



Les 3 parties du projet

Traitement du signal

Electronique

Assembleur

Projet : Traitement du signal

Savoir interpréter et utiliser une DFT :

Expliquer l'effet du fenêtrage rectangulaire

Expliquer l'effet de l'échantillonnage temporel sur le spectre

Documents associés :



Fiche_Log_SIMILNK_V4.pdf



FicheTheorie_TFD_v4.pdf

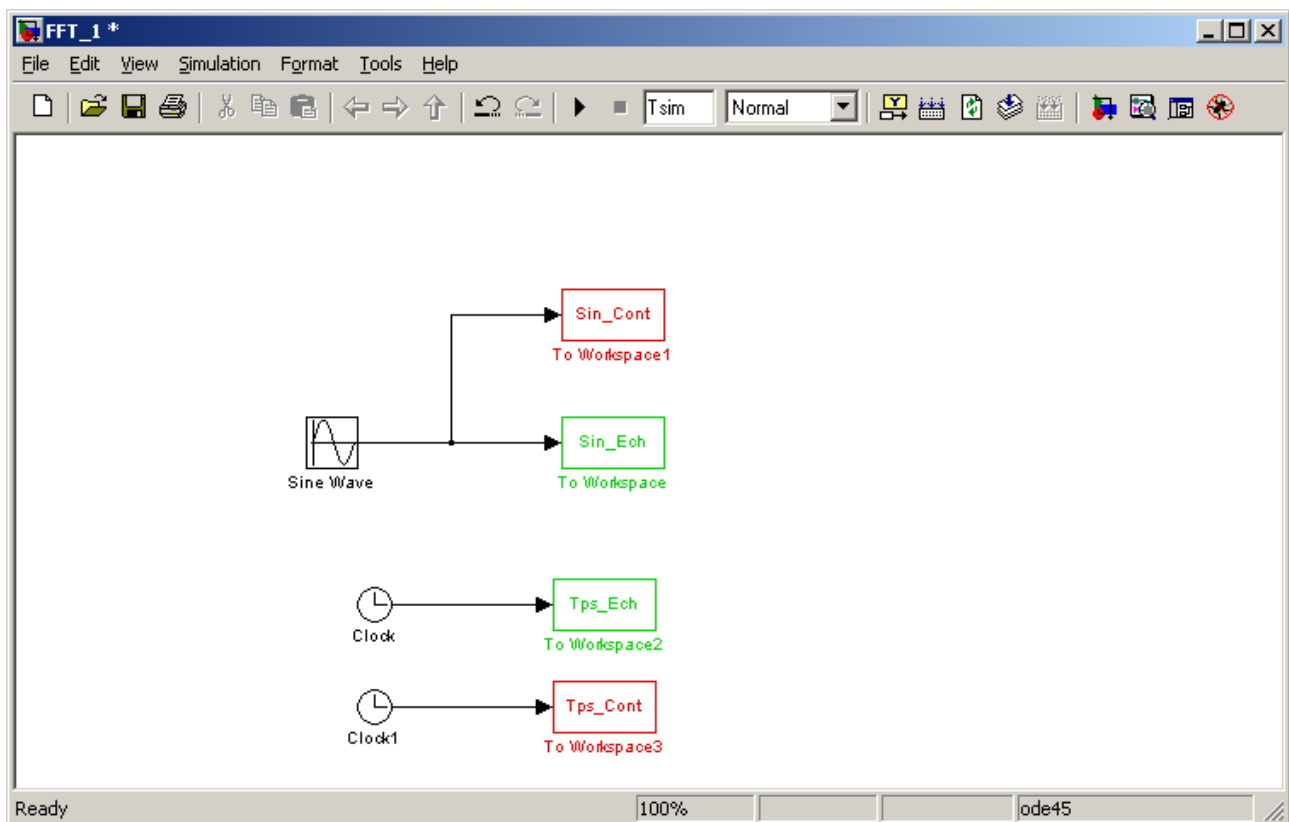
1. Prise en main Matlab et Simulink

1. Lisez la *fiche logiciel* La simulation avec SIMULINK



2. Edition sous SIMULINK

Lancez Matlab, puis tapez la commande *simulink*. Editez le schéma suivant sous Simulink puis en faire la sauvegarde :



On distingue dans ce schéma :

- une **source sinusoïdale**. Elle devra être paramétrée en amplitude et fréquence.
- Deux sources de types **horloge**. Elles génèrent, à partir de l'avancement du simulateur des tables de données (vecteurs) qui contiennent des dates (comprises entre 0 et *Tsim*) **NB**: *Tsim* est une variable générée dans MATLAB, qu'il faut placer dans SIMULINK pour spécifier le temps de simulation.
- 4 sondes (sink) de type **ToWorkspace** qui permettent de renvoyer des tableaux de données (vecteurs) vers MATLAB. Ces données seront donc traitées sous MATLAB. **NB**: Pour chaque sonde, on

prendra garde à bien régler son *TS* (*Sampling Time*) et on précisera la forme des variables : *Save format* : *Array*.

- Les sondes vertes (*Sin_Ech* et *Sin_Ech*) ont un *TS* égal à *Te*. Cela force le simulateur à adapter son pas de calcul à *Te* voire moins. Les points collectés par chacune des sondes seront espacés de *Te*. Ces deux sondes permettront de traiter sous MATLAB les signaux échantillonnés (FFT).
- Les sondes rouges (*Sin_Cont* et *Tps_Cont*) ont un *TS* égal à *Te/100*. Le simulateur sera donc obligé d'affiner encore plus son pas de calcul à mieux que *Te/100*. En toute rigueur, ces signaux ne sont pas continus (ils sont bien sûr discrets). Mais le pas est tellement fin, qu'on pourra considérer ces signaux continus, si on les compare à ceux issus des sondes vertes (discretisés à *Te*). Les sondes considérées continues, permettront de faire des comparaisons discret – continu.

3. Edition d'un script .m

L'exploitation des données se fera par l'intermédiaire d'un script dont l'extension est *.m*. Voici un exemple (rien à voir avec le TP actuel) :

```

10 % Cons_I(p)--> +( )--> |C(p)|--> |K=KMLI.KH |-->|1/R / (1+L.p/R)|-->I(p)
11 %
12 %
13 %
14 %% Grandeurs à saisir
15 % Il est demandé de rentrer les grandeurs suivantes:
16 % R (Ohm), L (H) : résistance et inductance de la Mcc
17 R=1;
18 L=1e-3;
19 % Sensibilité du capteur de courant (V/A)
20 SI=1.65/10; % 10A donne 1.65V
21 % Tension du hacheur (V)
22 E=24; % batterie de 24V
23 % Fréquence de transition souhaitée pour la FTBO (Hz)
24 FT=500;
25
26
27 %% Fonction de transfert du moteur
28 % I(p) = (1/R) / (1+ Tau_Elec.p) * (V(p) - E(p)) = F_Moteur(p)
29
30 Tau_Elec = L/R;
31 F_Moteur_Num=1/R;
32 F_Moteur_Den=[Tau_Elec 1];
33 F_Moteur=tf(F_Moteur_Num,F_Moteur_Den);
34
35 bode(F_Moteur);
36 figure;
37 bode(Cor_PI);
38 hold on
39
40
41
42 %% Synthèse du PI (mphi de 90°, on ne compte pas le filtre de retour)
43 % ConsI est la consigne de courant sismée exprimée en V. Voir SI pour la

```

Le fichier va permettre :

- de régler les paramètres du design saisis sous Simulink
- de lancer Simulink

- de visualiser les tableaux provenant de SIMULINK
- opérer des calculs sur ces données (notamment prendre la *fft*)

On demande : (chercher dans l'aide comment utiliser les commandes inconnues. Par exemple, tapez dans la fenêtre *Command Window*, *help sim*)

- De créer une variable **T** et de lui affecter la valeur 1 : C'est la fenêtre d'analyse, choisie à 1s.

NB: dans la syntaxe MATLAB, la déclaration (donc la création) se fait en même temps que l'affectation. Donc dans ce cas on écrira tout simplement

T=1;

Si le ';' n'est pas écrit, à l'exécution, l'opération se fait en écho dans la fenêtre "*Command Window*" de MATLAB.

- Créer le nombre de points pour la FFT : **M**=32
- Créer la période d'échantillonnage **Te** à partir de T et M
- Créer le temps de simulation **Tsim** qui sera égal à T-Te (on enlève Te, sinon le simulateur inclut le dernier point et donne 33 points au lieu de 32).
- Créer la fréquence **Fsin** de la sinusoïde. On lui donnera la valeur 3 (3Hz).
- Lancer la simulation SIMULINK avec la commande *sim*.
- Procéder à la transformée de Fourier rapide (*fft*) du signal sinusoïdal échantillonné.
- Afficher avec la commande *plot* :
 - la sinusoïde considérée continue
 - la sinusoïde discrète (vous ferez un affichage uniquement des points)
 - la *fft* (vous ferez un affichage uniquement des points)

NB: pour les affichages temporels, vous prendrez le vecteur *Tps_Ech* ou *Tps_Cont* comme abscisse dans les affichages de type *plot*.

Dernière chose à faire sous SIMULINK : placer toutes les variables (Tsim, Te, Te/100, Fsin...).

4. Faire un commentaire sur le résultat pour Fsin=3Hz.
5. Procéder à la même chose pour Fsin= 15Hz. Commenter la FFT et comparer les deux signaux temporels (continu et discret).

Pour les questions qui suivent, lire la fiche théorique La transformée de Fourier Discrète **chapitre 3 inclus qui concerne le fenêtrage.**



jusqu'au

6. Choisir $F_{\sin} = 5\text{Hz}$, puis $5,5\text{Hz}$. Commenter le résultat. Calculer la valeur du point à 5Hz , 6Hz , 7Hz .
7. Conclure sur les conditions à respecter pour lire exactement sur la FFT la valeur que donnerait une véritable transformée de Fourier.
8. Modifier votre script pour obtenir le spectre exact de la sinusoïde de $5,5\text{Hz}$. Quel paramètre doit-on modifier ?
9. Afin d'obtenir un tracé spectral avec, en abscisse, les valeurs de fréquences qui s'affichent (et non l'indice des points). Aidez vous de la commande *linspace*.

Pour les questions qui suivent, lire la fiche théorique La transformée de Fourier Discrète



en entier

10. Se replacer dans les conditions de départ. Utilisez toujours le vecteur d'abscisses fréquentielles. Choisir $F_{\sin}=16\text{Hz}$. Commenter
11. Refaire un dernier test pour $F_{\sin}=25\text{Hz}$. Commenter

2. Utilisation de la FFT de l'oscilloscope

A faire

3. Projet – traitement de signal

Chacun des 6 pistolets va émettre une onde sinusoïdale pendant un temps maximal de 100ms , à une fréquence bien déterminée. La discrimination de chacun des 6 faisceaux va se faire par une DFT pour chacune des 6 fréquences. Sous MATLAB, on utilisera la FFT (qui peut le plus peu le moins, dans le sens où la FFT calcule les M points fréquentiels, alors que dans le projet, on ne calculera que 6 points fréquentiels, ceux qui nous intéressent).

Les fréquences retenues sont :

$$F_1 = 85 \text{ kHz}$$

$$F_2 = 90 \text{ kHz}$$

$$F_3 = 95 \text{ kHz}$$

$$F_4 = 100 \text{ kHz}$$

$$F_5 = 115 \text{ kHz}$$

$$F_6 = 120 \text{ kHz}$$

1. On vous demande de paramétrer une DFT. On impose une fréquence d'échantillonnage de 320kHz. Choisir T et en déduire M. On impose que chaque signal reçu ne produise aucun parasitage sur les fréquences des autres canaux.
2. Construire un nouveau fichier SIMULINK et un nouveau script. Editer un schéma qui permette la simulation des 6 pistolets qui tirent en même temps (utiliser l'opérateur *Sum* dans SIMULINK) . Vérifier que la FFT est concluante. Vérifier en particulier que chaque pistolet n'influe pas sur la raie spectrale d'un autre pistolet. Chacune des 6 sources ne doit pas perturber le canal d'une autre (on appelle cela l'orthogonalité des signaux). En effet, il serait ennuyeux qu'une erreur de mesure entraîne la validation d'un tir voisin...

3. En réalité, l'émetteur possède une imprécision sur les canaux suivant :

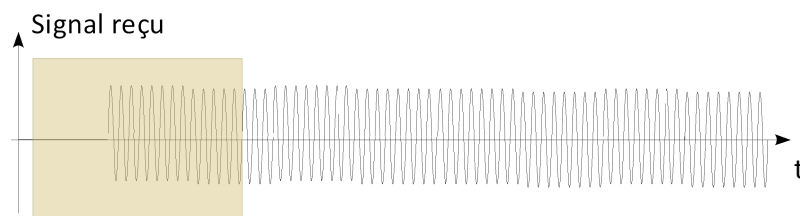
$$F_1 = 85005,9 \text{ Hz}$$

$$F_3 = 94986,8 \text{ Hz}$$

$$F_5 = 115015,9 \text{ Hz}$$

Estimer l'impact de l'erreur de fréquence sur les autres canaux. Passer en échelle log pour bien mesurer la différence.

4. Si pas de chance, il se peut que la fenêtre d'analyse de la DFT soit à cheval sur le début d'une salve :



Analyse FFT "à cheval" sur le front d'arrivée

Faire une FFT dans ce cas. Pour cela, on pourra truquer l'échantillonnage et placer artificiellement la première moitié du bloc à 0 (boucle for). Quelle conséquence sur le jeu ?

!! En réalité, les signaux issus du pistolet ne sont pas sinusoïdaux mais carrés.

5. Déterminer la série de Fourier d'un signal carré de fréquence F_0 .

6. Quel problème cela peut poser vis à vis de la DFT ? Faire un essai pour un **seul** signal à 85kHz mais de forme carrée et observer la DFT. Prendre une source de type *pulse Generator*. Expliquer la FFT obtenue.
7. On propose de placer en amont un filtre. Son rôle est de contourner le problème relevé précédemment. Quel est sa nature ? Que doit-il permettre ?

On propose d'implémenter un filtre dit de Chebychev, dont la fonction de transfert $F(p)$ est:

$$F(p) = \frac{1}{1,7483 \cdot 10^{-23} \cdot p^4 + 7,6663 \cdot 10^{-18} \cdot p^3 + 1,162 \cdot 10^{-11} \cdot p^2 + 3,0332 \cdot 10^{-6} \cdot p + 1}$$

8. Tracer le diagramme Bode à l'aide de la commande *bode*. Vous utiliserez aussi la commande *tf* (création de fonction de transfert). Mesurer dans le pire cas, le gain de filtre pour la première harmonique indésirable lors d'une attaque en signal carré.
9. Insérer le filtre passe-bas dans votre schéma SIMULINK (bloc *Transfer Fcn*). Refaire l'essai de la question 6. Conclusion. Observer le signal en sortie du filtre. Est-il bien sinusoïdal ?