2

Signification des composants graphiques

Dans le chapitre Prise en main, vous avez construit des schémas en assemblant des blocs et des liens. Ces schémas ne comportaient, à l'exception des traceurs de courbes, que des blocs en temps continu, si bien qu'une description sommaire de leur fonction ainsi que le flot d'informations de bloc en bloc indiqué par les flèches des ports suffisaient à appréhender la signification du diagramme et à comprendre les résultats des simulations, visualisés par les traceurs de courbes. Pour des schémas plus complexes, faisant en particulier intervenir des systèmes en temps discret ou des exécutions conditionnées, il est indispensable de bien comprendre ce qui se passe au niveau des blocs et des schémas pour que ceux-ci réalisent effectivement ce que vous attendez d'eux.

Ce chapitre a pour objectif de vous expliquer la signification des composants graphiques de Xcos pour que vous soyez à même de les utiliser en connaissance de cause.

2.1. Compréhension des blocs

Fonctions des blocs

Jusqu'à maintenant, nous avons manipulé des blocs réalisant des fonctions simples telles que la génération de signaux sinusoïdaux ou constants, l'intégrateur, le sommateur ou le traceur de courbes. En réalité, un bloc Xcos peut représenter un système dynamique complexe et inclure tout ou partie des principales fonctions suivantes :

- des fonctions calculant la sortie du bloc, de la forme : $y = F(t, x, z, u)$, comme par exemple $y = A\sin(\omega\, t + \phi)$ dans le bloc Générateur de signaux sinusoïdaux ;
- des équations différentielles de la forme : $\frac{\mathrm{d}\, x(t)}{\mathrm{d}\, t} = C(t, x, z, u)$ évoluant en fonction du temps, comme celle que l'on a vue dans le bloc Intégrateur ou encore des équations algébro-différentielles de la forme : $H(t, x, \frac{\mathrm{d}\, x(t)}{\mathrm{d}\, t}, z, u) = 0$;
- des équations de récurrence sur l'état discret : $z^+ = D_z(t, x, z^-, u)$ ou des équations de *saut* de l'état continu : $x^+ = D_x(t, x^-, z, u)$ évoluant en fonction d'événements

d'activation discrets en entrée et dépendant du ou des ports d'activation ayant reçu l'événement. Les notations x^- et x^+ représentent les valeurs de x juste avant et juste après la prise en compte de l'activation ;

- des fonctions définissant des surfaces critiques $G(t, x, z, u) = 0$. Il s'agit de surfaces frontières entre modèles dont on souhaite tester le franchissement, par exemple la surface du sol dans le cadre de la modélisation d'une balle rebondissante ou le passage par zéro de l'entrée du bloc calculant la valeur absolue ;
- des fonctions programmant des signaux d'activations en réaction à une activation d'entrée : $\Delta t = E(t, x, z, u)$ où $t + \Delta t$ représente la date du ou des événement(s) futur(s) à programmer ;

où t représente le temps, u les entrées du bloc, y ses sorties, x les états continus, z les états discrets.

Lors de la simulation, ces fonctions seront exécutées à tour de rôle selon un ordre établi par le séquenceur.

Les registres d'un bloc

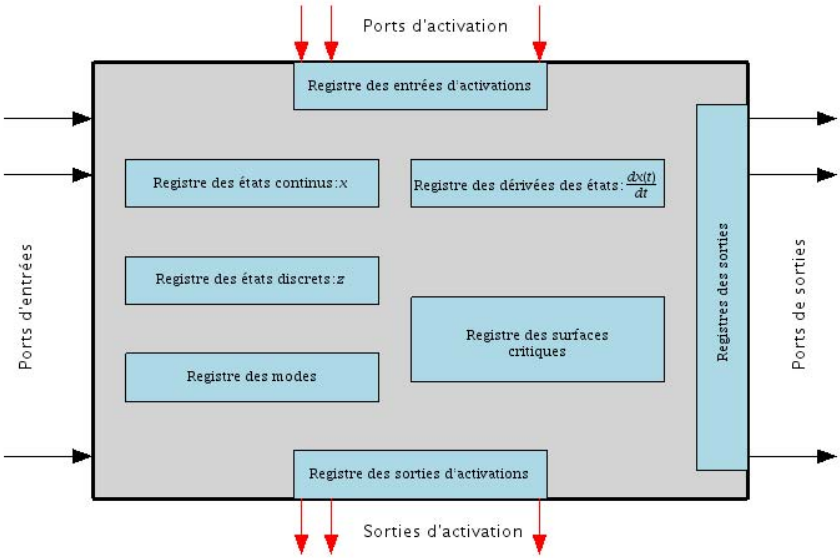
Les registres sont les zones de données attachées à chaque instance de chaque bloc.

Lorsque le bloc est activé le registre des entrées d'activation permet de connaître quel port a été activé (en cas d'activations simultanées, le registre des entrées d'activation définit quels sont les ports qui ont été activés simultanément).

Le registre des modes permet de mémoriser le modèle courant dans le cas d'un bloc pouvant supporter des changements de modèles en cours de simulation, ce registre est souvent associé à la gestion des surfaces critiques pour gérer les discontinuités. Le bloc SIGNUM  qui réalise $y = 1$ si u est positif et $y = -1$ si u est négatif en est un exemple.

Un bloc donné peut ne contenir que certains de ces registres. Ainsi le bloc GENSIN_f  ne contient que le registre de sortie. Le bloc INTEGRAL_m  contient un registre d'état continu, un registre d'entrée et un registre de sortie.

Figure 2.1 : Les différents registres d'un bloc



Rémanence des variables

Les valeurs contenues dans les registres d'états et de sorties d'un bloc sont supposées être toujours définies et sont maintenues constantes entre deux activations.

Les états sont initialisés grâce aux valeurs initiales spécifiées par l'intermédiaire du dialogue associé à chacun des blocs. Les valeurs contenues dans les registres de sorties sont initialisées au tout début de la simulation en fonction du temps et des états initiaux. Une procédure de point fixe permet de propager ces valeurs à travers les blocs ayant des transferts directs entrée/sortie en itérant jusqu'à ce que les valeurs dans les registres de sorties n'évoluent plus.

Paramètres

Hormis l'existence ou non d'états continus et/ou d'états discrets dans le bloc, deux autres de ses propriétés sont essentielles du point de vue de la sémantique :

- la dépendance directe entrée/sortie, qui indique si le bloc a un transfert direct entre l'entrée et la sortie comme, par exemple, le bloc Gain qui réalise $y = K * u$;
- la dépendance continue au temps, qui indique que la sortie du bloc dépend continûment du temps comme, par exemple, le bloc Générateur de sinusoïdes qui réalise $y = a * \sin(\omega * t + \phi)$.

Voir Section 2.3, Gestion des composantes temps continu pour plus de détails sur l'usage de ces paramètres.

2.2. Interprétation des diagrammes

Bien que les schémas Xcos semblent représenter un fonctionnement en flot de données, l'interprétation des diagrammes est en fait basée sur la notion d'*événement d'activation* : la simulation est pilotée par les activations des blocs et les contraintes de causalité indiquées par le sens des flèches sur les ports. Ces activations, explicites ou implicites, notifient l'exécution de chacun des blocs.

Causalité/acausalité

Les schéma que nous avons construits jusqu'alors où les liens dénotent des flots d'information orientés d'un port de sortie vers un port d'entrée correspondent à des modélisations causales. L'approche causale ou *orientée* permet de construire explicitement le modèle global par assemblage des *affectations* calculant les valeurs des registres qui sont réalisées par les fonctions de simulation de chacun des blocs. Par exemple, le système formé du bloc Générateur de sinusoïdes, dont l'affectation calculant la sortie est $y = a * \sin(\omega * t + \phi)$, connecté au bloc Gain, qui réalise $y = K * u$, implémente l'affectation $y = K * (a * \sin(\omega * t + \phi))$. Les affectations présentes dans les blocs sont triées suivant un ordre basé sur le fait qu'une affectation fournissant une entrée doit se trouver avant les affectations consommant cette entrée. Nous verrons plus précisément comment cela s'effectue aux Section 2.3, Gestion des composantes temps continu et Section 2.4, Gestion des composantes discrètes.

Nous verrons dans le chapitre à venir *La modélisation au niveau système* qu'il est aussi possible de modéliser des systèmes par des composants (blocs) acausaux. Les connections entre ces blocs ne représentant plus alors des flots d'information mais des contraintes entre certaines variables des blocs connectés.

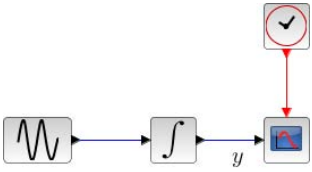
Xcos permet ainsi de combiner l'approche causale et acausale dans un même diagramme. Les blocs acausaux sont alors décrits par des *équations* faisant intervenir les *variables* portées par les ports (un bloc Gain pouvant, par exemple, être décrit par

l'équation $e1 - K * e2 = 0$ où $e1$ et $e2$ sont les variables portées par les ports du bloc). Dans ce cas, un prétraitement permet de convertir les composants acausaux du schéma en un seul bloc compatible avec l'approche causale.

Signaux d'activation et conditionnement

Dans le schéma ci-dessous le lien rouge représente l'activation explicite du traceur de courbes par le bloc Horloge. Les activations transmises par ce lien sont des activations discrètes (des tops). Les blocs en temps continus tels que le générateur de sinusoïdes et le bloc Intégrale sont eux aussi pilotés par un signal d'activation, mais il s'agit là d'activation continue (des intervalles de temps). Ces signaux d'activation continue ne sont pas représentés sur le schéma pour ne pas en alourdir la lecture.

Figure 2.2 : *Un schéma simple (`scilab-xcos-screen4.zcos`)*

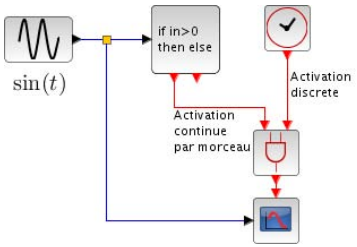


L'utilisateur attend de ce schéma qu'il affiche les valeurs de l'intégrale de la sinusoïde en entrée à chaque instant d'observation correspondant aux dates des événements produits par l'horloge.

Il n'existe pas d'implémentation qui permette dans un ordinateur de faire évoluer continûment un signal en fonction du temps. Mais il est possible d'agencer les calculs de telle sorte que les observations soient conformes aux résultats attendus. Par exemple, pour le schéma ci-dessus, à l'occurrence d'un top de l'horloge à la date t_k, Xcos va appeler un intégrateur d'équations différentielles pour évaluer numériquement la valeur de y à la date t_{k+1} correspondant à la date du top suivant de l'horloge. C'est cet intégrateur d'équation différentielle qui fait avancer le temps à l'intérieur de l'intervalle d'activation $[t_k, t_{k+1}]$.

Plus généralement, les signaux d'activation de Xcos peuvent être formés par l'union d'intervalles de temps et de points isolés.

Figure 2.3 : *Activation continue (`scilab-xcos-activation_continue.zcos`)*



La Figure 2.3 montre sur un exemple simple comment il est possible de conditionner l'exécution de composants discrets par un signal d'activation continu par morceau. Le bloc If-then-else IFTHEL_f est ici utilisé en mode continu, il génère sur son port *then* une activation continue par morceau sur tous les intervalles de temps ou $\sin(t)$ est positif.

Le bloc ANDBLK de la palette GESTION D'ÉVÉNEMENTS génère un signal d'activation lorsque ses deux ports d'entrée sont activés simultanément.

Le bloc CSCOPE n'est donc activé que pour les tops d'horloge tels que $\sin(t)$ est positif. Le bloc CSCOPE a été paramétré pour afficher d'une croix la valeur du signal entrant chaque fois qui est activé.

La simulation de ce schéma produit le graphique de la Figure 2.4.

Figure 2.4 : *Activation continue contrôlée*