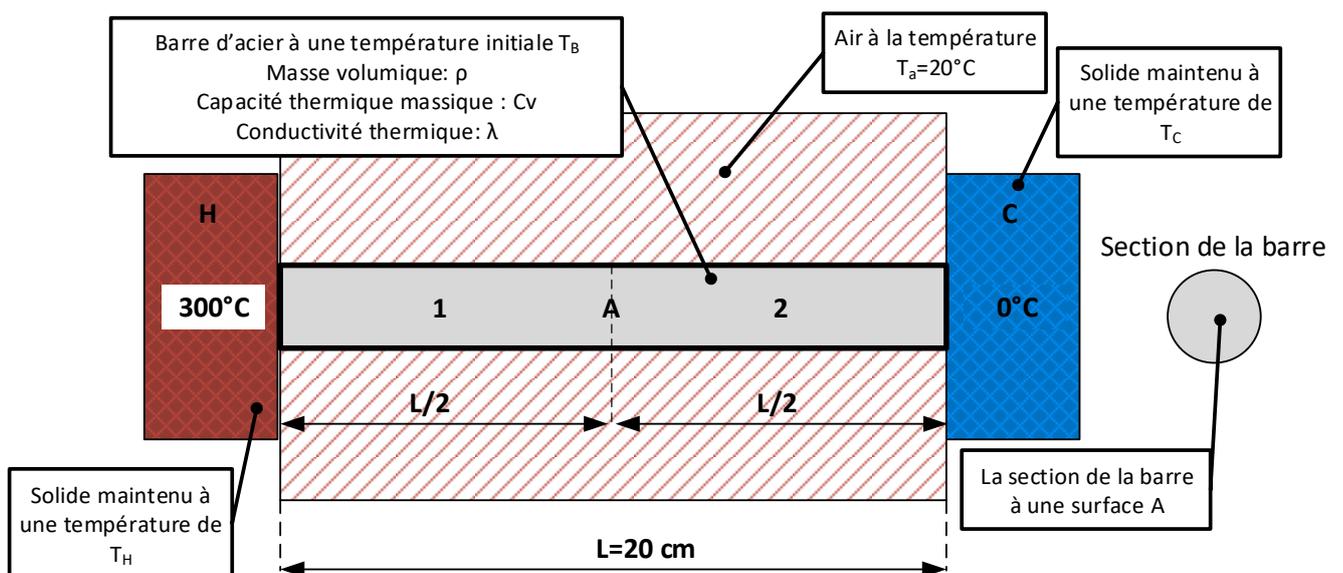


Figure 251 : chauffage d'une barre par conduction non isolée

Afin de modéliser l'échange de chaleur par convection entre la surface extérieure de la barre et l'air, nous allons ajouter un bloc **Convective Heat Transfer** et un bloc **Temperature Source** pour modéliser la température de l'air qui reste constante (Figure 252).

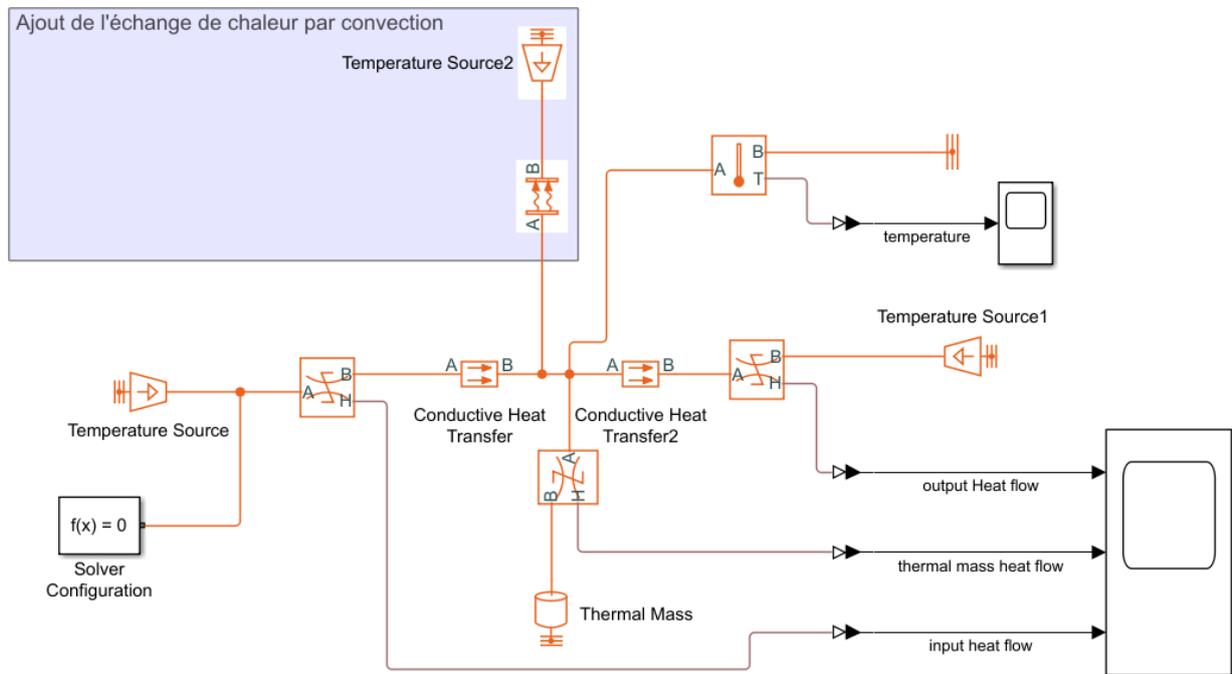


Figure 252 : modélisation de l'échange de chaleur par convection

Le fichier « **chauffage\_barre\_conv\_1.slx** » contient le modèle de la Figure 252.

### Paramétrage du bloc Ideal Temperature Source

Ideal Temperature Source



Simscape/Foundation Library/Thermal/Thermal Elements

Ce bloc permet de modéliser la température de l'air qui reste constante à 20°C.

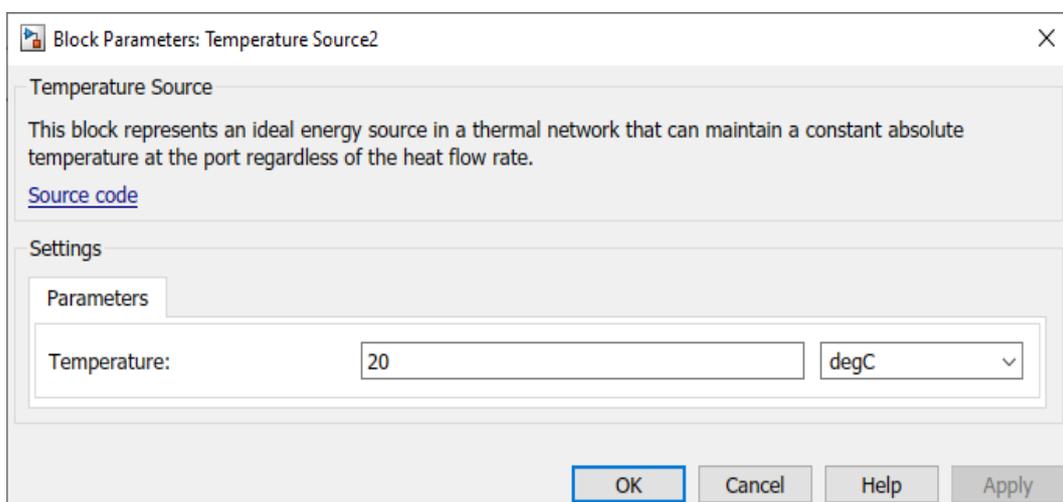


Figure 253 : paramétrage du bloc Temperature Source

## Paramétrage du bloc Convective Heat Transfert

Convective Heat  
Transfert



Simscape/Foundation  
Library/Thermal/Thermal Elements

Ce bloc permet de définir les paramètres de l'échange de chaleur par convection. Les paramètres à renseigner sont :

**Area** : la surface correspondant à la surface d'échange thermique par convection. Dans notre exemple cela correspond à la surface extérieure du cylindre  $S_{ext} = \pi DL = 201 \text{ cm}^2$ .

**Heat Transfer Coefficient** : il faut renseigner ici le coefficient de transfert thermique par convection entre l'air et l'acier pour les condition de l'expérience,  $k=32 \text{ W}/(\text{m}^2.\text{K})$ .

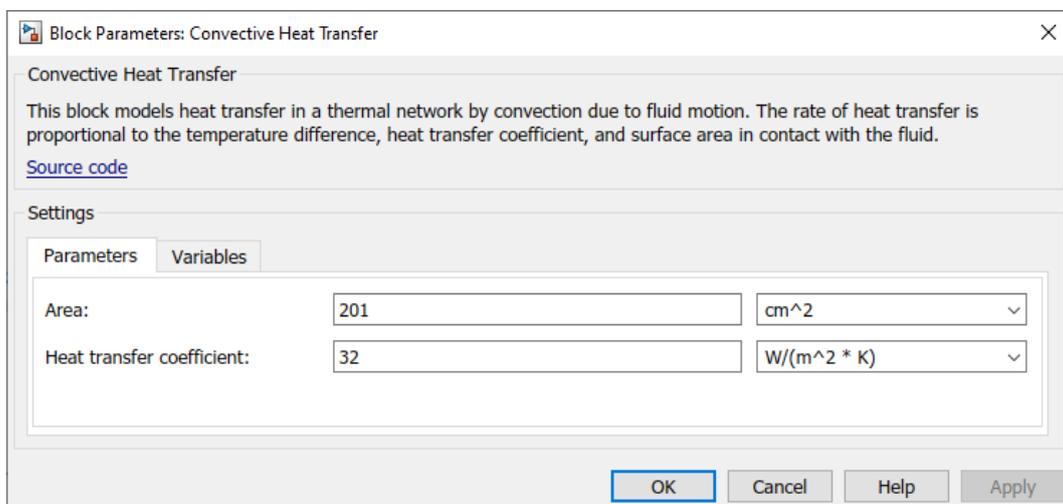


Figure 254 : paramétrage du bloc Convective Heat Transfer

Le temps de simulation étant réglé sur 3600\*2 secondes, **lancer** la simulation en appuyant sur  dans la barre de commande Figure 245. Ouvrir les scopes donnant la température au milieu de la barre et les flux thermiques. Pour mettre en forme les courbes dans les scopes, vous pouvez vous rendre à l'annexe paramétrage des scopes en fin d'ouvrage.

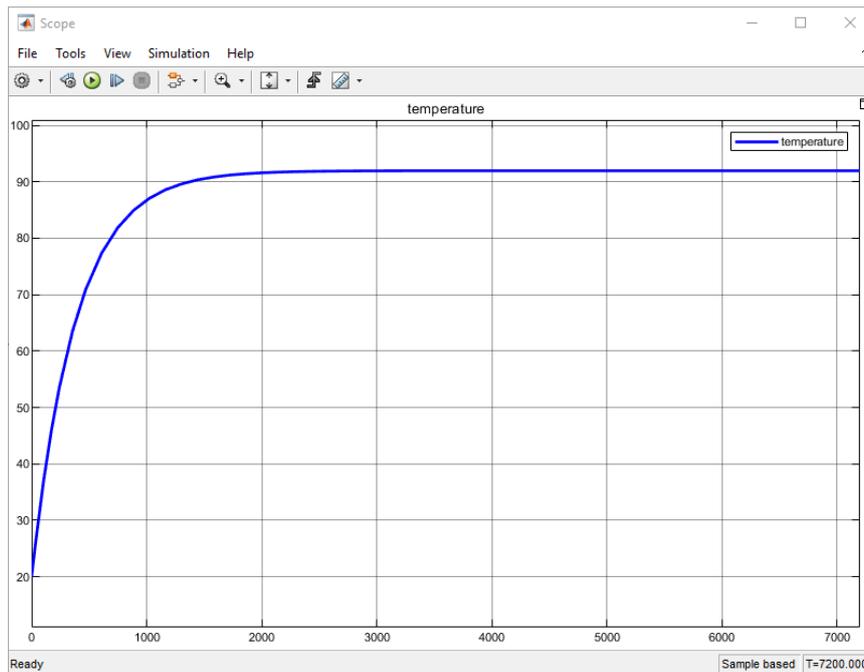


Figure 255 : température au milieu de la barre en fonction du temps avec pertes par convection

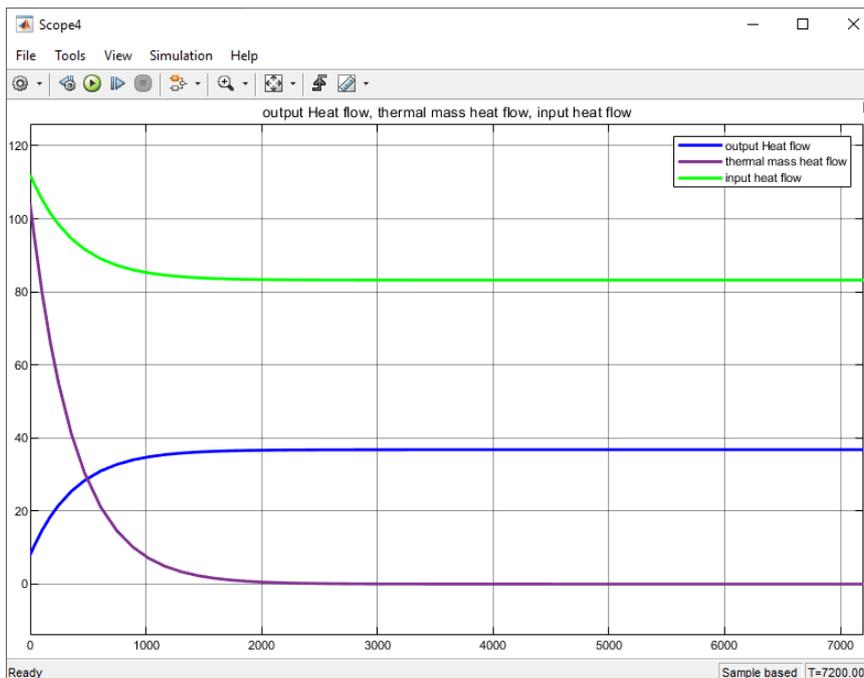


Figure 256 : flux thermiques mesurés sur le modèle avec pertes par convection

### Interprétation des résultats de la simulation

La Figure 255 nous permet de voir que le régime permanent s'établit au bout de  $t_{5\%} = 1053$  s avec une température au centre de la barre  $T_A = 92^\circ\text{C}$ . Nous constatons que la température de la barre en régime permanent est bien inférieure à celle obtenue lorsque la barre était isolée.

Nous pouvons remarquer que le flux entrant dans la barre au niveau de la source de température de  $300^\circ\text{C}$  (input heat flow) n'est plus égale au flux qui arrive au niveau de la source de température à  $0^\circ\text{C}$  (output heat flow). En effet le flux thermique de convection qui correspond aux pertes est maintenant pris en compte dans le modèle.

Pour évaluer ce flux thermique, il est possible d'ajouter un capteur de flux thermique dans notre modèle (Figure 257).

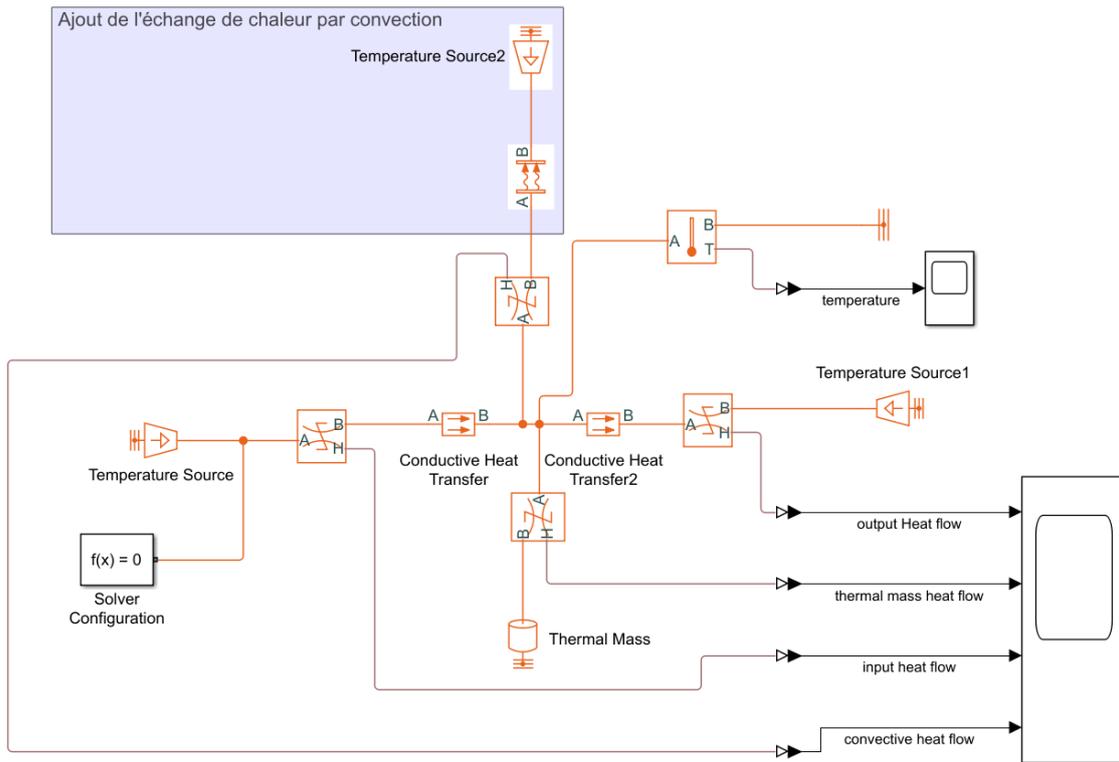


Figure 257 : ajout d'un capteur pour mesure le flux thermique de convection

Le fichier « **chauffage\_barre\_conv\_2.slx** » contient le modèle de la Figure 257.

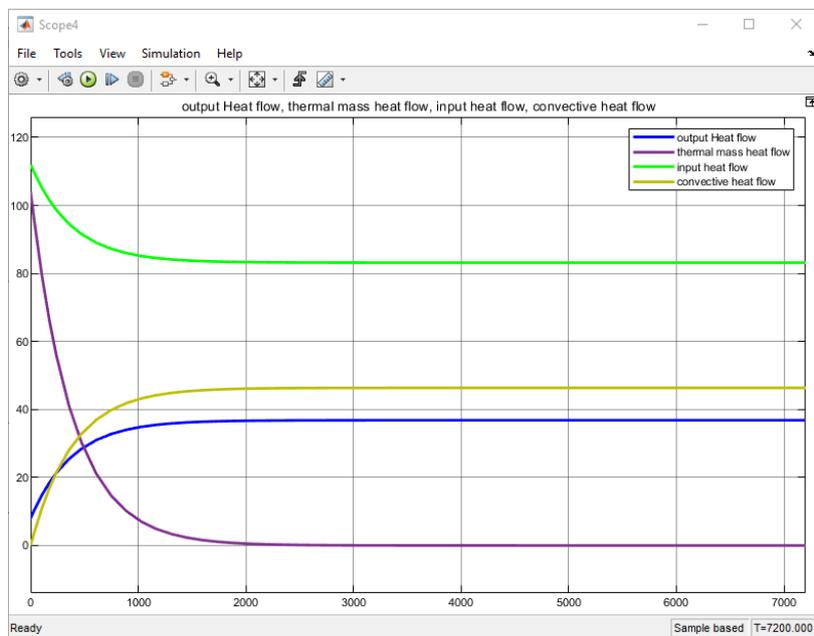


Figure 258 : visualisation de la conservation du flux thermique

Nous pouvons vérifier aisément sur le Figure 258 la conservation du flux thermique. Le flux thermique à l'entrée est bien égal à la somme de tous les autres flux.

## 17. Rappels théoriques sur les échanges thermiques à travers une paroi d'un bâtiment

Les échanges thermiques à travers une paroi d'un bâtiment sont de différentes natures. Des échanges par convection ont lieu sur la surface extérieure et sur la surface intérieure de la paroi. Des échanges thermiques par conduction ont lieu au travers des différentes couches de la paroi.

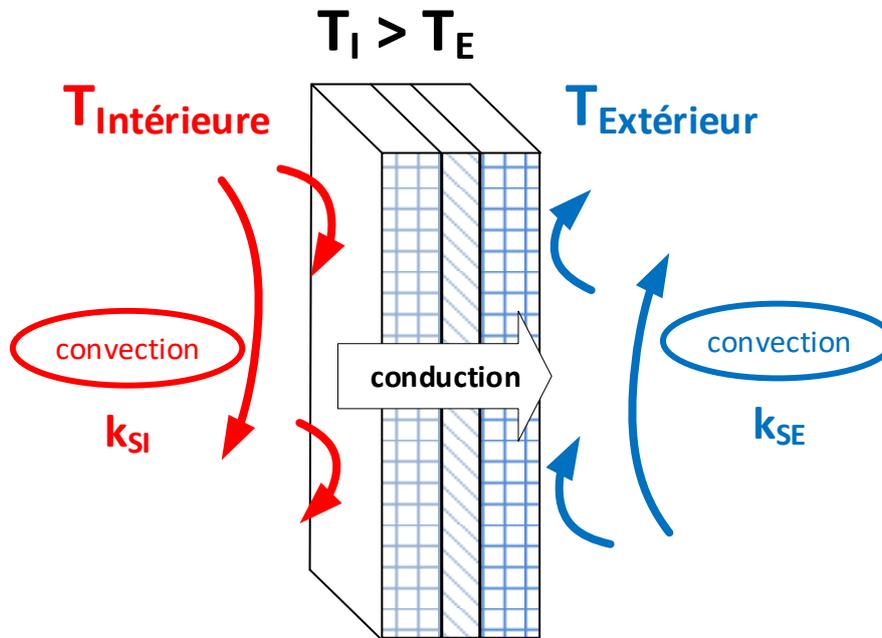


Figure 259 : les échanges thermiques à travers une paroi

### Modélisation des échanges par convection sur la surface de la paroi

Il faudra utiliser le bloc **Convective Heat Transfer** en utilisant les valeurs du coefficient d'échange thermique par convection **k** regroupées dans les tableaux de la Figure 260 et de la Figure 261.

Paroi opaque			
Coefficient de transfert thermique par convection			
	$k_{si}$ en $W/(m^2.K)$	$k_{se}$ en $W/(m^2.K)$	
Paroi verticale	7.7	25	
Paroi horizontale (flux ascendant)	10	25	
Paroi horizontale (flux descendant)	5.9	25	

Figure 260 : coefficient de transfert thermique par convection

Paroi vitrée		
Coefficient de transfert thermique par convection		
	$k_{SI}$ en $W/(m^2.K)$	$k_{SE}$ en $W/(m^2.K)$
Paroi verticale (flux horizontal)	7.7	25
Paroi horizontale (flux ascendant)	10	25

Figure 261 : coefficient de transfert thermique par convection

### Modélisation des échanges par conduction au travers des différentes couches de la paroi

Il faudra utiliser le bloc **Conductive Heat Transfer** en utilisant les valeurs de la conductivité thermique des matériaux usuels regroupées dans le tableau de la Figure 262.

Conductivité thermique			
Matériau	$\lambda$ en $W/(m.K)$	Matériau	$\lambda$ en $W/(m.K)$
Béton plein	1.75	Air	0.028
Brique pleine	0.74	Liège	0.05
Enduit ciment	1.15	Aluminium	230
Plâtre	0.5	Cuivre	380
Bois	0.23	Acier	50
Laine de verre	0.04	Eau	0.66

Figure 262 : conductivité thermique des matériaux usuels

## 18. Modélisation avec Simscape des échanges par conduction et convection dans une paroi monocouche

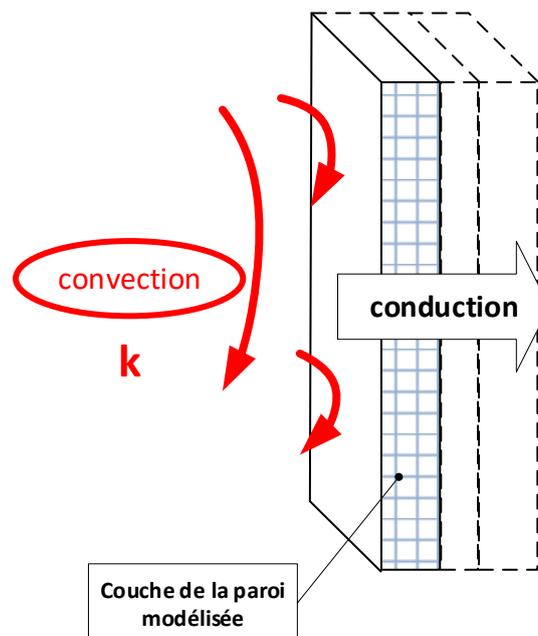


Figure 263 : modélisation d'une couche de la paroi

Il est possible de modéliser les échanges thermiques de deux manières différentes :

- En utilisant les blocs **Conductive Heat Transfer**, **Convective Heat Transfer**, **Radiative Heat Transfer**
- En utilisant le bloc **Thermal Resistance** qui introduit le concept de résistance thermique équivalente

#### Utilisation des blocs *Conductive Heat Transfer*, *Convective Heat Transfer*

La Figure 264 montre la correspondance entre les éléments de modélisation de Simscape et les phénomènes physiques intervenants dans les transferts thermiques au travers d'une couche de la paroi.

#### Caractéristiques de la couche de la paroi:

- Surface **A**
- Épaisseur **e**
- Conductivité thermique  $\lambda$
- Chaleur spécifique **C**
- Masse volumique  $\rho$

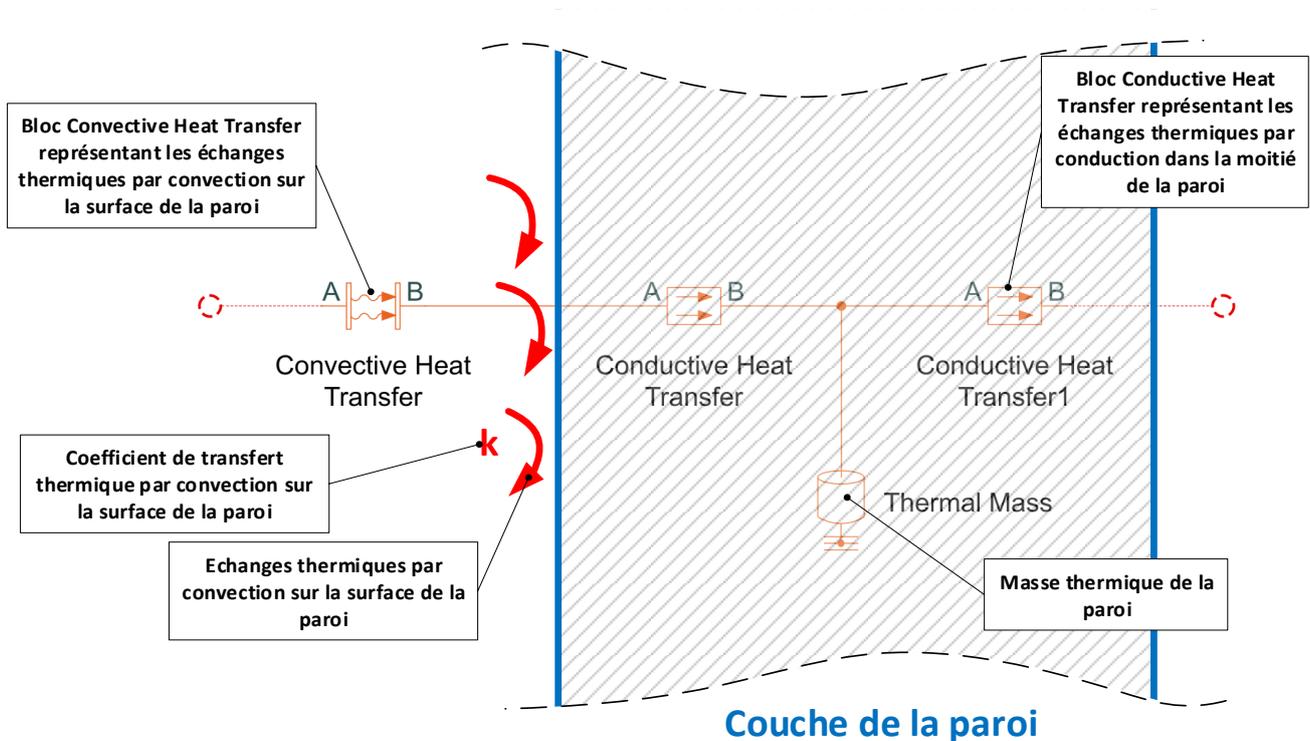


Figure 264 : modélisation Simscape d'une couche de la paroi

Les échanges thermiques par convection sur la surface de la paroi sont modélisés à l'aide du bloc **Convective Heat Transfert**.

Les échanges thermiques par conduction sont représentés par deux blocs **Conductive Heat Transfert** paramétrés de manière identique :

La masse thermique de la paroi est modélisée à l'aide du bloc **Thermal Mass**.

## Paramétrage du bloc Convective Heat Transfert

Convective Heat  
Transfert



Simscape/Foundation  
Library/Thermal/Thermal Elements

Ce bloc permet de définir les paramètres de l'échange de chaleur par convection entre l'air et la surface de la paroi.

**Area** : Surface d'échange thermique par convection. Pour l'exemple de la paroi cette surface correspond à la surface **S** de la paroi

**Heat Transfer Coefficient** : le coefficient de transfert thermique par convection entre l'air et la paroi est **k**.

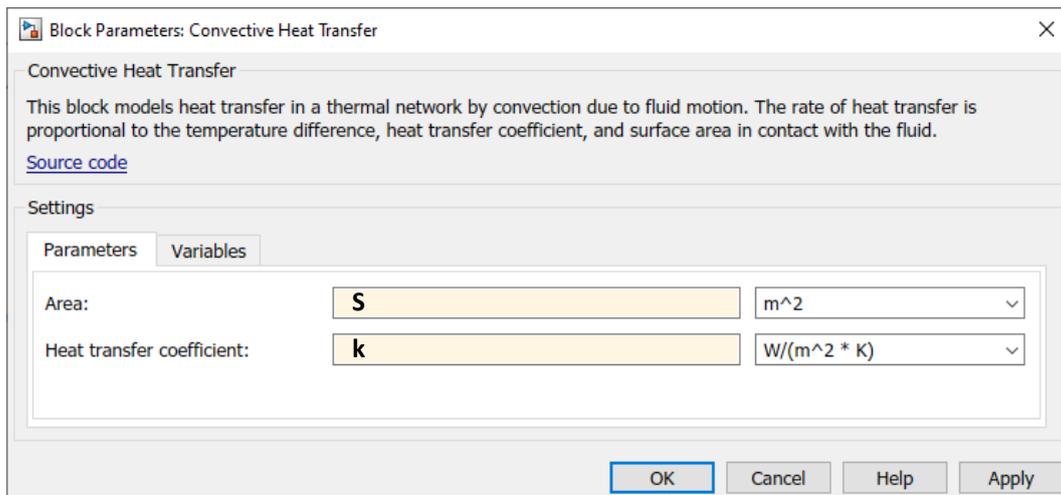


Figure 265 : paramétrage du bloc Convective Heat Transfer

## Paramétrage du bloc Conductive Heat Transfert

Conductive Heat  
Transfert



Simscape/Foundation  
Library/Thermal/Thermal Elements

Ce bloc permet de définir les paramètres de l'échange de chaleur par conduction au travers d'une moitié de la couche de la paroi.

**Area** : Surface correspondant à l'échange thermique. Dans cet exemple il s'agit de la surface de la paroi **S**.

**Thickness** : la longueur de l'éléments traversé par le flux thermique. Dans cet exemple, il s'agit de l'épaisseur de la paroi **e/2**.

**Thermal Conductivity** : il faut renseigner ici la conductivité thermique du matériau constitutif de la paroi  **$\lambda$** .

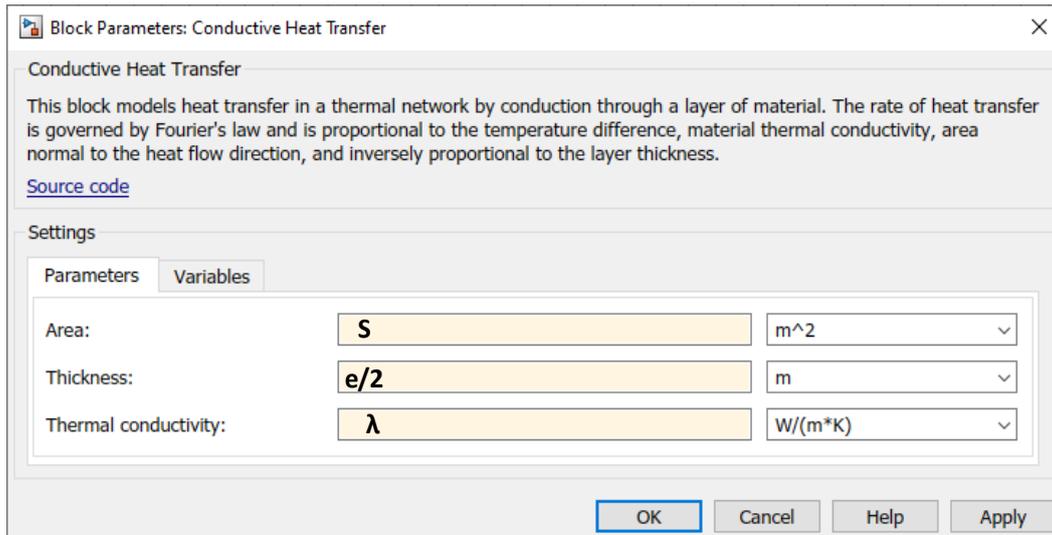


Figure 266 : paramétrage du bloc Conductive Heat Transfer

## Paramétrage du bloc Thermal Mass

Thermal Mass



Simscape/Foundation  
Library/Thermal/Thermal Elements

Ce bloc permet de modéliser la masse thermique de la couche de la paroi.

**Mass** : la masse de l'élément considéré. Dans la masse correspond à la masse de la couche de la paroi : **S.e.ρ**

**Specific Heat** : il faut renseigner ici la capacité thermique massique du matériau considéré **C**.

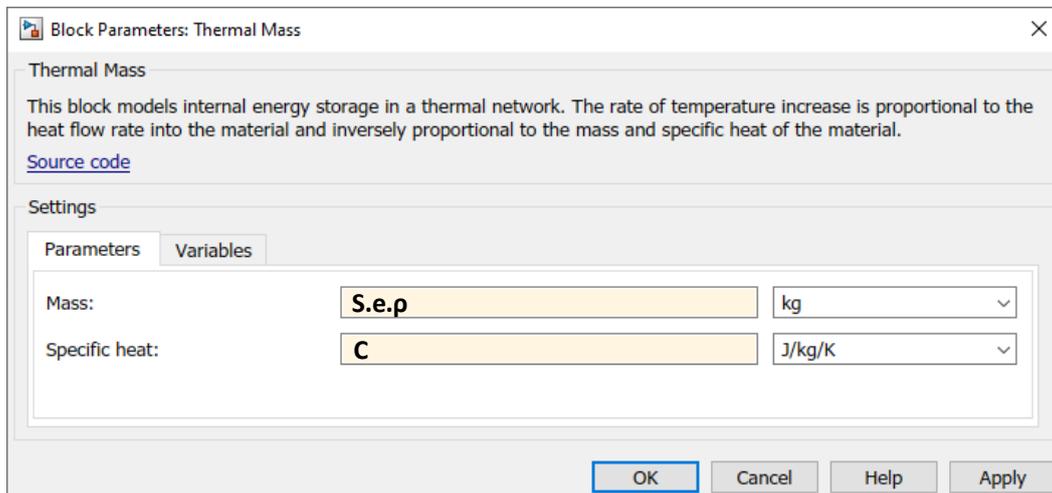


Figure 267 : paramétrage du bloc Thermal Mass

Il est ainsi possible de modéliser les différentes couches de la paroi. Il conviendra de repérer le type d'échange thermique qui intervient au niveau des surfaces (convection ou conduction).

## 19. Utilisation du bloc Thermal Résistance

La Figure 268 montre la correspondance entre les éléments de modélisation de Simscape et les phénomènes physiques intervenants dans les transferts thermiques au travers d'une couche de la paroi.

### Caractéristiques de la couche de la paroi:

- Surface A
- Épaisseur e
- Conductivité thermique  $\lambda$
- Chaleur spécifique C
- Masse volumique  $\rho$

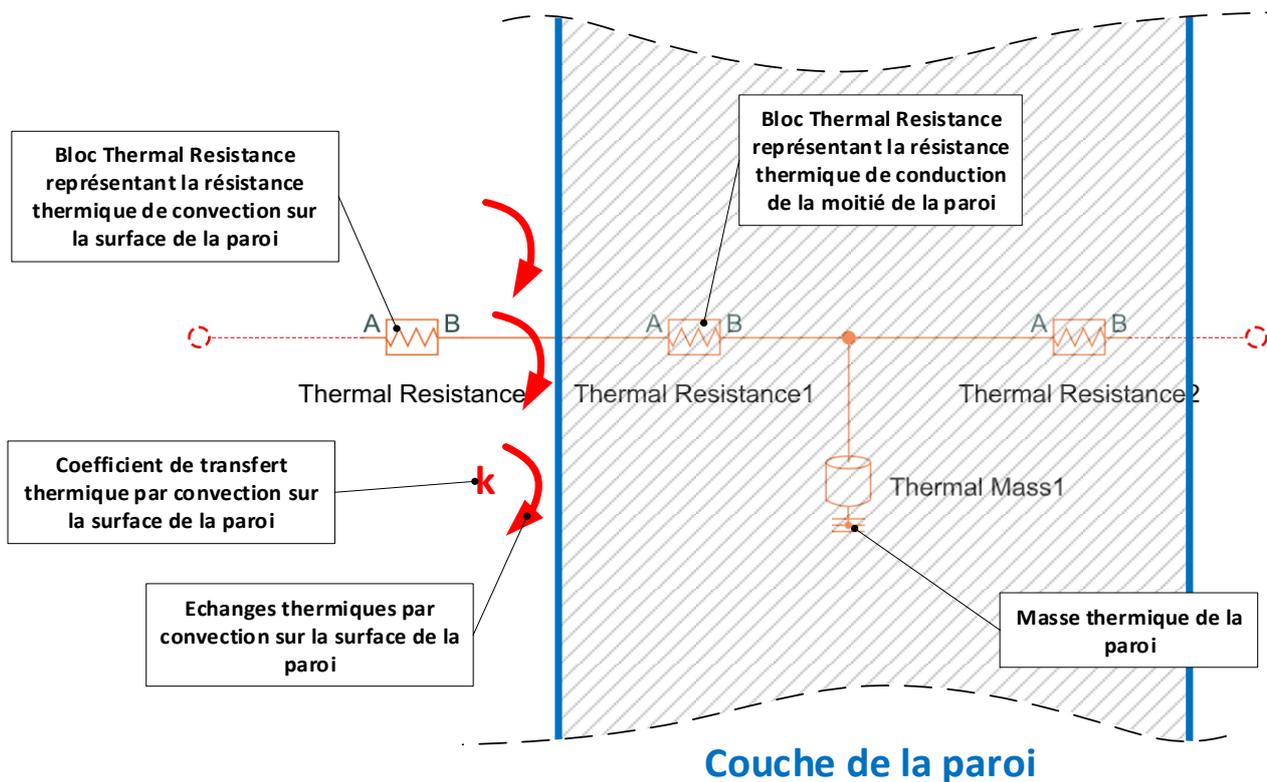


Figure 268 : modélisation Simscape d'une couche de la paroi

Le calcul des résistances thermiques de conduction et de convection équivalentes est détaillé en page 196.

Paramétrage du bloc Thermal Resistance représentant la résistance thermique de convection de la surface de la paroi

### Paramétrage du bloc Thermal Resistance pour la convection

Thermal Resistance  Thermal Resistance Simscape/Foundation  
Library/Thermal/Thermal Elements

La résistance thermique de convection est donnée par  $R_{conv} = \frac{1}{k.S}$  où S représente la surface d'échange

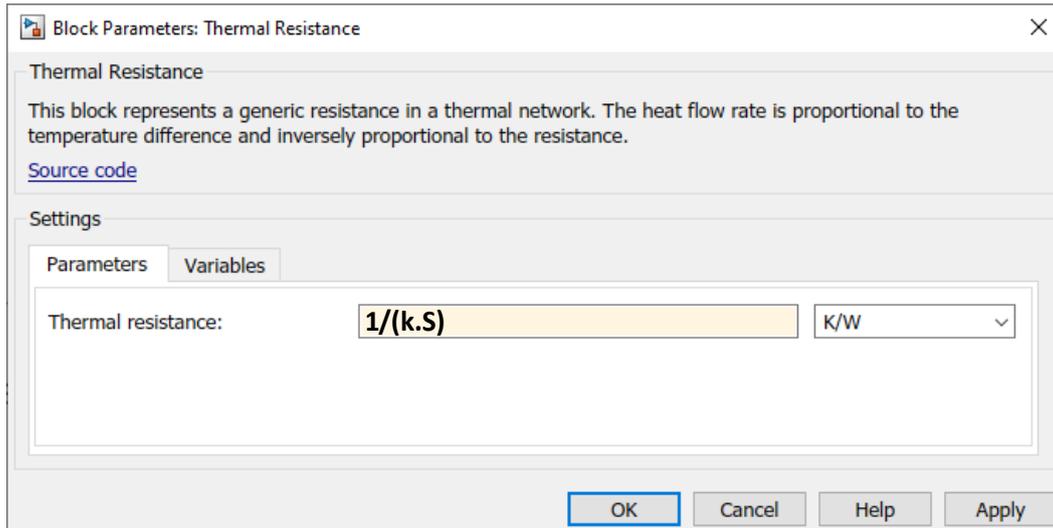


Figure 269 : Paramétrage du bloc Thermal Resistance pour la convection

Paramétrage du bloc Thermal Resistance représentant la résistance thermique de conduction d'une moitié de la couche de la paroi.

### Paramétrage du bloc Thermal Resistance pour la conduction

Thermal Resistance  Thermal Resistance Simscape/Foundation  
Library/Thermal/Thermal Elements

La résistance thermique de conduction de la moitié d'une couche de la paroi est donnée par

$$R_{conv} = \frac{(e/2)}{\lambda.S}$$

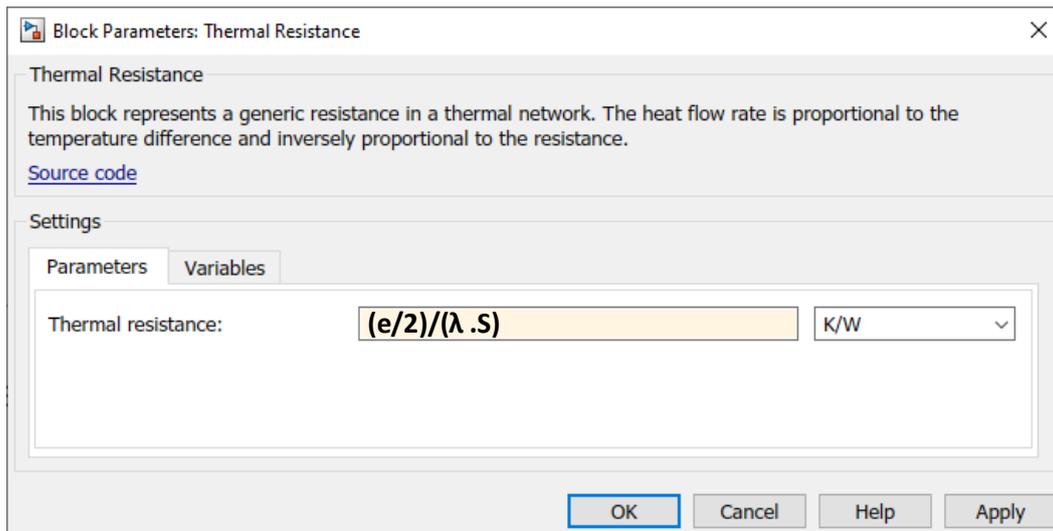


Figure 270 : Paramétrage du bloc Thermal Resistance pour la conduction

Ouvrir le fichier « **couche\_0.slx** ». Ce fichier contient la modélisation d'une couche d'une paroi. Le modèle propose les deux types de modélisation, en utilisant les blocs de transfert thermique par conduction et convection et la modélisation équivalente en utilisant le concept de résistance thermique (Figure 271).

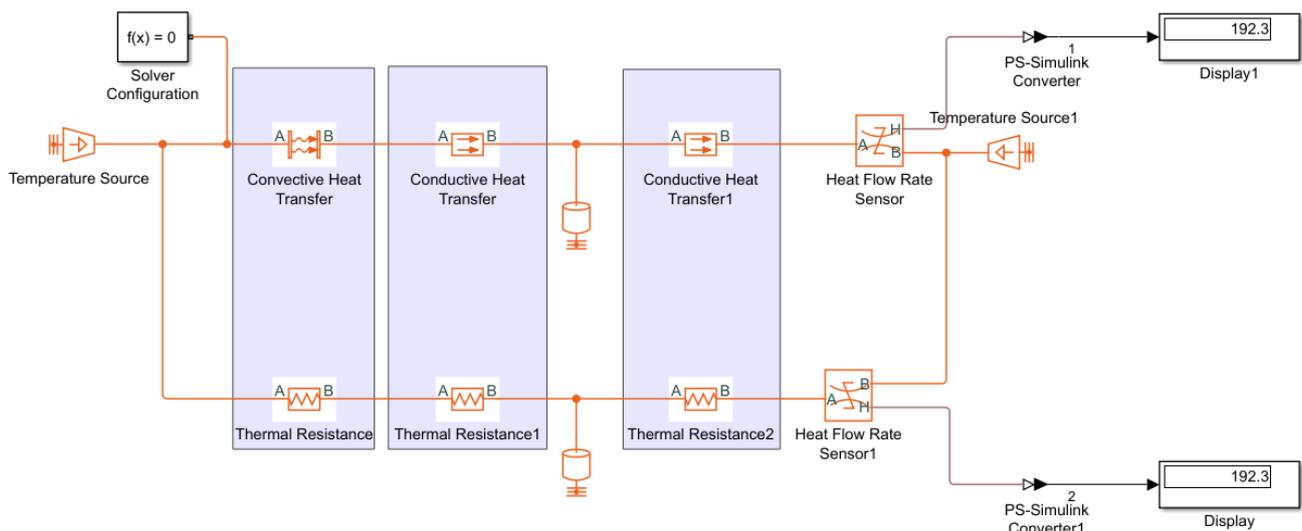


Figure 271 : modélisation d'une couche de paroi en utilisant

**Ouvrir** les différents blocs et observer le paramétrage effectué

**Lancer** la simulation et vérifier l'équivalence des deux modélisations en constatant l'égalité des flux thermiques qui traversent les parois.

## 20. Modélisation avec Simscape d'une paroi multicouche d'un bâtiment

La paroi que nous allons modéliser est constituée de trois couches (Figure 272). Les caractéristiques des différentes couches sont données Figure 273. La paroi est de hauteur  $h = 10$  m et de largeur  $L = 8$  m, sa surface est  $S = 80$  m<sup>2</sup>

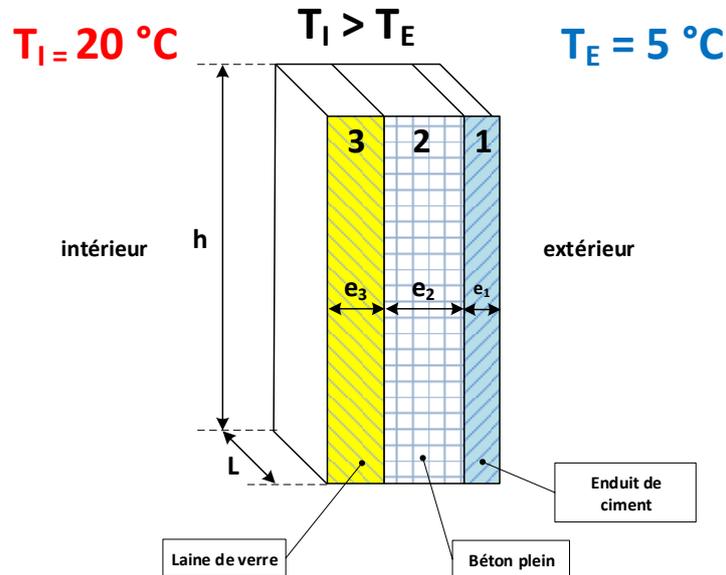


Figure 272 : modélisation d'une paroi d'un bâtiment

	Matériau	Épaisseur en cm	Masse volumique en kg/m <sup>3</sup>	Conductivité thermique en W/(K.m)	Chaleur spécifique en J.kg <sup>-1</sup> .K <sup>-1</sup>
<b>Couche 1</b>	Enduit ciment	$e_1 = 1$	$\rho_1 = 1800$	$\lambda_1 = 1.15$	$c_1 = 864$
<b>Couche 2</b>	Béton	$e_2 = 30$	$\rho_2 = 2500$	$\lambda_2 = 1.75$	$c_2 = 880$
<b>Couche 3</b>	Laine de verre	$e_3 = 10$	$\rho_3 = 20$	$\lambda_3 = 0.032$	$c_3 = 1030$

Figure 273 : caractéristiques physiques des différentes couches de matériau

Les coefficients de transfert thermique par convection sont donnés par le tableau de la Figure 274.

Paroi opaque		
Coefficient de transfert thermique par convection		
	$k_{SI}$ en W/(m <sup>2</sup> .K)	$k_{SE}$ en W/(m <sup>2</sup> .K)
	Intérieur / Paroi	Paroi / Extérieur
Paroi verticale	7.7	25

Figure 274 : coefficients de transfert thermique par convection

L'objectif de la modélisation est de connaître le flux thermique qui s'établit en régime permanent de l'intérieur vers la paroi. Ce flux thermique correspond à ce qui devra être compensé par le chauffage de l'intérieur du bâtiment.

Compte tenu du grand nombre de paramètres qui interviennent dans cette modélisation, nous allons regrouper l'ensemble des valeurs numériques dans un script qui sera associé au modèle. Le script sera

exécuté pour créer les variables dans le Workspace. Ces variables pourront alors être utilisées par le modèle Simscape.

Ouvrir le script « **Parametres\_multi\_couches.m** » et l'exécuter afin de créer les variables correspondant aux paramètres de modélisation. Vérifiez la création des variables dans le Workspace de MATLAB.

```

% Paramètres dimensionnels
h = 6; %hauteur de la paroi en m
L = 10; %largeur de la paroi en m
S = h*L; %surface de la paroi en m2

% Epaisseur des différentes couches de la paroi en cm
e1 = 1; e2 = 30; e3 = 10;

% Caractéristiques des matériaux des différentes couches
% Masse volumique en kg/m3
p1 = 1800; p2 = 2500; p3 = 20;

% Masse des couches de la paroi en kg
m1 = p1*e1*1e-2*S; m2 = p2*e2*1e-2*S; m3 = p3*e3*1e-2*S;

% Conductivité thermique en W/(K.m)
lam_1 = 1.15; lam_2 = 1.75; lam_3 = 0.032;

% Chaleur spécifique en J/kg/K
c1 = 864; c2 = 880; c3 = 1030;

% Coefficients de transfert thermique par convection en W/(m2.K)
ksi = 7.7; kse = 25;

% Température en degrés Celsius
T_int = 20; % température intérieure
T_ext = 5; % Température extérieure

```

Figure 275 : paramètres nécessaires à la construction du modèle de la paroi multicouche

Ouvrir le fichier « **modelisation\_pari\_multicouche\_0.slx** » (Figure 276).

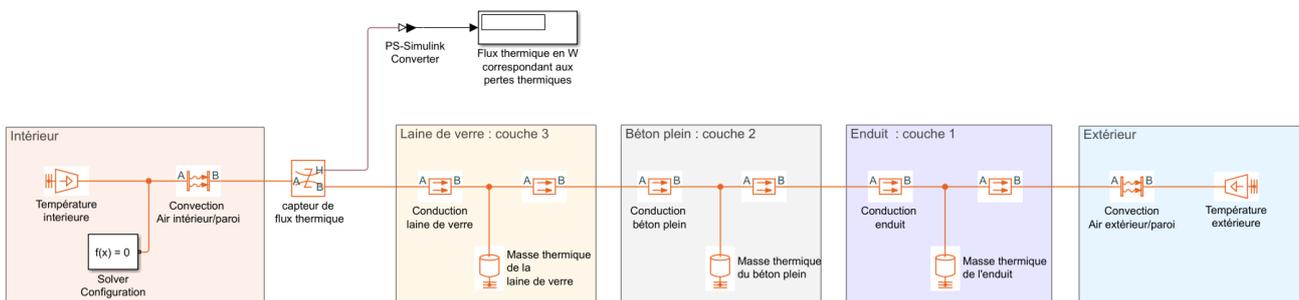


Figure 276 : modélisation d'une paroi multicouche, démarrage de la modélisation

Ce fichier représente la modélisation d'une paroi multicouche. Chaque couche a été modélisée conformément aux explications données au paragraphe 18.

Afin de pouvoir lancer la simulation il faut préalablement renseigner tous les blocs avec les paramètres de modélisation définis dans le script de la Figure 275.

### Paramétrage du bloc : *Température intérieure*

Ce bloc représente la température de l'intérieur de la pièce.

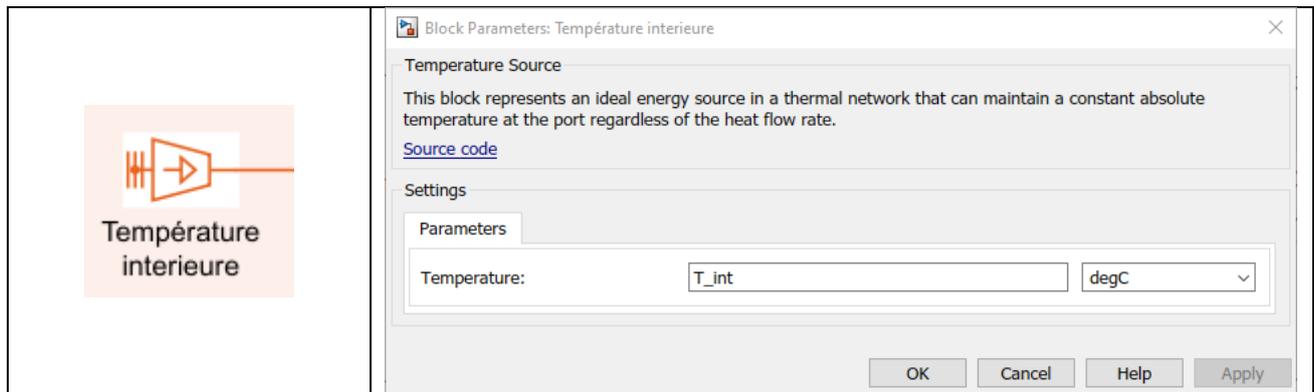


Figure 277 : paramétrage du bloc Température intérieure

### Paramétrage du bloc : *Convection Air intérieur/paroi*

Ce bloc modélise les échanges thermiques par convection entre la surface de la paroi et l'intérieur de la pièce.

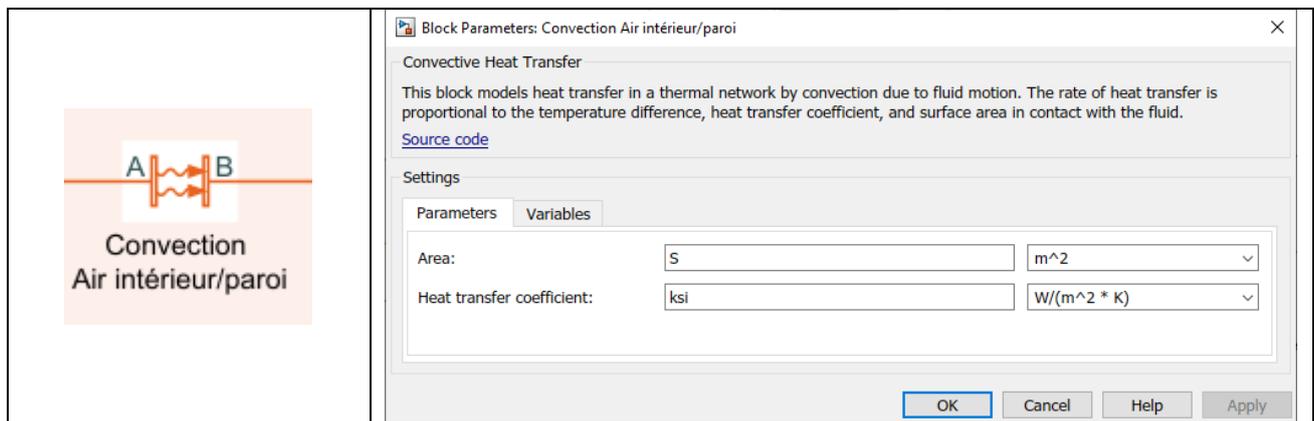


Figure 278 : paramétrage du bloc Convection Air intérieur/paroi

### Paramétrage du bloc : *capteur de flux thermique*

Ce bloc ne nécessite pas de paramétrage. Ce capteur permet de mesurer le flux thermique qui s'établit entre l'intérieur du bâtiment et la paroi.

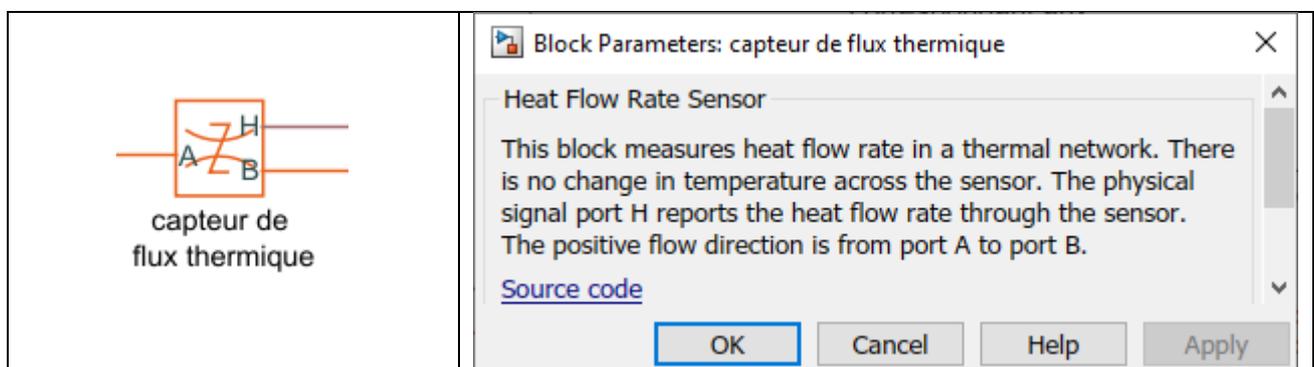


Figure 279 : paramétrage du bloc Capteur de flux thermique

### Paramétrage du bloc : *PS-Simulink Converter*

Il faut préciser dans ce bloc l'unité d'affichage du flux thermique (Watt).

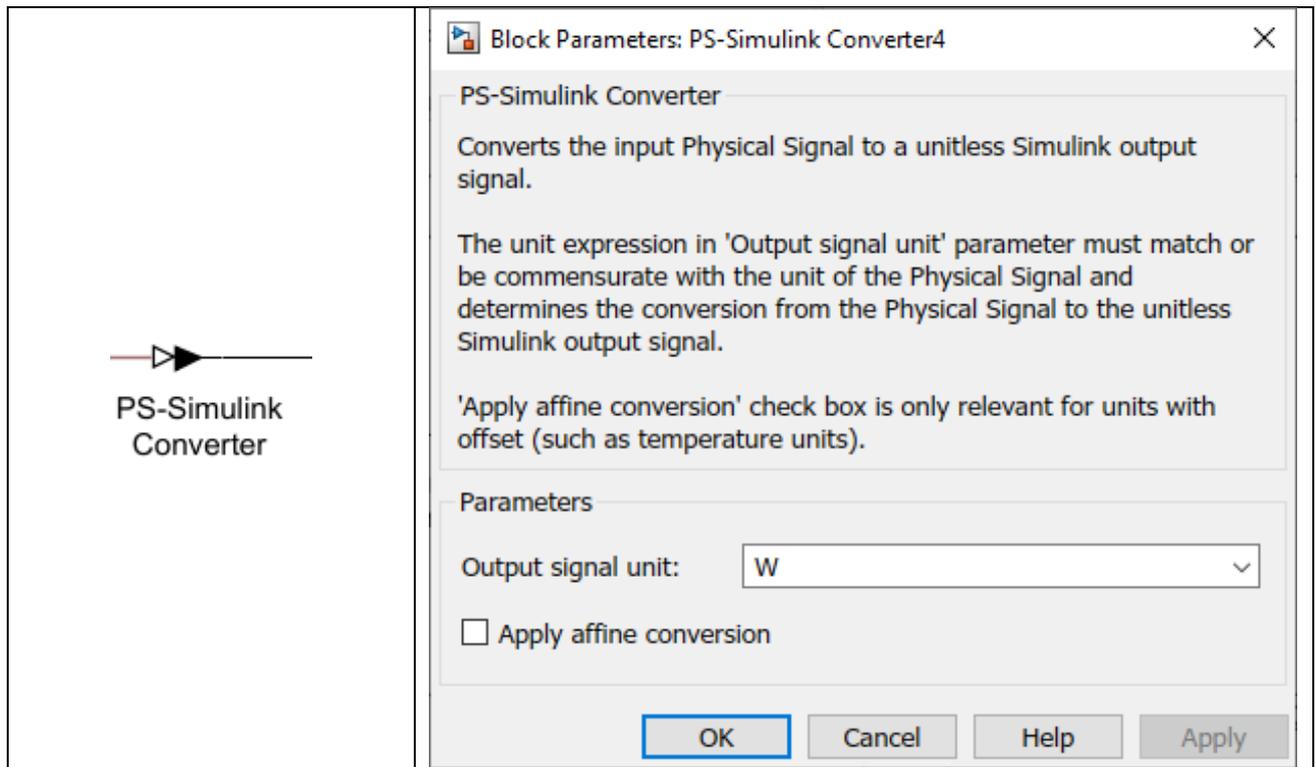


Figure 280 : paramétrage du bloc PS - Simulink Converter

### Paramétrage du bloc : *Conduction laine de verre*

Ces deux blocs modélisent le transfert thermique par conduction dans une demi épaisseur de laine de verre.

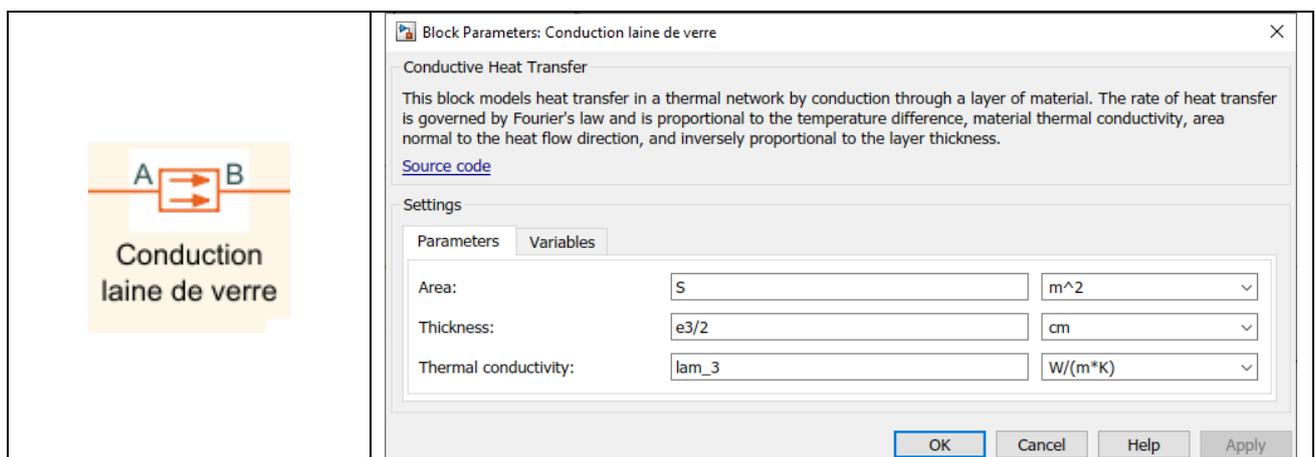


Figure 281 : paramétrage du bloc Conduction laine de verre

### Paramétrage du bloc : *Masse thermique de la laine de verre*

Ce bloc modélise la masse thermique de la laine de verre. Il y a deux onglets à paramétrer. L'onglet **Parameters** qui permet de renseigner la masse de la laine de verre et à sa chaleur spécifique. L'onglet **Variables** qui permet de fixer la température de la masse thermique en début de simulation. Nous considérons ici que la laine de verre est à la même température que l'intérieur de la pièce en début de simulation.

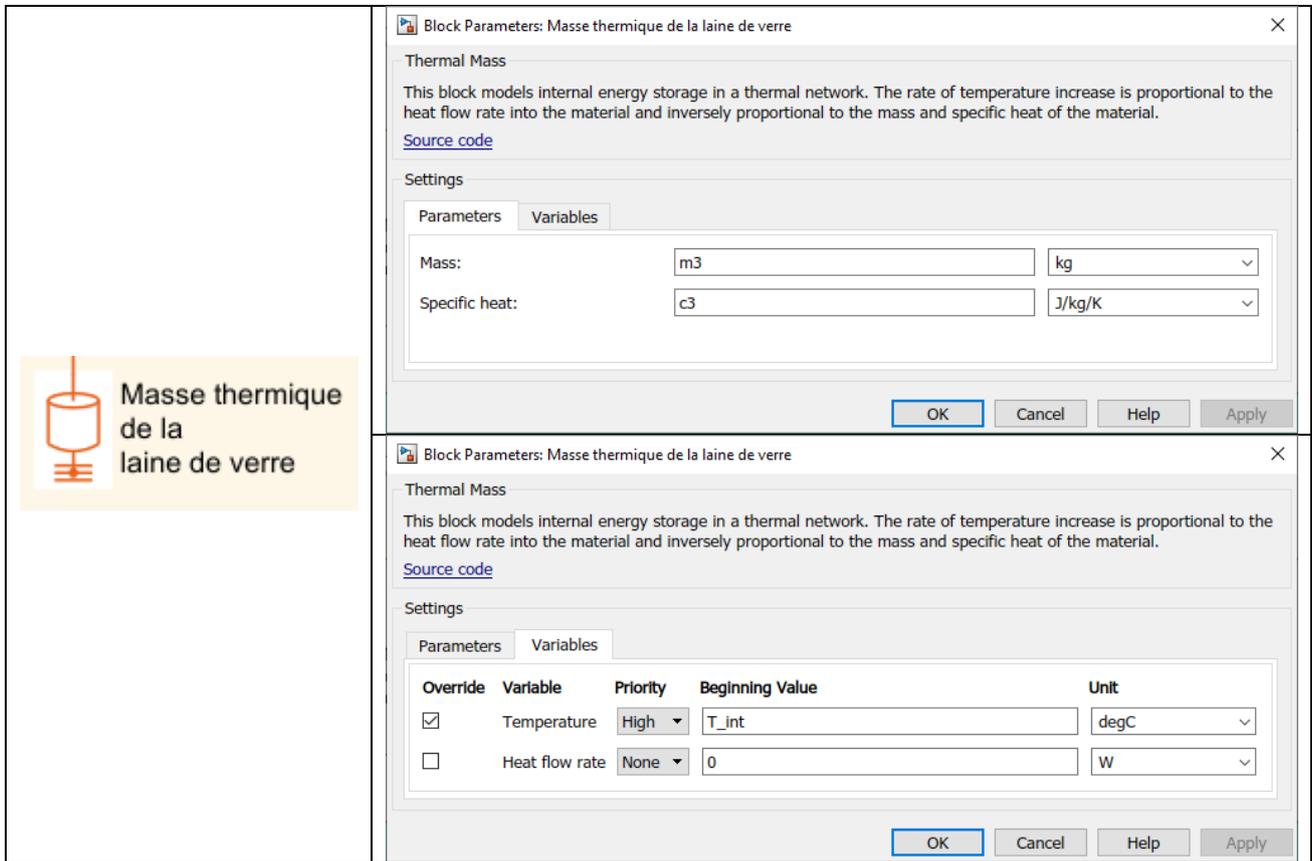


Figure 282 : paramétrage du bloc Masse thermique de la laine de verre

### Paramétrage du bloc : *Conduction Béton plein*

Ces deux blocs modélisent le transfert thermique par conduction dans une demi épaisseur de béton plein.

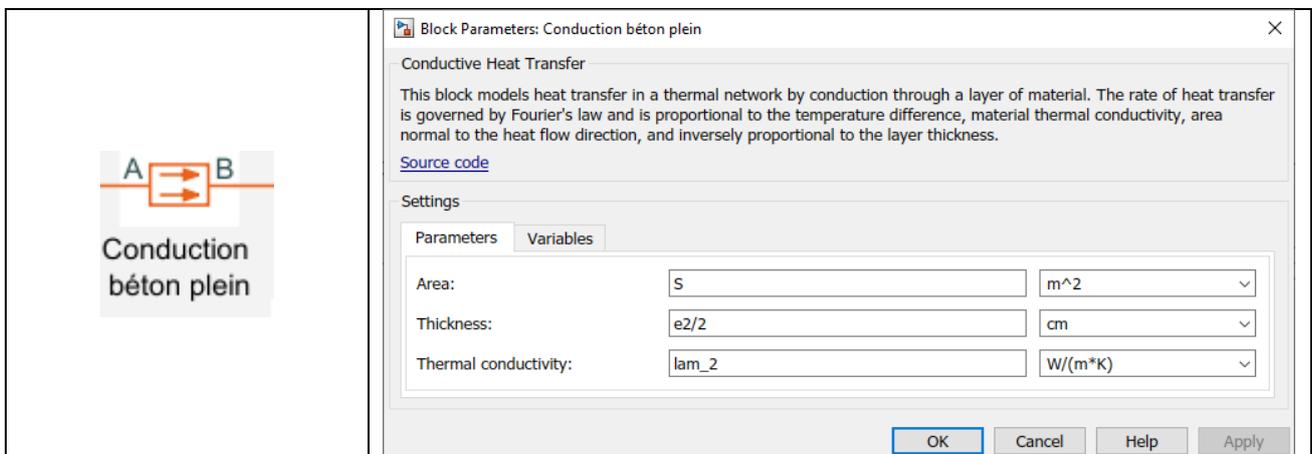


Figure 283 : paramétrage du bloc Conduction béton plein

### Paramétrage du bloc : *Masse thermique de béton plein*

Ce bloc modélise la masse thermique de la couche de béton plein. Il y a deux onglets à paramétrer. L'onglet **Parameters** qui permet de renseigner la masse du béton plein et à sa chaleur spécifique. L'onglet **Variables** qui permet de fixer la température de la masse thermique en début de simulation. Nous considérons ici que le béton plein est à une température  $(T_{int}+T_{ext})/2$  en début de simulation.

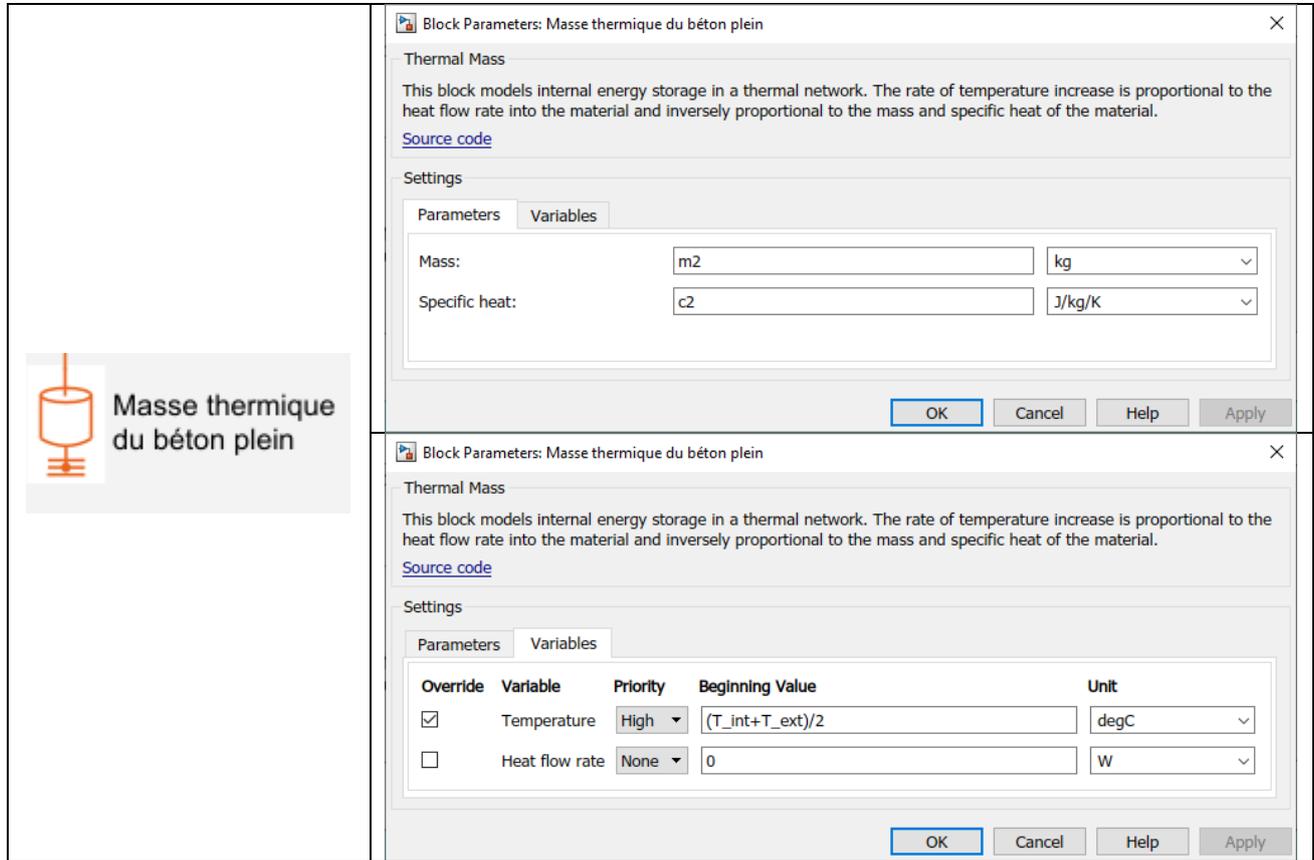


Figure 284 : paramétrage du bloc Masse thermique du béton plein

### Paramétrage du bloc : *Conduction enduit*

Ces deux blocs modélisent le transfert thermique par conduction dans une demi épaisseur d'enduit.

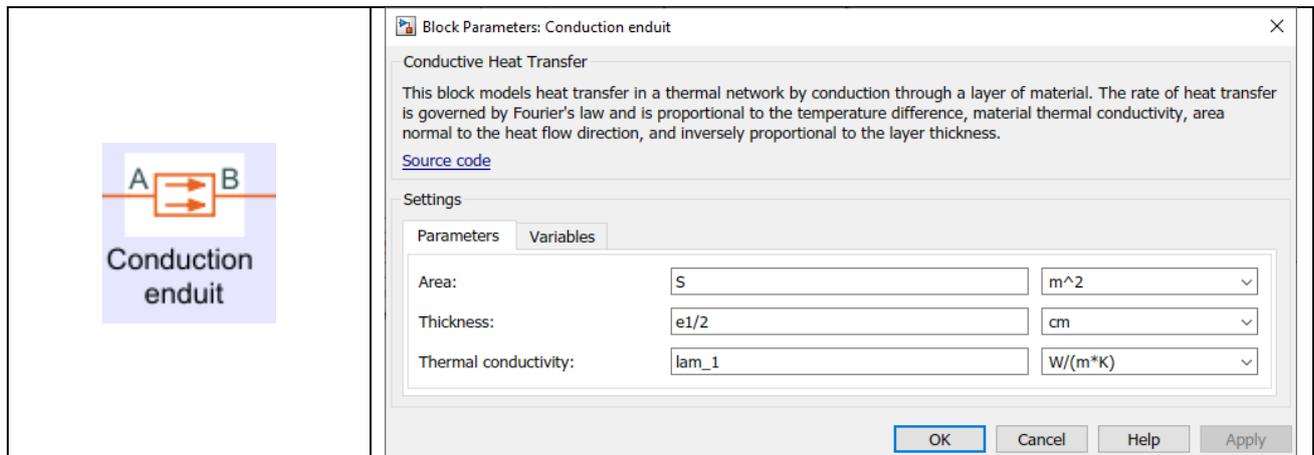


Figure 285 : paramétrage du bloc Conduction enduit

### Paramétrage du bloc : *Masse thermique de l'enduit*

Ce bloc modélise la masse thermique de la couche d'enduit. Il y a deux onglets à paramétrer. L'onglet **Parameters** qui permet de renseigner la masse de l'enduit et à sa chaleur spécifique. L'onglet **Variables** qui permet de fixer la température de la masse thermique en début de simulation. Nous considérons ici que l'enduit est à la température extérieure en début de simulation.

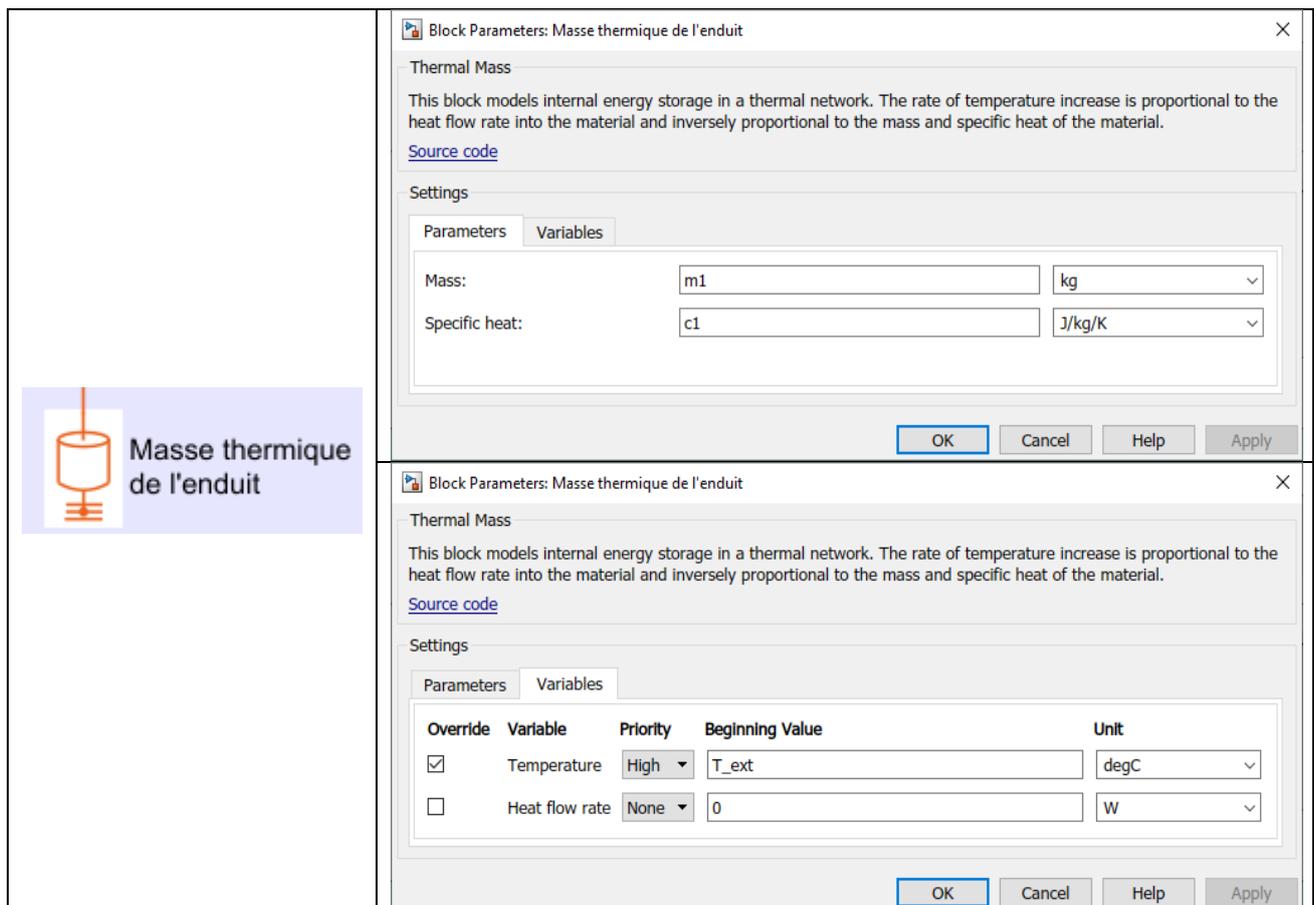


Figure 286 : paramétrage du bloc Masse thermique de l'enduit

### Paramétrage du bloc : *Convection Air extérieur/paroi*

Ce bloc modélise les échanges thermiques par convection entre la surface de la paroi et l'extérieur.

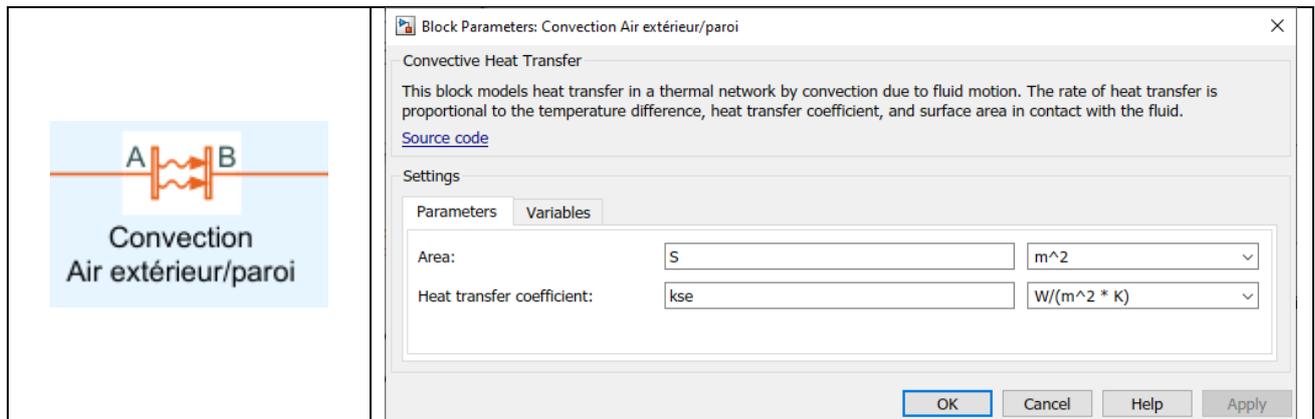


Figure 287 : paramétrage du bloc Convection Air extérieur/paroi

### Paramétrage du bloc : *Température extérieure*

Ce bloc représente la température de l'extérieur.

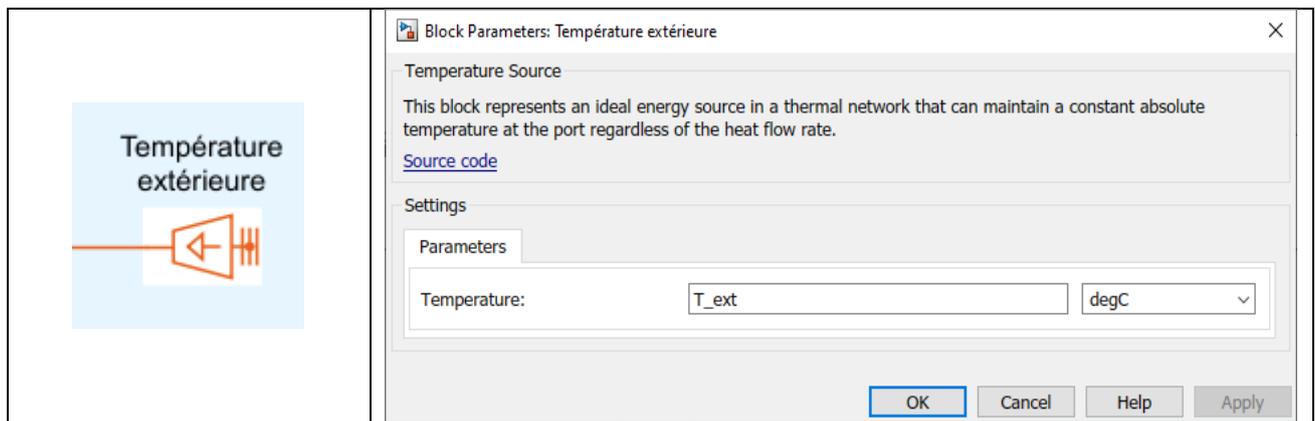


Figure 288 : paramétrage du bloc Température extérieure

Le fichier « **modelisation\_pari\_multi\_couche\_1.slx** » contient le modèle entièrement paramétré.

Pour être certain d'obtenir la valeur en régime permanent du flux thermique, il faut choisir un temps de simulation assez long. Nous allons donc régler le temps de simulation sur 2 jours soit  $48 \times 3600$  s.



Figure 289 : réglage du temps de simulation

Lancer la simulation en cliquant sur  .

Il est possible de visualiser le flux thermique en régime permanent, c'est-à-dire la puissance de chauffage qu'il faudra fournir pour maintenir la température de la pièce intérieure.

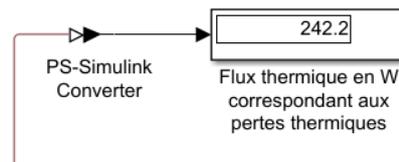


Figure 290 : visualisation du flux thermique en intérieur de paroi en régime permanent

Nous pouvons constater que le flux thermique s'établit à 242.2 W.

Afin d'évaluer l'influence des différentes couches, il est maintenant possible de modifier les paramètres du modèle regroupés dans le script.

Si nous voulons par exemple évaluer l'influence de l'épaisseur de la laine de verre (paramètre  $e_3=10$  cm), nous pouvons modifier le script en imposant  $e_3 = 20$  cm ce qui revient à doubler l'épaisseur de la laine de verre.

**Modifier** la valeur de  $e_3$  dans le script (Figure 291).

```
% Epaisseur des différentes couches de la paroi en cm
e1 = 1; e2 = 30; e3 = 20;
```

Figure 291 : modification d'un paramètre dans le script

**Exécuter** à nouveau le script pour générer la nouvelle variable  $e_3$  dans le Workspace. **Relancer** la simulation du modèle Simscape et relever la nouvelle valeur du flux thermique.

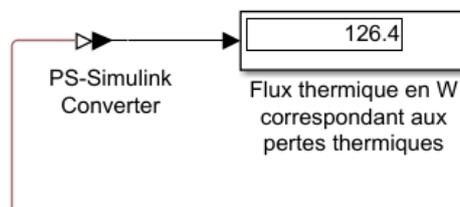


Figure 292 : visualisation de la nouvelle valeur du flux thermique

Nous pouvons constater la diminution du flux thermique dû à l'augmentation de l'épaisseur du matériau isolant. Il est possible en utilisant cette méthode d'évaluer l'influence de tous les paramètres du modèle.

Il est également pu choisir de modéliser tous les échanges thermiques en utilisant le concept de résistance thermique en modifiant le modèle.

**Ouvrir** le modèle « **modelisation\_paroι\_multicouche\_2.slx** » (Figure 293).

Ce fichier contient la modélisation de la paroi multicouche que nous venons de construire et son modèle équivalent qui utilise le concept de résistance thermique.

**Lancer** la simulation en cliquant sur .

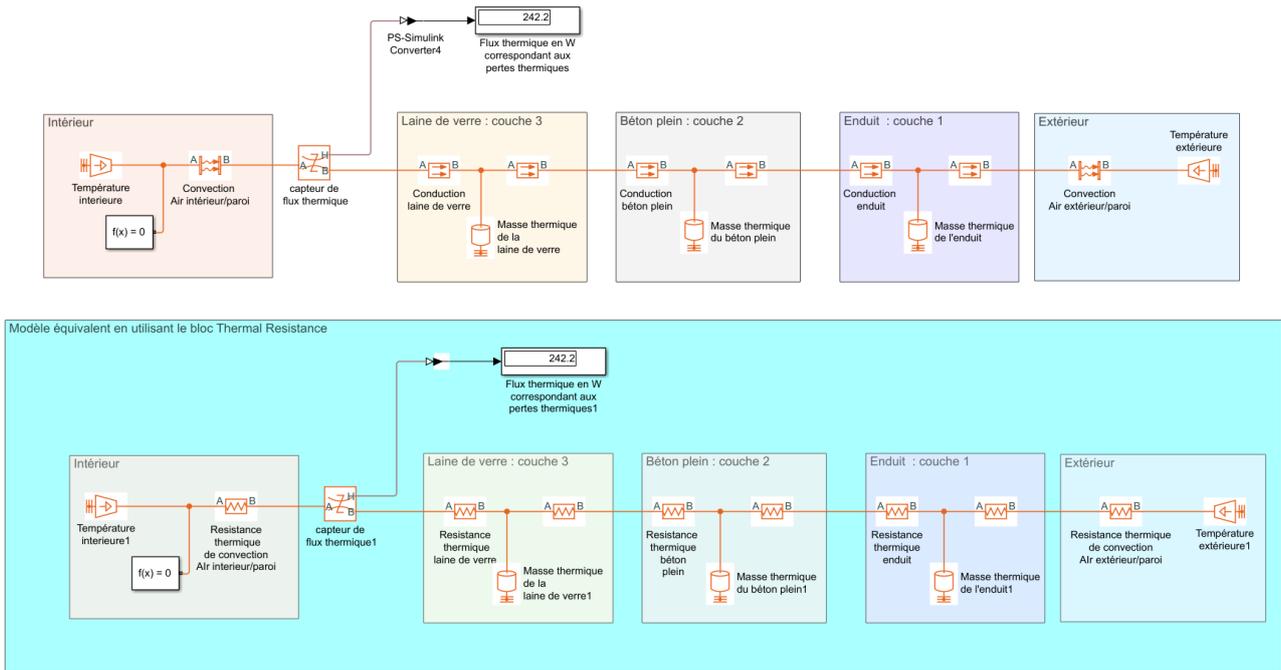


Figure 293 : modélisation équivalente avec des résistance thermiques

Le paramétrage des blocs « résistances thermiques » a été effectué conformément aux explications données paragraphe 19.

Paramétrage du bloc : *Résistance thermique de convection Air intérieur/paroi*

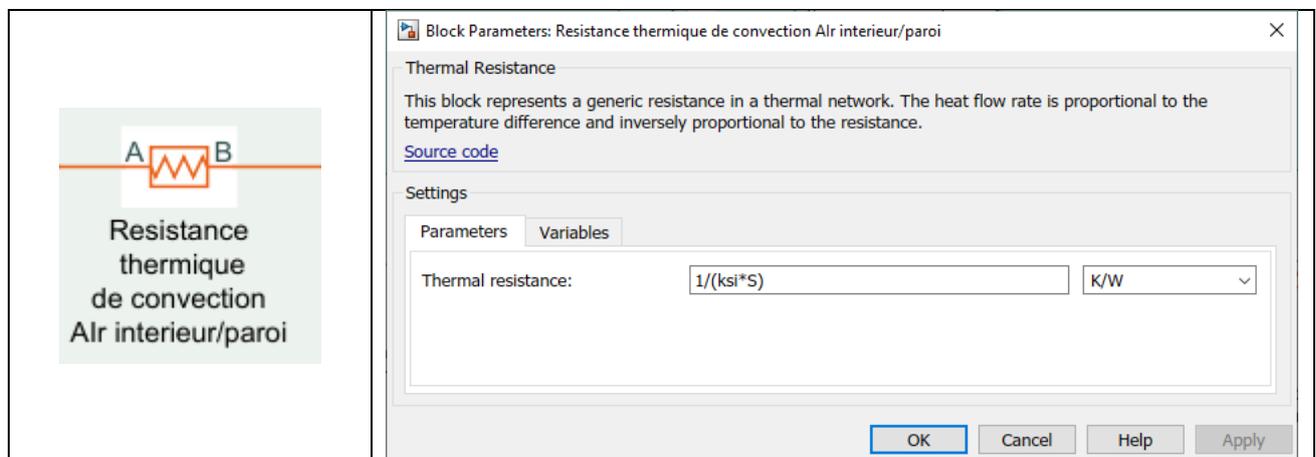


Figure 294 : paramétrage du bloc Résistance thermique de convection Air intérieur/paroi

Paramétrage du bloc : *Résistance thermique de conduction de la laine de verre*

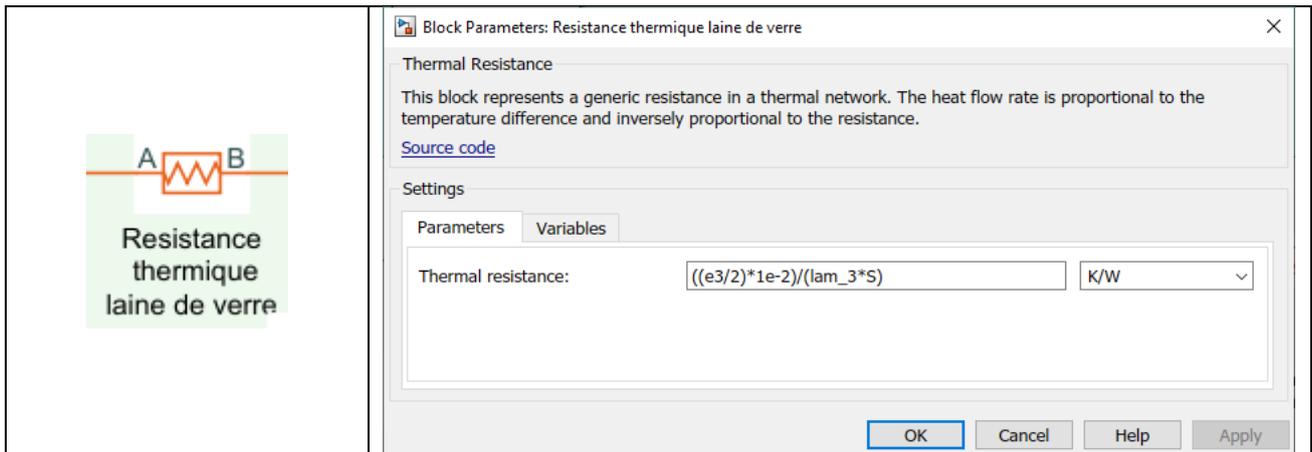


Figure 295 : paramétrage du bloc Résistance thermique de conduction de la laine de verre

Paramétrage du bloc : *Résistance thermique de conduction du béton plein*

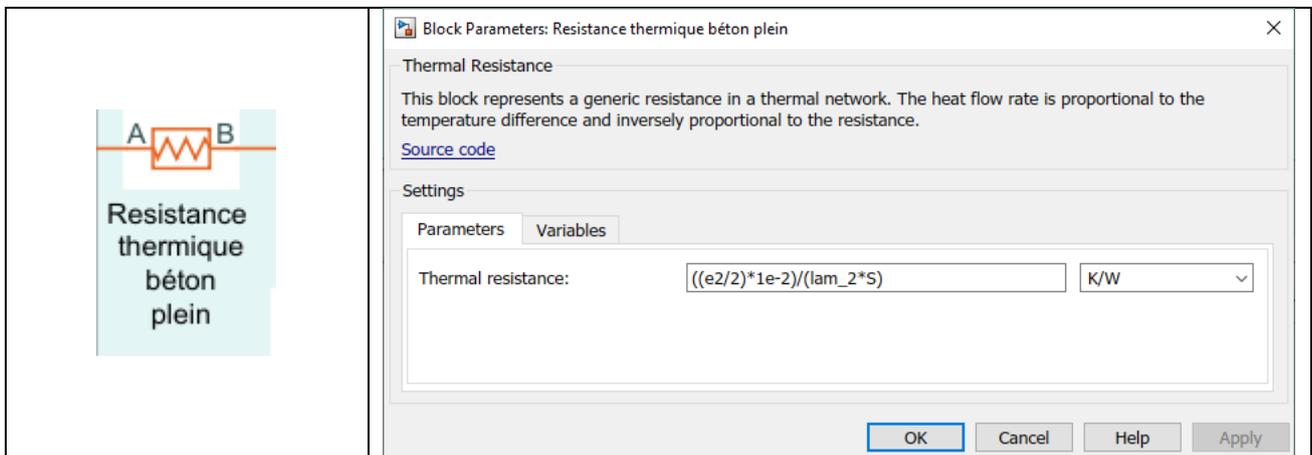


Figure 296 : paramétrage du bloc Résistance thermique de conduction du béton plein

Paramétrage du bloc : *Résistance thermique de conduction de l'enduit*

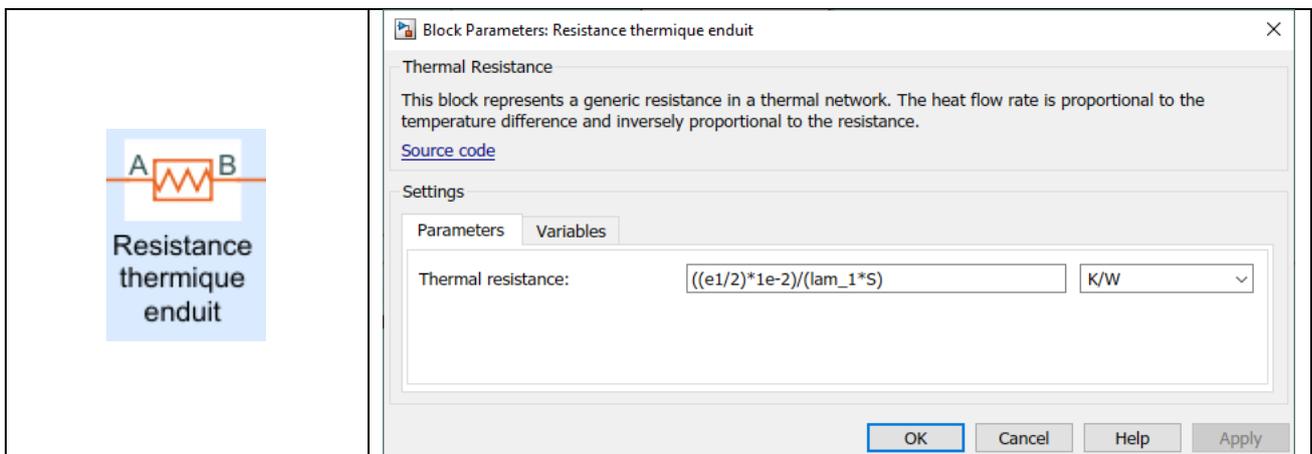


Figure 297 : paramétrage du bloc Résistance thermique de conduction de l'enduit

## Paramétrage du bloc : Résistance thermique de convection Air extérieur/paroi

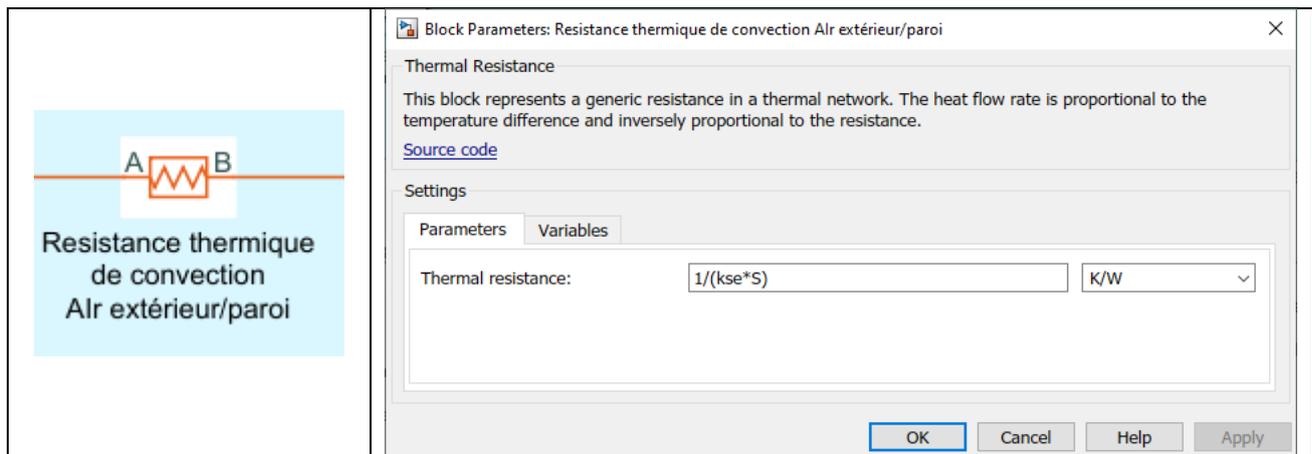


Figure 298 : paramétrage du bloc Résistance thermique de convection Air extérieur/paroi

## 21. Modélisation de l'efficacité d'un double vitrage avec couche d'argon

Nous souhaitons comparer l'efficacité thermique d'une paroi en double vitrage simple (Figure 299) et celle d'une paroi en double vitrage avec une couche d'argon entre les deux vitres (Figure 300).

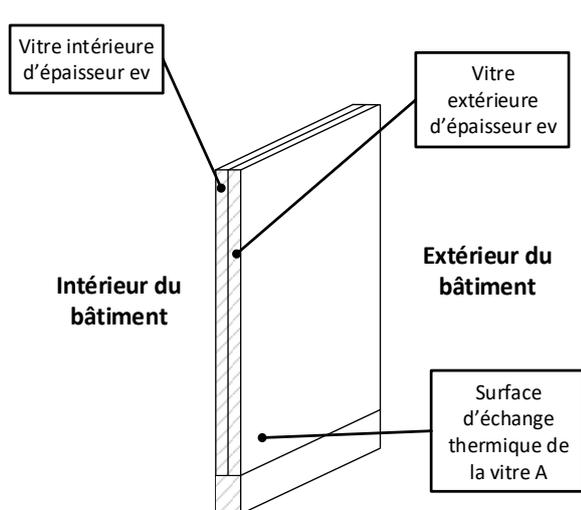


Figure 299 : double vitrage simple

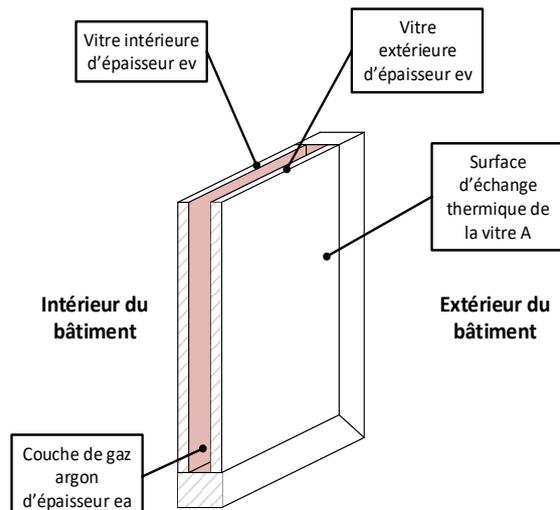


Figure 300 : double vitrage avec couche d'argon

La pièce intérieure à une surface  $S=60\text{m}^2$  et une hauteur de plafond  $h=2.5\text{ m}$  (Figure 301). Elle sera modélisée par une masse thermique constitué de l'air contenu dans la pièce. La vitre aura une surface  $A=2\text{ m}^2$ . Les seuls échanges thermiques pris en compte dans le modèle seront ceux qui auront lieu entre la paroi vitrée et l'intérieur de la pièce. La température de l'intérieur de la pièce est  $T_{\text{int}}=20^\circ\text{C}$  et la température extérieure est  $T_{\text{ext}}=10^\circ\text{C}$ .

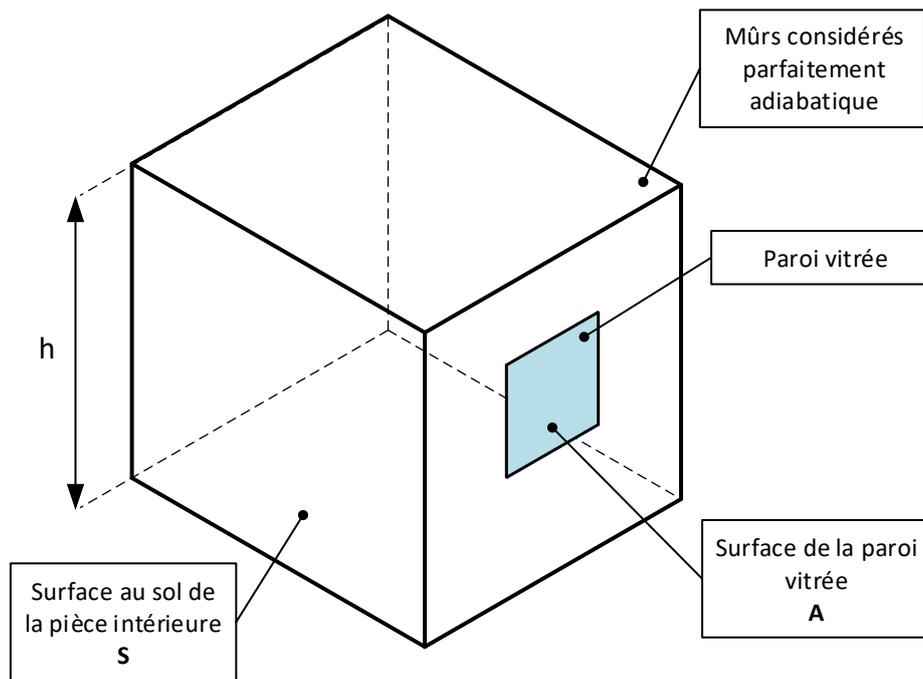


Figure 301 : modèle retenu pour la pièce intérieure et pour la paroi vitrée

Les caractéristiques des matériaux qui interviennent dans la modélisation sont données Figure 302

Matériau	Epaisseur en mm	Masse volumique en $\text{kg/m}^3$	Conductivité thermique en $\text{W/(K.m)}$	Chaleur spécifique en $\text{J.kg}^{-1}.\text{K}^{-1}$
Verre	$e_v = 8$	$\rho_v = 2530$	$\lambda_v = 1$	$c_v = 720$
Argon	$e_a = 10$	$\rho_a = 1.784$	$\lambda_a = 0.018$	$c_a = 320$
Air		$\rho_{\text{air}} = 1.255$	$\lambda_{\text{air}} = 0.025$	$c_{\text{air}} = 1030$

Figure 302 : caractéristiques des matériaux

L'objectif de la modélisation est de connaître l'évolution de la température à l'intérieur de la pièce pendant 24 heures.

Compte tenu du grand nombre de paramètres qui interviennent dans cette modélisation, nous allons regrouper l'ensemble des valeurs numériques dans un script qui sera associé au modèle. Le script sera exécuté pour créer les variables dans le Workspace. Ces variables pourront alors être utilisées par le modèle Simscape.

**Ouvrir** le script « Parametres\_double\_vitrage.m » et l'exécuter afin de créer les variables correspondant aux paramètres de modélisation. Vérifiez la création des variables dans le Workspace de MATLAB.

```

% Paramètres dimensionnels
A = 2 ; %surface de la vitre en m2

% Dimensions de la pièce intérieure en m
S = 60; h = 2.5;

% Volume d'air contenu dans la pièce
Vair = S * h;

% Epaisseur des différentes couches
ev = 8; % épaisseur de la couche de verre en mm
ea = 10; % épaisseur de la couche d'argon en mm

% Caractéristiques des matériaux des différentes couches
% Masse volumique en kg/m3
pv = 2530; % verre
pa = 1.784; % argon
pair = 1.225; % air

% Chaleur spécifique en J/kg/K
cv = 720; ca = 320; cair = 1030;

% Masse des couches de la paroi en kg
Mv = pv * A * ev * 1e-3;
Ma = pa * A * ea * 1e-3;

% Masse de l'air contenu dans la pièce
Mair = pair * Vair;

% Conductivité thermique en W/(K.m)
lam_v = 1; lam_a = 0.018; lam_air = 0.025;

% Coefficients de transfert thermique par convection en W/(m2.K)
ksi = 7.7; % entre intérieur et paroi vitrée
kse = 25; % entre extérieur et paroi vitrée

% Température en degrés Celsius
T_int = 20; % température intérieure
T_ext = 10; % Température extérieure

```

Figure 303 : paramètres nécessaires à la construction du modèle d'un double vitrage

Ouvrir le fichier « double\_vitrage\_0.slx » (Figure 304).

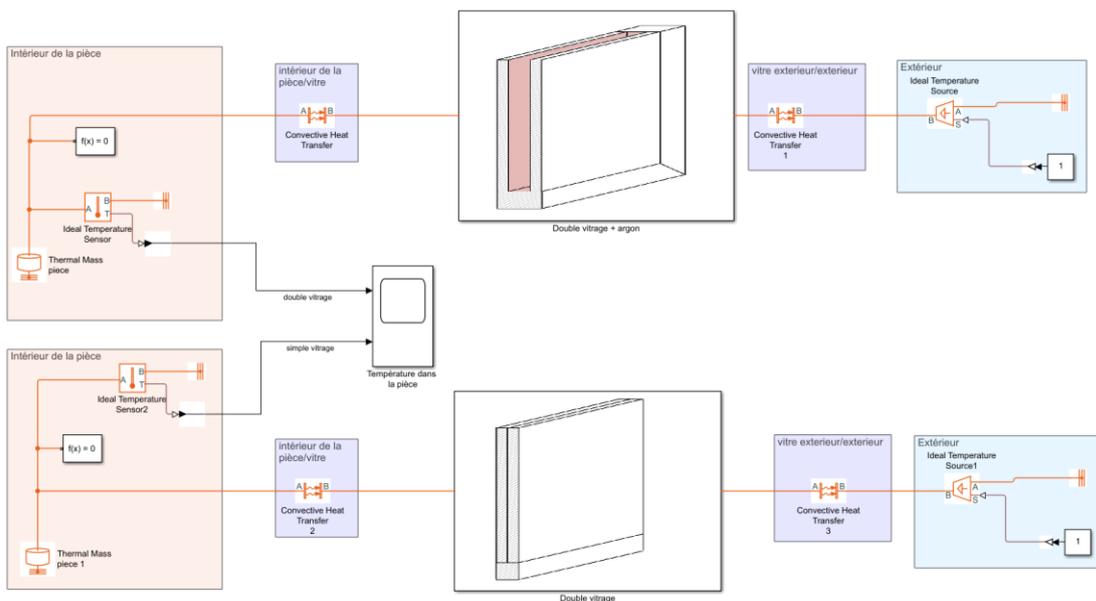


Figure 304 : modélisation d'une paroi multicouche, démarrage de la modélisation

Ce fichier contient la structure de la modélisation de l'efficacité thermique d'un double vitrage avec couche d'argon intermédiaire et d'un double vitrage simple. Le modèle contient 2 sous-systèmes « Double vitrage + argon » (Figure 305) et « Double vitrage » (Figure 306) que l'on peut développer en double cliquant sur chaque bloc.

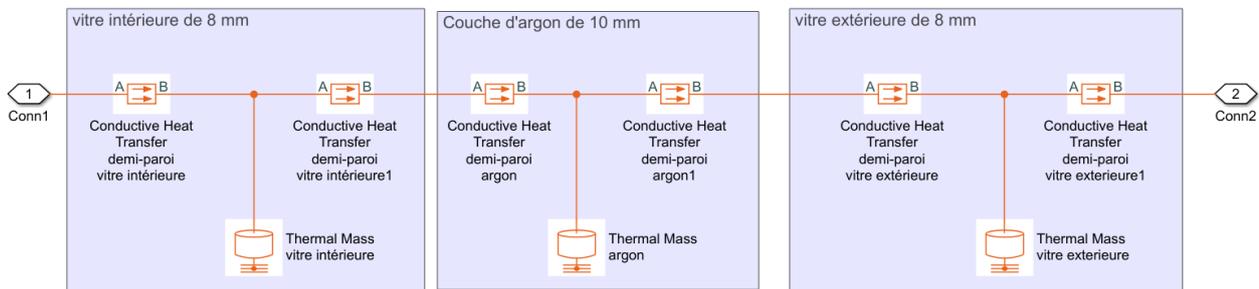


Figure 305 : sous-système « Double vitrage + argon »

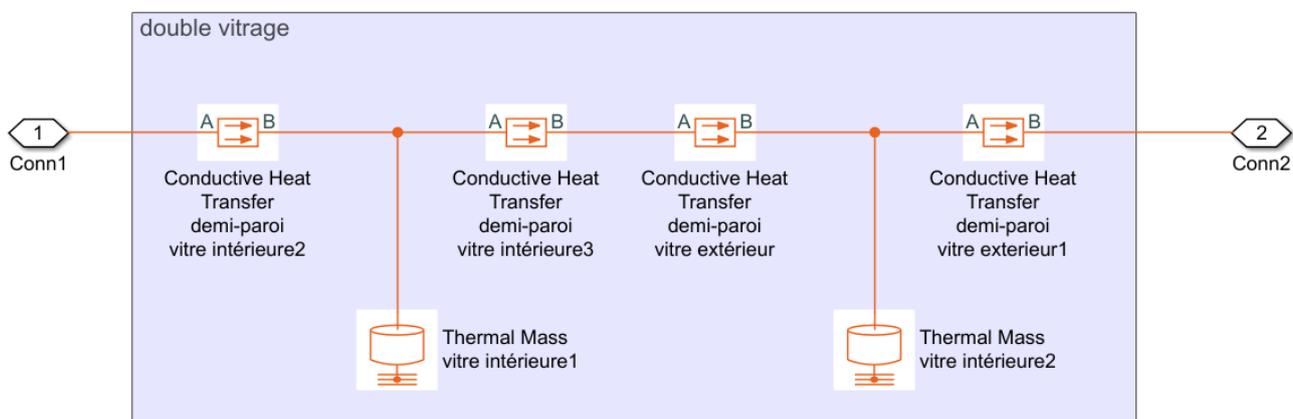
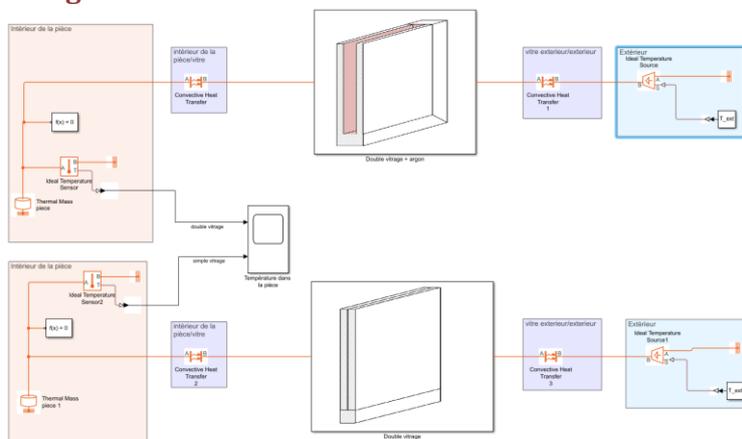


Figure 306 : sous-système « Double vitrage »

Afin de pouvoir lancer la simulation, il faut préalablement renseigner tous les blocs avec les paramètres de modélisation définis dans le script de la Figure 303.

### Paramétrage du modèle global



### Paramétrage du bloc : *Masse thermique de la pièce intérieure*

La pièce est modélisée par la masse thermique de l'air qu'elle contient. Les deux blocs « Thermal Mass pièce » et « Thermal Mass pièce 1 » sont identiques et recevront le même paramétrage.

L'onglet **Parameters** qui permet de renseigner la masse de l'air et à sa chaleur spécifique. L'onglet **Variables** qui permet de fixer la température de la masse thermique en début de simulation. Nous considérons ici que l'intérieur est à la température  $T_{int}$  en début de simulation.

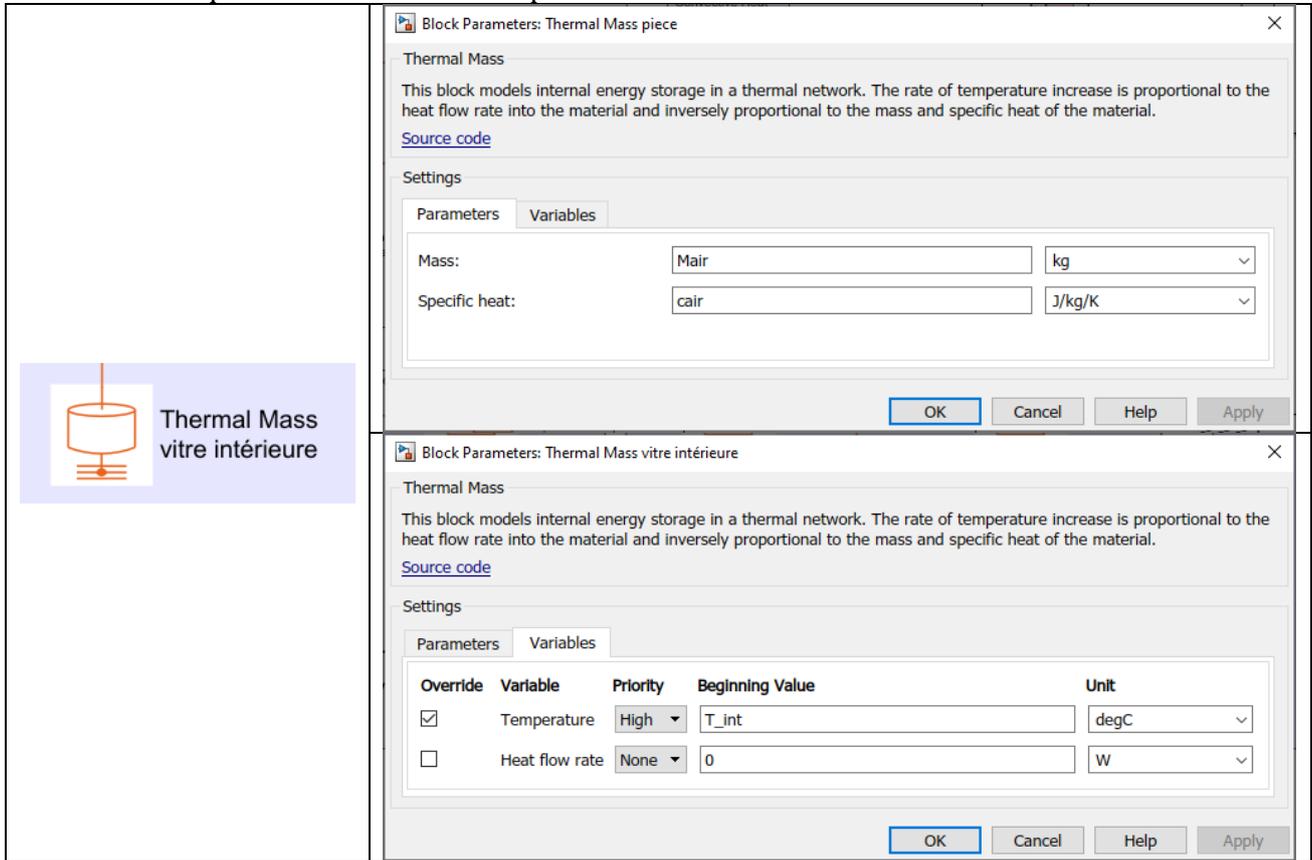


Figure 307 : paramétrage du bloc Masse thermique de l'air de la pièce intérieure

### Paramétrage du bloc : *Convection Air intérieur/paroi vitrée*

Ce bloc modélise les échanges thermiques par convection entre la surface de la paroi vitrée et l'intérieur de la pièce. Les deux blocs « Convective Heat Transfer » et « Convective Heat Transfer2 » sont identiques et recevront le même paramétrage.

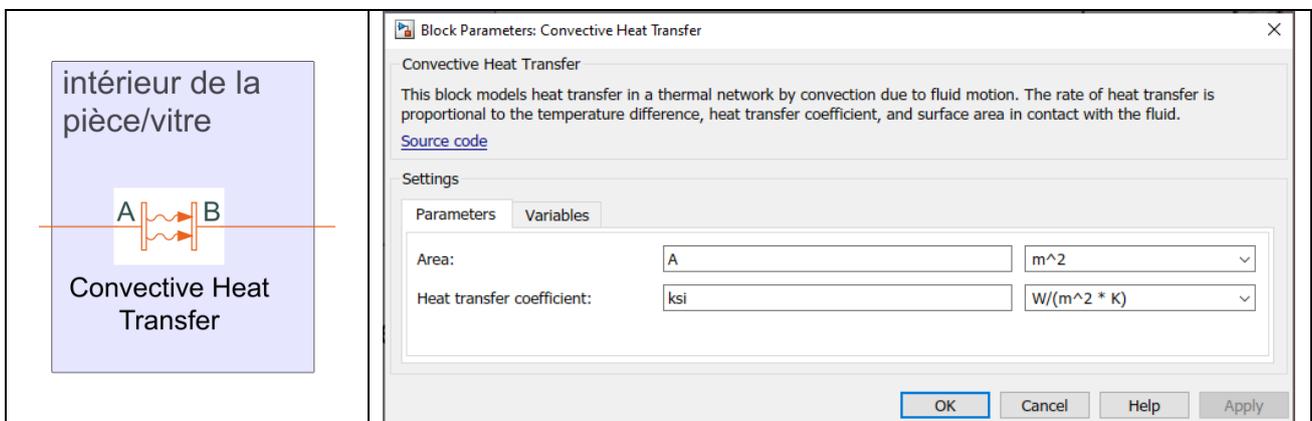


Figure 308 : paramétrage du bloc Convection Air intérieur/paroi

### Paramétrage du bloc : *Convection paroi vitrée/air extérieur*

Ce bloc modélise les échanges thermiques par convection entre la surface de la paroi vitrée et l'extérieur. Les deux blocs « Convective Heat Transfer1 » et « Convective Heat Transfer3 » sont identiques et recevront le même paramétrage.

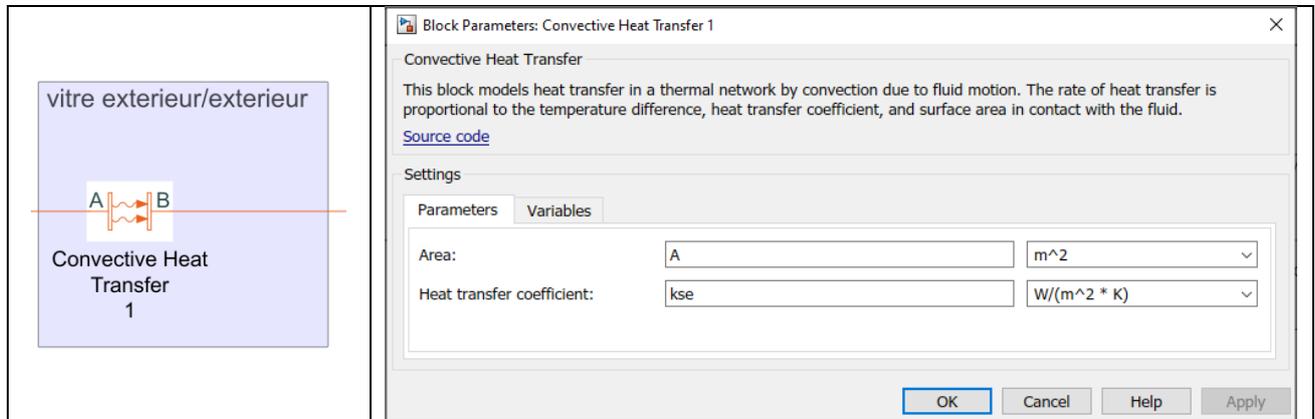


Figure 309 : paramétrage du bloc Convection paroi vitrée/air extérieur

### Paramétrage du bloc : *Température extérieure*

Ce bloc représente la température de l'extérieur. Il est modélisé par un bloc Constant.

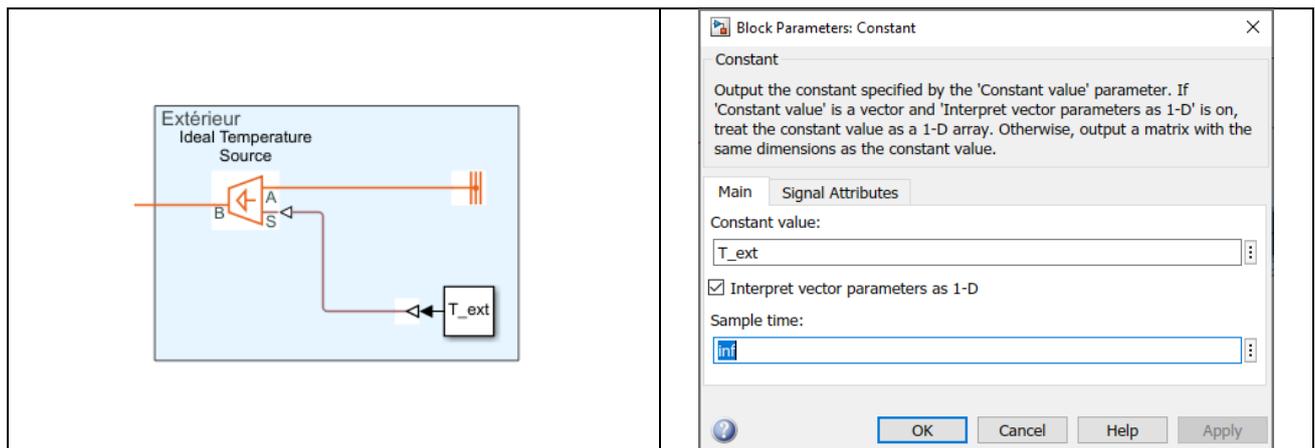


Figure 310 : paramétrage du bloc Température extérieure

### Paramétrage du bloc : *PS-Simulink Converter*

Il faut préciser dans ce bloc l'unité d'affichage de la température est le degré Celsius. Choisir l'unité **degC** et cocher **Apply affine conversion**.

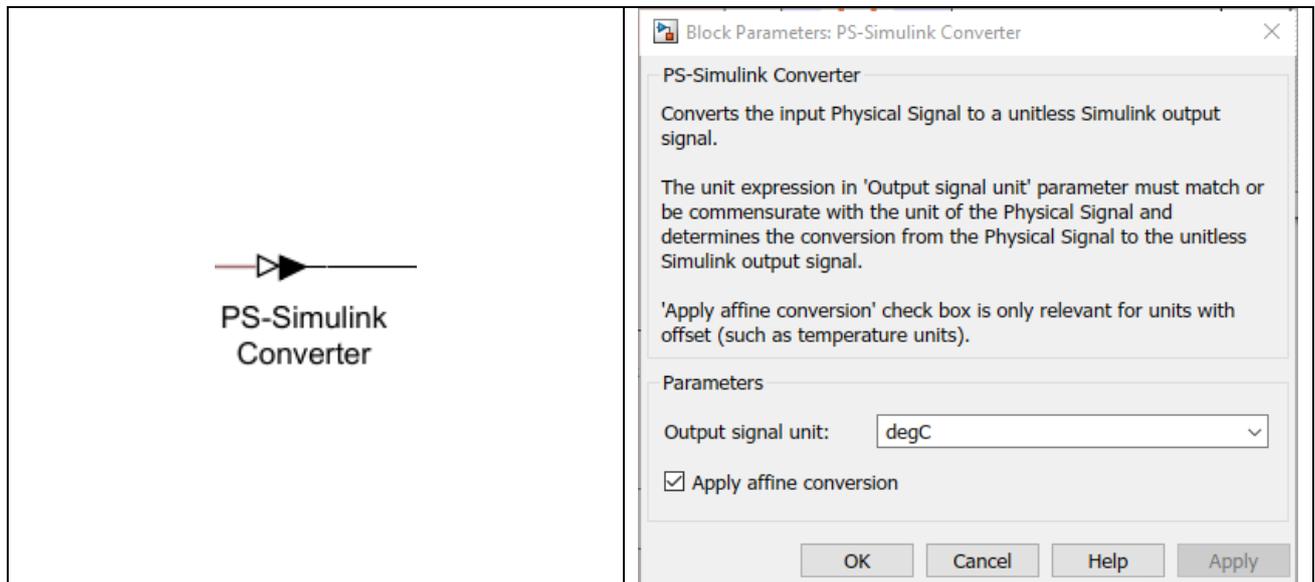
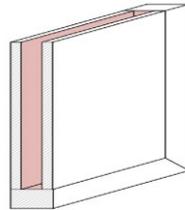


Figure 311 : paramétrage du bloc PS – Simulink Converter

### Paramétrage du sous-système « Double vitrage + argon »



### Paramétrage du bloc : *Conduction du verre de la vitre intérieure*

Ces deux blocs modélisent le transfert thermique par conduction dans une demi épaisseur de verre

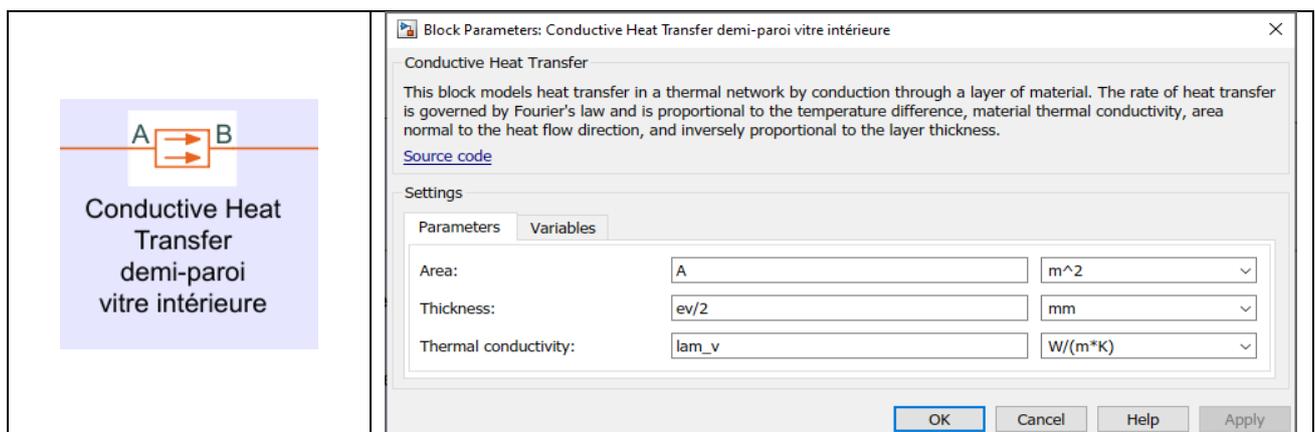


Figure 312 : paramétrage du bloc Conduction de la vitre intérieure

### Paramétrage du bloc : *Masse thermique de la vitre intérieure*

Ce bloc modélise la masse thermique du verre de la vitre intérieure. Il y a deux onglets à paramétrer. L'onglet **Parameters** qui permet de renseigner la masse du verre et à sa chaleur spécifique. L'onglet **Variables** qui permet de fixer la température de la masse thermique en début de simulation. Nous considérons ici que le verre de la vitre intérieure est à la même température que l'intérieur de la pièce en début de simulation.

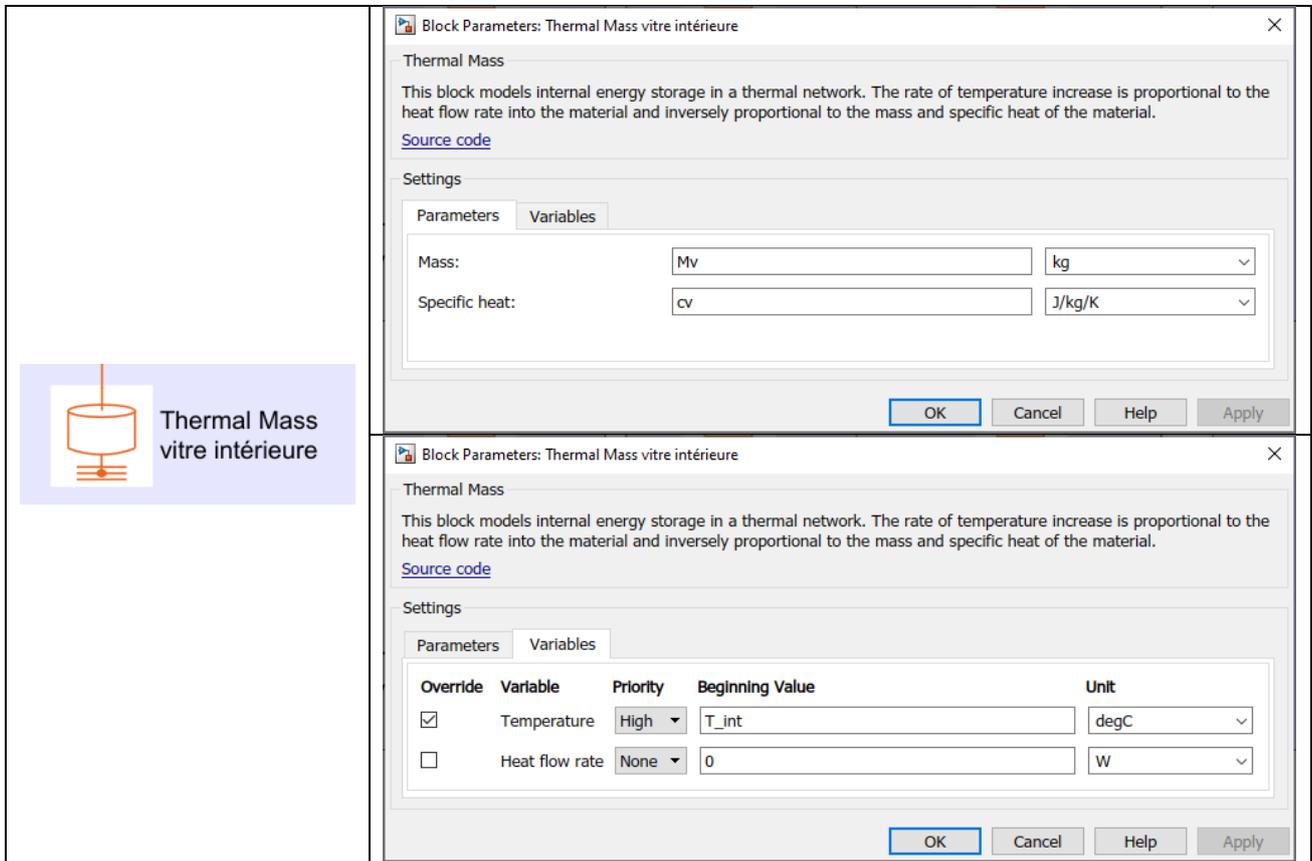


Figure 313 : paramétrage du bloc Masse thermique de la vitre intérieure

### Paramétrage du bloc : *Conduction de la couche d'argon*

Ces deux blocs modélisent le transfert thermique par conduction dans une demi épaisseur d'argon

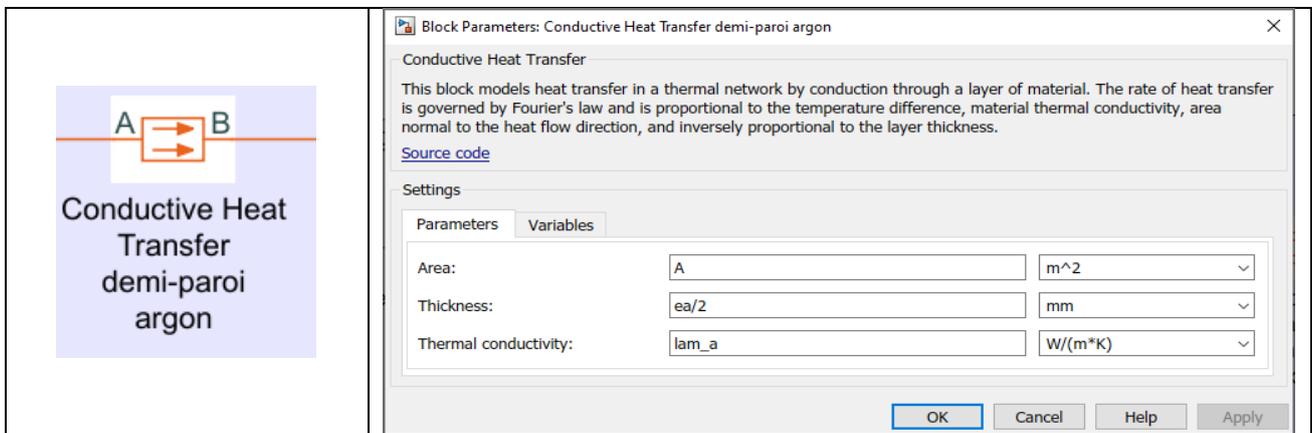


Figure 314 : paramétrage du bloc Conduction de la couche d'argon

### Paramétrage du bloc : *Masse thermique de l'argon*

Ce bloc modélise la masse thermique de l'argon. Il y a deux onglets à paramétrer. L'onglet **Parameters** qui permet de renseigner la masse de l'argon et à sa chaleur spécifique. L'onglet **Variables** qui permet de fixer la température de la masse thermique en début de simulation. Nous considérons ici que l'argon est à la température  $(T_{int}+T_{ext})/2$  en début de simulation.

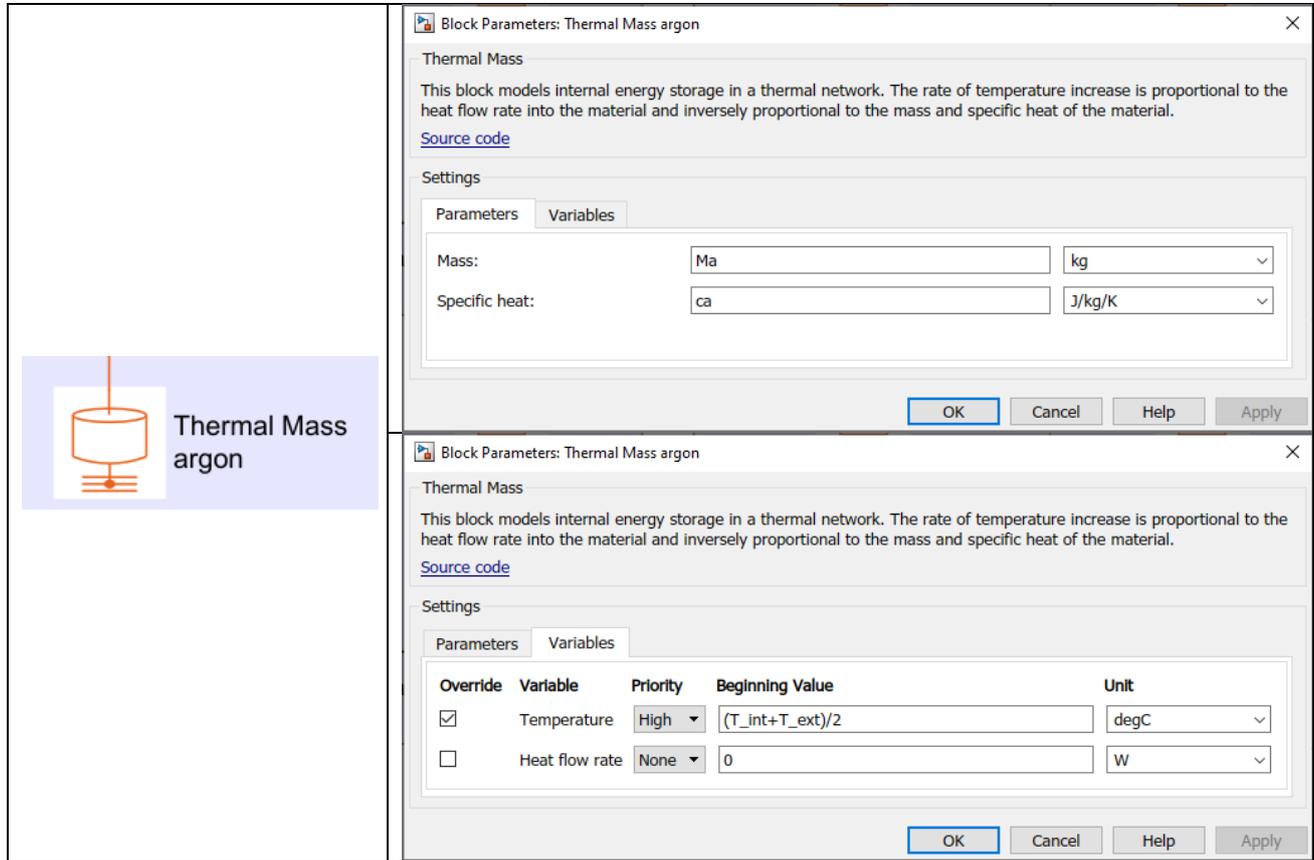


Figure 315 : paramétrage du bloc Masse thermique de l'argon

### Paramétrage du bloc : *Conduction du verre de la vitre extérieure*

Ces deux blocs modélisent le transfert thermique par conduction dans une demi épaisseur de verre de la vitre extérieure.

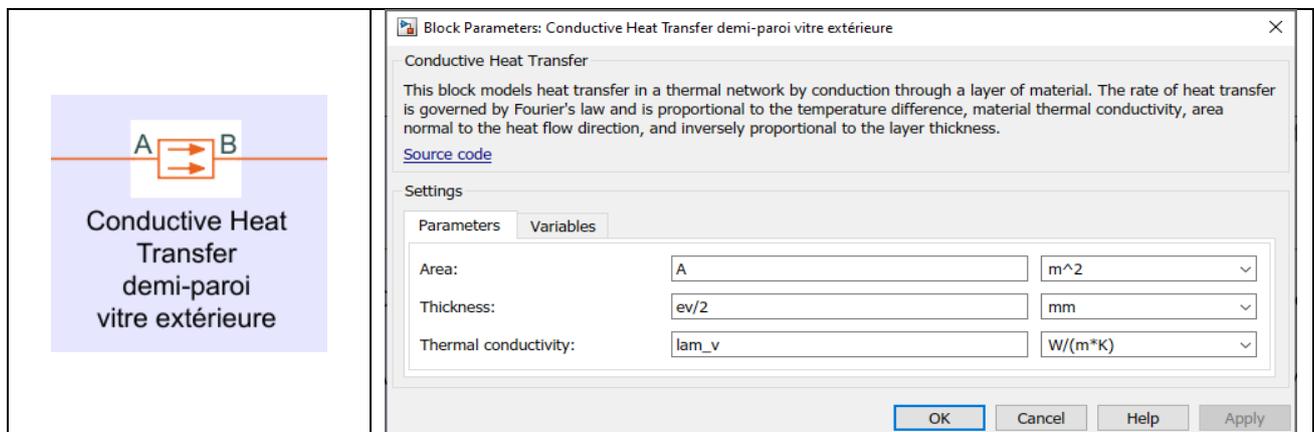


Figure 316 : paramétrage du bloc Conduction de la vitre extérieure

### Paramétrage du bloc : *Masse thermique de la vitre extérieure*

Ce bloc modélise la masse thermique du verre de la vitre extérieure. Il y a deux onglets à paramétrer. L'onglet **Parameters** qui permet de renseigner la masse du verre et à sa chaleur spécifique. L'onglet **Variables** qui permet de fixer la température de la masse thermique en début de simulation. Nous considérons ici que le verre de la vitre intérieure est à la même température que l'extérieur en début de simulation.

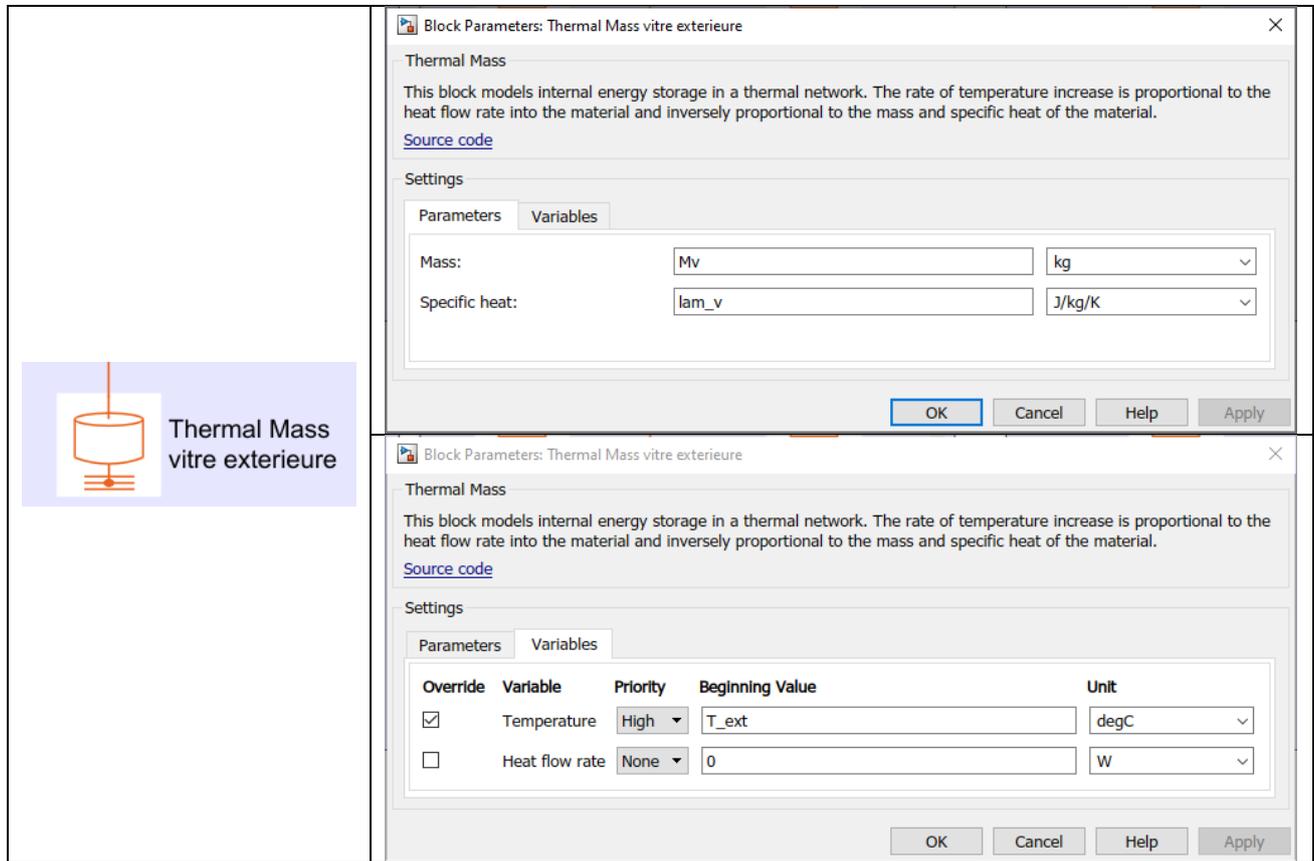
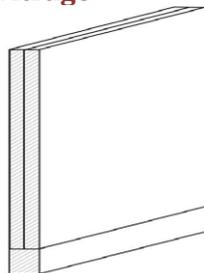


Figure 317 : paramétrage du bloc Masse thermique de la vitre extérieure

### Paramétrage du sous-système « Double vitrage »



*Paramétrage du bloc : Conduction du verre de la vitre intérieure/extérieure*

Ces quatre blocs modélisent le transfert thermique par conduction dans une demi épaisseur de verre.

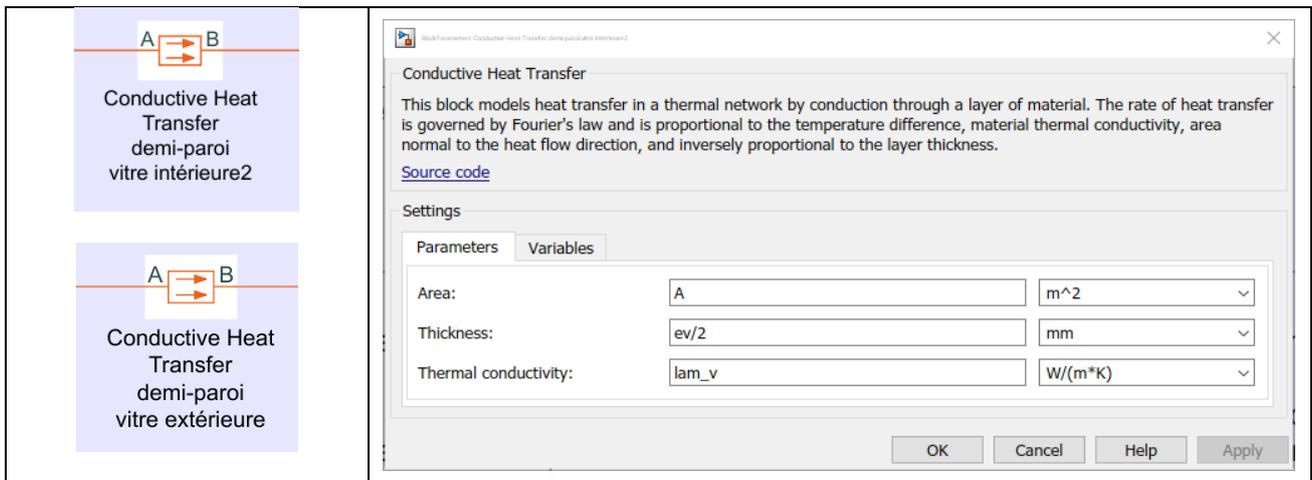


Figure 318 : paramétrage du bloc Conduction de la vitre intérieure

*Paramétrage du bloc : Masse thermique de la vitre intérieure*

Ce bloc modélise la masse thermique du verre de la vitre intérieure. Il y a deux onglets à paramétrer. L'onglet **Parameters** qui permet de renseigner la masse du verre et à sa chaleur spécifique. L'onglet **Variables** qui permet de fixer la température de la masse thermique en début de simulation. Nous considérons ici que le verre de la vitre intérieure est à la même température que l'intérieur de la pièce en début de simulation.

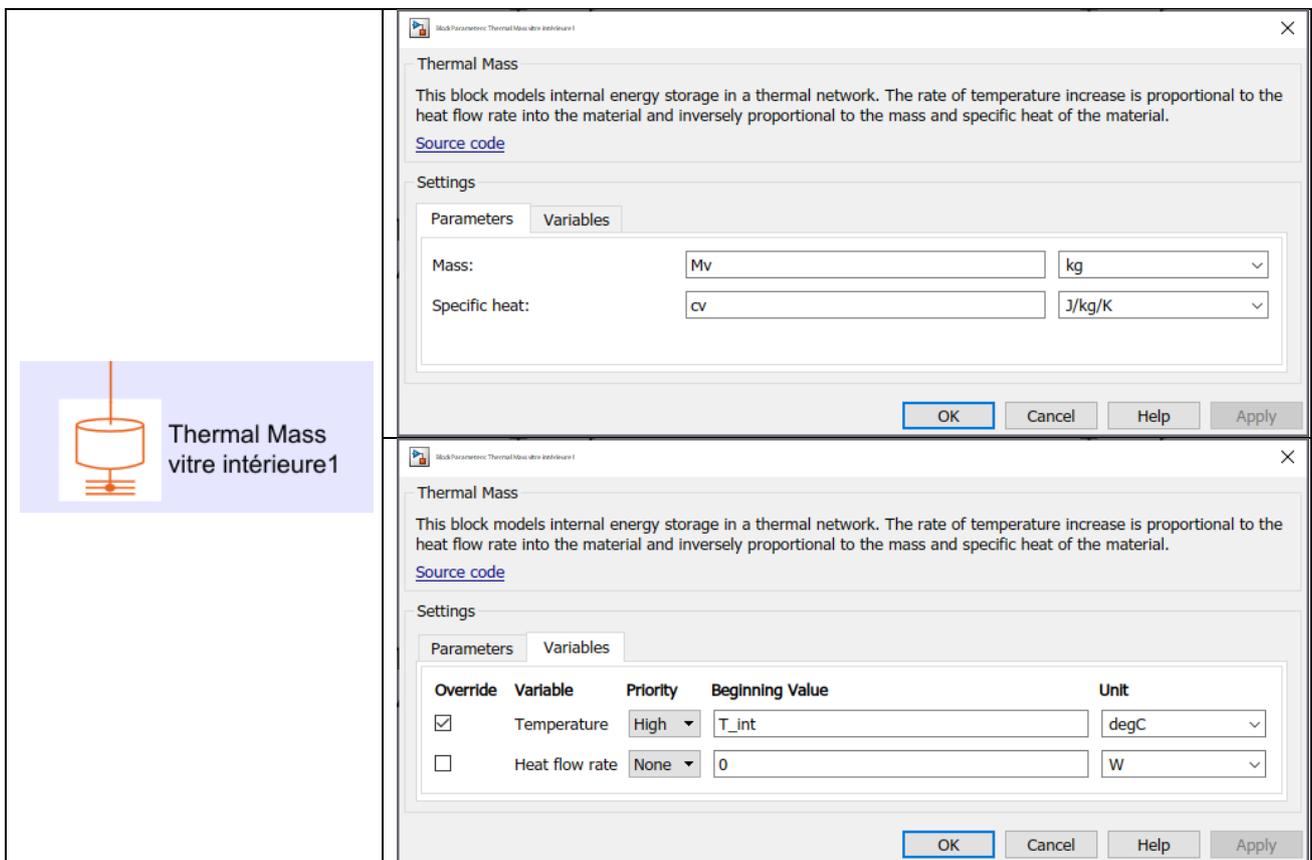
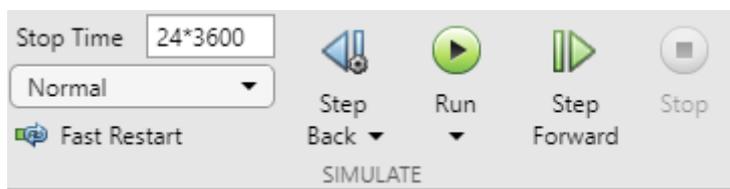


Figure 319 : paramétrage du bloc Masse thermique de la vitre intérieure

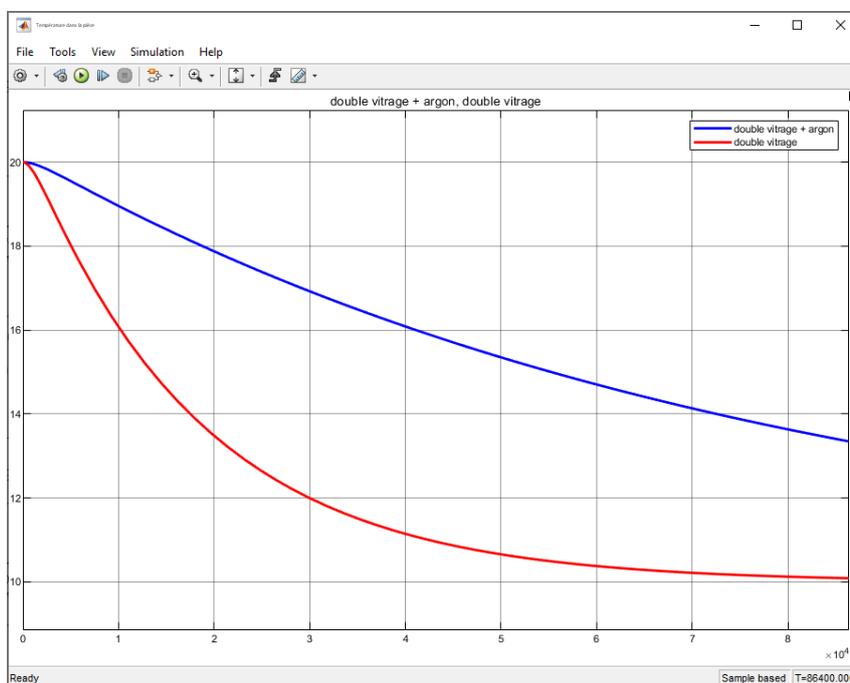
Le fichier « **double\_vitrage\_1.slx** » contient le modèle entièrement paramétré et prêt à être simulé.

Pour être certain d'obtenir la valeur en régime permanent du flux thermique, le temps de simulation est réglé à 24 heures.



**Figure 320 : réglage du temps de simulation**

Lancer la simulation en cliquant sur



**Figure 321 : évolution de la température dans la pièce en fonction du temps pour les deux types de vitrage**

Nous pouvons constater l'efficacité thermique du double vitrage + argon qui permet à la température de la pièce de se maintenir plus durablement.

Il est très simple à ce stade de choisir les paramètres du modèle à faire varier en modifiant les valeurs des paramètres dans le script.

## 22. Modélisation thermique d'une maison

Les paragraphes précédents ont présenté la théorie et les concepts permettant de modéliser les différents types d'échanges thermiques que l'on peut rencontrer dans la modélisation thermique d'une maison. Il est maintenant possible de comprendre l'exemple de la documentation Mathworks© qui modélise le chauffage d'une maison en prenant en compte les échanges thermiques par conduction et convection au travers du toit, des murs et des fenêtres.

Afin d'accéder à cet exemple, il faut ouvrir la documentation de **MATLAB** à partir de la barre de commande principale du logiciel.

- Cliquer sur le point d'interrogation  dans la barre de commande de **MATLAB** afin d'ouvrir la **Documentation**.
- Choisir l'onglet **Examples** dans la partie supérieure de la fenêtre et faire défiler le menu **Category** dans la partie gauche de l'écran pour faire apparaître **Simscape**
- Sélectionner **Simscape** afin que les exemples Simscape soit consultables sur la page principale
- Faire défiler les exemples jusqu'à voir apparaître les modèles thermiques de Simscape (Figure 322).

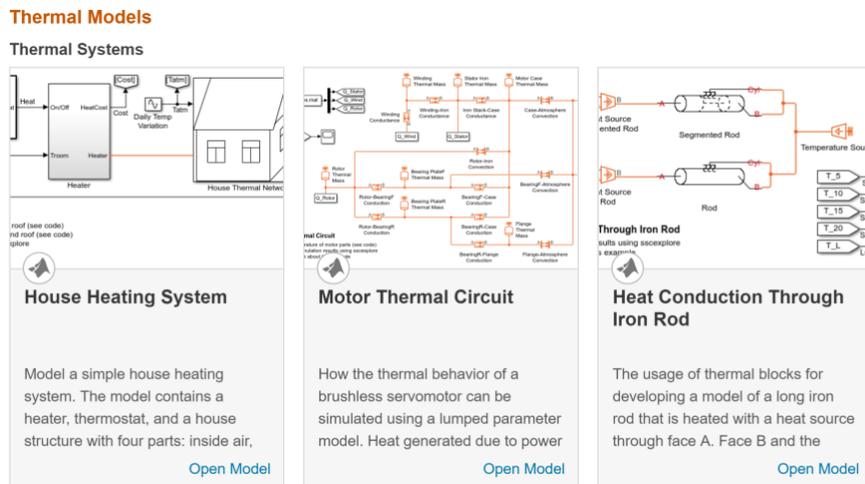


Figure 322 : modèles Simscape Thermal

Cliquer sur **Open Model** pour l'exemple **House Heating System**© afin de faire apparaître le modèle thermique d'une maison et la régulation de son système de chauffage.

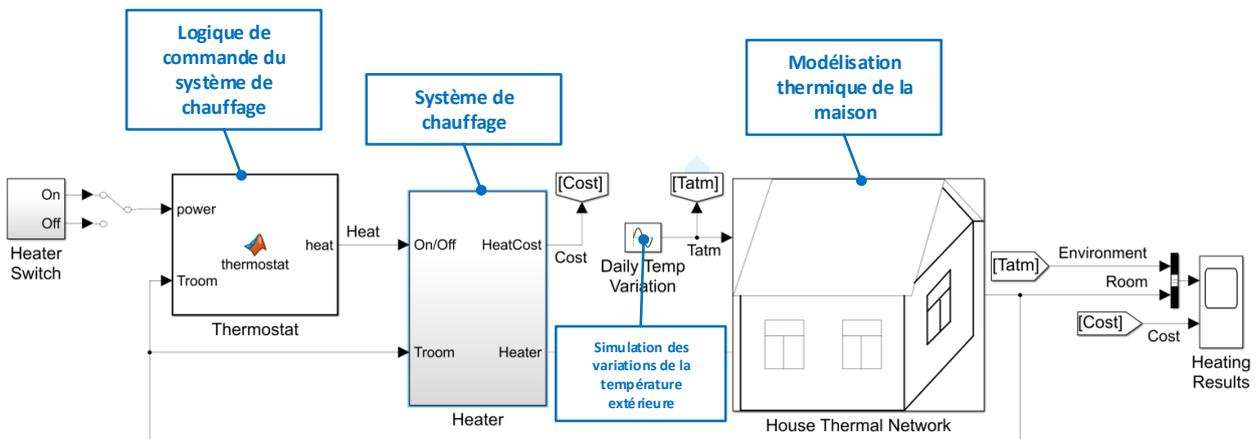


Figure 323 : modèle House Heating System , Mathworks ©

Ce modèle permet de simuler le fonctionnement du système de chauffage de la maison et de sa logique de commande. Le comportement thermique de la maison est entièrement modélisé à l'aide de Simscape. En cliquant sur le sous-système **House Thermal Networks**, il est possible de visualiser le modèle Simscape de la maison (Figure 324).

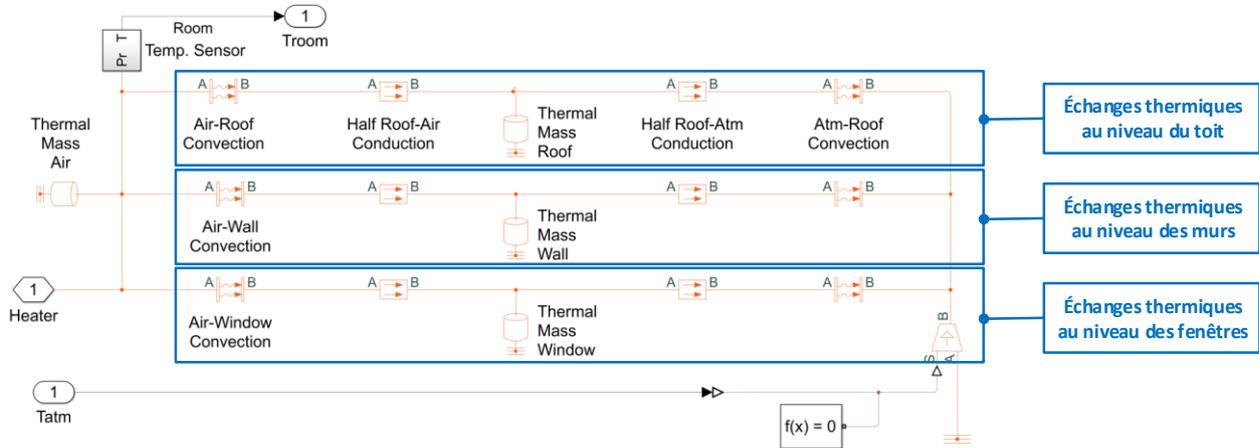


Figure 324 : modèle thermique de la maison réalisé avec Simscape

On retrouve sur la Figure 324 les structures des modèles présentées dans les paragraphes précédents et qui permettent de prendre en compte les échanges thermiques par conduction et convection au travers du toit, des murs et des fenêtres.

Le modèle permet de voir comment il est possible de stabiliser la température à l'intérieure de la maison en fonction des fluctuations de la température extérieure. Il sera alors possible de valider les performances du modèle :

- Fonctionnement de la logique de commande du système de chauffage
- Capacité du système de chauffage à fournir une puissance suffisante
- Fluctuation de la température à l'intérieure de la maison...

Il sera également possible d'avoir accès à de nombreuses informations techniques utiles lors du processus de conception :

- Température du mur, du toit des fenêtres au centre des parois
- Flux thermique au travers des différentes parois...

**Lancer** la simulation et ouvrir le scope (Figure 325).

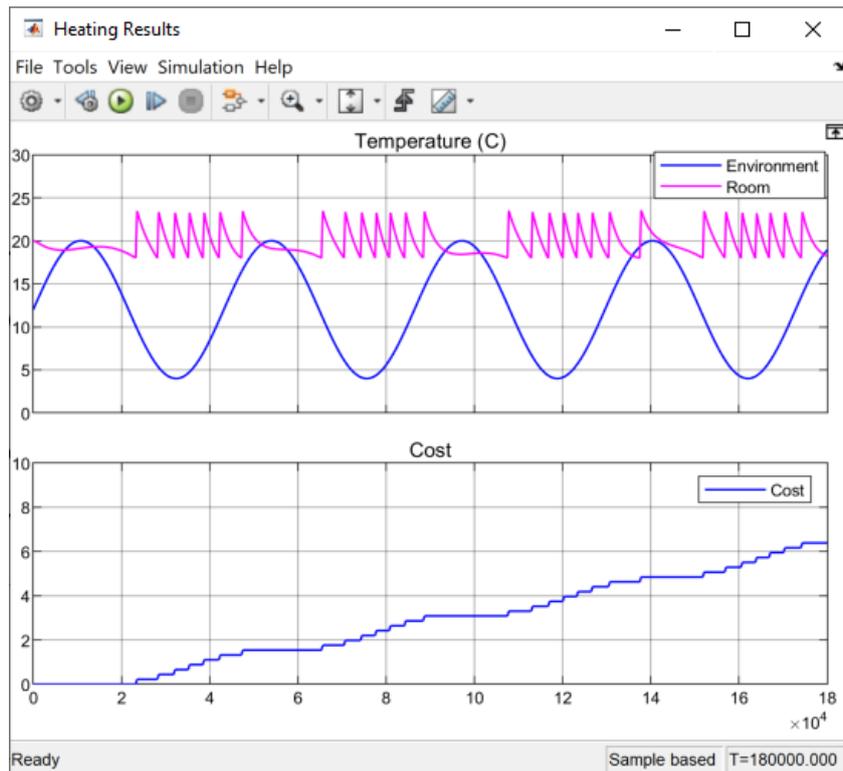


Figure 325 : évolution des températures intérieures et extérieures et du coût du chauffage

L'analyse de la Figure 325 nous permet de voir que la température à l'intérieur de la maison reste approximativement constante autour de 20°C avec une amplitude de variation de 5°C maximum alors que la température extérieure présente des variations de l'ordre de 15°C.

En cliquant sur **Plot Temperature** dans le modèle il est possible de visualiser les températures des différentes parois de la maison (Figure 326).



Figure 326 : température des différentes parois de la maison

En cliquant sur **Plot Heat Flow** dans le modèle il est possible de visualiser les flux de chaleur au travers des différentes parois de la maison (Figure 327).

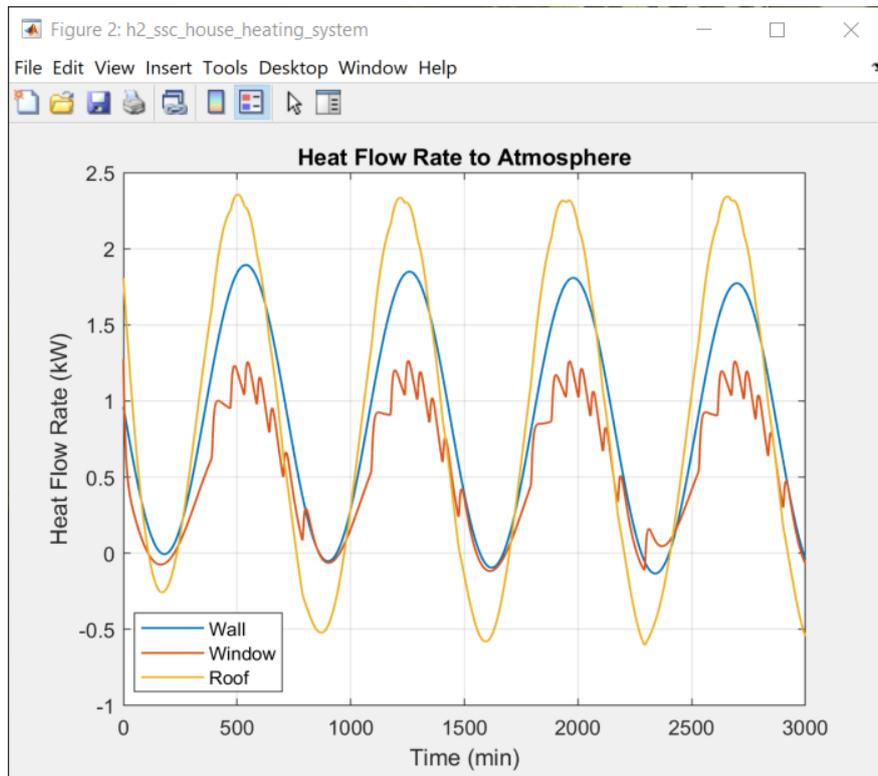


Figure 327 : flux de chaleur au travers des différentes parois de la maison

Il est également possible en double-cliquant sur l'interrupteur **Heater Switch** de couper le système de chauffage.

**Placer** l'interrupteur sur la position **Off** et **relancer** la simulation.

**Ouvrir** le scope et visualiser le résultat sur la Figure 328.

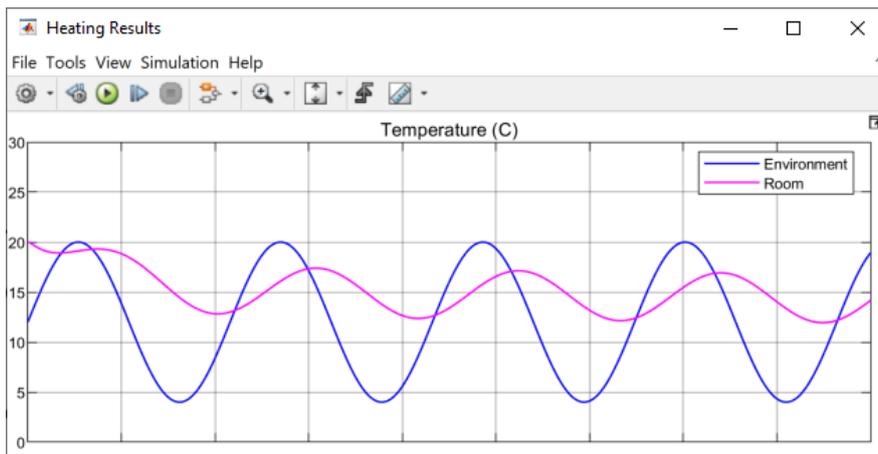


Figure 328 : évolution de la température sans système de chauffage

Nous constatons que sans le système de chauffage la fluctuation de la température intérieure augmente et que la température moyenne est plutôt de l'ordre de 15°C.

Il est ainsi possible de tester un grand nombre de scénarios afin d'optimiser les performances du modèle et le coût du système de chauffage.

## V. Application pédagogique

Les paragraphes précédents ont montré comment utiliser les fonctionnalités de Simscape et de ses bibliothèques pour construire des modèles exploitables. Afin d'utiliser ces modèles à des fins pédagogiques, il est nécessaire d'approfondir l'analyse et d'expliquer la stratégie à mettre en œuvre pour utiliser ces modèles comme supports de formation.

La simple présentation d'un modèle à des étudiants n'est pas suffisante pour garantir une efficacité dans le processus d'apprentissage. Un travail de réflexion pédagogique de la part de l'enseignant est impératif pour construire une séquence pertinente autour de l'exploitation d'un modèle numérique. Dans l'exemple qui va suivre, nous allons voir comment il est possible de passer de la simple conception d'un modèle à son exploitation pédagogique avec des étudiants. Cette démarche sera illustrée en exploitant le modèle du hacheur série.

### A. Présentation du hacheur série

Les hacheurs sont des convertisseurs directs du type continu-continu. Ils permettent d'obtenir une tension continue réglable à partir d'une tension continue fixe. Ils sont particulièrement utilisés pour la commande des machines à courant continu (Figure 329).

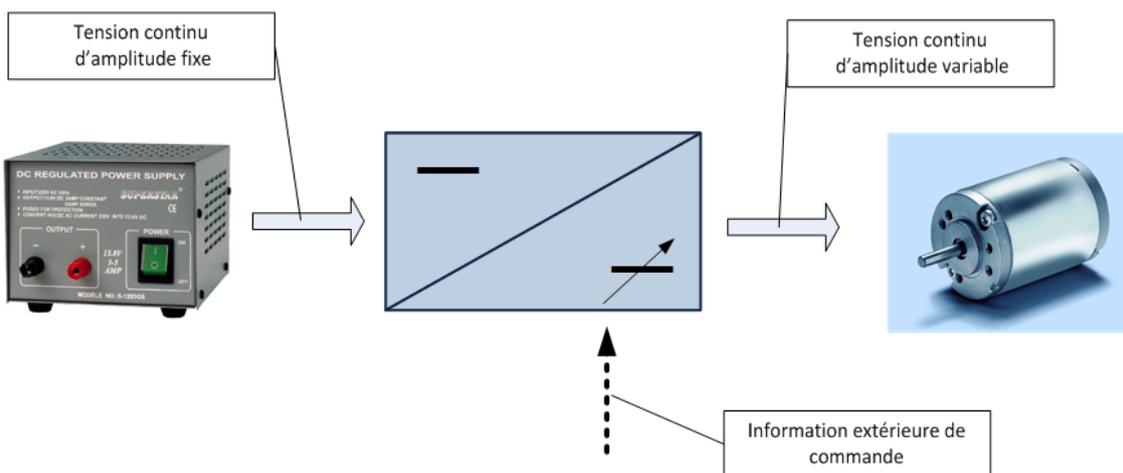


Figure 329 : présentation du hacheur série

Le hacheur va convertir une tension continue en une tension qui prend la forme d'un signal carré (Figure 330)

- $T$ : période de hachage (ou de découpage)
- $\alpha = \frac{t_{on}}{T}$ : rapport cyclique, si  $\alpha = 1$ , la tension de sortie est continu

Le signal obtenu en sortie du hacheur est appelé signal PWM (Pulse Width Modulation – Modulation de la largeur d'impulsion).

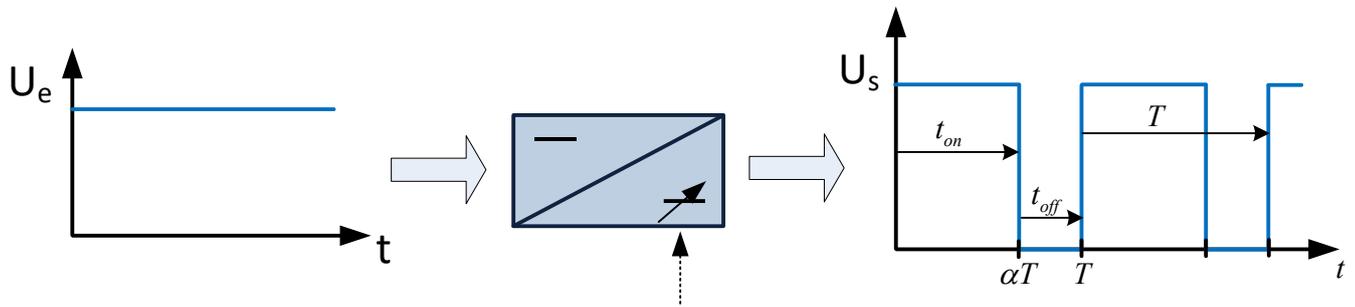


Figure 330 : principe de fonctionnement du hacheur série

Une source d'entrée délivre une tension continue  $U_e$ . Périodiquement, on ferme un interrupteur K pour obtenir la tension  $U_s$  aux bornes de la charge. En pratique, l'interrupteur K est un transistor qui travaille en commutation.

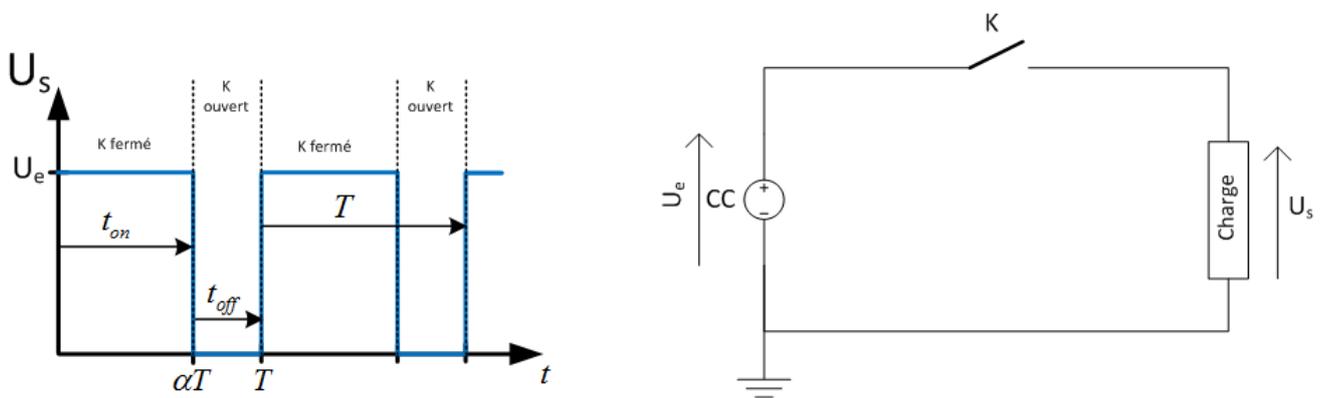


Figure 331 : schéma de principe du hacheur série

Durant une période  $T$  :

- la charge est alimentée durant la durée  $t_{on}$ , c'est la phase active
- l'alimentation est coupée durant la durée  $t_{off}$ , c'est la phase de roue libre

En pratique, la fréquence de hachage est de l'ordre de quelques kHz et la charge se comporte comme si elle était alimentée par une tension moyenne  $\langle U_c \rangle = \alpha U_e$ .

Le hacheur série permettant de commander un moteur à courant continu doit comporter 2 interrupteurs (**Figure 332**) :

- l'interrupteur  $K_1$  est un transistor de type MOS ou IGBT, unidirectionnel en tension et en courant (commutation commandée)
- l'interrupteur  $K_2$  est une diode, unidirectionnelle en tension et en courant (commutation spontanée)

En phase active, le transistor  $K_1$  est équivalent à un interrupteur fermé. La diode  $K_2$  est bloquée ( $V_K > V_A$ ) et se comporte comme un interrupteur fermé.

En phase de roue libre, le transistor  $K_1$  est équivalent à un interrupteur ouvert. La diode  $K_2$  est passante ( $V_A = V_K$ ). Une diode ainsi utilisée est appelée « diode de roue libre ».

$K_1$  est un transistor de type MOS ou IGBT (commutation commandée)

$K_2$  est une diode (commutation spontanée)

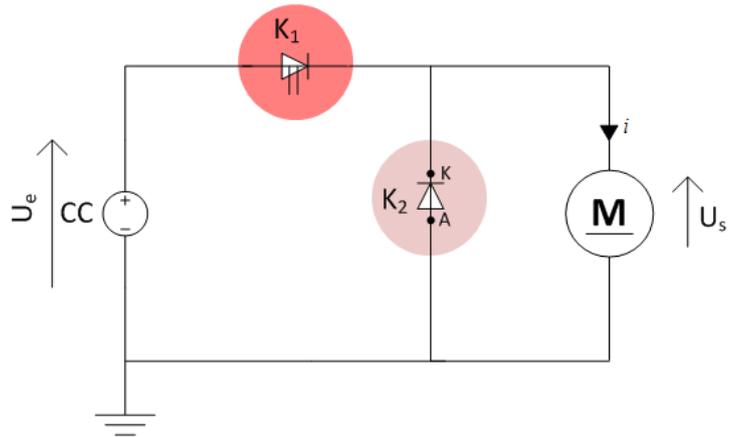


Figure 332 : schéma de commande d'un moteur à courant par un hacheur série

## B. Objectifs pédagogiques

A l'issue de la séquence l'étudiant doit être en mesure de comprendre les principes de fonctionnement du hacheur série :

**Objectif 1 : Comprendre la circulation du courant dans le circuit en phase active et en phase de roue libre (Figure 333).**

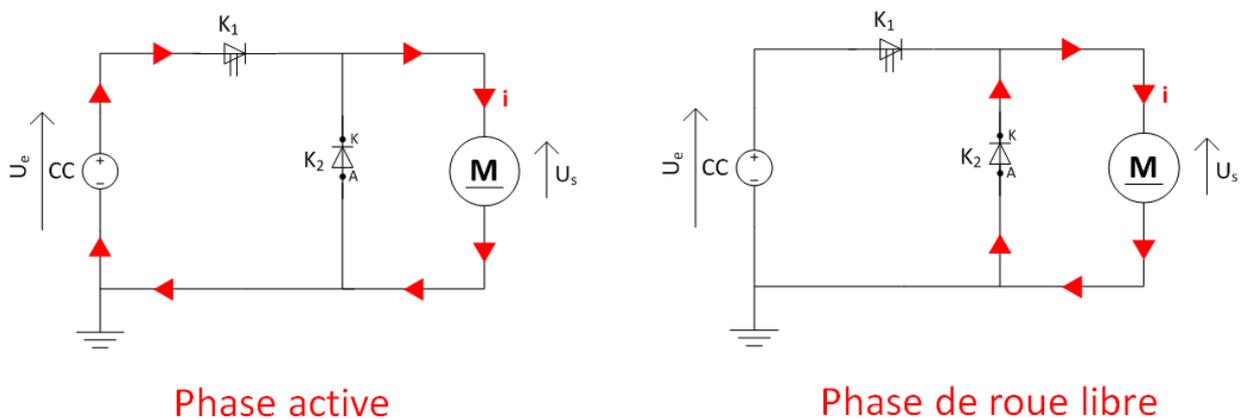


Figure 333 : circulation du courant dans le circuit en phase active et en phase de roue libre

**Objectif 2 : Visualiser et évaluer l'influence du rapport cyclique sur l'ondulation du courant (Figure 334)**

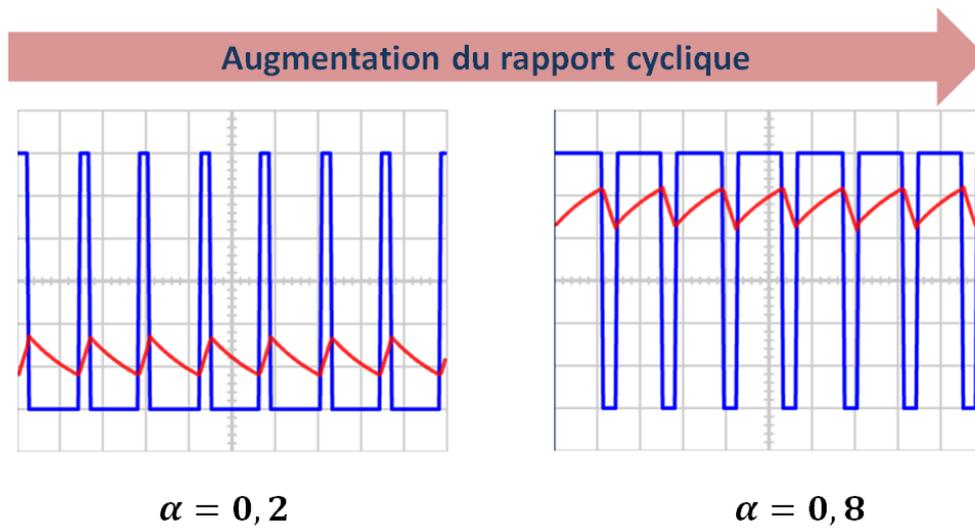


Figure 334 : visualisation de l'influence du rapport cyclique sur l'ondulation de courant

**Objectif 3 : Visualiser et évaluer l'influence de la fréquence de hachage sur l'ondulation du courant (Figure 335)**

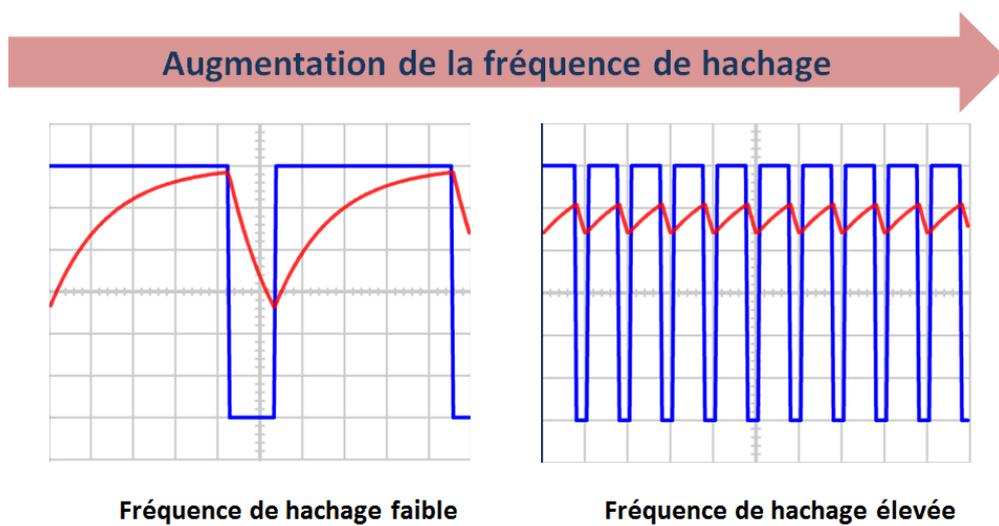


Figure 335 : visualisation de l'influence de la fréquence de hachage sur l'ondulation de courant

**Objectif 4 : Visualiser et évaluer l'influence de la valeur de l'inductance sur l'ondulation du courant (Figure 336)**

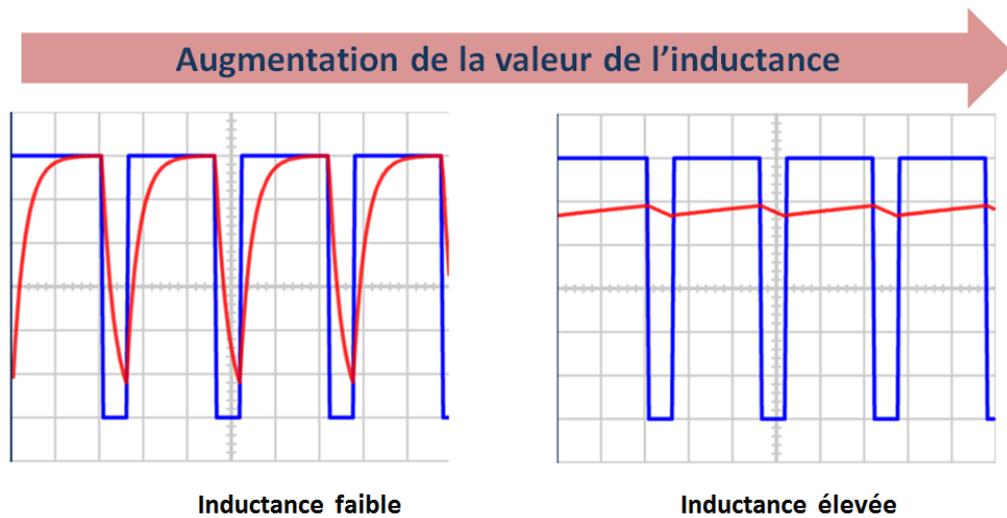
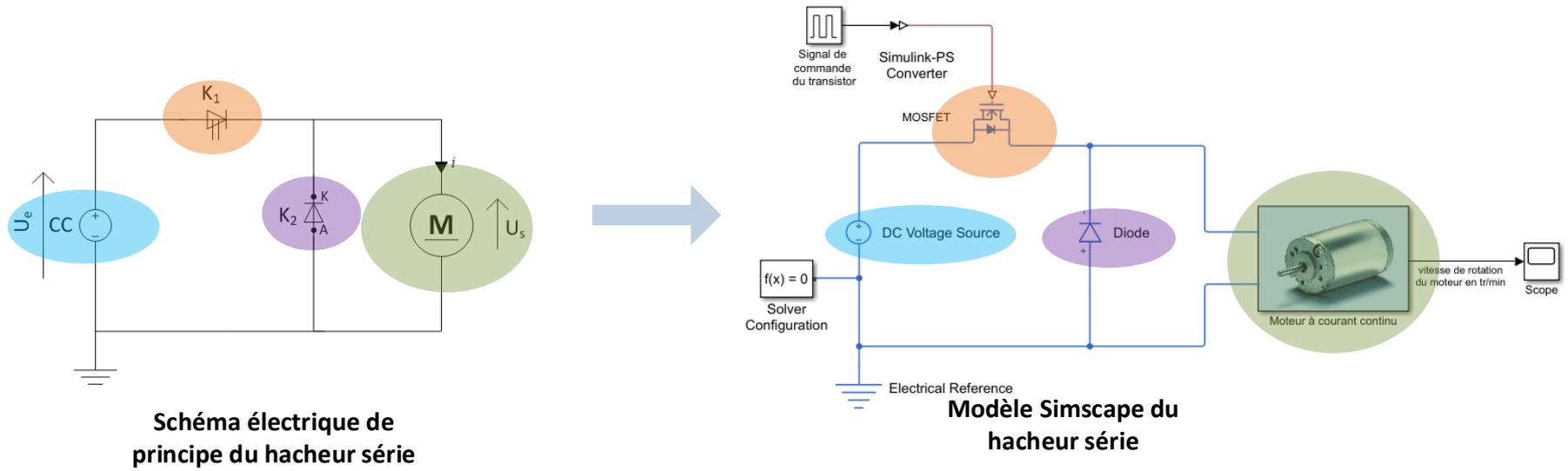


Figure 336 : visualisation de l'influence de la valeur de l'inductance sur l'ondulation de courant

### C. La construction du modèle

---

Le modèle numérique doit être construit, de manière à créer une analogie visuelle avec la schématisation afin de faciliter sa prise en main par les étudiants (Figure 337). L'étudiant identifie facilement tous les composants et peut faire le lien avec ses connaissances théoriques.



**Schéma électrique de principe du hacheur série**

**Modèle Simscape du hacheur série**

**Figure 337 : analogie entre la forme du modèle Simscape et le schéma électrique de principe**

Ouvrir le modèle « hacheur-serie\_0.slx ».

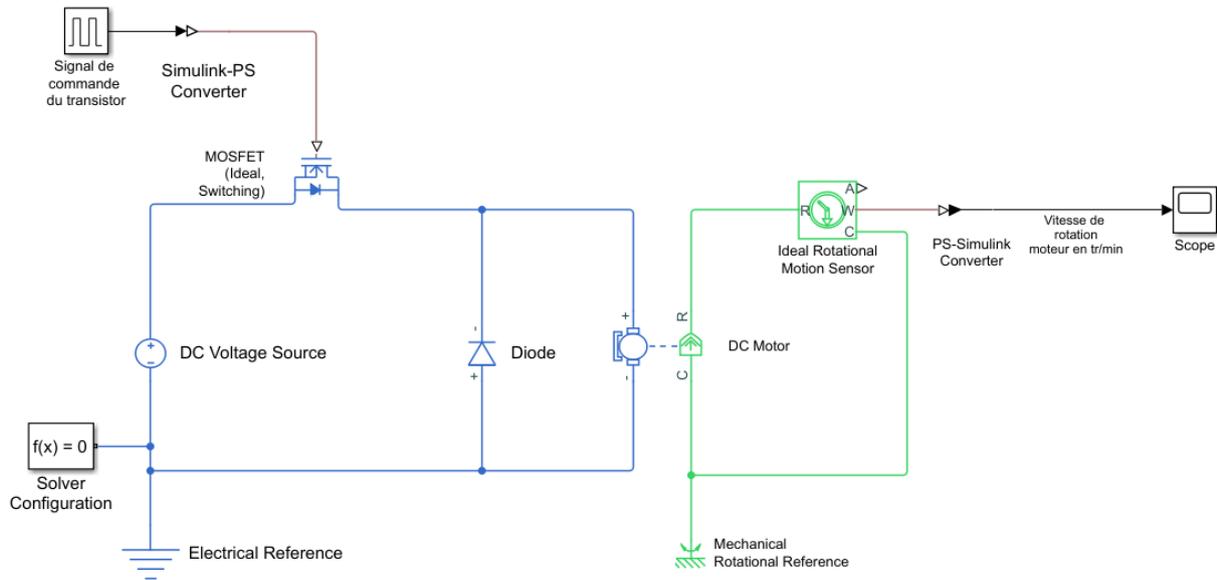


Figure 338 : modèle Simscape du hacheur série

Seuls sont référencés dans le tableau de la Figure 339 les composants qui n'ont pas encore été utilisés dans l'ouvrage.

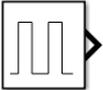
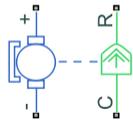
Fonction du composant	Représentation	Bibliothèque
Transistor MOSFET (Ideal Switching)	 MOSFET (Ideal, Switching)	Simscape/Electrical/Components/SemiConductors & Converters
Diode	 Diode	Simscape/Foundation Library/Electrical/Electrical Elements
Générateur de signaux carrés	 Pulse Generator	Simulink/Sources

Figure 339 : les composants nécessaires à la modélisation du hacheur série

Ce modèle fait intervenir le domaine électrique et le domaine mécanique de rotation.

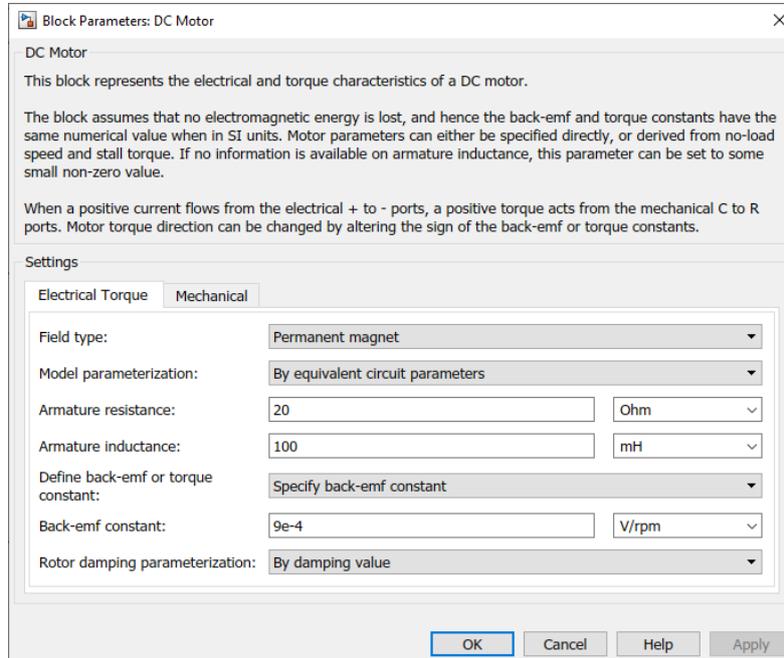
## Paramétrage

DC MOTOR



DC Motor

Simscape/Electrical/Electromechanical/Brushed Motors



## Paramétrage

DIODE



Diode

Simscape/Foundation Library/Electrical/Electrical Elements

Ce composant représente une diode. Le paramétrage se limite aux caractéristiques minimales du composant.

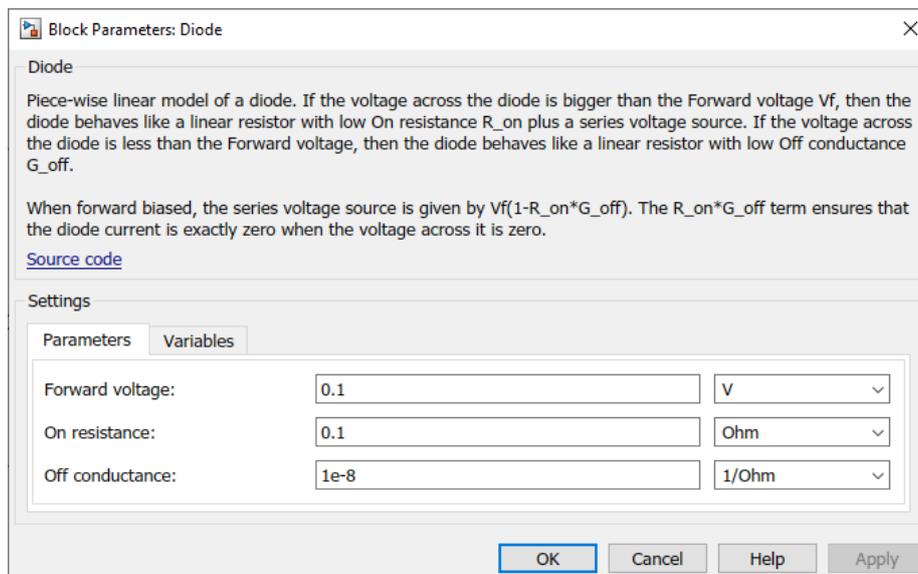


Figure 340 : paramétrage du bloc Diode

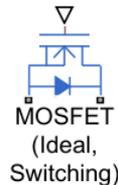
**Forward Voltage** : tension minimale à appliquer aux bornes de la diode pour la rendre passante (tension de seuil)

**On Resistance** : resistance de la diode à l'état passant

**Off Conductance** : conductance de la diode à l'état bloqué

## Paramétrage

MOSFET



Simscape/Electrical/ Components/SemiConductors & Converters

Ce composant représente un transistor de type MOSFET qui travaille en commutation. Le paramétrage se limite aux caractéristiques minimales du composant.

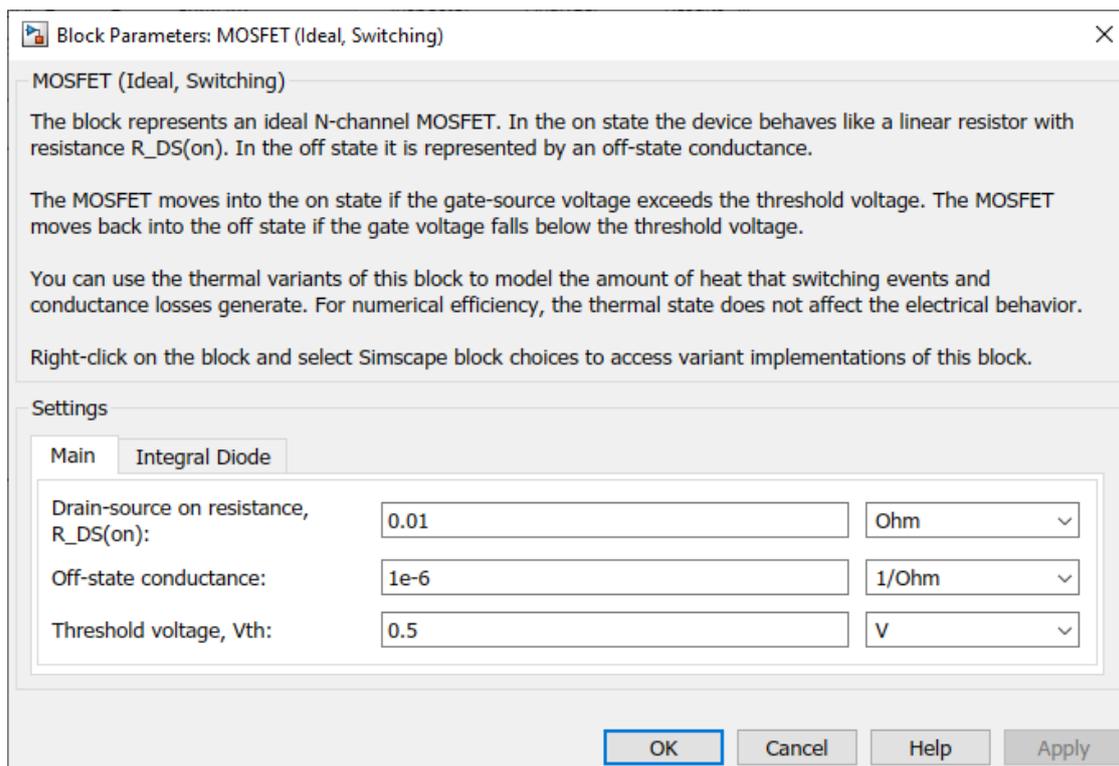


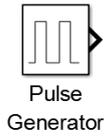
Figure 341 : paramétrage du bloc MOSFET

**Drain-source on resistance** : résistance à l'état passant

**Off-State Conductance** : conductance à l'état bloqué

**Threshold voltage** : tension de seuil de commutation

PULSE GENERATOR



Simulink/Sources

Ce composant permet de modéliser une source de signaux carrés en spécifiant les caractéristiques du signal.

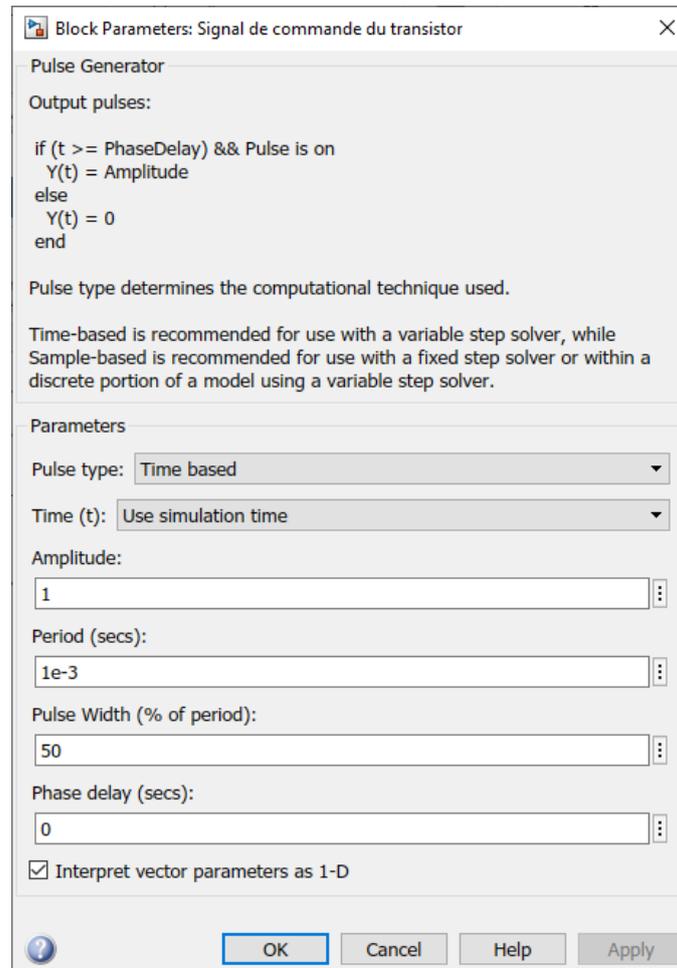


Figure 342 : paramétrage du bloc Pulse Generator

**Amplitude** : amplitude du signal carré

**Period** : période du signal carré

**Pulse Width** : rapport cyclique pouvant varier de 0% à 100%

**Phase Delay** : retard

Lancer la simulation et observer dans le scope l'évolution de la vitesse de rotation du moteur.

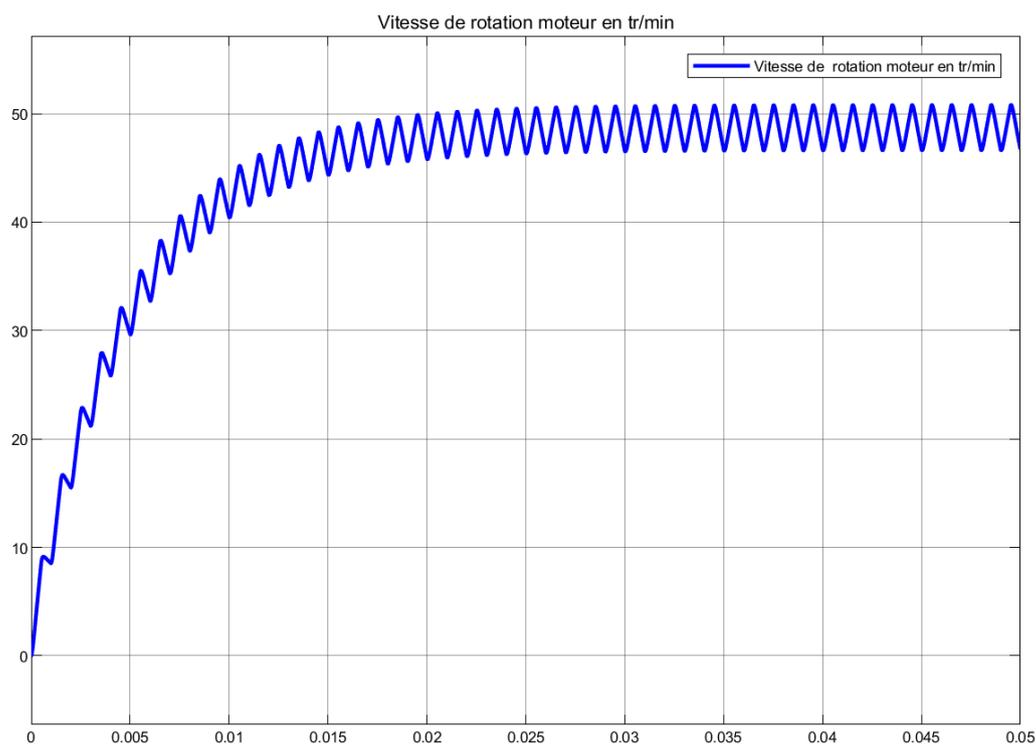


Figure 343 : évolution de la vitesse de rotation du moteur commandé par un hacheur série

La fréquence de hachage étant ici fixée à 1KHz et l'inertie de la charge étant très faible, la fréquence de commutation entraîne des irrégularités dans la vitesse de rotation du moteur.

Plusieurs étapes vont maintenant nous permettre de « **didactiser** » le modèle, c'est-à-dire de le rendre accessible à des étudiants et de le rendre compatible avec les objectifs de l'apprentissage visé.

## D. La didactisation du modèle

### 1. Création d'un sous-système et ajout d'une image

**Ouvrir** le modèle « **hacheur\_serie\_1.slx** ».

Ce modèle est le même que le précédent. Un sous-système représentant le moteur a été créé rendant plus visuel ce composant. Cette opération doit être réalisée à chaque fois que cela est possible pour améliorer la lisibilité d'un modèle.

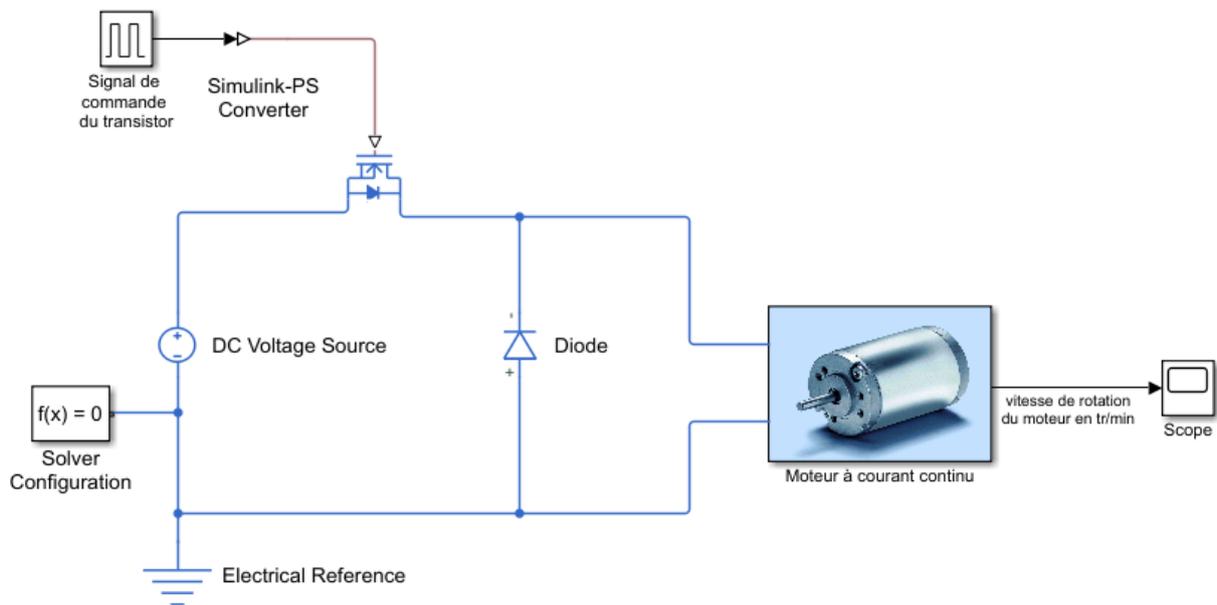


Figure 344 : création d'un sous-système pour didactiser le modèle

### 2. L'instrumentation du modèle

L'un des objectifs est de faire comprendre aux élèves comment circulent les courants dans le hacheur, nous allons mettre en place des capteurs de courant dans chacune des branches du circuit (Figure 345).

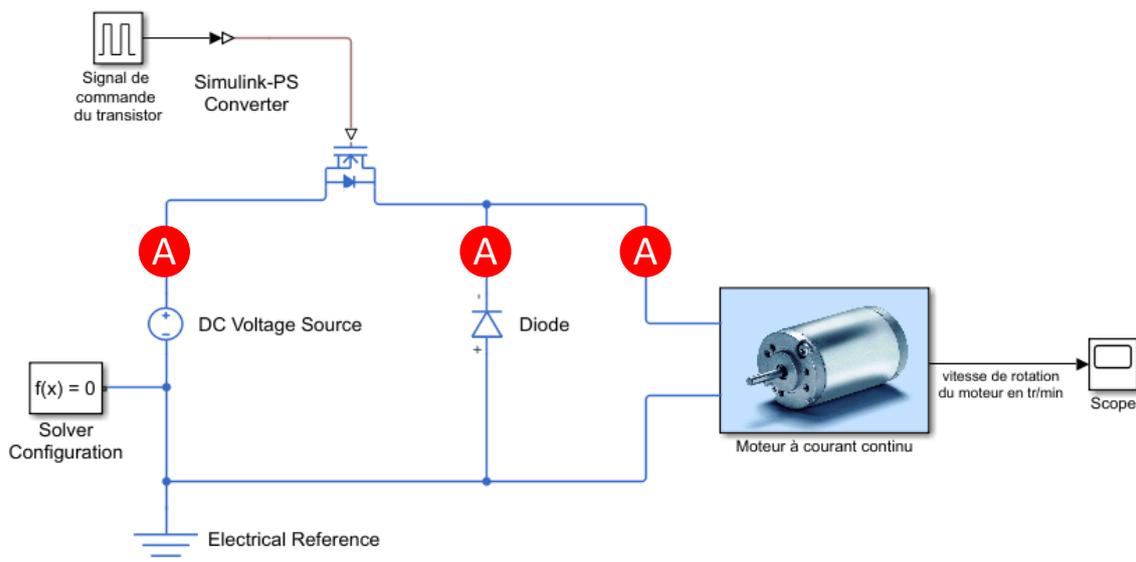


Figure 345 : mise en place des capteurs de courant dans le circuit

Nous pouvons mettre en place le premier capteur pour obtenir le modèle de la Figure 346.

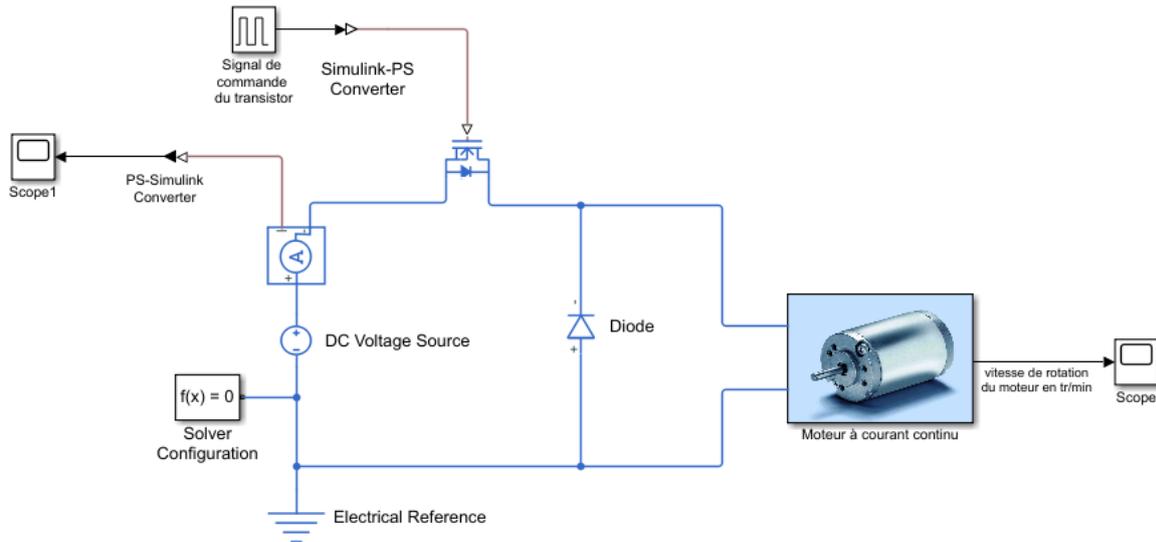


Figure 346 : ajout d'un capteur sur le modèle

Ce type d'instrumentation présente l'inconvénient majeur de surcharger le modèle en ajoutant un grand nombre de composants. Cette surcharge de composants va nuire à la lisibilité du modèle.

Afin d'optimiser cette phase et de gagner en lisibilité, il est conseillé de construire un sous-système réalisant la fonction souhaitée et de router les signaux avec des tags en les regroupant dans un même scope (les fonctionnalités de routage des signaux sont présentées page 161).

**Ouvrir** le fichier « **hacheur\_serie\_2.slx** » et observer le travail qui a été effectué pour instrumenter le modèle.

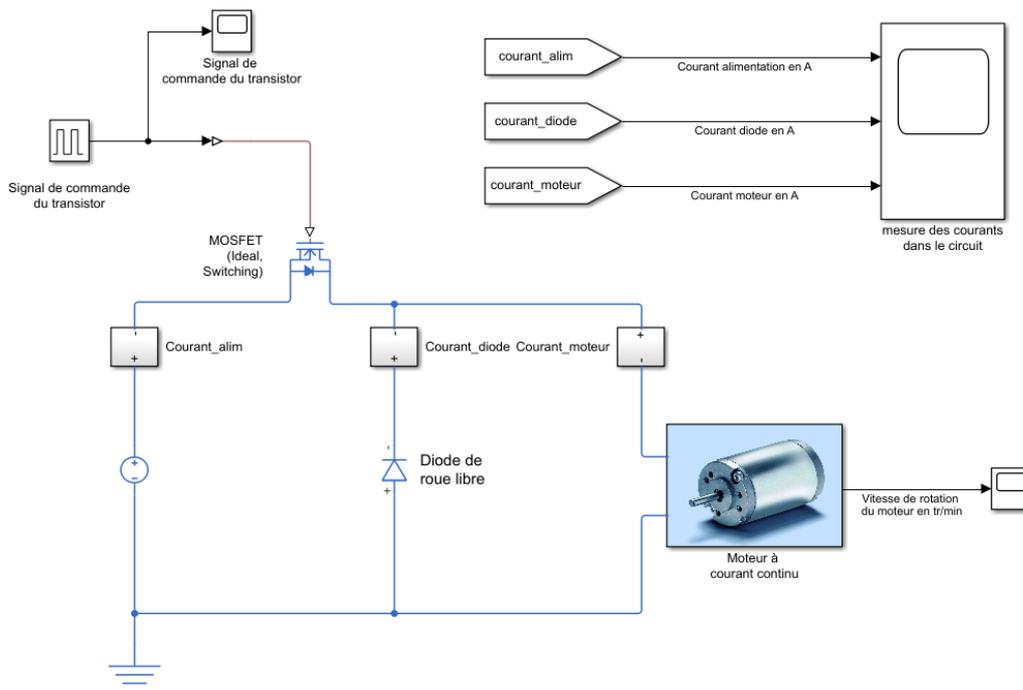
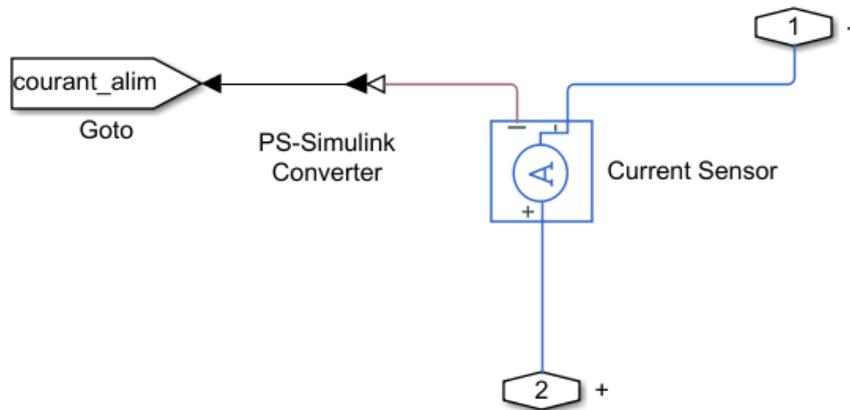


Figure 347 : modèle instrumenté et didactisé

**Double-cliquer** sur le capteur de courant « **Courant\_alim** » et observer le contenu du sous-système (Figure 348).



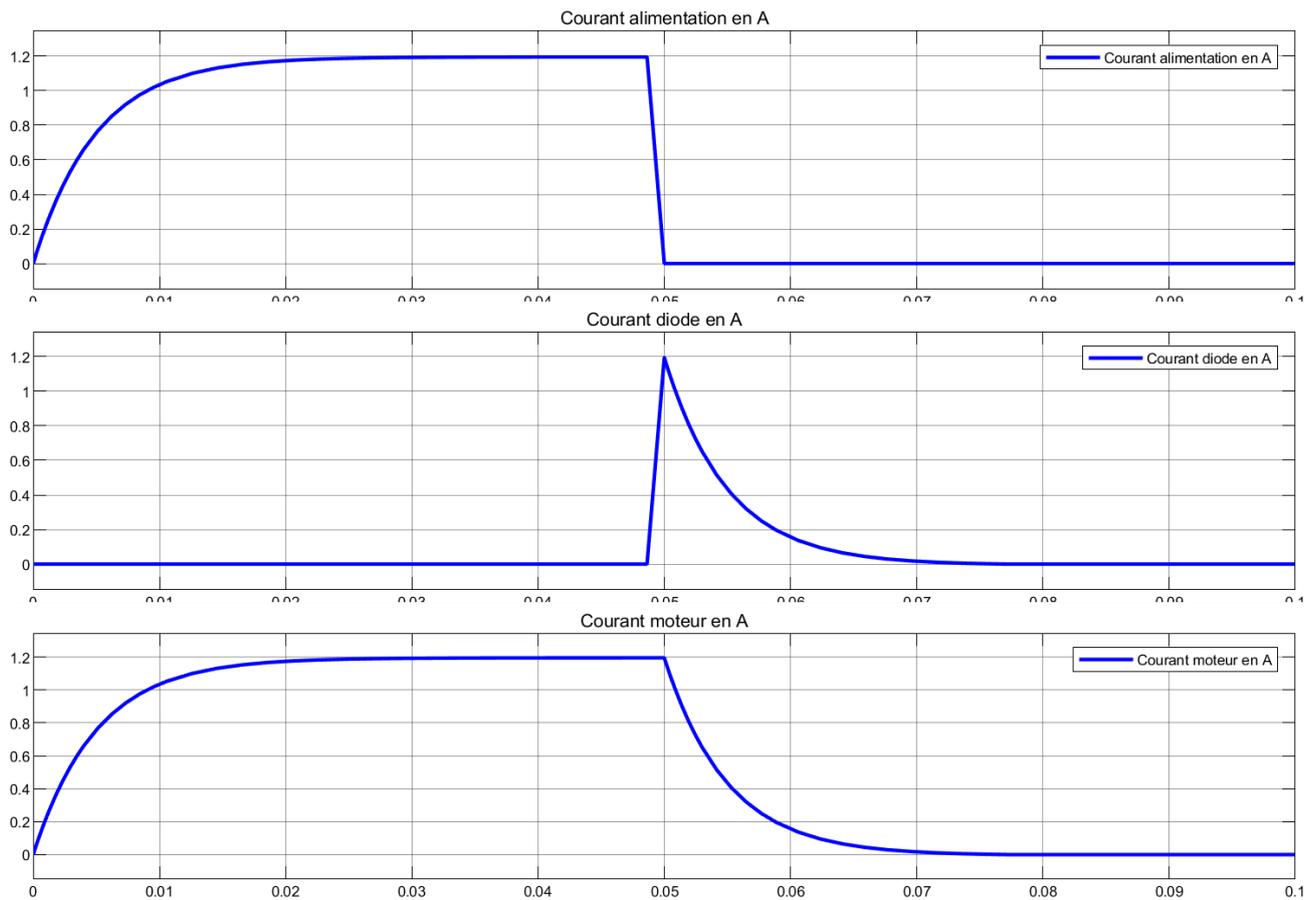
**Figure 348 : sous-système « capteur de courant »**

Ce capteur permet de relever l'intensité du courant dans une branche du circuit et route le signal à l'aide d'un tag qui sera repris par le scope.

L'avantage est que cette structure peut être recopiée et utilisée dans les deux autres branches du circuit afin de minimiser l'ajout de composants sur le modèle.

A ce stade le modèle devient exploitable par l'étudiant.

**Lancer** la simulation et visualiser l'allure des courants dans les différentes branches du circuit.



**Figure 349 : évolution du courant dans les trois branches du circuit**

Rappel :

Objectif 1 : Comprendre la circulation du courant dans le circuit en phase active et en phase de roue libre.

L'étudiant peut alors facilement faire le lien entre les résultats obtenus en simulation et le sens du passage du courant dans les différentes branches du circuit. La phase active et la phase de roue libre peuvent être identifiées. L'objectif 1 d'apprentissage peut être atteint.

### 3. Conclusion sur la didactisation du modèle

Les différentes phases d'amélioration de la lisibilité du modèle, que nous avons menées, participent à la « didactisation du modèle ». C'est une phase importante du travail qui permet de passer d'un modèle difficilement lisible par les étudiants à un modèle parfaitement lisible. Ces différentes étapes demandent un travail supplémentaire de la part de l'enseignant mais sont indispensables à la réussite de la séquence.

Le modèle didactisé :

- Il est construit pour permettre une prise en main rapide par un élève
- Il est facilement modifiable
- Il permet à l'élève de se focaliser sur l'objectif d'apprentissage visé
- Il doit dans la mesure du possible reproduire le descripteur

Le modèle non didactisé :

- Il donne les mêmes informations que le modèle didactisé
- L'objectif est centré sur les résultats exploitables
- Il est difficile à lire et à modifier
- Il ne doit pas être donné aux élèves

La Figure 350 montre le modèle non didactisé et permet de comprendre les difficultés de lecture et d'exploitation que l'utilisation de ce modèle pourraient engendrer.

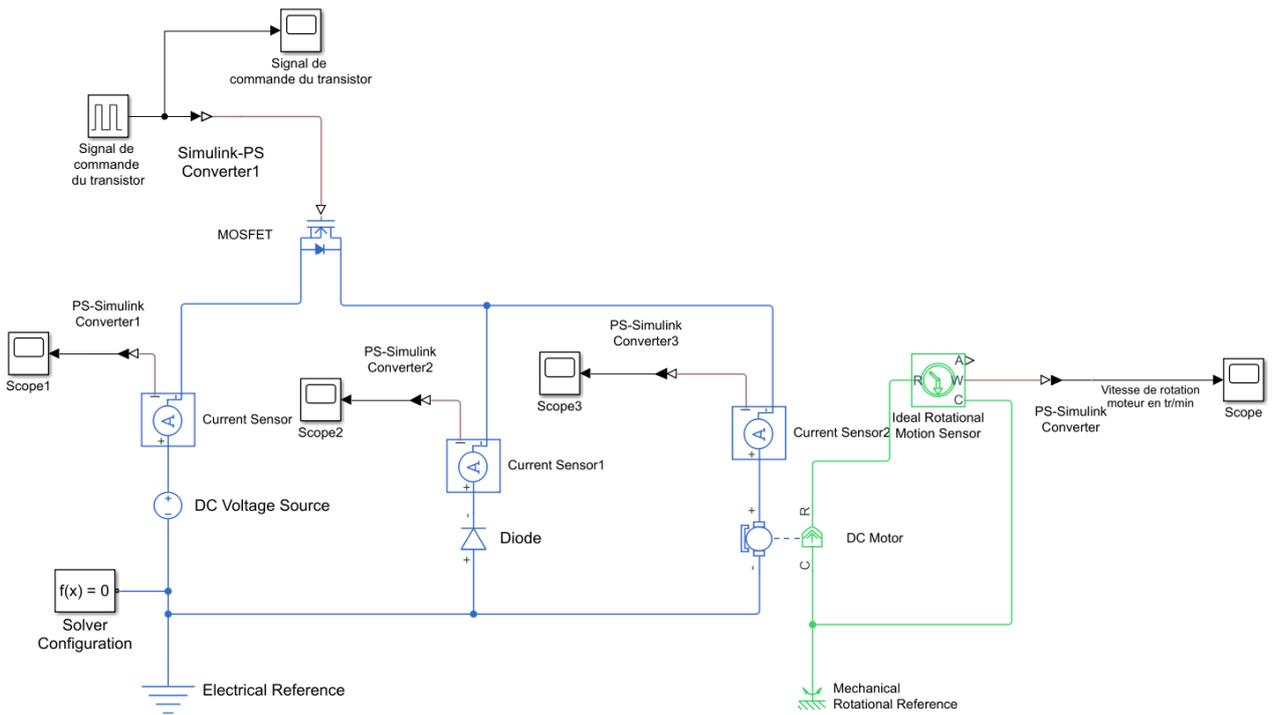


Figure 350 : le modèle non didactisé

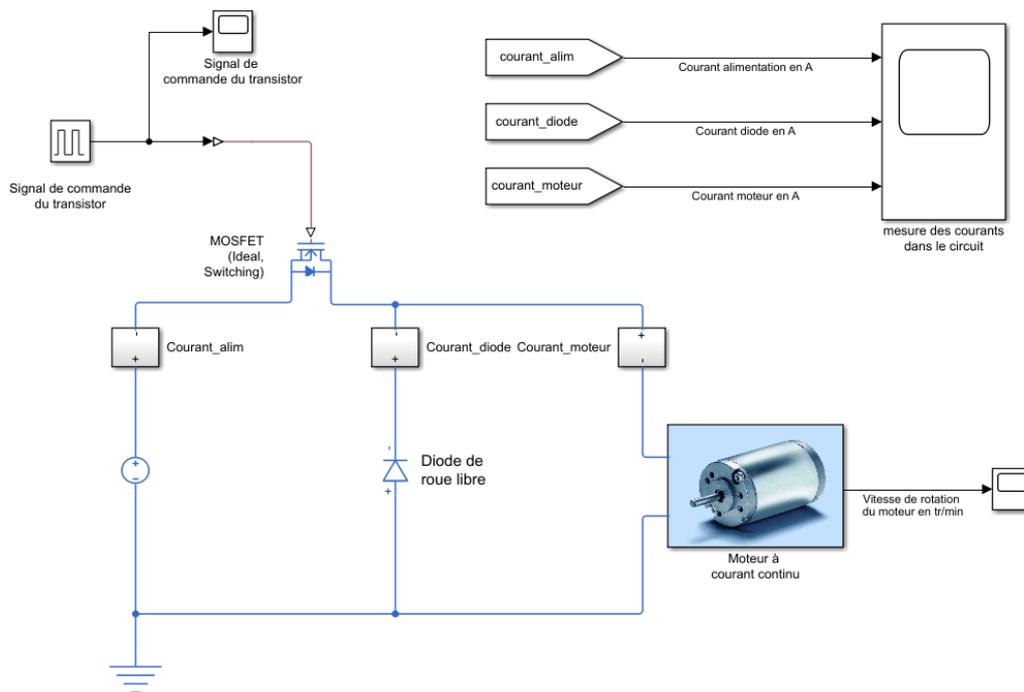


Figure 351 : le modèle didactisé

#### 4. Optimiser la didactisation du modèle en fonction de l'objectif d'apprentissage visé

Nous avons vu que la forme du modèle didactisé précédemment établie est parfaitement adaptée à la compréhension du premier objectif d'apprentissage de la séquence qui était de permettre aux élèves de comprendre la circulation du courant dans le hacheur série en phase active et en phase de roue libre.

Afin de poursuivre la séquence, et passer à l'objectif d'apprentissage suivant, il est nécessaire d'adapter la didactisation de notre modèle et de s'interroger sur les améliorations éventuelles à apporter pour atteindre les objectifs suivants.

Rappel :

Objectif 2, 3 et 4 : Visualiser et évaluer l'influence des différents paramètres sur l'ondulation du courant.

- Rapport cyclique
- Fréquence de hachage
- Valeur de l'inductance de la charge

Il faut focaliser l'attention des étudiants sur le courant moteur pour les objectifs suivants, les deux capteurs de courant dans les autres branches peuvent être supprimés. De plus il est nécessaire de disposer de la valeur moyenne du courant pour avoir une image du couple moteur (Figure 352).

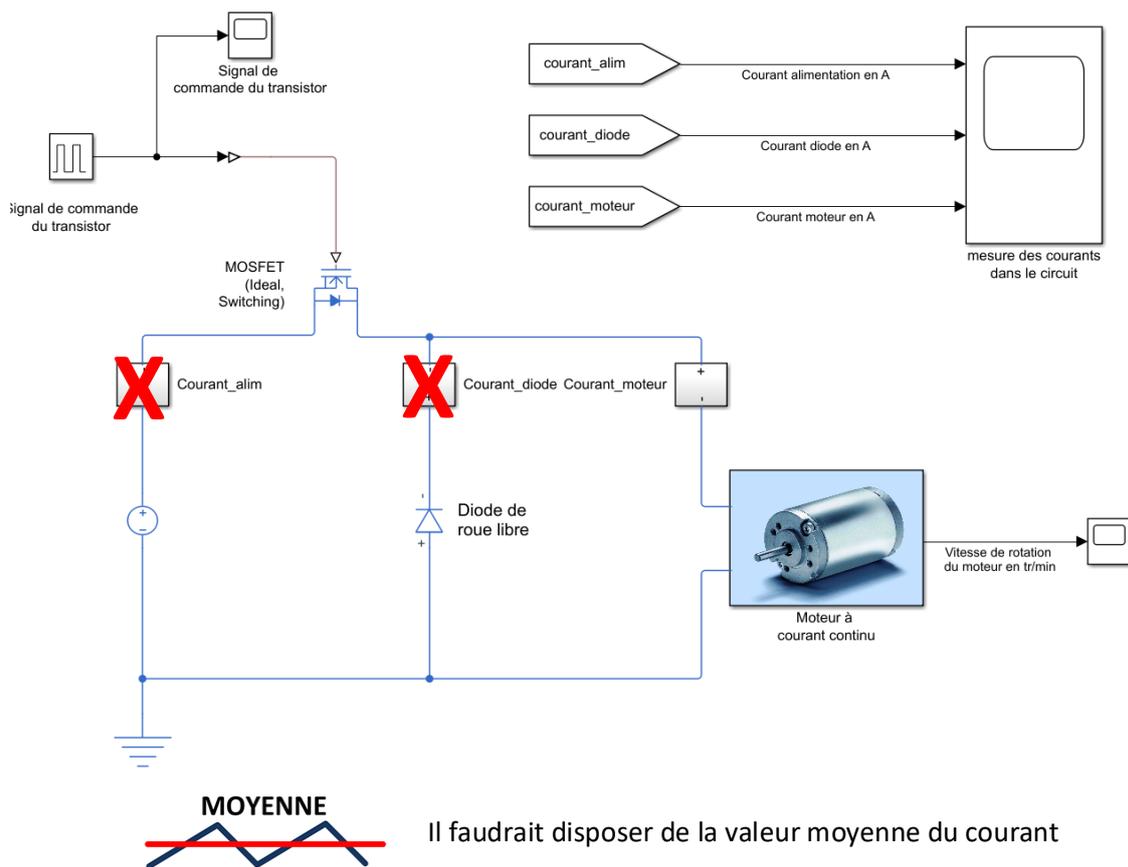
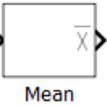


Figure 352 : amélioration de la didactisation du modèle

Le bloc **Mean** permet de calculer automatiquement la valeur moyenne d'un signal.

Fonction du composant	Représentation	Bibliothèque
Calcul de la valeur moyenne		Simscape/Electrical/Specialized Power Systems/Control and Measurements Library/Measurements
Affichage		Simulink/Sinks

**Ouvrir** le fichier « **hacheur\_serie\_3.slx** ». Ce fichier contient le modèle didactisé avec les améliorations proposées. Un **masque** a été mis sur le bloc **Mean** pour rendre plus explicite sa fonction (Figure 353).

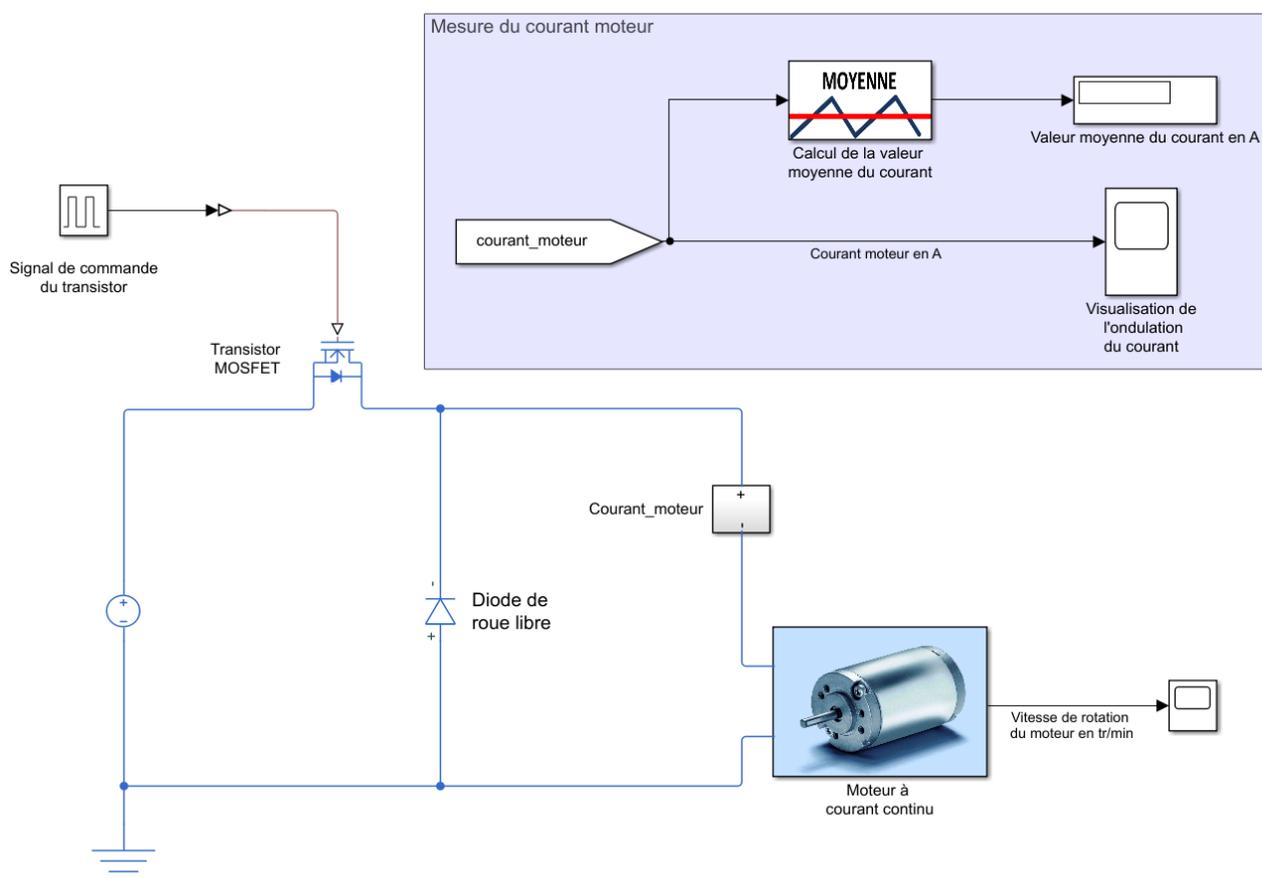
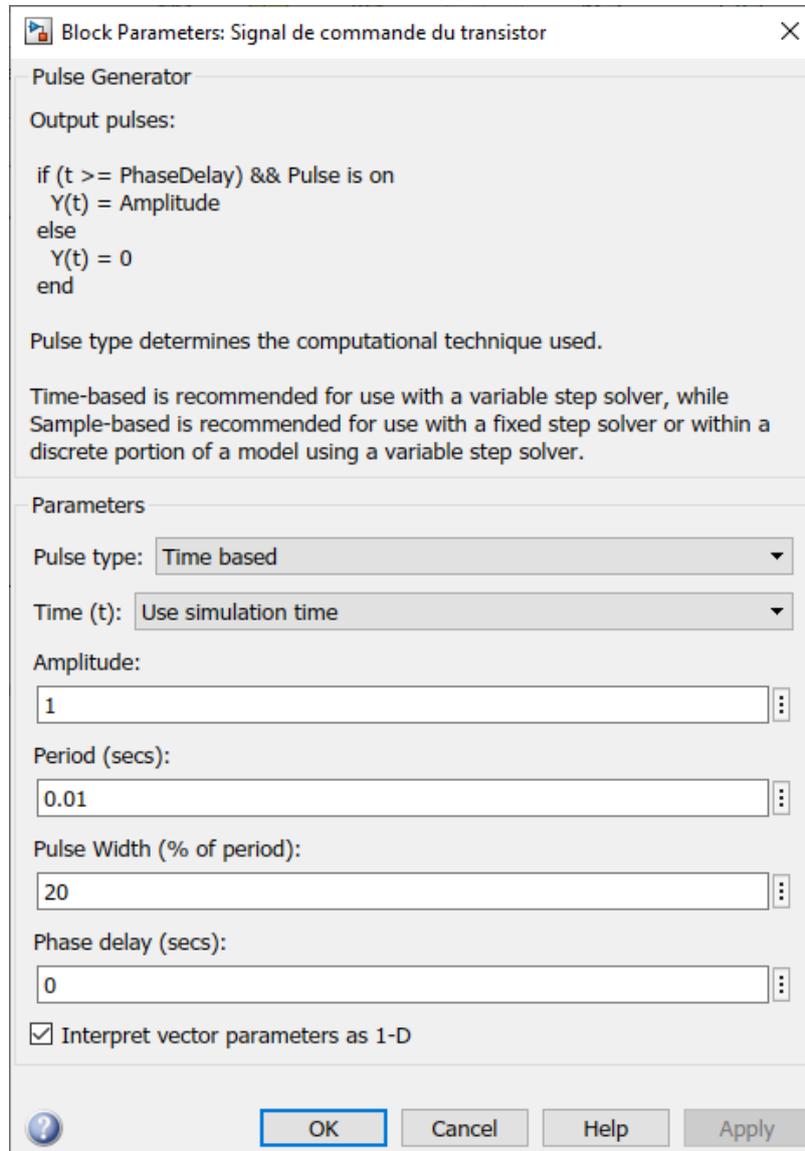


Figure 353 : le modèle didactisé et optimisé

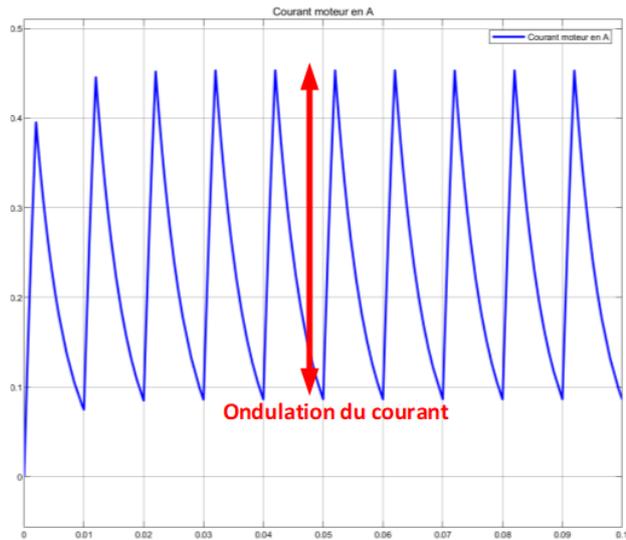
**Double cliquer** sur le bloc **Signal de commande du transistor**.

Régler la **période de commutation** du transistor sur **0.01 s** et spécifier un **temps de simulation** de **0.1 s**.

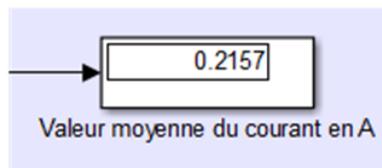


**Figure 354 : réglage de la période de commutation du transistor**

**Lancer** la simulation et constater que la valeur moyenne et l'allure du courant instantané sont maintenant disponibles dans le modèle. La valeur instantanée du courant est visualisable dans le scope et la valeur moyenne dans l'afficheur (Figure 355).



Valeur instantanée du courant moteur avec visualisation de l'amplitude de l'ondulation de courant



Valeur moyenne du courant moteur

**Figure 355 : visualisation de la valeur moyenne et instantanée du courant moteur**

Nous pouvons constater que les informations nécessaires à l'atteinte des objectifs sont très facilement accessibles par les étudiants. Ils pourront alors faire varier les paramètres du modèle comme la fréquence de hachage (période de commutation du transistor) ou le rapport cyclique.

Pour visualiser l'influence de l'inductance de la charge, il faudra faire une modification du modèle en insérant une inductance en série avec le moteur.

**Ouvrir** le fichier « **hacheur\_serie\_4.slx** ». Ce fichier contient le modèle précédent auquel a été ajoutée une inductance en série avec le moteur. L'étudiant pourra alors faire varier la valeur de l'inductance pour visualiser son influence sur le courant moteur (Figure 356).

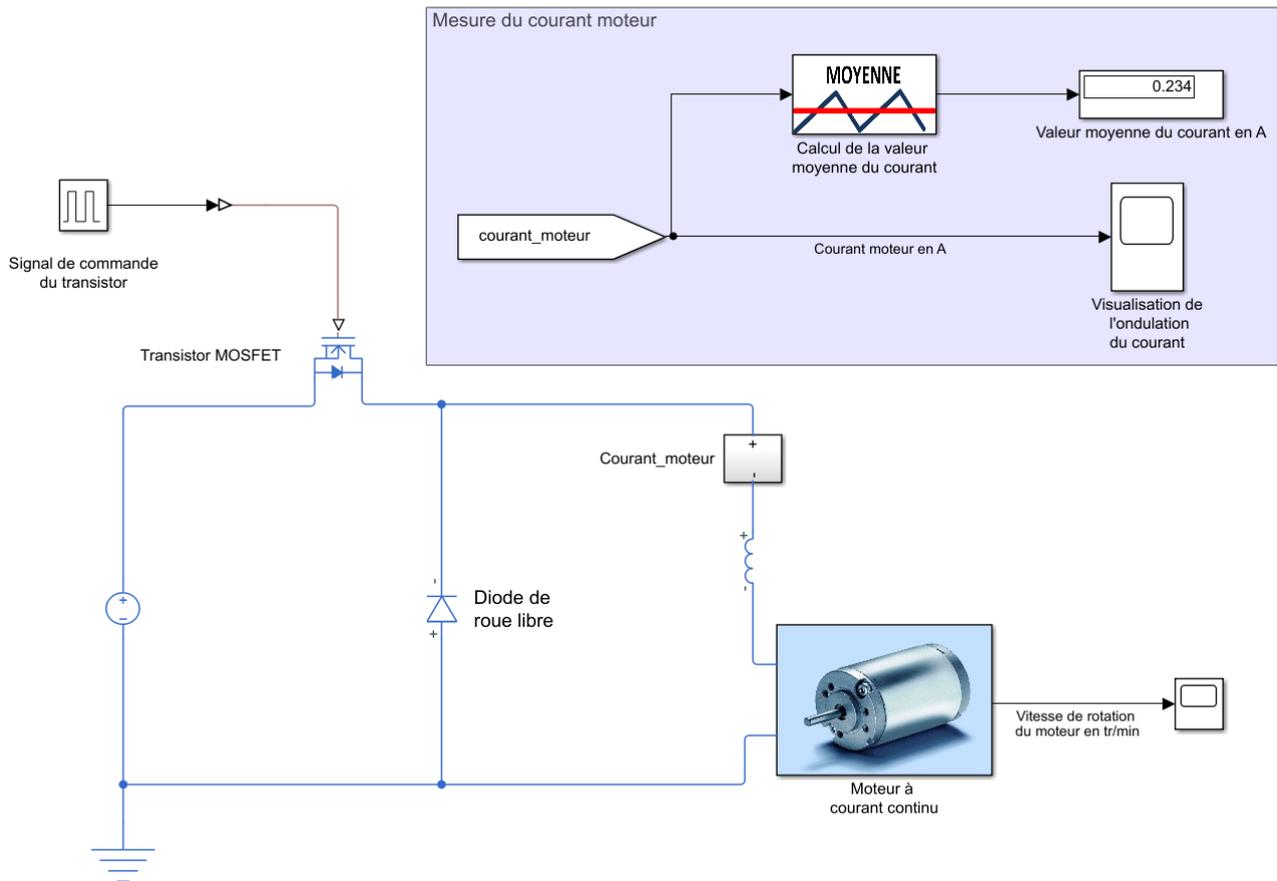


Figure 356 : ajout d'une inductance de lissage en série avec le moteur

## E. Exploitation des résultats issus de la simulation du modèle

Dans cette partie, les résultats issus de la simulation des différents modèles sont présentés et montre ce que l'étudiant doit être capable de mettre en évidence au travers de l'exploitation des différents modèles.

### 1. Objectif 1 : Comprendre la circulation du courant dans le circuit en phase active et en phase de roue libre

La simulation du modèle « **hacheur\_serie\_2.slx** » (Figure 357) permet de mettre en évidence les résultats suivants :

- En phase active, le courant qui traverse le moteur est identique à celui qui traverse l'alimentation. Le courant dans la branche de la diode est nul.
- En phase de roue libre, le courant qui traverse le moteur est identique à celui qui traverse la branche de la diode. Le courant qui traverse l'alimentation est nul.

# Circulation du courant

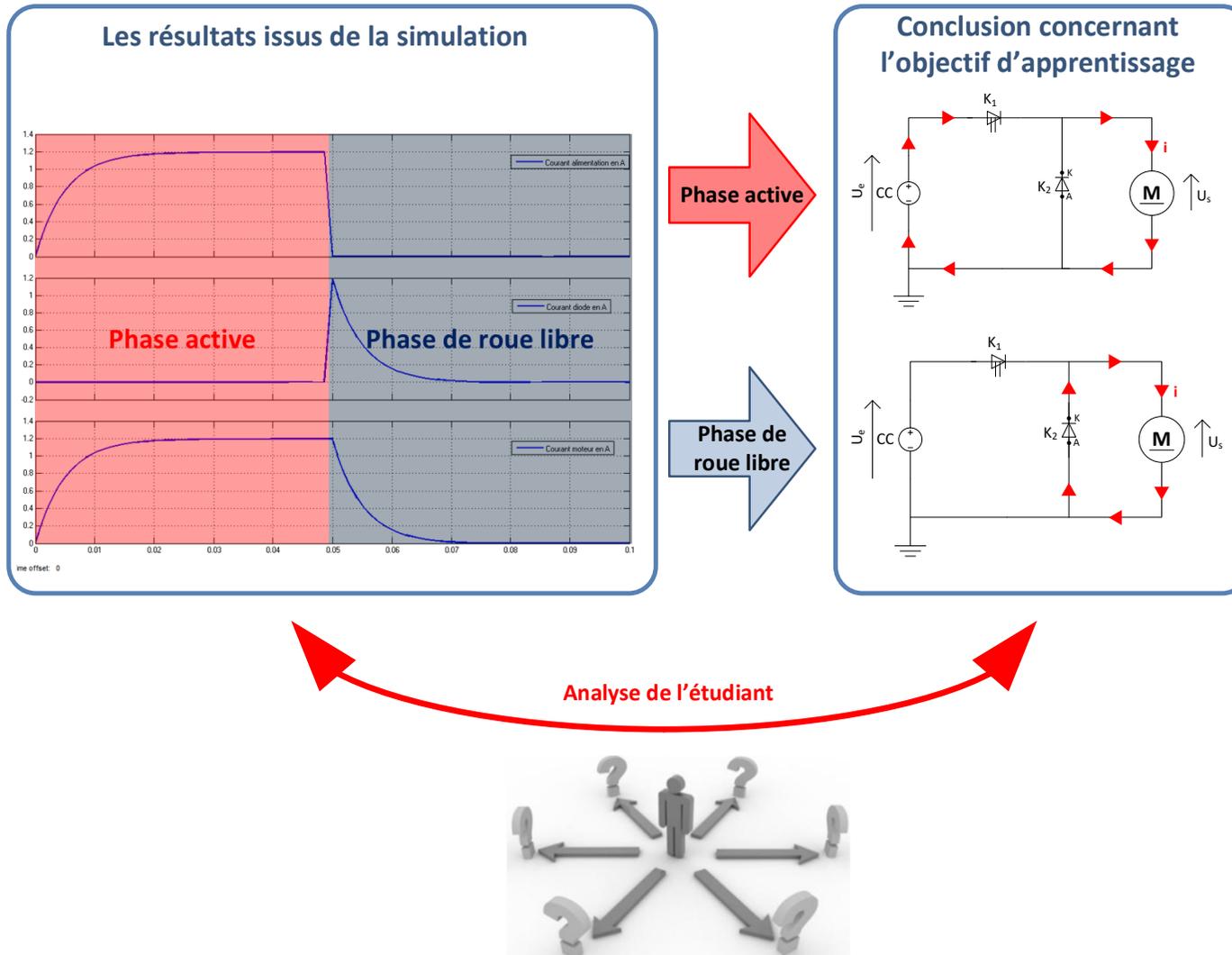


Figure 357 : exploitation et visualisation de la circulation du courant dans le hacheur série

## 2. Objectif 2 : Visualiser et évaluer l'influence du rapport cyclique sur le courant moteur

La simulation du modèle « **hacheur\_serie\_3.slx** » (Figure 359) permet de mettre en évidence l'influence du rapport cyclique.

**Régler** la période de commutation du transistor sur **0,0001 s**

**Régler** le rapport cyclique sur  $\alpha = 50\%$  (Figure 358)

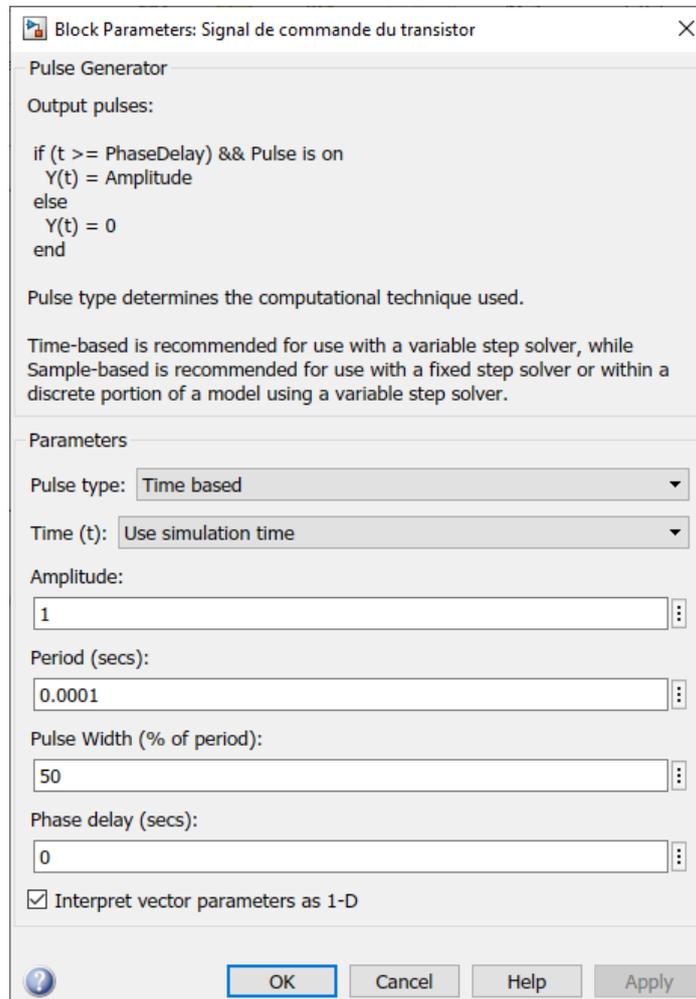


Figure 358 : réglages des paramètres de la simulation

**Lancer** la simulation et relever l'allure instantanée du courant moteur et sa valeur moyenne.

**Relancer** la simulation en modifiant la valeur du rapport cyclique sur  $\alpha = 90\%$ .

# Influence du rapport cyclique

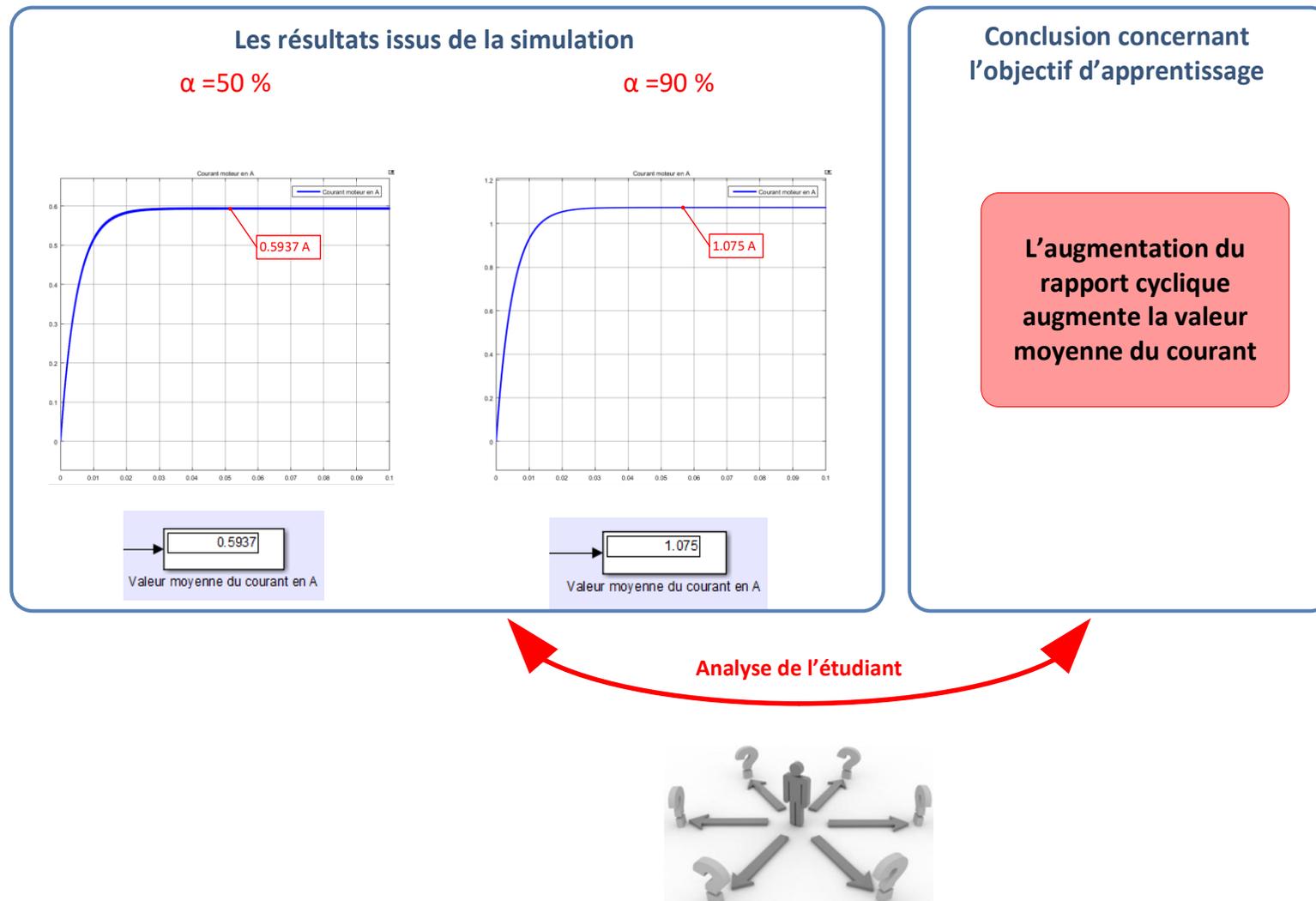


Figure 359 : exploitation et visualisation de l'influence du rapport cyclique sur le courant moteur

### 3. Objectif 3 : Visualiser et évaluer l'influence de la fréquence de hachage sur l'ondulation du courant

La simulation du modèle « **hacheur\_serie\_3.slx** » (Figure 361) permet de mettre en évidence l'influence du rapport cyclique.

**Régler** la période de commutation du transistor sur **0,001 s**

**Régler** le rapport cyclique sur  $\alpha = 50\%$

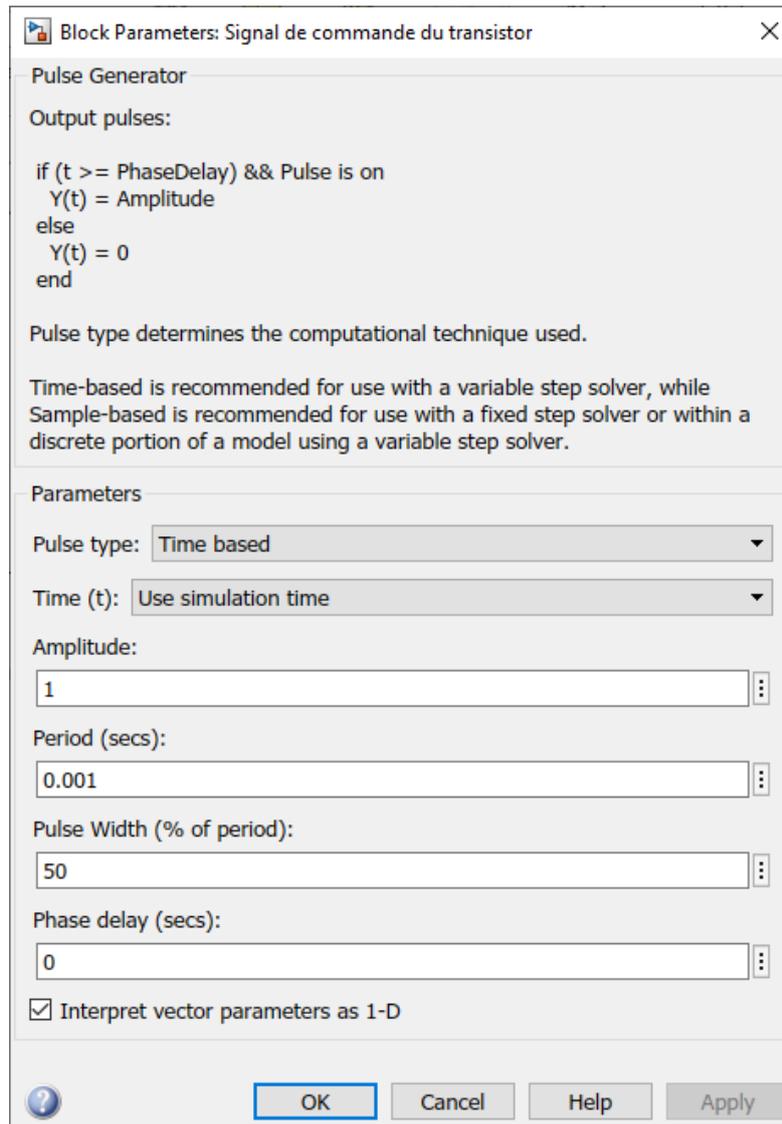
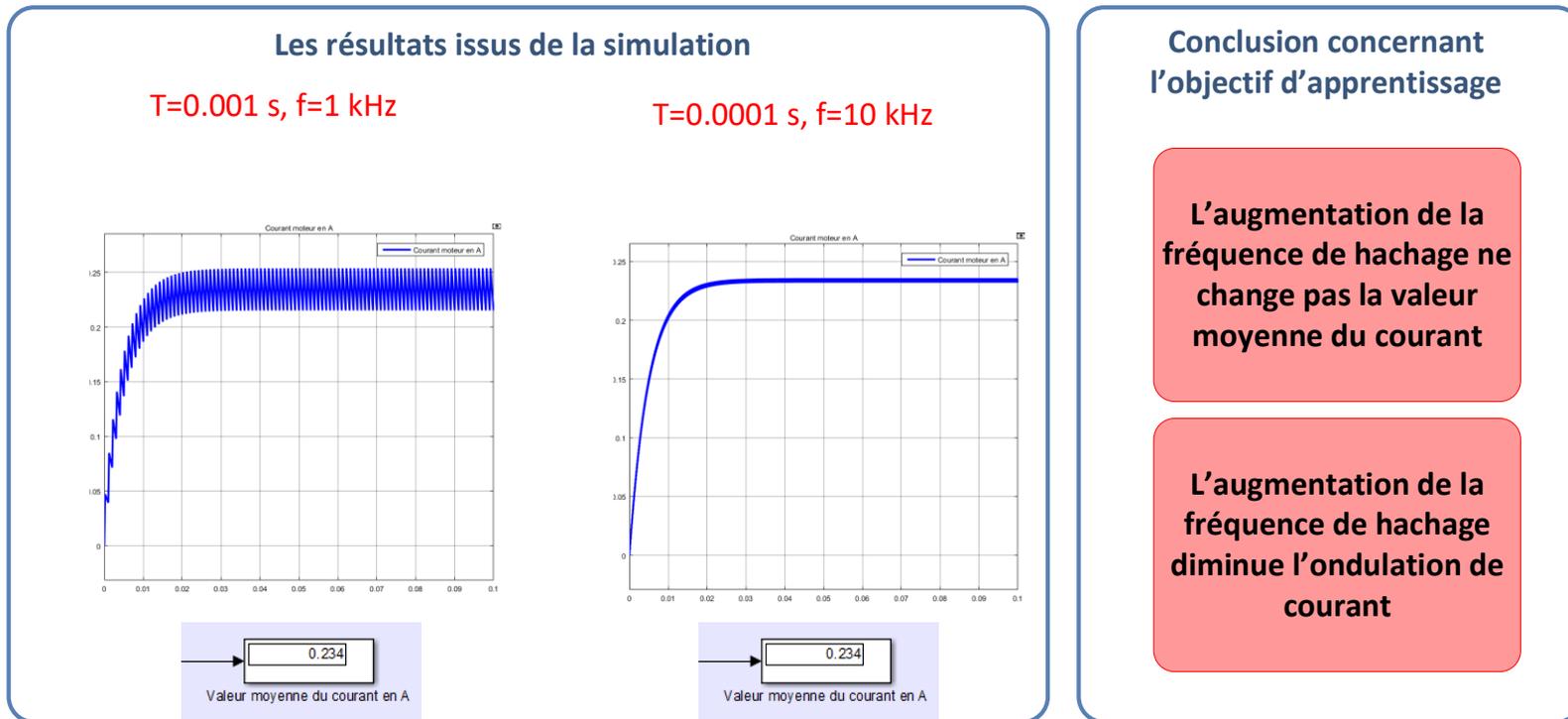


Figure 360 : réglage des paramètres de la simulation

**Lancer** la simulation et relever l'allure instantanée du courant moteur et sa valeur moyenne.

**Relancer** la simulation en modifiant la période de commutation sur **T=0.0001 s**

# Influence de la fréquence de hachage



**Analyse de l'étudiant**

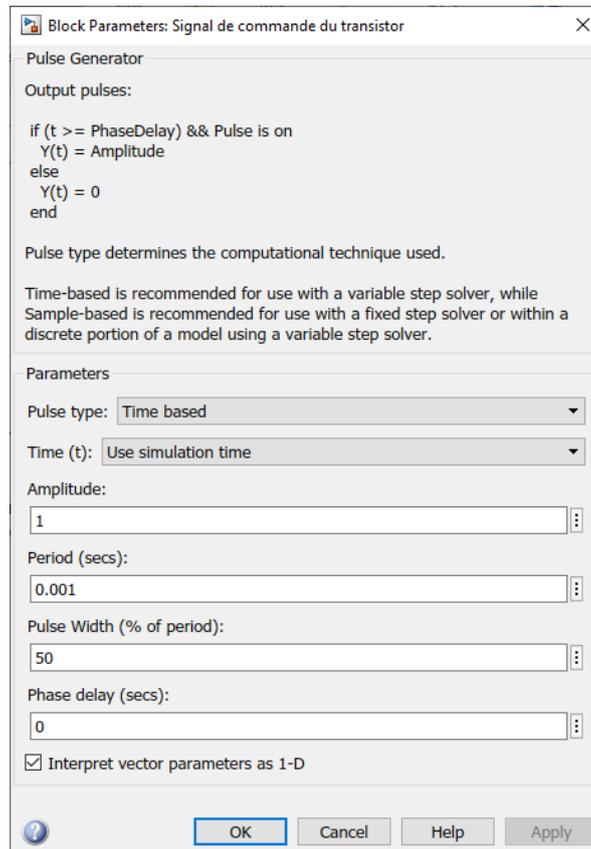


**Figure 361 : exploitation et visualisation de l'influence de la fréquence de hachage sur le courant moteur**

#### 4. Objectif 4 : Visualiser et évaluer l'influence de l'inductance de la charge sur l'ondulation du courant

La simulation du modèle « **hacheur\_serie\_4.slx** » (Figure 364) permet de mettre en évidence l'influence de l'inductance de la charge.

**Paramétrer** le bloc **Pulse Generator** conformément à la Figure 362.



**Figure 362 : paramétrage du bloc Pulse Generator**

**Régler** la valeur de l'inductance ajoutée à **1 mH**

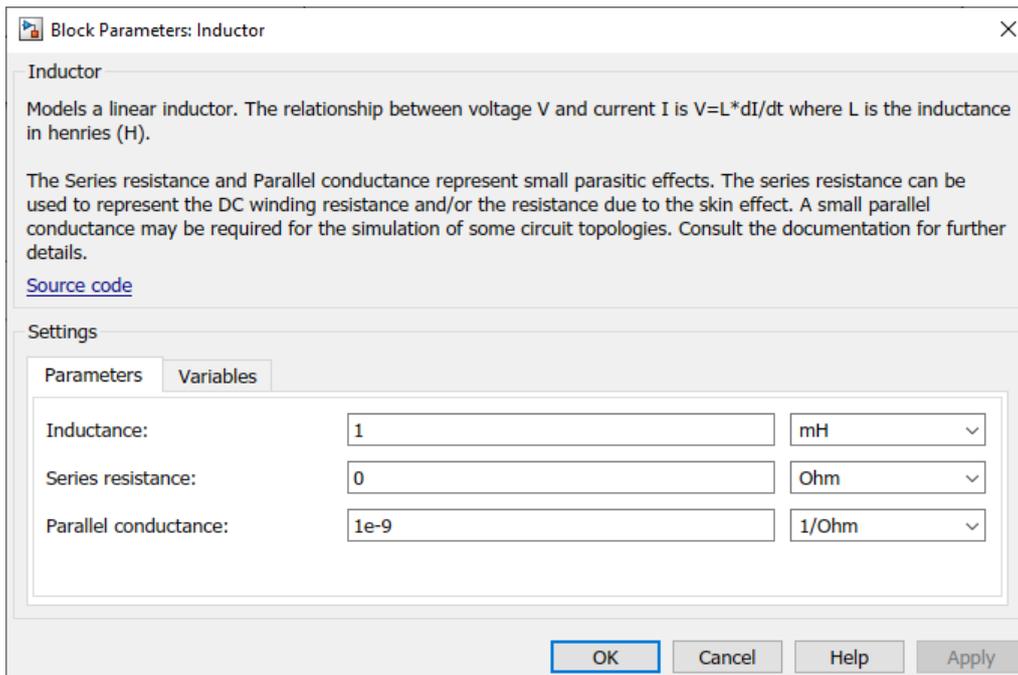


Figure 363 : réglage des paramètres de la simulation

**Lancer** la simulation et relever l'allure instantanée du courant moteur et sa valeur moyenne.  
**Relancer** la simulation en modifiant la valeur de l'inductance à **1000 mH**

## Influence de l'inductance de la charge

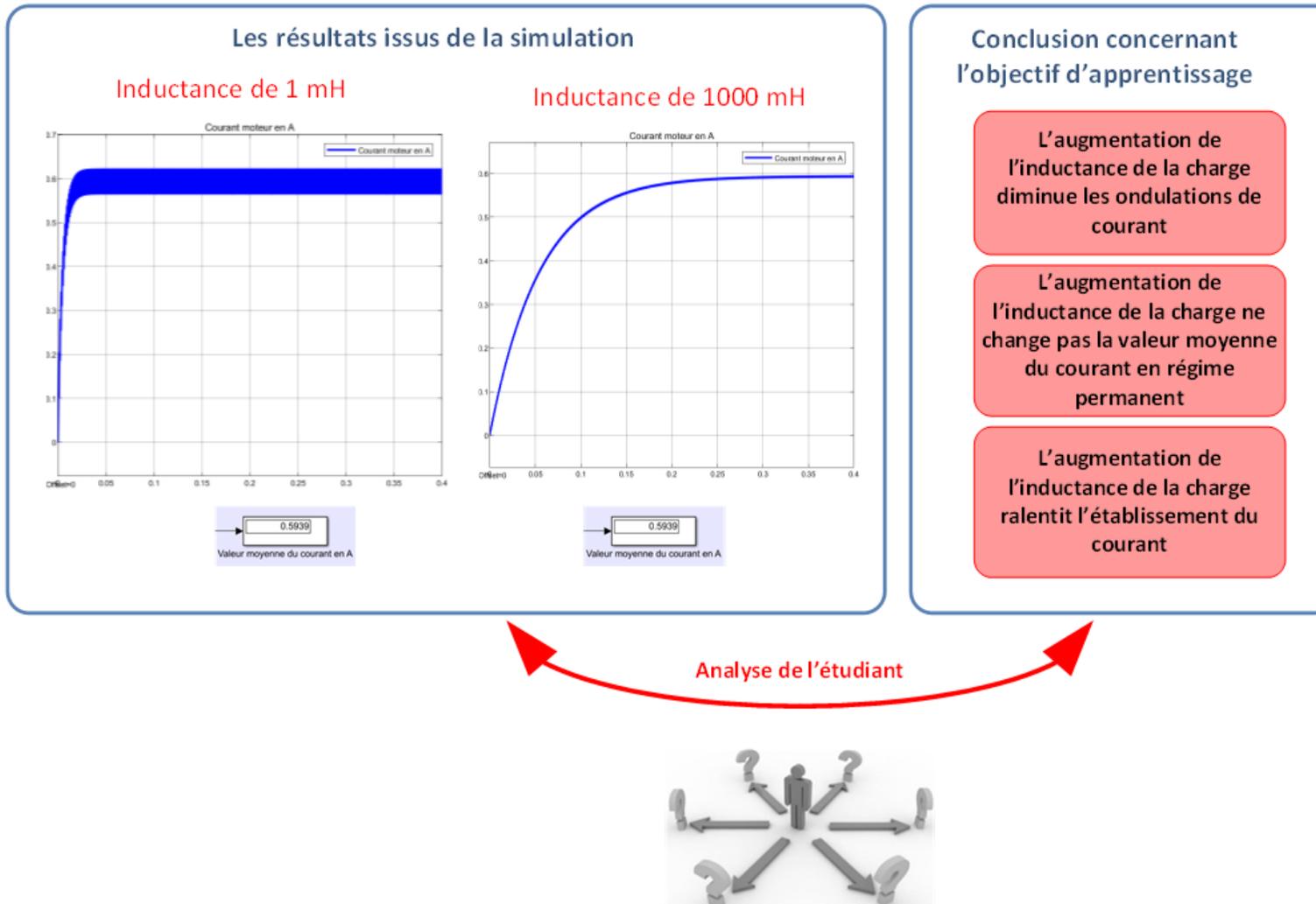


Figure 364 : visualisation et exploitation de l'influence de l'inductance de la charge sur le courant moteur

## F. Conclusion

---

Le succès d'une séquence pédagogique ne se limite pas à la construction des modèles et à leur simulation. Un travail de didactisation doit être effectué sur le modèle. Cette didactisation doit permettre d'optimiser la forme du modèle à l'objectif d'apprentissage visé. Les étudiants sont alors dans les meilleures conditions pour focaliser leur attention sur l'atteinte des objectifs d'apprentissage.

# Chapitre 5 : Prise en main de MATLAB

---

## I. Introduction

---

Ce chapitre a pour objectif de présenter les fonctionnalités de base de **MATLAB** et de manipuler les commandes les plus utiles pour débiter. Les possibilités de calcul de **MATLAB** étant très vastes, il s'agit ici de présenter les commandes de bases qui peuvent être utiles dans le cadre des calculs orientés Sciences de l'Ingénieur.

Toutes les commandes qui suivent seront saisies dans la « fenêtre de commande » **MATLAB**. Chaque ligne de commandes doit être validée par la touche « entrée » du clavier pour être prise en compte par le logiciel.

### A. Création de variable

---

Création de la variable a :

```
>> a=1 (valider)
a =
    1
```

MATLAB fait la différence entre majuscule et minuscule **a** et **A** seront deux variables distinctes. On remarque que MATLAB répond qu'il vient de créer la variable **a**. Cette nouvelle variable apparaît dans la fenêtre du **Workspace** et la commande que l'on vient de saisir apparaît dans la fenêtre **Command History**.

Création de la variable b :

```
>> b=5 ;
```

Le point-virgule à la fin de la ligne de commande indique que l'on ne souhaite pas de réponse à notre commande. La variable b apparaît dans le **Workspace**.

Il est facile de réaliser une opération entre deux variables, par exemple une multiplication.

```
>> a*b
ans =
    5
```

En appuyant sur les flèches haut et bas du clavier, il est possible de rappeler toutes les commandes de la fenêtre **Command History**.

## B. Création de vecteur

---

Considérons la série de données du tableau de Figure 365.

x	0	10	20	30	40	50
y1	0	16	35	49	25	6
y2	0	12	21	36	49	56

Figure 365 : tableau de données

Si l'on souhaite exploiter ces données il faut les saisir dans MATLAB sous la forme de vecteurs.

Le séparateur peut être un espace :

```
>> x=[0 10 20 30 40 50];
```

Le séparateur peut être une virgule :

```
>> y1=[0,16,35,49,25,6];
```

```
>> y2=[0 12 21 36 49 56];
```

Chaque composante  $i$  du vecteur  $x$  est stockée dans la variable  $x(i)$ .

```
>> x(2)
ans =
    10
```

**MATLAB** propose également des fonctions de création automatique de vecteurs

```
>> u=[8 :0.1 :50]
```

Cette commande permet de créer le vecteur  $u$  qui aura comme composantes toutes les valeurs comprises entre **8** et **50** avec un pas de **0.1**. L'utilisation de cette commande permet de spécifier l'espacement entre deux composantes, le nombre de composantes étant calculé en fonction du pas et des bornes spécifiées.

Si le pas n'est pas spécifié, il prend par défaut la valeur de 1.

```
>> u=[8 :50]
```

Cette commande permet de créer le vecteur  $u$  qui aura comme composantes toutes les valeurs comprises entre **8** et **50** avec un pas de **1**.

Si au contraire, on souhaite spécifier le nombre exact de composantes, la fonction *linspace* peut être utilisée.

```
>> u=linspace(8,50,1000)
```

Cette commande permet de créer le vecteur  $u$  qui aura 1000 composantes régulièrement espacées entre les bornes 8 et 50.

## C. Indexation des composantes d'un vecteur

---

Il est possible d'accéder facilement aux composantes d'un vecteur en utilisant l'indexation par ligne.

La variable  $u(i)$  contient la valeur de la  $i^{\text{ème}}$  composante du vecteur  $u$ .

```
>> u(56)
ans =
    10.313
```

## D. Tracés de courbes

Deux vecteurs de même longueur peuvent être tracés à l'aide de la commande *plot* :

```
>> plot(x,y1) trace le vecteur y1 en fonction du vecteur x
```

Une fenêtre graphique « Figure 1 » s'ouvre avec la courbe demandée.

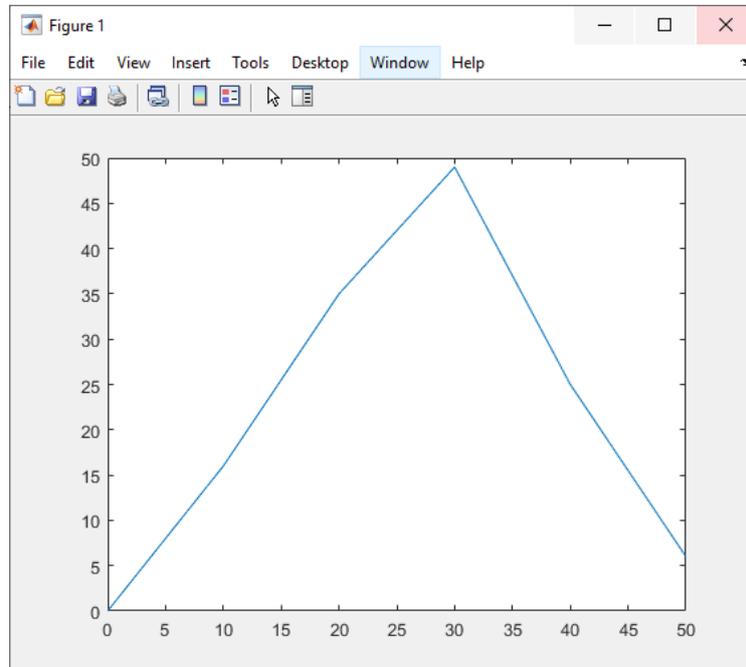


Figure 366 : tracé de deux vecteurs

On peut tracer le vecteur y2 en fonction du vecteur x

```
>> plot(x,y2) trace le vecteur y2 en fonction du vecteur x
```

On remarque que sur la fenêtre « Figure 1 », seule la courbe y2 en fonction de x est affichée. La courbe précédente a été supprimée.

Pour conserver les courbes dans une même fenêtre, on peut utiliser la commande *hold on*.

```
>> hold on  
>> plot(x,y1)
```

La commande *hold off* annule la commande *hold on*.

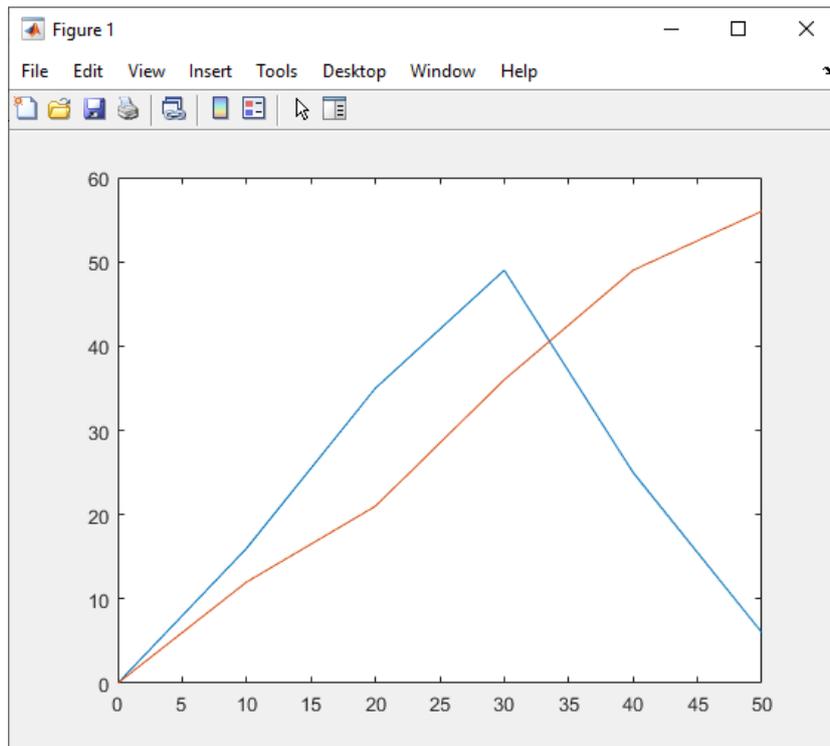


Figure 367 : tracé de deux courbes dans la même fenêtre graphique

La commande *grid on* permet d'afficher la grille

>>grid on

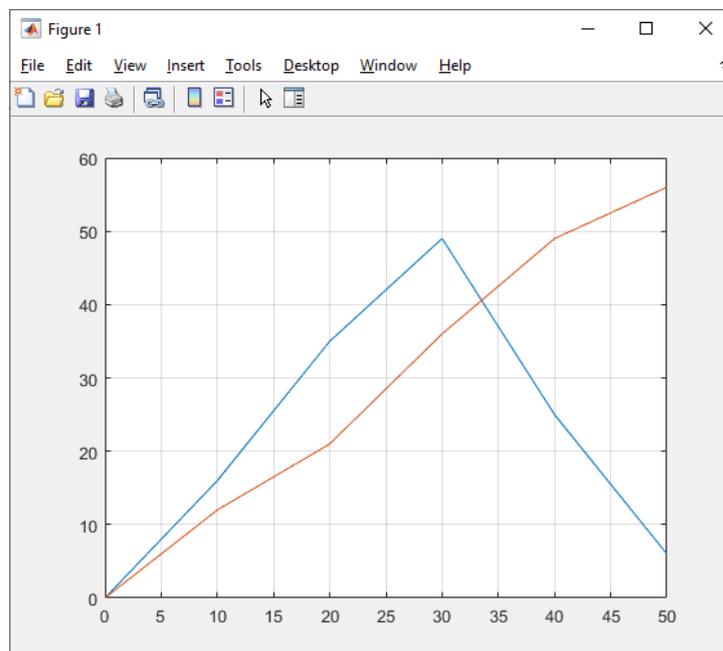


Figure 368 : affichage de la grille sur un graphique

La commande *grid off* permet de masquer la grille.

Si l'on souhaite ouvrir une nouvelle fenêtre graphique, il faut utiliser la commande *figure*

>> figure

Une fenêtre graphique vide nommée figure2 s'ouvre et il est possible travailler dans cette nouvelle fenêtre (Figure 369).

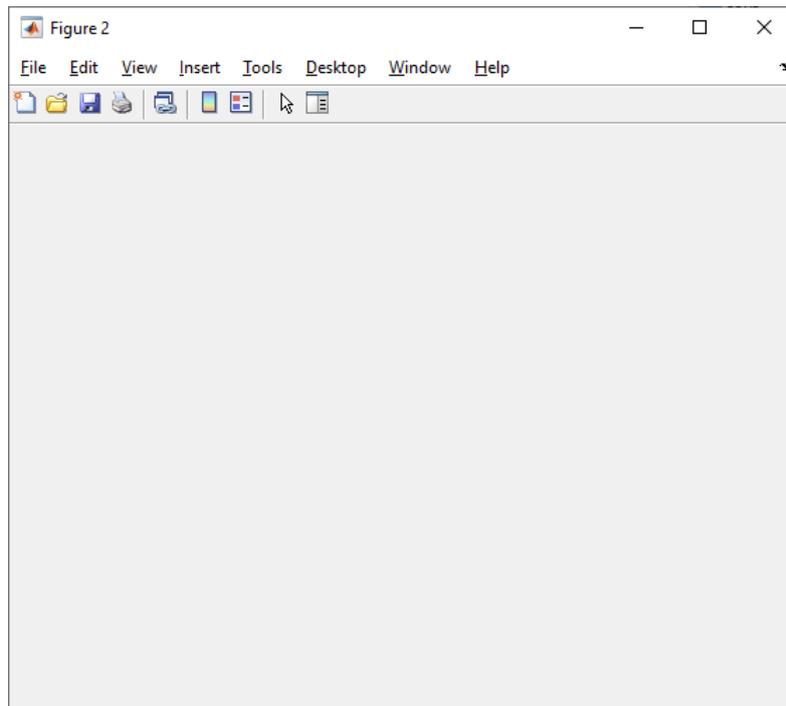


Figure 369 : ouverture d'une nouvelle fenêtre graphique

### E. Mise en forme élémentaires des courbes

Des fonctions de mise en forme des courbes sont disponibles. La fonction *plot* accepte un troisième argument facultatif qui vous permet de spécifier une mise en forme pour les courbes.

Le tableau de la Figure 370 présente les codes que l'on peut saisir pour obtenir la mise en forme des courbes. Ces codes peuvent être combinés dans un ordre quelconque.

Couleur		Marqueur		Style de ligne	
<b>b</b>	Bleu	.	Point	-	Continu
<b>g</b>	Vert	o	Cercle	--	Trait
<b>r</b>	Rouge	x	Croix diagonale	:	Pointillés
<b>c</b>	Cyan	+	Croix verticale	-.	Trait-point
<b>m</b>	Magenta	*	Etoile		<b>Remarque</b> : Si un style de marqueurs est spécifié sans aucun style de ligne aucune ligne ne sera tracée.
<b>y</b>	Jaune	s	Carré		
<b>k</b>	Noir	d	Losange		
<b>w</b>	blanc	v	Triangle vers le bas		
		^	Triangle vers le haut		
		<	Triangle à gauche		
		>	Triangle à droite		
		p	Pentagramme		
		h	hexagramme		

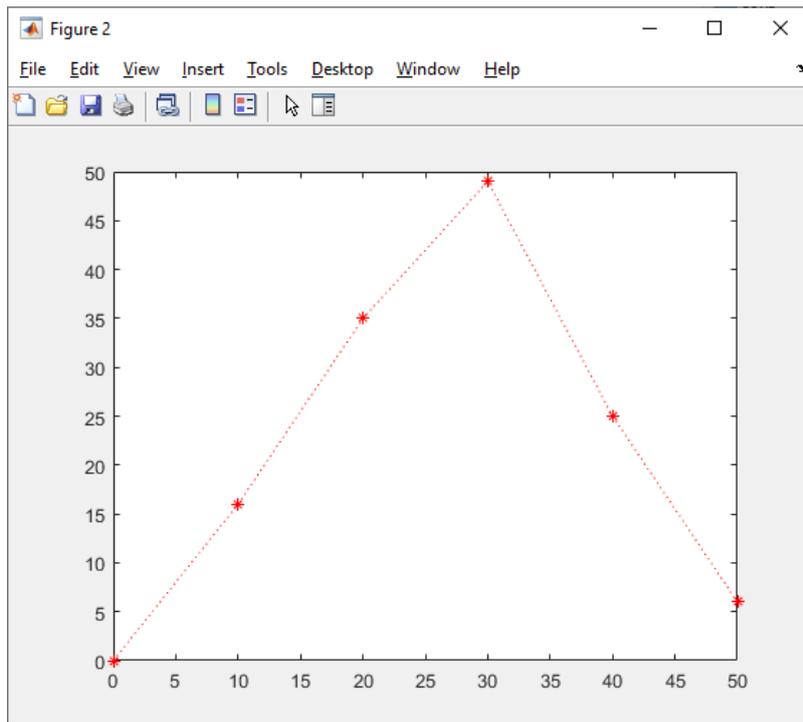
Figure 370 : codes de mise en forme des courbes

Vous pouvez faire quelques essais :

```
>>hold off
```

```
>>plot(x,y1,'r*:')
```

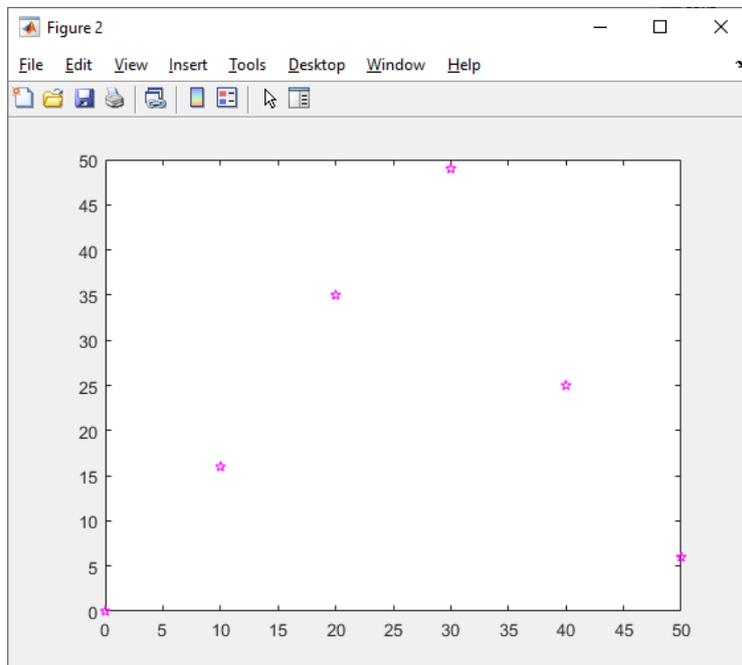
r pour rouge, \* pour les marqueurs et : pour le style de ligne pointillés



**Figure 371 : mise en forme d'une courbe : modification de la couleur et du style de ligne**

```
>>plot(x,y1,'mp')
```

Aucun style de ligne n'est spécifié, ce sont uniquement les marqueurs qui apparaissent.



**Figure 372 : mise en forme d'une courbe : ajout de marqueurs**

Il est également possible de modifier l'épaisseur du trait avec l'argument *LineWidth*

```
>> plot(x,y1,'mo-.','LineWidth',3)
```

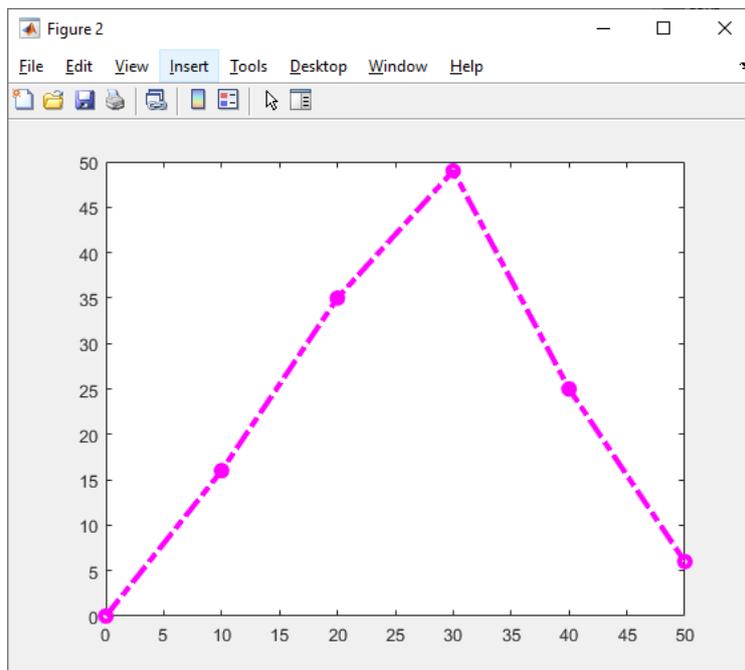


Figure 373 : mise en forme d'une courbe : choix de l'épaisseur de trait

## F. Annotation des graphiques

Il est également possible d'annoter le graphique:

```
>> title('ma première courbe')  
>> xlabel('vecteur x')  
>> ylabel('vecteur y1')  
>> legend('y1 en fonction de x')
```

*affiche un titre au graphique*  
*étiquette de l'axe des abscisses*  
*étiquette de l'axe des ordonnées*  
*ajout d'une légende*

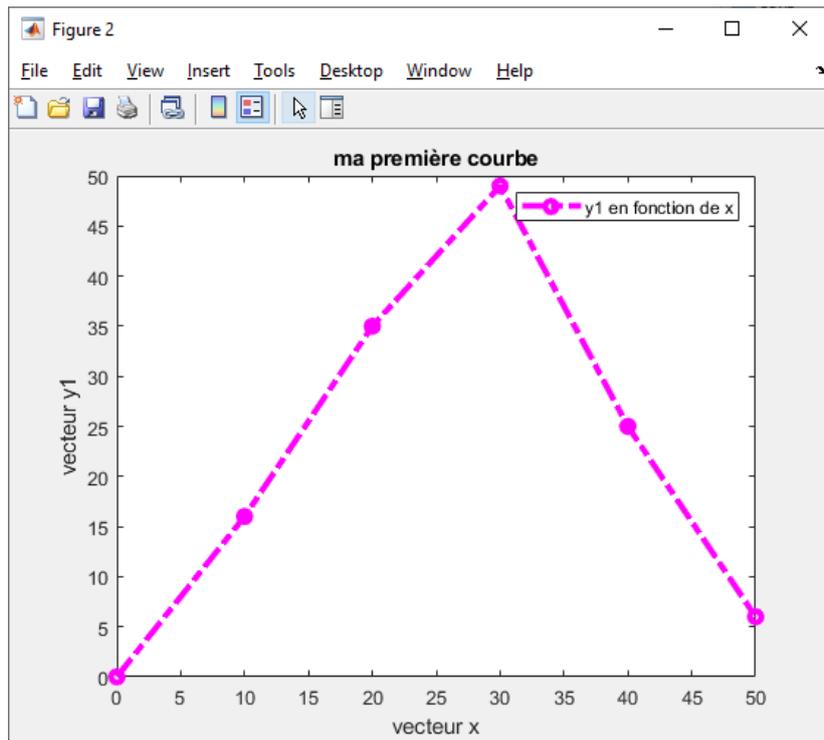


Figure 374 : mise en forme d'une courbe : légende et annotation des axes

La commande `axis` permet de définir manuellement les échelles des axes.

```
axis([0 100 -10 60])
```

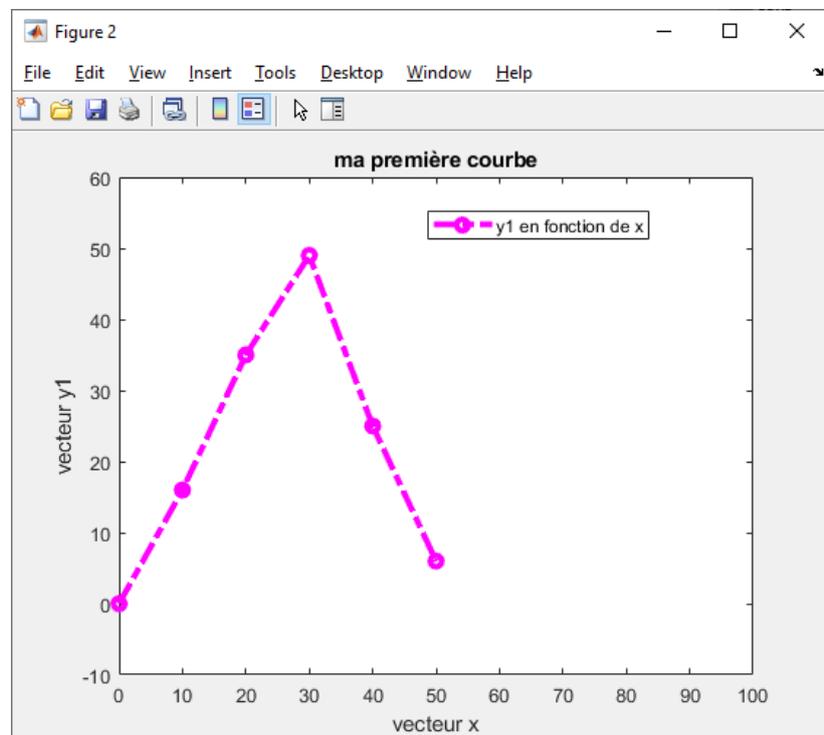


Figure 375 : mise en forme d'une courbe : modification des échelles des axes

## G. Création de matrices

La manipulation des données peut également se faire en stockant des nombres dans des matrices.

Pour créer la variable M contenant la matrice  $M = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$  :

```
>> M = [1 2 3 ; 4 5 6 ; 7 8 9]
```

M =

```
1 2 3
4 5 6
7 8 9
```

Pour accéder à un élément de la matrice on utilise le système d'indexation  $M(n_{\text{ligne}}, n_{\text{colonne}})$ . Les lignes et les colonnes sont numérotées de 1 à n (Figure 376).

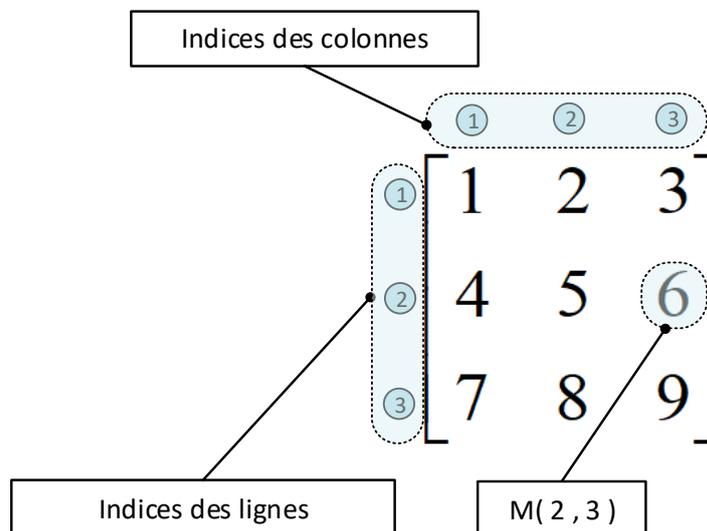


Figure 376 : indexation des lignes et des colonnes d'une matrice

```
>> M(2,3)
```

ans =

```
6
```

```
>> M(1,3)
```

ans =

```
3
```

Il est également possible d'accéder à un élément d'une matrice en utilisant une indexation à un seul indice. Les éléments sont numérotés en partant du premier élément de la première ligne qui porte l'indice 1. Les indices sont ensuite incrémentés en se déplaçant du haut vers le bas dans la colonne. Arrivé en bas

de colonne, l'indexation se poursuit en partant du premier élément en haut de la colonne suivante, et ainsi de suite (Figure 377).

$$M = \begin{bmatrix} \textcircled{1} 1 & \textcircled{4} 2 & \textcircled{7} 3 \\ \textcircled{2} 4 & \textcircled{5} 5 & \textcircled{8} 6 \\ \textcircled{3} 7 & \textcircled{6} 8 & \textcircled{9} 9 \end{bmatrix}$$

Figure 377 : indexation des éléments d'une matrice avec un indice unique

```
>> M(1)
```

```
ans =
```

```
1
```

```
>> M(6)
```

```
ans =
```

```
8
```

```
>> M(2)
```

```
ans =
```

```
4
```

Il est ainsi possible d'affecter une valeur à un élément de la matrice de plusieurs manières.

```
>> M(2,3)=666
```

```
M =
```

```
1 2 3  
4 5 666  
7 8 9
```

```
>> M(4)=777
```

```
M =
```

```
1 777 3  
4 5 666  
7 8 9
```

Si on affecte une valeur à un élément dont l'indice est supérieur à la taille de la matrice, MATLAB augmente automatiquement la matrice afin de lui donner la bonne dimension en ajoutant des 0.

```
>> M(5,5)=888
```

M =

```
1 777 3 0 0
4 5 666 0 0
7 8 9 0 0
0 0 0 0 0
0 0 0 0 888
```

Cette fonctionnalité s'applique également aux vecteurs. Ici le vecteur T n'est pas défini dans le Workspace

```
>> T(6) = 0
```

T =

```
0 0 0 0 0 12
```

Il est possible d'extraire des parties de la matrice en utilisant la notation suivante :  $A = M(2:4, 1:3)$ . Cela signifie que la matrice A sera constituée des lignes 2 à 4 et des colonnes 1 à 3 de la matrice M (Figure 378).

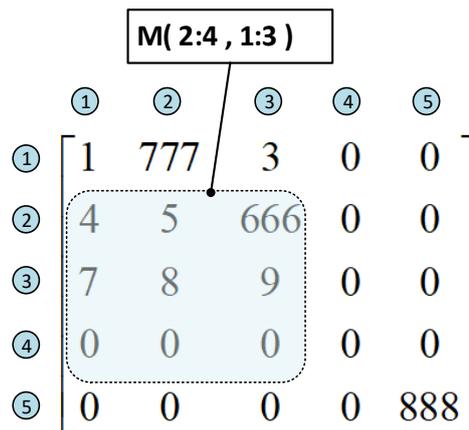


Figure 378 : extraction d'une matrice

```
>> A = M(2:4,1:3)
```

A =

```
4 5 666
7 8 9
0 0 0
```

Il est également possible d'extraire simplement des lignes ou des colonnes entières en utilisant la notation « : » qui signifie « **tous les indices de la ligne ou de la colonne** » (Figure 379).

$M(:, 3)$  permettra d'extraire tous les éléments de la troisième colonne.

$M(2, :)$  permettra d'extraire tous les éléments de la deuxième ligne.

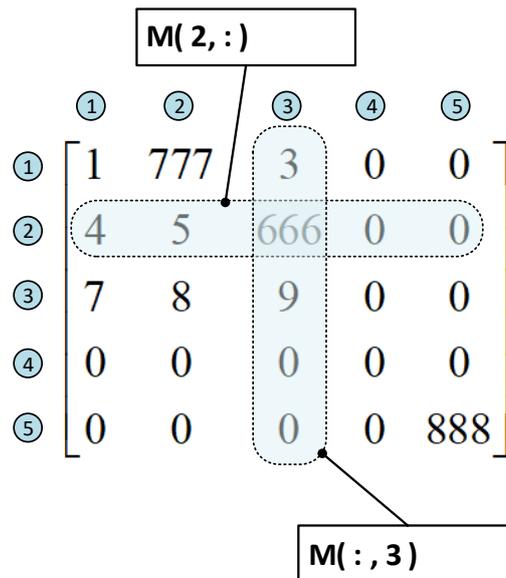


Figure 379 : extraction d'une ligne entière ou d'une colonne entière

```
>> M(:,3)
ans =
    3
   666
    9
    0
    0

>> M(2,:)
ans =
    4    5   666    0    0
```

Pour obtenir la transposée d'une matrice on utilise le signe « ' »

```
>> M'
ans =
    1    4    7    0    0
   777    5    8    0    0
    3   666    9    0    0
    0    0    0    0    0
    0    0    0    0   888

>> C = M(:,2)
C =
   777
```

```

5
8
0
0

>> C'

ans =

777 5 8 0 0

```

Pour créer des matrices des 0, on utilise la fonction **zeros**.

```

>> B = zeros(4,4)

B =

0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0

```

Pour créer des matrices de 1, on utilise la fonction **ones**.

```

>> D = ones (4,4)

D =

1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1

```

Pour créer une matrice identité on utilise la fonction **eye**.

```

>> E = eye(5)

E =

1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

```

Pour créer une matrice avec des nombres choisis de manière aléatoire on utilise la fonction **rand**.

```

>> rand(3,3)

ans =

0.8147 0.9134 0.2785
0.9058 0.6324 0.5469
0.1270 0.0975 0.9575

```