

# SIMULATION SYSTÈME

**Date :** 19 décembre 2022

**Auteur(s) :** Benoît Souyris

**Copyright :** B. Souyris, université de technologie de Compiègne

**Licence :** CC 4.0 BY-NC-SA [<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.fr>] + licence commerciale ET-LIOS [<https://et-lios.s-mart.fr/licencecommerciale/>]

# Table des matières

1. Introduction à la simulation multi-physique .....	4
1.1. Introduction	4
1.2. Enjeux industriels	4
1.3. Situation de la simulation multi-physique dans le cycle de vie du produit	8
1.4. Objectifs de la simulation	9
1.5. Exemple d'application : ascenseur grande vitesse	9
1.6. Liens avec l'ingénierie système	12
1.7. Architecture d'un système multi-physique	19
1.7.1. Notion de diagramme fonctionnel : la chaîne d'information et la chaîne d'énergie	19
1.7.2. Composition des blocs fonctionnels	22
1.7.3. Assemblage des blocs	24
1.7.4. Architecture mixte	26
1.8. Notion de paramètres et de variables	28
1.9. Notion de domaines d'études	29
1.10. Notion de couplages multi-physiques et de solveurs	29
1.11. Performances et optimisation du système : Simulée-Mesurée-Souhaitée	32
1.12. Utilisation des jumeaux numériques	34
1.13. Ce qu'il faut retenir	35
2. Modélisation par schéma blocs .....	37
2.1. Introduction	37
2.2. Concepts fondamentaux	37
2.2.1. Modélisation système multi-physique	37
2.2.2. Mise en équation du système	39
2.2.3. Notion de <b>fonction de transfert</b>	41
2.2.4. Notion de <b>causalité</b>	42
2.2.5. Notion de lien entre les blocs	43
2.2.6. Schémas blocs	46
2.3. Quelques modèles d'éléments simples	47
2.3.1. Quelques modèles <b>causals</b> d'éléments <b>passifs</b>	48
2.3.2. Quelques modèles <b>causals</b> d'éléments de <b>jonction</b>	51
2.4. Méthode de modélisation par schéma blocs, exemple d'une pompe de maintien	54
2.4.1. Introduction	54
2.4.2. Présentation d'une pompe de maintien de niveau	54
2.4.3. Modélisation d'une pompe de relevage	55
2.4.3.1. Modéliser chaque bloc	55
2.4.3.2. Modéliser le réservoir 2	56
2.4.3.3. Modéliser la pompe	57
2.4.3.4. Modéliser le réducteur	58
2.4.3.5. Modéliser le moteur électrique	60
2.4.3.6. Synthèse	62
2.4.4. Méthode de <b>modélisation par schéma blocs</b>	63
2.5. Ce qu'il faut retenir	66

3. Modélisation orientée objet .....	67
3.1. Introduction	67
3.2. Exemple d'application : ascenseur grande vitesse	67
3.3. Etude proposée : vérifier la performance de la cabine d'ascenseur	70
3.4. Mise en place d'un environnement DYMOLA dans 3DEXperience	71
3.4.1. Les étapes de conception avec 3DEXperience	71
3.4.2. Configuration de Dymola	72
3.4.3. Créer le cadre de l'étude : espace logique de référence	74
3.4.4. Créer un modèle Dymola	76
3.5. Modélisation avec DYMOLA	78
3.5.1. Comment se présente un modèle sur Dymola ?	78
3.5.2. Les liens par domaine	78
3.5.3. Equations de liaison	79
3.5.4. Structure d'un composant à deux ports (ou bornes)	80
3.5.5. Paramétrage interne des composants	81
3.5.6. Comportement d'un composant	82
3.5.7. Couplage entre composants élémentaires	83
3.5.8. Regroupement de composants	84
3.6. Application : étude de la cabine d'ascenseur	86
3.6.1. Construction du système simplifié pour prédimensionnement	86
3.6.2. Construction du système complexe	89
3.6.3. Résultats de simulation	94
Synthèse .....	98
Glossaire .....	99

# 1. Introduction à la simulation multi-physique

## 1.1. Introduction

### Objectifs pédagogiques

L'objectif de ce module est de présenter le besoin industriel ainsi que les étapes et méthodes permettant de mener à bien l'étude comportementale globale d'un système complexe multi-physique.

### Présentation rapide des différentes étapes

- **Présentation des enjeux industriels** : la nécessité de performance des systèmes complexes multi-domaines et les problématiques de coût de développement ont contraint les industriels à revoir leur approche qui, par le passé, était plutôt cloisonnée, afin de favoriser une approche holistique <sup>[p.99]</sup> ;
- **Objectifs de la simulation système** : les objectifs sont multiples allant de **valider** à **optimiser** une conception ;
- **Situation de la simulation système dans le cycle de vie du produit** : elle sera utilisée dans plusieurs étapes du cycle de vie du produit ;
- **Lien avec l'Ingénierie Système** : l'ingénierie système utilise des diagrammes permettant de faire des choix de conception. Certains diagrammes permettront de construire une maquette numérique ayant pour but de valider la conception ;
- **Architecture d'un modèle multi-physique** : l'architecture d'un modèle multi-physique dépend de l'objectif que l'on s'est fixé ;
- **Notion de domaine et de réseau de composants** : chaque système est constitué d'un réseau de composants ;
- **Notion de paramètres et de variables** : il est important de différencier paramètres et variables ;
- **Performances des systèmes** : on abordera les notions de **solveurs**.

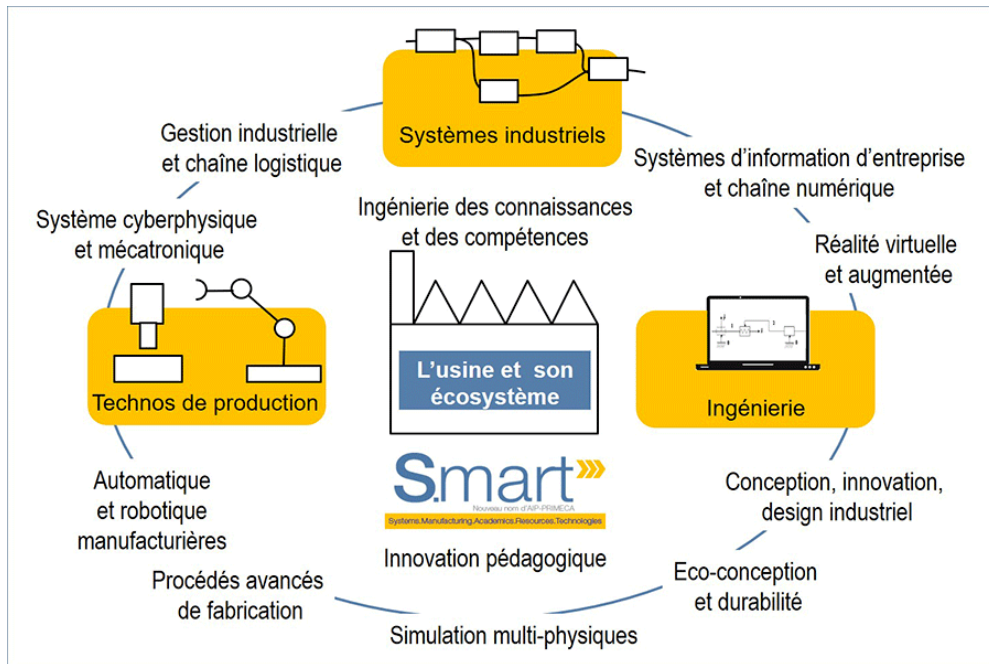
## 1.2. Enjeux industriels

### Problématique des coûts de fabrication et du temps de développement des prototypes

Les exigences des cahiers des charges de plus en plus complexes ainsi que les coûts de production de plus en plus élevés, obligent les entreprises à innover en permanence.

Créer un nouveau produit devient de plus en plus risqué. Il est donc nécessaire de tester la conception globale d'un système avant même de lancer la fabrication d'un prototype, puis sa production à grande

échelle.



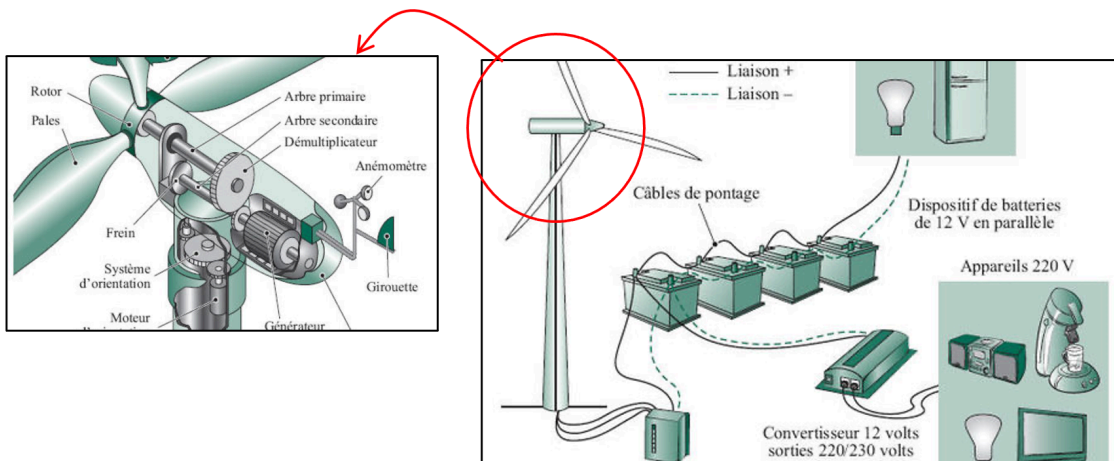
Vision holistique de l'Industrie du Futur

Pour répondre à ces problématiques, les départements R&D des entreprises se tournent de plus en plus vers la simulation système pour des raisons de coût et de temps de développement.

## Des systèmes toujours plus complexes

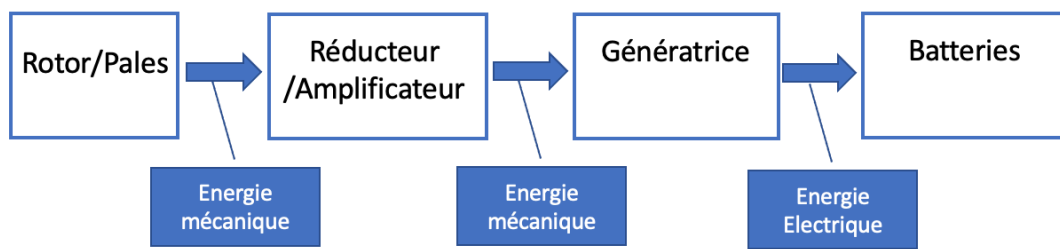
### Exemple Les éoliennes

Les éoliennes sont des exemples de systèmes où les moindres pertes énergétiques peuvent altérer fortement la production. Il est donc primordial d'avoir correctement caractérisé chaque élément qui les compose pour que l'exploitation soit rentable.



Mécanisme d'une éolienne

Chaîne d'énergie :



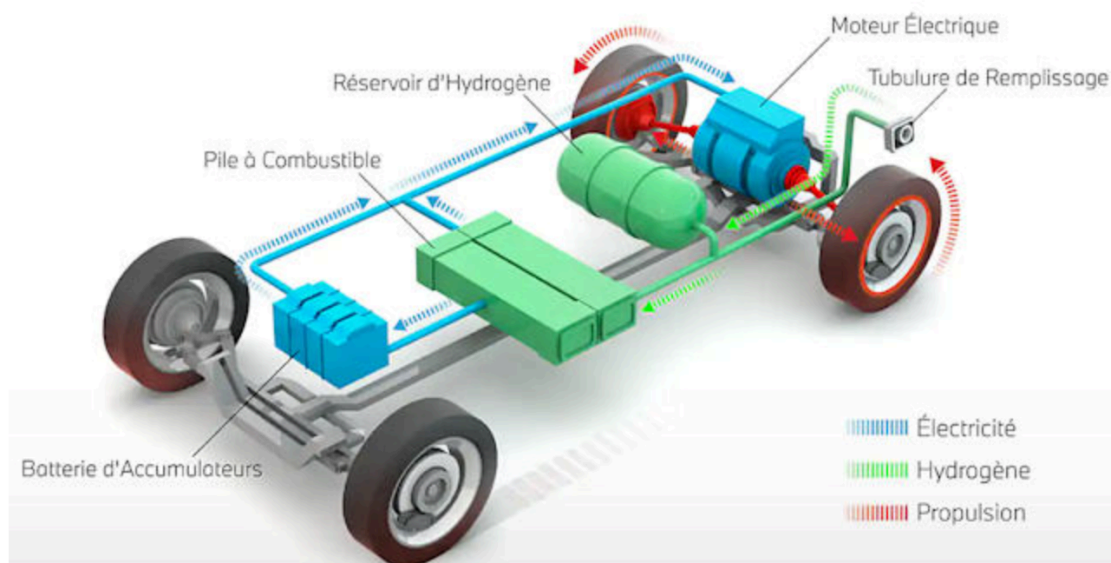
Chaîne d'énergie d'une éolienne

### Exemple L'automobile hybride

Aujourd'hui, une voiture à hydrogène est une voiture électrique qui puise son énergie d'une pile à combustible alimentée avec ce gaz liquéfié. La question de performance énergétique est d'une importance primordiale car elle touche à l'autonomie du véhicule. Anticiper les performances du véhicule devient crucial compte tenu des coûts de fabrication d'un prototype.

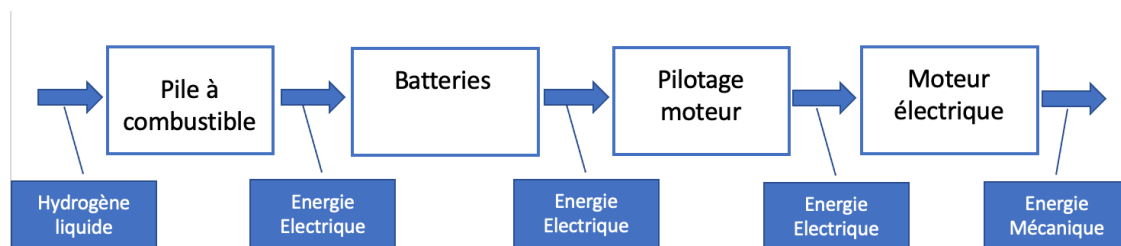
La difficulté d'un tel système est qu'il regroupe différents domaines tels que la chimie, l'électricité et la mécanique.

Pour prévoir son comportement, il faut donc être spécialiste dans tous ces domaines.



Fonctionnement d'une voiture à hydrogène

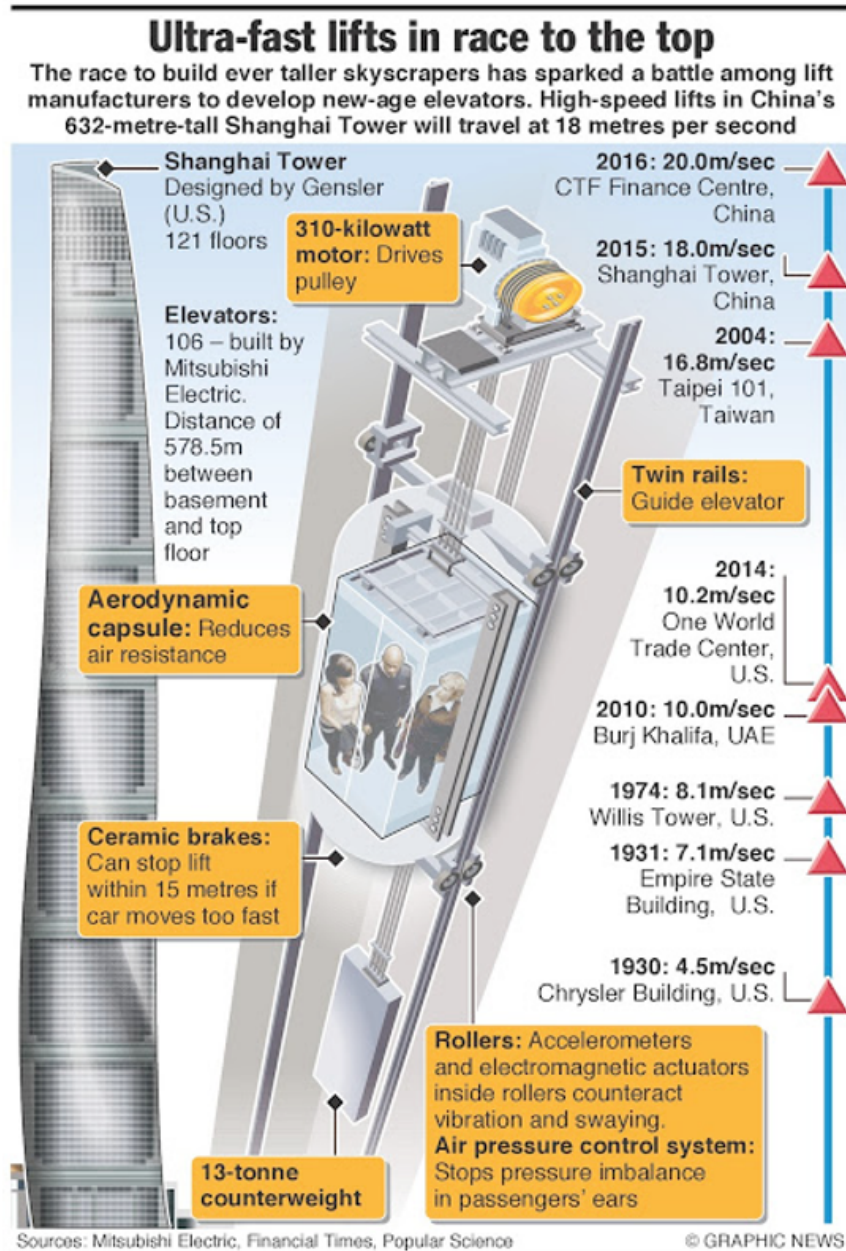
Chaîne d'énergie :



Chaîne d'énergie d'une voiture à hybride à hydrogène

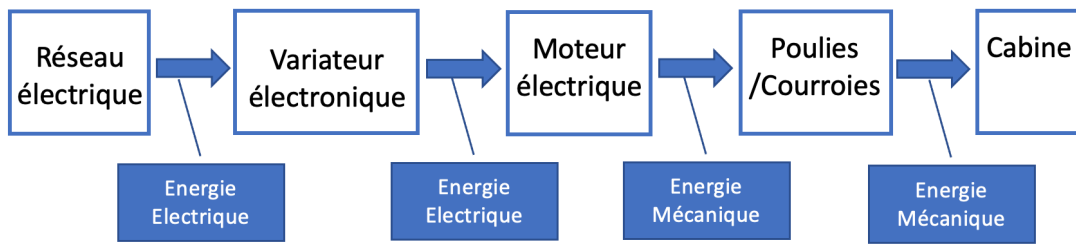
## Exemple L'ascenseur

L'étude des performances des ascenseurs est aussi très complexe, d'autant plus s'il s'agit d'ascenseurs grande vitesse pour lesquels des problématiques de mécanique des fluides ou d'échanges thermiques peuvent venir perturber le comportement jusqu'ici classique des ascenseurs d'ancienne génération.



Ascenseur ultra haute vitesse

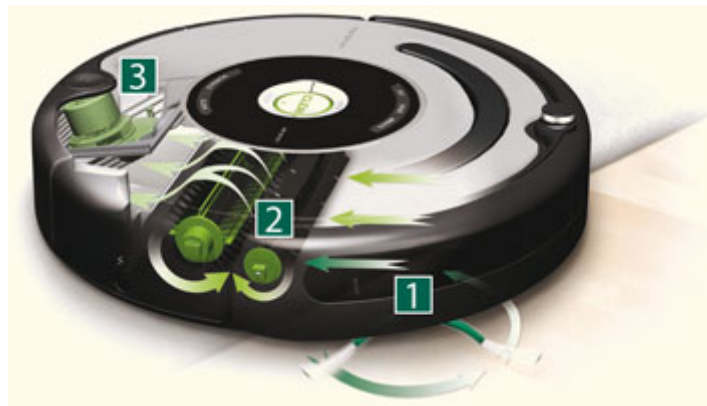
Chaîne d'énergie :



Chaîne d'énergie d'un ascenseur

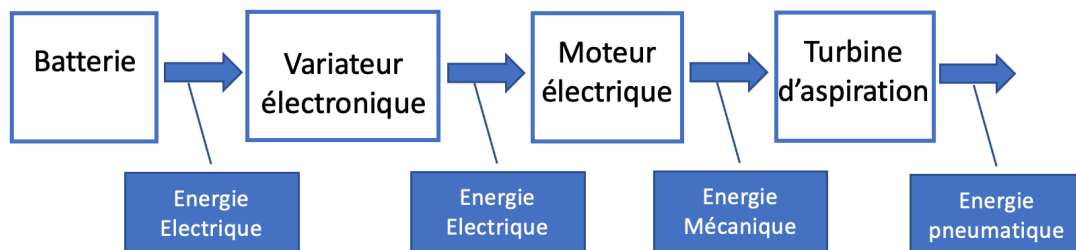
### Exemple L'électroménager

Les robots aspirateurs sont un bon exemple d'évolution, réduisant les corvées ménagères et devenant capable de nettoyer les sols d'une habitation en complète autonomie.



Fonctionnement des aspirateurs Roomba

Chaîne d'énergie :

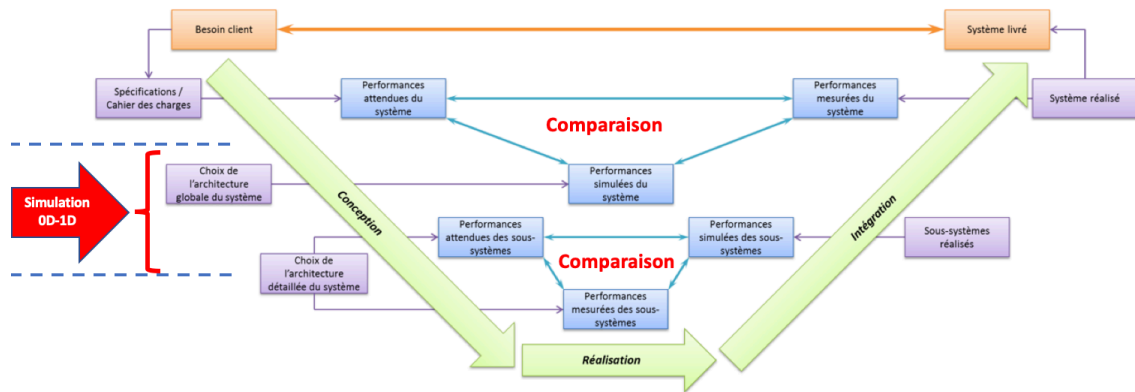


Chaîne d'énergie d'un aspirateur

## 1.3. Situation de la simulation multi-physique dans le cycle de vie du produit



## Tester un concept



La simulation de systèmes multi-physique permet donc de simuler un système complet avant de réaliser un prototype

### Affiner la précision du système

Un fois le prototype construit, il sera tout de même intéressant de comparer les performances du prototype avec le système simulé.

## 1.4. Objectifs de la simulation

1. **Valider les choix d'architecture** : tester les performances globales d'un système lorsque les composants sont en réseau ;
2. **Valider la conception et autoriser le prototypage** : lorsque la phase de conception est validée, on peut passer au prototypage afin de tester les performances réelles du système ;
3. **Comparer le comportement du système simulé par rapport au système réel** : en utilisant un jumeau numérique, on peut vérifier la caractérisation de chaque composant afin de valider la maquette numérique en comparant les performances du réel avec les résultats de la simulation ;
4. **Remplacer le banc de test réel** : lorsque le banc de test n'est pas accessible, on peut se fier à la simulation qui peut se faire sur un poste informatique ;
5. **Optimiser les performances du système** : affiner les caractéristiques de chaque composant afin de mieux traduire son comportement ;
6. **Réduire le temps et le coût de production** : utiliser l'architecture système et le cahier des charges pour concevoir plus rapidement des systèmes multi-physiques.

## 1.5. Exemple d'application : ascenseur grande vitesse

## Contexte

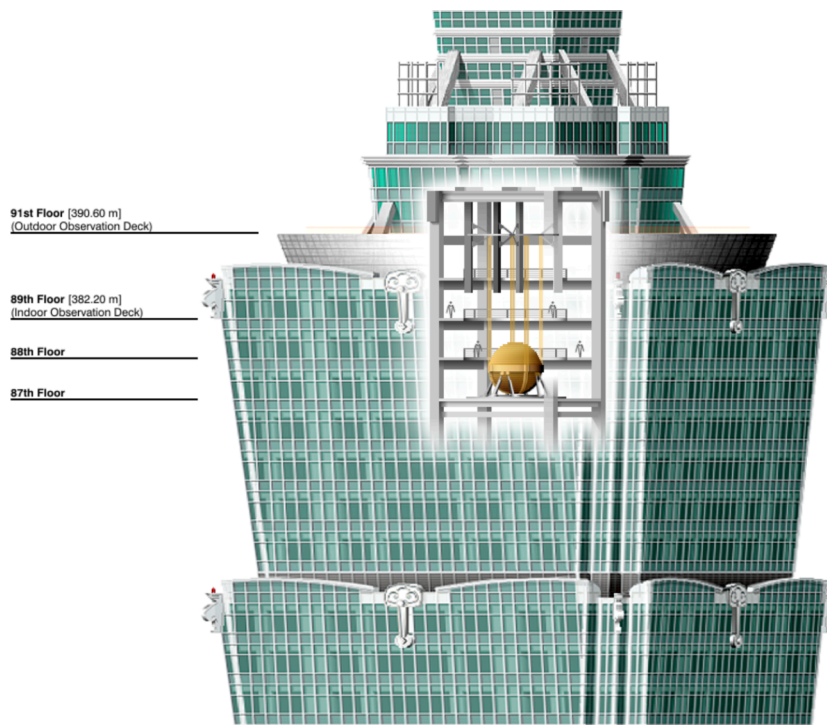
Le Taipei 101 est le premier immeuble à dépasser le demi kilomètre de hauteur (508 mètres).



*Tour Taipei 101*

Du fait de ce record de hauteur :

- La tour a besoin d'un **dispositif de stabilisation massif** pour faire face aux vents et aux séismes. Celui-ci prend la forme d'une énorme boule d'acier de 800 tonnes, accrochée en haut de l'immeuble et visible du restaurant et du bar, qui encerclent l'espace autour du dispositif.



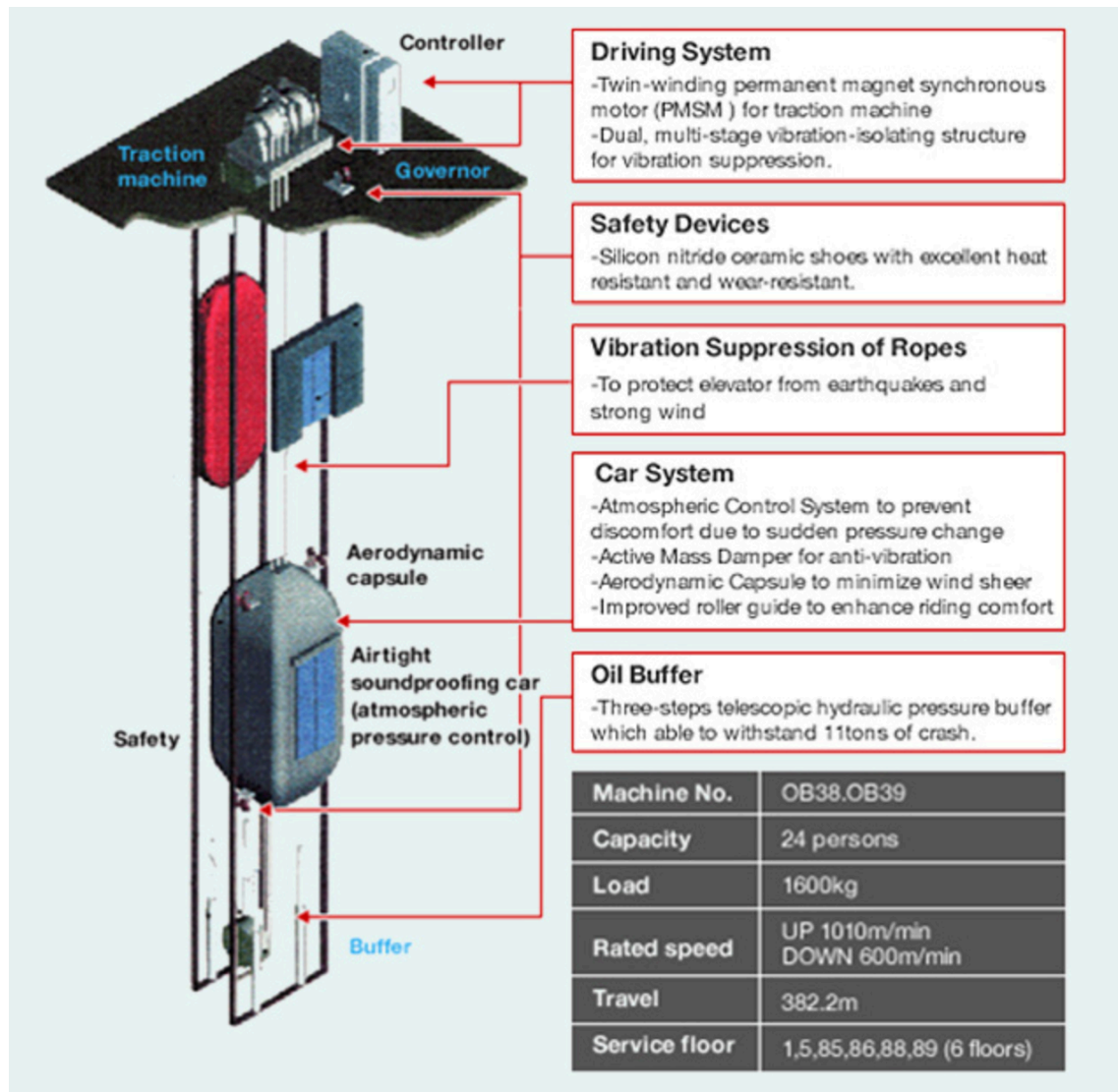
*Taipei 101: Une pendule comme amortisseur harmonique dans la tête de la tour*

- On a aussi besoin d'**ascenseurs performants** pour assurer un déplacement rapide des personnes du rez-de-chaussée au sommet en garantissant un confort optimal.

### Performances attendues des ascenseurs pour l'observatoire

Les ascenseurs desservant les 101 étages du Taipei 101 doivent atteindre une vitesse maxi en montée de 1010 mètres par minute (60.60 km/h) et descendant à 610 mètres par minute (36.60 km/h).

Ces performances sont limitées par des normes d'accélération (environ  $< 1\text{m/s}^2$ ), de pressions acoustiques et de vibrations maximales autorisées.



Ascenseur ultra haute vitesse Toshiba, installé au sein de la tour Taipei 101

Chaque ascenseur doit transporter jusqu'à 24 personnes du rez-de-chaussée au 101<sup>ème</sup> étage.

### ⚠ Attention

Les objectifs d'études peuvent être multiples.

En effet, selon si on veut dimensionner le système de levage ou le système de mise en pression de l'intérieur de la cabine, le modèle sera totalement différent.

L'objectif de cette simulation système est de construire un modèle numérique permettant de simuler le comportement dynamique du système de levage.

## 1.6. Liens avec l'ingénierie système

Pour construire le modèle simulé, on pourra utiliser certains diagrammes de l'ingénierie système :

- Le **diagramme des exigences** permet d'identifier les données et les objectifs du cahier des charges.
- Les **diagrammes d'état** et **de séquence** permettent de construire et valider la logique de commande.
- Le **diagramme de blocs internes** permet construire et valider la chaîne d'énergie.

## Diagramme d'exigences

≈ req

≈ requirement diagram

Les diagrammes d'exigences servent à préciser ce que l'on attend du système, généralement dérivées du cahier des charges.

### Spécificités

Il fait le bilan des attentes du cahier de charge en faisant la liste des critères et niveaux à respecter.

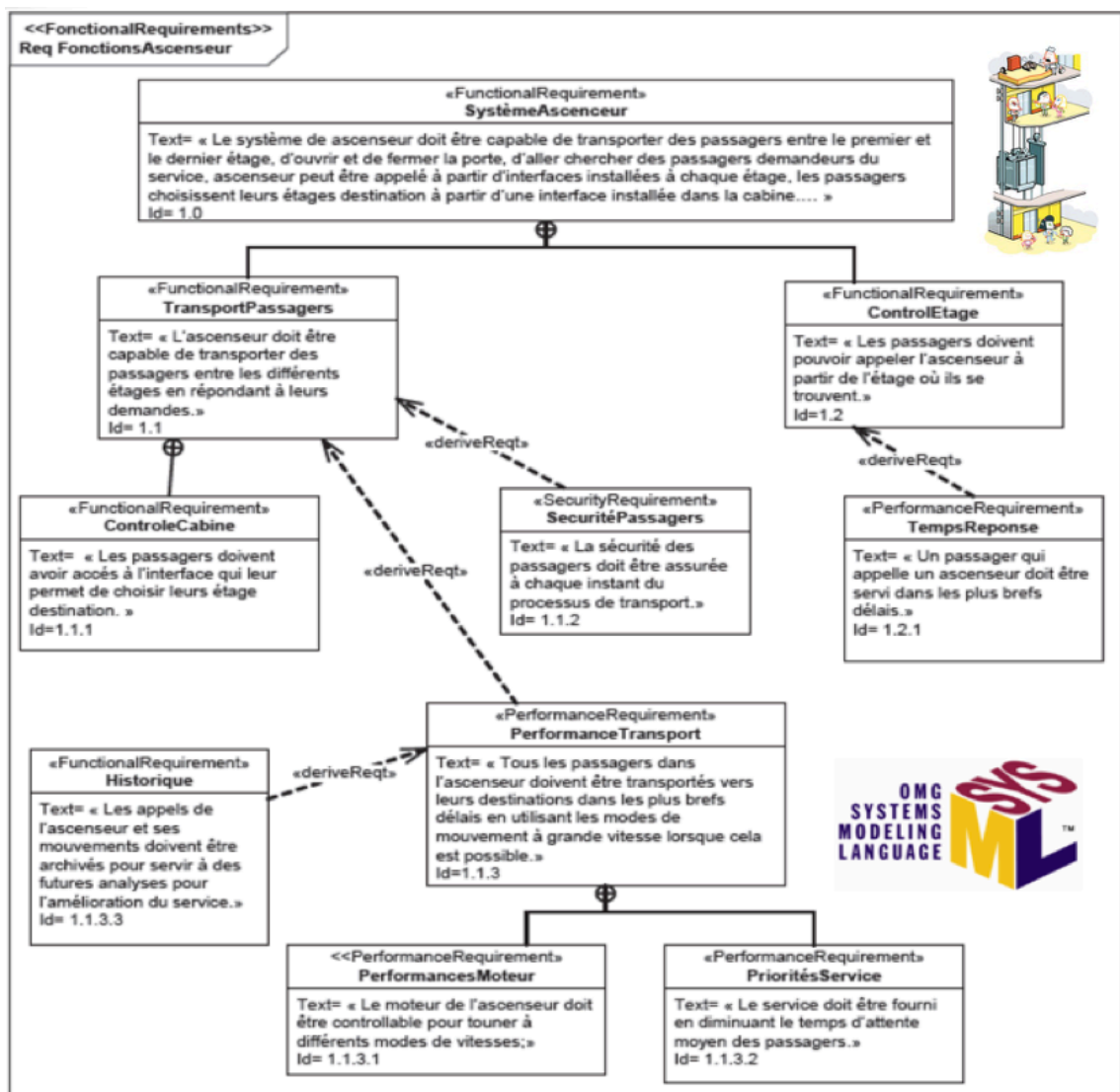


Diagramme d'exigences d'une cabine d'ascenseur

## Diagramme de machine à état

≈ *state machine diagram*

Le diagramme de machine à état décrit les transitions d'état et les actions qu'un système ou ses parties effectuent en réponse à des événements.

### Spécificités

Ce type de diagramme permet de traduire la commande des systèmes séquentiels.

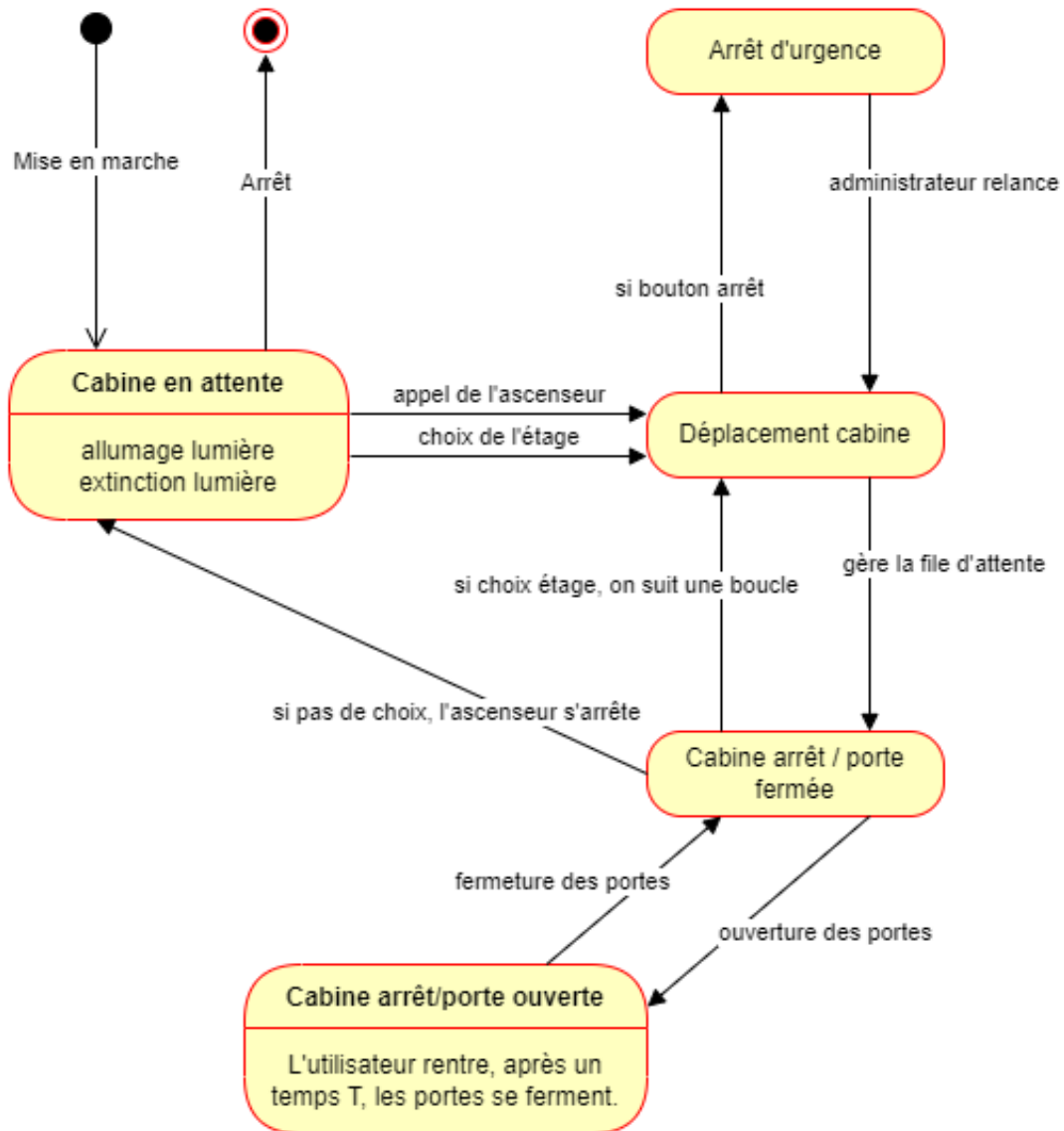


Diagramme d'état d'une cabine d'ascenseur

## Diagramme de séquences

≈ *sequence diagram*

Le diagramme de séquences représente l'interaction entre les parties collaboratrices d'un système.

## Spécificités

Ils complètent le diagramme d'état en donnant une notion temporelle.

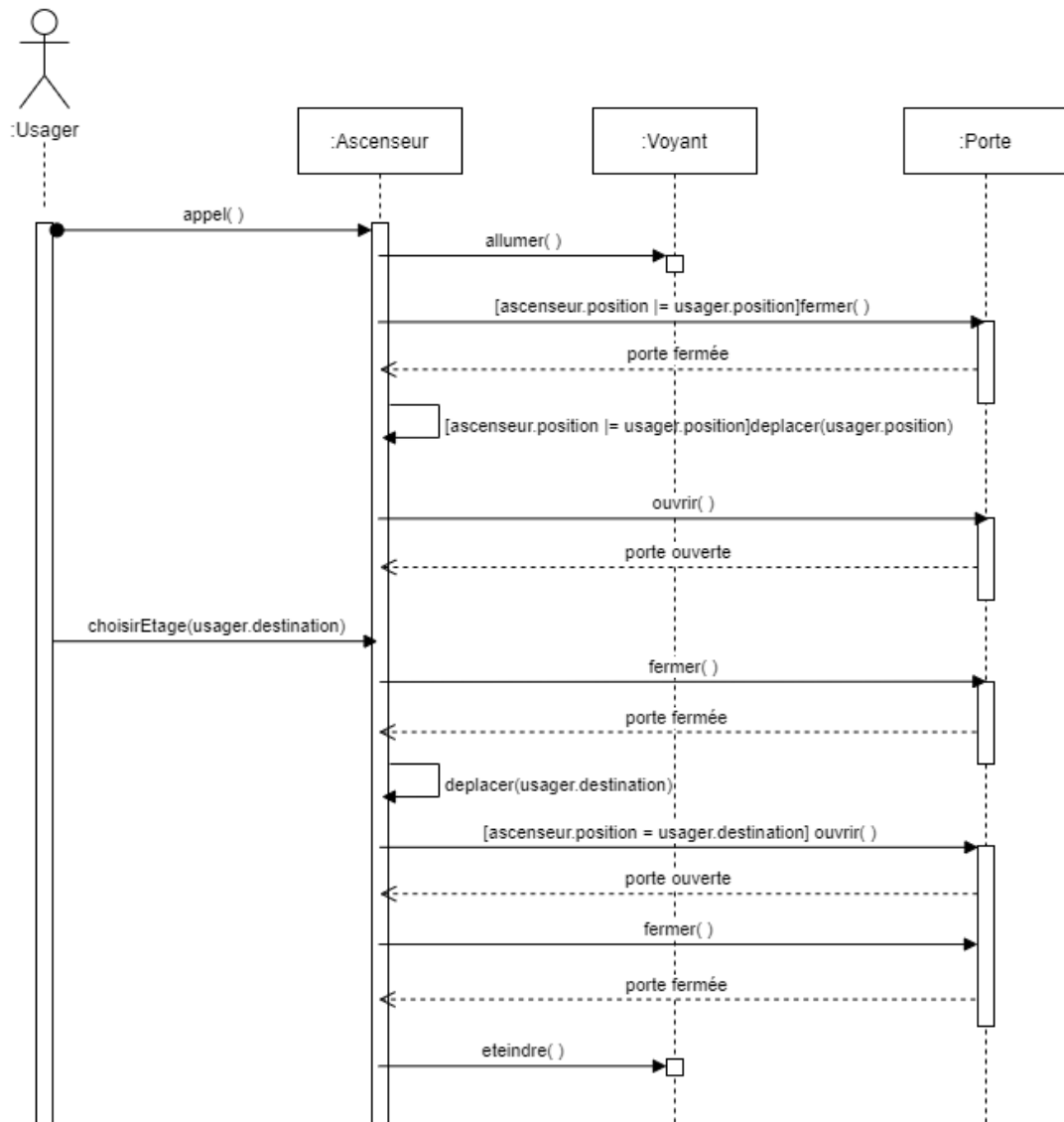


Diagramme de séquences d'une cabine d'ascenseur



## Diagramme de blocs internes

≈ *ibd*

≈ *internal block diagram*

Un diagramme de blocs internes est un diagramme structurel statique.

L'objectif des diagrammes de blocs internes (IBD) est de montrer le contenu structurel encapsulé (composants, propriétés, connecteurs, ports, interfaces) des blocs afin qu'ils puissent être décomposés de manière récursive et "reliés" à l'aide d'interfaces. Autrement dit, un IBD est une vue "boîte blanche" (je comprends et je décris ce qu'il y a à l'intérieur du système considéré) d'un bloc qui a jusqu'alors été décrit en "boîte noire" (je ne sais pas ce qui se passe à l'intérieur, je ne peux

que constater son comportement).

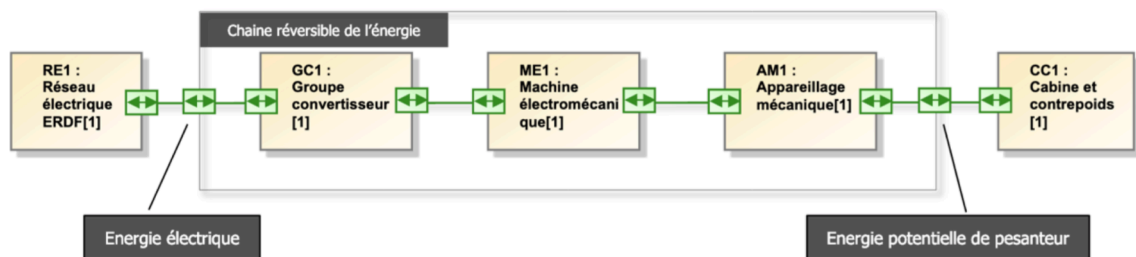
## Spécificités

Le diagramme de blocs internes décrit et donne la liste des différents éléments par lesquels transite l'énergie.

On notera que :

- La nature de l'énergie en transit peut varier, d'où le nom de système multi-physique.
- L'énergie qui transite entre les blocs peut être à sens unique ou à double sens, selon si la chaîne est **réversible** ou **irréversible**.

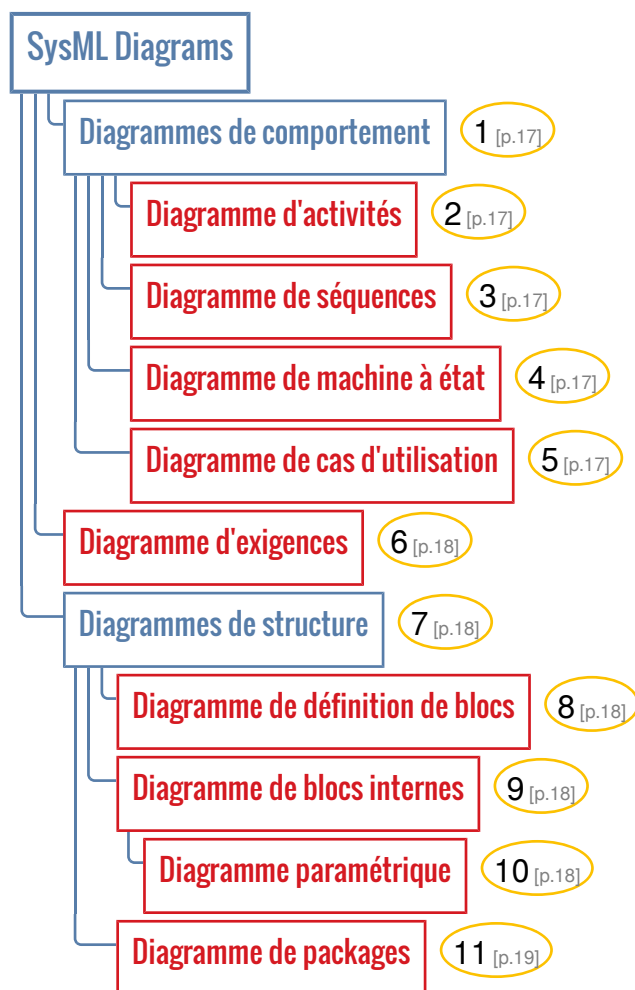
*Si l'énergie est positive, alors elle transite de gauche à droite →*  
*Négative, elle transite de droite à gauche ←*



*Diagramme de blocs internes du système de levage d'une cabine d'ascenseur*



## ❖ Pour aller plus loin Les différents diagrammes de l'ingénierie système



### 1 Diagrammes de comportement

≈ *behavior diagrams*

Les diagrammes de comportement comprennent le diagramme de cas d'utilisation, le diagramme d'activité, le diagramme de séquence et le diagramme de machine à états.

### 2 Diagramme d'activités

≈ *activity diagram*

Le diagramme d'activités représente le flux de données et de contrôle entre les activités.

### 3 Diagramme de séquences

≈ *sequence diagram*

Le diagramme de séquences représente l'interaction entre les parties collaboratrices d'un système.

### 4 Diagramme de machine à état

≈ *state machine diagram*

Le diagramme de machine à état décrit les transitions d'état et les actions qu'un système ou ses parties effectuent en réponse à des événements.

## 5 Diagramme de cas d'utilisation

≈ UC

≈ use case

Un diagramme de cas d'utilisation fournit une description de haut niveau de la fonctionnalité qui est réalisée par l'interaction entre les systèmes ou les parties du système.

Les diagrammes de cas d'utilisation intègrent des relations standardisées :

- <<include>> : le cas d'utilisation de base « ne peut se faire sans » ou « impose que »
- <<extend>> : le cas d'utilisation de base « peut éventuellement se faire avec »
- <<spécialisation>> : il permet de définir la spécialisation d'un cas d'utilisation

## 6 Diagramme d'exigences

≈ req

≈ requirement diagram

Les diagrammes d'exigences servent à préciser ce que l'on attend du système, généralement dérivées du cahier des charges.

## 7 Diagrammes de structure

≈ structure diagrams

Les diagrammes de structure servent à définir l'architecture du système.

Le bloc est l'unité de base de la structure en SysML et peut être utilisé pour représenter le matériel, le logiciel, les installations, le personnel ou tout autre élément du système. La structure du système est représentée par des diagrammes de définition de blocs et des diagrammes de blocs internes. Les packages servent à organiser le modèle.

## 8 Diagramme de définition de blocs

≈ bdd

≈ block definition diagram

Un diagramme de définition de blocs est un diagramme structurel statique.

L'objectif des diagrammes de définition de blocs est de spécifier les composants du système, leur contenu (propriétés, comportements, contraintes), leurs interfaces et leurs relations.

## 9 Diagramme de blocs internes

≈ ibd

≈ internal block diagram

Un diagramme de blocs internes est un diagramme structurel statique.

L'objectif des diagrammes de blocs internes (IBD) est de montrer le contenu structurel encapsulé (composants, propriétés, connecteurs, ports, interfaces) des blocs afin qu'ils puissent être décomposés de manière récursive et "reliés" à l'aide d'interfaces. Autrement dit, un IBD est une vue "boîte blanche" (je comprends et je décris ce qu'il y a à l'intérieur du système considéré) d'un bloc qui a jusqu'alors été décrit en "boîte noire" (je ne sais pas ce qui se passe à l'intérieur, je ne peux que constater son comportement).

**10** Diagramme paramétrique

≈ *parametric diagram*

Le diagramme paramétrique représente les contraintes sur les valeurs des propriétés du système, telles que les performances, la fiabilité, les propriétés de masse, etc.

**11** Diagramme de packages

≈ *pkg*

≈ *package diagram*

Un diagramme de packages est un diagramme structurel statique.

L'objectif d'un diagramme de packages est d'organiser les modèles complexes et de grande taille.

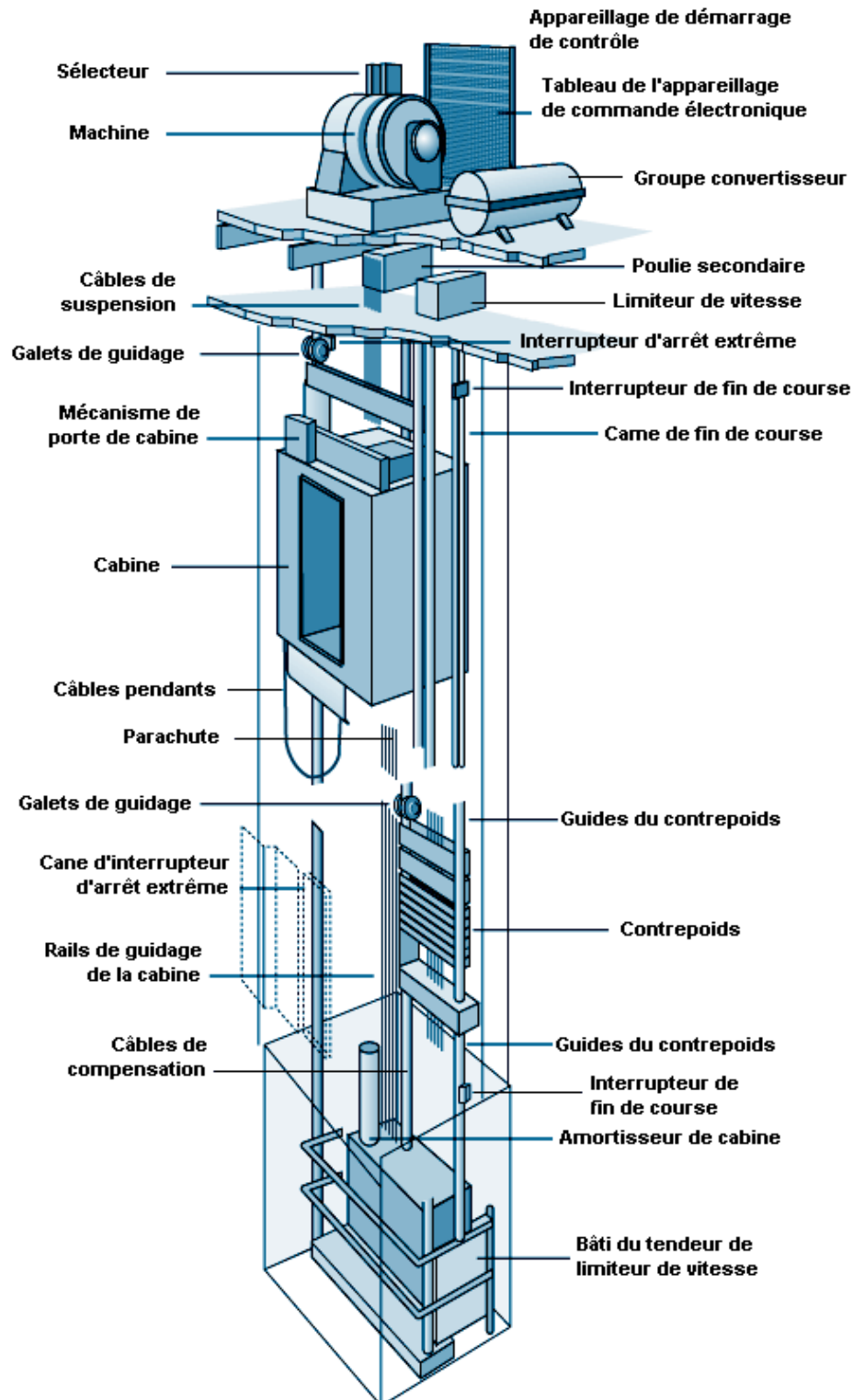
## 1.7. Architecture d'un système multi-physique

---

### 1.7.1. Notion de diagramme fonctionnel : la chaîne d'information et la chaîne d'énergie

---

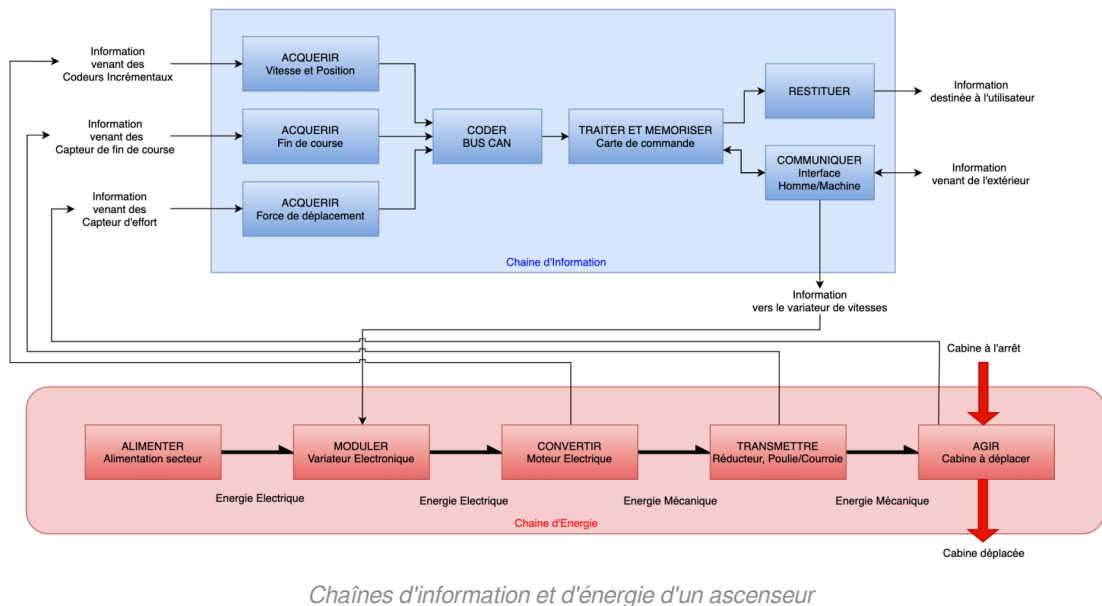
## Exemple L'ascenseur



Source: d'après Otis Elevator Company, New York.

*Mécanisme d'un ascenseur*

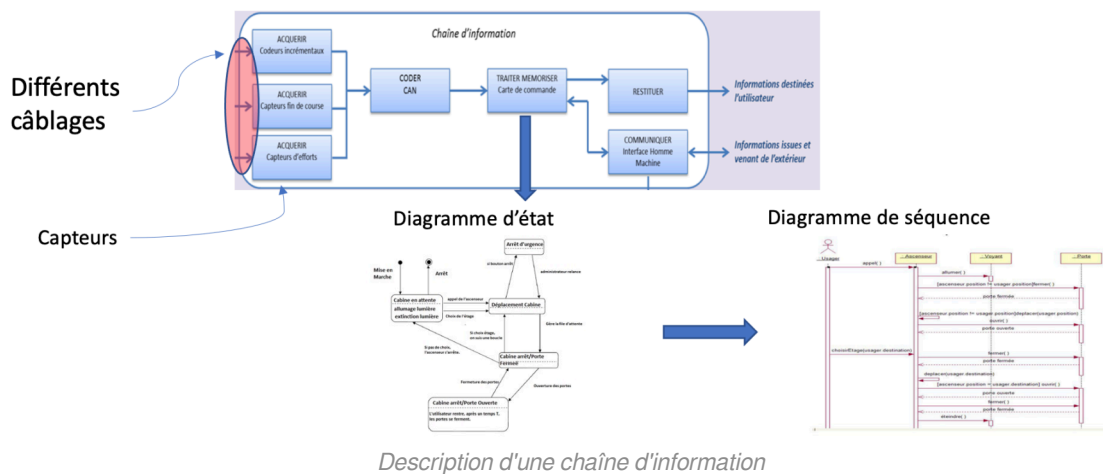
Pour une meilleure lisibilité, on dissociera la chaîne d'information de la chaîne d'énergie :



### Description de la chaîne d'information

L'architecture globale de la chaîne d'information du système multi-physique a pour base le diagramme de blocs internes de l'étude IS.

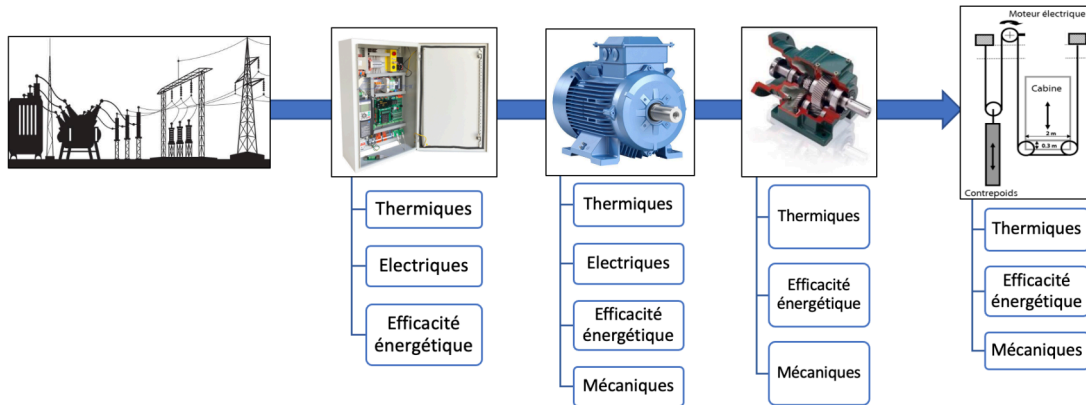
Il sera donc composé de blocs fonctionnels représentant les composants par lesquels transite **l'information**.



### Description de la chaîne d'énergie

L'architecture globale de la chaîne d'énergie du système multi-physique a pour base le diagramme de blocs internes de l'étude IS.

Il sera donc composé de blocs fonctionnels représentant les composants par lesquels transite **l'énergie** et/ou l'information.



Description de la chaîne d'énergie d'un ascenseur

## 1.7.2. Composition des blocs fonctionnels

Dans les logiciels de simulation système, on trouve différents types de blocs permettant de modéliser le comportement des différentes parties d'un système.

Par assemblage des blocs, on peut ainsi construire le système complet.

On notera l'existence de différents types de blocs, tels que les blocs « mathématiques » et les blocs « physiques ».

### Blocs mathématiques => causal

Dans les logiciels de simulation système, on trouve différents types de blocs tels que les blocs « mathématiques » représentant des relations mathématiques entre une entrée et une sortie.

Ils peuvent être utilisés dans différents domaines : temporels ou fréquentiels.

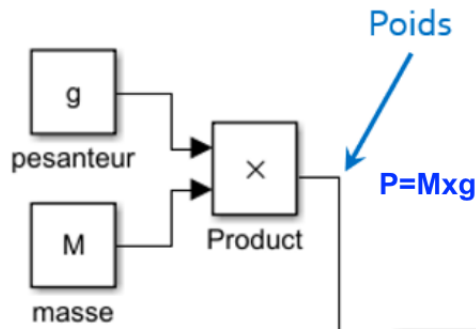
Ils se présentent sous la forme suivante :

#### Réglementaire

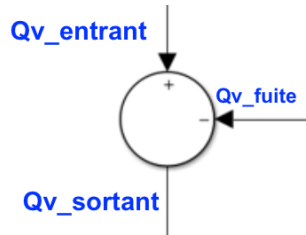
- En électricité l'équation  $U=R \cdot I$  traduit le comportement d'une résistance :



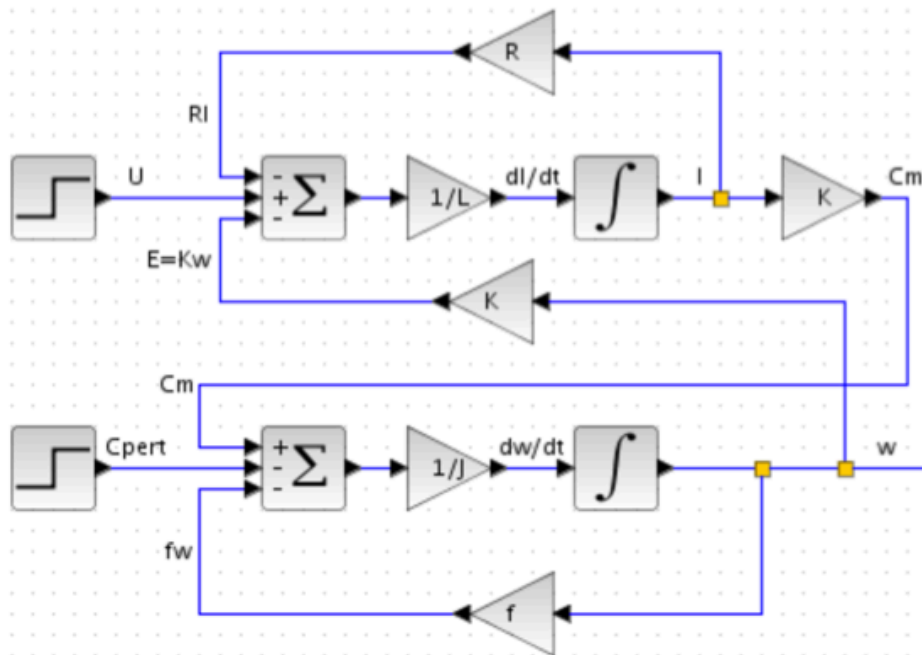
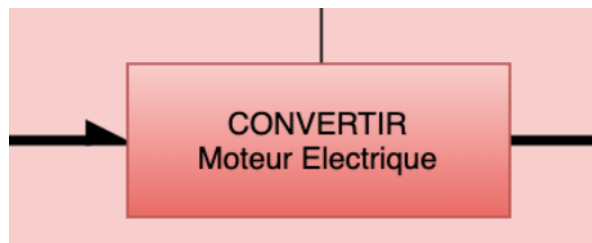
- En mécanique  $P=M \times g$  représente le calcul du poids à partir de la masse de la cabine et de la gravité



- En hydraulique, le débit total est la somme des débits  $Q_v = Q_{v1} + Q_{v2}$



**Exemple** Modèle du moteur à courant continu réalisé à partir de blocs mathématiques



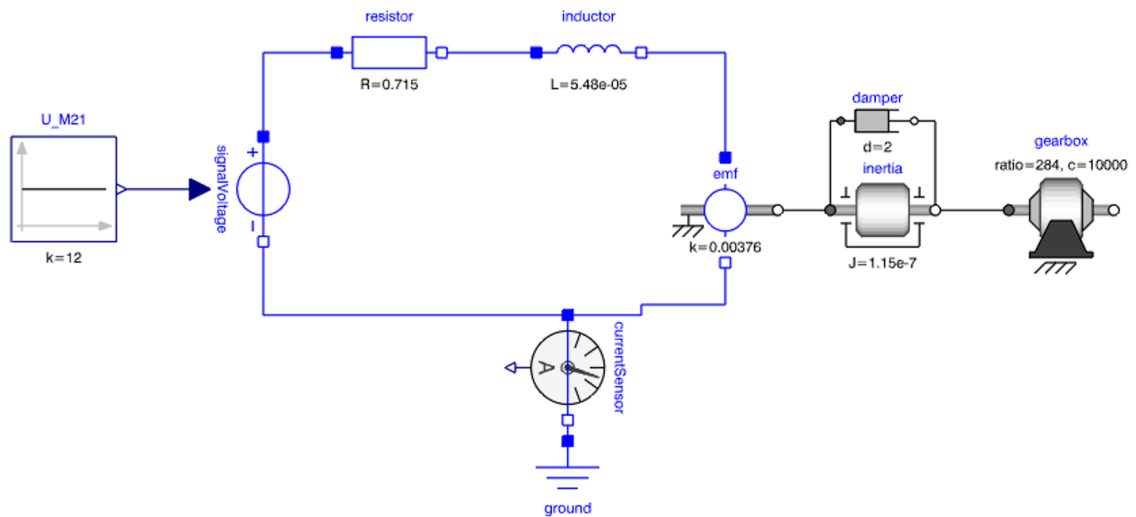
Modèle moteur courant continu (schéma bloc temporel)

## Blocs physiques <=> causal ou acausal

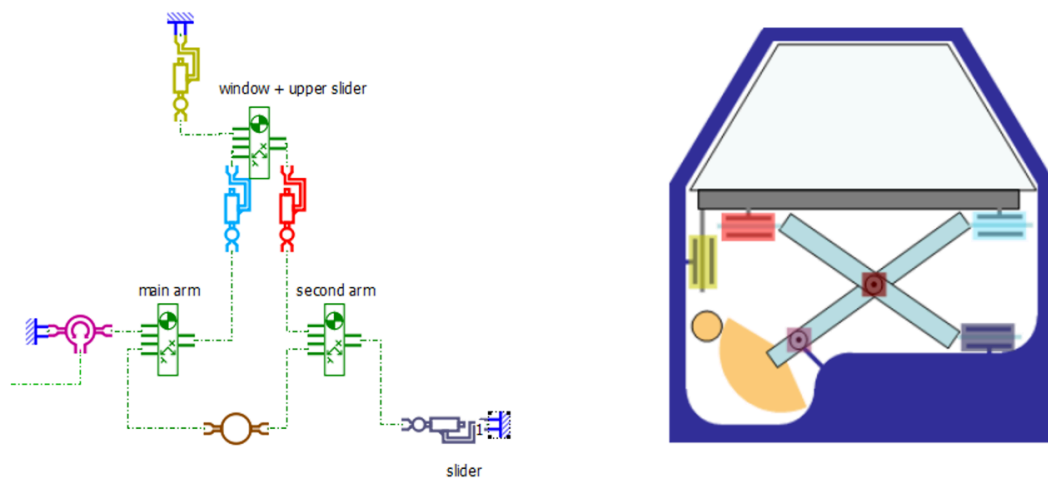
Orientés objets, ils représentent des objets (résistances électriques, masses, moteurs électriques, flexibles hydrauliques, résistances thermiques, etc.) contenant les équations physiques du domaine concerné et portant des « ports » leur permettant de communiquer les uns avec les autres.

Ils se présentent sous la forme suivante :

### Exemple Modèle du moteur à courant continu réalisé à partir de blocs physiques



### Exemple Exemple de mécanisme sur AMESIM



Exemple de mécanisme sur AMESIM

## 1.7.3. Assemblage des blocs



## Continuité des variables

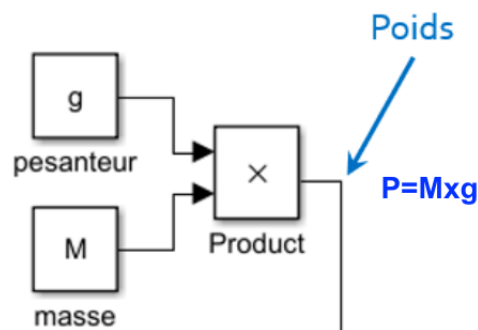
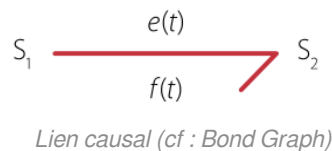
Il est important de noter que deux blocs ne peuvent être reliés que par une flèche transmettant le même type de puissance (mécanique, hydraulique, thermique, etc.), d'où l'importance de faire figurer l'unité de la variable transmise par la flèche.

## Communication des blocs mathématiques

L'assemblage des blocs « mathématiques » se fait grâce à des liens unidirectionnels permettant de transmettre les résultats des calculs effectués dans un bloc au suivant.

### Exemple Lien à sens unique

Lien transportant une puissance calculée à partir des deux variables d'état  $e(t)$  (effort) et  $f(t)$  (flux) :



### Remarque

Il est impératif de permettre la continuité des unités transmises.

## Communication des blocs physiques

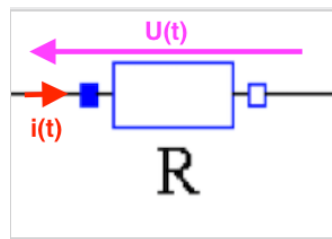
L'assemblage des blocs physiques se fait grâce à des liens physiques bidirectionnels permettant de transmettre des variables calculées à l'intérieur des blocs, correspondant à la physique des ports à relier.

### Exemple Lien à double sens OpenModélica

Lien physique transportant une quantité électrique entre deux éléments de la bibliothèque d'objets électriques :



Lien acausal (OpenModélica)

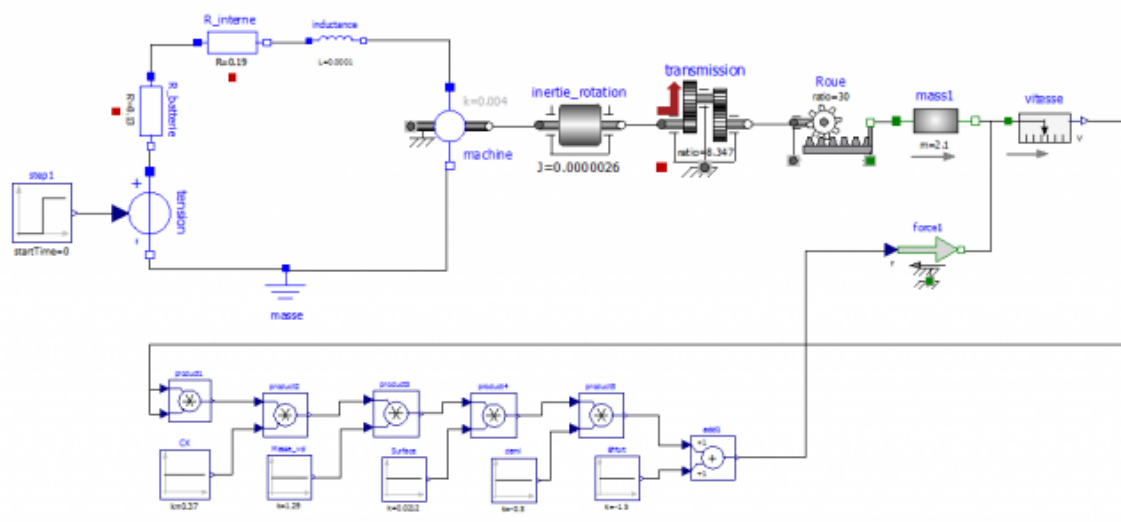


## 1.7.4. Architecture mixte

### Principe

En mélangeant les deux types de blocs (mathématiques et physiques), on obtient un modèle mixte. Les modèles mixtes sont des schémas composés des deux types de blocs.

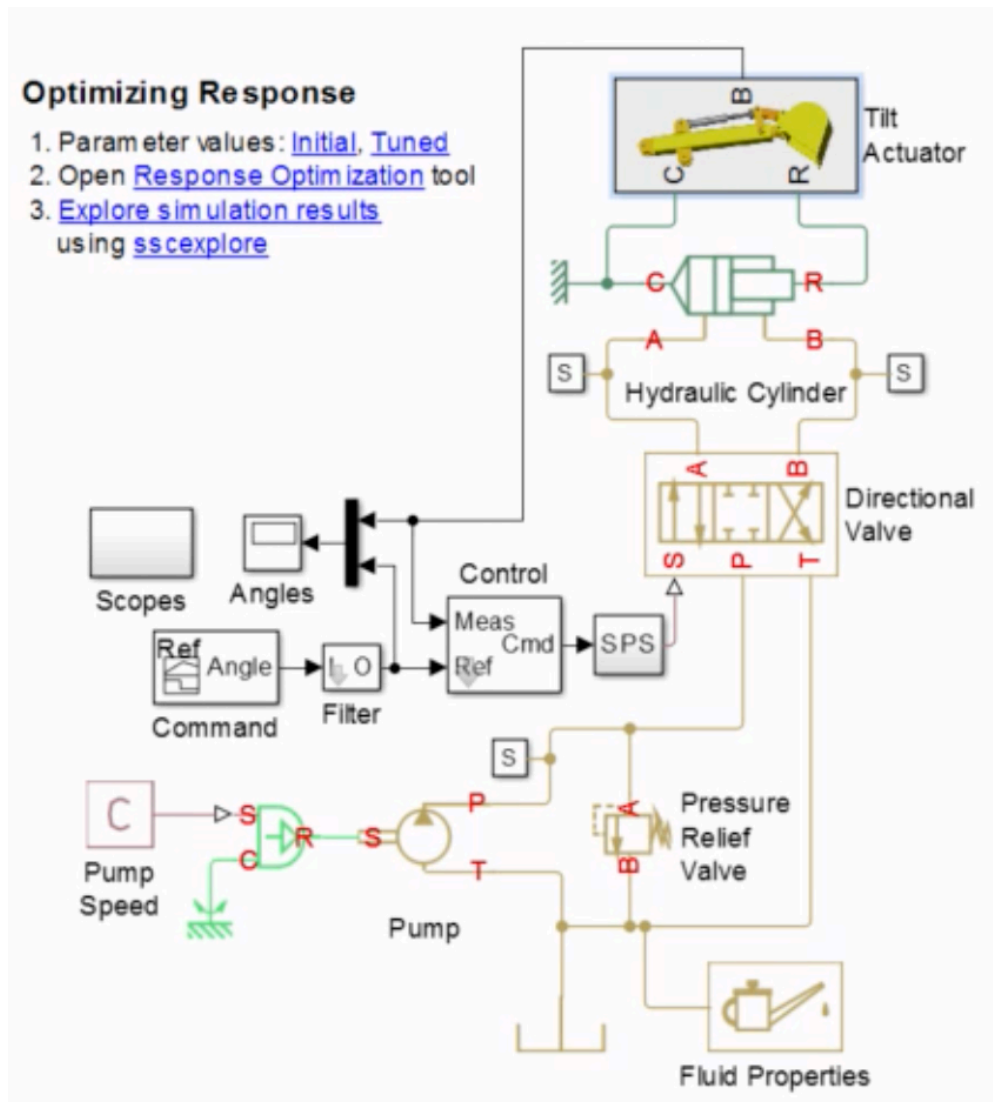
### Exemple Modèle mixte de la cabine d'ascenseur prenant en compte les effets aéro-dynamiques



Prise en compte de la résistance de l'air :

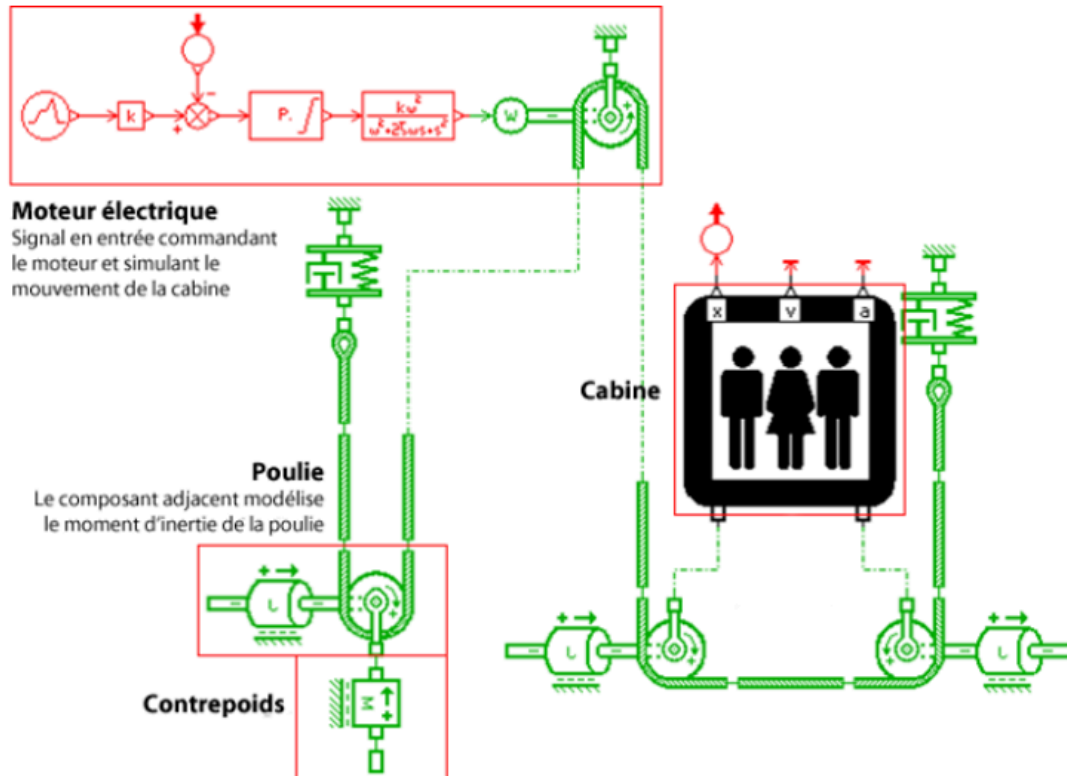
$F = 1/2 \cdot C_x \cdot m_v \cdot S \cdot V^2$ . L'effort aérodynamique  $F = 1/2 \cdot C_x \cdot m_v \cdot S \cdot V^2$  est ajouté à l'effort constant.

## Exemple Modèle mixte d'un vérin de pèle mécanique sur Simscape



Modèle mixte d'un vérin de pèle mécanique sur Simscape

## Exemple Modèle mixte de l'ascenseur sur AMESim



Modèle mixte de l'ascenseur sur AMESim

## 1.8. Notion de paramètres et de variables

### Paramètre

Quel que soit le type de blocs, ils devront tous contenir les caractéristiques des éléments représentés que l'on appellera **paramètres**.

#### Paramètres localisés

Ce sont des paramètres choisis en fonction de l'objectif de l'étude.

#### Paramètres distribués

Ce sont des paramètres qui décrivent la géométrie 2D ou 3D, ainsi que les lois de comportement des matériaux ou des fluides.

### Variable

Valeur d'un paramètre variant au cours du temps pendant le fonctionnement du système.

### Variable de flux

Ce sont les variables qui permettent de transmettre un flux.

### Variable d'efforts

Ce sont les variables qui permettent de transmettre un effort.

## 1.9. Notion de domaines d'études

### Les différents domaines

Un système est généralement composé d'une multitude de domaines.

Chaque domaine utilise ses propres techniques pour modéliser les comportements des éléments qui le composent.

- **Mécanique** : étude des mécanismes en prenant en compte les masses, les inerties des pièces et les liaisons entre pièces
- **Electrique** : étude des circuits électriques
- **Thermique** : étude des comportement thermiques
- **Mécanique des fluides** : étude du comportement des fluides en statique ou en dynamique
- **Logique** : permet de tester le comportement séquentiel et la logique de la commande

### Dualité entre les domaines

Domaine	Puissance	"Effort"		"Flux"	
Translation	$F \times V$	Force	N	Vitesse	$m \times s^{-1}$
Rotation	$\Gamma \times \omega$	Couple	$N \times m$	Vitesse angulaire	$rad \times s^{-1}$
Électricité	$U \times I$	Tension	V	Intensité	$A (C \times s^{-1})$
Hydraulique	$p \times Q$	Pression	Pa	Débit	$m^3 \times s^{-1}$
Thermique	$T \times j$	Température	K	Courant thermique	$J \times s^{-1}$

*Dualité entre les variables*

## 1.10. Notion de couplages multi-physiques et de solveurs

### Couplages multi-physiques

Le fonctionnement de la plupart des produits industriels met en jeu des interactions entre différents domaines (mécanique, thermique, chimique, électromagnétique...). Lors d'une simulation, il est important de permettre à ces domaines de communiquer (couplage).

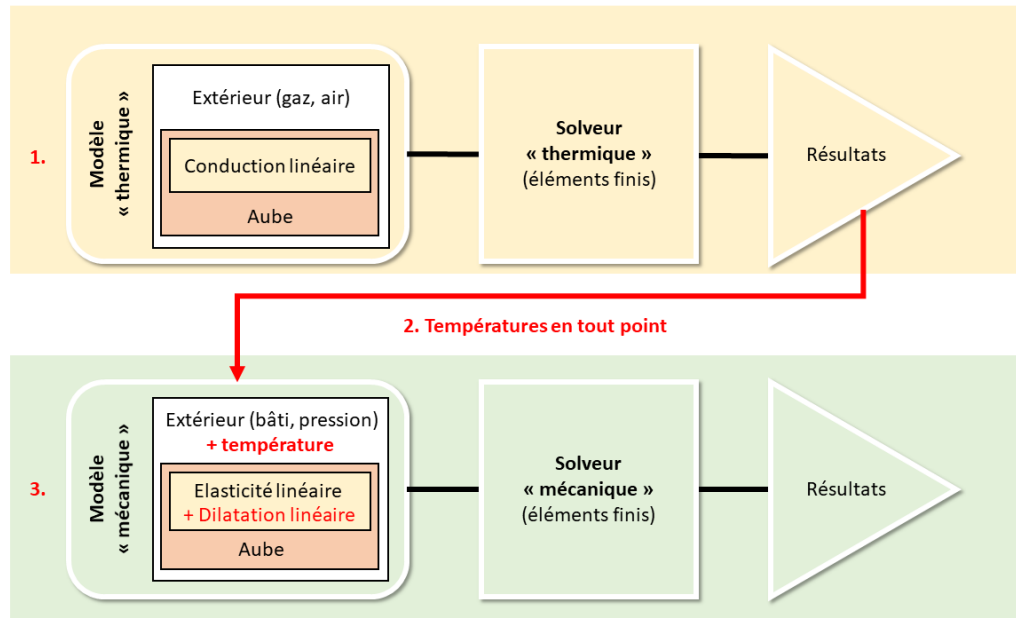
Il existe différents types de couplages.

## Couplage faible

Lors d'une simulation d'interaction entre deux physiques, l'un des deux modèles ne dépend pas de l'autre, il peut être analysé en premier.

### Exemple Simulation thermo-mécanique du comportement de l'aube de turbine

La démarche de résolution serait la suivante :

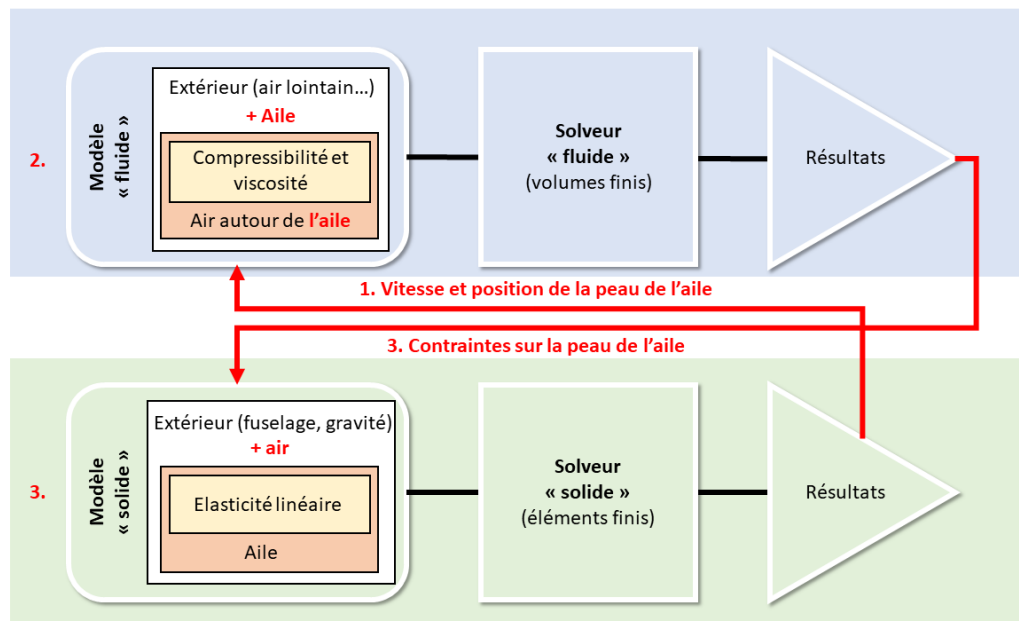


Exemple : simulation du comportement thermo-mécanique faiblement couplé d'une aube de turbine

## Couplage fort

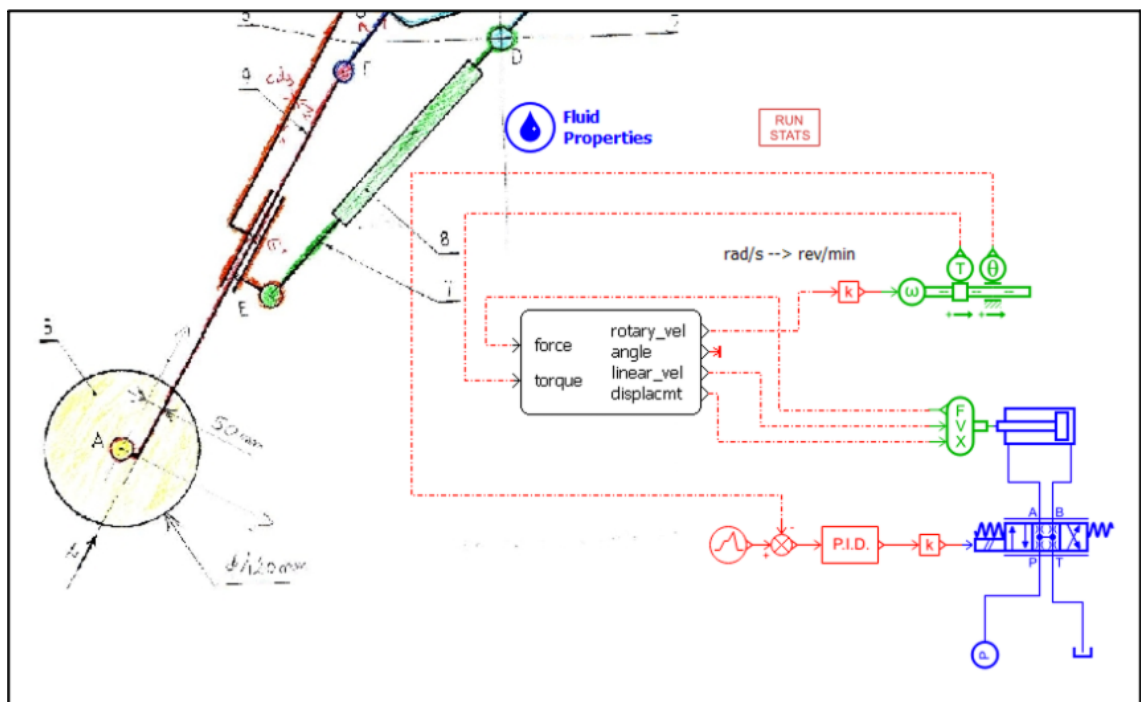
Lors d'une simulation d'interaction entre deux physiques, chacune des deux physiques dépend de l'autre. Aucune simulation ne peut être menée à bien indépendamment. Il est donc nécessaire d'échanger des résultats entre les deux solveurs, dans les deux sens, afin d'assurer le couplage.

## Exemple Déformation d'une aile d'avion soumise à un flux d'air



Exemple : simulation du comportement fluide-solide fortement couplé d'une aile d'avion

## Exemple Exemple de co-simulation



Co-simulation entre ADAMS et AMESim

## Logiciels de simulation système multi-physique

Chaque domaine nécessitera l'emploi de solveurs adaptés.

## Solveur

En règle générale un logiciel de simulation multi-physique transforme un modèle graphique (sous forme de blocs) en une représentation ODE (Ordinary Differential Equations) pour effectuer une simulation en utilisant différentes méthodes d'intégration numérique.

Ces méthodes d'intégration sont aussi appelées solveur et se distinguent par leur caractéristique :

- explicit vs. implicit
- order
- step size control
- multi step

NB : Ces caractéristiques sont plus ou moins adaptées à la physique considérée.

### Solveurs DYMOLA

DASSL est le solveur par défaut dans DYMOLA, pour plusieurs raisons. Il s'agit d'un solveur multi-étapes implicite d'ordre supérieur avec un contrôle de taille de pas et avec ses propriétés. Il est assez stable pour une large gamme de modèles. De plus, il a un code source mature, qui a été développé à l'origine dans les années 80.

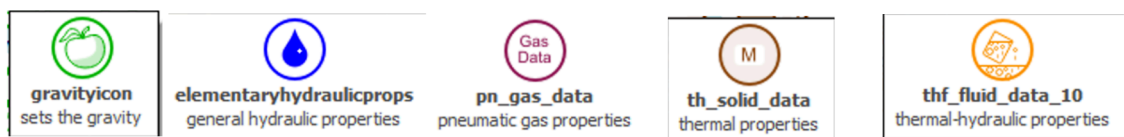
Ce solveur est basé sur la formule de différenciation vers l'arrière (BDF), qui est une famille de méthodes implicites d'intégration numérique. L'implémentation utilisée est appelée DASPK2.0 et elle est traduite automatiquement en C par f2c.

Les indicateurs de simulation suivants peuvent être utilisés pour ajuster le comportement du solveur pour des problèmes de simulation spécifiques : jacobian, noRootFinding, noRestart, initialStepSize, maxStepSize, maxIntegrationOrder, noEquidistantTimeGrid.

Pour en savoir plus, suivre le lien suivant : OpenModélica<sup>[<https://openmodelica.org/doc/OpenModelicaUsersGuide/OpenModelicaUsersGuide-latest.pdf>]</sup>

### Solveurs AMESIM

Dans AMESIM, il existe différents solveurs adaptés aux bibliothèques utilisées.



*Exemples de solveurs dans le logiciel AMESIM*

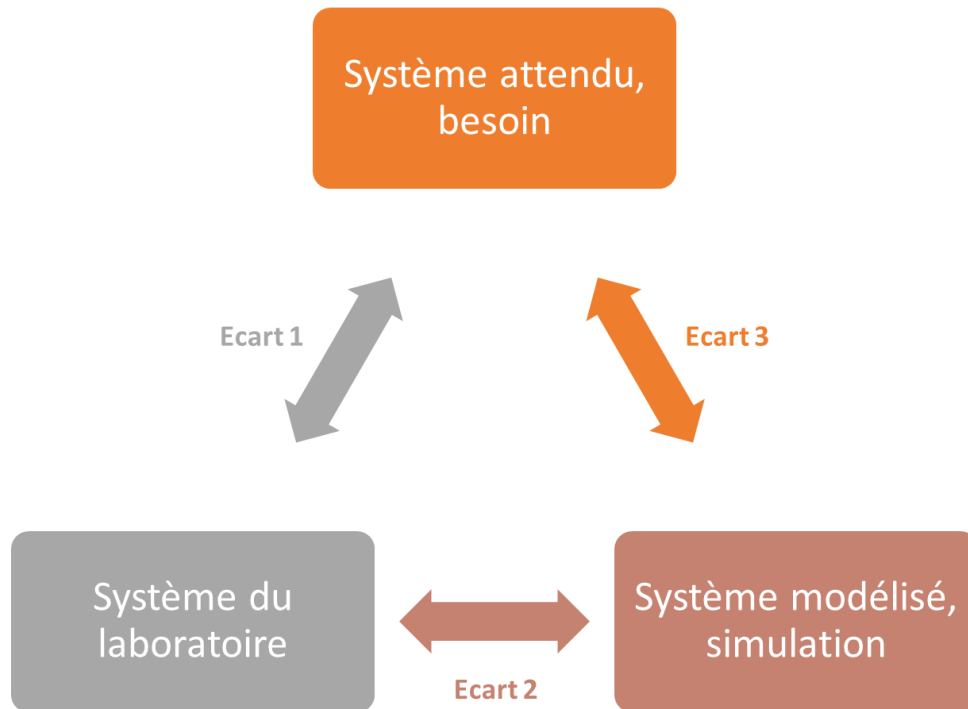
- Le solveur **Gravity**, utilisé avec la bibliothèque mécanique 1D, il permet de prendre en compte la gravité dans un mécanisme 1D.
- Le solveur **Elementaryhydraulicprops**, utilisé avec la bibliothèque hydraulique 1D permet de prendre en compte le type de fluide utilisé (viscosité de l'huile, par exemple).

## 1.11. Performances et optimisation du système : Simulée-Mesurée-Souhaitée



## Performances du système

L'objectif est d'être capable de réduire les écarts entre la simulation et le système réel tout en respectant le cahier des charges.



### Performances du système

Un première analyse vise à tester les performances du système sans la commande.

### Performances de la commande

Un seconde analyse vise à tester la commande et ainsi vérifier que les correcteurs sont bien choisis.

## Optimisation du système

Les logiciels de simulation multi-physique proposent des outils interactifs et des blocs pour analyser et ajuster les paramètres des modèles. Vous pouvez déterminer la sensibilité du modèle, l'adapter aux données de test et l'ajuster pour répondre aux exigences.

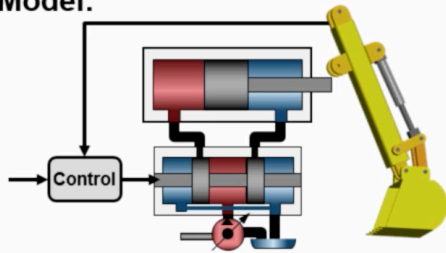
Vous pouvez optimiser conjointement les paramètres physiques et les gains algorithmiques ou du contrôleur pour améliorer les caractéristiques de conception du système comme le temps de réponse, la bande passante et la consommation d'énergie.

### Exemple Optimisation des paramètres

Dans cet exemple on utilisera un logiciel d'optimisation afin de régler les coefficients du correcteur PI, pour que la position du godet corresponde en tout point à la courbe en pointillé visible ci-dessous.

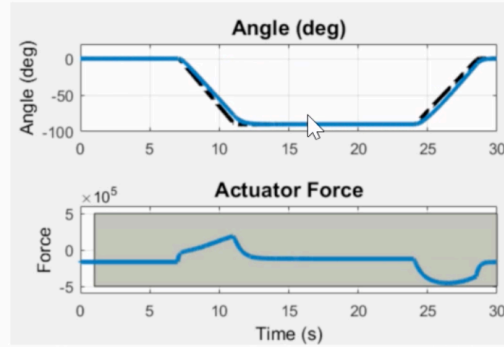
## Optimizing System Performance

**Model:**



**Problem:** Optimize the position controller to meet system requirements

**Solution:** Tune controller parameters with [Simulink Design Optimization](#)



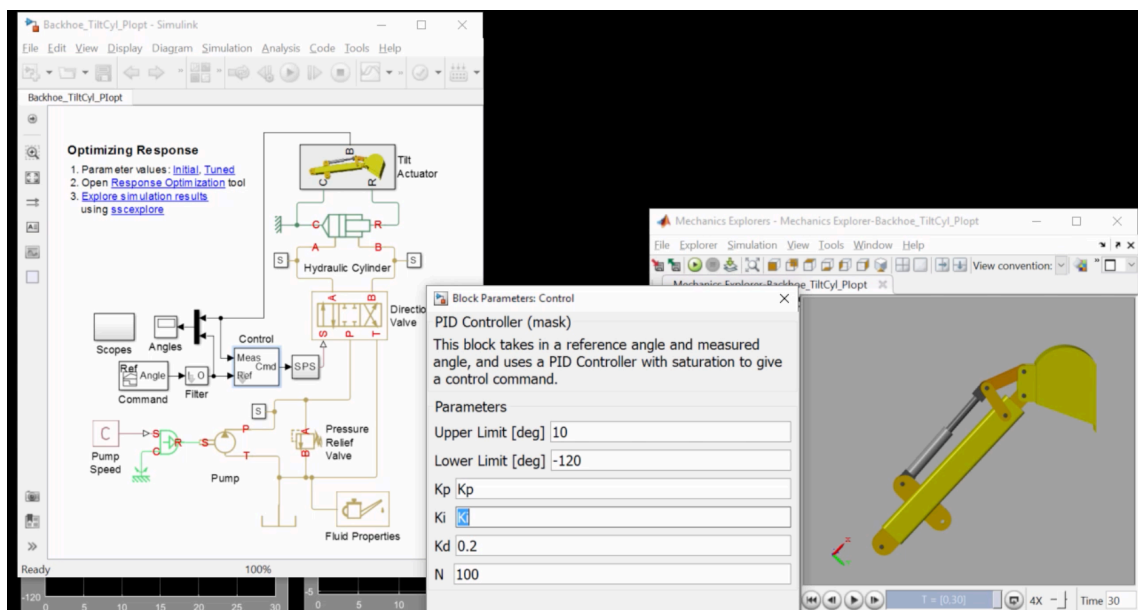
PI Position Control	$K_p$	$K_i$
	23.4	3.67

Après quelques simulations automatiques, le logiciel converge vers un couple de coefficients  $K_p$  et  $K_i$  idéaux.

La courbe bleu s'aligne sur le courbe cible en pointillé ci-dessus.

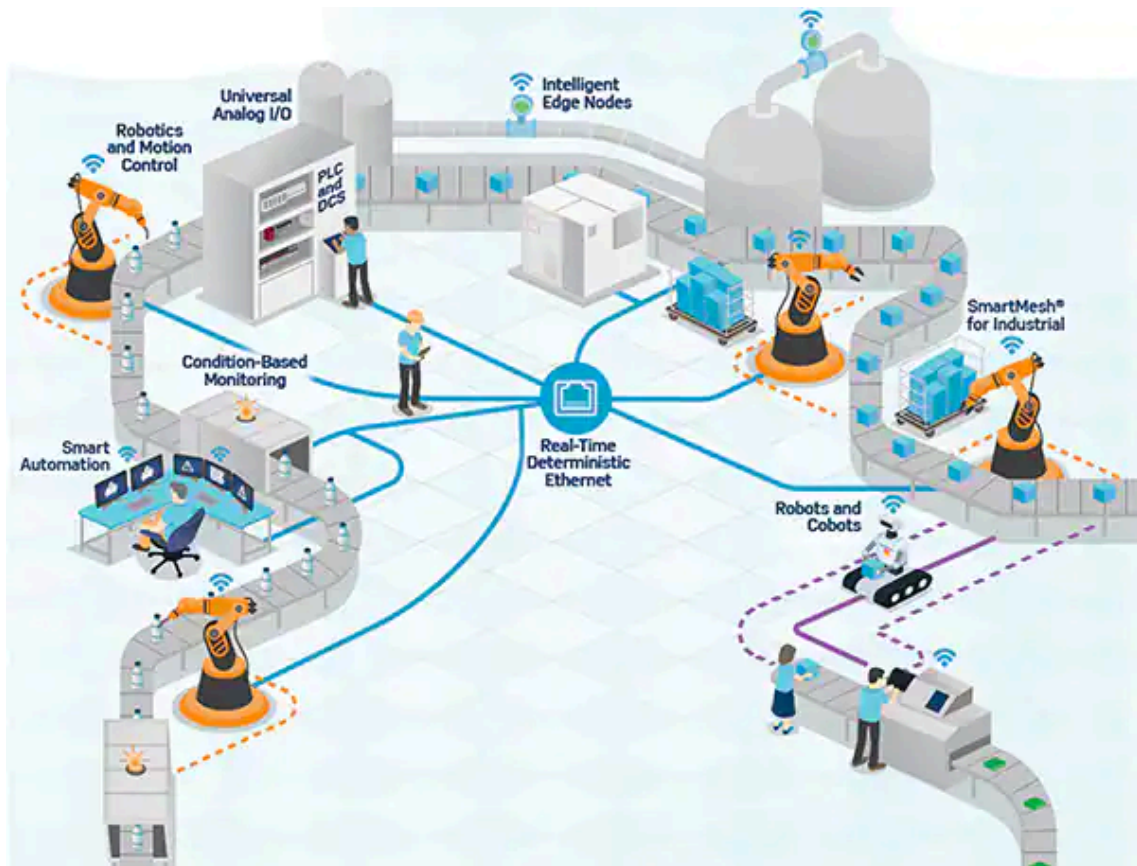
### Exemple Tolérances sur l'erreur de position

Sur l'image ci-dessous, on peut voir la valeur de la tolérance de l'erreur de position acceptée.



## 1.12. Utilisation des jumeaux numériques

Le concept de jumeau numérique permet de créer un modèle virtuel très complexe de la réplique exacte d'un élément physique réel.



Jumeau numérique

Dans la figure ci-dessus, les moniteurs de processus utilisent des données pour piloter le système réel. Grâce à des capteurs judicieusement placés, on pourra non seulement mesurer les performances du système réel afin de les confronter aux résultats attendus par le client, mais aussi vérifier les performances du système virtuel afin de caractériser au mieux le comportement de chacun de ses composants.

L'objectif étant d'améliorer les performances du système actuel puis des futures évolutions.

### Exemple IRON BIRD

On pourra utiliser des jumeaux numériques pour simuler le fonctionnement lorsque le système réel n'est pas accessible ou pour qualifier des composants avant de les insérer dans le système réel.

Airbus fait parti des précurseurs dans le domaine en construisant l'IRON BIRD, qui n'est rien d'autre qu'un banc de test incluant tous les actionneurs de l'A380.

Le même banc a été réalisé plus tard pour l'A350.

Pour plus de détail, suivre le lien suivant :

<https://www.airbus.com/en/newsroom/news/2017-05-taking-flight-with-the-airbus-iron-bird>

## 1.13. Ce qu'il faut retenir

Le modèle de la simulation système contient :

- L'architecture définie par l'IS lors de la phase de conception.
- Un assemblage structuré de **briques** contenant des équations et traduisant le comportement physique des différents éléments constituant le système.
- Des **solveurs** permettent d'effectuer des calculs avec la précision souhaitée et de faire communiquer des différentes briques entre elles.

Le rôle de la simulation système :

- **Analyser** le comportement d'un système complet de manière collaborative.
- **Optimiser** le dimensionnement des éléments constituant le système.

## 2. Modélisation par schéma blocs

---

### 2.1. Introduction

---

#### Objectifs pédagogiques

L'objectif de ce module est de présenter **les étapes de construction d'un modèle mathématique causal**<sup>[p.100]</sup> **d'un système multi-physique complexe**, en vue de simuler son comportement.

A l'issue de ce module les étudiants doivent être capable de :

- Choisir les variables du modèle
- Mettre le système en équation
- Transcrire les équations en blocs fonction de transfert
- Construire un schéma bloc
- Simuler son fonctionnement
- Analyser son comportement

#### Déroutement Présentation rapide des différentes étapes

- Les concepts fondamentaux
- Quelques modèles causals d'éléments simples
- Application : modélisation d'une pompe de relevage

## 2.2. Concepts fondamentaux

---

### 2.2.1. Modélisation système multi-physique

---

#### Attention Modélisation 1D

L'état de l'art actuel en modélisation multi-physique consiste à simplifier les modèles judicieusement afin de se ramener à des systèmes d'équations où les variables d'intérêt dépendent uniquement du temps.

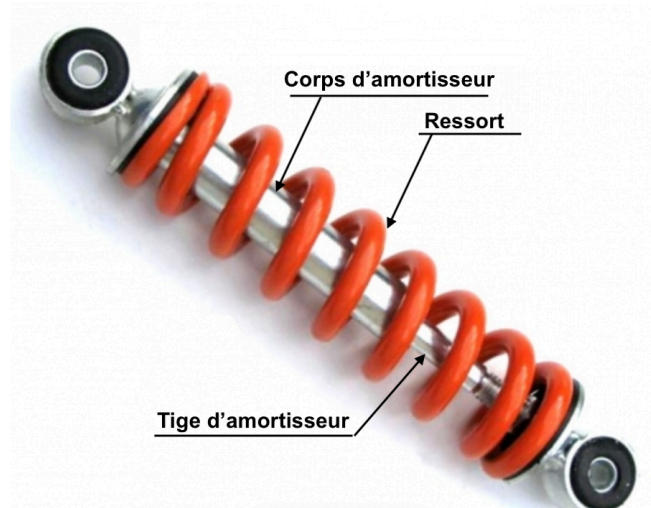
## Paramètres localisés

Ce sont les paramètres que l'on choisit pour représenter les caractéristiques des éléments composants le système étudié.

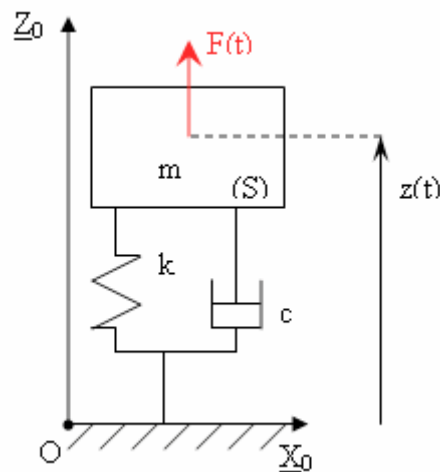
Il s'agit ici de localiser dans la chaîne d'énergie les éléments caractéristiques du système, dont on va suivre l'évolution pendant l'étude.

## Exemple

Pour représenter une suspension de véhicule automobile, on utilisera un système « masse-ressort-amortisseur »



Exemple d'un amortisseur



Modélisation paramétrée d'un système Masse/Ressort/Amortisseur

Le paramètre  $k$  représentant la raideur de la suspension ne prendra en compte que la rigidité du ressort sur un seul axe, mais pas les déformations circonférentielles du corps de l'amortisseur, ou une éventuelle flexion de la tige de ce dernier.

## Choix des variables d'entrée et de sortie

Ce sont les variables qui permettent de représenter le comportement du système. On distingue deux sortes de variables.

## Variable de flux

≈ *Through variables*

Elle représentent un flux passant au travers d'un élément, par exemple un débit dans une canalisation, une intensité de courant dans un fil de cuivre, la vitesse d'un solide en déplacement, ou bien un courant thermique dans une pièce, etc.

## Variable d'effort

≈ *Across variable*

Elle représente une différence de potentiel, permettant de créer un flux.

Par exemple une force (mécanique), une tension (électrique), une pression (hydraulique) ou une température (thermique), etc.

## Truc & astuce Dualité des variables

Domaine	Puissance	"Effort"		"Flux"	
Translation	$F \times V$	Force	N	Vitesse	$m \times s^{-1}$
Rotation	$\Gamma \times \omega$	Couple	$N \times m$	Vitesse angulaire	$rad \times s^{-1}$
Électricité	$U \times I$	Tension	V	Intensité	$A (C \times s^{-1})$
Hydraulique	$p \times Q$	Pression	Pa	Débit	$m^3 \times s^{-1}$
Thermique	$T \times j$	Température	K	Courant thermique	$J \times s^{-1}$

*Dualité entre les variables*

## 2.2.2. Mise en équation du système

### Introduction

La mise en équation est une étape primordiale de la modélisation du système, elle dépend des hypothèses et de techniques de modélisation du domaine concerné :

- Mécanique
- Electricité
- Hydraulique
- Thermique

Ces différents domaines vont nous fournir des équations représentant les lois d'évolution des variables d'intérêt.

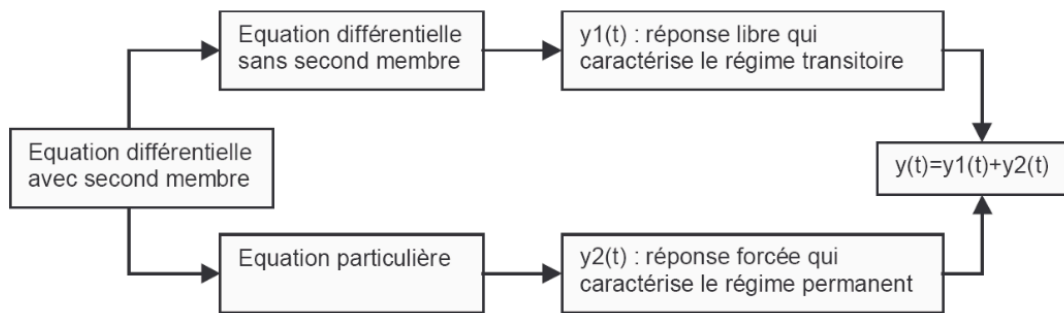
On aura souvent à faire à des équations différentielles, d'ordre plus ou moins important.

### Résolution classique des équations différentielles

Exemple d'équation différentielle :

$$\frac{d^2s(t)}{dt^2} + 5 \frac{ds(t)}{dt} + 6s(t) = e(t)$$

Pour résoudre les équations différentielles obtenues, on pourra utiliser les technique vues en cours de mathématiques.



### Résolution des équations différentielles dans le domaine de Laplace

Exemple de transformation d'équation différentielle dans le domaine de Laplace :

$$\frac{d^2s(t)}{dt^2} + 5 \frac{ds(t)}{dt} + 6s(t) = e(t)$$

avec  $s(0) = 2, s'(0) = 2$  et  $e(t) = 6 u(t)$

On applique la transformation de Laplace à cette équation:

$$\mathcal{L}\left[\frac{d^2s(t)}{dt^2}\right] + 5 \mathcal{L}\left[\frac{ds(t)}{dt}\right] + 6 S(p) = E(p)$$

$$p^2 S(p) - p s(0) - s'(0) + 5 [p S(p) - s(0)] + 6 S(p) = E(p)$$

$$p^2 S(p) - 2 p - 2 + 5[p S(p) - 2] + 6 S(p) = \frac{6}{p}$$

soit 
$$S(p) = \frac{2p^2 + 12p + 6}{p(p^2 + 5p + 6)} = \frac{2p^2 + 12p + 6}{p(p+2)(p+3)}$$

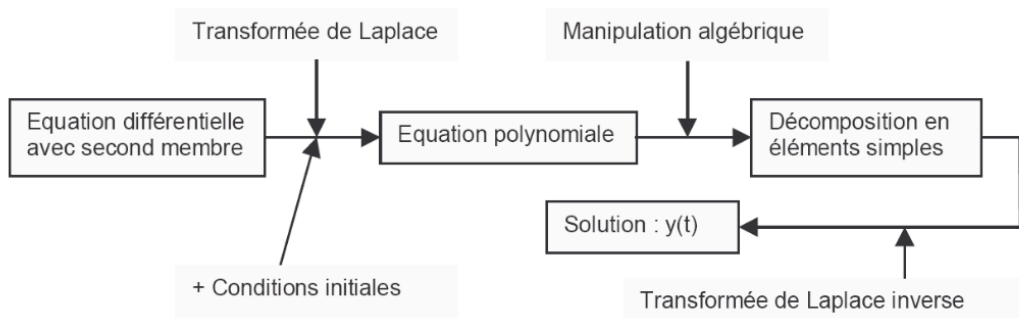
On décompose cette fraction en éléments simples: 
$$S(p) = \frac{A}{p} + \frac{B}{p+2} + \frac{C}{p+3}$$

Par identification, on trouve 
$$S(p) = \frac{1}{p} + \frac{5}{p+2} - \frac{4}{p+3}$$

On retourne au domaine temporel en prenant les transformées inverses, d'où:

$$s(t) = (1 + 5 e^{-2t} - 4 e^{-3t}). u(t)$$

Pour résoudre les équations différentielles obtenues, on pourra utiliser la technique suivante :





### ◆ Remarque Quelques transformées de Laplace

$f(t)u(t)$	$F(p)$	$f(t)u(t)$	$F(p)$
$\delta(t)$	1	$\sin \omega t$	$\frac{\omega}{p^2 + \omega^2}$
K	$\frac{K}{p}$	$\cos \omega t$	$\frac{p}{p^2 + \omega^2}$
Kt	$\frac{K}{p^2}$	$\text{sh} \omega t$	$\frac{\omega}{p^2 - \omega^2}$
$e^{-at}$	$\frac{1}{p + a}$	$\text{ch} \omega t$	$\frac{p}{p^2 - \omega^2}$
$t^n$	$\frac{n!}{p^{n+1}}$	$e^{-at} \sin \omega t$	$\frac{\omega}{(p + a)^2 + \omega^2}$
$1 - e^{-t/\tau}$	$\frac{1}{p(1 + \tau p)}$	$e^{-at} \cos \omega t$	$\frac{p + a}{(p + a)^2 + \omega^2}$
$e^{at} t^n$	$\frac{n!}{(p - a)^{n+1}}$		

## 2.2.3. Notion de fonction de transfert

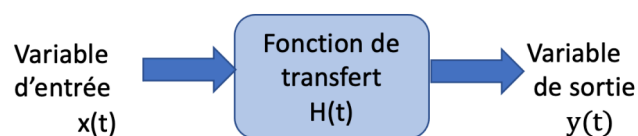
### ✎ Fonction de transfert

Une fonction de transfert est une relation mathématique permettant de relier une variable de sortie  $y$  avec une variable d'entrée  $x$ .

Elle respecte le **principe de causalité**.

#### Représentation

On représente la fonction de transfert sous la forme suivante, appelée **Bloc** :



#### Expressions

- Expression temporelle :

$$H(t) = \frac{y(t)}{x(t)} = \frac{\textit{Sortie}}{\textit{Entrée}}$$

- Expression fréquentielle (domaine de Laplace) :

$$H(p) \frac{y(p)}{x(p)} = \frac{\text{Sortie}}{\text{Entrée}}$$

Fonction de transfert du premier ordre

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

Fonction de transfert du second ordre

$$H(p) = \frac{K}{1 + \frac{2\xi}{\omega_0} p + \frac{1}{\omega_0^2} p^2}$$

## 2.2.4. Notion de causalité

### Attention Type de causalité

Un système multi-physique peut être **causal** ou **acausal**.

### Causalité

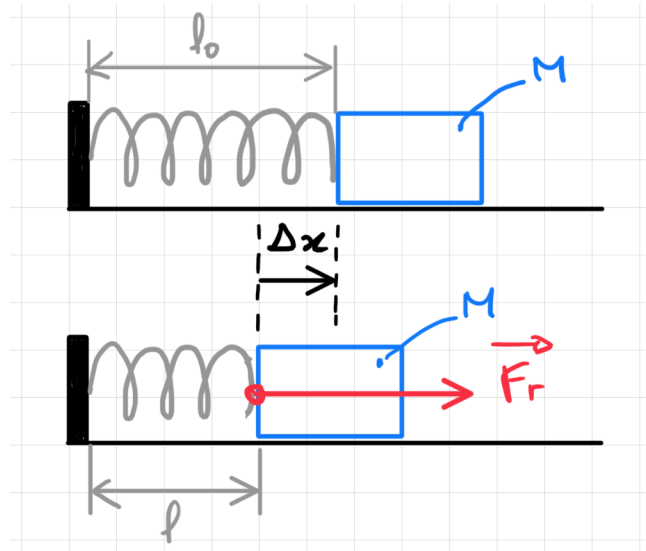
La notion de causalité permet à l'outil informatique de séquencer les équations modélisant la physique du système et de les traiter dans un ordre logique.

Elle consiste à désigner les variables d'entrée et de sortie d'un modèle et de chacun de ses sous-modèles.

L'objectif est de créer une relation de cause à effet entre les entrées et les sorties du modèle.

### Exemple

Si on prend l'exemple d'un ressort de raideur  $k$  et de longueur à vide  $L_0$  :



L'effort  $F_r$  peut être exprimé en fonction de la raideur  $k$  du ressort et de sa déformation  $\Delta x$  par la relation :  $F_r = k \cdot \Delta x$ .

La déformation  $\Delta x$  peut aussi être exprimée en fonction de la raideur  $k$  et de l'effort  $F_r$  par la relation :  $\Delta x = F_r / k$ .

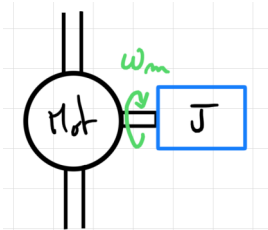
Il est donc possible de choisir la causalité  $F_r = f(\Delta x)$  ou  $\Delta x = f(F_r)$ .

### 💡 Truc & astuce Causalité intégrale

Lorsque l'équation sera différentielle, **on privilégiera toujours la causalité intégrale**, pour éviter des problèmes de calcul numérique.

### 🔑 Exemple

Soit un moteur de couple moteur  $C_m$ , entraînant en rotation une inertie  $J$  :



Le principe fondamental de la dynamique nous permet d'écrire :

- $C_m = J \cdot (d(\text{Oméga}) / dt)$

On préférera écrire :  $\text{Oméga} = \text{Intégrale de } (C_m / J)$

## 2.2.5. Notion de lien entre les blocs

### ⚠ Attention Choix de la causalité transmise par un lien

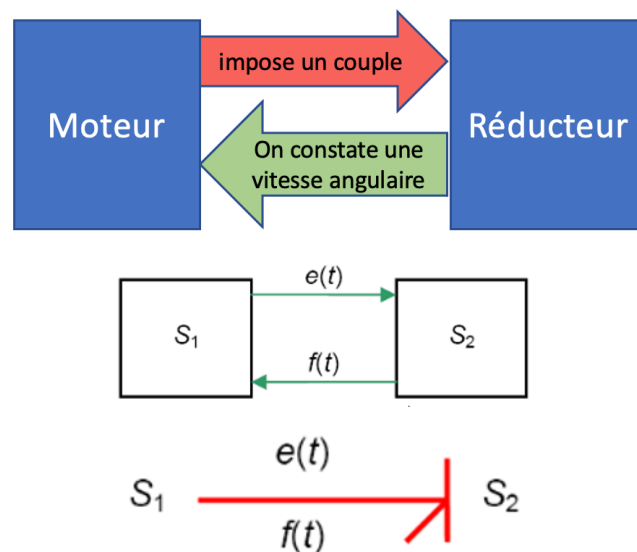
Les liens entre les blocs d'un système **causal** sont unidirectionnels, il est donc important de fixer le type de causalité utilisé pour répondre à l'objectif fixé :

- si on cherche à commander un système à partir d'un effort, on imposera une causalité de type **effort**.
- si on souhaite commander notre système avec un flux, on imposera une causalité de type **flux**.

## Effort imposé

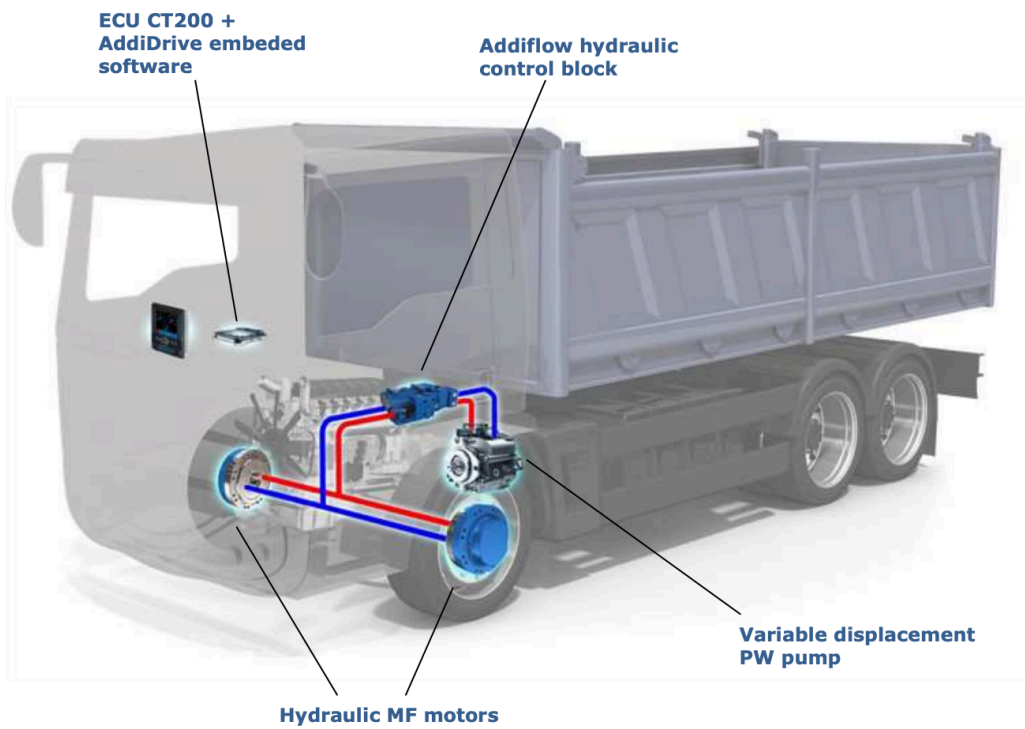
### 📌 Conseil

Dans un système, on peut choisir d'imposer la variable de type **Effort** et de regarder les effets sur la variable de type **Flux**.



### 🔧 Exemple Pilotage du couple moteur

Dans certaines applications telles que les moteurs roues utilisés sur quelques camions et engins de chantier, on pilote le couple moteur au niveau des roues afin de garder l'adhérence des roues sur terrains glissants.



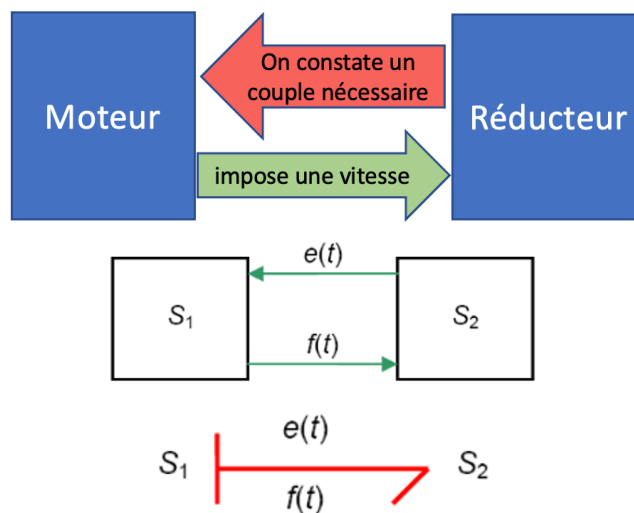
Exemple d'un cas de pilotage couple moteur

On mesure donc en permanence les différences de vitesses des roues pour adapter le couple moteur en conséquence.

## Flux imposé

### Conseil

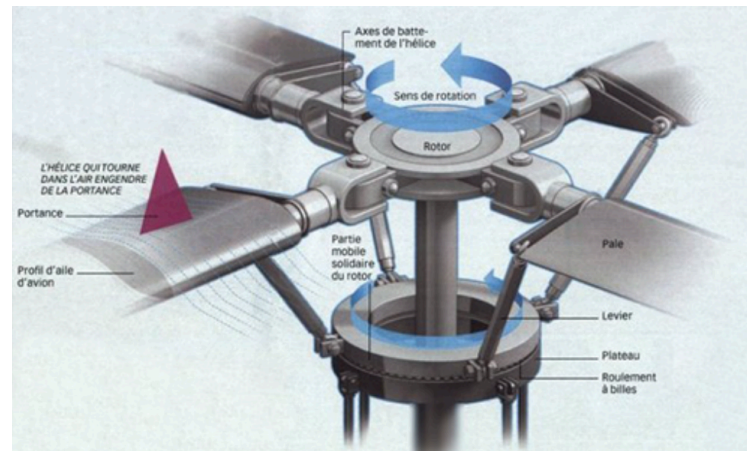
Dans un système, on peut choisir d'imposer la variable de type **Flux** et de regarder les effets sur la variable de type **Effort**.



## Exemple Pilotage en vitesse

Pour certains systèmes, comme les hélicoptères, c'est la vitesse de rotation des rotors qui permet d'avoir la portance nécessaire pour voler.

Il est donc primordial de maintenir une vitesse constante et mesurer le couple obtenu.



Principes de fonctionnement d'une tête de rotor d'hélicoptère

## 2.2.6. Schémas blocs

### Schéma bloc

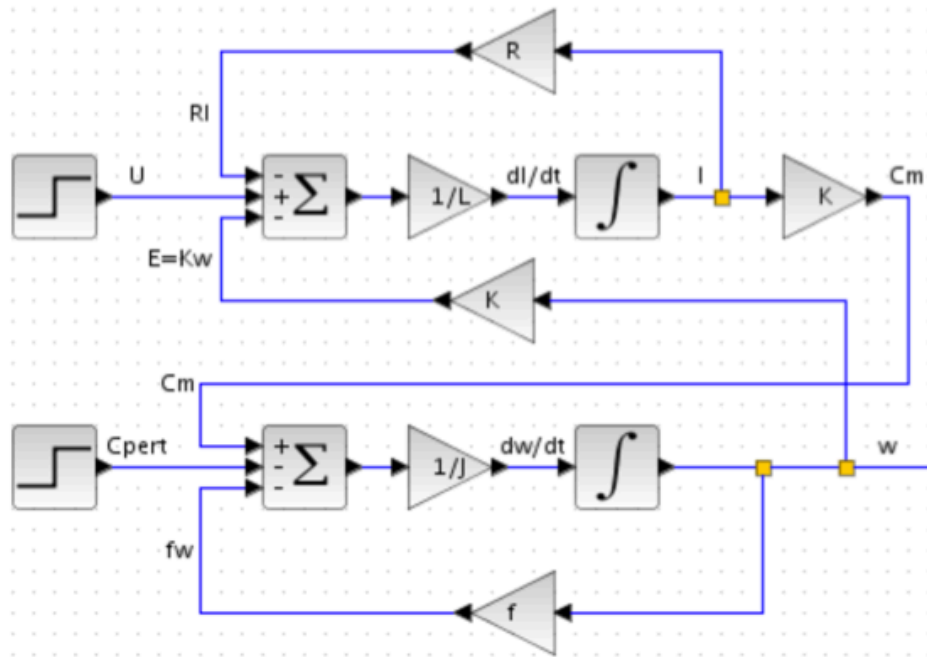
Le schéma bloc est un outil pour décrire et comprendre le comportement dynamique d'un système décrit par des équations différentielles qui peuvent être écrites et résolues dans deux domaines différents :

- Domaine temporel
- Domaine fréquentiel

### Schémas blocs temporels

#### Principe

Les équations différentielles peuvent être écrites directement sur les logiciels de simulation système sous forme de schémas blocs.



Modèle moteur courant continu (schéma bloc temporel)

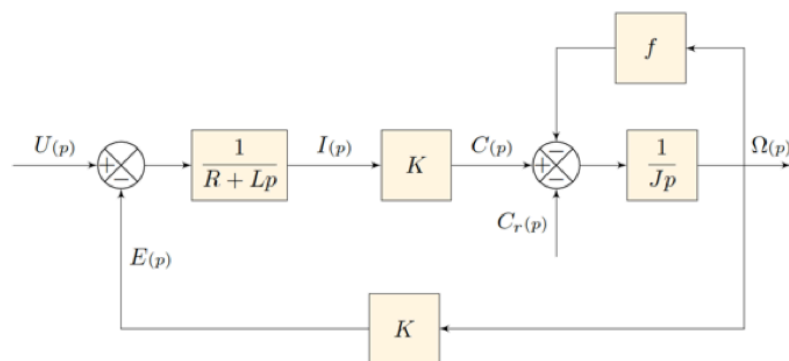
On pourra ainsi simuler le comportement du système afin d'observer l'évolution des variables transitant au travers des différents blocs.

## Schéma blocs fréquentiels (Domaine de Laplace)

### Principe

Par la transformation de Laplace, il est ainsi très aisé de résoudre l'équation différentielle et d'en déduire la fonction de transfert globale du système, ses zéros, ses pôles, ses comportements temporels et fréquentiels dans le cas des systèmes linéarisés.

Les liens entre les blocs d'un schéma bloc « transportent » une seule information et ne peuvent donc traduire que de façon incomplète les transferts d'énergie dans un système.



Modèle moteur courant continu (schéma bloc fréquentiel)

## 2.3. Quelques modèles d'éléments simples

## 2.3.1. Quelques modèles causals d'éléments passifs

### ✎ Éléments passifs

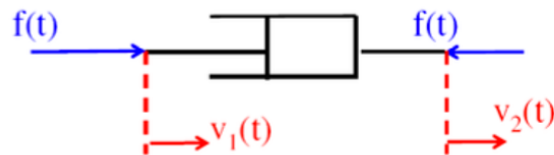
Ce sont des éléments qui reçoivent de l'énergie.

### ■ Éléments passifs-résistifs

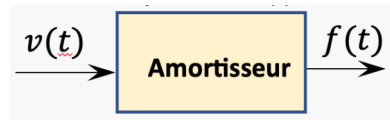
#### ◆ Remarque

Les éléments passifs-résistifs vont dissiper de l'énergie.

#### Amortisseurs

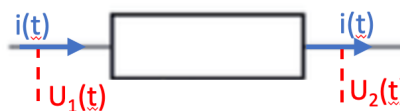


Schéma\_amortisseur



$$f(t) = b \cdot v(t)$$

#### Résistances



$$u(t) = R i(t)$$



## Autres éléments résistifs

Domaine	Interprétation	Formulation
Translation	Frottement visqueux	$F = rV$
	Frottement sec	$F = r \times \text{sgn}(V)$
Rotation	Frottement visqueux	$\Gamma = r\omega$
	Frottement sec	$\Gamma = r \times \text{sgn}(\omega)$
Hydraulique	Restriction pour un écoulement laminaire	$p = rQ$
	Restriction pour un écoulement turbulent	$p = rQ^2$
Électricité	Résistance	$U = rI$

## ◆ Remarque

Les grandeurs d'entrée et de sortie peuvent être inversées, il n'y a pas de causalité. La relation est valable dans les deux sens.

## ■ Éléments passifs-capacitifs

## ◆ Remarque

Les éléments passifs-capacitifs vont emmagasiner de l'énergie.

## Les ressorts

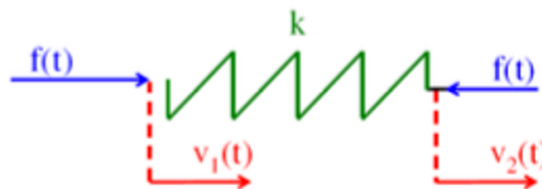
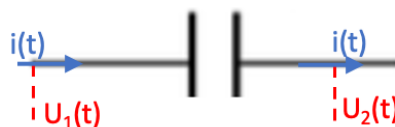


Schéma ressort



$$F(t) = -K (x(t) - x_0(t))$$

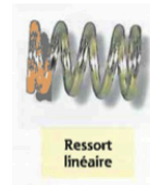
## Les condensateurs



$$i(t) = C \cdot \frac{dU(t)}{dt}$$

Autres éléments capacitifs

Domaine	Interprétation	Formulations
Translation	Ressort linéaire	$V = \frac{1}{k} \frac{dF}{dt}$ $F = k \int_0^t V(\tau) d\tau$
Rotation	Ressort de torsion	$\omega = \frac{1}{k} \frac{d\Gamma}{dt}$ $\Gamma = k \int_0^t \omega(\tau) d\tau$
Hydraulique	Réservoir	$Q = C \frac{dp}{dt}$ $p = \frac{1}{C} \int_0^t Q(\tau) d\tau$
Électricité	Condensateur	$I = C \frac{dU}{dt}$ $U = \frac{1}{C} \int_0^t I(\tau) d\tau$



◆ Remarque

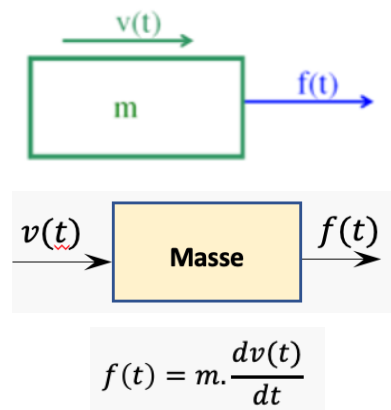
La grandeur d'entrée cause une variation de grandeur de sortie : on parle d'élément causal.

■ **Éléments passifs-inductifs**

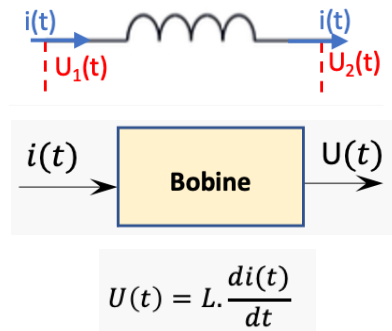
◆ Remarque

les éléments passifs-inductifs reçoivent de l'énergie, la force induit un flux.

Les masses



## Les bobines



### Conseil Autres éléments inductifs

Domaine	Interprétation	Formulations	
Translation	Masse en translation	$F = M \frac{dV}{dt}$	$V = \frac{1}{M} \int_0^t F(\tau) d\tau$
Rotation	Masse en rotation	$\Gamma = J \frac{d\omega}{dt}$	$\omega = \frac{1}{J} \int_0^t \Gamma(\tau) d\tau$
Hydraulique	Tube	$p = I \frac{dQ}{dt}$	$Q = \frac{1}{I} \int_0^t p(\tau) d\tau$
Électricité	Inductance	$U = L \frac{dI}{dt}$	$I = \frac{1}{L} \int_0^t U(\tau) d\tau$



### Remarque

La grandeur d'entrée cause une variation de grandeur de sortie : on parle d'élément causal.

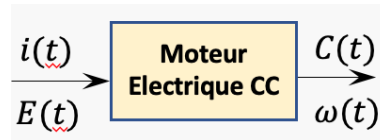
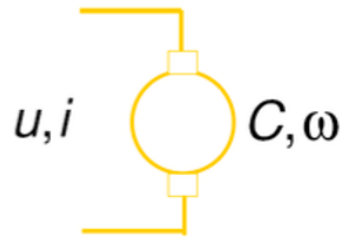
## 2.3.2. Quelques modèles causals d'éléments de jonction

### Les convertisseurs

### Remarque

Les convertisseurs modifient la nature de l'énergie qui y transite.

## Les moteurs électriques en courant continu



$$C = ki$$

$$e = k\omega$$

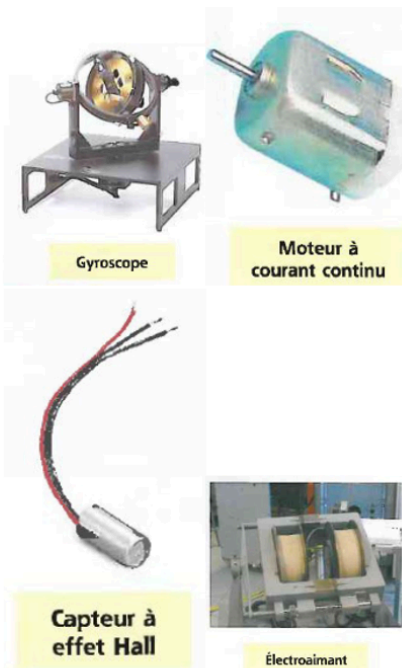
## Autres convertisseurs

### Domaine

Rotation  
Mécano-électrique  
Mécano-électrique  
Électricité

### Gyrateur

Gyroscope  
Machine à courant continu  
Électro-aimant  
Capteur à effet Hall



### ◆ Remarque

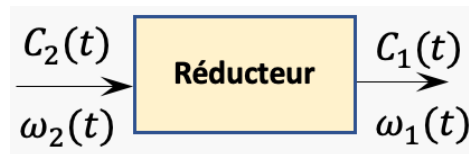
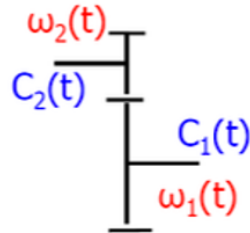
L'énergie électrique devient de l'énergie mécanique.  
Ils peuvent être réversibles ou irréversibles.

## Les transformateurs

### ◆ Remarque

Les transformateurs transmettent la puissance, avec ou sans rapport de transmission.

Les réducteurs/amplificateurs mécaniques



$$\frac{\omega_1}{\omega_2} = \lambda$$

$$\frac{C_2}{C_1} = \lambda$$

### ⚠ Attention

Relation valable s'il n'y a pas de pertes énergétiques entre l'entrée et la sortie.

Autres transformateurs

Domaine	Transformateur
Translation	Levier
Rotation	Engrenage
Électricité	Transformateur électrique
Mécano-hydraulique	Vérin



### ◆ Remarque

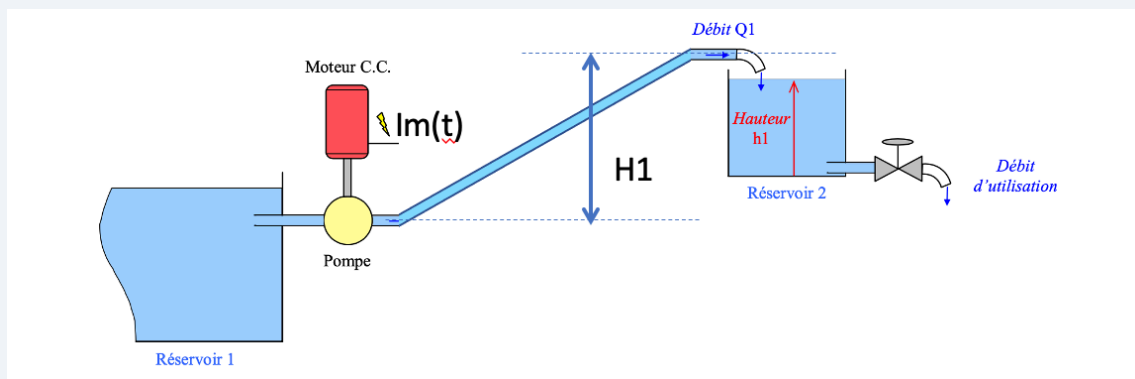
La **puissance** reste de **même nature** entre l'entrée et la sortie.

## 2.4. Méthode de modélisation par schéma blocs, exemple d'une pompe de maintien

### 2.4.1. Introduction

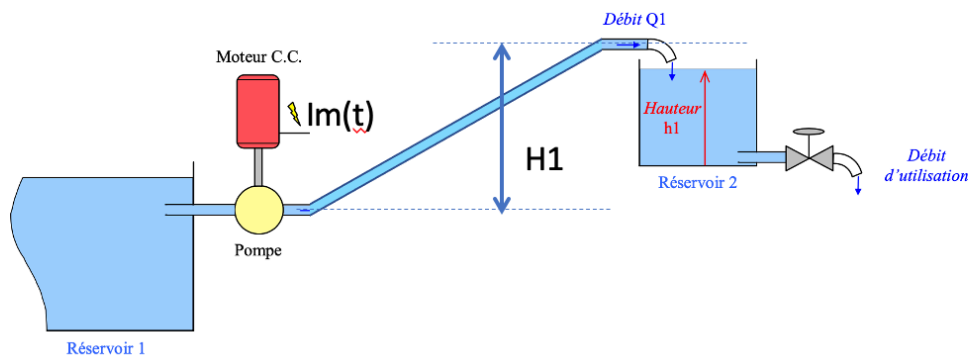
#### Objectifs pédagogiques : Cas d'étude : pompe de maintien de niveau

Pour maintenir le niveau  $h_1$  constant dans le réservoir 2 malgré le débit d'utilisation, on doit établir une relation entre la grandeur d'entrée (ici  $U_m$ ) et la grandeur de sortie (ici  $h_1$ ).



### 2.4.2. Présentation d'une pompe de maintien de niveau

#### Schéma de l'installation



## Fonctionnement

La pompe visible ci-dessus en jaune, actionnée par le moteur à courant continu visible ci-dessus en rouge, permet de déplacer du liquide présent dans le réservoir 1 vers le réservoir 2.

La fonction principale de ce mécanisme est d'assurer un niveau constant ( $h_1$ ) du réservoir 1.

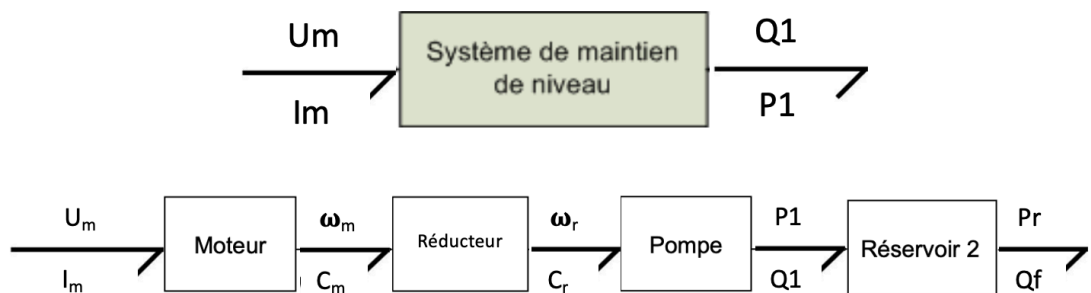
## 2.4.3. Modélisation d'une pompe de relevage

### 2.4.3.1. Modéliser chaque bloc



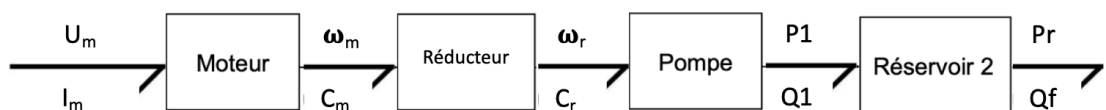
#### 1 Identifier les différents domaines d'études

Ici les domaines sont :



- Mécanique : moteur et réducteur
- Hydraulique : pompe et circuit et réservoir
- Electrotechnique : moteur électrique

#### 2 Identifier les paramètres localisés pour chaque bloc



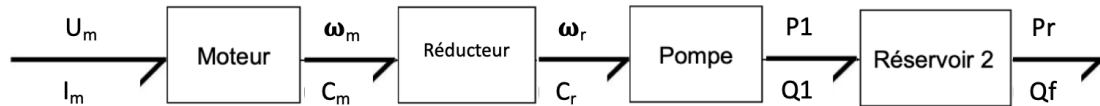
Ici :

- Moteur :  $K$  (Constante de couple),  $R$  (résistance),  $L$  (Inductance) et  $J$  (Inertie du rotor)
- Réducteur :  $r$  (rapport de réduction)
- Pompe :  $C_{yl}$  (Cylindrée)

#### 3 Identifier les variables d'intérêt pour chaque bloc

Lors de cette étape on fera attention à mettre en évidence les paramètres localisés communs à chaque domaine, pour pouvoir connecter les différents blocs appartenant à différents

domaines.



Ici :

- $U_m$ ,  $\omega_m$ ,  $\omega_r$ ,  $Q_1$  et  $Q_f$  sont des variables de **flux**.
- $I_m$ ,  $C_m$ ,  $C_r$ ,  $P_1$  et  $P_r$  sont des variables d'**effort**.

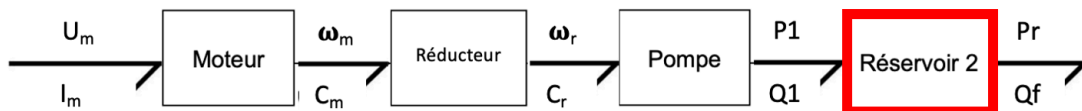
#### 4 Ecrire les équations pour chaque bloc

En utilisant les principes de chaque domaine, écrire les équations faisant intervenir les paramètres localisés précédemment identifiés.

#### 5 Mettre les équations sous forme de blocs

Sous forme de bloc, on peut assembler les équations afin de construire le schéma bloc du système.

### 2.4.3.2. Modéliser le réservoir 2



**Attention** On choisit d'imposer le **flux**.

Cela veut dire que l'on imposera la tension moteur  $U_m$  et on pourra mesurer le débit  $Q_1$  obtenu et la pression  $P_1$  nécessaire.

#### 1 Ecrire l'équation permettant de relier le niveau du réservoir 2, au débit de la pompe.

On écrit le principe de conservation du débit, sachant que la vanne du réservoir 2 permet un débit de fuite  $Q_f$  (non nul).

- Equations de comportement des variables **flux**  $v$  du **réservoir 2**.

$$v_1 \cdot S_{res} = Q_{remplissage} = Q_1 - Q_f ; \quad \text{Et} \quad \int v_1 \cdot dt = h_1$$

Les variables **flux** d'entrées sont  $Q_1$  et  $Q_f$  ; et la variable **flux** de sortie est  $h_1$ .



**2** Ecrire l'équation permettant de relier la pression en sortie du réservoir 2 à la pression en sortie de la pompe.

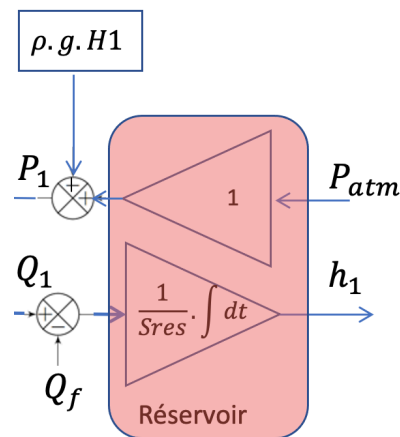
- Equation du comportement des variables **Effort**  $f$  du **réservoir 2** :

$$P_1 = P_{atm} + \rho \cdot g \cdot H1$$

La variable **effort** d'entrée est  $P_{atm}$  et la variable **effort** de sortie est  $P_1$ .

( $\rho$ ,  $g$ , et  $H1$  étant constantes)

**3** Ecrire le bloc correspondant.

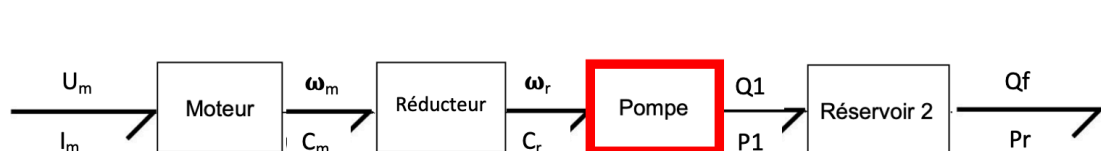


### ⚠ Attention

Attention aux unités :

- $Q_1$  et  $Q_f$  [ $m^3/s$ ]
- $P_1$  et  $P_{atm}$  [ $Pa$ ]
- $h_1$  [ $m$ ]

## 2.4.3.3. Modéliser la pompe



**1** Ecrire l'équation permettant de relier le débit de la pompe, à la vitesse de rotation du réducteur.

On écrit le débit  $Q_1$  en fonction de la cylindrée de la pompe  $C_{y1}$ .

- Equations de comportement des variables **flux** de la **pompe** :

$$\frac{cyl. \omega_{pompe}}{2\pi} = Q_1$$

$$\omega_r = \omega_{pompe}$$

La variable **flux** d'entrée est  $\omega_r$  et la variable **flux** de sortie est  $Q_1$ .

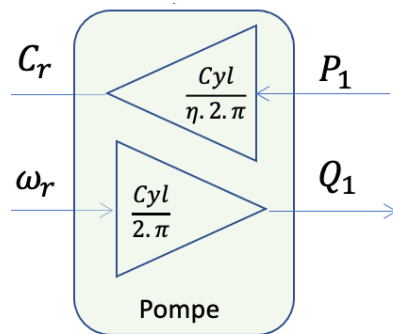
**2** Ecrire l'équation permettant de relier la pression en sortie de pompe  $P_1$  au couple appliqué en sortie de réducteur  $C_r$ .

- Equation du comportement des variables **Effort** de la pompe :

$$\frac{Q_1 \cdot P_1}{C_r \cdot \omega_r} = \eta_p$$

La variable **effort** d'entrée est  $P_1$  [Pa] et la variable **effort** de sortie est  $C_r$  [N.m].

**3** Ecrire le bloc correspondant.

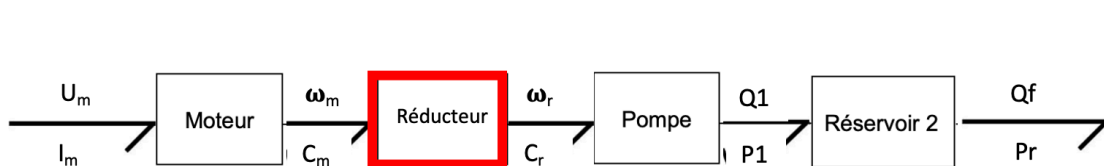


**Attention**

Attention aux unités :

- $C_{yl}$  [m<sup>3</sup>]
- $P_1$  [Pa]
- $C_r$  [N.m]

**2.4.3.4. Modéliser le réducteur**



**1** Ecrire l'équation permettant de relier la vitesse de rotation du réducteur à la vitesse de rotation du moteur.

On écrit le rapport de transmission du réducteur.

- Equation de comportement des variables **flux** du **réducteur** :

$$r = \frac{\omega_r}{\omega_m}$$

La variables **flux** d'entrée est  $\omega_m$  et la variable **flux** de sortie est  $\omega_r$ .

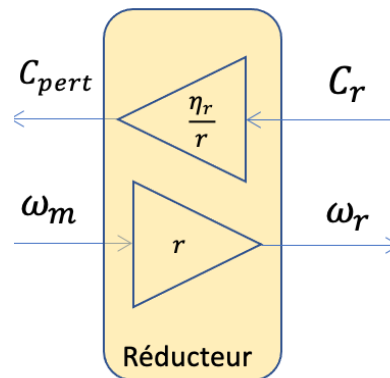
**2** Ecrire l'équation permettant de relier le couple appliqué en entrée du réducteur  $C_m$  au couple appliqué en sortie de réducteur  $C_r$ .

- Equation du comportement des variables **Effort** du **réducteur** :

$$\frac{C_r \cdot \omega_r}{C_m \cdot \omega_m} = \eta_r$$

La variable **effort** d'entrée est  $C_r$  [N.m] et la variable **effort** de sortie est  $C_m$  [N.m].

**3** Ecrire le bloc correspondant

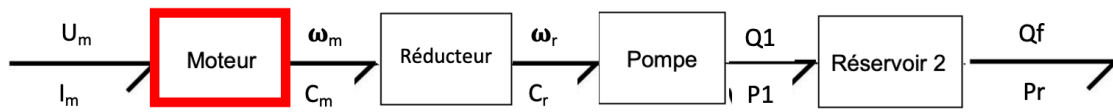


### ⚠ Attention

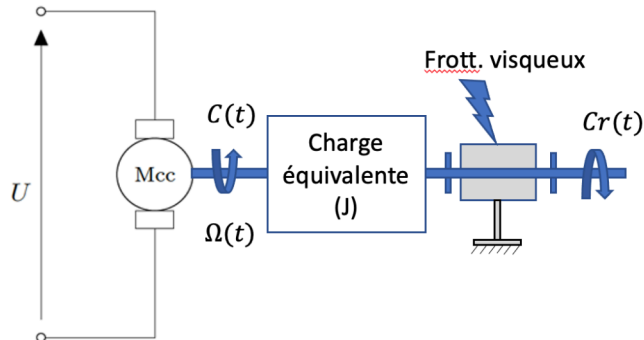
Attention aux unités :

- Oméga\_m [Rad/s]
- Oméga\_r [Rad/s]
- $C_m$  [N.m]
- $C_r$  [N.m]

## 2.4.3.5. Modéliser le moteur électrique



1 Réaliser le schéma mécanique du moteur.



2 Ecrire l'équation mécanique du moteur.

### Truc & astuce

On écrit le théorème du moment dynamique en projection sur l'axe moteur.

- Equation du comportement dynamique de l'arbre moteur **flux** du **moteur** :

$$C(t) - C_r(t) - f\Omega(t) = J \frac{d\Omega(t)}{dt}$$

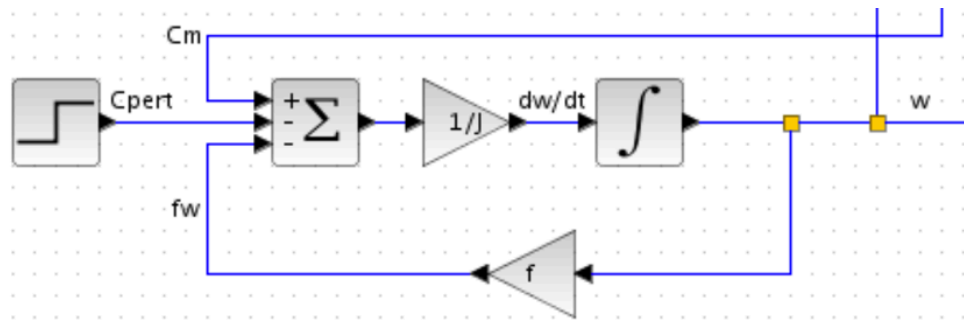
Coëf. De Frott. visqueux

On en déduit l'expression de  $\omega(t)$  en fonction de  $C_m(t)$  et de  $C_{pert}$  :

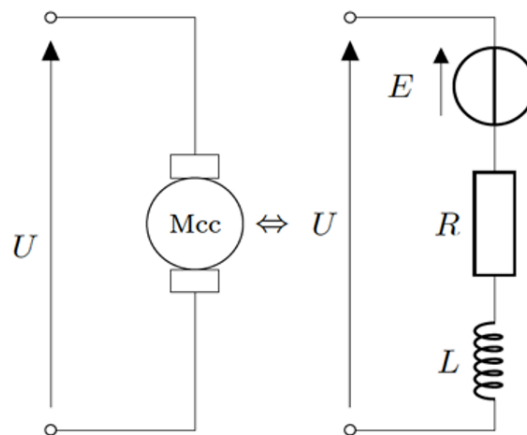
$$\omega = \frac{1}{J} \int (C_m - C_{pert} - f \cdot \omega) dt$$

La variable **flux** de sortie est  $\omega_m$  et la variable **effort** d'entrée est  $\omega_r$ .

3 Réaliser le schéma bloc associé.



4 Réaliser le schéma mécanique du moteur.



5 Ecrire les équations électriques du moteur.

On écrit la loi des mailles :

- Equation du comportement des variables **effort** du **réducteur** :

$$U(t) = E(t) + R.I(t) + L \frac{dI(t)}{dt}$$

On en déduit l'expression de  $I(t)$  en fonction de  $U(t)$  :

$$I = \frac{1}{L} \int (U - E - R.I) dt$$

La variable **flux** d'entrée est  $U(t)$  [Volt] et la variable **effort** de sortie est  $I$  [Ampères].

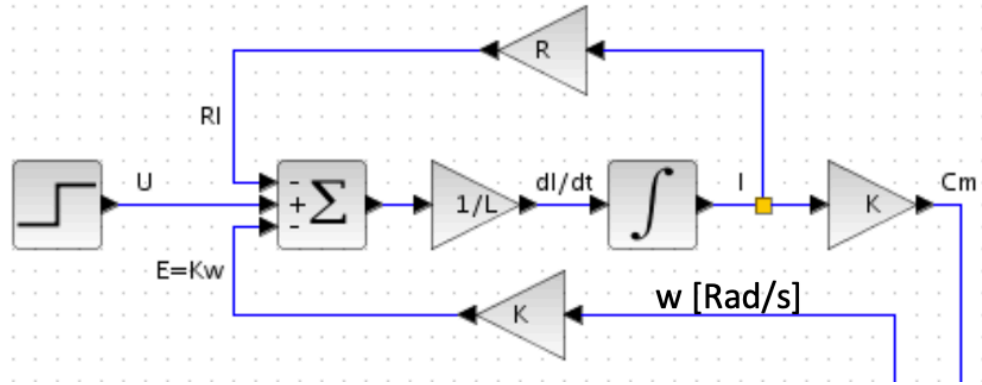
6 Ecrire les équations électro-mécaniques du moteur.

- Equation du comportement des variables **effort** du **réducteur** :

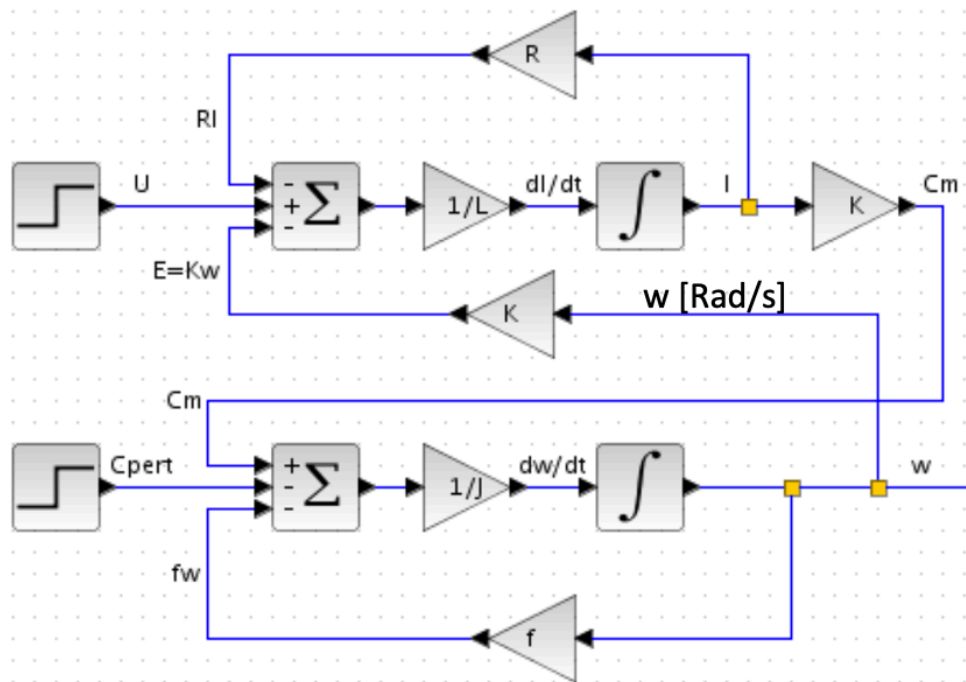
$$\begin{aligned} E(t) &= K\Omega(t) \\ C(t) &= KI(t) \end{aligned}$$

La variable **effort** d'entrée est  $C_r$  [N.m] et la variable **effort** de sortie est  $C_m$  [N.m].

**7** Réaliser le schéma bloc associé.

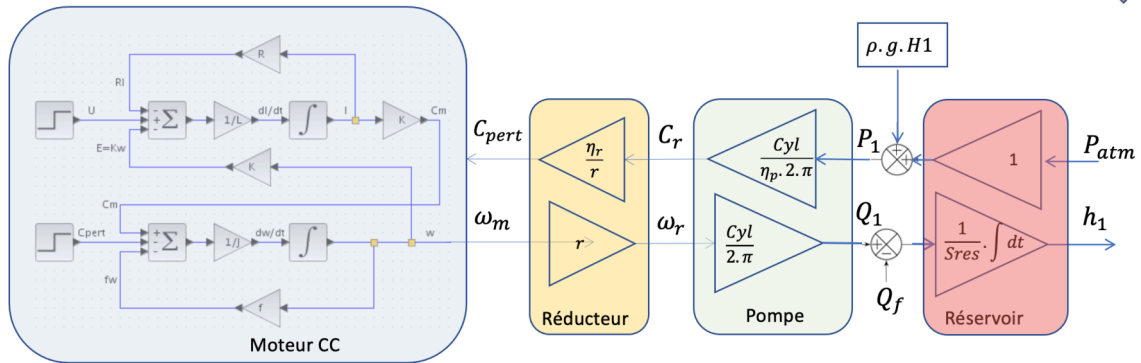


**8** Réaliser le schéma bloc du moteur courant continu correspondant.

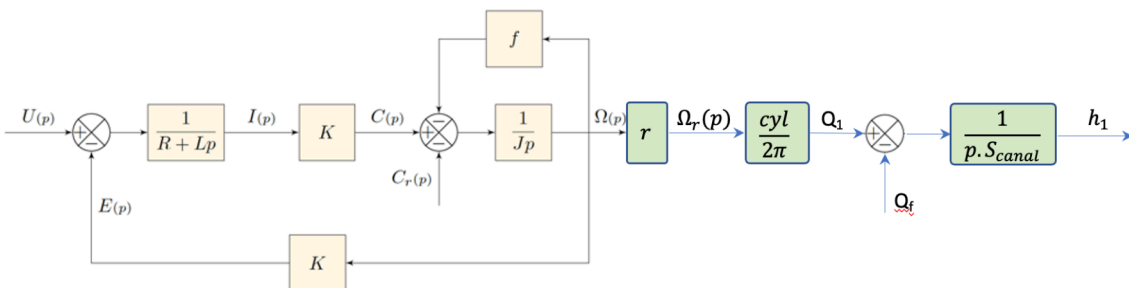


### 2.4.3.6. Synthèse

## Modèle mathématique causal temporel



## Modèle mathématique causal fréquentiel



### 2.4.4. Méthode de modélisation par schéma blocs

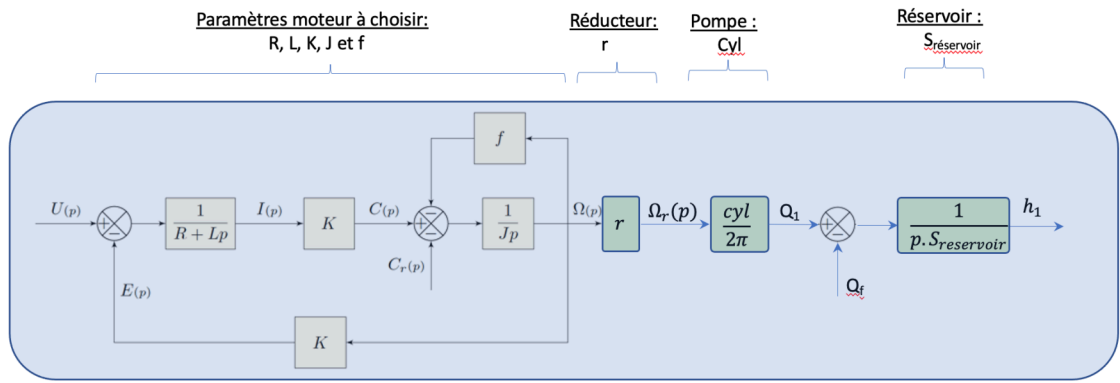


#### 1 Choix des paramètres de blocs

Pré-dimensionnement de chaque bloc

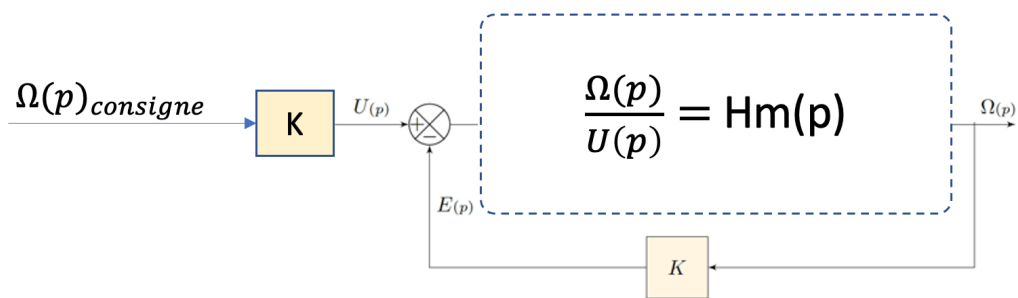
Etude spécifique par domaine pour caractériser chaque bloc, et évaluer la valeur des paramètres localisés :

- Pour le moteur :  $R$ ,  $L$ ,  $K$ ,  $J$  et  $f$
- Pour la pompe :  $Cyl$
- Pour le réducteur :  $r$
- Pour le réservoir :  $S_{reservoir}$



**2** Mise en place d'un asservissement, si nécessaire

**Exemple** Asservissement en vitesse d'un moteur à courant continu

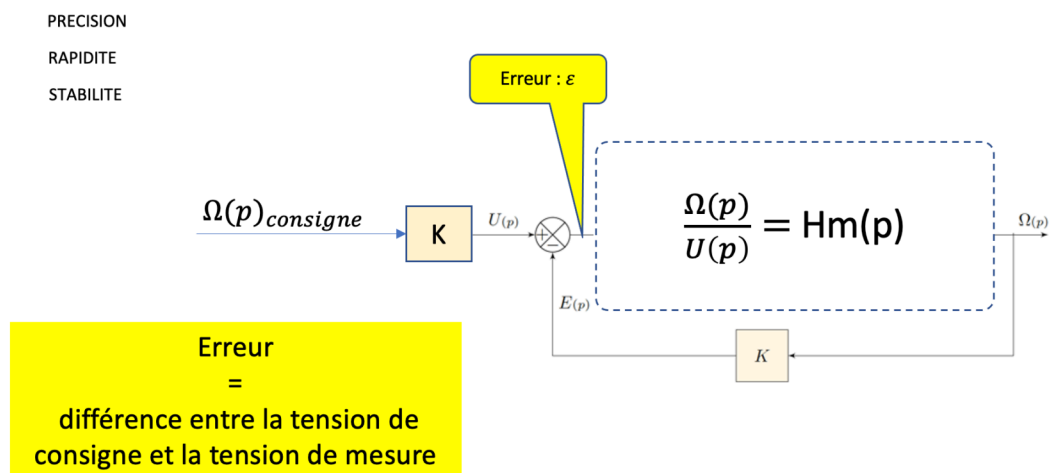


**3** Etude des performances du système

Les performances mesurées

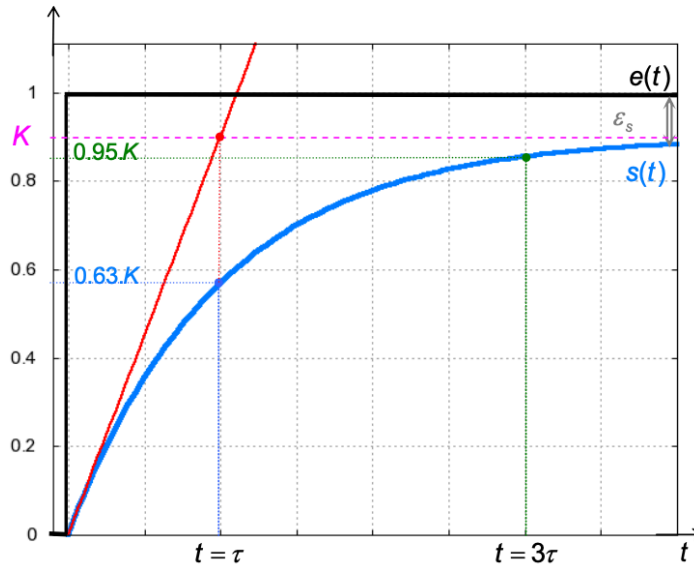
- Rapidité
- Précision
- Stabilité

**Exemple** Architecture d'un système asservi





### Exemple Réponse d'un système asservi du premier ordre



Avec :

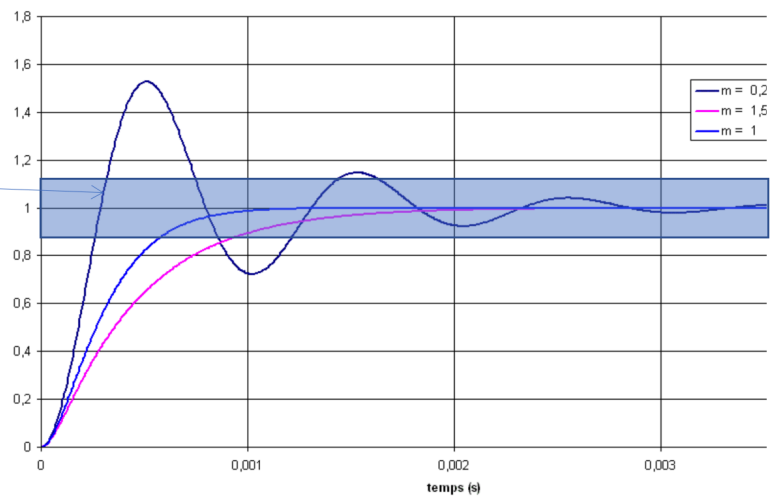
- Epsilon : erreur statique du système
- Tau : Temps de réponse à 5%
- K : Gain statique

### Exemple Réponse d'un système asservi du second ordre

- PRECISION
- RAPIDITE
- STABILITE

Tolérance

Réponse indicielle d'un système du 2ème ordre

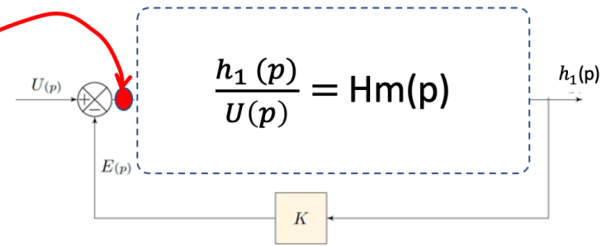


## Optimisation des performances

### Etude de sensibilité des paramètres

- Sur la commande
  - PRECISION
  - RAPIDITE
  - STABILITE
- En agissant sur les paramètres (sensibles) du modèle

### Insertion et réglage d'un correcteur SI NECESSAIRE



## 2.5. Ce qu'il faut retenir

### Avantages :

- La modélisation repose sur des lois physiques
- Les équations peuvent parfois être simplifiées
- La maîtrise des hypothèses utilisées permet d'établir les équations
- Temps de calcul plus rapides

Très utile lorsque l'amplitude de la réponse oscille autour de l'équilibre = Systèmes asservis

### Inconvénients :

- Equations mathématiques souvent lourdes et difficiles à établir
- **La relation de causalité entre l'entrée et la sortie peut être un frein à l'optimisation des paramètres sensibles.**

## 3. Modélisation orientée objet

---

### 3.1. Introduction

---

#### Objectifs pédagogiques

L'objectif de module est de présenter un logiciel de simulation multi-physique **permettant de construire un modèle sans en écrire les équations**.

On y découvre les étapes de construction d'un modèle multi-physique d'un système complexe, en vue de simuler son comportement.

A l'issue de ce module, les étudiants doivent être capables de :

- Choisir les composants du modèle dans une bibliothèque
- Assembler les différents composants
- Caractériser ces composants
- Simuler son fonctionnement
- Analyser son comportement

#### **Déroulement** Présentation rapide des différentes étapes

1. Présentation du système étudié
2. Présentation du problème technique étudié
3. Présentation du logiciel utilisé et de sa bibliothèque de composants
4. Fonctionnement de la bibliothèque Modélica
5. Application

### 3.2. Exemple d'application : ascenseur grande vitesse

---

#### Contexte

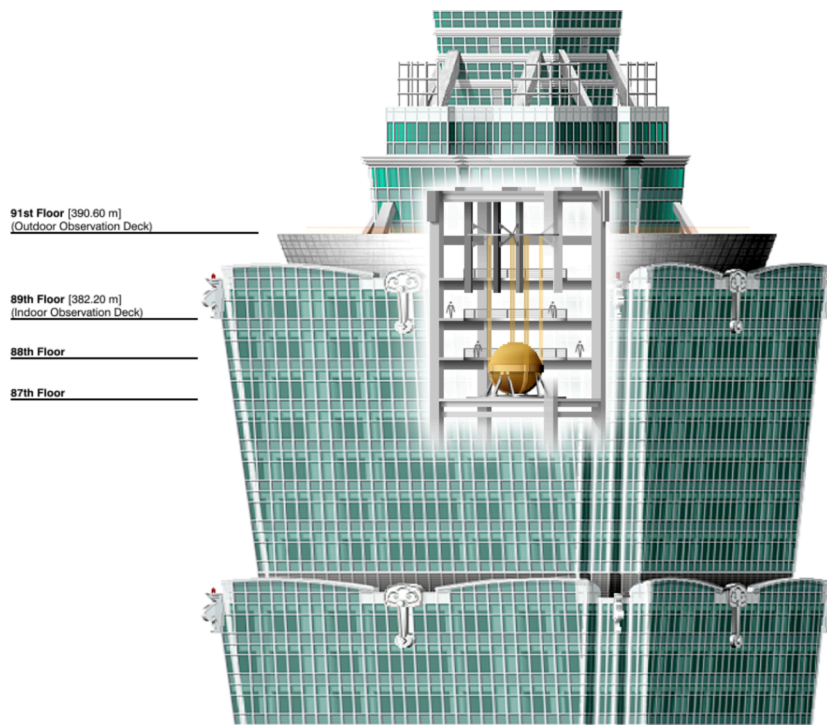
Le Taipei 101 est le premier immeuble à dépasser le demi kilomètre de hauteur (508 mètres).



*Tour Taipei 101*

Du fait de ce record de hauteur :

- La tour a besoin d'un **dispositif de stabilisation massif** pour faire face aux vents et aux séismes. Celui-ci prend la forme d'une énorme boule d'acier de 800 tonnes, accrochée en haut de l'immeuble et visible du restaurant et du bar, qui encerclent l'espace autour du dispositif.



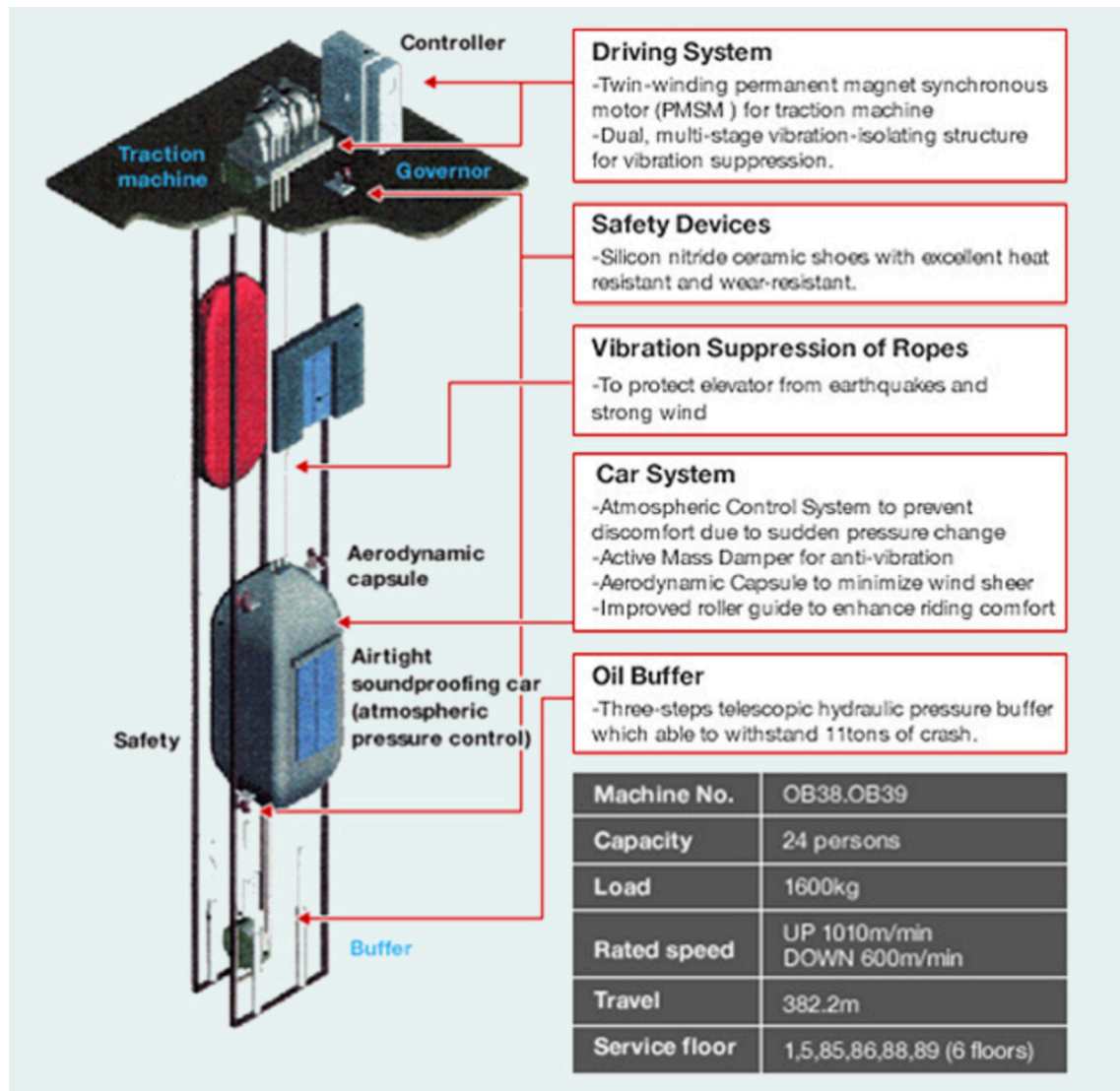
*Taipei 101: Une pendule comme amortisseur harmonique dans la tête de la tour*

- On a aussi besoin d'**ascenseurs performants** pour assurer un déplacement rapide des personnes du rez-de-chaussée au sommet en garantissant un confort optimal.

### Performances attendues des ascenseurs pour l'observatoire

Les ascenseurs desservant les 101 étages du Taipei 101 doivent atteindre une vitesse maxi en montée de 1010 mètres par minute (60.60 km/h) et descendant à 610 mètres par minute (36.60 km/h).

Ces performances sont limitées par des normes d'accélération (environ  $< 1\text{m/s}^2$ ), de pressions acoustiques et de vibrations maximales autorisées.



Ascenseur ultra haute vitesse Toshiba, installé au sein de la tour Taipei 101

Chaque ascenseur doit transporter jusqu'à 24 personnes du rez-de-chaussée au 101<sup>ème</sup> étage.

### ⚠ Attention

Les objectifs d'études peuvent être multiples.

En effet, selon si on veut dimensionner le système de levage ou le système de mise en pression de l'intérieur de la cabine, le modèle sera totalement différent.

L'objectif de cette simulation système est de construire un modèle numérique permettant de simuler le comportement dynamique du système de levage.

## 3.3. Etude proposée : vérifier la performance de la cabine d'ascenseur

## Procédure : Vérifier la performance du système : la cabine d'ascenseur

Lors du fonctionnement du système, on souhaite obtenir un mouvement le plus fluide possible, soit sans oscillation.

### **1** Evaluation du couple moteur nécessaire pour déplacer la cabine à la vitesse souhaitée

Construction du mécanisme

On valide la cinématique du mécanisme.

On impose un flux et on observe l'effort moteur.

Cette étape nous permettra de choisir le moteur dans un catalogue.

### **2** Tester le comportement du moteur CC

On teste le fonctionnement du moteur seul.

Cette étape nous permet de valider les performances du moteur.

### **3** Assemblage des deux modèles

On teste le système au regard des performances attendues par le cahier des charges.

Cette étape nous permet de vérifier que l'ensemble fonctionne correctement et correspond aux attentes du cahier des charges.

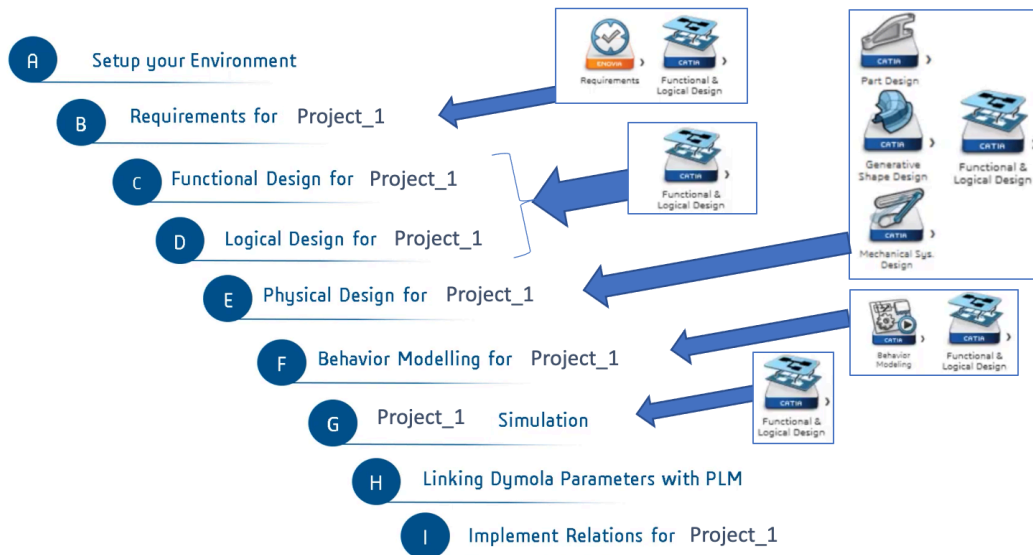
### **4** Analyse des performances

Impact de la rigidité de la mécanique sur le système

Dans cette étude, on va mettre en évidence l'impact de la rigidité de la transmission sur le déplacement de la cabine.

## 3.4. Mise en place d'un environnement DYMOLA dans 3DEXperience

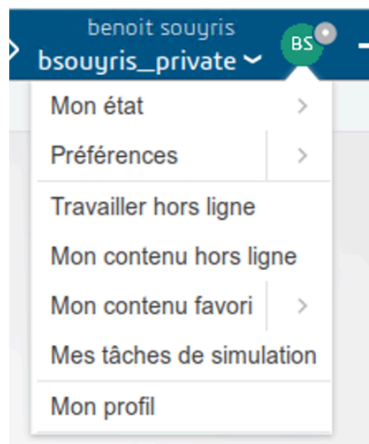
### 3.4.1. Les étapes de conception avec 3DEXperience



### 3.4.2. Configuration de Dymola

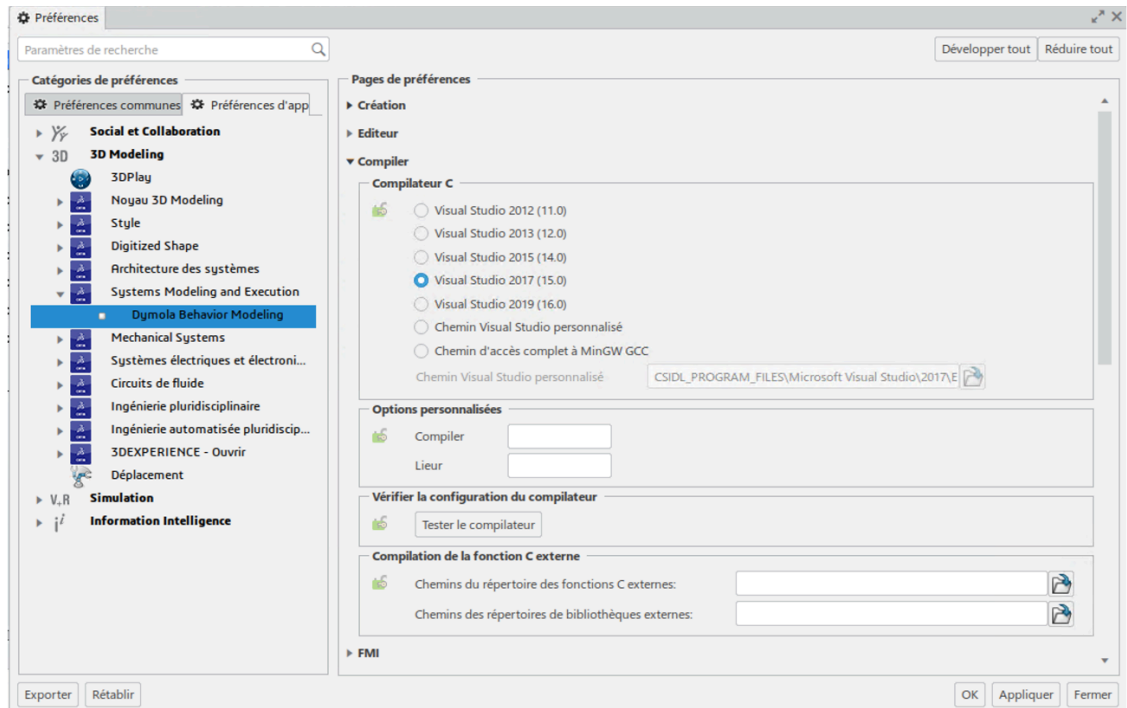
**Procédure :** Choisir le compilateur d'équations et sélectionner les bibliothèques utilisées

**1** Clic-droit sur le profil





## 2 Sélectionner Préférences

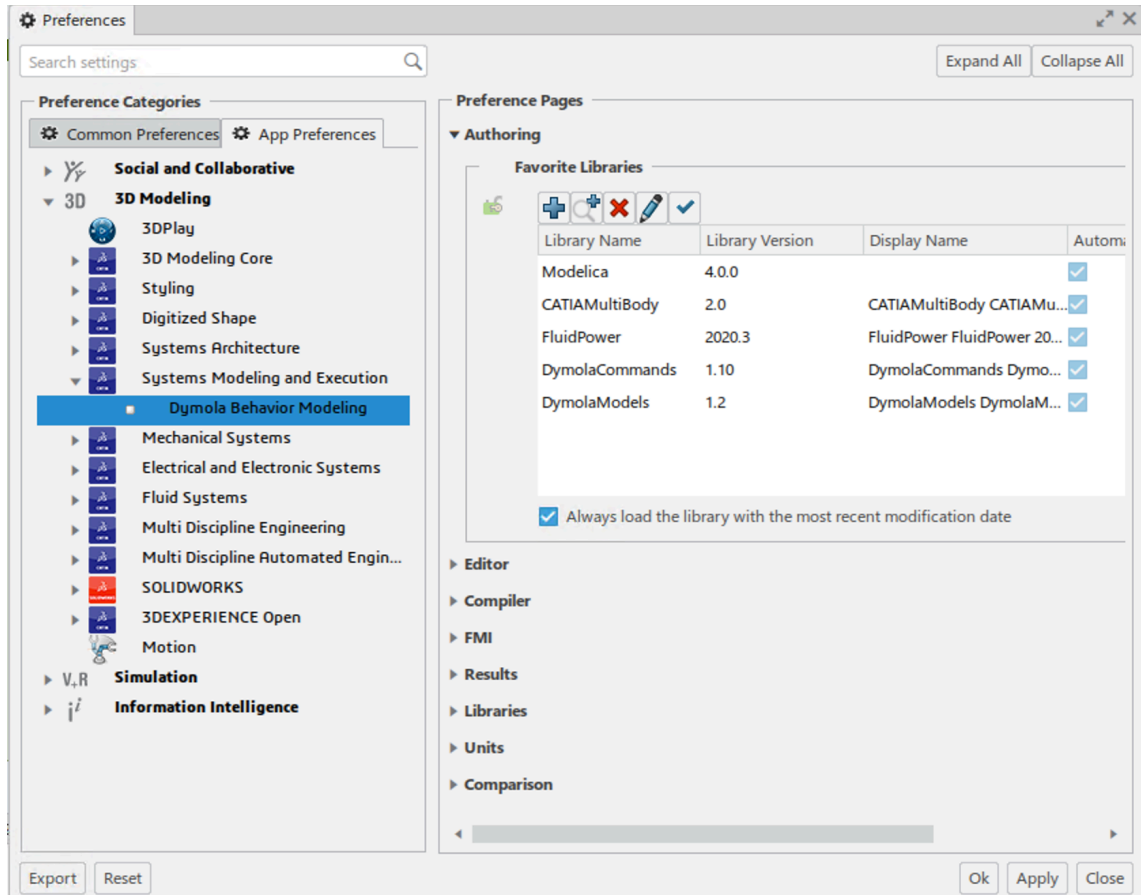


## 3 Ouvrir le panel Compiler et choisir : Visual Studio 2017

## 4 Tester le compilateur

## 5 Ouvrir le panel Authoring et ajouter les bibliothèques à utiliser

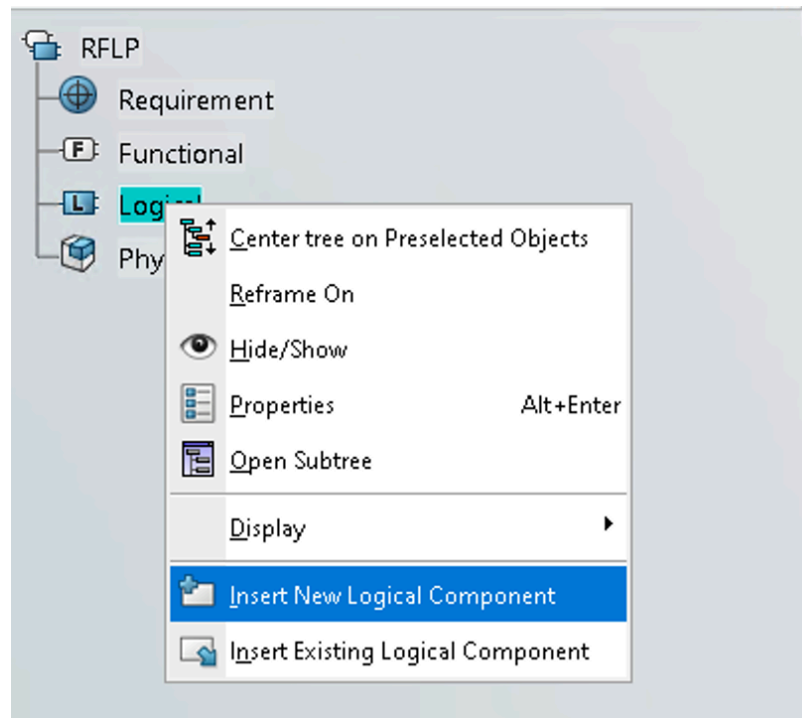
- Modelica
- DymolaCommands
- DymolaModels



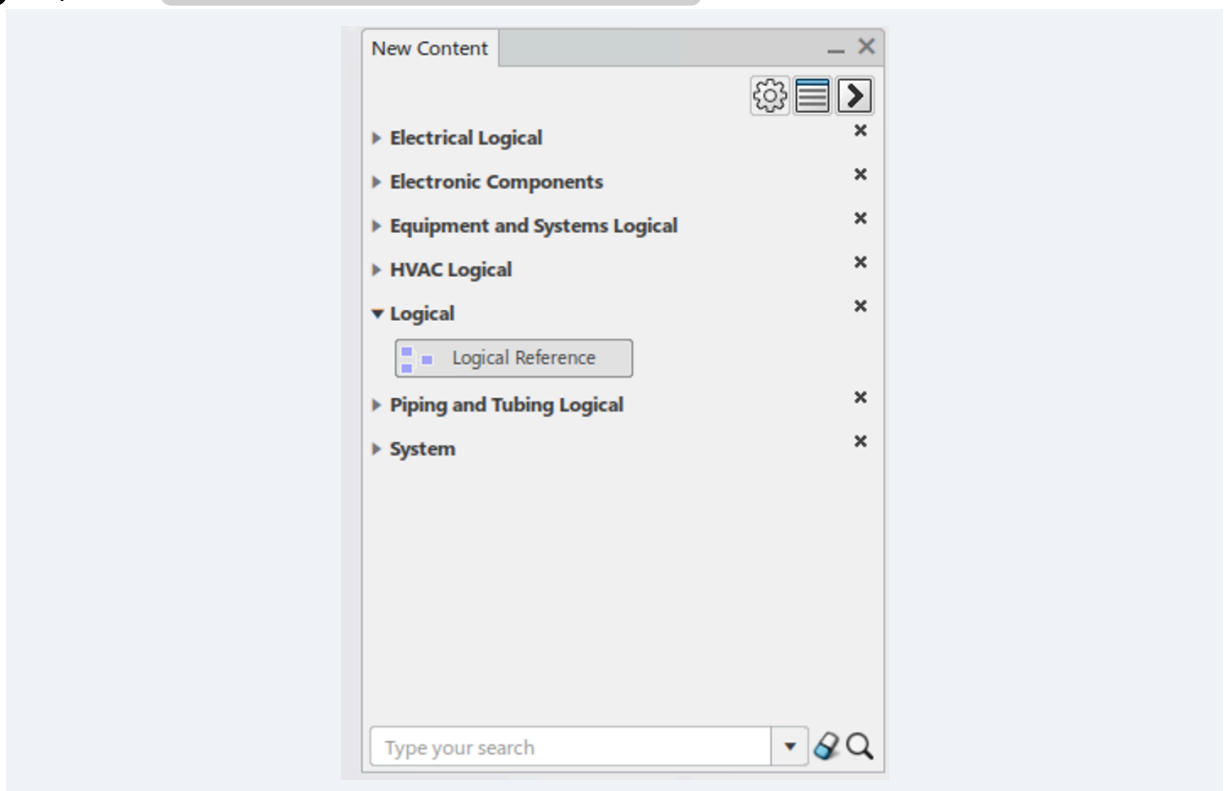
### 3.4.3. Créer le cadre de l'étude : espace logique de référence

## ❗ Procédure : Insérer un espace logique pour définir le système

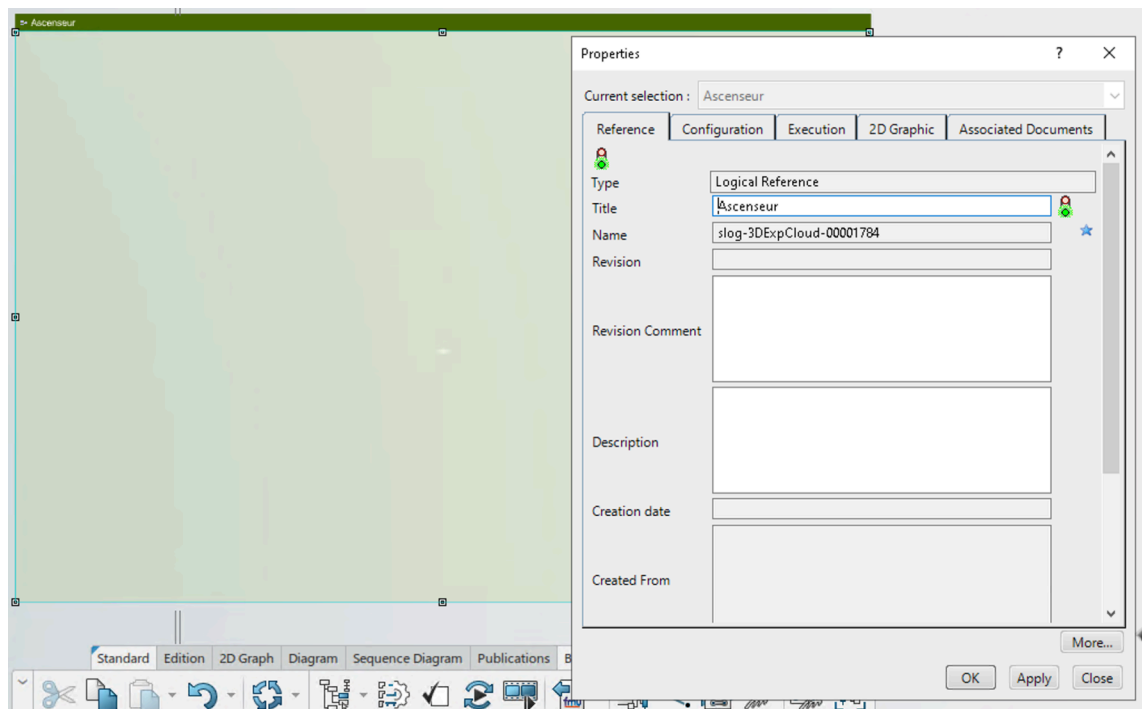
1 Clic-droit sur **Logical**



2 Cliquer sur **Insert new logical component**



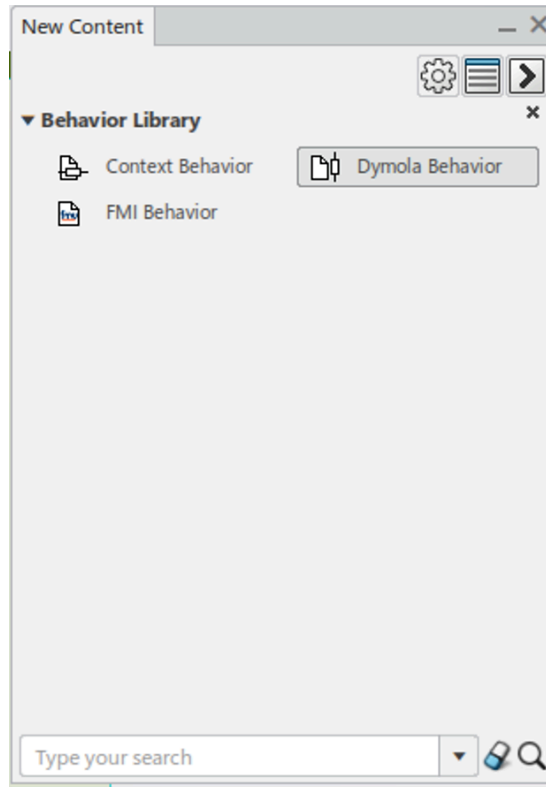
3 Sélectionner **Logical reference**

**4** Clic-droit sur la référence créée dans l'arbre de création, et changer le nom du projet

### 3.4.4. Créer un modèle Dymola

#### Procédure : Etapes de création d'un modèle Dymola

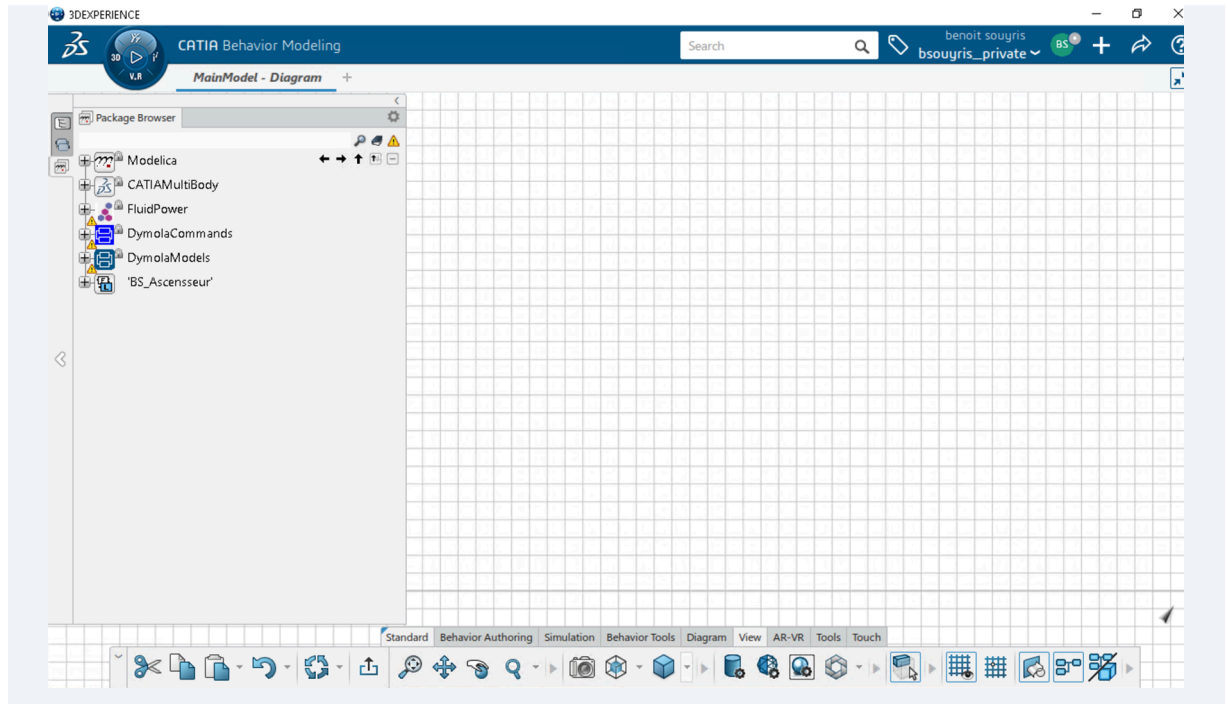
**1** Cliquer sur le cadre vert avant de cliquer sur l'icône suivante :

**2** Sélectionner `Dymola Behavior`

Le « totem » ci-dessous apparaît dans le coin bas gauche du rectangle vert :

**3** Cliquer sur le « totem » puis double-cliquer sur `MainModel`

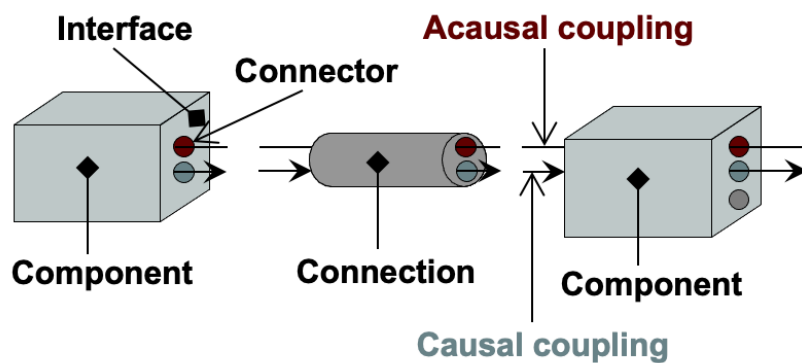
Ouverture de Dymola :



## 3.5. Modélisation avec DYMOLA

### 3.5.1. Comment se présente un modèle sur Dymola ?

Réseaux de composants orientés objets

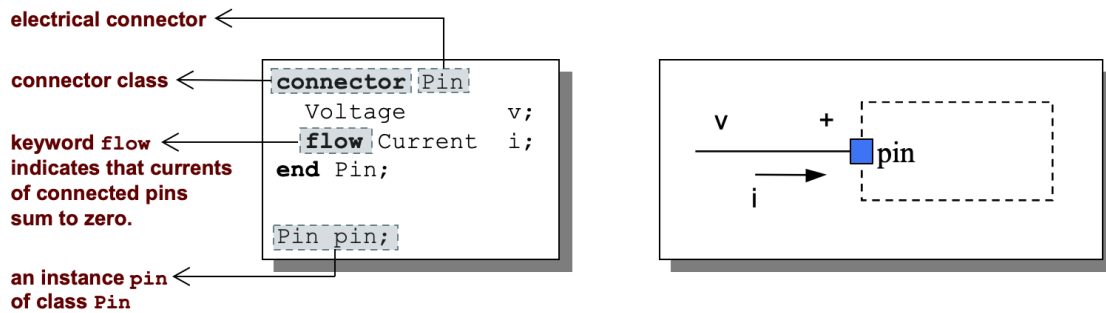


Une classe de composants doit être définie indépendamment de son environnement, c'est essentiel pour sa réutilisation.

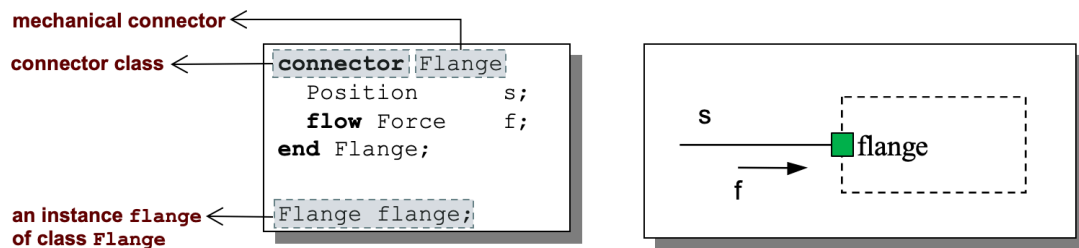
Les systèmes complexes sont généralement constitués d'un grand nombre de composants connectés.

### 3.5.2. Les liens par domaine

### Connecteur Pin : composant Électrique



### Connecteur Flange : composant Mécanique

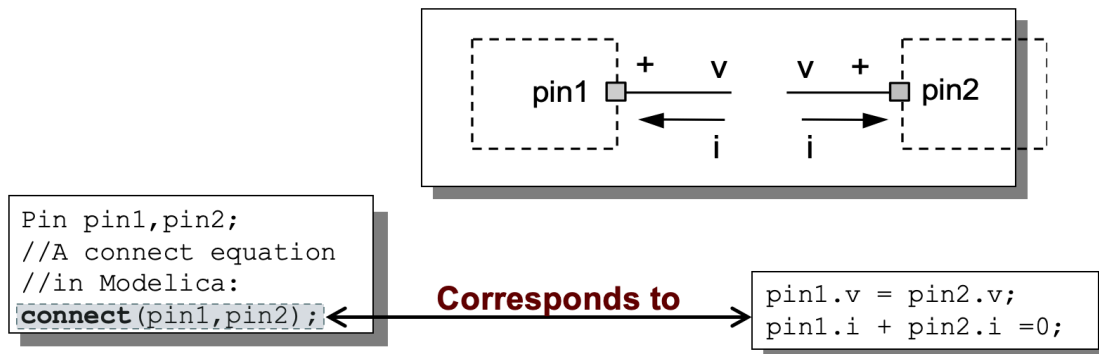


### Autres connecteurs, par domaine

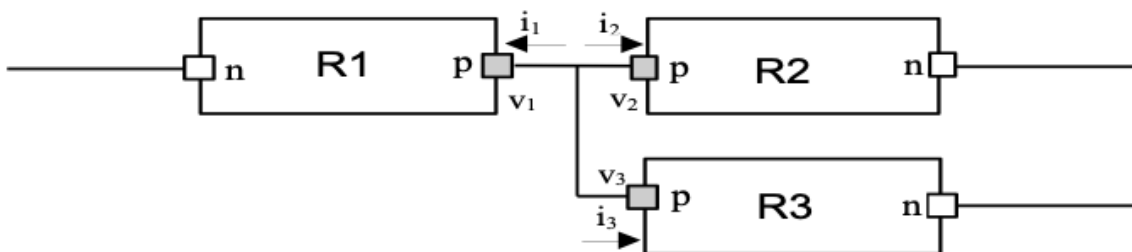
Domaine	Potentiel	Flux	Connexion	Modelica Librairie
Electrical	Voltage	Current		Electrical. Analog
Translational	Position	Force		Mechanical. Translational
Rotational	Angle	Torque		Mechanical. Rotational
Magnetic	Magnetic potentiel	Magnetic flux rate		Magnetic
Hydraulic	Pression	Volume flow		OpenHydraulics
Heat	Temperature	Heat flow		HeatFlow1D
<u>Mathematic</u>		signal		Math

## 3.5.3. Equations de liaison

### Lois de Kirschoff



◆ **Remarque** Les connections multiples sont possibles

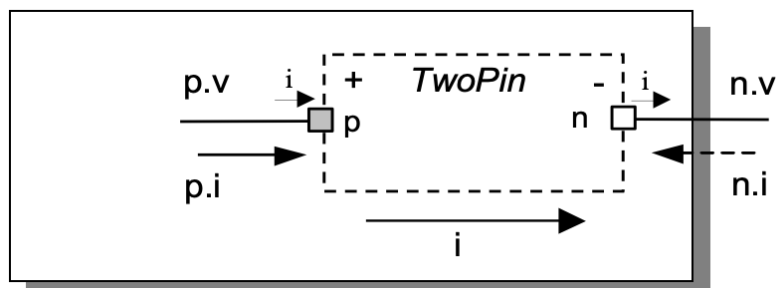


- A chaque nœud de **potentiel**, on peut écrire la loi des Mailles suivante :  
 $V_1 = V_2 = V_3 = \dots = V_n$
- A chaque nœud de **flux**, on peut écrire la loi des nœuds suivante :  
 $i_1 + i_2 + i_3 = 0$

### 3.5.4. Structure d'un composant à deux ports (ou bornes)

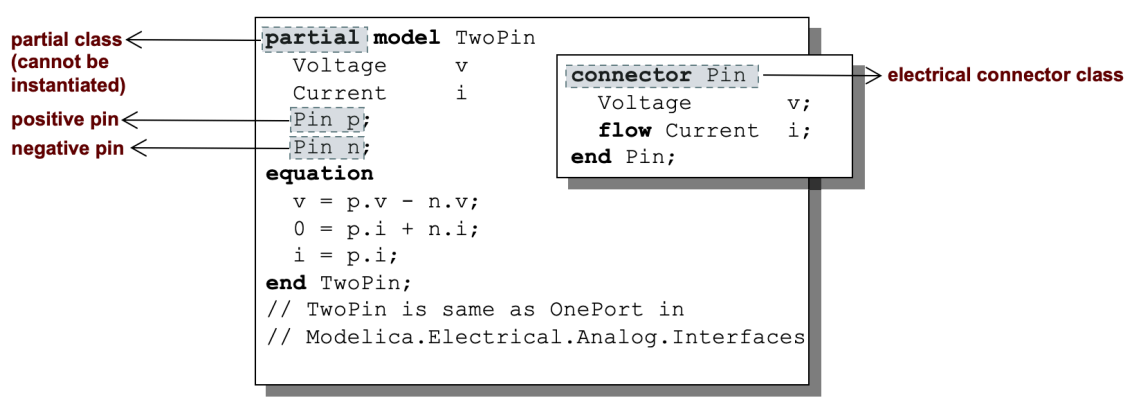
Traitement des infos venant des ports

Les deux ports d'un tel composant seront appelés bornes **p** (positive) et **n** (négative) :



Code associé :



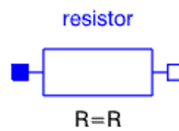


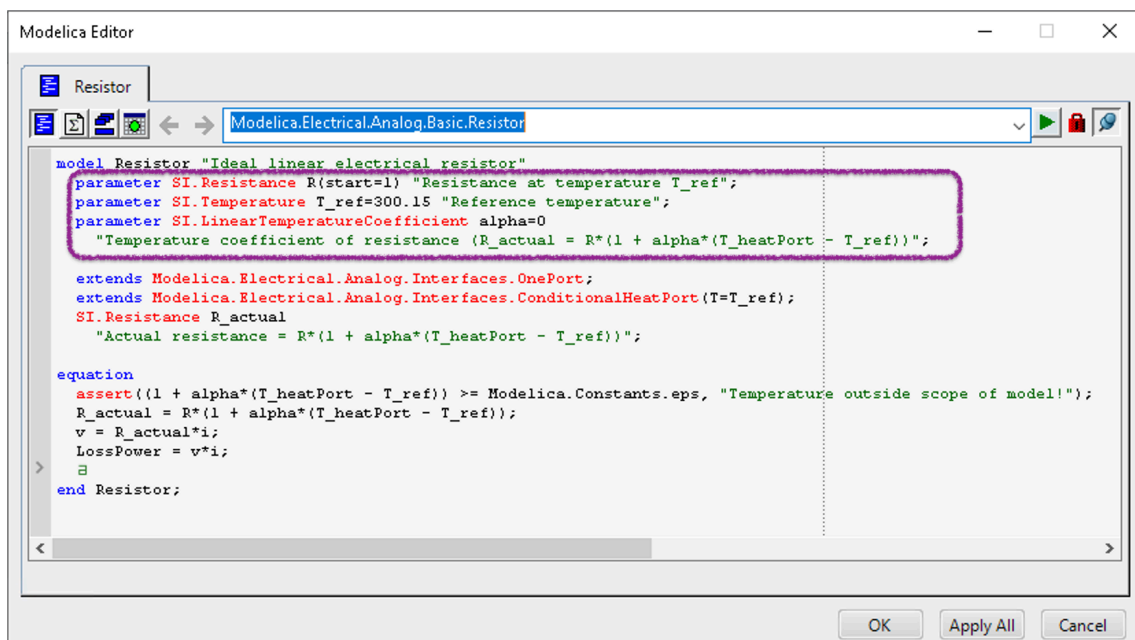
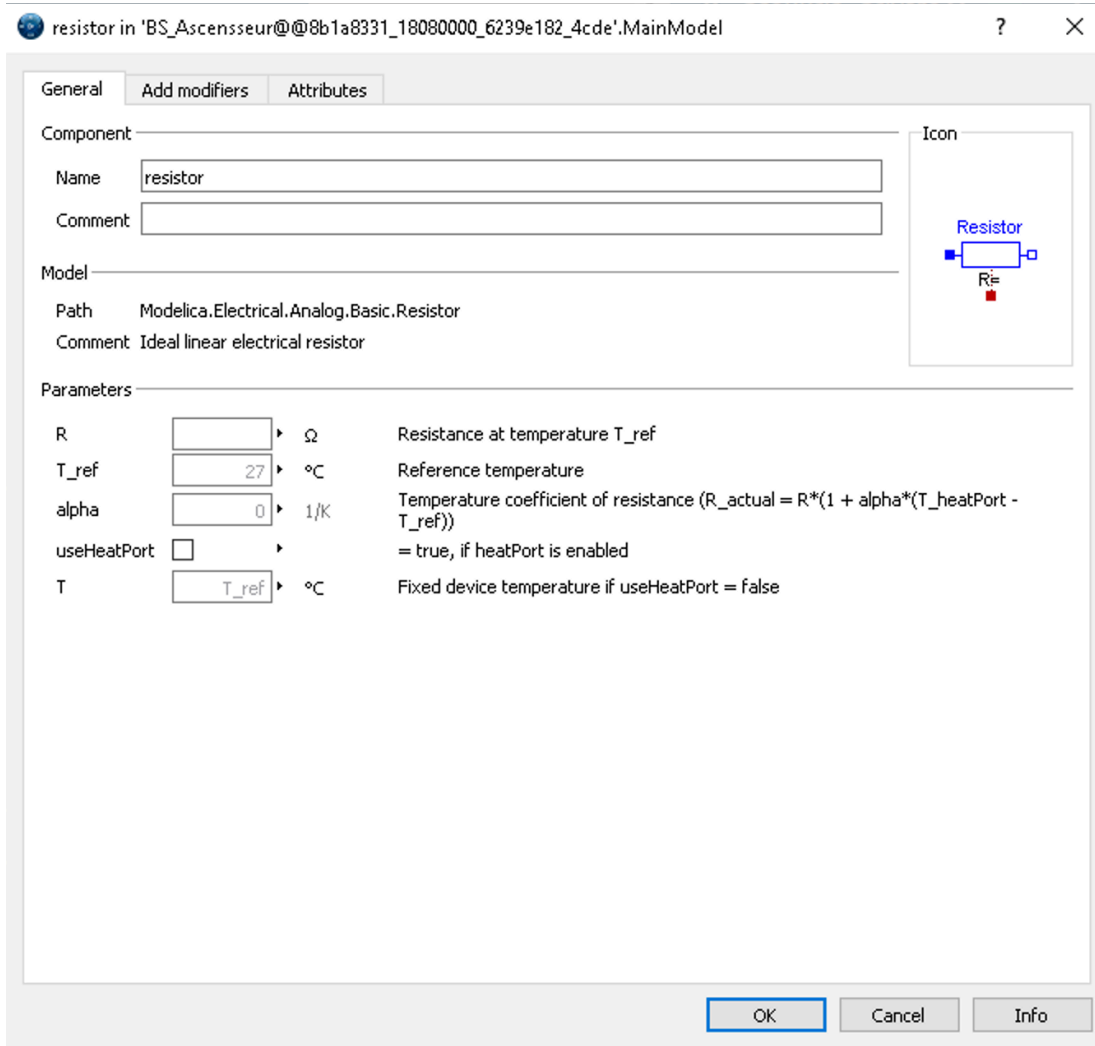
### 3.5.5. Paramétrage interne des composants

#### Paramètres des composants

Ce sont les valeurs qui caractérisent le composant.

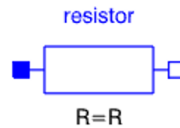
#### Exemple Paramètres internes d'une résistance





### 3.5.6. Comportement d'un composant

## Exemple Equations de comportement d'un résistance électrique



```

Modelica Editor
Resistor
Modelica.Electrical.Analog.Basic.Resistor

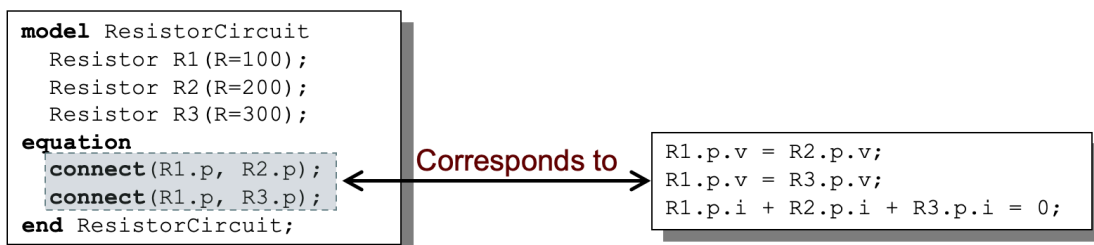
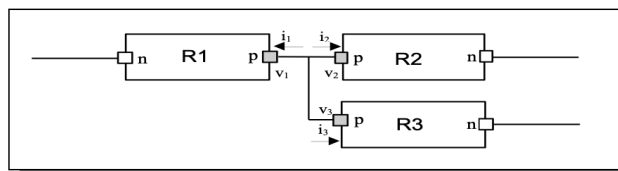
model Resistor "Ideal linear electrical resistor"
  parameter SI.Resistance R(start=1) "Resistance at temperature T_ref";
  parameter SI.Temperature T_ref=300.15 "Reference temperature";
  parameter SI.LinearTemperatureCoefficient alpha=0
    "Temperature coefficient of resistance (R_actual = R*(1 + alpha*(T_heatPort - T_ref)))";

  extends Modelica.Electrical.Analog.Interfaces.OnePort;
  extends Modelica.Electrical.Analog.Interfaces.ConditionalHeatPort(T=T_ref);
  SI.Resistance R_actual
    "Actual resistance = R*(1 + alpha*(T_heatPort - T_ref))";

equation
  assert((1 + alpha*(T_heatPort - T_ref)) >= Modelica.Constants.eps, "Temperature outside scope of model!");
  R_actual = R*(1 + alpha*(T_heatPort - T_ref));
  v = R_actual*i;
  LossPower = v*i;
end Resistor;
  
```

## 3.5.7. Couplage entre composants élémentaires

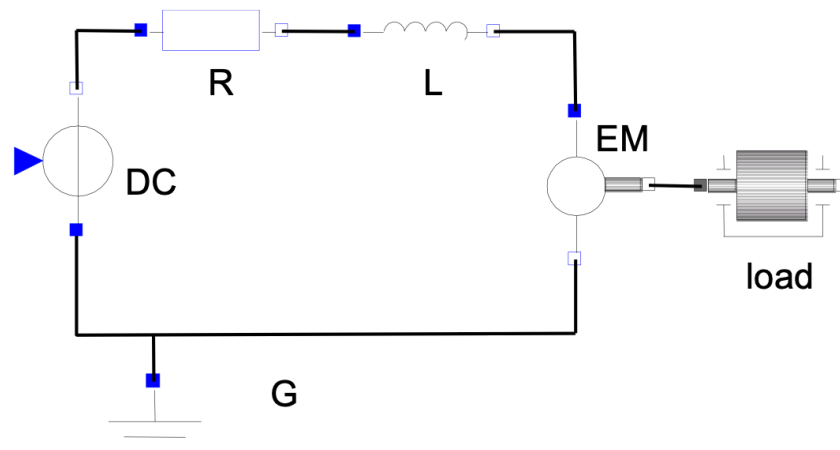
### Circuit électrique entre des résistances



### Couplage entre composants élémentaires

Les modèles acausaux<sup>[p.100]</sup> se présentent sous la forme d'un réseau d'objets connectés par des liens à double sens.

Ces liens relient les objets du réseau par des ports (connectors) placés sur les contours des objets et ayant généralement une forme et une couleur permettant d'identifier un domaine de la physique bien précis (Electricité, Mécanique, Hydraulique, Thermique, etc.).



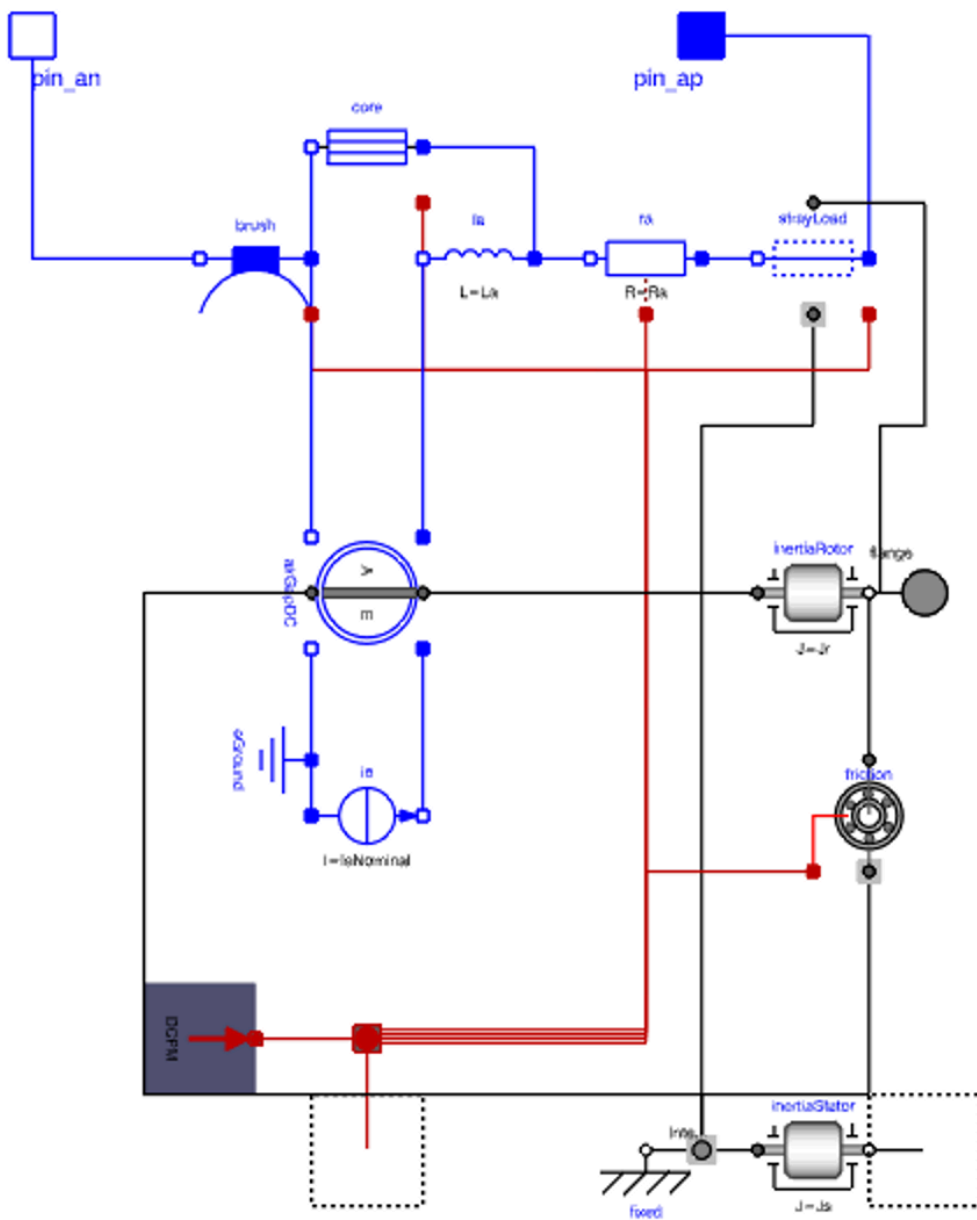
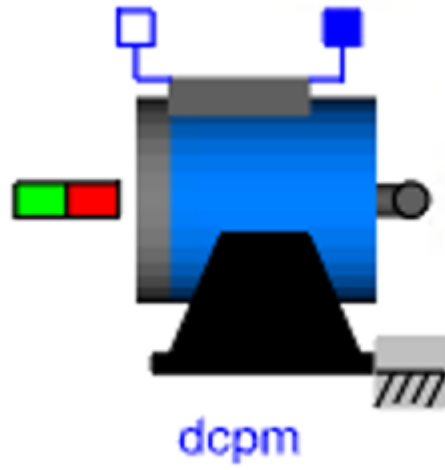
```

model DCMotor
  Resistor R(R=100);
  Inductor L(L=100);
  VsourceDC DC(f=10);
  Ground G;
  ElectroMechanicalElement EM(k=10,J=10, b=2);
  Inertia load;
equation
  connect(DC.p,R.n);
  connect(R.p,L.n);
  connect(L.p, EM.n);
  connect(EM.p, DC.n);
  connect(DC.n,G.p);
  connect(EM.flange,load.flange);
end DCMotor

```

### 3.5.8. Regroupement de composants

Un composant peut être constitué en interne d'autres composants, c'est-à-dire une modélisation hiérarchique.



## 3.6. Application : étude de la cabine d'ascenseur

### 3.6.1. Construction du système simplifié pour prédimensionnement



#### Hypothèses d'étude

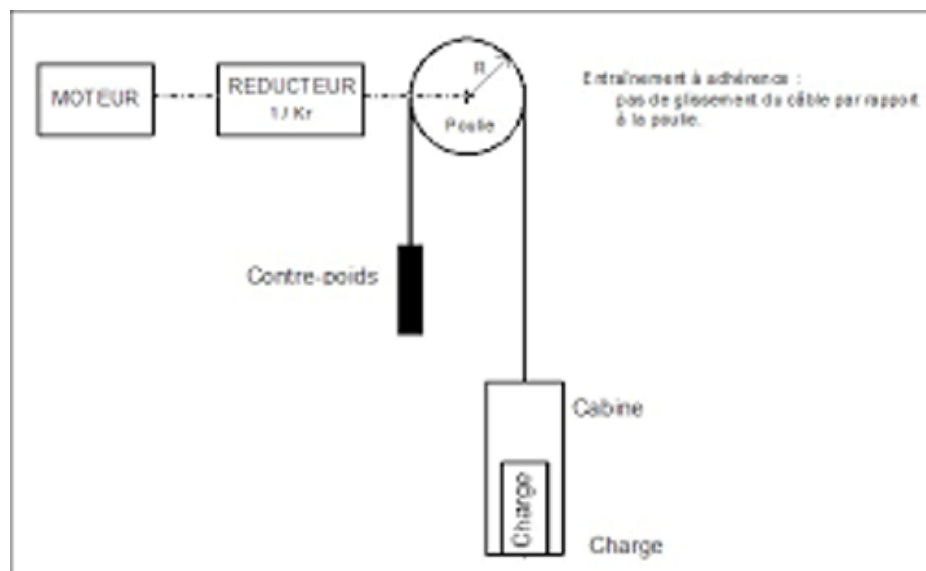
On néglige :

- Les effets aérodynamiques sur la cabine
- L'inertie des galets de guidage cabine

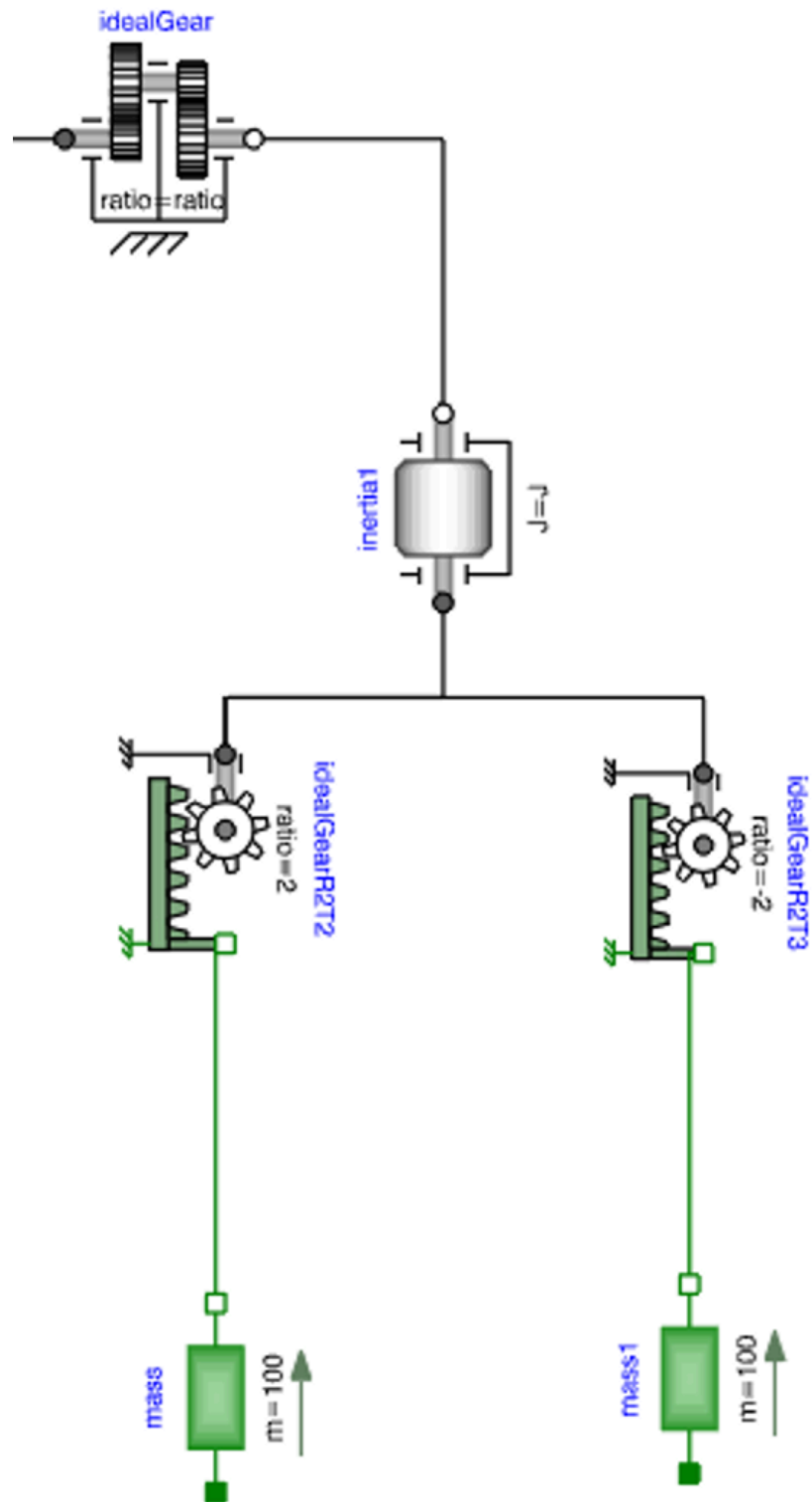
Rigidité de l'arbre poulie, non négligé

#### 1 Domaine mécanique

##### Schéma mécanique

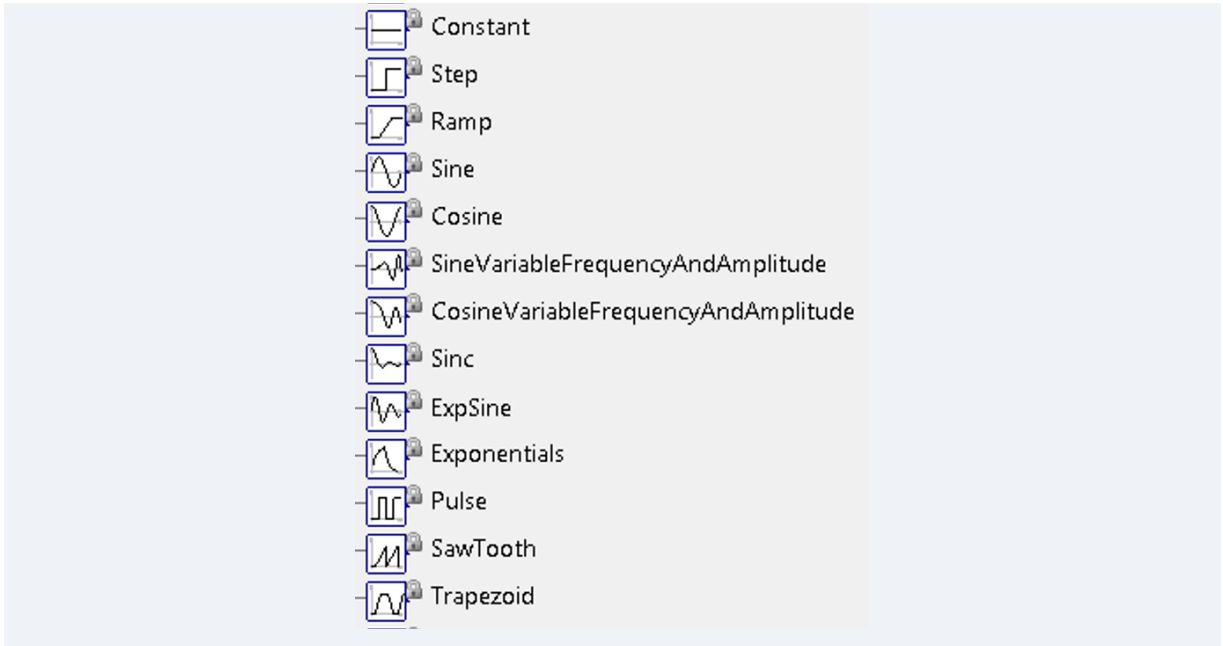


## Schéma Modélica



Test de performance du modèle mécanique simplifié

Choix d'un signal de consigne :



**2** Domaine électrique  
Schéma Électrique

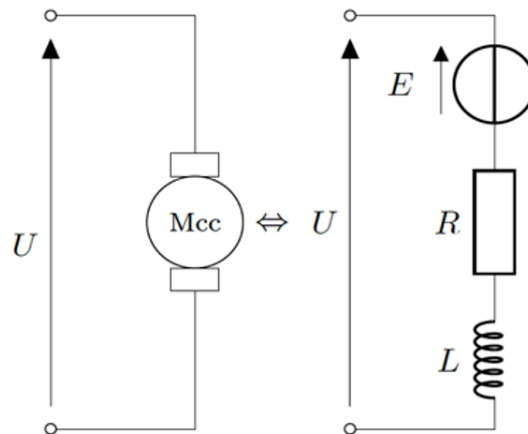
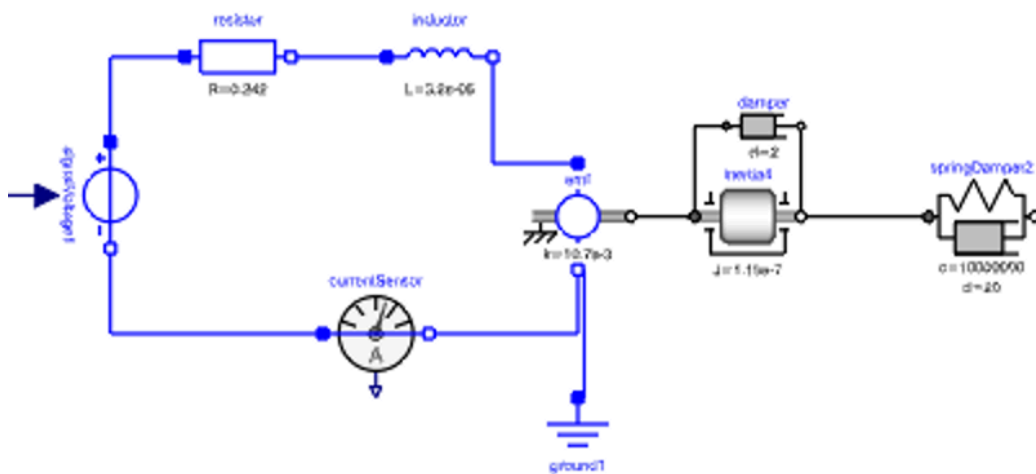


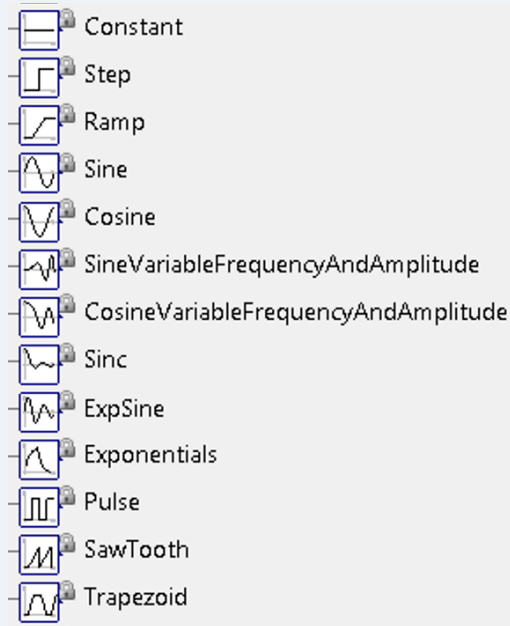
Schéma Modélica





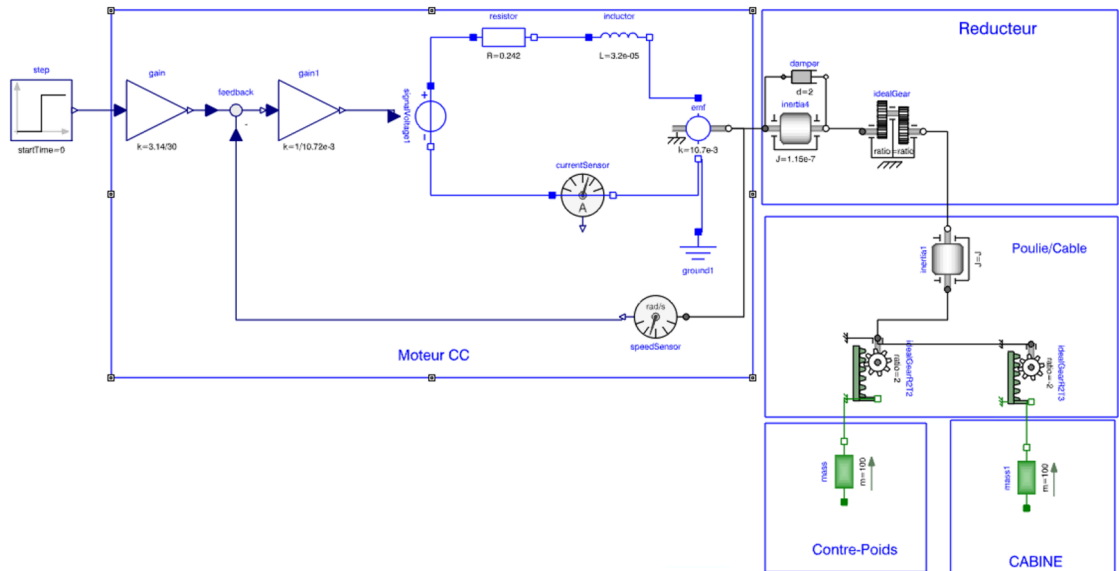
## Test de performance du modèle Moteur CC

Choix d'un signal de consigne :



### 3 Assemblage pour simulation du système complet (simplifié)

Grâce à cette maquette, on peut vérifier que le moteur pré-dimensionné correspond au cahier des charges et permet de déplacer la mécanique en **approchant** les performances attendues.



## 3.6.2. Construction du système complexe

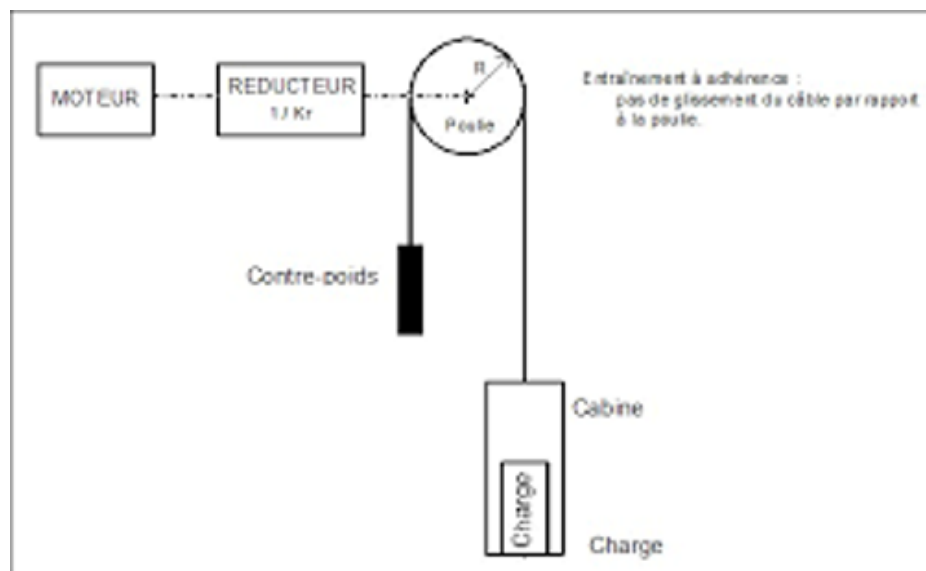


## Hypothèses d'étude

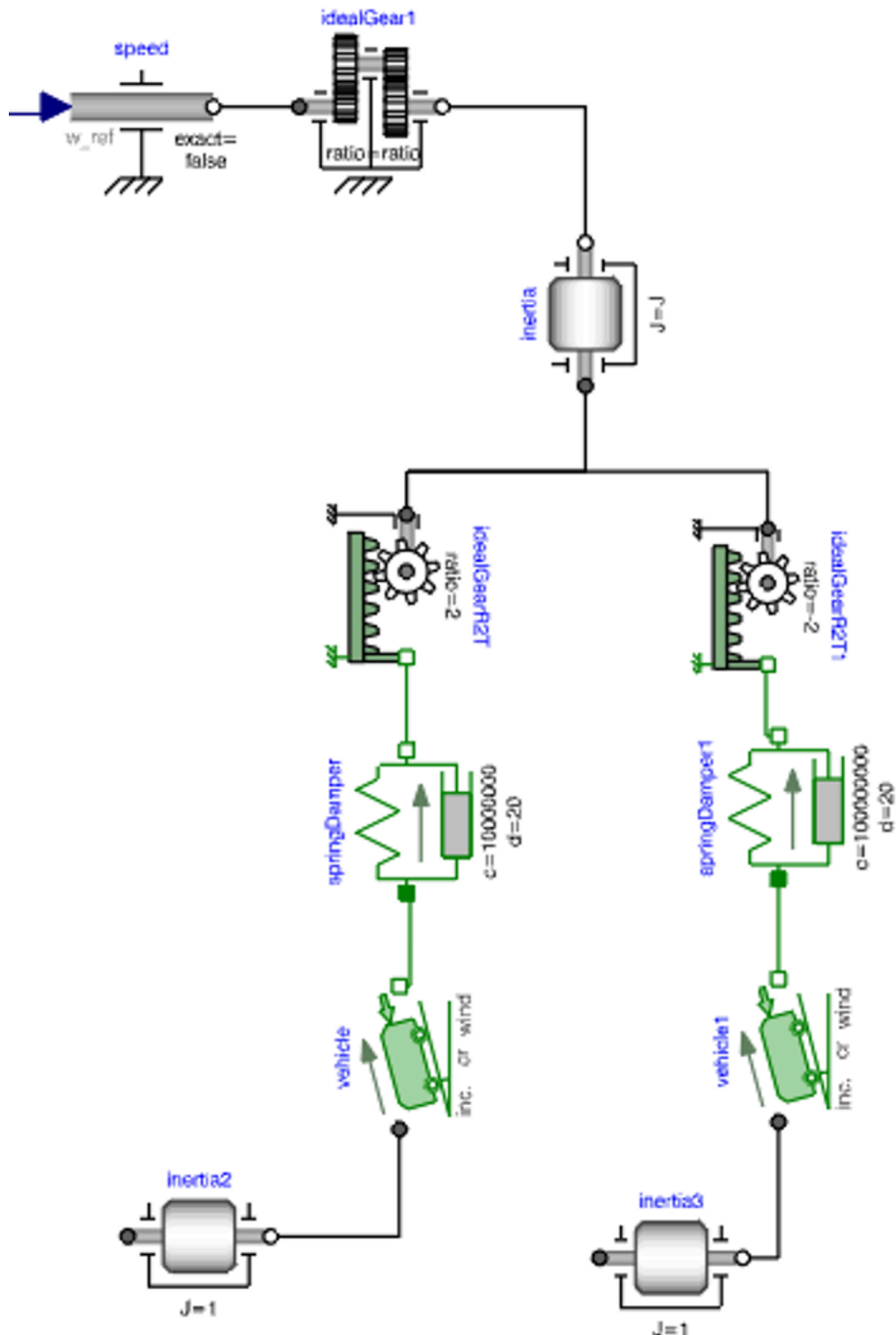
- Prise en compte des effets aérodynamiques sur la cabine
- Prise en compte de l'inertie des galets de guidage cabine
- Rigidité de l'arbre poulie, non négligé

### 1 Domaine mécanique

#### Schéma mécanique

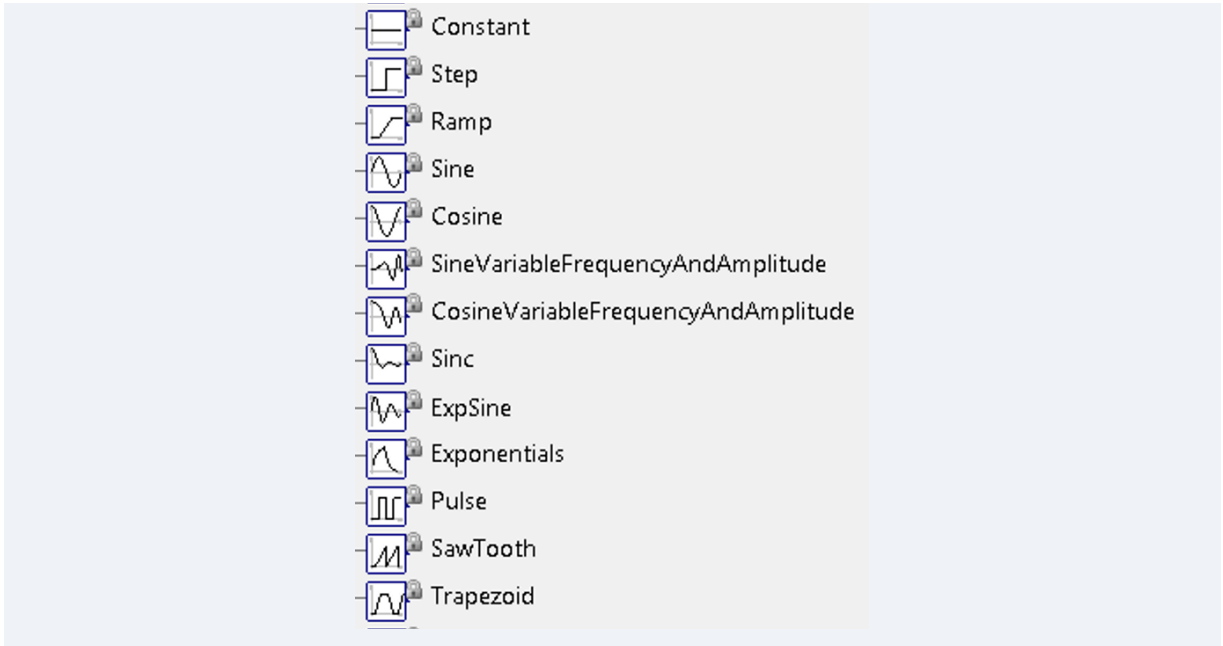


## Schéma Modélica



Test de performance du modèle mécanique simplifié

Choix d'un signal de consigne :



**2** Domaine électrique  
Schéma Électrique

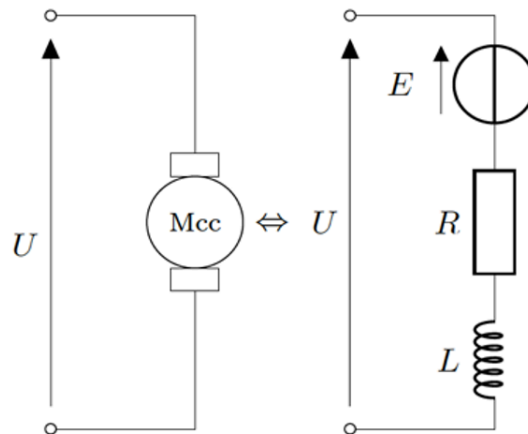
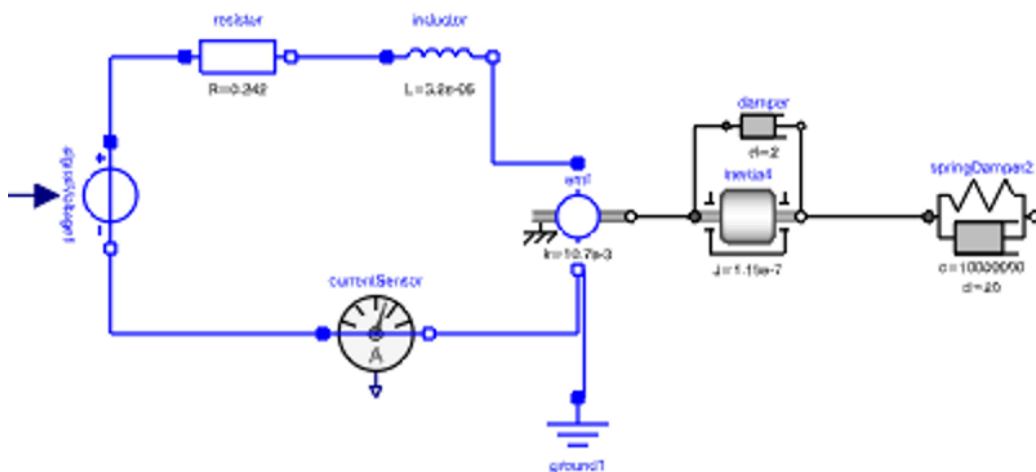
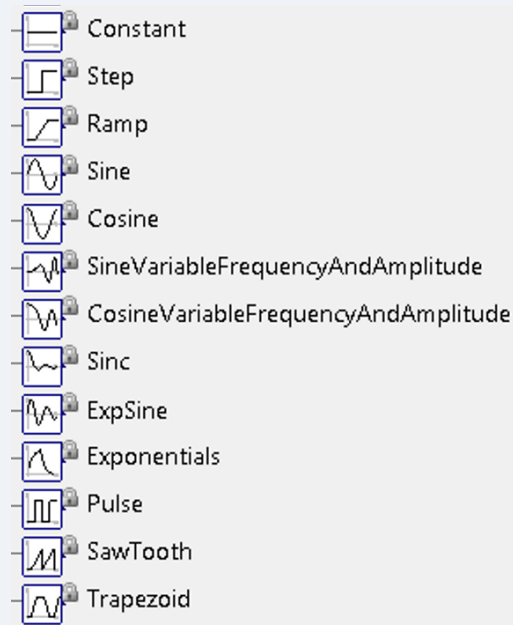


Schéma Modélica



## Test de performance du modèle Moteur simplifié

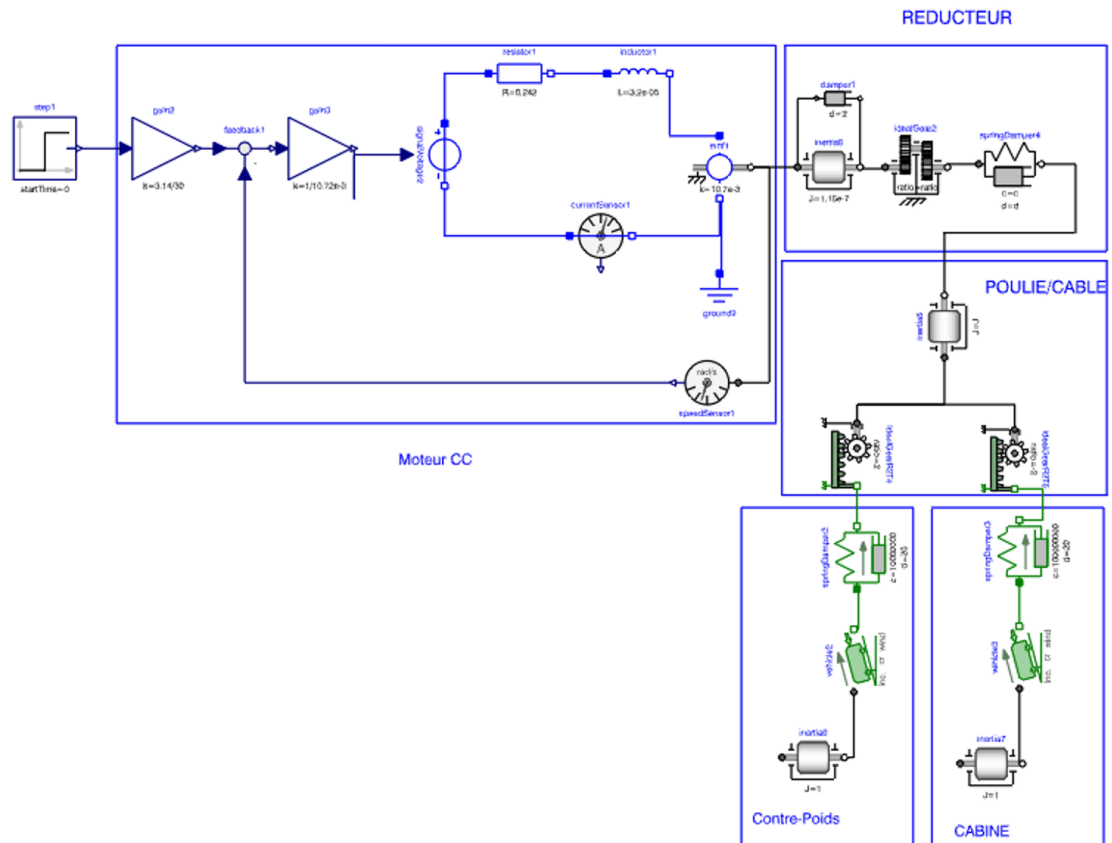
Choix d'un signal de consigne :



### 3 Assemblage pour simulation du système complet (raffiné)

Grâce à cette maquette, on peut vérifier que le moteur pré-dimensionné correspond au cahier des charges et permet de déplacer la mécanique en **respectant** les performances attendues.

Ici, on prend en compte des phénomènes aérodynamiques et des inerties négligées par la maquette simplifiée :

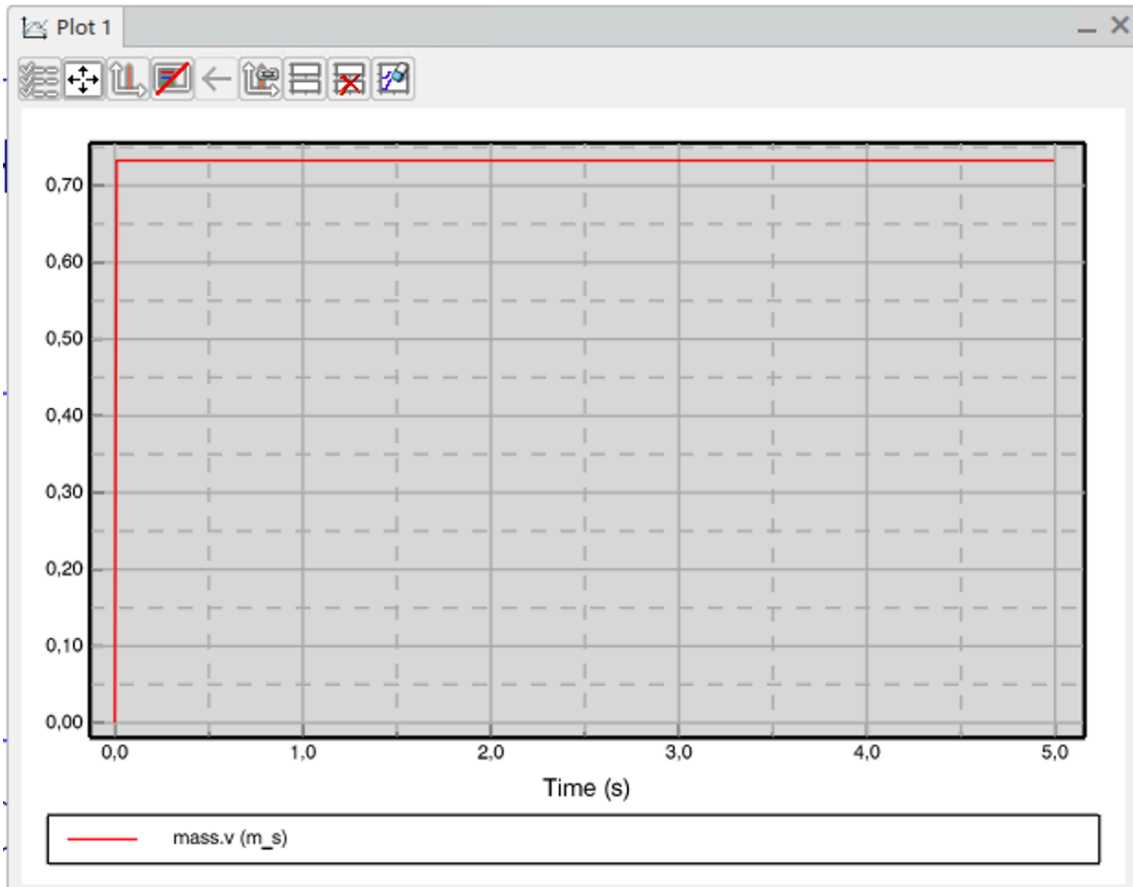


### 3.6.3. Résultats de simulation

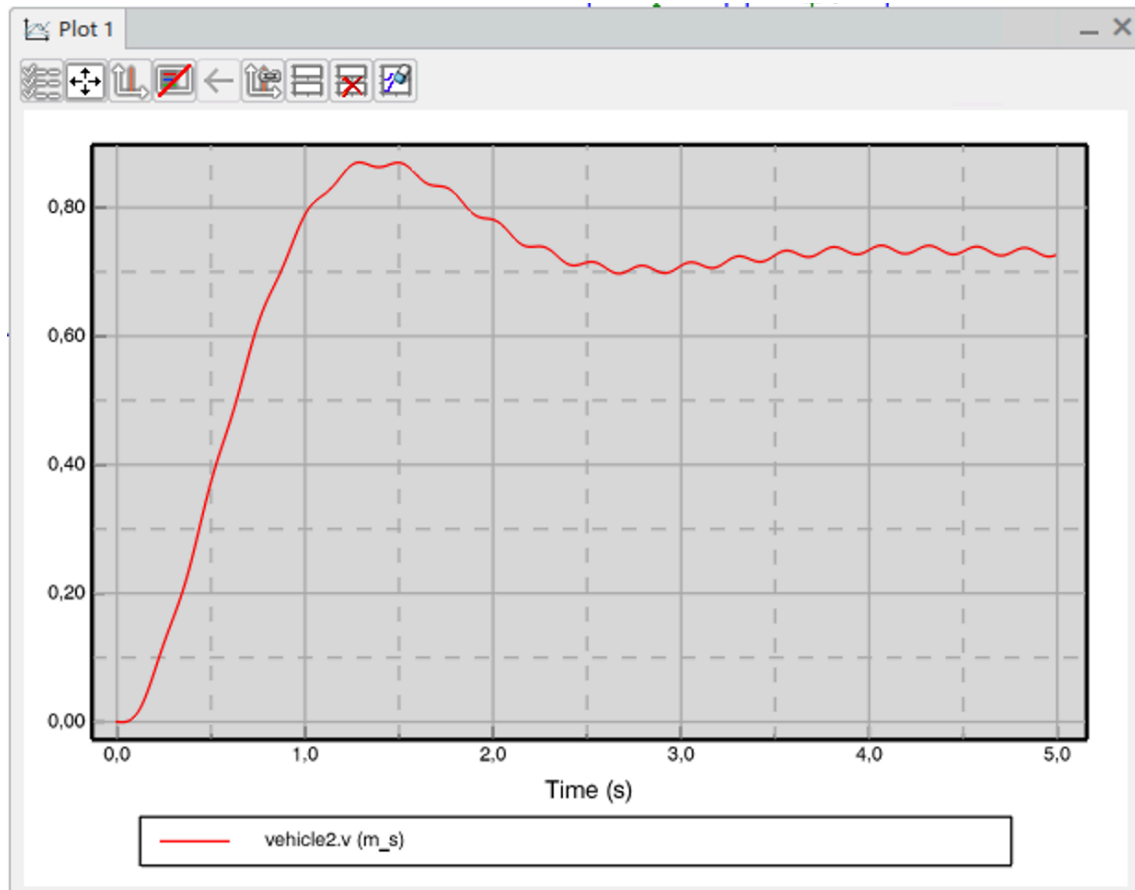
#### Définition de l'excitation

On met volontairement une excitation sous forme d'un step (donc brutal), pour observer l'impact de la rigidité de la transmission sur les performances en vitesse de la cabine.

## Vitesse cabine pour simulation n°1 (sans rigidité câble)

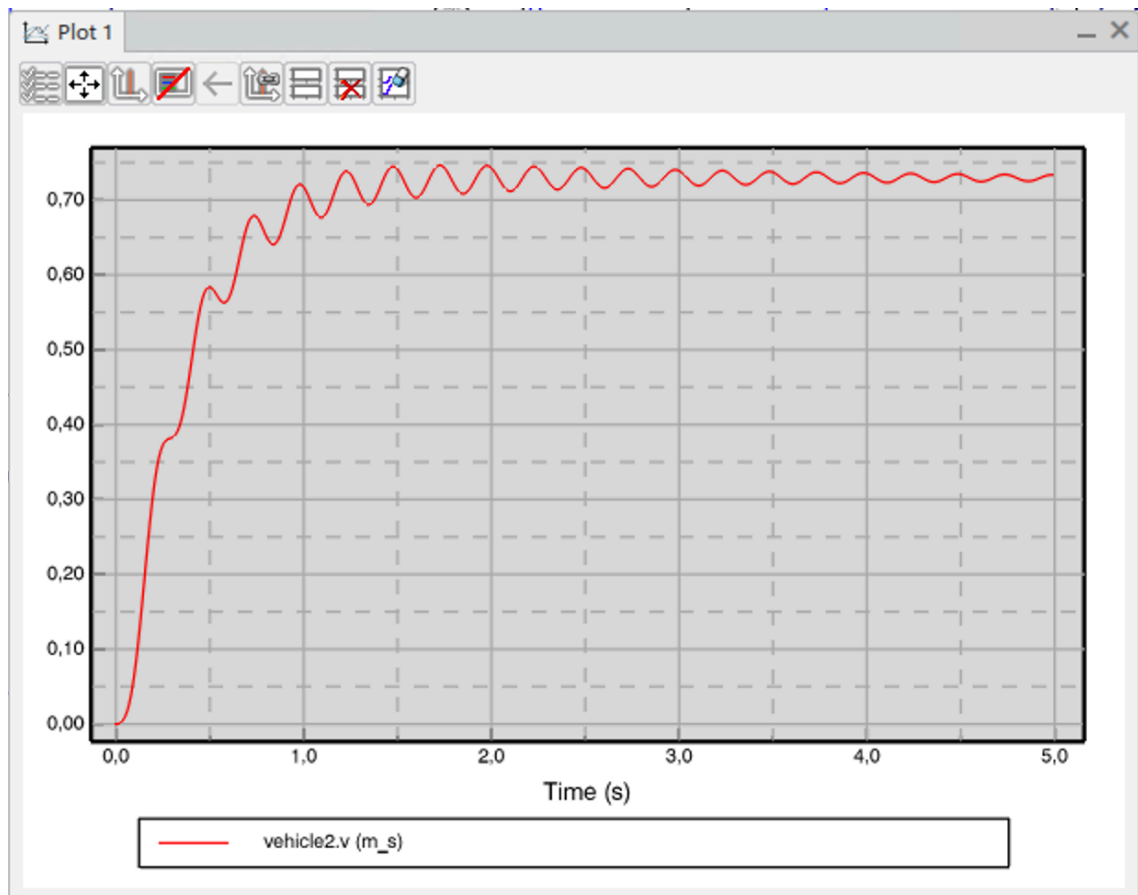


## Vitesse cabine pour simulation n°2 (faible rigidité câble)





## Vitesse cabine pour simulation n°3 (forte rigidité câble)



# Synthèse

---

## Évaluation du module

Le module que vous avez étudié fait partie d'un projet en phase de test.

Nous vous invitons à compléter le formulaire d'évaluation dans une perspective d'amélioration continue.

Accès au formulaire :

- via l'URL du formulaire [\[https://forms.office.com/r/DPN2yFcKeH\]](https://forms.office.com/r/DPN2yFcKeH)

- via QR code : 

# Glossaire

<b>Approche holistique</b>	L'approche holistique permet d'envisager une étude de système dans sa globalité et non domaine par domaine.
<b>Blocs mathématiques =&gt; causal</b>	Dans les logiciels de simulation système, on trouve différents types de blocs tels que les blocs « mathématiques » représentant des relations mathématiques entre une entrée et une sortie. Ils peuvent être utilisés dans différents domaines : temporels ou fréquentiels.
<b>Blocs physiques &lt;=&gt; causal ou acausal</b>	Orientés objets, ils représentent des objets (résistances électriques, masses, moteurs électriques, flexibles hydrauliques, résistances thermiques, etc.) contenant les équations physiques du domaine concerné et portant des « ports » leur permettant de communiquer les uns avec les autres.
<b>Causalité</b>	La notion de causalité permet à l'outil informatique de séquencer les équations modélisant la physique du système et de les traiter dans un ordre logique. Elle consiste à désigner les variables d'entrée et de sortie d'un modèle et de chacun de ses sous-modèles. L'objectif est de créer une relation de cause à effet entre les entrées et les sorties du modèle.
<b>Couplage faible</b>	Lors d'une simulation d'interaction entre deux physiques, l'un des deux modèles ne dépend pas de l'autre, il peut être analysé en premier.
<b>Couplage fort</b>	Lors d'une simulation d'interaction entre deux physiques, chacune des deux physiques dépend de l'autre. Aucune simulation ne peut être menée à bien indépendamment. Il est donc nécessaire d'échanger des résultats entre les deux solveurs, dans les deux sens, afin d'assurer le couplage.
<b>Diagramme d'exigences</b> <i>≈ req</i> <i>≈ requirement diagram</i>	Les diagrammes d'exigences servent à préciser ce que l'on attend du système, généralement dérivées du cahier des charges.
<b>Diagramme de blocs internes</b> <i>≈ ibd</i> <i>≈ internal block diagram</i>	Un diagramme de blocs internes est un diagramme structurel statique. L'objectif des diagrammes de blocs internes (IBD) est de montrer le contenu structurel encapsulé (composants, propriétés, connecteurs, ports, interfaces) des blocs afin qu'ils puissent être décomposés de manière récursive et "reliés" à l'aide d'interfaces. Autrement dit, un IBD est une vue "boîte blanche" (je comprends et je décris ce qu'il y a à l'intérieur du système considéré) d'un bloc qui a jusqu'alors été décrit en "boîte noire" (je ne sais pas ce qui se passe à l'intérieur, je

	ne peut que constater son comportement).
<b>Diagramme de machine à état</b> <i>≈ state machine diagram</i>	Le diagramme de machine à état décrit les transitions d'état et les actions qu'un système ou ses parties effectuent en réponse à des événements.
<b>Diagramme de séquences</b> <i>≈ sequence diagram</i>	Le diagramme de séquences représente l'interaction entre les parties collaboratrices d'un système.
<b>Éléments passifs</b>	Ce sont des éléments qui reçoivent de l'énergie.
<b>Fonction de transfert</b>	Une fonction de transfert est une relation mathématique permettant de relier une variable de sortie $y$ avec une variable d'entrée $x$ . Elle respecte le <b>principe de causalité</b> .
<b>Modèle acausal</b>	La modélisation dite acausale, contrairement à la modélisation causale, permet de décrire les systèmes sans anticiper l'orientation des liens entre les composants du phénomène. Comme son nom l'indique, un modèle acausal ne représente pas le lien de cause à effet. Il n'y a notamment pas de choix particulier de variables échangées entre composants, ni de notion d'entrées sorties.
<b>Modèle causal</b>	Un modèle multi-physique causal traduit le comportement d'un système physique réel (composé d'éléments mécanique, électrique, acoustique, optique, etc.) sous forme d'équations mathématiques. On le représente généralement sous la forme de schémas composés de blocs représentant des fonctions de transfert. On le qualifie de « causal » en ce sens que la sortie dépend de l'entrée.
<b>Paramètre</b>	Quel que soit le type de blocs, ils devront tous contenir les caractéristiques des éléments représentés que l'on appellera <b>paramètres</b> .
<b>Paramètres localisés</b>	Ce sont les paramètres que l'on choisit pour représenter les caractéristiques des éléments composants le système étudié. Il s'agit ici de localiser dans la chaîne d'énergie les éléments caractéristiques du système, dont on va suivre l'évolution pendant l'étude.
<b>Schéma bloc</b>	Le schéma bloc est un outil pour décrire et comprendre le comportement dynamique d'un système décrit par des équations différentielles qui peuvent être écrites et résolues dans deux domaines différents : <ul style="list-style-type: none"> <li>● Domaine temporel</li> <li>● Domaine fréquentiel</li> </ul>
	En règle générale un logiciel de simulation multi-physique transforme un modèle graphique (sous forme de blocs) en une représentation ODE (Ordinary Differential Equations) pour effectuer une simulation

<p><b>Solveur</b></p>	<p>en utilisant différentes méthodes d'intégration numérique. Ces méthodes d'intégration sont aussi appelées solveur et se distinguent par leur caractéristique :</p> <ul style="list-style-type: none"> <li>● explicit vs. implicit</li> <li>● order</li> <li>● step size control</li> <li>● multi step</li> </ul> <p>NB : Ces caractéristiques sont plus ou moins adaptées à la physique considérée.</p>
<p><b>Variable</b></p>	<p>Valeur d'un paramètre variant au cours du temps pendant le fonctionnement du système.</p>
<p><b>Variable d'effort</b> ≈ <i>Across variable</i></p>	<p>Elle représente une différence de potentiel, permettant de créer un flux. Par exemple une force (mécanique), une tension (électrique), une pression (hydraulique) ou une température (thermique), etc.</p>
<p><b>Variable de flux</b> ≈ <i>Through variables</i></p>	<p>Elle représentent un flux passant au travers d'un élément, par exemple un débit dans une canalisation, une intensité de courant dans un fil de cuivre, la vitesse d'un solide en déplacement, ou bien un courant thermique dans une pièce, etc.</p>