

Projet Programmation

Documentation

Auteurs

Gonzague LENFANT (scriper n°236781) et Eklavya SARAQR (n°236312)

Section Génie Mécanique - Année Propédeutique - EPFL

Plateforme et compilateur employés

OSX et Xcode

Résumé

Le but du projet de programmation est de générer les valeurs d'un signal, le filtrer grâce à un filtre moyenne mobile, puis finalement l'afficher. Ceci est découpé en trois parties, utilisant chacun un des trois logiciels vu en cours, c'est-à-dire, un compilateur de C++, LabVIEW et MatLab.

Fichiers principales présents

- *Projet Programmation, partie Labview.vi* : VI principal à exécuter en premier.
- *Genere_cpp* : exécutable mac lancé par le VI principal, crée la période.
- *main.cpp* : le code source C++, utilisé par l'exécutable pour créer la période.
- *Genere_cpp.xcodeproj* : le code C++ dans le compilateur Xcode.
- *F_période.txt* : fichier créé par l'exécutable contenant les valeurs de la période du signal.
- *Second VI.vi* : VI secondaire lancé par le VI principal qui filtre le signal.
- *F_signal.m* : fichier créé par Labview contenant les valeurs du signal.
- *Options.m* : fichier créé par Labview contenant les options graphiques pour la courbe.
- *MP_LaunchMatlabScript.vi* : VI secondaire lancé pour lancer Matlab et exécuter le script.
- *Projet_programmation_script_matlab.m* : script Matlab permettant l'affichage de la courbe.

Le flow de données

Tout d'abord, il faut sélectionner, dans le *Control Panel* du *Projet Programmation*, partie *Labview.vi*, les options d'affichage graphique et les valeurs souhaitées pour les caractéristiques du signal, comme la forme, le nombre de points, l'amplitude, l'offset, le bruit.

Une fois démarré, celui-ci lance ensuite l'exécutable *Genere_cpp* avec les valeurs du signal choisies comme paramètres dans la console. Celui-ci utilise le code C++ écrit dans le *main.cpp*, qui utilise, à son tour, plusieurs bibliothèques et fonctions pour créer et enregistrer un fichier texte *F_periode.txt* (dans le fichier *Genere_cpp*) contenant toutes les valeurs des points du signal, en fonction d'un intervalle de temps régulier. De plus, des indicateurs ont été mis dans le LabVIEW pour contrôler l'entrée et le chemin de la commande, permettant donc de s'assurer que la commande envoyée dans la console est celle souhaitée. *F_periode.txt* est ensuite lu par le VI principal et en sort un *string* et ensuite un *array* (tableau) contenant aussi les mêmes valeurs de la période. Un booléen qui indique si le fichier *F_periode.txt* existe, et d'autres indicateurs du *output* (retour) du *Genere_cpp* ont aussi été installés.

Le VI principal va ensuite utiliser une boucle *for* pour dupliquer l'*array* des points du signal selon le nombre de fois souhaité par l'utilisateur dans l'option *nombre periodes* pour obtenir un *array* du signal final. Il va ensuite lancer *Second VI.vi* qui va filtrer le signal grâce à un filtre moyenne-mobile et le mettre dans un autre *array*. Selon le choix du booléen *filtre*, un des deux *array* filtré ou non-filtré sera affiché sur le *Front Panel* du VI principale.

Les valeurs des deux *array*s sont ensuite mises ensemble dans un *appended array* (un tableau de deux colonnes), qui sont subséquemment transférées dans un *string*, pour pouvoir créer le fichier *F_signal.m* contenant en deux colonnes les valeurs des points du signal filtré et non-filtré. Le choix de l'affichage de chaque signal, c'est-à-dire, la couleur, la représentation des points, et le style sont sélectionnés par l'utilisateur dans le *Control Panel* du VI principal au départ, et les données sont transférées dans le fichier *Options.m*.

Le VI principal lance ensuite *MP_LaunchMatlabScript.vi* qui va démarrer le script *Projet_programmation_script_matlab.m* pour au final afficher le signal filtré et non-filtré grâce à Matlab, en tenant en compte les fichiers *F_signal.m* et *Options.m*.

Indications aux utilisateurs

- Ne surtout pas supprimer ou déplacer les fichiers impliqués dans le dossier *Sarkar_Lenfant*, car les chemins sont prédéfinis selon leur position par rapport au VI principal.
- Choisir tous les paramètres de départ dans le *Control Panel* du VI principal avant de lancer le programme.

- Si le programme ne fonctionne pas, appuyer sur stop si nécessaire et regarder les indications dans les outputs (retours) du `Genere_cpp` ou Matlab. Les erreurs potentielles seront indiqués dedans.
- En cas d'erreur, vérifier que le fichier `F_période.txt` existe grâce au booléen, qui sera vert si le fichier existe.

Gestion des erreurs

Dans le code C++, toutes les erreurs sont sorties avec `cerr` et sont visibles dans output du `Genere_cpp` dans le LabVIEW. En cas d'erreur, la fonction doit retourner un code d'erreur 1 dans `Return Code`, ainsi que le message d'erreur (écrit auparavant dans le C++) dans `Standard Error`. Il en va de même pour les erreurs de Matlab, qui sont affichées dans les outputs du Matlab.

Remarques

- Tout les accents visibles par l'utilisateur ont été enlevés des trois logiciels car ils se changeaient en d'autres caractères lors des changements entre différents OS.
- Le nom et chemin du fichier contenant l'exécutable `Genere_cpp` est long et compliqué car il était créé automatiquement par Xcode dans les fichiers du système OS X. Nous étions encore des novices de Mac et de la gestion des fichiers créés par Xcode, et nous pensions que tous les fichiers surplus du data étaient nécessaires pour que le compilateur marche.

Nous avons, plus tard, essayé de changer ou supprimer le nom et chemin de ces fichiers mais cela empêchait le VI et l'exécutable de marcher correctement car ils étaient déjà reliés à ce(s) document(s). Après quelques essais sans succès, nous avons décidé de garder les fichiers tels quels. Nous nous excusons pour ce désagrément, même si tout les programmes marchent parfaitement.

- Nous avons eu beaucoup de plaisir à apprendre la programmation et à travailler sur ce projet !