Sciences de l'Ingénieur
TP n°2 - Réponse temporelle des SLCI

Nous avons vu, à la première séance de TP, comment saisir un schéma-bloc sous Simulink et déterminer la fonction de transfert équivalente sous Matlab. Nous allons voir aujourd'hui comment les exploiter pour déterminer la réponse temporelle d'un SLCI. Les deux méthodes présentent leurs avantages et leurs inconvénients :

- la réponse temporelle est plus facile à déterminer sous Simulink, mais pour des types de consignes limités (échelon, rampe, sinusoïde), et elle est plus difficile à exploiter,
- la réponse temporelle est plus difficile à déterminer sous Matlab (puisqu'il faut utiliser diverses commandes) mais, une fois tracée, il est possible d'en tirer des informations intéressantes telles que le temps de réponse à 5%. Elle peut en outre être déterminée pour, en théorie, n'importe quelle consigne.

1 Détermination de la réponse temporelle sous Simulink

Il est possible de déterminer la réponse indicielle, en rampe, ou harmonique (à une consigne sinusoïdale) d'un SLCI à l'aide de Simulink. En revanche, comme nous l'avons remarqué au TP n°1, **il est impossible de déterminer la réponse impulsionnelle** puisque l'impulsion de Dirac n'est pas physiquement réalisable et donc non modélisable sous Simulink.

Il vous suffit de placer en entrée du système le composant associé à la réponse souhaitée : un échelon pour une réponse indicielle, une rampe pour une réponse en rampe, une consigne sinusoïdale pour une réponse harmonique. En utilisant en sortie un multiplexeur et un scope, il vous est ensuite possible de faire apparaître sur un même écran la consigne et la réponse du système à cette consigne. Pour cela, il suffit de cliquer sur le bouton  (run) pour lancer la simulation, puis de double-cliquer sur le scope. La durée de la simulation peut être modifiée en cliquant sur le bouton  . Cependant, une fois ces courbes tracées, il est impossible de déterminer avec précision les coordonnées d'un point de la courbe, ou même le temps de réponse du système. Il est juste possible de zoomer ou de dézoomer.

2 Détermination de la réponse temporelle sous Matlab

Matlab offre de plus grandes possibilités en matière de réponses temporelles, bien que cela nécessite la connaissance de la fonction de transfert. Un préliminaire à toute détermination de réponse temporelle est donc la détermination sous Matlab de la fonction de transfert équivalente au schéma-bloc étudié.

Une fois cette fonction de transfert déterminée, il est possible de déterminer la réponse à une consigne :

- impulsionnelle avec la commande impulse,
- indicielle avec la commande step,
- quelconque avec la commande lsim.

Les deux premières commandes sont les plus simples. Une fois le système défini sous Matlab, par exemple sous le nom sys, il suffit d'utiliser les commandes impulse(sys) ou step(sys) pour obtenir la réponse impulsionnelle ou indicielle du système (voire impulse(sys1,...,sysn) ou step(sys1,...,sysn) pour obtenir la réponse impulsionnelle ou indicielle de plusieurs systèmes dans une même fenêtre). C'est cependant Matlab qui choisit automatiquement la durée de la simulation, même si cette dernière pourra être modifiée plus tard.

Si vous voulez modifier l'intervalle de temps au cours duquel la simulation a lieu, vous pouvez le faire en utilisant les commandes impulse(sys,t) ou step(sys,t). **Attention**, cependant : la variable t qui, pour nous, correspond au temps, est pour Matlab une variable comme une autre, à laquelle il faut donc imposer une valeur pour que Matlab comprenne ces commandes. La variable t va donc être définie comme un vecteur contenant les différentes dates auxquelles les valeurs de la réponse impulsionnelle ou de la réponse indicielle seront calculées. Si les dates sont également espacées, il n'est pas nécessaire de toutes les lister puisque la commande matlab $[a : p : b]$ (ou, tout simplement, $a : p : b$) permet de construire automatiquement le vecteur $[a, a + p, a + 2p, \ldots, a + kp]$ avec k la plus grande valeur entière telle que $a + kp \leqslant b$. Ainsi, par exemple, si $t = [0 : 0.01 : 10]$, les valeurs de la réponse temporelle seront calculées à des dates comprises entre 0 et 10 et séparées de 0,01, ce qui correspond à 1001 valeurs. Vous pouvez tester par vous-même et observer l'impact du nombre de valeurs sur la forme de la réponse obtenue. Le pas p doit-il être petit ou grand (et par rapport à quoi ?) pour que la réponse tracée par Matlab corresponde fidèlement à la vraie réponse temporelle ?

La commande lsim utilise le même principe : la commande lsim(sys,u(t),t) demande à Matlab de tracer la réponse du système sys à une consigne qui est une fonction de t dont l'expression est u(t) (là aussi, on peut également utiliser la commande lsim(sys1,...,sysn,u(t),t) pour obtenir la réponse à la consigne u(t) de plusieurs systèmes dans une même fenêtre). Le vecteur t doit être défini comme précédemment. **Attention**, cependant. En fait, la commande lsim(sys,u(t),t) demande en fait à Matlab de tracer un graphe avec en abscisse les valeurs de t et en ordonnée les valeurs de la réponse à la consigne u(t) : t étant un vecteur permettant d'indiquer à Matlab à quelles dates la valeur de la réponse doit être calculée, la valeur de u doit être connue aux mêmes dates, et u(t) doit donc être un vecteur de même dimension que t. Néanmoins, Matlab est très bien conçu, ce qui fait que si le vecteur t est défini, l'application d'une fonction à t fournira un vecteur de même dimension, ce qui fait que l'écriture transparente lsim(sys,sin(t),t), par exemple, ne posera aucun problème pour Matlab. Cependant, on pourrait penser que l'on pourrait obtenir la réponse indicielle d'un système

en utilisant la commande lsim(sys,1,t), et ce serait alors faux du fait que 1 est un scalaire et non un vecteur. Il faudrait donc définir un vecteur rempli de 1, ce qui peut être fait en utilisant la commande ones(1,n), où n est la dimension du vecteur t, ou bien « tromper » Matlab en lui faisant calculer un vecteur de dimension n et toujours égal à 1, par exemple en lui fournissant une consigne du type $1 + t - t$ ou $1 + 0. * t$.

Quelle que soit la commande utilisée, les courbes tracées sous Matlab peuvent ensuite être exploitées de diverses manières :

- la valeur du dépassement maximal peut être obtenue en faisant un clic droit dans la fenêtre de la courbe puis en cliquant sur Characteristics puis sur Peak Response,
- une grille peut être rajoutée pour faciliter la lecture des coordonnées en faisant un clic droit dans la fenêtre de la courbe puis en cliquant sur Grid (ou en tapant « grid » dans la fenêtre de commande de Matlab),
- les intervalles de tracé de la courbe en abscisse et en ordonnée peuvent être modifiés en faisant un clic droit dans la fenêtre de la courbe puis en cliquant sur Properties puis sur Limits (en tapant sur la touche Entrée pour valider les changements),
- enfin, un curseur peut être ajouté en double-cliquant sur la courbe, puis ce curseur peut être déplacé à l'aide de la souris.

De plus, si ce sont les commandes impulse ou step qui sont utilisées, il est possible d'obtenir le temps de réponse du système en faisant un clic droit dans la fenêtre de la courbe puis en cliquant sur Characteristics puis sur Settling Time. **Attention**, cependant. Le temps de réponse déterminé par Matlab est par défaut le temps de réponse à 2%. Il est possible d'obtenir le temps de réponse à 5% en faisant un clic droit dans la fenêtre de la courbe puis en cliquant sur Properties puis sur Options, et en sélectionnant « Show settling time within 5% », puis en tapant sur la touche Entrée. Si c'est la commande step qui est utilisée, il est également possible d'obtenir la valeur du dépassement **relatif** maximal avec Peak Response.

3 Exercices d'entraînement

3.1 Réponses temporelles d'un système d'ordre 1

Tracer, en utilisant Matlab et/ou Simulink, les réponses temporelle (impulsionnelle, indicielle, en rampe) d'un système du 1er ordre, et observer l'influence du gain et de la constante de temps sur les réponses temporelles.

3.2 Réponses temporelles d'un système d'ordre 2

Tracer, en utilisant Matlab **et** Simulink, la réponse indicielle d'un système du 2^{e} ordre pour des valeurs du facteur d'amortissement de 0, 0.1, 0.2, ..., 2 **sur le même graphe**. Faire de même avec les réponses impulsionnelle et en rampe.

La démarche à mettre en œuvre ici sous Matlab est un peu plus compliquée que celle que nous avons vue précédemment, du fait qu'il nous faut tracer plusieurs courbes sur un même graphe. Il

nous faut tout d'abord définir un vecteur (par exemple xi) contenant toutes les valeurs du facteur d'amortissement. Il nous faut ensuite programmer une boucle for sous Matlab qui va nous permettre de faire varier le facteur d'amortissement et de « mémoriser » la réponse temporelle du système pour chaque valeur considérée. Pour ce faire, si la variable de la boucle for est par exemple notée i, xi(i) fournira la i-ème coordonnée du vecteur xi (et donc la i-ème valeur du facteur d'amortissement). Il nous faut ensuite définir une fonction de transfert sys fonction de xi(i), puis obtenir sa réponse temporelle. La difficulté réside dans la mémorisation des valeurs des différentes réponses temporelles. L'astuce consiste à remplir une matrice à l'aide de ces valeurs en utilisant, par exemple, la commande « y(:,i)=step(sys,t) », commande qui dira à Matlab de remplir la i-ème colonne de la matrice y avec les valeurs de la réponse indicielle aux différentes dates renseignées dans le vecteur t. L'ensemble des courbes peut enfin être tracé à l'aide de la commande « plot(t,y) ».

3.3 Diagramme de Bode d'un système d'ordre 2 avec résonance

Tracer, en utilisant Matlab et/ou Simulink, le diagramme de Bode d'un système du 2^{e} ordre de facteur d'amortissement $\xi = 0,5$ **uniquement à l'aide de sa réponse à une consigne sinusoïdale**.

3.4 Précision et robustesse de systèmes d'ordre 1 et 2

Retrouver, en utilisant Matlab et/ou Simulink, la valeur de l'erreur en régime permanent de systèmes d'ordre 1 et 2 pour une consigne en échelon. Retrouver ensuite la relation entre le nombre d'intégrateurs dans la boucle ouverte et la précision/robustesse de ces systèmes en boucle fermée avec retour unitaire pour une consigne/perturbation donnée.