Modélisations de composants spécifiques de la chaîne d'information

publié le 30/04/2020 - mis à jour le 17/11/2020

Favoriser l'usage de composants paramétrables pour modéliser les fonctions de la chaîne d'information

Descriptif :

Les fonctions de la chaîne d'information, très souvent linéaires, peuvent être assimilées à de simples coefficients, ce qui, dans Sinusphy, est prévu par des composants génériques (ampli_kp, correcteur P, capteur...), mais il est avantageux d'avoir des modèles plus spécifiques qui rappellent les caractéristiques de la fonction.

Sommaire :

- 1- Création de 2 modèles paramétrables de CAN (ADC) :
- 2- Création de 2 modèles paramétrables de CNA (DAC) :
- 3- Création d'un modèle paramétrable de MLI (PWM) :
- 4- Création d'un modèle paramétrable de potentiomètre de recopie :
- 5- Documents ressources liés à cet article :

Quatrième volet d'une série de 7 articles amenant à maîtriser la modélisation de composants de la chaîne d'énergie et de la chaîne d'information dans le but d'effectuer la simulation d'un asservissement numérique de position la plus précise possible afin que le comportement réel soit totalement prévisible.

Les fonctions de la chaîne d'information, très souvent linéaires, peuvent être assimilées à de simples coefficients, ce qui, dans Sinusphy, est prévu par des composants génériques (*ampli_kp, correcteur P, capteur...*), mais il est avantageux d'avoir des modèles plus spécifiques qui font référence aux caractéristiques de la fonction. La prise en main par les élèves en est d'autant plus simple et devient une application du cours.

Cet article va permettre de comprendre comment simuler, forcément en CAUSAL, des fonctions spécifiques comme des CAN -convertisseurs analogiques numériques- (*ADC*), des CNA -convertisseurs numériques analogiques- (*DAC*), des MLI -modulation de largeur d'impulsion- (*PWM*) et également un capteur de position (*position sensor*) connu sous le nom de potentiomètre de recopie (*feedback potentiometer*).

Ces fonctions sont généralement décrites par leurs aspects techniques (*quantum, résolution, format binaire, nombre de combinaison, codage, tension de référence...*) mais plus rarement par leur fonction de transfert (*coefficient directeur et décalage à l'origine*). Le but est de réaliser des modèles qui déduisent l'équation reliant la sortie à l'entrée en fonction des valeurs des paramètres.

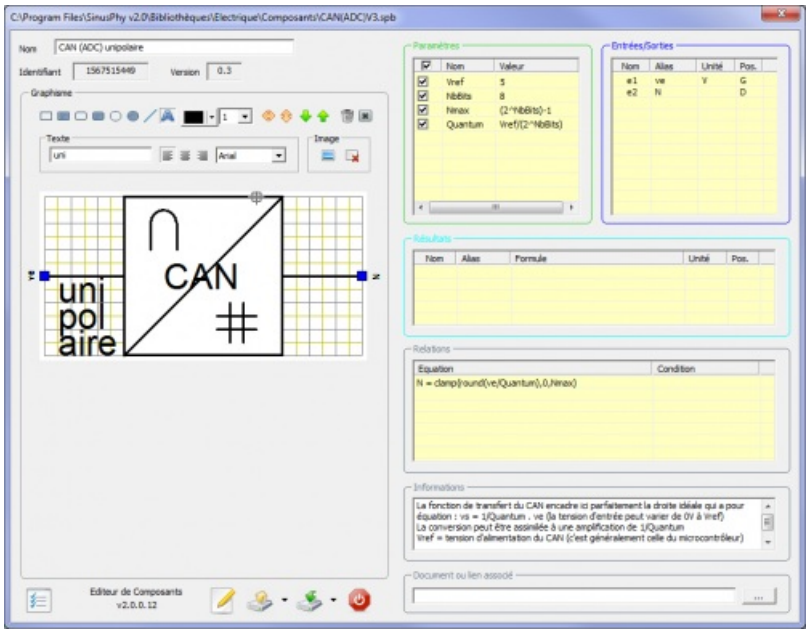
1- Création de 2 modèles paramétrables de CAN (ADC) :

Sans parler de technologie, il existe 2 types de CAN :

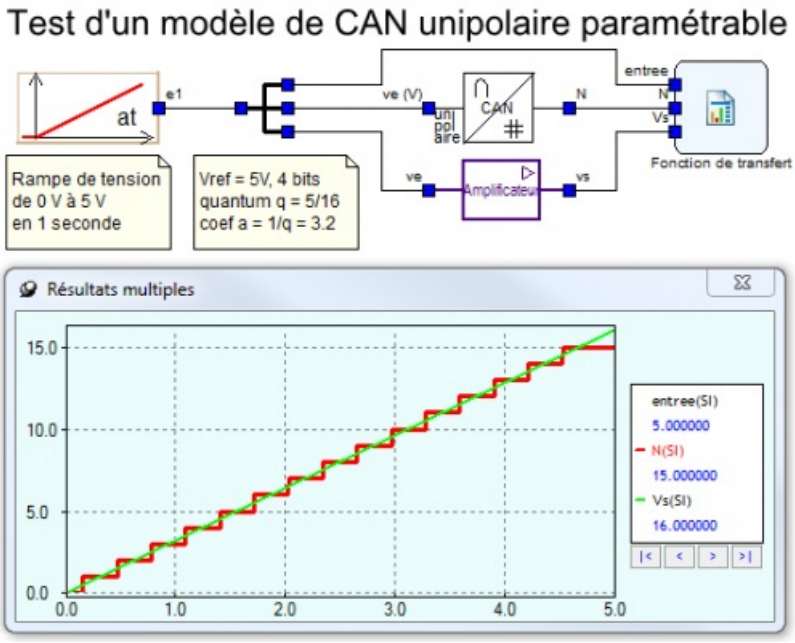
- Les CAN unipolaires (*intégrés dans les micro-contrôleurs*) où la tension d'entrée ne varie qu'entre 0 V et une tension de référence (*généralement la tension d'alimentation du micro-contrôleur : soit 5 V ou 3,3 V*)
- Les CAN bipolaires (*principalement utilisés dans le traitement de signal : acquisition du son, oscilloscopes numériques...*) où le signal d'entrée est, par nature, centré autour de 0 V.

Un CAN est principalement défini par le nombre de bits et la tension de référence (*on néglige généralement le temps de conversion, forcément optimisé*).

Pour un CAN unipolaire, il est possible de déduire le nombre de combinaisons, le nombre maximum, le quantum (*appelé aussi résolution*) et le coefficient directeur de la fonction de transfert coef a = 1/quantum.



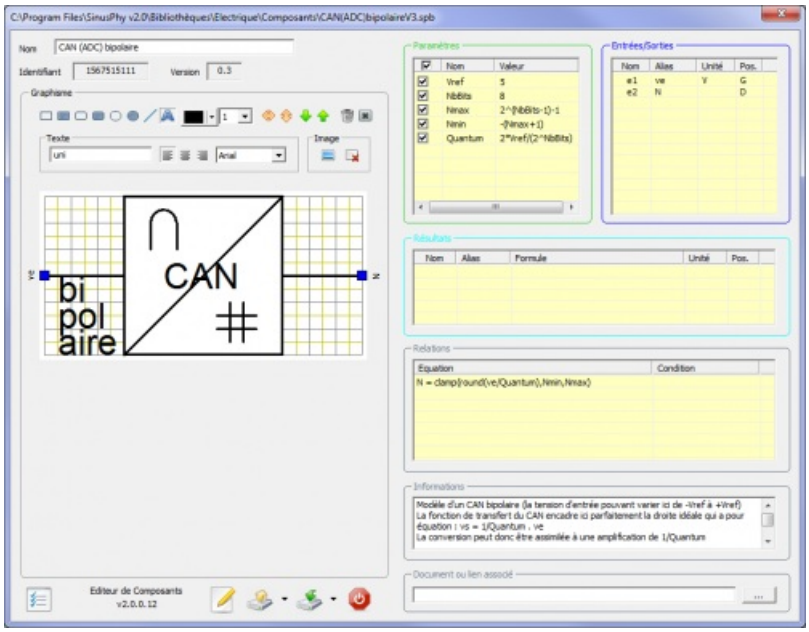
- **Nmax = (2^NbBits)-1**
- **Quantum = Vref/(2^NbBits)**
- **N = clamp(round(ve/Quantum), 0, Nmax)**
  La fonction round arrondit au nombre entier le plus proche, ce qui a pour effet d'obtenir une fonction de transfert idéale, c'est à dire entourant la droite de pente 1/quantum.
- La modélisation des composants de la chaîne d'information est principalement de type CAUSAL. Ceci est obtenu ici par l'utilisation de fonctions particulières (*clamp, round*) qui rendent l'équation parfaitement irréversible (*une éventuelle modification de N ne pourra pas modifier la tension ve*).



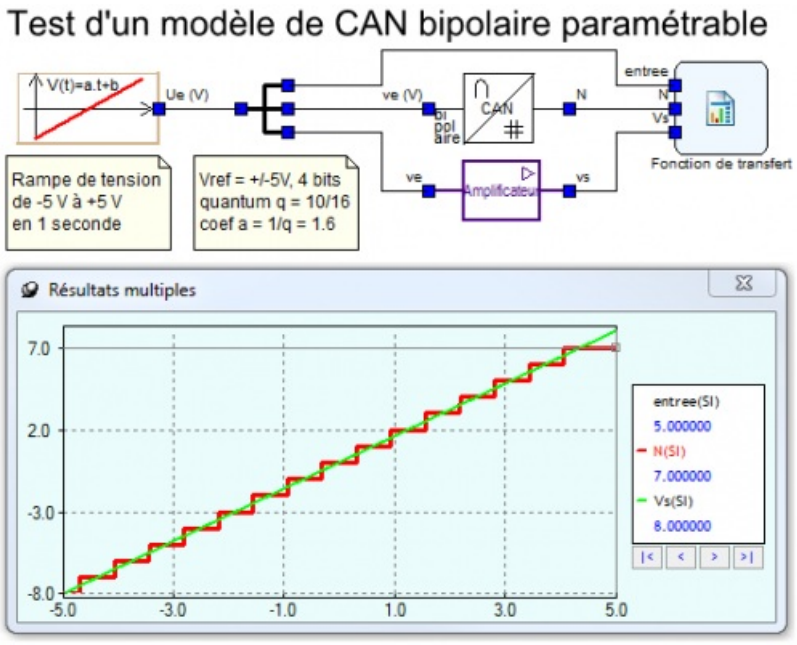
Le schéma de test, ci-dessus, valide le calcul du quantum et du coefficient directeur (*l'amplificateur témoin est paramétré avec coefficient directeur du CAN (coef a = 1/quantum)*). Il montre ainsi que le CAN pourrait, en effet, être simplifié par un simple coefficient si le nombre de bits est suffisamment important (*10 ou 12 bits en général*) afin que la hauteur des "marches" soit de l'ordre du mV. Si les variations de l'entrée dépassent les limites, le modèle

limitera N dans l'intervalle [0, Nmax].

Dans sa version bipolaire, les équations sont légèrement différentes. Comme la valeur de N est signée, la moitié des combinaisons est positive ou nulle et l'autre moitié est strictement négative. Ici, le choix a été fait d'autoriser le signal d'entrée à varier, comme dans beaucoup de CAN bipolaires, de -Vref à +Vref, ce qui a pour effet de doubler le quantum (*cela correspond à la solution classique qui consiste à transformer un CAN unipolaire en CAN bipolaire grâce à un montage à 2 résistances parfaitement identiques*).



- **Nmax = 2^(NbBits-1)-1**
- **Nmin = -(Nmax+1)**
- **Quantum = 2*Vref/(2^NbBits)**
- **N = clamp(round(ve/Quantum), Nmin, Nmax)**
  La valeur du nombre entier N est limitée à l'intervalle [Nmin, Nmax].
  Exemple : Sur 10 bits, N restera compris entre -512 et +511 même si la tension d'entrée sort de l'intervalle [-Vref, +Vref]



Ici aussi, le schéma de test, avec son amplificateur témoin, permet de valider le calcul du quantum et du coefficient directeur.

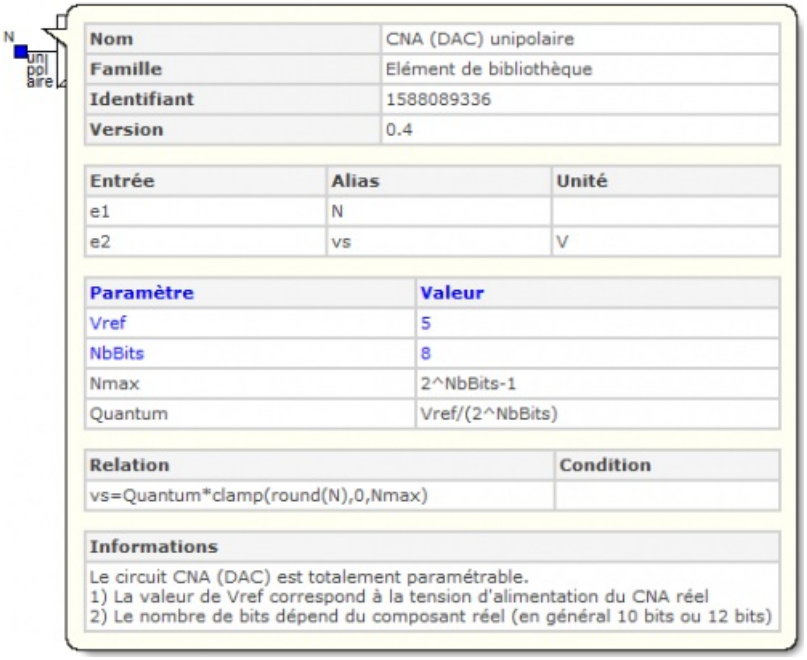
● 2- Création de 2 modèles paramétrables de CNA (DAC) :

▸ Il existe également 2 types de CNA :

- Les CNA unipolaires (*certains micro-contrôleurs récents en intègrent un ou deux*) où la tension de sortie peut varier entre 0 V et une tension de référence correspondant à la tension d'alimentation du composant,
- Les CNA bipolaires (*principalement utilisés dans le traitement de signal*) où la tension de sortie est centrée autour de 0 V et le code binaire est interprété, soit en code complément à 2 (*CPL2 ou two's complement*), soit en code binaire décalé (*offset binary*).

▸ Les CNA, comme les CAN, sont donc définis par le format binaire (*nombre de bits et codage*) et la tension de référence.

▸ Pour le modèle de CNA unipolaire, il faut calculer Nmax et le quantum pour en déduire la tension de sortie vs.
vs = Quantum*clamp(round(N), 0, Nmax)



Nom	CNA (DAC) unipolaire
Famille	Elément de bibliothèque
Identifiant	1588089336
Version	0.4

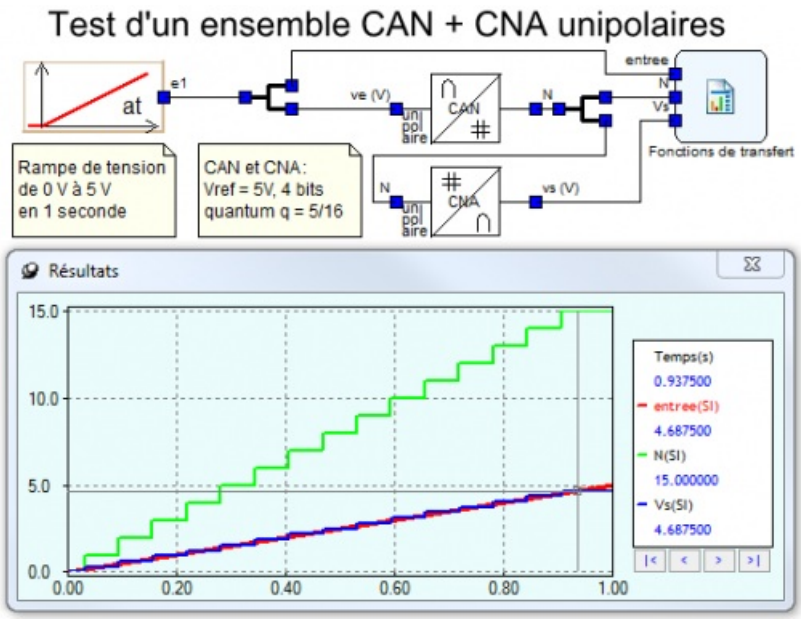
Entrée	Alias	Unité
e1	N	
e2	vs	V

Paramètre	Valeur
Vref	5
NbBits	8
Nmax	2^NbBits-1
Quantum	Vref/(2^NbBits)

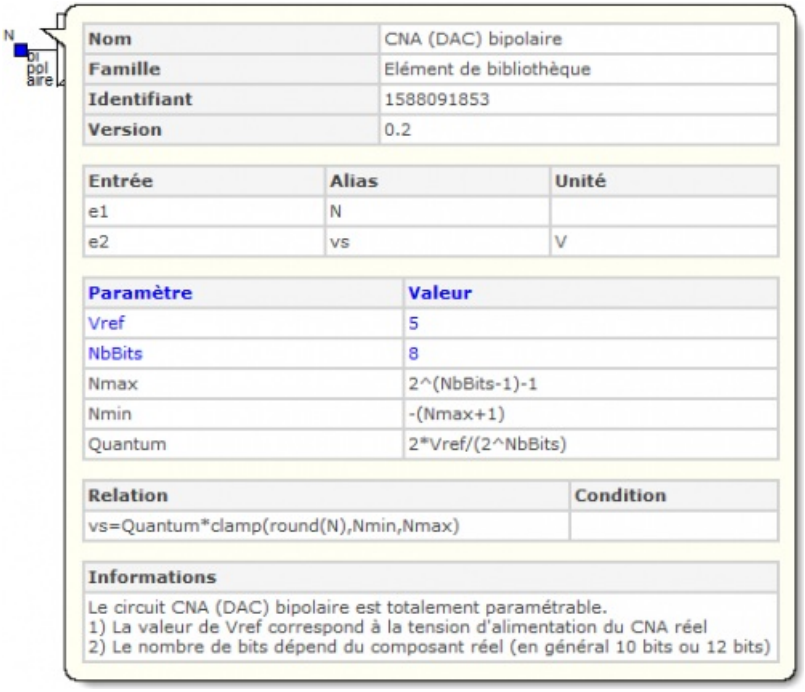
Relation	Condition
vs=Quantum*clamp(round(N),0,Nmax)	

Informations
Le circuit CNA (DAC) est totalement paramétrable. 1) La valeur de Vref correspond à la tension d'alimentation du CNA réel 2) Le nombre de bits dépend du composant réel (en général 10 bits ou 12 bits)

▸ Le schéma de test, ci-dessous, utilise le CAN validé précédemment pour vérifier le fonctionnement du CNA. Les 2 modèles de composants (*CAN et CNA*) sont configurés de la même manière : Vref = 5 V, NbBits = 4
Cette double conversion existe dans la plupart des systèmes numériques chargés de traiter des grandeurs analogiques. Cela montre que le signal qui ressort du CNA peut être quasiment le même (*à basse fréquence seulement !*) que le signal d'entrée si le nombre de bits est important (*10 bits à 12 bits*).



Pour le CNA bipolaire, l'équation de la tension de sortie, vs est légèrement différent :
vs = Quantum*clamp(round(N), Nmin, Nmax)



Nom	CNA (DAC) bipolaire
Famille	Elément de bibliothèque
Identifiant	1588091853
Version	0.2

Entrée	Alias	Unité
e1	N	
e2	vs	V

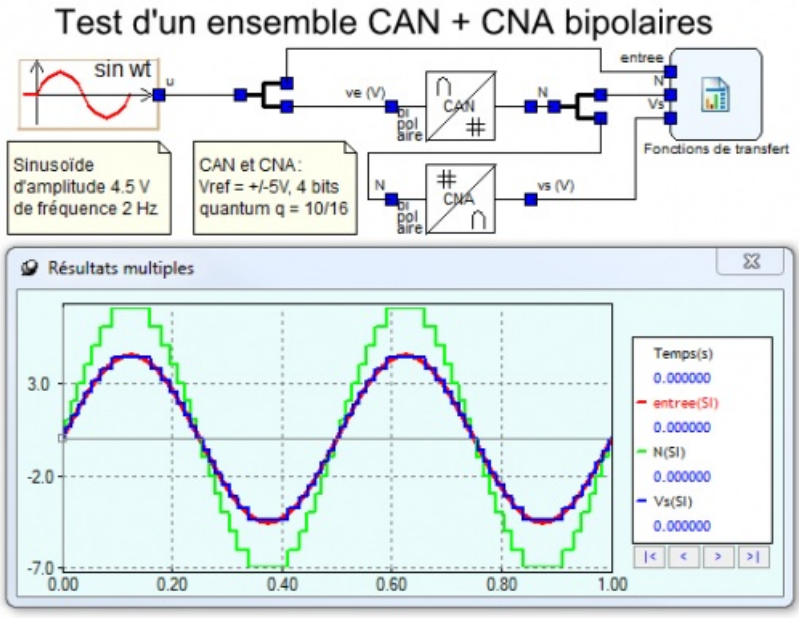
Paramètre	Valeur
Vref	5
NbBits	8
Nmax	2^(NbBits-1)-1
Nmin	-(Nmax+1)
Quantum	2*Vref/(2^NbBits)

Relation	Condition
vs=Quantum*clamp(round(N),Nmin,Nmax)	

Informations

Le circuit CNA (DAC) bipolaire est totalement paramétrable.
1) La valeur de Vref correspond à la tension d'alimentation du CNA réel
2) Le nombre de bits dépend du composant réel (en général 10 bits ou 12 bits)

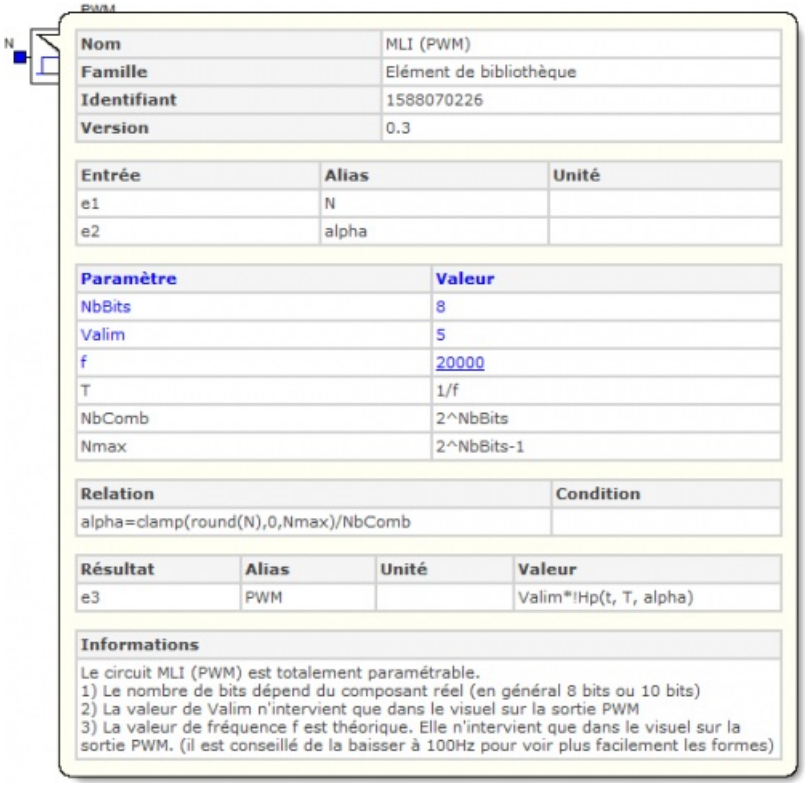
Le schéma de test ci-dessous et les courbes obtenues montrent que la forme de la tension vs, en bleu, entoure parfaitement la tension ve, en rouge. La fréquence est suffisamment basse (*ici 2 Hz, représentative, par exemple, d'une évolution lente d'une tension issue d'un capteur (température, pression...)*) pour que le signal soit suffisamment échantillonné (*critère de SHANNON largement respecté*).
Les 2 modèles de composants (*CAN et CNA*) sont configurés de la même manière : Vref = 5 V, NbBits = 4
L'erreur liée aux marches sera diminuée en augmentant le nombre de bits (*généralement de 10 à 12 bits*)



● 3- Création d'un modèle paramétrable de MLI (PWM) :

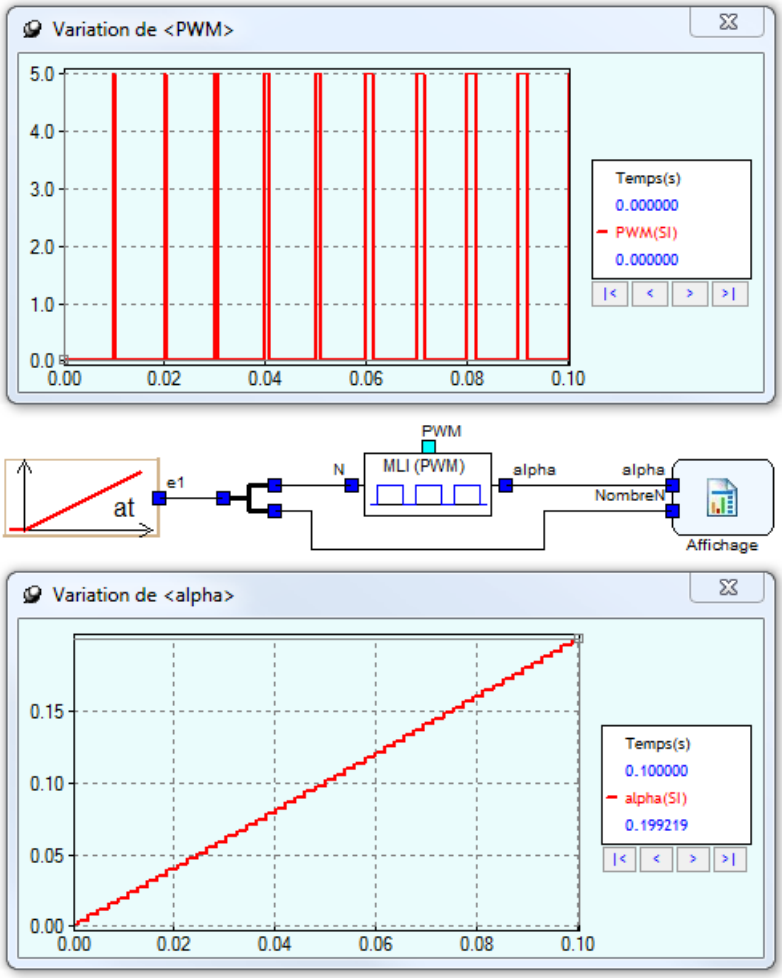
▸ Le rôle de ce composant est multiple :

- Fournir un **signal alpha directement utilisable par le composant hacheur**,
- Montrer l'**allure du signal réel** sur la sortie résultat PWM,
- Permettre de rendre le **script python, utilisé dans Sinusphy (*thème du prochain article*), très similaire à la version qui sera implantée dans le micro-contrôleur cible**. Ainsi, dans les 2 cas, la valeur numérique sera convertie par un composant spécifique en une valeur comprise entre 0 et 100%.



▸ Sur les 3 paramètres, un seul est vraiment nécessaire. En effet, le composant n'a réellement besoin que du **nombre de bits déterminant sa résolution**, les 2 autres paramètres ne servant qu'à configurer le signal PWM disponible sur la sortie résultat (*tension d'alimentation et fréquence*).

▸ Le schéma de test ci-dessous, permet de valider le composant et de montrer qu'il est possible de suivre l'évolution de la forme du signal PWM correspondant à la sortie alpha.



● 4- Création d'un modèle paramétrable de potentiomètre de recopie :

▸ Le potentiomètre de recopie est un capteur de position angulaire simple à utiliser. IL existe classiquement en monotour (*3/4 de tour en général soit environ 270 degrés de course*) et également en multitours (*généralement 10 tours*)

▸ Placé en fin de chaîne d'énergie, le potentiomètre de recopie permet de connaître l'angle absolu obtenu (*angle du bras de robot, angle de direction...*) ou une longueur parcourue (*vérin électrique par exemple*)

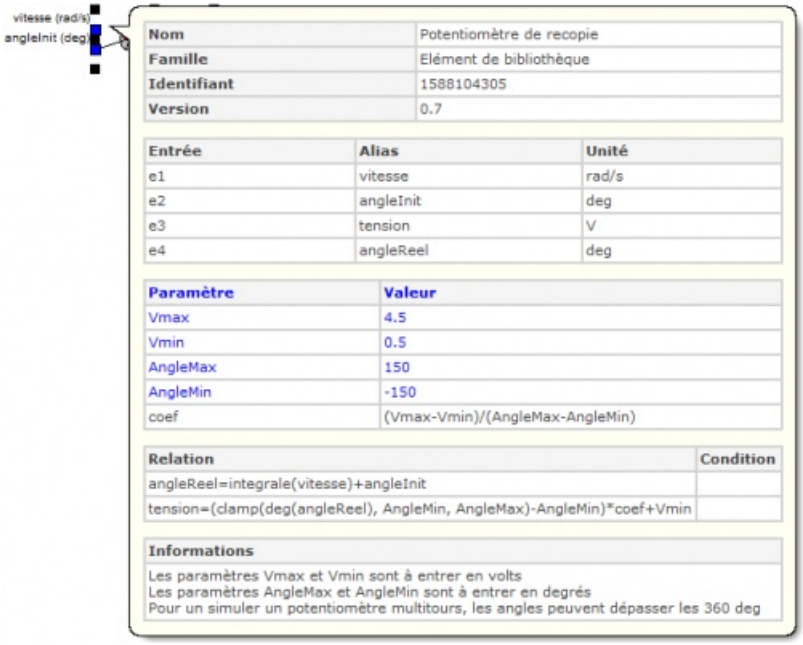
▸ Description du modèle :

- Pour adapter le modèle à tous types de d'utilisation, il possède **une entrée vitesse de rotation** (*reliée à la sortie vitesse de rotation d'un réducteur par exemple*) et **une entrée angle** (*reliée à une maquette méca3D par exemple, ou pour imposer une valeur d'angle initiale*)
- ⚠ **Les valeurs réelles des entrées angles sont en radians** (*pas en degrés !*)
- L'intérêt du modèle réside aussi dans son **paramétrage intuitif** (*à partir des mesures des angles et des tensions sur la maquette réelle*) et sa **polyvalence** (*entrée vitesse et/ou angle*)

▸ C'est l'occasion de soulever le problème lié à la confusion classique que l'on fait avec les conversions d'unités dans Sinusphy, lorsque l'on crée un modèle :

- Ne pas mettre d'unité pour une entrée permet de récupérer la valeur brute,
- Mettre une unité, permet de "voir" la valeur réelle autrement (*exemple : en tr/min au lieu de rad/s*),
- La valeur interne liée à l'entrée ne changera pas, il faut donc parfois la convertir en interne pour certains usages (*ce sera le cas ici*),

- ⚠ Les paramètres ne sont pas liés à une unité particulière, il faut documenter leur usage.

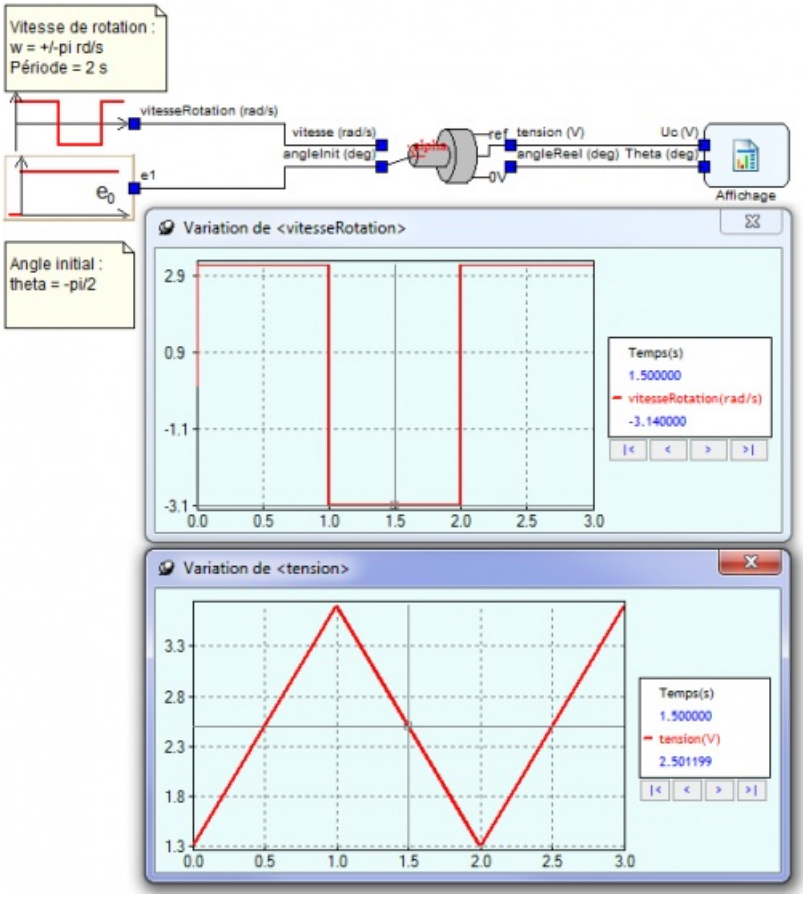


L'étude de ce modèle va permettre d'approfondir ces points :

- Le fait de noter **angleReel = integrale(vitesse) + angleInit** signifie que les angles sont calculés en radians car la vitesse de rotation est forcément en rad/s (SI),
- A noter que la fonction integrale() n'est pas documentée dans l'aide de Sinusphy bien qu'elle soit utilisée dans plusieurs modèles de la bibliothèque de Sinusphy,
- L'unité deg imposée pour les entrées angleInit et angleReel permet, à l'utilisateur, d'avoir une vision en degrés s'il visualise directement ces signaux à la sortie du composant,
- A noter que le composant "sortie", utilisé pour l'affichage, définit, ici, également, Theta en degrés. Si on oublie de le faire, on visualisera alors l'angle en radians (*CQFD*)
- L'équation de la tension fait appel à la valeur de l'angle réel (*en radians*) et la traduit, en interne, en degrés avant de l'utiliser avec les paramètres qui, eux, ont été saisis, par choix, en degrés.
  tension=(clamp(deg(angleReel), AngleMin, AngleMax)-AngleMin)*coef+Vmin

Le schéma de test ci-contre montre que l'état du potentiomètre de recopie peut dépendre directement de la grandeur vitesse et/ou d'une valeur d'angle. Il fournit l'angle réel évalué et la tension correspondante.

On voit bien l'influence de l'angle initial (*ici -pi/2 => tension initiale = 1,3 V*) et de la variation de l'angle simulée grâce à la fonction intégrale().



5- Documents ressources liés à cet article :

Le fichier zip associé contient les copies d'écran en qualité d'origine, les fichiers de simulation et les modèles de composants.

ressourcesarticlesinusphyn5 (Zip de 64 ko)

Académie de Poitiers
Avertissement : ce document est la reprise au format pdf d'un article proposé sur l'espace pédagogique de l'académie de Poitiers.
Il ne peut en aucun cas être proposé au téléchargement ou à la consultation depuis un autre site.