

TD1&TD2 MASTERS

Cahier des charges n°1: COMMANDES & INDICATEURS & SOUS-PROGRAMMES

Ecrire un VI qui calcule le rapport de deux nombres réels x et y

⇒ **Lancer LabVIEW**

⇒ **Ouverture VI vierge** (mettre côte à côte diagramme et face-avant).

Présentation des différentes palettes (commandes, fonctions, outils). Une palette est commune aux deux fenêtres: la **palette d'outils**. Pour éviter de se tromper, LabVIEW ne permet pas d'accéder à la palette de commandes à partir du diagramme et ne permet pas d'accéder à la palette de fonctions à partir de la face-avant.

⇒ **Déposer commandes et indicateur sur face-avant**

Montrer l'utilisation palette outils et l'option caractère A pour renommer une commande en x , l'autre en y et l'autre en s pour l'indicateur. Faire remarquer qu'une icône apparaît sur le diagramme à chaque création d'objets sur face-avant. Faire remarquer que les commandes sont dessinées avec un trait épais alors que les indicateurs sont en trait fin.

⇒ **Déposer opérateur 'diviser' sur diagramme'**.

Utiliser aide en ligne pour connaître action opérateur divisé (pour le 'diviser le fil le plus haut divisé par le fil le plus bas).

⇒ **Réaliser câblage**

⇒ Montrer comment réaliser un angle droit.

⇒ Montrer comment rechercher, à partir de la face-avant une commande sur le diagramme (un clic droit et sélectionner rechercher sur ou double clic directement).

⇒ **EXECUTER LE PROGRAMME ($x=1$ et $y=2$ donc $s=0.5$)**

⇒ Montrer influence, sur le résultat, du type de données (faire utiliser le menu local de l'indicateur afin de transformer la représentation de l'indicateur en entier **I32**). **EXECUTER LE PROGRAMME ($x=1$ et $y=2$ donc $s=0$!)** Montre donc qu'il existe un menu local associé à chaque objet, que LabVIEW utilise un code de couleurs selon le format et que le résultat peut être erroné si le format est mal choisi.

⇒ Construire sous les instructions précédentes une application identique en créant, à partir du diagramme **UNIQUEMENT** et à partir de l'opérateur division, les indicateurs et les commandes.

⇒ Que se passe-t-il si on crée une commande sur une sortie ? Labview crée la commande mais ne la connecte pas. Si on relie par un fil le fil est brisé. Cliquer sur la flèche brisée pour en savoir plus. Faire control B pour enlever le fil, faire control Z pour annuler l'opération précédente et faire changer en indicateur pour que tout fonctionne correctement.

⇒ Montrer comment sélectionner, sur le diagramme, ces nouvelles instructions (commandes +opérateur +indicateur) et comment les supprimer.

⇒ **Construction de sous-programme** : pour finir transformer le programme en sous-programme, le sauver sous un nom mnémotechnique puis montrer comment l'utiliser.

NB : pour le définir en tant que sous-programme, il faut cliquer droit sur l'icône en haut à droite de la fenêtre face-avant ; **pas accessible à partir du diagramme !!!**). Associer un connecteur à une commande ou indicateur associé (Clic Gauche = CLKG sur connecteur puis CLG sur commande ou indicateur et ainsi de suite). Sauver le fichier.

Ouvrir un nouveau programme puis appeler le VI précédent à partir de la palette de fonctions >> Sélection d'un VI (en bas à gauche).

⇒ Montrer un problème lors de l'utilisation du sous-programme : sur le programme appelant, même si l'indicateur est un DBL alors 1/2 donne 0. Raison : l'indicateur du sous programme a été défini comme I32. Pour résoudre le problème, il suffit de modifier le sous-programme. Constatez (c'est remarquable) que LabVIEW reporte la modification dans le VI appelant automatiquement ! EN relançant le programme appelant : 1 / 2 = 0.5 !

Cahier des charges n°2: Application des notions précédentes

Ecrire un VI qui calcule $y = a * x^2 - b * x + 3 * \pi / x$

Les étudiants le réalisent en totale autonomie.

Contraintes : x commande numérique (DBL), a glissière (I32), b bouton (I32) et y vumètre

NB : leur montrer que la face-avant peut ressembler à la face-avant d'un multimètre (justification du nom face-avant).

NB : la constante π se trouve dans la palette numérique.

⇒ Montrer fonctionnement en boucle et phénomène de dépassement de calibre sur le vumètre. Régler le problème de dépassement de calibre en leur montrant comment modifier la plage du vumètre.

Cahier des charges n°3: une première structure : la boîte de calcul

En parallèle au diagramme précédent, réaliser un diagramme de même fonctionnalité mais qui utilise une boîte de calcul. On utilisera, pour alimenter la boîte les commandes précédentes.

Sans rien enlever l'enseignant insère la boîte de calcul, montre comment créer 3 entrées et 1 sortie sur la boîte, comment nommer chaque entrée et chaque sortie et comment écrire l'équation (en particulier **ne pas oublier** de terminer l'expression par ; (un point-virgule).

En lançant le programme, l'étudiant vérifie que les deux méthodes donnent le même résultat.

NB : noter dans l'expression de y le terme $3\pi/x$; une étudiante dans la boîte de calcul a écrit $3*$ puis a déposé l'icône représentant π ce qui ne marche pas !!

Un autre a écrit bx ce qui ne marche pas car il faut insérer le signe de multiplication entre b et x

Cahier des charges n°4: Structure WHILE et REGISTRE A DECALAGE

Calculer la suite $x(n)=x(n-1)+(n-2)*x(n-2)$ avec $x(0)=1$ et $x(1)=2$

On utilisera une boucle while, l'utilisateur aura la possibilité de sortir de la boucle à l'instant de son choix. On utilisera un indicateur qui permettra de visualiser chaque nouveau terme calculé.

Après rappel sur registre à décalage en électronique numérique, donner intérêt d'un registre ici. Montrer création du registre puis ajout d'une case afin que le registre comporte 2 éléments.

Montrer initialisation registre et placer indicateurs pour montrer évolution état du registre au cours de l'itération.

NB : le terme (n-2) permet de leur montrer l'intérêt de l'indice de boucle i (en bas à gauche).

NB : pour le calcul de $x(2)$, $n-2 = 0$, pour le calcul de $x(3)$, $n-2 = 1$ etc ... il suffit donc d'utiliser l'indice de boucle.

NB : ajouter un indicateur sur la boîte de transit du registre à décalage (boîte sur la bordure droite (montrer qu'elle n'affiche que la dernière valeur en sortie de boucle et rien pendant).

NB : initialiser le registre (attention $x(1)$ en haut et $x(0)$ en dessous ! Créer les commandes à partir de la face-avant.

Il faut maintenant gérer l'arrêt de la boucle : créer, à partir du diagramme, une commande reliée au terminal d'arrêt. Faire remarquer que la commande est entourée de vert (variable booléenne)

- ⇒ Montrer en explorant le sous-menu du terminal conditionnel que la boucle peut s'arrêter si la variable est vraie ou fausse. Demander la condition d'arrêt actuelle.
- ⇒ Montrer en explorant le sous-menu de la commande booléenne que l'on peut fixer son état par défaut. Demander l'état par défaut actuel. Laisser cet état (qui est sauvé si l'on sauve le programme). En déduire le comportement du programme si les valeurs par défaut sont conservées. Vérifier en lançant le programme.
- ⇒ Faire constater que LABVIEW affiche rapidement un over flow (inf). Insérer un ralentisseur dans la boucle (délai de 0.1 s)

NB : avant de sauver montrer comment, à partir du menu local des commandes, définir une valeur par défaut (qui sera enregistrée lors de la sauvegarde).

Cahier des charges n°5: TABLEAU

L'application précédente a un défaut : on perd les valeurs calculées sauf la dernière !

Idée : stocker les données dans un tableau

- ⇒ Montrer comment créer un tableau à partir de la face-avant (déposer tableau puis insérer à l'intérieur un type de commande qui définira le type du tableau)
- ⇒ Montrer où placer le tableau et la connexion à la structure while
- ⇒ Montrer nécessité d'activer l'indexation afin de stocker les données dans le tableau.
- ⇒ Montrer que le trait change de taille.
- ⇒ Lancer le programme et faire constater que le contenu du tableau ne s'affiche que quand on arrête la boucle
- ⇒ Montrer que le tableau peut-être créé à partir du diagramme.

Cahier des charges n°6: OUTILS TABLEAUX

L'application précédente a un défaut : le tableau ne contient pas les deux premiers éléments!

Idée : ajouter ces éléments au tableau précédent

- ⇒ Montrer la procédure (montrer l'existence d'une palette qui permet de manipuler les tableaux).
- ⇒ Dégager l'idée de polymorphisme des fonctions.
- ⇒ Terminer en leur demandant d'extraire la troisième valeur du tableau complet et le nombre de ses éléments.

Cahier des charges n°7: Structure FOR

L'application précédente a un défaut : on ne peut pas fixer à l'avance le nombre d'itérations !

- ⇒ Montrer comment **transformer** la boucle while en boucle for.
- ⇒ Montrer que la commande logique ne sert plus rien et qu'il faut donc l'enlever.
- ⇒ Fixer le nombre de boucles de manière à calculer les 100 premiers éléments de la suite. Valeur de N ?

⇒ Exécuter le programme et vérifier.

Cahier des charges n°8: Structure IF

Modifier l'application afin de calculer	$x(n)=x(n-1)+(n-2)*x(n-2)$	si	$x(n-1) < 10$
	$x(n)=0$	si	$x(n-1) > 10$

NB : l'algorithme doit fournir l'indice iB pour lequel le basculement à lieu

A PROPOS DE L'ENCHAINEMENT DES OPERATIONS

Le câblage impose une chronologie dans les opérations exécutée par le programme.

Cette chronologie est partiellement contrôlée par l'utilisateur. Le vérifier à l'aide du programme précédent.

- ⇒ Lancer le programme précédent en mode pas-à-pas et suivre la propagation des « bulles » sur les fils.
- ⇒ Montrer que l'on peut mettre des points d'arrêts.

Cahier des charges n°9 : Structure SEQUENCE

Dans certaines applications, l'utilisateur peut et/ou doit exécuter certaines actions avant d'autres actions. Sous LABVIEW, il dispose pour cela de la structure Séquence.

Réaliser l'exercice suivant :

Calculer le temps mis par un programme pour évaluer la fonction $y=f(x)$ suivante :

$$y = (\sin x)^2 + \sin x - 1$$

En 10^5 points couvrant uniformément l'intervalle $[-2, +2]$.

Avant de faire l'exercice on peut faire un VI de base de 3 séquences la première et la troisième servent à prendre le temps T1 et T2 et l'on peut mettre une temporisation sur la séquence 2 et démontrer que $T2-T1=Tempo$.

Laisser faire les étudiants en totale autonomie (en intervenant uniquement pour leur montrer où se trouvent les objets).

*Sur l'exercice proposé i varie de 0 à (10^5-1) , on veut que x varie de -2 à +2 donc $x=a*i+b$.*

$$x = [4/(10^5-1)]*i - 2$$

Cahier des charges n°10 : SIGNAUX : création affichage

Soit la fonction $y(t)=y_1(t)+y_2(t)$ avec $y_1(t)=\sin(w*t+ \pi/2)$ et $y_2(t)= 2*\sin(3*w*t)$

Calculer en utilisant une boucle FOR le tableau des valeurs de t, y1(t), puis y2(t) puis y(t) pour t variant de 0 à 20 msec par pas de 2 s. ($w=2*\pi*fe$ avec $fe=100\text{hz}$).

NB : le programme ne comportera aucune commande

- ⇒ Tracer le graphe de y en fonction de l'indice i.
- ⇒ Tracer les graphes superposés de y, y1 et y2 en fonction de l'indice i (noter que build array propose une option (cf menu local): choisir celle qui permet d'obtenir en sortie un tableau de tableau = double fil !
- ⇒ Tracer, en utilisant le tableau des temps t, le graphe y(t)
- ⇒ Tracer, sans utiliser le tableau des temps t, le graphe y(t)
- ⇒ Tracer y1 en fonction de y2 (Figure dite de Lissajous utilisée pour déterminer rapport fréquence ici 3). Graphe X-Y

Cahier des charges n°11 : SIGNAUX : traitement FFT et affichage

Modifier le programme précédent afin d'introduire deux degrés de liberté dans la paramétrisation du VI à savoir :

- a) une commande qui fixe la fréquence d'échantillonnage f_{ech} .
- b) une commande fixant le temps d'acquisition T_{acq} .
- c) une commande qui fixe la fréquence d'entrée f_e . ($w=2*\pi*f_e$)

Fixer les valeurs adéquates de ces commandes afin de retrouver les résultats précédents.

Compléter alors le VI afin d'afficher le tableau contenant le spectre d'amplitude de f(t) en fonction de l'indice.

Représenter graphiquement ce tableau en fonction de la fréquence.

- ⇒ Utiliser ces outils pour caractériser les structures.
 - Domaine affiché ? Dépend de quoi ? Le vérifier
 - Symétries ? Par rapport à quoi ? Valeur ?
 - Explication position des raies ?
 - ⇒ Que se passe-t-il si f_e diminue (à T_{acq} constant=20ms)?
 - ⇒ Comparer, chaque fois, fréquence signal réel et positions en fréquence des raies ? Conclusion ?
- Interprétation : faire rappel sur TF, Discrétisation TF (la fonction est-elle continue ou discrète ?), aliasing, TFD, FFT avec schémas au tableau (signal quelconque, en dessous spectre d'amplitude avec f_{max} , positionner f_0 et $3f_0$, en dessous spectre signal échantillonné pour cas $f_e > 2f_{max}$, en dessous spectre signal échantillonné pour $f_e < 2f_{max}$)
- ⇒ Demander alors de justifier, à l'aide des schémas précédents les positions des raies pour deux cas ($f_e > 2f_{max}$ et $f_e < 2f_{max}$).

Cahier des charges n°12 : SIGNAUX : traitement Corrélation

Auto corrélation, Inter corrélation et Mesure d'un retard

Etape1 : Ecrire un VI comportant 3 commandes : tmin, tmax, nbpts et qui calcule (via une boîte de calcul) la fonction $x(t)=\exp(-t)*\cos(4*\pi*t)$. Afficher le résultat en fonction du temps.

NB : si la fonction n'est pas centrée, ceci signifie que t démarre à 0 et non à tmin !

Etape2 : Compléter le VI par le calcul de $y(t)=0.5*x(t-d)$ avec $t_d=t+délai$ délai étant une commande supplémentaire de la face avant Superposer ce graphe au graphe précédent.

Etape3 : Réaliser le produit d'inter corrélation entre $x(t)$ et $y(t)$. Afficher le tableau résultant en fonction de l'indice (nombre de points ?) puis créer un autre graphe pour l'afficher en fonction du temps.

Etape4 : Fixer le décalage délai (un temps !) à zéro. Ou se trouve le maximum et quel est l'indice associé i_0 ? (utiliser LabVIEW pour obtenir ces deux informations)

Etape5 : Fixer un décalage à une valeur différente x de 0. Ou se trouve le max et quel est l'indice associé i_x ? Retrouver la valeur du décalage temporel à partir de la différence entre i_x et i_0 .

NB : le produit d'inter corrélation est utilisé dans la reconnaissance d'une forme dans une image.

Pour en savoir plus sur l'inter corrélation :

Commencer par tracer la fonction $f(t) = 0$ si $t < 0$, $f(0)=1$, $f(1/2)=1/2$ et $f(1)=0$ puis $f(t)=0$ si $t > 1$

Leur demander de tracer $y(t)=f(t+1)$ et $g(t)=f(t-1)$

Donner la définition du produit d'inter corrélation $h(t)$ entre deux fonctions $f(t)$ et $g(t)$

Donner, à titre de comparaison, la définition du produit de convolution

Expliquer graphiquement la « mécanique » associée au calcul du produit d'inter corrélation

Comment LabVIEW calcule-t-il le produit d'inter corrélation

$X(t)$ se présente sous la forme d'un tableau de taille N (indice variant de 0 à $N-1$)

$Y(t)$ se présente sous la forme d'un tableau de taille M (indice variant de 0 à $M-1$)

NB : N peut être égal ou différent de M

Question : quelle est la dimension du tableau contenant le produit d'inter corrélation ?

Réponse : dessiner le tableau Y en dessous du tableau X (tableau 1 ligne et N ou M colonnes) les éléments $x(0)$ et $y(0)$ étant alignés. Envisager alors toutes les translations possibles du tableau Y (d'abord vers la gauche, comme $x(t+1)$, ce qui donne $M-1$ translations possibles PUIS vers la droite, comme $x(t-1)$, ce qui donne $N-1$ translations possibles.

Bilan : en comptant l'absence de translation (translation de 0 indice) le nombre de possibilités est donc $(M-1)+(N-1)+1=M+N-1$

Montrer que si deux signaux ont des formes identiques alors le produit d'inter corrélation est maximum quand les deux signaux sont superposés (décalage nul). Illustrer en dessinant deux fonctions porte identiques l'une en dessous de l'autre PUIS en refaisant le dessin mais en décalant la fonction porte du dessous (-au moins de deux largeurs).

Gestion des indices : LabVIEW ne gère pas les indices négatifs (qui seraient utilisés ici pour gérer les translations vers la gauche). Une ré-indexation est donc nécessaire. Sous LabVIEW elle s'écrit (cf documentation sur la cross corrélation tirée du help en ligne de LabVIEW) : $Rxy_i = h_i - (N-1)$ pour $i = 0, 1, 2, \dots, N+M-1$. Le premier indice Rxy_0 est donc associé à $y(t-(N-1))$ c'est-à-dire à une translation de y maximale vers la droite ! Ce tableau est ensuite traduit vers la gauche.