

OpenModelica

La modélisation multiphysique en
Sciences de l'ingénieur

Volume 2 – Activités complémentaires

Sanchez Pablo

Introduction

OpenModelica est un logiciel libre de simulations numériques de système physique. Basé sur le langage de modélisation Modelica, il permet la modélisation causale (schéma fonctionnel) ou acausale (schéma structurel), la simulation, l'optimisation et l'analyse de systèmes physiques complexes (électrique, mécanique, thermique, pneumatique, chimique, hydraulique, etc.).

La communauté d'OpenModelica poursuit activement le développement du logiciel notamment à travers une organisation à but non lucratif.

Le logiciel est aussi bien utilisé par le monde universitaire que par le secteur industriel. De nombreuses bibliothèques (base de données composants) permettent la modélisation de systèmes automobiles, traitement de l'eau, aéronautique, défense, énergie, process, équipement industriel...

OpenModelica permet de simuler le fonctionnement de systèmes physiques pour analyser leur comportement. L'utilisateur modélise chaque sous-ensemble du système par des composants issus d'une base de données intégrée au logiciel.

Chaque sous-ensemble du système est modélisé par des composants issus d'une base de données intégrée au logiciel. Ce schéma est transformé en un ensemble d'équations que le simulateur associé résout.

De ce fait, l'utilisateur peut analyser toutes les caractéristiques de chaque composant du système.

Dans ce volume 2, nous allons voir comment OpenMolica peut se mettre au service de nombreuse thématique du programme de sciences de l'ingénieur, comme l'automatisme et la modulation et démodulation de signaux.

Le recueil est découpé en 3 parties :

- Les systèmes pluri-technologiques
- Les systèmes asservis
- La modélisation des modulations et démodulation de signaux

Tables des matières

Introduction	1
Tables des matières	2
Premier pas avec OpenModelica	4
Téléchargement	4
Interface	4
Affichage des résultats	6
Les blocs fonctionnels	7
Connecteurs	7
Grande famille	7
Activité 1 - Déterminer la tension d'alimentation d'un moteur à courant continu pour assurer la bonne vitesse de montée	8
Modèle Multiphysique	9
Utilisation du modèle pour estimer la tension d'alimentation	10
Démarche analytique	10
Activité 2 - VTT de descente électrique	11
Analyse du VTT	11
Modéliser le VTT et l'actionneur	13
Vérifier les performances avec ajout de l'assistance	14
Conclusion : retour sur le cahier des charges	15
Activité 3 – la chaine d'énergie de la voiture ZOE	16
Activité 4 - Actionneur ElectroMécanique (EMA) de la tuyère de la fusée VEGA	19
Analyse de l'actionneur	19
Problématique	20
Description structurelle	20
Modéliser l'actionneur	21
Régulation	22
Influence du gain proportionnel	23
Perturbations	23
Activité 5 - Etude de l'asservissement d'un télésiège	24
Identification des paramètres de la fonction de transfert	25
Asservissement de la vitesse de montée	25
Influence du correcteur PID	26
Rôle du limiteur	27
Activité 6 - Modélisation de la régulation en température de la tête d'impression d'une imprimante 3D	28
Structure utilisée pour la partie chauffante de l'imprimante 3D	28

La tête d'impression	28
Paramètres physiques de la tête d'impression.....	29
Essai en Boucle Ouverte – Identification de modèle de comportement.....	30
Régulation de température – Tout ou Rien.....	31
Régulation de température – Correcteur Proportionnel Intégral Dérivé (PID).....	32
Réglage des correcteurs	32
Comparaison avec le modèle réel.....	33
Calcul du coût d'une impression 3D.....	33
Déterminer le temps d'une impression avec CURA	34
Activité 7 - L'Internet des Objets	35
Cas des modulations numériques (1 symbole = 1 bit)	36
Signal numérique à transmettre	36
Cas des modulations numériques à symbole constitué de plus de 1bit.....	39
Cas de la modulation M-QAM	42
Document réponse.....	43

Premier pas avec OpenModelica

Téléchargement

Vous pouvez retrouver le logiciel sur : <https://openmodelica.org/>

Puis :

- Ouvrir OMEdit
- Cliquer sur « Nouvelle Classe Modélica »

ATTENTION : Règle d'or

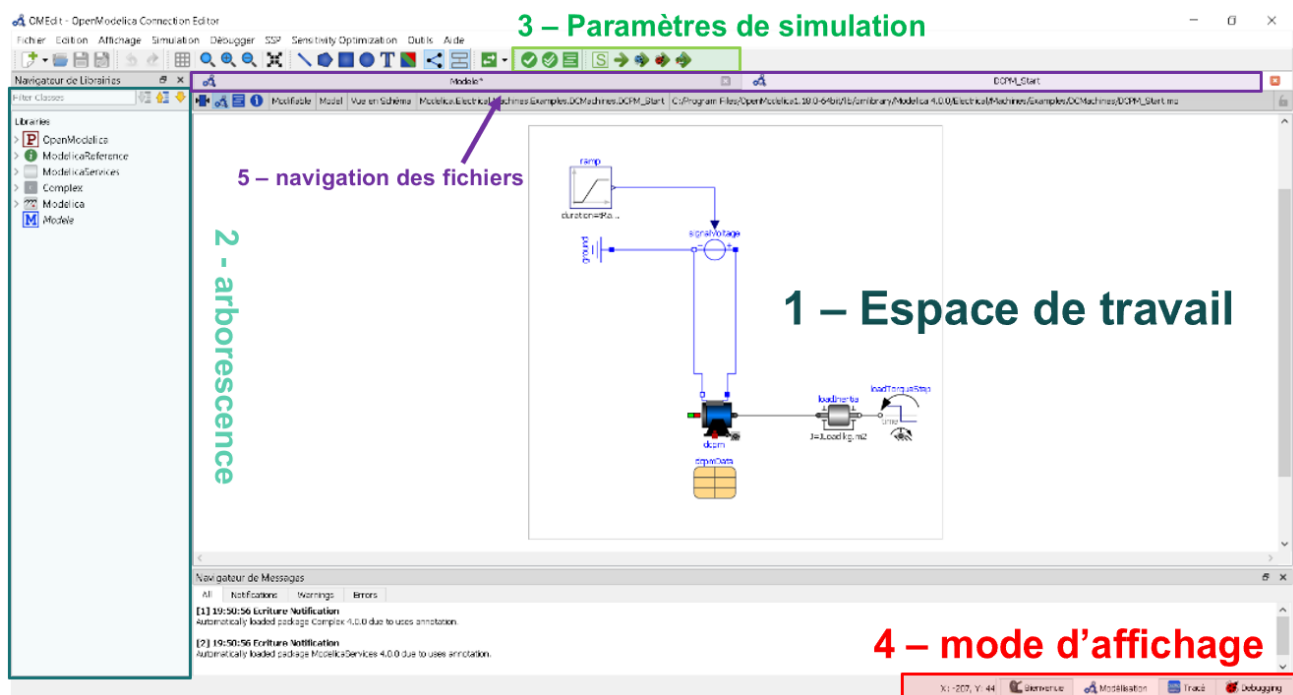
Lors de l'enregistrement, vos modèles ne doivent contenir **ni accent ni espace**.

L'enregistrement des modèles doit se faire dans un chemin de dossier **sans aucun espace ni accent !**

Pensez toujours à enregistrer vos modèles dans un dossier facilement accessible et proche de la racine.

Exemple : D:\Lycee\ModelesOM

Interface




L'interface Modélisation de OpenModelica

1-Espace de travail

La zone Espace de travail permet de construire les modèles. Pour de déplacer dans la zone :

- Molette Haut-Bas : faire défiler en haut/bas
- Shift + Molette Haut-Bas : faire défiler en droite/Gauche
- Ctrl + Molette Haut-Bas : Zoomer/dézoomer



 : Permet de recentrer votre modèle

 : Permet de dessiner des formes simples

2-arborescence

L'arborescence est l'un des espaces les plus importants. Il est aussi appelé le **Navigateur de Bibliothèques**

Vous y retrouverez :

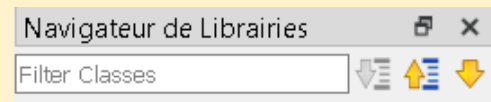
-  Modelica : La bibliothèque des blocs
-  Modèle : Vos modèles

Les bibliothèques de modèles sont divisées en grande famille selon la nature du flux :

- Magnétique
- Mécanique
- Fluide
- Thermique
- Electrique
- Etc....

Astuce

Pour rechercher les blocs demandés, vous utiliserez la barre de recherche :



3-Paramètres de simulation



Configuration de la simulation

Lancer la simulation

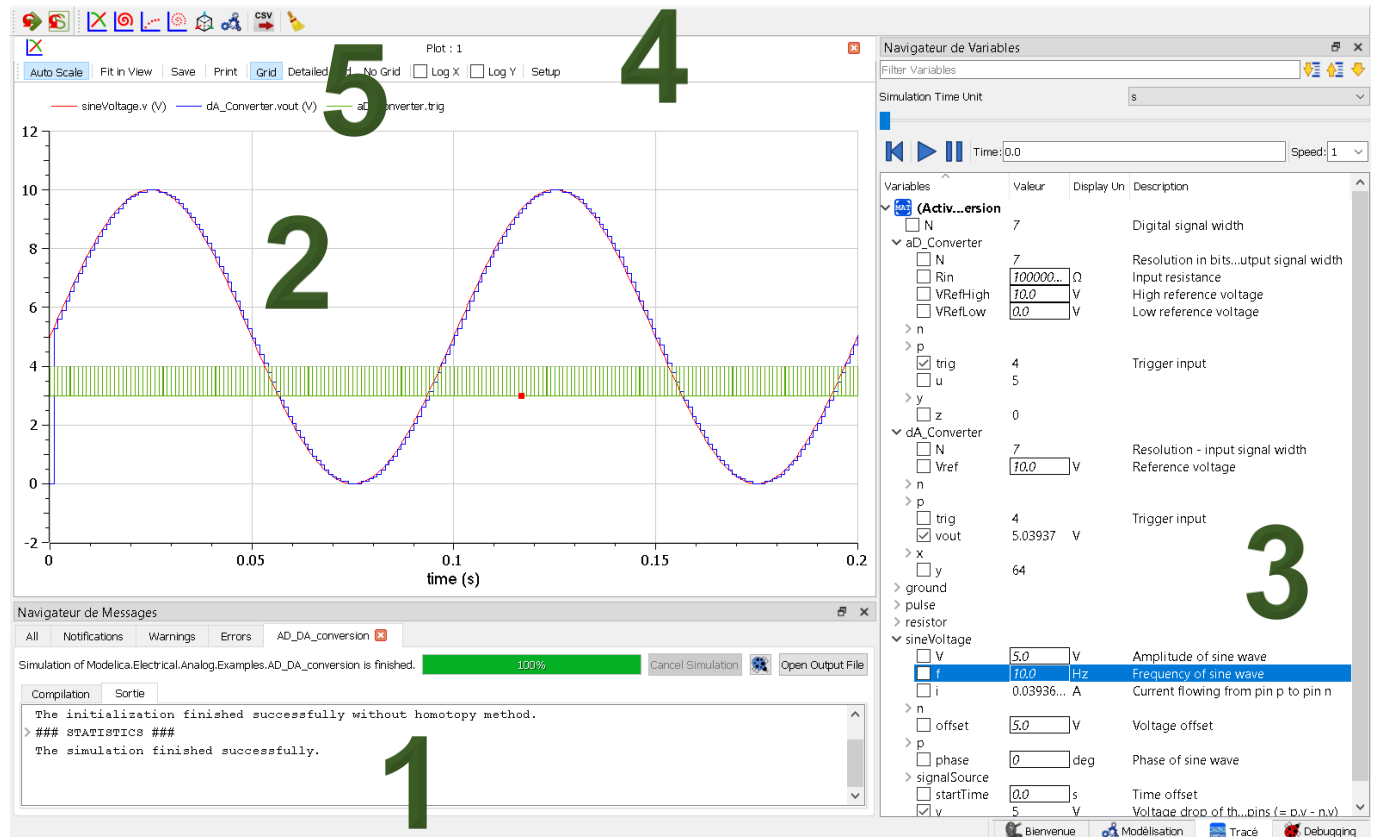
4-mode d'affichage

Une fois la simulation lancée, vous pouvez observer les résultats dans l'onglet **Tracé**.

Vous pouvez revenir sur votre modèle à tout moment en cliquant sur **Modélisation**

Affichage des résultats

Dans l'onglet **Tracé**



1. Navigateur de message. Une simulation réussie affichera le message « *The simulation finished successfully.* »

2. Affichage des courbes

3. Menu de sélection des grandeurs observables :

Le menu de sélection des courbes peut être déroutant de premier abord. Vous avez la possibilité de choisir l'affichage de toutes les grandeurs observables dans votre modèle. Soyez très attentif à ce que vous voulez afficher.

Aidez-vous des unités !

4. Réglage de la vue. Le bouton « **Fit in view** » vous permet de zoomer au mieux sur la courbe. Le bouton « **Setup** » permet de choisir les options d'affichage de vos courbes.

5. Sélection des courbes



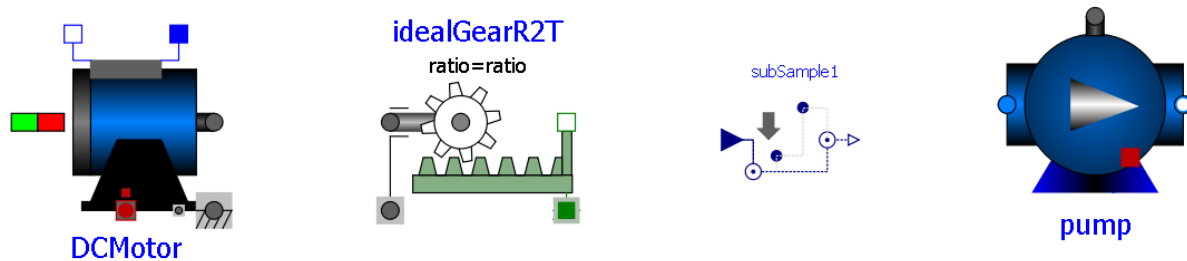
Création d'une nouvelle courbe

Création d'une courbe Paramétrique (maintenir shift pour choisir la variable en abscisse)

Export en CSV

Nettoyer la fenêtre

Les blocs fonctionnels



Pour ajouter un bloc dans votre espace de travail, maintenez et glissez le bloc depuis l'arborescence.

Double-clic pour accéder aux paramètres du bloc.

Dans certain cas, les paramètres du modèle sont associés à un tableau de valeur :



Connecteurs

Les blocs sont reliés entre eux par des connecteurs représentant le flux d'énergie.

La connexion doit se faire par forme équivalente (un carré vers un carré, un rond vers un rond etc...)



La connexion « Flèche » représente un signal. La construction des différents signaux d'entrées se fait dans la librairie Blocks>Sources

Grande famille

Grande famille	Couleur
Electrique	Bleu
Mécanique (translation)	Vert
Mécanique (rotation)	Gris
Thermique	Rouge
Magnétique	Orange
Hydraulique	Bleu clair
Logique Booléenne	Rose

Important : Vous devez toujours relier les blocs par couleur correspondant à leurs connecteurs !

Partie 1 – Les systèmes pluri-technologiques

Activité 1 - Déterminer la tension d'alimentation d'un moteur à courant continu pour assurer la bonne vitesse de montée.

La présente activité a pour objectif de déterminer la tension d'alimentation d'un moteur à courant continu d'un système de levage, grâce à l'utilisation d'un modèle multi-physique fourni (à paramétrer partiellement) et également de manière analytique (papier et crayon) pour atteindre une vitesse de translation à la montée de 1 m.s^{-1}

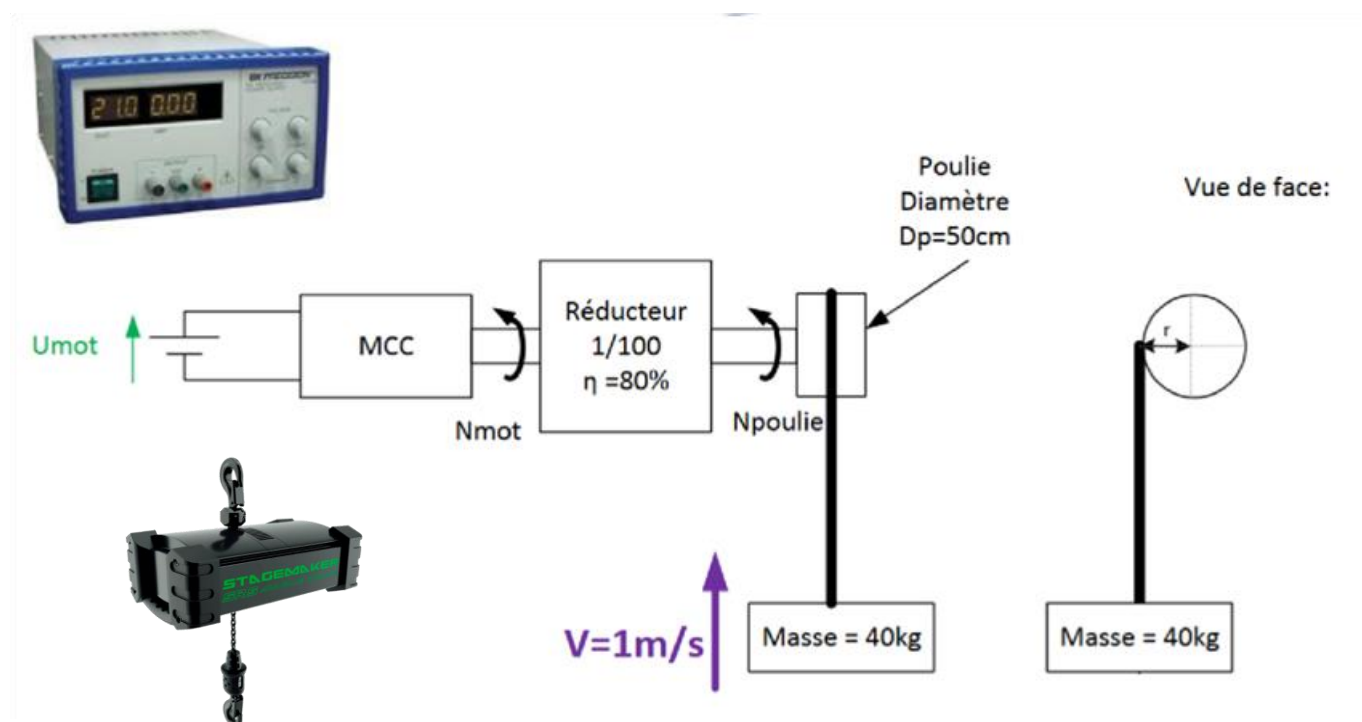


Figure 1

Vous disposez des ressources suivantes :

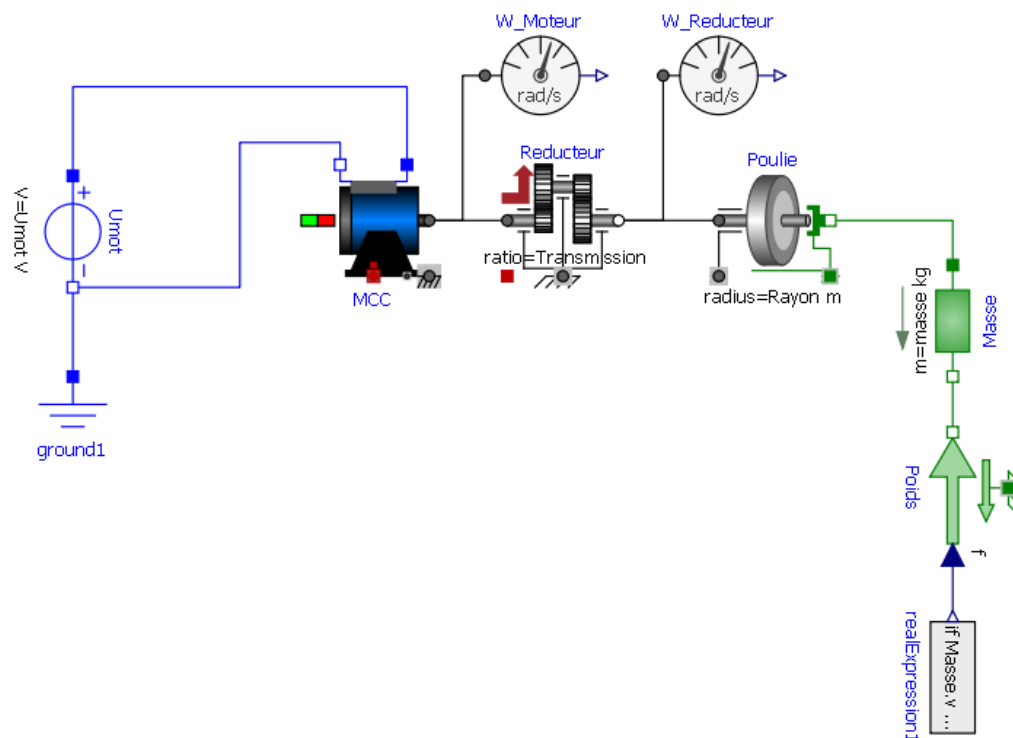
- Un extrait de la documentation constructeur du moteur à courant continu
- Le modèle du multi-physique du système de levage Modelisation_Moteur_eleve.mo dans le dossier ModeleOM

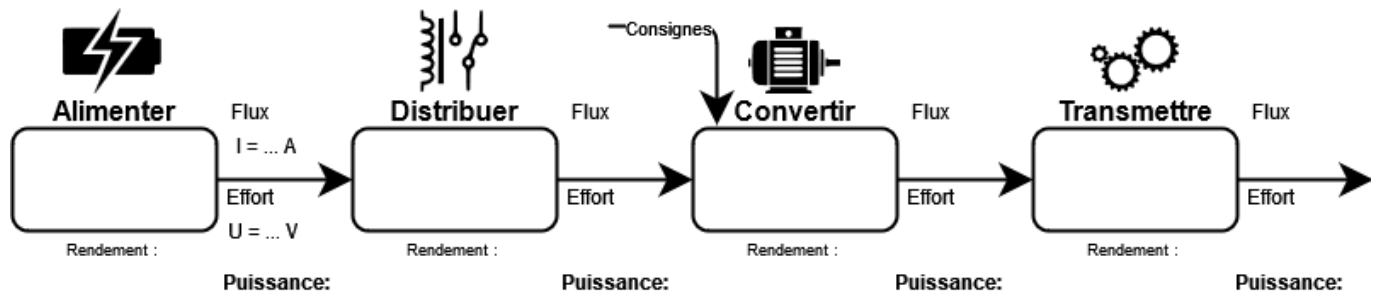
Caractéristiques du moteur

Couple en rotation lente	3.1	<i>N.m</i>	M_O
Courant permanent rotation lente	10.9	<i>A</i>	I_O
Tension d'alimentation de définition	92	<i>V</i>	U
Vitesse de définition	2700	<i>tr/mn</i>	N
Tension maximale	130	<i>V</i>	U_{max}
Vitesse maximale	4200	<i>tr/mn</i>	N_{max}
Courant impulsionnel	32	<i>A</i>	I_{max}
Fem par 1000 tr/mn (25°C)	31	<i>V</i>	K_e
Coefficient de couple électromagnétique	0.296	<i>N.m/A</i>	K_t
Couple de frottement sec	13	<i>N.cm</i>	T_f
Coefficient de viscosité par 1000tr/mn	0.92	<i>N.cm</i>	K_d
Résistance du bobinage (25°C)	0.4	Ω	R_b
Inductance du bobinage	2.34	<i>mH</i>	L
Inertie rotor	0.00135	<i>kg.m²</i>	J
Constante de temps thermique	17.8	<i>min</i>	T_{th}
Masse moteur	6.3	<i>kg</i>	M

Figure 2

Modèle Multiphysique





Question 1: Compléter la chaîne d'énergie avec les différents éléments de la Figure 1. Indiquer les grandeurs de flux et d'effort, ainsi que la nature des puissances entre chaque bloc.

Question 2: Dessiner le modèle multi-physique d'un moteur à courant continu et écrire les équations instantanées fondamentales d'un moteur à courant continu (1 équation électrique, 2 équation électro-mécaniques).

Question 3: Ouvrir le modèle multi-physique du système de levage, faire un clic droit sur le moteur MCC puis « **open class** ». Le modèle du moteur correspond -il à votre modèle ? Expliquer les différences.

Utilisation du modèle pour estimer la tension d'alimentation

Question 4: Paramétrer le modèle du système de levage à partir des documents et de la figure 1 et 2 (Moteur, réducteur, poulie, masse).

Aide : le rapport de réduction « r » d'un réducteur est par définition $r = \frac{\omega_{\text{sortie}}}{\omega_{\text{entrée}}}$

Le logiciel OpenModelica utilise le **rapport de transmission**, un ratio qui vaut $r = \frac{\omega_{\text{entrée}}}{\omega_{\text{sortie}}}$

Question 5: A partir du modèle paramétré, trouver par dichotomie la tension à appliquer pour satisfaire à une vitesse de 1 m.s^{-1} .

Démarche analytique

Question 6: Donner l'équation moteur simplifiée lorsque l'intensité du courant est continue dans le temps.

Question 7: Déterminer de manière analytique la tension U_{mot} nécessaire à assurer une vitesse de montée de 1 m.s^{-1} .

Activité 2 - VTT de descente électrique

Analyse du VTT

Cahier des charges

L'utilisation de VTT de descente est adaptée pour dévaler les pentes et encaisser les sauts. Par contre l'ascension est beaucoup plus laborieuse car ces modèles sont lourds.

Le VTT électrique rend la discipline VTT accessible aux personnes qui n'ont pas ou plus la capacité physique pour la pratique tout terrain sur un VTT traditionnel. La caractéristique essentielle d'un moteur électrique efficace pour le VTT est la puissance qu'il délivre face à la résistance qu'oppose une côte ou un obstacle. L'architecture la plus adaptée est de relier le moteur directement au pédalier. L'avantage de cette solution est que le travail du moteur peut être optimisé en fonction de l'obstacle à franchir en ajustant le rapport de vitesse du vélo.

Haibike est considéré comme le leader européen sur le marché des VTT électriques.

En 2015, la marque allemande dévoile une nouvelle gamme baptisée SDuro à assistance Yamaha. Avec leurs moteurs intégrés aux cadres et l'intégration de suspension Fox, ces modèles se déclinent en version 27,5 ou 29 pouces, en tout suspendus ou semi-rigides.

Ces modèles qui se veulent haut de gamme, doivent répondre à un cahier des charges représenté par le diagramme d'exigence partiel.

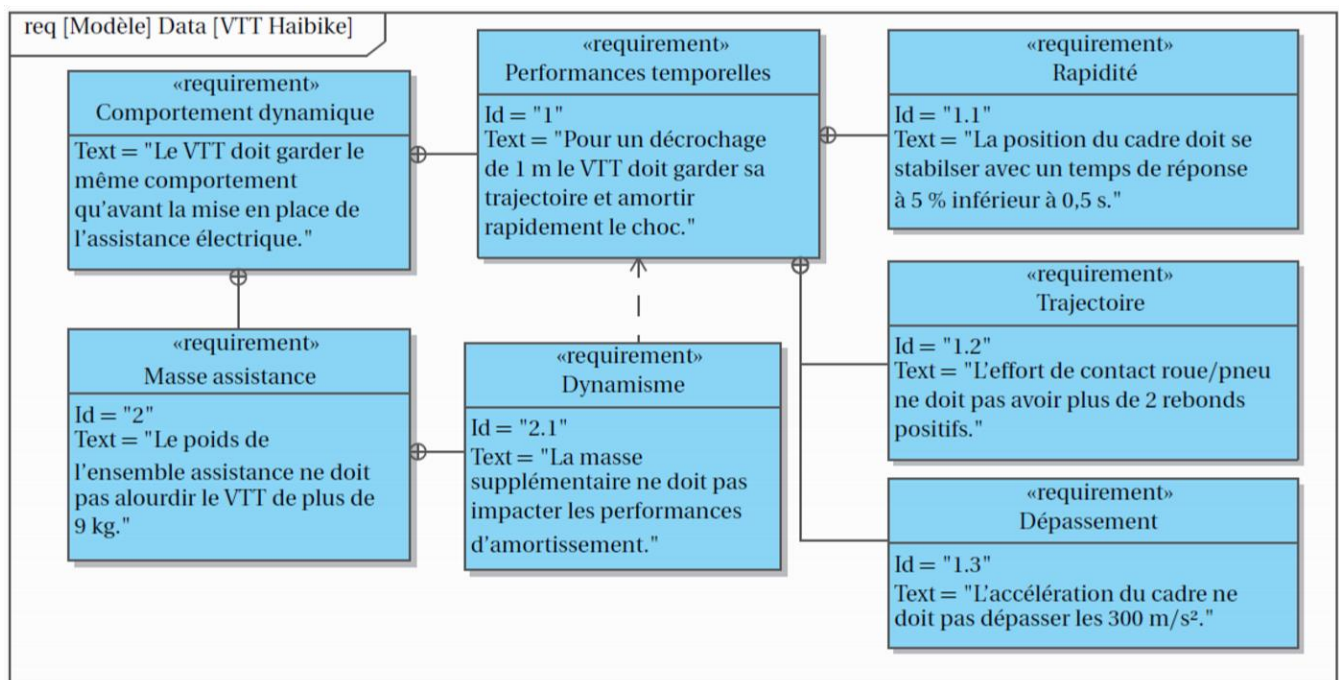


Figure 1 : diagramme des exigences partiel

Problématique

L'objectif est de vérifier les performances de trajectoire, de rapidité et de dépassement décrites dans le diagramme des exigences. Une attention particulière sera apportée à la vérification de l'exigence 2.1 : « l'ensemble d'assistance ne doit pas impacter les performances dynamiques du VTT ».

Description structurelle

Voici la description des différents composants du VTT.

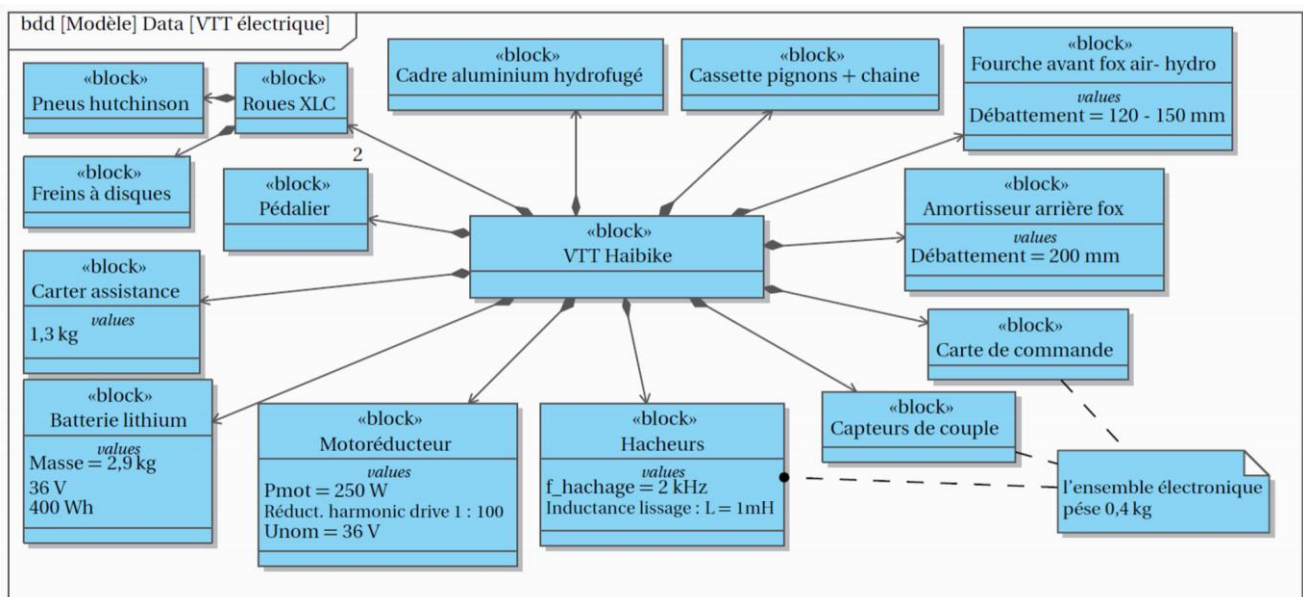


Figure 2 : diagrammes de définition des blocs

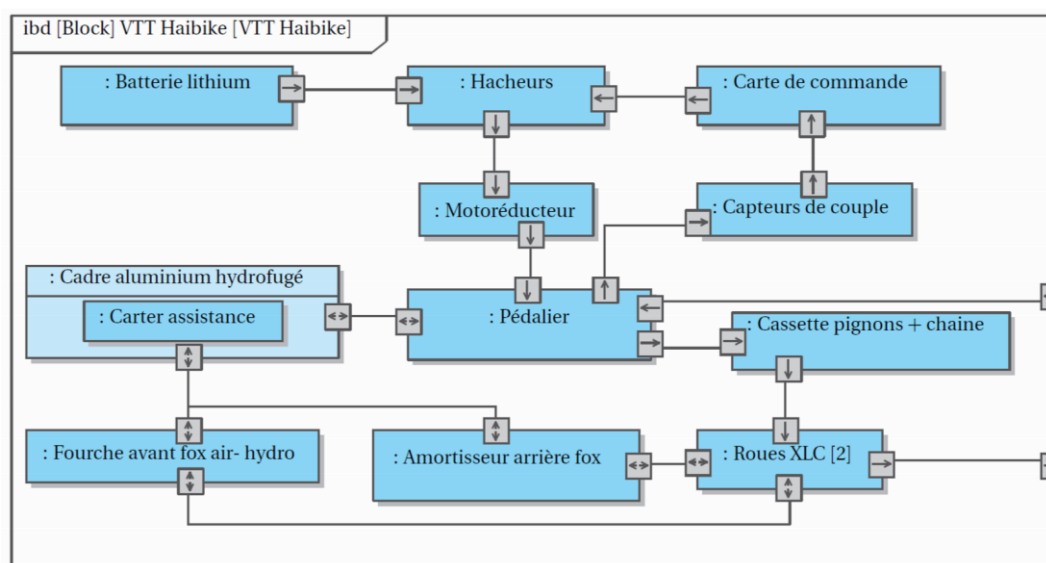


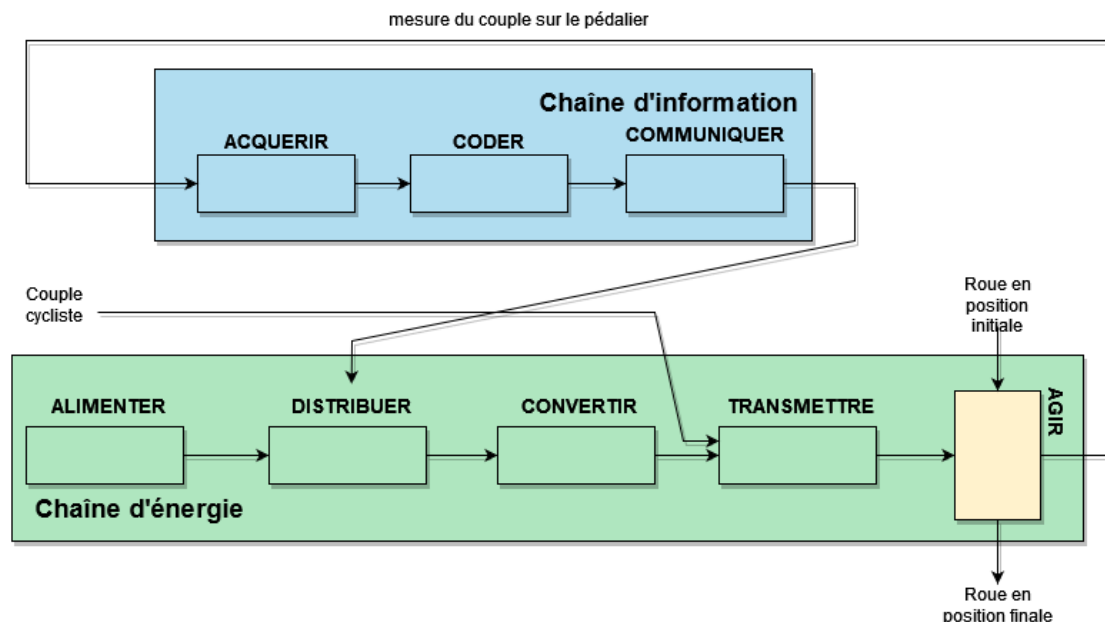
Figure 3 : diagrammes de définition des blocs internes

Modéliser le VTT et l'actionneur

Question 1: Sachant que le poids total du VTT est de **21.9 kg**, que celui-ci pesait **13.8 kg** avant la mise en place de l'assistance électrique, déterminer la masse du motoréducteur. Il faut tenir compte de l'ensemble des éléments ajoutés pour l'assistance électrique, dont les caractéristiques sont données dans le diagramme de définition des blocs.

Question 2: Indiquer sur chaque flux du diagramme de définition des blocs internes s'il s'agit de matière, d'énergie ou d'information.

Question 3: En vous aidant du diagramme de définition des blocs internes, compléter le diagramme de type « chaîne d'information et chaîne d'énergie ». Identifier sur ce schéma, pour chacun des composants du vélo avec assistance, la fonction correspondante ainsi que les grandeurs physiques d'entrées/sorties.



On se propose de représenter dans un premier temps les masses et les suspensions, avant et après implantation de l'assistance électrique.

On modélise les jantes des roues par des masses de **0.7 kg**, l'ensemble cadre (comprenant le pédalier et les autres éléments tels que potence, guidon, cassette ...) par une masse de **8.5 kg**, les pneus par des ressorts-amortisseurs visqueux en parallèle ainsi que les suspensions avant et arrière avec :

- Pneus : raideur **1800 N/m** ; coefficient d'amortissement **1 N.s/m**
- Suspension avant : raideur **200 N/m** ; coefficient d'amortissement **100 N.s/m**
- Suspension arrière : raideur **400 N/m** ; coefficient d'amortissement **50 N.s/m**

Ouvrir le modèle VTT_eleve.mo

Question 4: Compléter les paramètres du modèle avec les données à votre disposition.

Question 5: Vérifier et expliquer que la réponse à un échelon nul (aucun décrochage du sol) correspond à un déplacement du cadre VTT. On ne prend pas en compte ici la masse du cycliste qui allonge le temps réponse car on souhaite étudier uniquement le comportement du VTT, indépendamment des différents cyclistes.

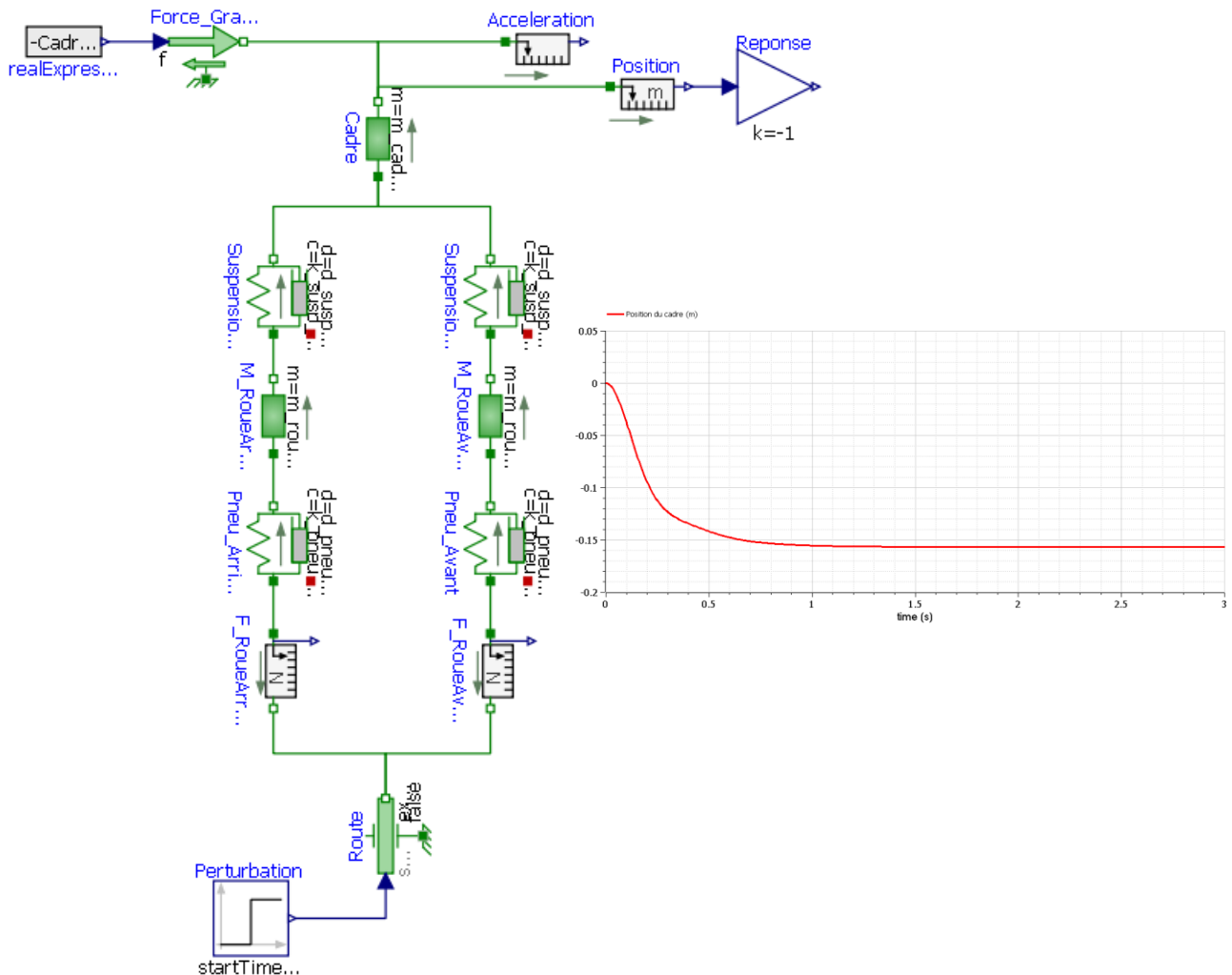


Figure 4 : modèle du VTT et réponse à une sollicitation nulle

Vérifier les performances avec ajout de l'assistance

Mettre un échelon de 1 m (décrochage du sol) afin d'obtenir un déplacement du cadre VTT. Puis afficher les courbes suivantes :

- **Cadre.a** : Accélération (m.s^{-2})
- **F_RoueAvant.f** et **F_RoueArrière.f** (N)
- **Cadre.s** : position du cadre (m)

Question 6: Mettre un échelon de 1 m (décrochage du sol) afin d'obtenir un déplacement du cadre VTT. Vérifier alors les performances de stabilité, de rapidité et de dépassement attendues par le cahier des charges avant l'installation de l'ensemble de l'assistance. Vérifier également l'exigence 1.2 en expliquant les conséquences des rebonds sur l'effort de contact.

Ajouter maintenant la masse de l'ensemble d'assistance à celle du cadre et relancer la simulation.

Question 7: Expliquer les différences d'allures des courbes ? Le cahier des charges est-il toujours respecté ?

Conclusion : retour sur le cahier des charges

Question 8: Déterminer l'écart entre la performance de rapidité attendue et la performance simulée en pourcentage.

Question 9: Quels sont les paramètres sur lesquels il est possible d'intervenir pour diminuer cet écart ? Modifier ces paramètres sur le modèle pour valider les critères du cahier des charges.

Activité 3 – la chaine d'énergie de la voiture ZOE

On s'intéresse à comparer les différentes performances annoncées par le constructeur.

Conditions d'essai :

- Les essais constructeurs étant faits sur circuit, nous mènerons une étude sur terrain plat, pas de pente.
- Modèle : **R110**
- Les pneus sont des Michelin **185 / 65 R15 92T**, la masse d'une roue pneu+jante est estimée à **18kg**
- Un conducteur (**100kg** avec équipement)



Ressources

- Extrait du dossier technique constructeur de la Zoé : [Document Technique ZOE](#)
- Modèle de la voiture Zoé (logiciel OpenModélica) : [ZOE eleve.mo](#)
- Site internet pour le dimensionnement des pneus : <https://www.toutcalculer.com/automobile/dimension-pneu.php#calcul>

Question 1: Rappeler l'équation de la Puissance mécanique d'un solide en translation.

Sur un essai sur terrain plat, la puissance que doit fournir la voiture pour avancer est noté Puissance utile P_{utile} :

$$P_{\text{utile}} = \frac{1}{2} \rho \cdot V^3 \cdot SC_x + m \cdot g \cdot C_{rr} \cdot V$$

- ρ = Masse volumique de l'air (kg/m^3)
- V = vitesse du véhicule (m/s)
- SC_x = Coefficient de pénétration dans l'air compte tenu de la surface frontale du véhicule (**sans unité**)
- m = masse du véhicule (kg)
- g = constante de gravité (m/s^2)
- C_{rr} = Coefficient de résistance au roulement des pneumatiques, ici **$C_{rr} = 0.006$**

Question 2: Faire le bilan des forces extérieures exercées sur la voiture ZOE

Question 3: A partir du modèle multiphysique et de l'équation de la Puissance Utile, retrouver les équations de la **résistance au roulement** et de la **trainée aérodynamique** ?

Question 4: A partir du document technique et des ressources mis à disposition, compléter le tableau suivant, puis renseigner les valeurs dans le modèle multiphysique (ne remplir que les valeurs du tableau)

Paramètre	Variable	Valeur	Unité
Rayon d'une roue	Rayon		
Moment d'inertie d'une roue	I_{roue}		
Masse du véhicule et conducteur	Masse		
Coefficient de résistance au roulement	C_{rr}		Sans unité
Constante de gravité terrestre	g		
Masse volumique de l'air	Masse_Volumique_Air		
Coefficient de pénétration dans l'air	SCx		
Tension de la batterie	Tension_Totale		

Remarque : pour simplifier le modèle, la batterie est modélisée par un générateur idéal et une résistance interne.

Lancer une simulation pour 100s

Rappel : un **régime permanent (ou régime établi)** est le régime d'un système stable observable après un certain temps, lorsque le **régime transitoire** est terminé. Cette durée d'extinction du régime transitoire dépend de l'**amortissement**.

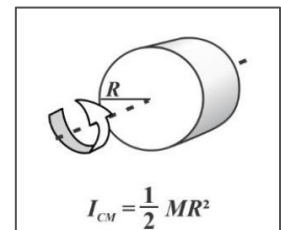
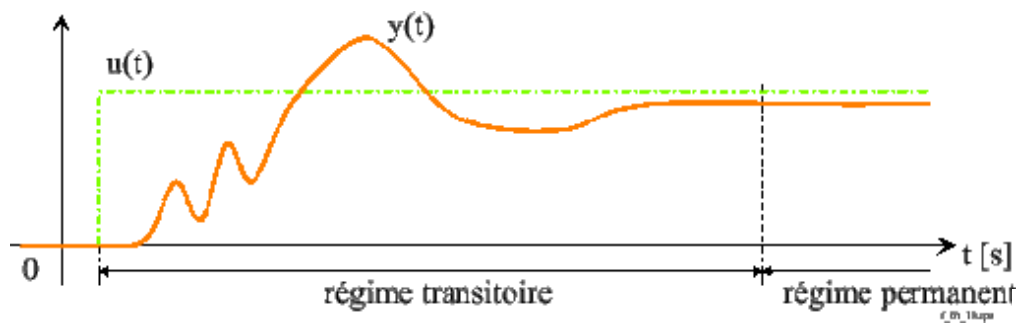


Figure 2 : Rappel Moment d'inertie

Question 5: Quel est la vitesse max (km/H) du véhicule en régime permanent ?

Question 6: Comment s'appelle le mouvement du véhicule en régime permanent ?

Question 7: Comparaison des performances de vitesses afin de noter quelques écarts entre le modèle et le réel.

Question 8: Copier et remplir ce tableau dans votre compte rendu. Commenter les écarts de performances.

Vitesse (km/h)	Temps constructeur (réel)	Temps simulation
0 – 50		
0 – 80		
0 – 100		
Vitesse max		

Question 9: Augmenter la masse à la charge **maxi autorisée**. Commenter l'influence de l'augmentation de cette valeur sur l'accélération du véhicule.

Question 10: A l'aide du bloc *Modelica.Blocks.Math.Division*, tracer la courbe du rendement total du véhicule.

Question 11: Tracer la courbe paramétrique de la Puissance mécanique (W) en fonction de la vitesse du véhicule (km/h). Pour quelle vitesse V_p la puissance mécanique est maximale ?

Question 12: Trouver la tension à appliquer sur le bloc *Tension* pour que le véhicule atteigne la vitesse V_p en régime permanent.

Question 13: Déterminer la Puissance électrique $P_{\text{electrique}}$ en régime établi.

Question 14: Calculer l'autonomie de la batterie en heures puis en km. Comparer cette valeur avec celle du constructeur. Commenter les écarts.

Partie 2 – Les systèmes asservis

Activité 4 - Actionneur ElectroMécanique (EMA) de la tuyère de la fusée VEGA

Analyse de l'actionneur

Cahier des charges

Le développement du lanceur européen VEGA a démarré en 1998 et s'est achevé en 2011. Ce projet répondait à une demande de mise en orbite basse et polaire, à coûts réduits, de satellites scientifiques dont la masse peut aller jusqu'à 2000 kg. La minimisation des coûts s'est appuyée sur l'intégration de technologies avancées déjà disponibles et l'utilisation des installations des lanceurs Ariane.

Une des innovations de ce projet concerne le système de contrôle vectoriel de poussée (en Anglais : « Thrust VectorControl ») du premier étage de propulsion P80. D'une longueur de dix mètres, le P80 est chargé de 88 tonnes de propergol solide. Ceci lui permet de disposer d'une poussée maximale de 3000 kN et d'un temps de combustion de 107 secondes. Alors que sur Ariane 5 le pilotage vectoriel de la poussée est assuré par des dispositifs à source de puissance hydraulique, sur le P80 cette tâche est assurée par des dispositifs à source de puissance électrique (en Anglais : « Power ByWire »). La tuyère est reliée à l'étage de propulsion, par une liaison qui permet d'orienter la tuyère autour du point fixe nommé point de rotulage.

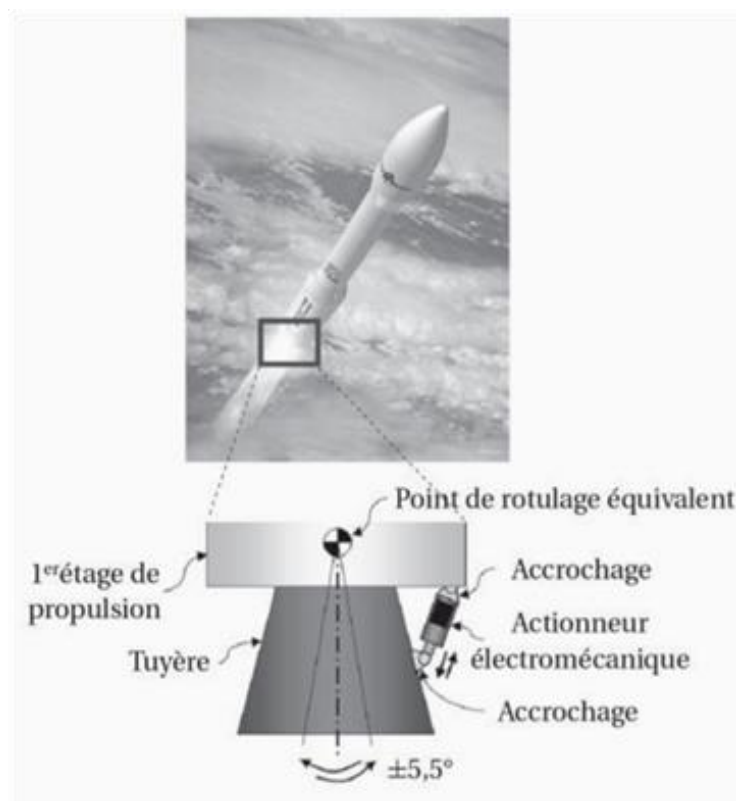


Figure 3 : Actionneur électromécanique d'orientation de la poussée vectorielle du premier étage de propulsion du lanceur européen VEGA.

Afin de bien contrôler la trajectoire de la fusée, il est indispensable d'orienter très rapidement et très précisément la tuyère. Le diagramme des exigences partiel de la figure 2 présente les valeurs des performances temporelles que doit réaliser l'actionneur.

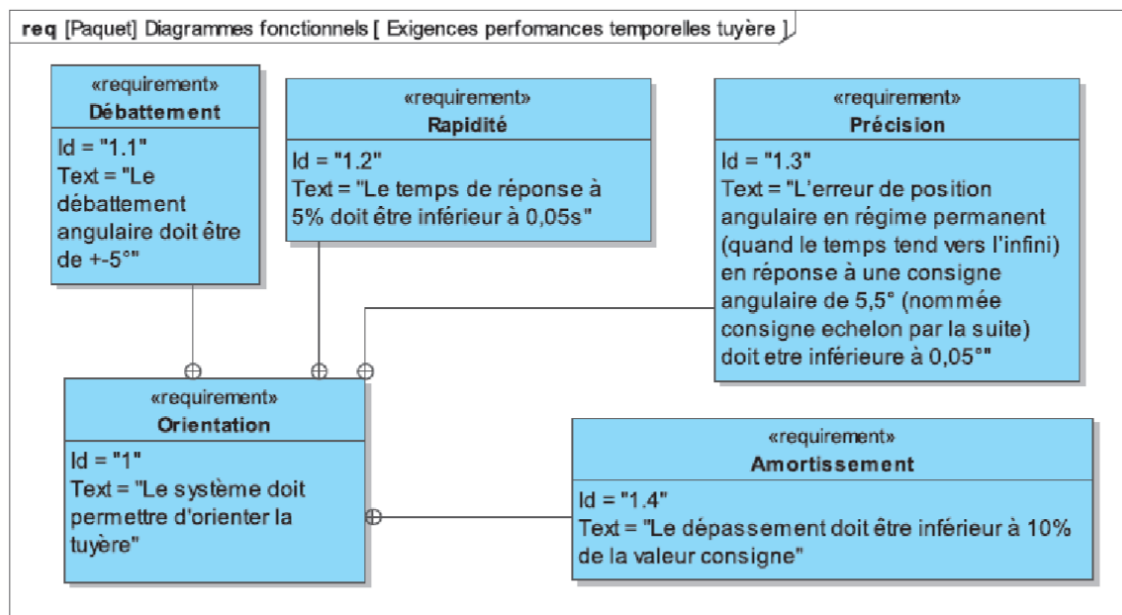


Figure 4: Diagramme des exigences

Problématique

L'objectif est de vérifier les performances de rapidité et de précision décrites dans le diagramme des exigences (figure 2). Un modèle multiphysique de l'actionneur va être mis en place afin de simuler le comportement de l'actionneur afin de valider ou non son comportement vis à vis du cahier des charges.

Description structurelle

Voici la description des différents composants de l'actionneur.

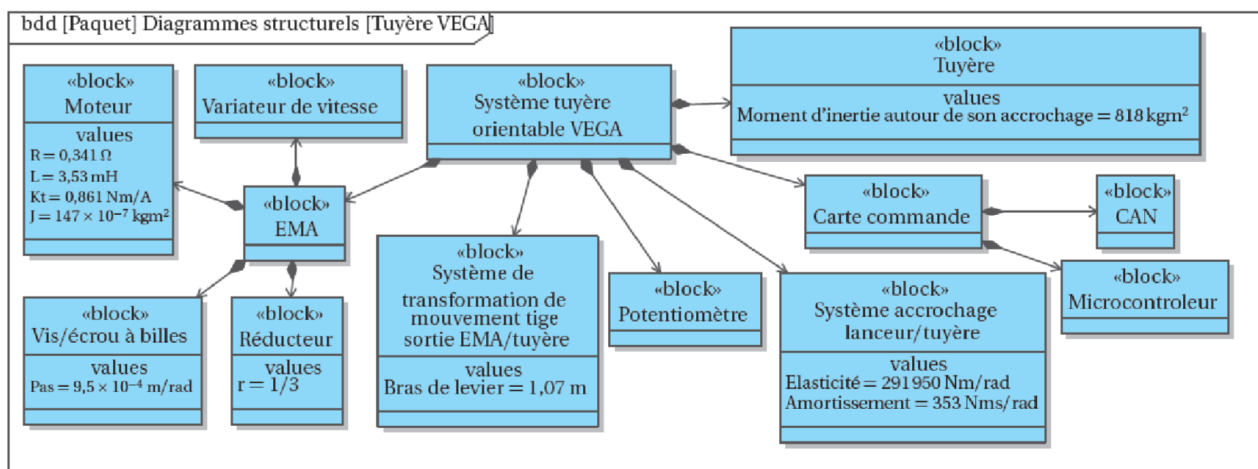


Figure 5 : Diagramme de définition des blocs

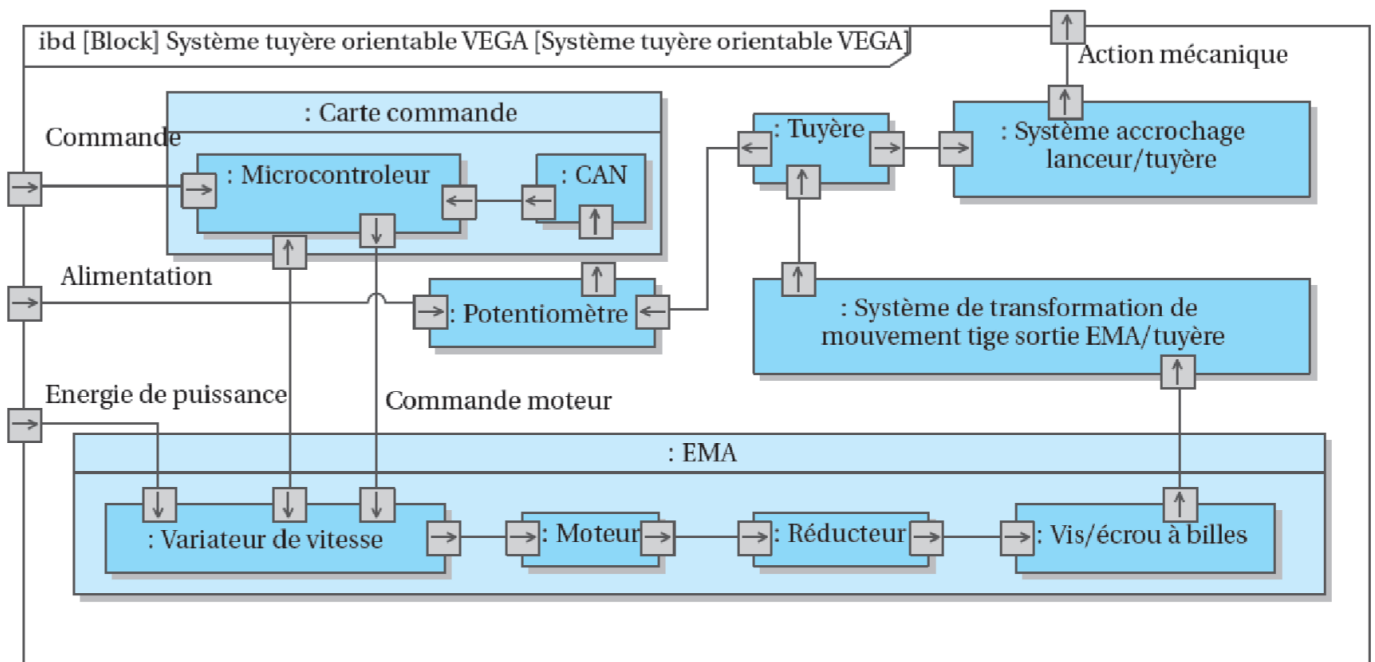
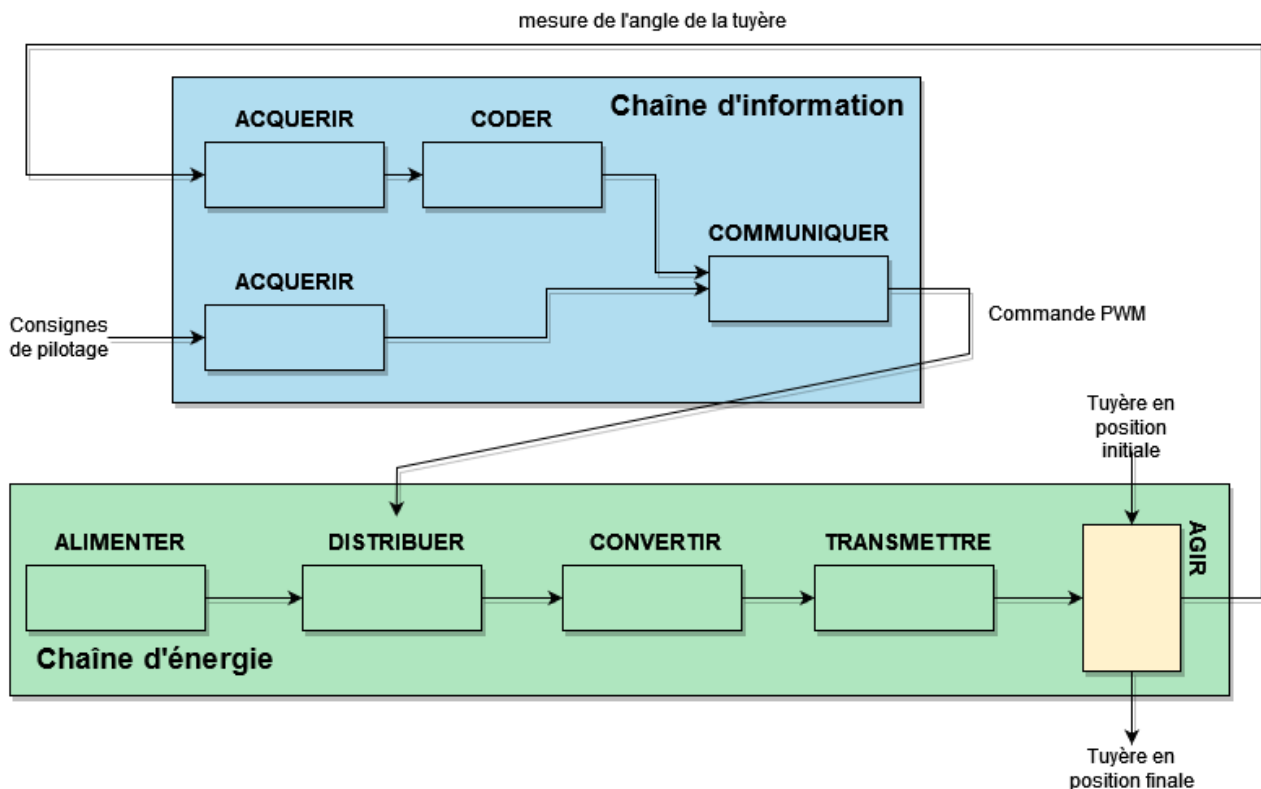


Figure 6 : Diagramme des blocs internes

Modéliser l'actionneur

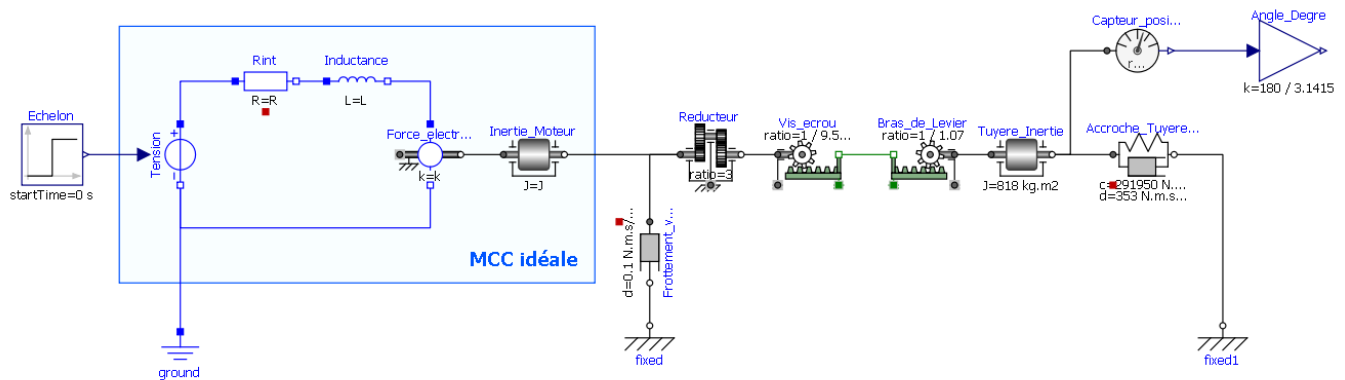
Question 1: Compléter les blocs de chaîne fonctionnelle (d'énergie et d'information) à partir du Diagramme des blocs internes et du Diagramme de définition des blocs.



Afin de simuler le comportement dynamique de l'actionneur, il est nécessaire de représenter la charge qu'il devra mettre en mouvement. Pour un solide en rotation on parle **d'inertie**. Cette grandeur s'exprime en kg.m^2 et quantifie la résistance d'un corps soumis à une accélération angulaire (le pendant de la masse qui caractérise la résistance d'un corps soumis à une accélération linéaire). Un frottement visqueux de l'arbre moteur est ajouté pour modéliser la manière dont est lubrifié le mécanisme.

Ouvrir le modèle : fusee_Vega_eleve.mo

Lorsque vous lancerez les simulations, vous sélectionnerez les courbes de *Echelon.y* et *Angle_Degre.y*



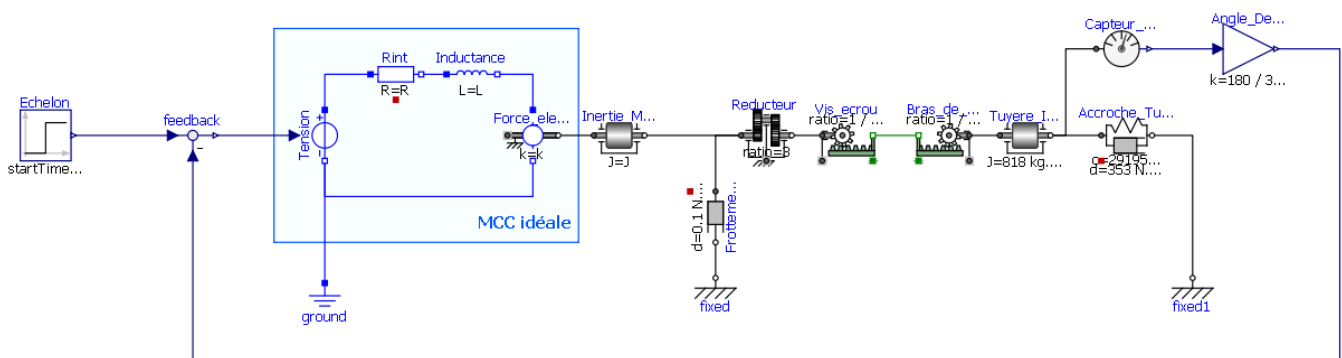
Question 2: Pour simplifier le modèle, le moteur a été représenté par le modèle d'une MCC Idéale. Compléter les valeurs des blocs de la MCC avec les données du diagramme de définition des blocs

Question 3: Vérifier que la réponse à un échelon de tension de 400V est une rotation de l'inertie de la tuyère.

Question 4: Mettre plusieurs valeurs de frottement visqueux (0.1 ; 0.5 ; 1 ; 5 Nm.s.rad^{-1}) et identifier son influence sur la rapidité de réponse du système.

Régulation

Question 5: Rajouter le feedback permettant la régulation du système :

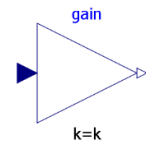


Question 6: Dans le bloc « Echelon », renseigner la valeur 5,5. Quelle est son unité ? A quel critère du diagramme des exigences fait-il référence ?

Question 7: Lancer la simulation et déterminer le **temps de réponse** et l'**écart statique**. Conclure quant au respect du cahier des charges

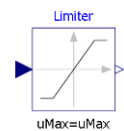
Influence du gain proportionnel

Pour corriger cette erreur nous allons ajouter un **Gain proportionnel** (bloc *Modelica.Blocks.Math.Gain*) entre le feedback et la source de tension.



Question 8: Déterminer par dichotomie la valeur du gain proportionnel pour atteindre un temps de réponse et un écart statique en raccord avec le cahier des charges. Conclure sur l'intérêt du gain proportionnel.

Question 9: Dans la réalité, le moteur ne peut pas accepter une tension supérieure à 400V. Ajouter un bloc « Limiteur » en sortie du Gain. Configurer le bloc pour limiter la tension à +400V et -400V.



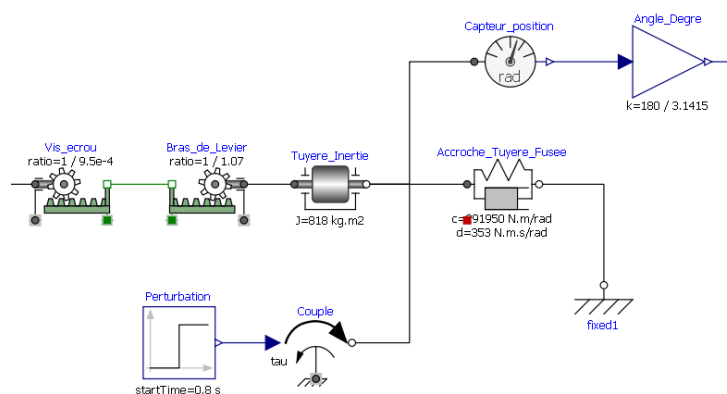
Question 10: Relancer la simulation et comparer les résultats. Le cahier des charges est-il toujours respecté ?

Question 11: Quels sont les paramètres mécaniques à changer pour améliorer le temps de réponse du système ? Changer les paramètres des blocs et relancer une simulation pour valider vos hypothèses.

Perturbations

Nous allons simuler une perturbation du système, par exemple une action mécanique extérieure qui viendrait déstabiliser le système (vent, mauvaise orientation, choc...).

Rajouter un générateur de couple (*Modelica.Mechanics.Rotational.Sources.Torque*) pour simuler cette perturbation. On lui assigne une valeur de -20 000N.m au bout de 0.8s



Question 12: Que doit faire le moteur pour encaisser cette perturbation.

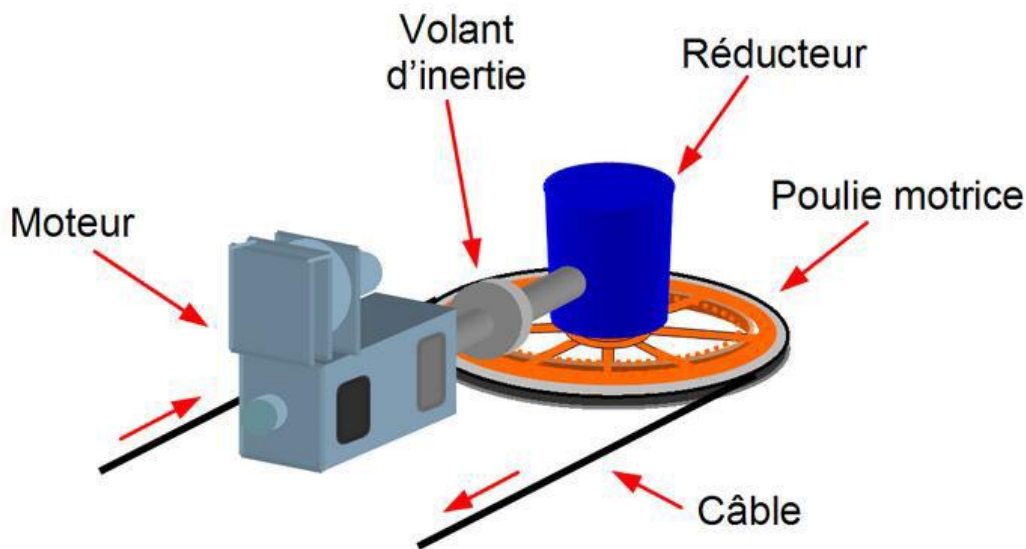
Question 13: Quel est l'écart statique après perturbation.

Par sécurité, l'intensité qui traverse le système électrique ne doit pas excéder 100A.

Question 14: En affichant la courbe de l'intensité, conclure sur le respect des conditions de sécurité.

Activité 5 - Etude de l'asservissement d'un télési

La présente activité a pour finalité de vous familiariser avec la structure PID très utilisée en régulation ou asservissement en étudiant un télési



Le choix du constructeur s'est porté sur un moteur de référence **LS 200 L** :

Puissance mécanique nominale P_n (kW)	30
Vitesse nominale N_n (tr/min)	2946
Couple nominale C_n (N.m)	97.2
Intensité nominale $I_{n(400V)}$ (A)	88.1
Intensité de démarrage/Intensité nominale I_d/I_n sous 400V	8.6
Couple de démarrage/Couple nominale C_d/C_n	2.7
Couple maximale/Couple nominale C_m/C_n	3.4
Bruit LP (dB)	73



Poulie

Rayon (m)	1,2
Masse (kg)	96

Le moment d'inertie du moteur sera négligé face au moment d'inertie J_p de la Poulie

Equations du moteur

- Tension aux bornes de la MCC : $U_{cc}(t) = K_m \cdot \omega_m(t) + R_{int} \cdot i(t)$
- Force électromotrice : $E(t) = K_m \cdot \omega_m(t)$
- Couple : $C_n(t) = K_c \cdot i(t)$

Pour ce moteur on considérera $K_m = K_c$

Identification des paramètres de la fonction de transfert

Question 1: Calculer la constante de couple K_c (N.m/A)

Question 2: Calculer la résistance interne R_{int} du moteur.

Question 3: Calculer le moment d'inertie de la poulie J_p

Question 4: Exprimer le théorème du moment dynamique sur l'arbre moteur.

Question 5: Exprimer $i(t)$ à partir de l'expression du moment dynamique

Question 6: Exprimer $i(t)$ à partir de l'équation caractéristique du moteur.

Question 7: En combinant les deux expressions de $i(t)$, retrouver l'équation différentielle caractéristique de la fonction de transfert de la vitesse de rotation du moteur $\omega_m(t)$ en fonction de tensions d'alimentation $U_{cc}(t)$

Rappel : L'équation temporelle qui régit un système du 1er ordre est une équation différentielle linéaire du 1er ordre (à coefficients constants), elle s'écrit :

$$\tau \frac{ds(t)}{dt} + s(t) = K \cdot e(t)$$

Question 8: Par identification, retrouver les coefficients τ (constante de temps) et K (gain)

Asservissement de la vitesse de montée

Pour assurer un débit de skieurs convenable, le téléski doit pouvoir maintenir une vitesse de montée de 2 m.s^{-1}

Le rapport de réduction entre le moteur et la poulie est $r = 200$

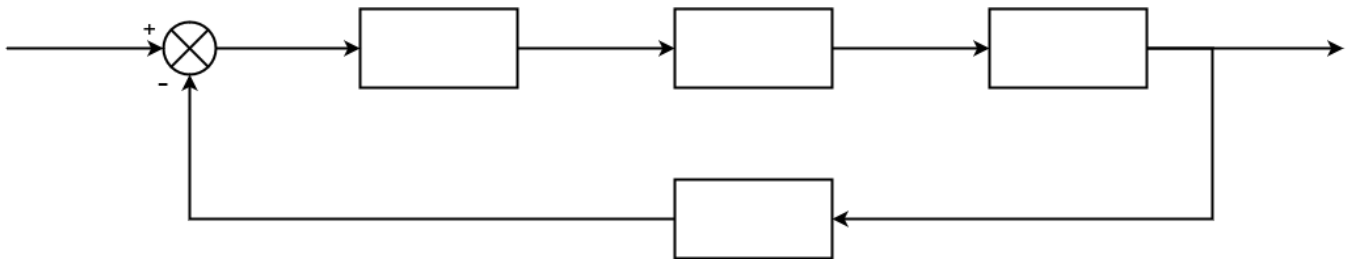
Question 9: Compléter le schéma bloc de l'asservissement en vitesse avec les éléments suivants :

Éléments fonctionnels

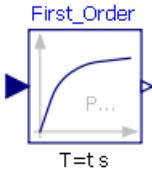
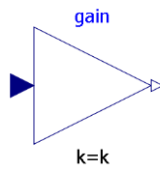
- Moteur
- Réducteur
- Poulie
- Capteur de vitesse linéaire

Flux

- Consigne : $V_{cons}(t)$
- Ecart : $\varepsilon(t)$
- Vitesse de rotation du moteur : $\omega_m(t)$
- Vitesse de rotation du réducteur : $\omega_r(t)$
- Vitesse linéaire : $V_s(t)$
- Vitesse mesurée : $V_{capt}(t)$



Question 10: Transposer votre schéma bloc sur OpenModelica et lancer une simulation sur 100s.

Fonction de Transfert 1 ^{er} Ordre	Gain
	
<i>Modelica.Blocks.Continuous.FirstOrder</i>	<i>Modelica.Blocks.Math.Gain</i>
Le bloc « fonction de transfert du 1 ^{er} Ordre » permet la modélisation d'une MCC. Les paramètres à remplir sont la constante de temps T_m et le gain calculé à la question 8.	Un simple gain permet la modélisation du réducteur et de la poulie.

Rappel : La consigne se modélise avec un échelon (*Modelica.Blocks.Sources.Step*) et le comparateur avec un feedback (*Modelica.Blocks.Math.Feedback*)

Question 11: Ce modèle de simulation est-il causal ou acausal ? Justifier votre réponse.

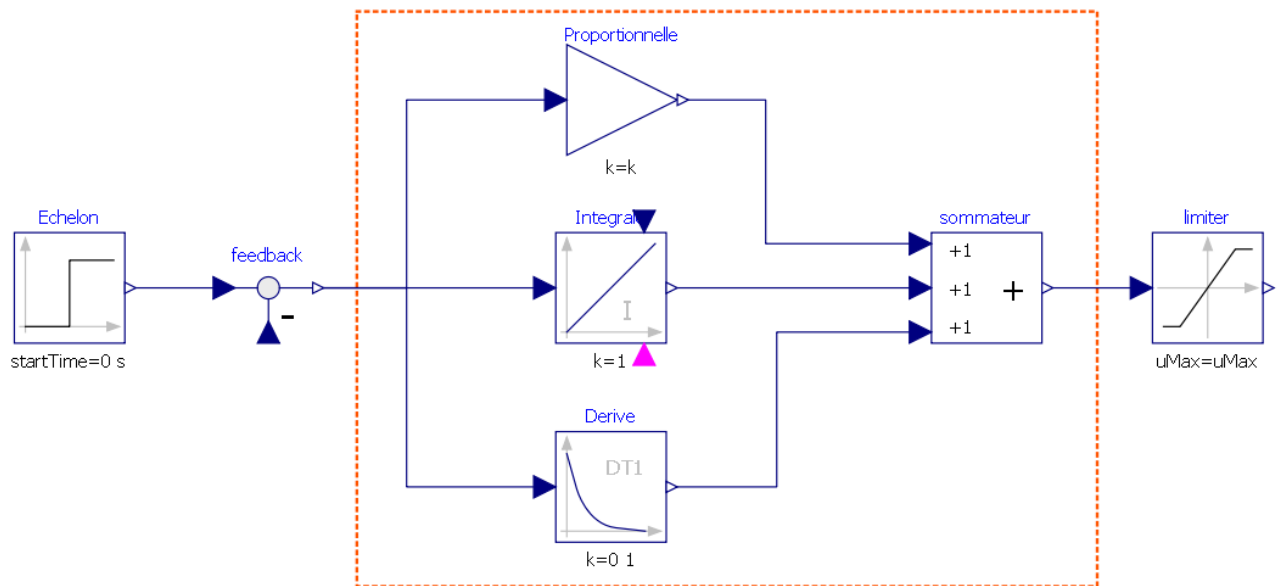
Question 12: Lancer une simulation pour 300s. Observer l'échelon et la vitesse de sortie. Ces résultats vous semblent-ils cohérent ?

Influence du correcteur PID

Afin de rectifier l'écart entre la consigne et la réponse du système, nous allons rajouter un correcteur PID entre le feedback et la fonction de transfert du 1^{er} Ordre

Rappel structure d'un correcteur PID :

Correcteur PID



Blocs utilisés :

- *Modelica.Blocks.Math.Gain*
- *Modelica.Blocks.Continuous.Integrator*
- *Modelica.Blocks.Continuous.Derivative*
- *Modelica.Blocks.Math.Add3*
- *Modelica.Blocks.Nonlinear.Limiter*

Rôle du limiteur

Le limiteur permet de limiter la valeur max d'un signal. Il peut s'agir d'une mesure visant à protéger les composants de très hautes tensions, ou tout simplement à représenter le fait qu'un système ne peut pas atteindre physiquement des valeurs trop hautes !

Le limiteur va donc influencer grandement les paramètres de notre PID.

Dans notre cas, la tension aux bornes du moteur ne peut pas excéder **800V**.

Question 13: Rajouter le PID à votre modèle.

Question 14: Ajuster les paramètres du PID pour arriver une vitesse de sortie stable avec :

- Un temps de réponse < 30s
- Un écart statique < 2%
- Un dépassement < 10%

Activité 6 - Modélisation de la régulation en température de la tête d'impression d'une imprimante 3D

Structure utilisée pour la partie chauffante de l'imprimante 3

L'imprimante 3D, objet de l'étude, réalise des objets par dépôt de couches successives de matière sur un plateau.

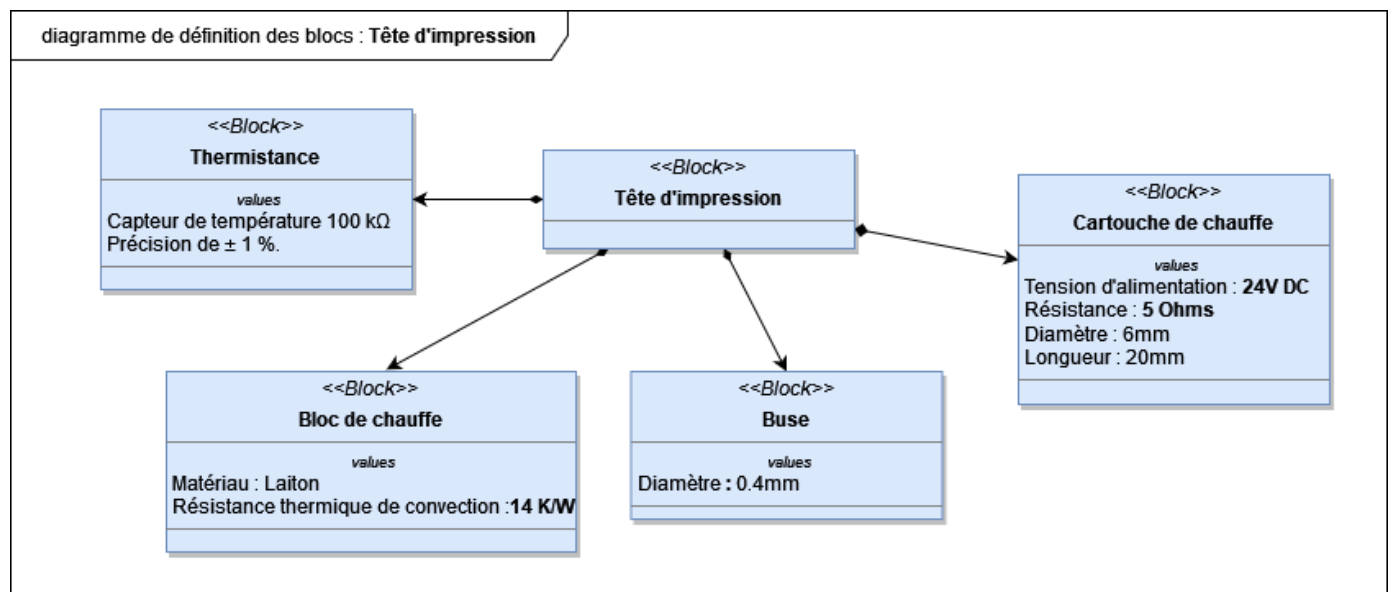
La majorité des imprimantes 3D personnelles fonctionnent selon de FDM (acronyme anglais de Fused Deposition Modeling qui signifie « modelage par dépôt de filament en fusion »). Ce procédé qui a été inventé en 1988 par la société Stratasys, est une marque déposée. On parle aussi de FFF (Fused Filament Fabrication) voir même de MPD (Molten Polymer Deposition) qui sont eux des termes libres de droits. Cette technique consiste à déposer couche par couche un filament de matière thermoplastique fondu à 200°C (en moyenne) qui en se superposant donne forme à l'objet.

La tête d'impression se déplace selon les coordonnées X, Y et Z (longueur, largeur et hauteur) transmises par un fichier 3D correspondant au modèle 3D de l'objet à imprimer.

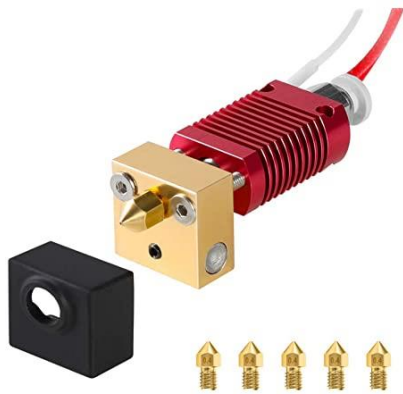
Le modèle d'imprimante 3D utilisé est une Créality Ender 3

La tête d'impression

Diagramme de définition des blocs :

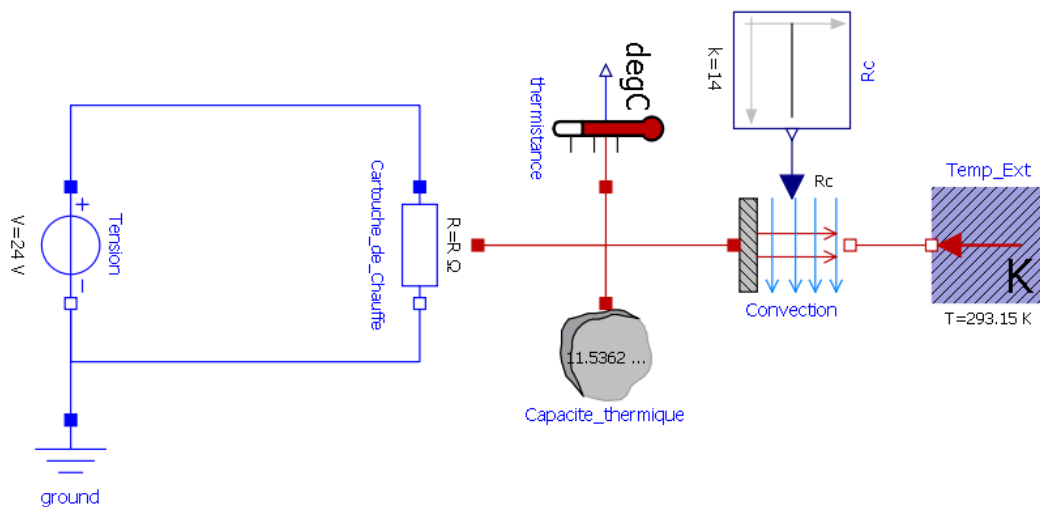


La résistance est alimentée par la carte de puissance située sur la carte électronique. La tension est stabilisée à 24V.



Fichiers à disposition :

- Modèles 3D de la tête d'impression
- Modèle Multiphysique : HeadPrint3D_part1.mo



Question 1: Compléter la valeur de la résistance de la Cartouche de chauffe à l'aide du diagramme de définition des blocs.

Pour l'ensemble de l'étude, on considérera l'ensemble bloc chauffant + cartouche de chauffe comme une pièce unique.

Paramètres physiques de la tête d'impression

La capacité thermique (ou anciennement capacité calorifique) d'un corps est une grandeur qui mesure la **chaleur** qu'il faut lui transférer pour **augmenter sa température d'un kelvin**. Inversement, elle permet de quantifier la possibilité qu'a ce corps d'absorber ou de restituer de la chaleur au cours d'une transformation pendant laquelle sa température varie.

Elle s'exprime en **Joule/Kelvin [J/K]** et se note **C**

C'est une grandeur extensive : plus la quantité de matière est importante, plus la capacité thermique est grande.

Ainsi :

$$C = c_p * m$$



$$C = c_p * \rho * V$$

Avec

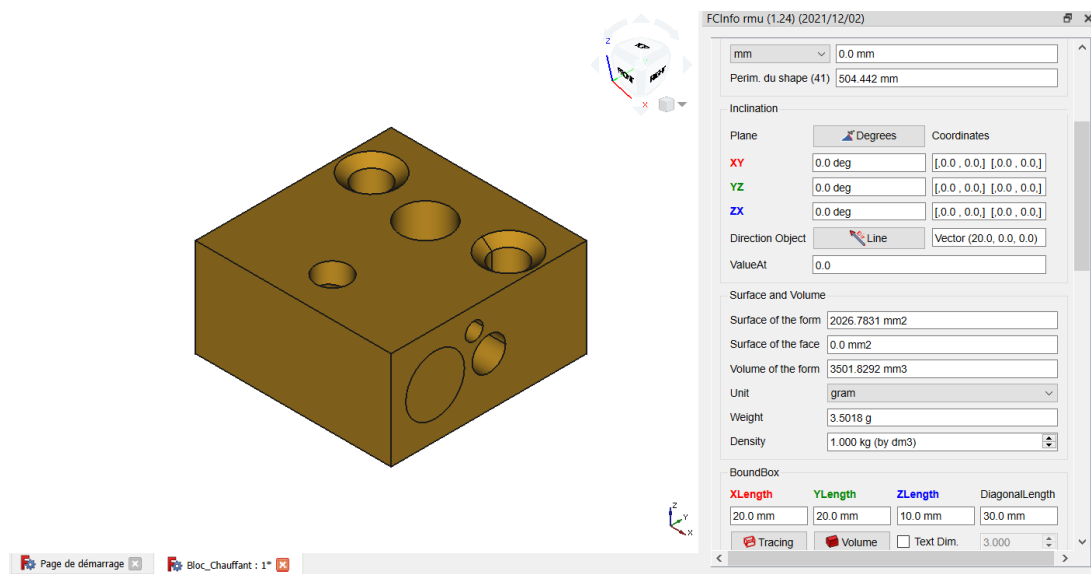
- C = Capacité thermique [J/K]
- m = masse du corps [Kg]
- ρ = masse volumique [Kg.m^{-3}]
- V = Volume [m^3]
- c_p = **capacité thermique massique** (anciennement appelée **chaleur spécifique**) en [$\text{J.K}^{-1}.\text{kg}^{-1}$]

Chaque matériau possède une valeur propre de capacité thermique massique :

https://fr.wikipedia.org/wiki/Capacit%C3%A9_thermique_massique

Question 2: Rechercher la valeur de la **Capacité thermique massique** et de la **masse volumique** du laiton ? (Rappel $1 \text{ g.cm}^{-3} = 1 \text{ kg.dm}^{-3}$)

Question 3: À L'aide du modèle FreeCad du bloc chauffant et de la macro **FCInfo** (voir la fiche ressource), déterminer la masse de la pièce ainsi que sa Capacité thermique. Renseigner cette valeur dans le modèle dans le bloc *Capacite_thermique*. Renseigner également la température ambiante.



Essai en Boucle Ouverte – Identification de modèle de comportement

Lancer une simulation pour 1000s et observer la courbe de température de la thermistance (*thermistance.T*).

Question 4: La réponse du système est-il une fonction de transfert du 1^{er} ou du 2^e Ordre ? Justifier votre réponse.

Question 5: Calculer le **Temps de réponse à 5 %**, le **gain K** et la **constante de temps τ** du système par rapport à la consigne de 24V

Régulation de température – Tout ou Rien

Maintenant que l'on dispose d'un modèle en boucle ouverte, nous pouvons réfléchir au type d'asservissement utilisé. Le cahier des charges stipule de pouvoir réguler la température à **+2 degrés**.

La correction de type **bang-bang** ou **tout ou rien** est la plus simple à mettre en œuvre. Elle consiste à chauffer à plein régime jusqu'à atteindre la valeur désirée puis à arrêter de chauffer. La température va donc baisser naturellement. On chauffe à nouveau si on descend de 2° par rapport à la valeur demandé.

On réalise cette régulation à l'aide de la fonction **hystérésis** (*Modelica.Blocks.Logical.Hysteresis*)

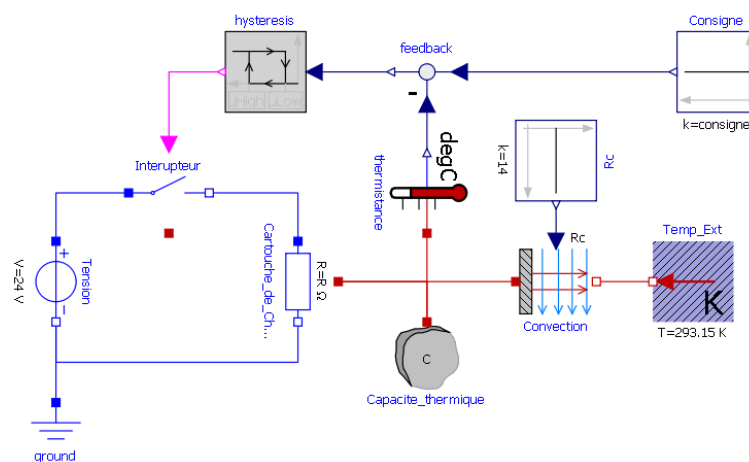
Ce bloc transforme un signal d'entrée réel en un signal de sortie booléen tout-ou-rien qui va ouvrir et fermer l'interrupteur avec la logique suivante :

- Lorsque la sortie était **fausse** et que l'entrée devient **supérieure** au paramètre **uHigh**, la sortie passe à **vrai**.
- Lorsque la sortie était **vraie** et que l'entrée devient **inférieure** au paramètre **uLow**, la sortie passe à **faux**.

Ouvrez le modèle HeadPrint3D_part2.mo.

Question 6: Compléter les paramètres du bloc hystérésis en fonction des données du problème.

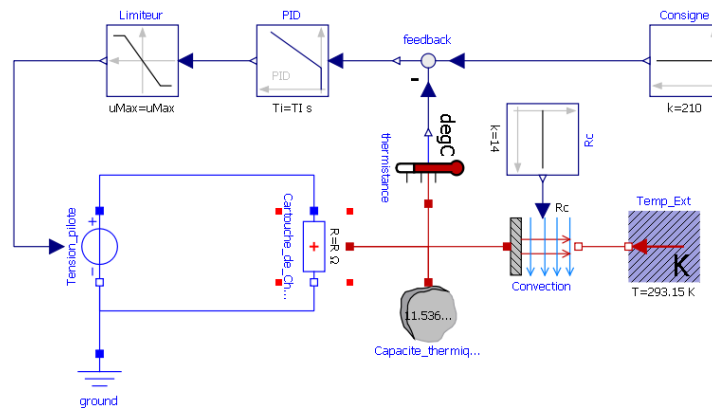
Question 7: Lancer une simulation sur 300s avec une consigne de 210°C. Commenter l'allure de la courbe de la température. La valeur consigne est-elle respecté ?



Régulation de température – Correcteur Proportionnel Intégral Dérivé (PID)

Afin de limiter la consommation électrique de la machine suite à la régulation tout ou rien, nous pouvons utiliser un correcteur de type proportionnel intégral dérivé (**PID**) afin de piloter directement la source de tension.

Ouvrir le modèle HeadPrint3D_part3.mo



Le limiteur traduit les limites physiques du générateur de tension. Ainsi les bornes de notre source de tension sont comprises en 0 et 24V.

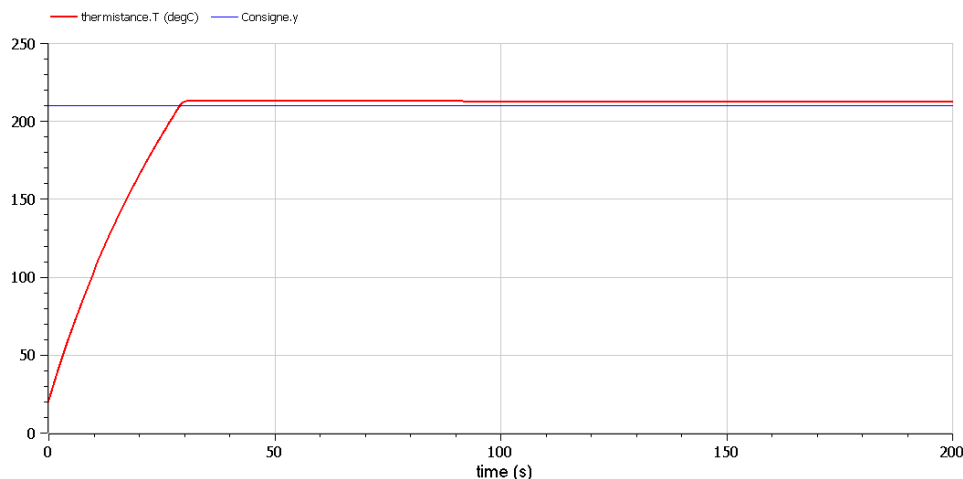
Question 8: Configurer le bloc limiteur.

Réglage des correcteurs

Le réglage de ces correcteurs, pour un tel processus, se fait de façon empirique. La méthode de réglage recommandée pour le P.I.D. est de régler d'abord K_p avec l'expression suivante :

$$K_p = \epsilon * \text{coef} \text{ (Ici coef = 0,3 pour optimiser la réponse) puis } K_i = \frac{K_p}{\tau} \text{ et } K_d = \frac{K_p}{2}$$

Question 9: A partir de ces expressions, en déduire les valeurs de K_p , T_i et T_d utilisées pour le réglage du correcteur afin d'arriver le plus proche de la réponse suivante :



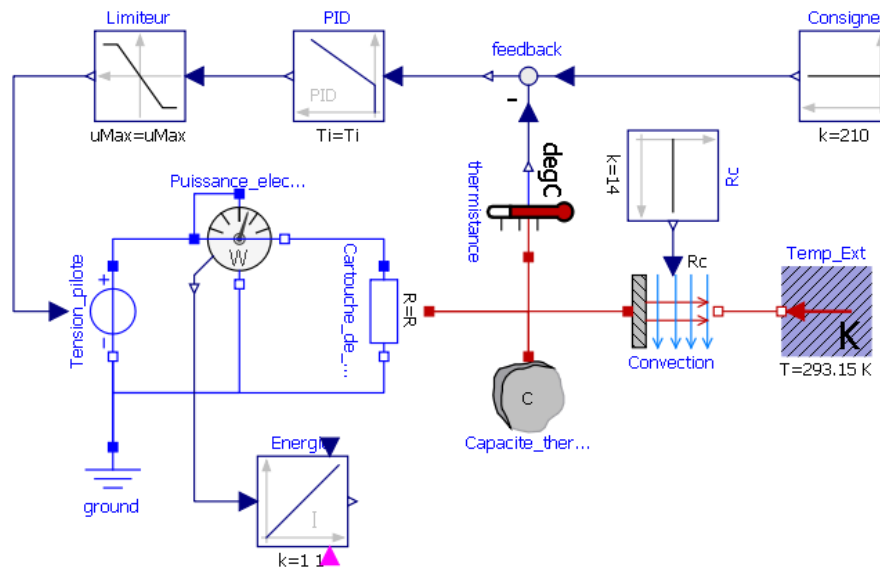
Question 10: Au bout de combien de temps le système se stabilise à la consigne sur la simulation.

Comparaison avec le modèle réel.

A l'aide de votre professeur, d'une caméra thermique et d'un chronomètre, configurer l'imprimante 3D pour une consigne de 210°C. Relever le temps que met le système à se stabiliser à 210°C. Comparer ce temps avec celui de la simulation. Commenter les écarts.

Calcul du coût d'une impression 3D

Ouvrir le modèle HeadPrint3D_part4.mo



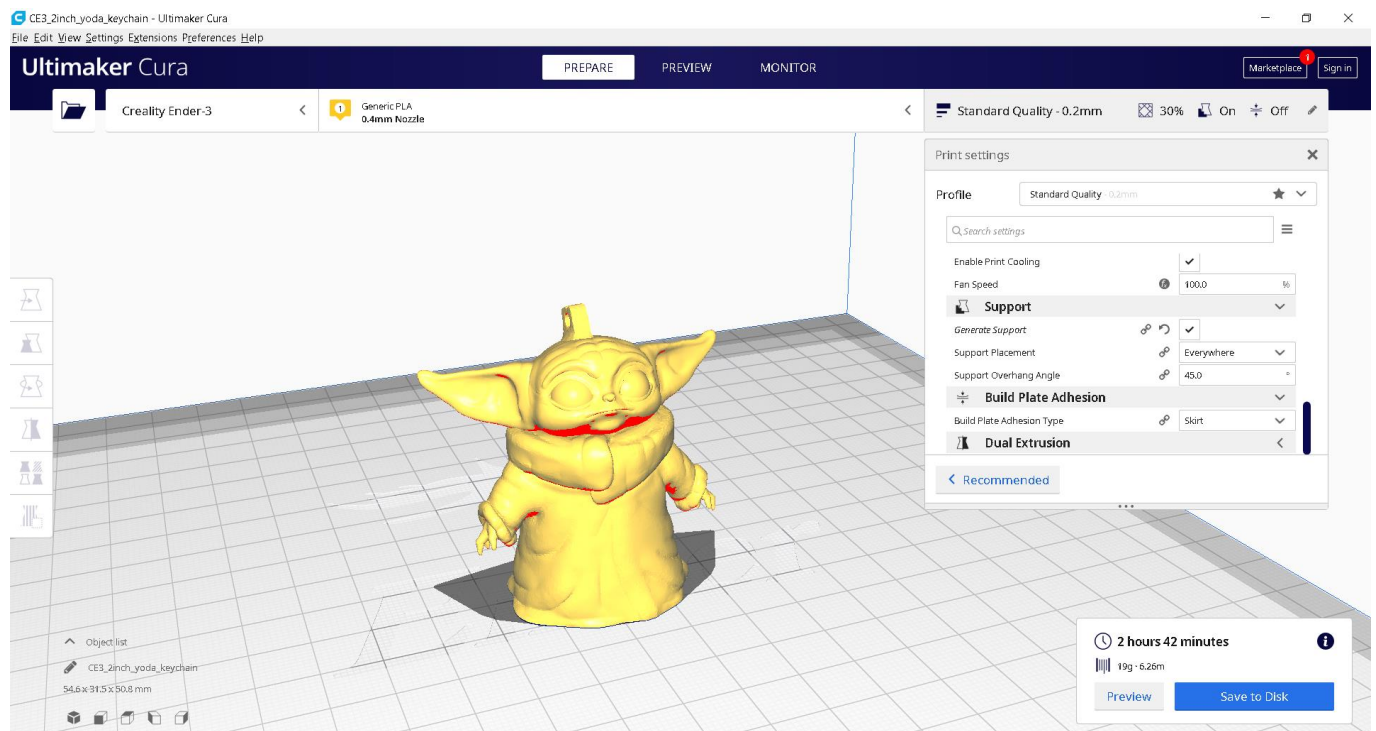
Dans ce modèle nous avons rajouté un capteur de puissance ainsi qu'un bloc *Integrator* pour calculer l'énergie électrique consommée par le système.

Question 11: Expliquer le rôle du bloc *Integrator* pour le calcul de l'énergie consommée ?

Question 12: Quel est l'unité de cette énergie ?

Question 13: Rajouter un gain (*Modelica.Blocks.Math.Gain*) à la sortie du bloc *Integrator* pour afficher l'énergie consommée en KWh.

Déterminer le temps d'une impression avec CURA



Ouvrez CURA puis le modèle Baby_Yoda.stl

Question 14: A l'aide de la fiche [ressource CURA](#), configurer le slicer pour une impression avec les paramètres suivant :

- Qualité Standard
- Température d'impression : 210°
- Température du plateau : 50°C
- Générer un support pour les porte-à-faux de plus de 45°

Question 15: Lancer le Slicer puis renseigner le temps d'impression dans le temps de simulation d'Openmodelica en seconde (vous ajouterez à ce temps le temps de chauffe de la tête d'impression).

Question 16: Sur internet, trouver le prix actuel du KWh en France.

Question 17: Déterminer le cout (€) en électricité pour l'impression de ce modèle.

Partie 3 – Modulation et démodulation des signaux

Activité 7 - L'Internet des Objets

Il n'existe pas une définition universelle de l'Internet des Objets (abréviation IdO en français ou encore IoT pour Internet of Things en anglais). Nous proposons la suivante :

« L'Internet des Objets est un dispositif pour lequel les objets –conçus par l'homme- mais également les animaux et les personnes se voient attribués d'identifiants uniques leur permettant de transmettre des données sur le réseau internet sans nécessiter aucune action humaine à humain ou humain à machine ».

Actuellement, de très nombreux acteurs se lancent dans l'Internet des Objets :

Les opérateurs de télécommunications ;

Les consortiums institutionnels tels que ceux à l'origine des normes standards :

- **WiFi** (IEEE 802.11),
- **Bluetooth** (IEEE 802.15.1),
- **Zigbee** (IEEE 802.15.4).

Un certain nombre de start-up tentant d'imposer leur standard.

Les deux premiers acteurs proposent une évolution de leurs produits alors que les derniers proposent une technologie dédiée à l'IdO en natif

Toutefois, il semble que se détachent très clairement trois technologies dont on peut penser qu'elles vont se partager une grande part du marché de l'IdO, au moins dans un avenir très proche. Il s'agit des technologies :

- **NB-IoT** (Narrow Band Internet of Things) proposée par l'organisme de standardisation des télécommunications 3GPP (3rd Generation Partnership Project) pour laquelle la transmission radio s'effectue via les réseaux cellulaires des opérateurs de télécommunications « classiques » ;
- **Sigfox**, proposée par l'opérateur de télécommunications toulousain éponyme dont l'usage est exclusivement dédié à l'Internet des Objets ;
- **LoRaWAN**, développée par le fondateur de Silicium américain Semtech, et qui propose des modules Radio-Fréquences dédiés à l'Internet des Objets.

Dans tous les cas ces communications sont maintenant quasi-exclusivement toutes **numériques** !



Cas des modulations numériques (1 symbole = 1 bit)

Le principe général de la modulation numérique est que le signal numérique, au niveau de l'émetteur, est modulé pour produire un signal de haute fréquence, transmis par ondes électromagnétiques, puis démodulé au niveau du récepteur pour être reconstitué à nouveau sous la forme d'un signal en bande de base.

La modulation a donc pour rôle d'adapter le signal au canal de transmission (milieu physique) sur lequel il doit être émis.

Remarque : Pour simplifier la simulation, les fréquences des porteuses seront volontairement prises très basses. Ce qui nous éloignera des réelles valeurs de fréquence utilisées dans les diverses normes citées ci-dessus. De la même façon, les séquences numériques seront transmises au rythme de seulement un symbole (un bit par exemple) par seconde.

Tous les modèles devront être rigoureusement enregistrés. Toutes les captures d'écran des résultats de simulation devront apparaître dans le compte-rendu.

Signal numérique à transmettre

Dans un message numérique, **deux états sont à coder**, le **0 logique** et le **1 logique**.

La **séquence binaire** à transmettre est composée d'une suite de symboles "**0**" et "**1**" (par exemple « 0 1 1 0 1 1 0 0 ») qui n'a pas de signification physique. Il s'agit d'un signal Booléen.

Pour transmettre l'information qu'elle représente, il faut représenter les "0" et "1" par un signal physique $s_0(t)$ et $s_1(t)$, appelé **forme d'onde du codage**.

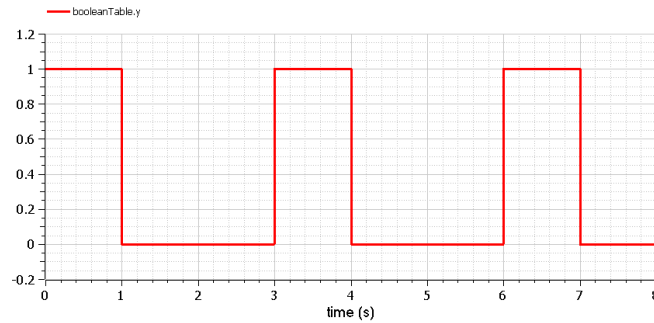
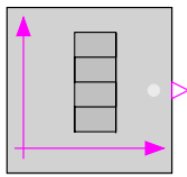
Question 1: Notre objet connecté souhaite émettre la valeur « **117** ». Coder cette valeur en binaire et remplir le chronogramme 1 du document réponse sachant que le débit binaire est de 1b/s.

Ouvrez le logiciel OpenModelica. Pour modéliser cet octet, nous allons utiliser le bloc BooleanTable (*Modelica.Blocks.Sources.BooleanTable*).

Fonctionnement du bloc : le bloc utilise une table de valeur (suite de nombre entre crochets séparés par des virgules {m,n} qui représente des instants (en seconde). À chaque t instant, la sortie change sa valeur à la valeur complémentaire de la précédente.

Exemple avec la Table = {0, 1, 3, 4, 6, 7}

booleanTable



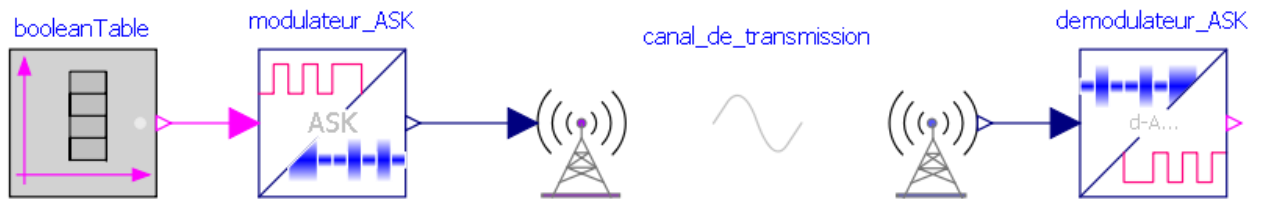
Question 2: Configurer le bloc booleanTable pour générer l'octet de la question 1

Modulation ASK

Charger le package : Modulation_Demodulation_Numerique.mo

Dans ce package vous retrouver l'ensemble des blocs permettant la modulation/démodulation numérique.

Question 3: Saisir le modèle suivant puis lancer la simulation sur 8s avec un nombre d'intervalles de 2500.



Configuration du bloc modulateur et démodulateur :

- $A_p = 1$
- $F_p = 5$
- Decalage = 0,2

Question 4: Afficher la courbe *modulateur_ASK.S*. Indiquer à quelle modulation correspond ce modèle ? Décrire la forme d'onde du « 0 » et du « 1 » logique et reproduire la courbe dans votre document réponse.

Question 5: Expliquer le principe de la modulation ASK. Vous pouvez ouvrir le modèle du bloc *modulateur_ASK* avec « clic droit » puis « Open Class » pour vous aider dans la compréhension.

Question 6: Afficher la courbe *demodulateur_ASK.S*. La sortie correspond-elle bien à l'octet à transmettre. Que s'est-il passé entre l'octet d'entrée et l'octet de sortie ? Ouvrez le modèle et expliquer ce phénomène ?

Question 7: Remplacer le Canal de transmission par un Canal de transmission **Bruité** ? Observer et commenter l'allure du signal à l'entrée du démodulateur ? Est-ce ce que ce bruit induit des erreurs dans la démodulation du signal ? Si oui calculer en % le taux d'erreur ?

Question 8: Quels sont les phénomènes pouvant être source de « bruit » dans la transmission du signal ?

Modulation FSK

Question 9: Saisir le modèle suivant puis lancer la simulation sur 8s avec un nombre d'intervalles de 2500.



Configuration du bloc modulateur et démodulateur :

- $A_{p1} = 1$
- $F_{p1} = 10$
- $A_{p2} = 1$
- $F_{p2} = 5$

Question 10: **Afficher** la courbe *modulateur_FSK.S*. **Indiquer** à quelle modulation correspond ce modèle ? **Décrire** la forme d'onde du « 0 » et du « 1 » logique et **reproduire** la courbe dans votre document réponse.

Question 11: **Expliquer** le principe de la modulation FSK. Vous pouvez ouvrir le modèle du bloc *modulateur_FSK* avec « clic droit » puis « Open Class » pour vous aider dans la compréhension.

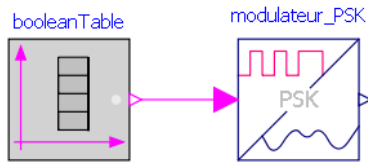
Question 12: **Afficher** la courbe *demodulateur_FSK.S*. La sortie correspond-elle bien à l'octet à transmettre. Que s'est-il passé entre l'octet d'entrée et l'octet de sortie ? Ouvrez le modèle et **expliquer** ce phénomène ?

Question 13: **Remplacer** le *Canal de transmission* par un *Canal de transmission Bruité* ? **Observer** et **commenter** l'allure du signal à l'entrée du démodulateur ? Est-ce ce que ce bruit induit des erreurs dans la démodulation du signal ? Si oui **calculer** en % le taux d'erreur ?

Question 14: **Rechercher** des systèmes ou des technologies utilisant une modulation/démodulation FSK ?

Modulation PSK

Question 15: Saisir le modèle suivant puis lancer la simulation sur 8s avec un nombre d'intervalles de 2500.



Question 16: Afficher la courbe *modulateur_PSK.S*. Indiquer à quelle modulation correspond ce modèle ? Décrire la forme d'onde du « 0 » et du « 1 » logique et reproduire la courbe dans votre document réponse.

Question 17: Expliquer le principe de la modulation PSK. Vous pouvez ouvrir le modèle du bloc *modulateur_PSK* avec « clic droit » puis « Open Class » pour vous aider dans la compréhension.

Cas des modulations numériques à symbole constitué de plus de 1bit

Dans la partie précédente, nous avons découvert comment émettre et recevoir une séquence binaire composée de symboles d'un bit "0" et "1".

Pour améliorer le débit binaire de la transmission nous avons intérêt à considérer des symboles composés de plusieurs bits (augmentation de l'efficacité spectrale).

Par exemple, la séquence binaire 1011100111110001, peut être vue :

- Comme une séquence de 16 symboles de longueur 1 bit (partie précédente)
- Ou comme une séquence de 8 symboles de longueur **2 bits** valant respectivement :

S1	S2	S3	S4
00	01	10	11
0	1	2	3

Pendant le même temps d'émission du symbole T_s (ici 1seconde), on transmet alors 2 bits.

Ainsi il est possible de multiplier par 2 le débit binaire ! Ainsi le signal modulé peut prendre plus de deux valeurs discrètes. Les symboles sont appelés des **polytons**.

Ce type de transmission **augmente cependant**, entre autres, la **complexité du système** pour réaliser la modulation/démodulation.

Dans notre exemple nous voulons maintenant transmettre une information en hexadécimale « **A38B** »

Question 18: Convertir la valeur en hexadécimal « A38B » en binaire. Découper ce mot binaire avec la méthode des symboles à plusieurs bits. On prendra 2 bits par symbole (donc des symboles de 0 jusqu'à 3). Sachant le temps d'émission d'un symbole est de 1s, reproduire l'allure de la courbe sur votre document réponse.

Pour modéliser cet octet, nous allons utiliser le bloc integerTable.

Modelica.Blocks.Sources.IntegerTable

Fonctionnement du bloc : Ce bloc génère un signal de sortie de nombre entier en utilisant une **table**.

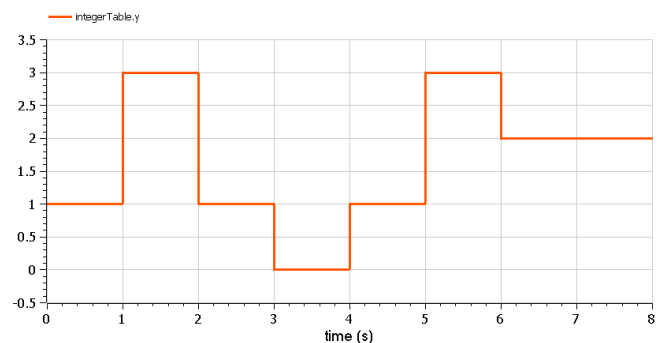
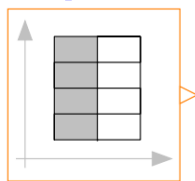
Les points temporels et les valeurs y sont stockés dans une table matricielle $[i,j]$, où la première colonne i contient le temps t et la deuxième colonne contient la valeur entière de la sortie y à ce moment t .

Les colonnes i et j sont séparées par des virgules et les lignes par des points-virgules.

Exemple avec la Table = [0, 1; 1, 3; 2, 1; 3, 0; 4, 1; 5, 3; 6, 2; 7, 2]

Temps (s)	Valeur
0	1
1	3
2	1
3	0
4	1
5	3
6	2
7	2

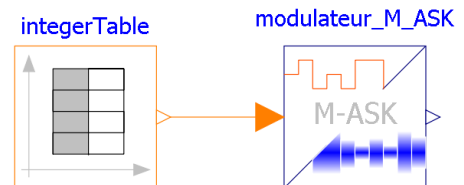
integerTable



Question 19: Configurer le bloc integerTable pour générer la courbe de la question 16

Cas de la modulation 4-ASK

Ici la modulation d'une séquence binaire composée de symboles de 2 bits soit 4 polytons.

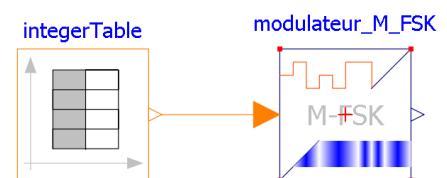


Question 20: Saisir le modèle suivant puis lancer la simulation avec une valeur de $F_p = 5$

Question 21: Afficher la courbe modulateur_M_ASK.S et décrire la forme d'onde des symboles « 0 », « 1 », « 2 » et « 3 ». Reproduire la courbe dans votre document réponse. Expliquer l'intérêt de la modulation 4-ASK par rapport à la modulation ASK.

Cas de la modulation 4-FSK

Ici la modulation d'une séquence binaire composée de symboles de 2 bits soit 4 polytons.



Question 22: Saisir le modèle suivant puis lancer la simulation avec $F_g = 1$

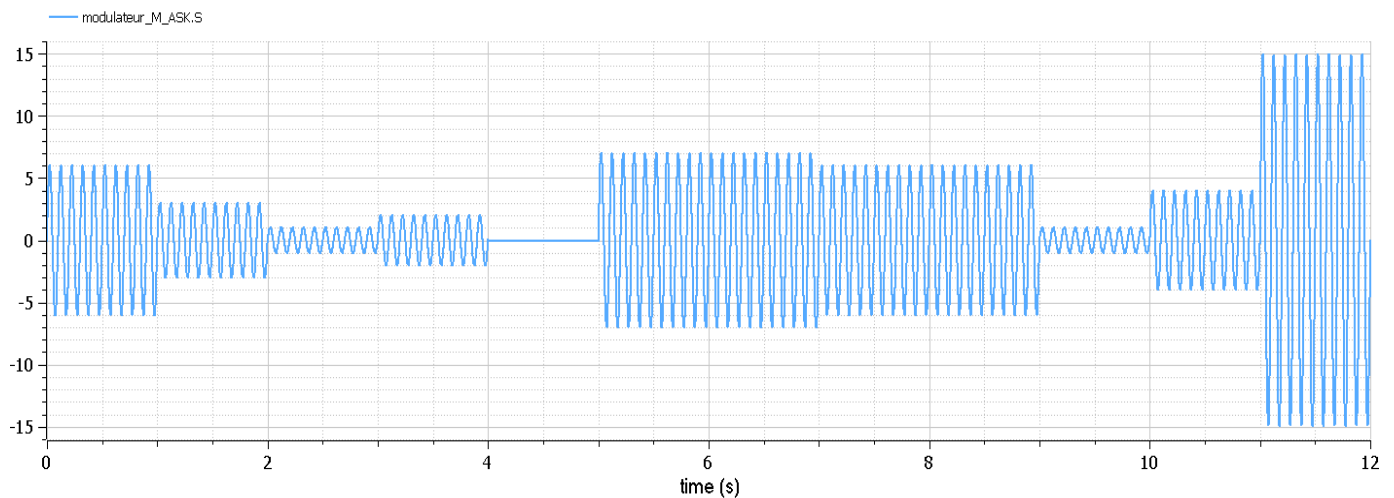
Question 23: Afficher la courbe modulateur_M_FSK.S et Décrire la forme d'onde des symboles « 0 », « 1 », « 2 » et « 3 ». Reproduire la courbe dans votre document réponse. Expliquer l'intérêt de la modulation 4-FSK par rapport à la modulation FSK.

Cas de la modulation 16-ASK

Dans de nombreux cas, le codage du message à transmettre se fait sur 16 symboles (16 Polytons).

Question 24: Combien de bits possèdent chaque symbole ?

Question 25: Un message ASCII codé en modulation 16-ASK a été intercepté. Décoder le message à partir de la courbe ci-dessous :



Rappel table ASCII :

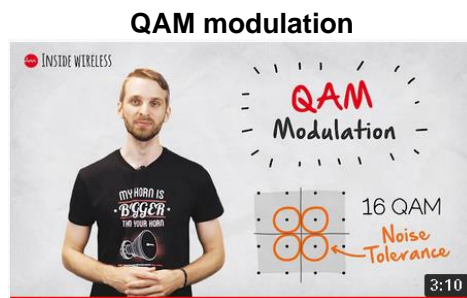
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	<u>BEL</u>	<u>BS</u>	<u>HT</u>	<u>LF</u>	VT	FF	<u>CR</u>	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	<u>ESC</u>	FS	GS	RS	US
2	<u>SP</u>	!	"	#	\$	%	&	'	()	*	±	•	=	~	/
3	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	:	:	≤	≡	≥	?
4	<u>@</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>	<u>G</u>	<u>H</u>	<u>I</u>	<u>J</u>	<u>K</u>	<u>L</u>	<u>M</u>	<u>N</u>	<u>O</u>
5	<u>P</u>	<u>Q</u>	<u>R</u>	<u>S</u>	<u>T</u>	<u>U</u>	<u>V</u>	<u>W</u>	<u>X</u>	<u>Y</u>	<u>Z</u>	[\]	^	_
6	`	<u>a</u>	<u>b</u>	<u>c</u>	<u>d</u>	<u>e</u>	<u>f</u>	<u>g</u>	<u>h</u>	<u>i</u>	<u>j</u>	<u>k</u>	<u>l</u>	<u>m</u>	<u>n</u>	<u>o</u>
7	<u>p</u>	<u>q</u>	<u>r</u>	<u>s</u>	<u>t</u>	<u>u</u>	<u>v</u>	<u>w</u>	<u>x</u>	<u>y</u>	<u>z</u>	{		}	~	<u>DEL</u>

Cas de la modulation M-QAM

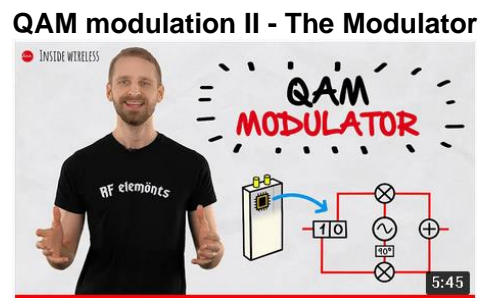
Parmi toutes les modulations numériques, il faut souligner l'existence de la modulation M-QAM que l'on va nommer simplement « **modulation de phase et d'amplitude** ».

Les modulations M-QAM sont **très largement utilisées** en raison de leur « efficacité » de fonctionnement (plus faible bande passante notamment, sensibilité au bruit, etc...).

Vidéo explicative :



<https://youtu.be/lbUflaeJcU8>



<https://youtu.be/YnWCRUoTEAI>

Une modulation nommée 4-QAM aura $2^2 = 4$ symboles différents.

Lorsqu'on utilise des symboles de longueurs 6 bits, on parle de modulation 64-QAM.

La modulation 64-QAM (6 bits par symbole) est largement très utilisée dans les communications numériques notamment pour la télévision numérique terrestre TNT mais aussi pour la 3G/4G.

Pour profiter de la modulation 256-QAM, il faut un smartphone compatible. Cette compatibilité est indiquée sur les fiches techniques des smartphones.

Extrait d'article : <https://www.universfreebox.com/article/52188/Technologies-utilisees-par-Free-qu-est-ce-que-la-modulation-256-QAM-recemment-activee-par-l-operateur>

« Après avoir proposé la modulation 64 QAM (Quadrature Amplitude Modulation, soit Modulation d'Amplitude en Quadrature en français), Free a récemment activé la modulation 256 QAM sur son réseau mobile en France Métropolitaine, et plus récemment encore à La Réunion. Mais à quoi sert cette technologie ?

Dans les faits, cela permet de booster les débits. Le débit maximum théorique descendant sur le réseau Free Mobile en France passe ainsi de 337,5 Mbit/s (en 64 QAM) à 440,55 Mbit/s (en 256 QAM). Une augmentation de l'ordre de 30 %, ce qui est loin d'être négligeable.

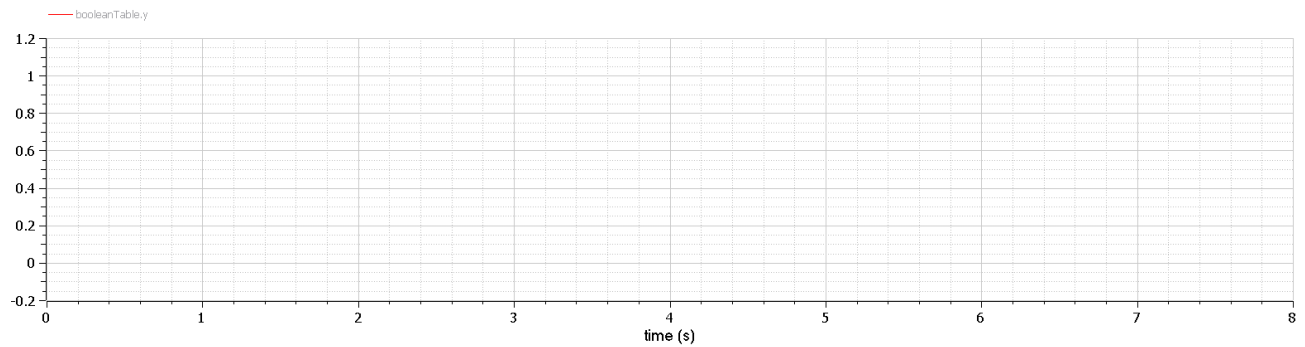
Sur le plan technique, il s'agit dans les grandes lignes d'augmenter le nombre de données transmises sur une période de signal. Le 64 QAM permet en effet d'encoder jusqu'à 64 valeurs (encodées sur 6 bits) par période, contre 256 valeurs (encodées sur 8 bits) par période pour le 256 QAM.

En pratique, cet exploit nécessite un signal de bonne qualité, puisque la lecture du signal se fait avec des marges beaucoup plus réduites (4 fois plus réduites que le QAM 64), ce qui augmente le risque de lire une donnée fausse due aux parasites électromagnétiques. »

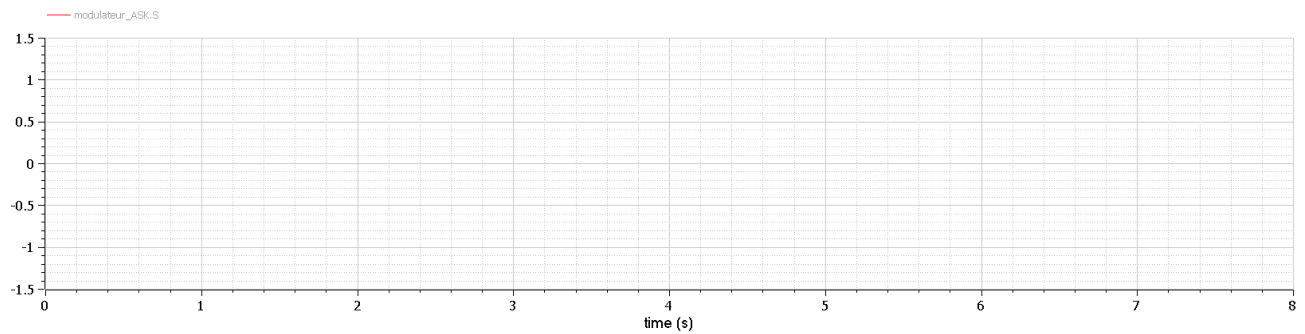
Question 26: Essayer d'expliquer dans les grandes lignes le principe de fonctionnement de cette modulation. Quels sont les **avantages** et les **inconvénients** par rapport aux autres principes de modulation ? **Citez** quelques technologies utilisant la modulation 256-QAM ?

Document réponse

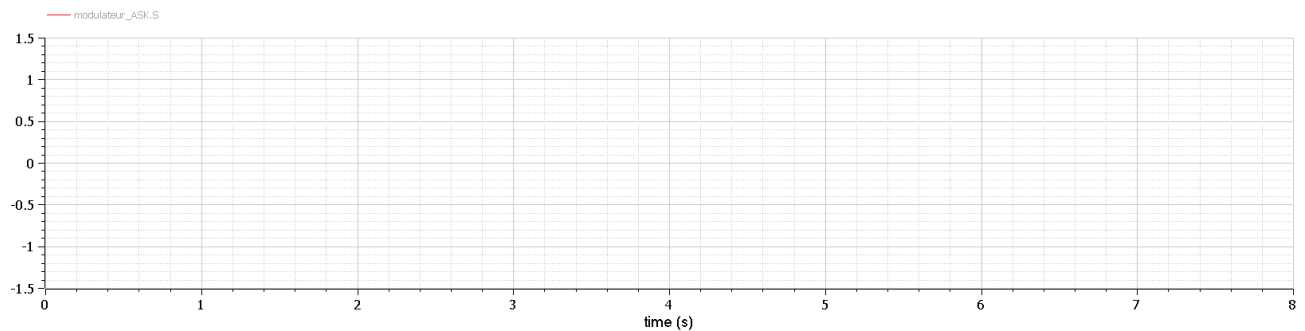
Octet



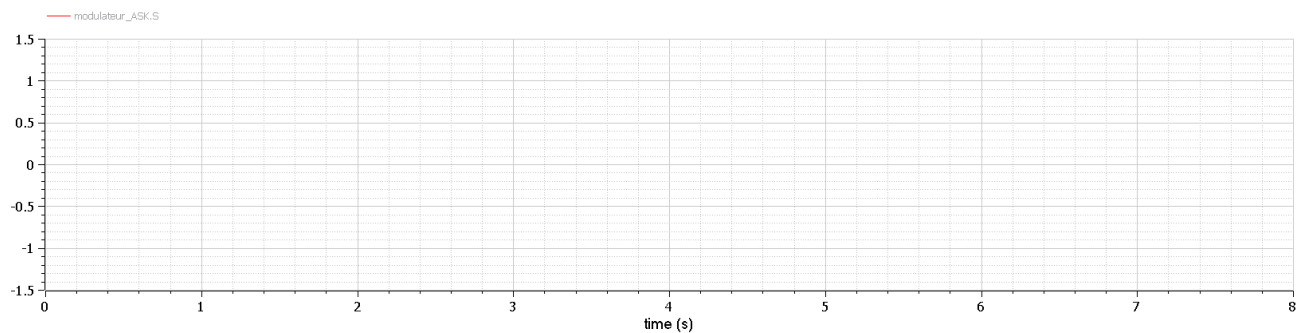
Modulation ASK



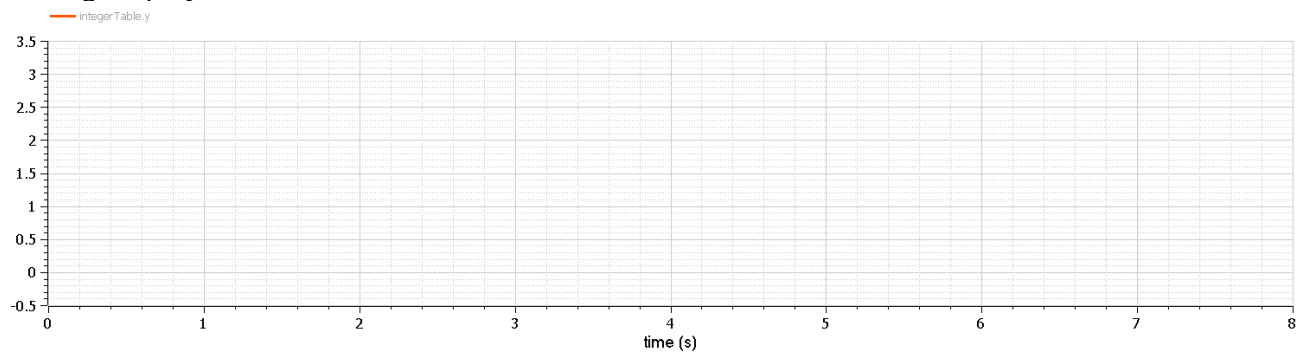
Modulation FSK



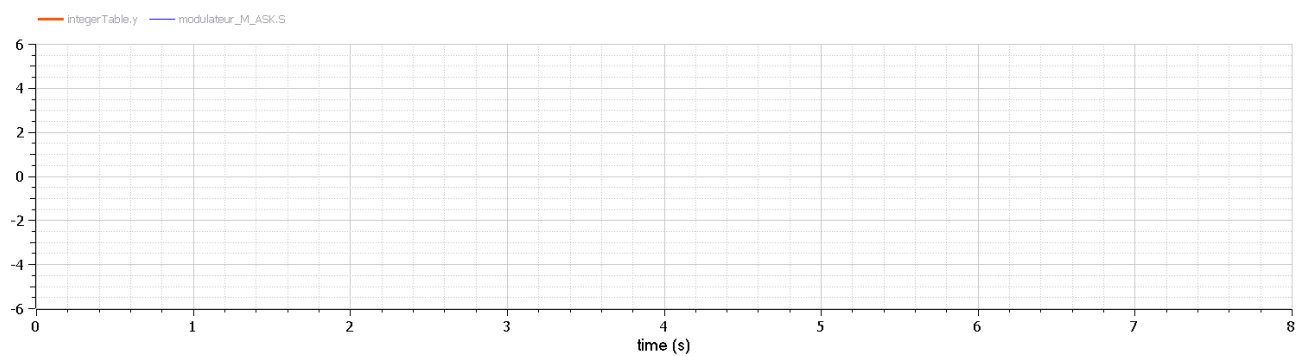
Modulation PSK



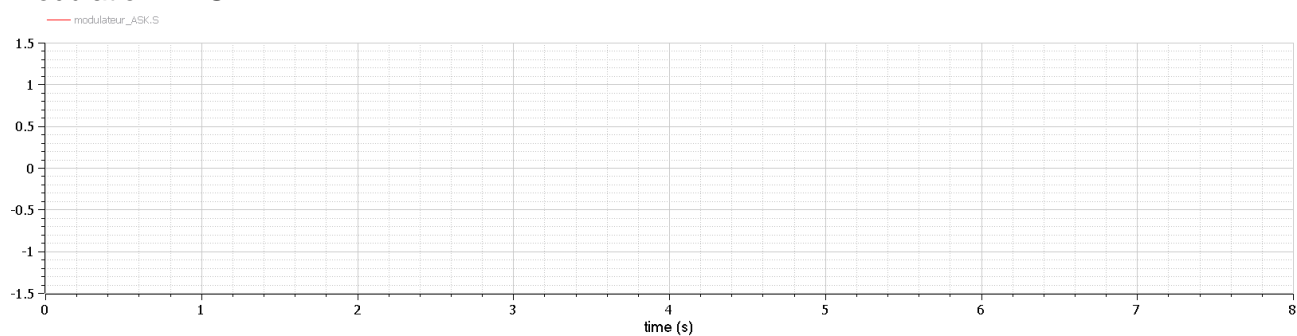
Codage 4 polytons



Modulation 4-ASK



Modulation 4-FSK



Décodage 16-ASK

