

UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD

TW51 : ANALYSE FREQUENTIELLE

Rapport de projet

LEFEBVRE Maxime

Département Génie Mécanique et Conception
Filière Conception des Systèmes Mécatroniques

Sommaire

Sommaire	2
1 Objectifs du projet	3
2 Réalisation du projet	4
2.1 Présentation de LabVIEW.....	4
2.2 Le programme : code.....	4
2.2.1 <i>Utilisation des machines d'états</i>	4
2.2.2 <i>Etape 0 : initialisation des données</i>	5
2.2.3 <i>Etape 1 : Réalisation de l'essai (vertical ou horizontal)</i>	6
2.2.4 <i>Etape 2 : Traitement des données et affichage des résultats</i>	9
2.2.5 <i>Etape 3 : sauvegarde des résultats</i>	11
2.2.5.1 Mise en forme des données opérateur.....	12
2.2.5.2 Mise en forme des résultats.....	12
2.2.5.3 Sauvegarde des courbes au format Word.....	13
2.2.5.4 Enregistrement des courbes au format Excel.....	14
2.2.6 <i>Etape 4 : Fin de l'essai</i>	15
3 Le programme : manuel d'utilisation	17
3.1.1 <i>Onglet d'informations</i>	17
3.1.2 <i>Onglet acquisition</i>	18
3.1.3 <i>Onglet analyse des résultats</i>	19
4 Conclusion	21

1 Objectifs du projet

L'UV TW51 est centrée autour d'un projet proposé par un industriel. La réalisation de ce projet a pour but de prouver nos capacités à traiter un problème industriel pour concevoir un produit à partir de son cahier des charges. Le but du projet présenté dans ce rapport est de concevoir un logiciel de test pour l'analyse fréquentielle sur des arbres de transmission automobiles. A partir d'une procédure donnée par la société Fuji Autotech (voir Figure 1), basée à Mandeuve, j'ai travaillé à définir une solution de mesure en adéquation avec le cahier des charges suivant :

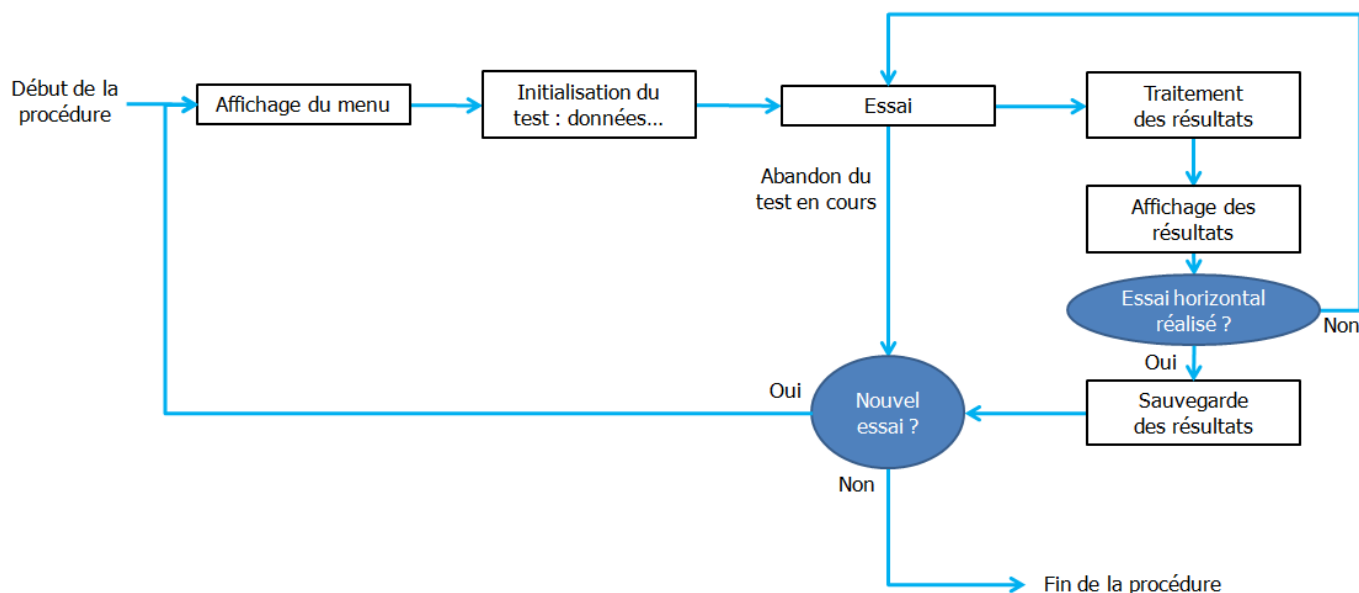


Figure 1 : Procédure de test souhaitée

Fonctionnalité souhaitées :

- Un cycle complet est constitué de plusieurs essais avec une excitation verticale puis autant d'essais avec une excitation horizontale. Le nombre d'essais dans chaque direction est fourni par l'opérateur. Pour chaque direction on doit obtenir ou afficher :
 - La courbe moyenne du signal issu du capteur,
 - Pour chaque excitation, la courbe du spectre instantané du signal issu du capteur,
 - Le pic de Dirac qui sert de signal d'excitation de l'arbre de transmission.
- La force d'excitation (hauteur du Dirac) doit être comprise entre 150 et 200 Newtons.
- L'opérateur doit pouvoir définir les seuils de conformités entre lesquels la fréquence propre de l'arbre doit se situer. Ces seuils sont définis dans les deux directions et afficher sur les graphes de résultats.
- L'opérateur a la possibilité d'arrêter le cycle à tout moment s'il juge le spectre instantané non-satisfaisant.
- Les résultats doivent être sauvegardés au format Word et les courbes au format Excel.
- Lorsqu'un nouveau essai est demandé, les données d'entrée sont conservées.

Matériel disponible :

- Un arbre de transmission de référence muni d'une masse simulant le poids du volant.
- Un capteur type accéléromètre positionnable sur la masse à l'aide de cire.
- Un marteau d'effort muni d'une tête de frappe molle
- Un conditionneur afin de traduire les signaux générés par le capteur et le marteau
- Une carte d'acquisition National Instrument NI-6221.
- Le logiciel LabVIEW.

2 Réalisation du projet

La suite de ce rapport s'attachera à présenter le travail réalisé ainsi que la méthodologie employée. Dans un premier temps je présenterai rapidement le logiciel LabVIEW. Ensuite j'aborderai le programme réalisé ainsi que les choix effectués pour résoudre les problèmes posés. Enfin je développerai étape par étape le fonctionnement de la procédure.

2.1 Présentation de LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) est un logiciel de développement d'applications d'instrumentation. Mis au point par la société américaine National Instrument, ce logiciel, utilisable dans un grand nombre de domaines, est plus particulièrement destiné à l'acquisition de données et au traitement du signal. En effet, il offre de larges possibilités de communication entre l'ordinateur et le monde physique (par cartes d'acquisitions analogiques ou numériques, cartes GPIB, réseau, liaisons série et parallèle, USB etc.) ainsi que d'importantes bibliothèques mathématiques permettant de réaliser de multiples traitements sur les signaux mesurés.

L'idée de LabVIEW est de remplacer les instruments de mesures et d'analyse d'un laboratoire par un ordinateur muni de cartes spécifiques et d'un logiciel approprié. Dans le cadre de la mesure, les cartes permettent de convertir des signaux électriques provenant de capteurs, en données numériques. Ainsi, un seul ordinateur muni d'une carte d'acquisition analogique et de LabVIEW est capable de remplacer un voltmètre, un fréquencemètre ou un oscilloscope. De plus, on pourra traiter, analyser et archiver sur disque automatiquement les mesures effectuées. Il s'agit donc du logiciel idéal afin de récupérer et traiter les signaux issus du capteur et du marteau.

2.2 Le programme : code

2.2.1 Utilisation des machines d'états

Nous l'avons vu sur la Figure 1, la procédure de test nécessite que nous revenions à l'étape "Essai" afin d'effectuer les mesures dans la direction horizontale. Il a donc fallu proposer une solution qui permettrait d'effectuer ce retour de façon simple. C'est pourquoi, j'ai décidé de mettre en place un système à machines d'états. Ce modèle simple facilite la définition de la séquence d'exécution des sections du code. L'implémentation spécifique détermine l'état suivant en fonction des décisions prises dans l'état actuel. La conception de ce modèle facilite l'insertion de nouvelles sections de code, la suppression de sections de code existantes ou le changement de l'ordre d'exécution des sections, le tout sans devoir modifier considérablement la structure de l'application.

La mise en place de machines d'états est simple. Il suffit de positionner une boucle `if` ou structure condition dans une boucle `while` (voir Figure 2). On crée ensuite une commande virtuelle qui va envoyer sur le registre à décalage de la boucle `while` l'état auquel accéder ensuite. Dans l'exemple suivant, la prochaine étape sera l'étape "Wait for Event".

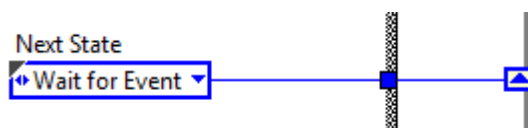


Figure 2 : Etat et machine d'états

On crée ensuite toutes les étapes qui vont nous être utile en ajoutant des conditions à la boucle `if`. En réalisant les actions nécessaires dans chaque étape puis en passant aux étapes situées immédiatement après dans l'ordre d'exécution, nous sommes capables de dérouler la procédure telle qu'elle est décrite dans le cahier des charges. Le changement d'onglet entre les différentes étapes s'effectue à l'aide d'un variable locale qui reçoit le nom de l'onglet à afficher comme le montre la Figure 3.



Figure 3 : Utilisation de la commande onglet

Les étapes de la machines d'états, présentées ci-après, correspondent toutes à une onglet spécifique dans le programme.

2.2.2 Etape 0 : initialisation des données

Lors de l'étape d'initialisation, nous devons récupérer auprès de l'opérateur les données de l'essai : nom de l'opérateur, référence du projet, numéro de la demande d'essai, etc. Pour cela nous allons lire en permanence les commandes affichées sur la face-avant du programme LabVIEW (voir Figure 4).

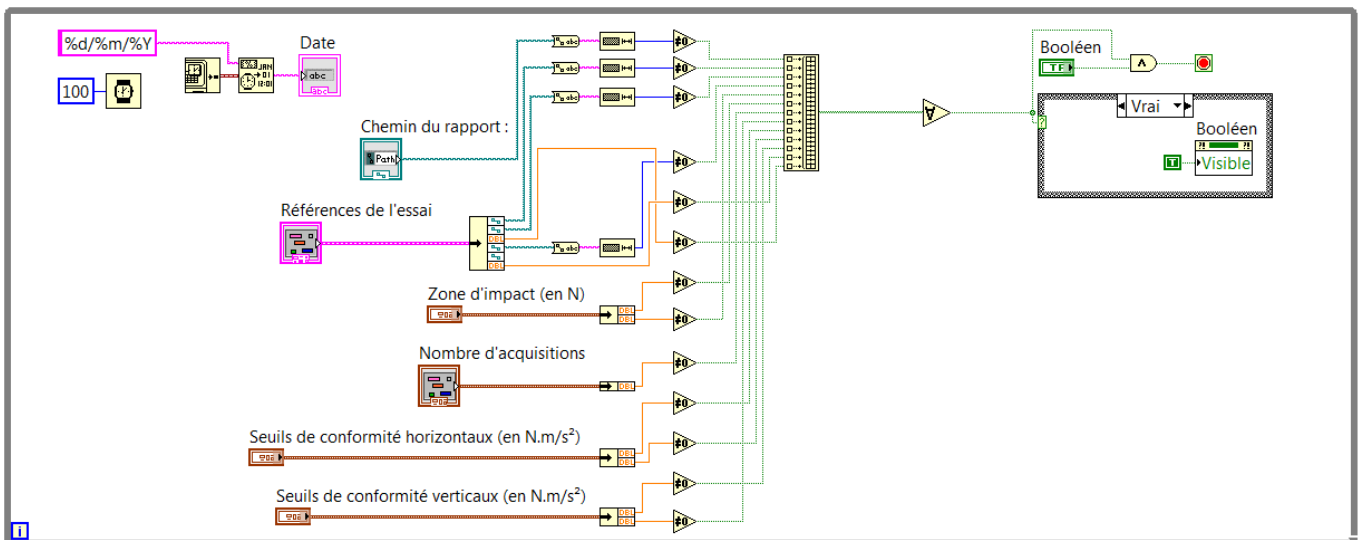


Figure 4 : Initialisation des données

Dans une boucle `while`, nous venons lire les commandes. Ces commandes sont soit des chaînes de caractères ou bien des nombres et sont placées dans des clusters pour des raisons de présentation. Lors du démarrage du programme, les commandes sont initialisées avec des valeurs données par l'industriel. L'opérateur modifie ensuite ces valeurs en fonction des besoins du test.

On effectue ensuite un ensemble de vérifications sur ces données et notamment qu'elles ne sont pas nulles. En effet, le test ne doit pas pouvoir démarrer si les données sont erronées. Pour effectuer ce test, on construit un tableau de valeurs logiques, résultats de la comparaison des commandes avec 0. On réalise ensuite un ET logique entre toutes les valeurs du tableau. Si le résultat est **VRAI**, alors on autorise l'affichage du bouton de démarrage de l'essai. Lors de l'appui sur ce bouton, la boucle s'arrête et on passe à l'étape suivante.

2.2.3 Etape 1 : Réalisation de l'essai (vertical ou horizontal)

Une fois les données fournies par l'opérateur, nous commençons une première série de mesures selon la direction verticale. Pour cela nous utilisons cette fois une séquence empilée dans laquelle nous allons définir l'ordre d'exécution des tâches nécessaires.

Tout d'abord nous devons créer la tâche d'acquisition sous NI MAX (voir Figure 5). Cette tâche est chargée de l'acquisition des signaux en provenance du marteau et du capteur. Selon les recommandations du client, nous prenons ici une fréquence d'échantillonnage de 800 Hz et un période d'acquisition de 2 secondes, soit 1600 échantillons. On choisit dans notre cas de récupérer le signal du capteur sur la voie ai0 et celui du marteau sur la voie ai1.



Figure 5 : Définition de la tâche NI MAX

Le conditionneur renvoie sur la carte un signal dimensionné selon les besoins du test. Cependant, les signaux ne sont pas exprimés dans l'unité souhaitée puisqu'ils sont encore en Volts. J'ai donc créé deux échelles linéaires afin d'obtenir les bonnes valeurs pour les courbes sous LabVIEW (voir Figure 6 et Figure 7) :

- Sur le capteur : sensibilité de 1.018 mV.s²/m soit un facteur d'échelle de $1 / 0.001018 = 982.32$.
- Sur le marteau : sensibilité de 11.20 mV/N soit un facteur d'échelle de $1 / 0.0112 = 89.29$

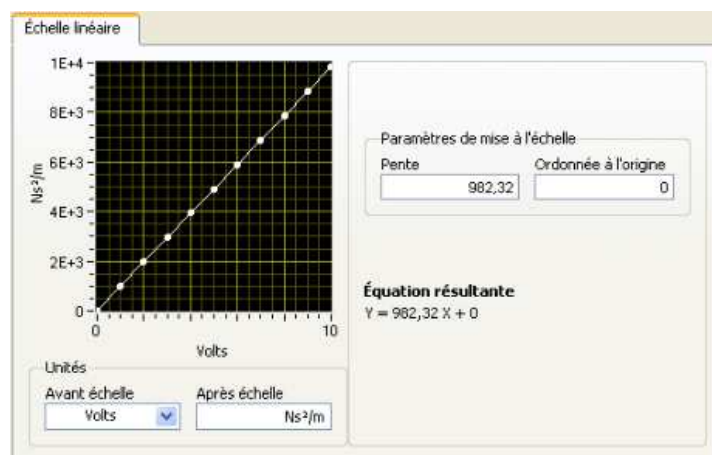


Figure 6 : Paramétrage des mises à l'échelle pour l'accéléromètre

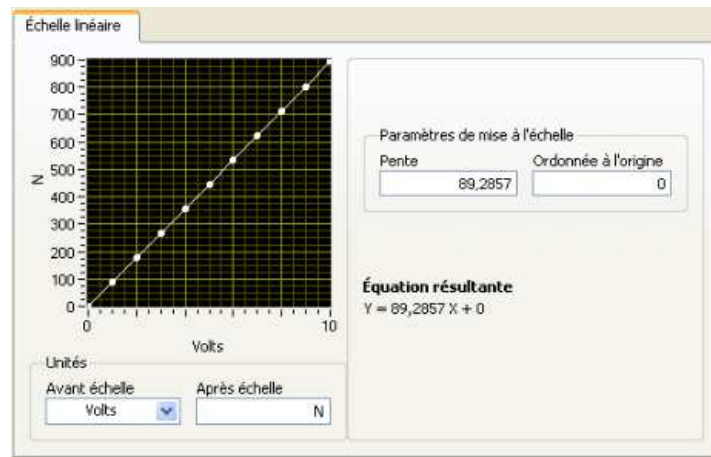


Figure 7 : Paramétrage des mises à l'échelle pour le marteau

La tâche correctement créé, j'ai mis en place la séquence empilée dans laquelle on trouve trois étapes (voir Figure 8) :

1. Lancement de la tâche,
2. Attente de fin d'initialisation du test
3. Réalisation de l'essai n° i

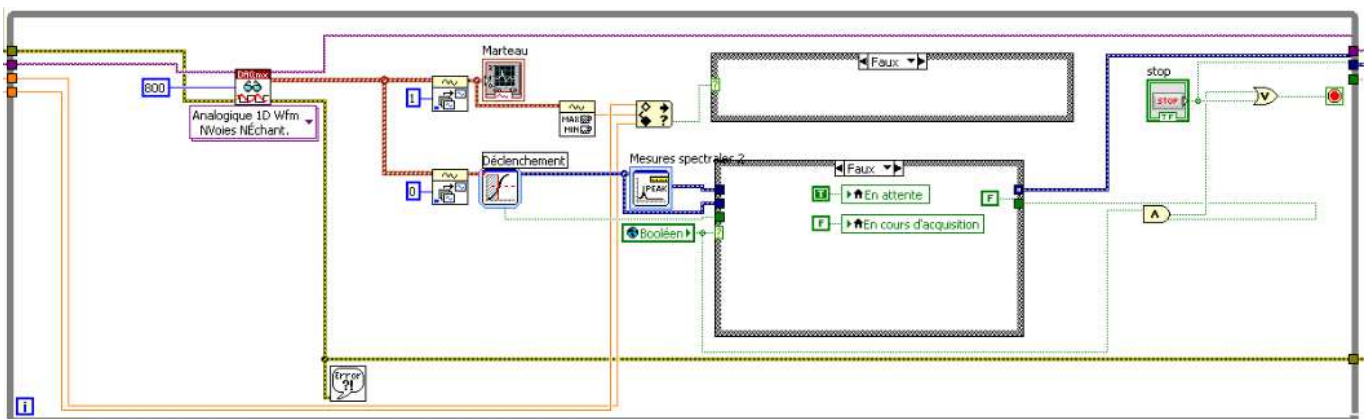


Figure 8 : Réalisation de l'essai n° i

Dans cette dernière étape, les deux voies sont lues de façon continue par paquet de 1600 échantillons. On détermine ensuite la valeur maximale de la courbe du marteau afin de déterminer la hauteur du Dirac d'excitation. Si cette valeur est dans la zone d'impact définie alors on retourne **VRAI**.

Sur le signal du capteur, on vérifie que l'excitation a bien été reçue. Pour cela on place un VI de déclenchement qui prend en argument la valeur de trigger ainsi que le recalage souhaité (ici 120 échantillons avant déclenchement). En effet, nous allons devoir faire la moyenne de ce signal sur plusieurs réalisations. Il faut donc s'assurer que le déclenchement de la voie soit toujours le même afin que les réponses aux excitations successives se superposent correctement. Pour cela on utilise les paramètres donnés sur la Figure 9.

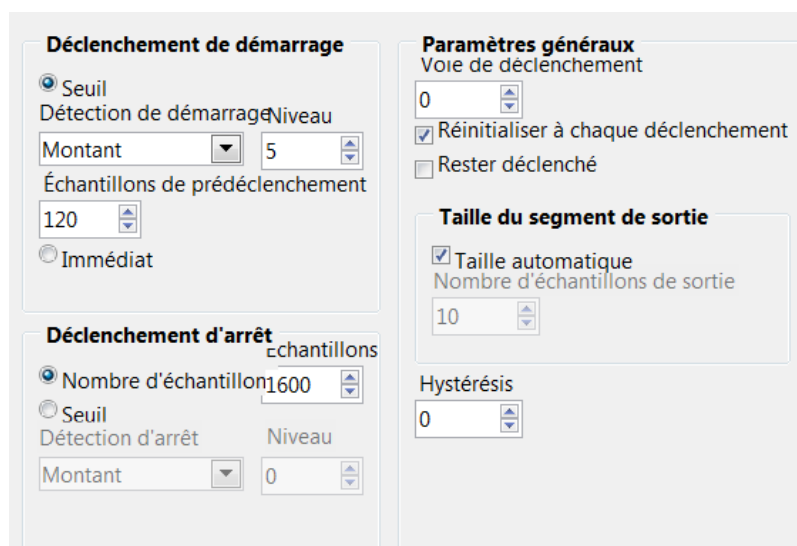


Figure 9 : Paramètres du VI de déclenchement

Lors du déclenchement ce bloc va renvoyer un booléen pour indiquer que des données sont disponibles. Si c'est le cas, on calcule le spectre d'amplitude du signal en dB (fenêtre de Hanning). On l'affiche alors dans le graphe du spectre instantané pour que l'opérateur puisse le visualiser. Le booléen est ensuite envoyé vers un ET logique afin d'indiquer au programme que le premier essai est terminé.

C'est lors de la détection du déclenchement que j'ai rencontré la première vraie difficulté de ce projet. En théorie ce ET logique entre la détection de l'excitation et le trigger devrait retourner **VRAI** lorsque l'essai se déroule correctement, mais ce n'est pas le cas. En réalité, après avoir placé un indicateur sur chaque booléen je me suis rendu compte que le trigger avait lieu au tour de boucle suivant celui de Dirac d'excitation. Les deux booléens n'étaient donc jamais en état logique haut simultanément. Pour régler ce problème, j'ai utilisé une variable globale. Lorsque le Dirac d'excitation est suffisamment haut (entre 150 et 200 N toujours), je stocke un booléen **VRAI** dans la variable globale afin de valider le fait que l'excitation a eu lieu. Cette variable gardera cette valeur jusqu'au prochain test. Ainsi lorsque le déclenchement sur le signal du capteur a lieu, le booléen associé passe à **VRAI** et le ET logique aussi. La boucle peut s'arrêter correctement.

Pour finir on réalise simplement la moyenne des signaux précédents à l'aide d'un registre à décalage. Cette méthode usuelle consiste à stocker la somme des signaux de 0 à $i-1$ dans le registre et à lui ajouter le $i^{\text{ème}}$ signal avant de diviser le tout par la valeur de l'incrément de boucle. Une fois la moyenne calculée pour l'instant t , si il reste de essais à faire on recommence l'étape 1, on arrête la tâche puis on la relance en attente d'un nouveau Dirac d'excitation.

On obtient ainsi en sortie la moyenne sur plusieurs réalisations de l'accélération observée par le capteur sur l'arbre de transmission.

NB : comme demandé par le client, l'essai peut-être arrêté à tout moment par l'opérateur. L'appui sur le bouton entraîne l'arrêt de la boucle d'acquisition ainsi que celui de la boucle du comptage d'essais (N essais). L'utilisateur est ensuite amené directement à l'étape 4.

2.2.4 Etape 2 : Traitement des données et affichage des résultats

Nous avons maintenant le signal issu du capteur et moyenné, nous pouvons traiter les données en sortie de la phase de test afin d'en extraire les résultats intéressants. Pour cela j'ai décidé de travailler en deux étapes. Tout d'abord je génère le spectre d'amplitude du signal avec le bloc vu précédemment (les réglages restent inchangés). Ensuite, en fonction de si je réalise les essais en vertical ou en horizontal, j'affiche le graphe correspondant et je récupère les données opérateur liées, notamment les seuils de conformité de l'essai. La Figure 10 montre la structure condition qui permet de réaliser la première partie.

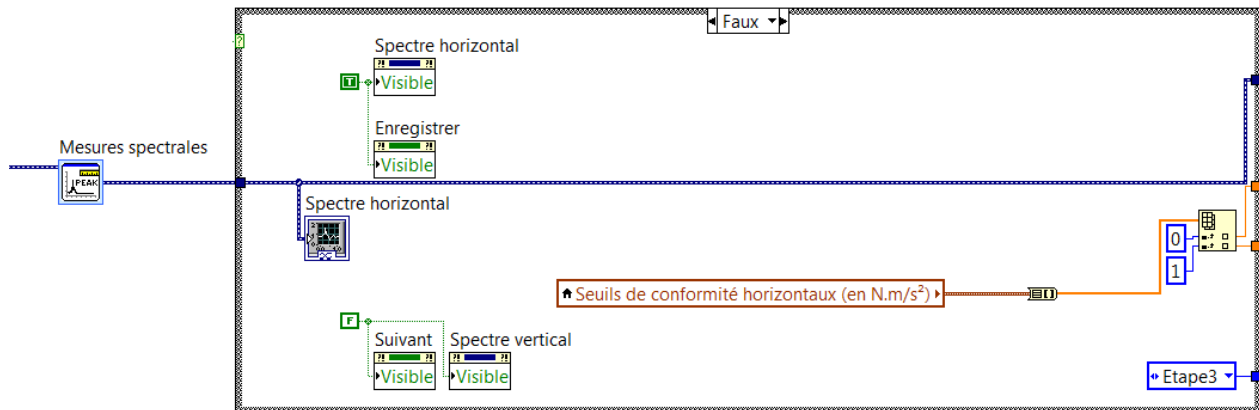


Figure 10 : Calcul du spectre et récupération des informations

Dans un deuxième temps ces données doivent être traitées afin de déterminer la fréquence propre de l'arbre de transmission. S'il est relativement simple de déterminer la valeur maximale d'une courbe (ex : pour le marteau) avec le VI correspondant, il est en revanche plus compliqué de connaître l'abscisse associée à cette valeur maximale.

Pour pouvoir obtenir cette valeur j'ai procédé de la manière suivante (voir Figure 11) :

- Extraction d'une partie seulement de la courbe. Lors des séries de tests effectuées, je me suis rendu compte que la fréquence propre du support de test s'exprimait parfois fortement. Cette fréquence propre étant basse, j'ai décidé d'exclure de la courbe les premiers échantillons jusqu'à environ 10 Hz. Les résultats attendus se situent autour de 50 à 70 Hz cela ne pose donc pas de problème.
- Calcul de la valeur maximale de la courbe. Cette valeur maximale est censée être l'amplitude du spectre à la fréquence de résonance.
- Recherche de l'abscisse associée à la valeur maximale de la courbe. J'utilise le VI "Rechercher dans une waveform" auquel je fournis la courbe ainsi que la valeur trouvée précédemment. Le bloc me renvoie le numéro de l'échantillon auquel on observe la valeur maximale.
- Décomposition de la waveform afin de récupérer le pas dt.
- Multiplication du numéro de l'échantillon par dt.
- Comparaison avec les seuils de conformité.

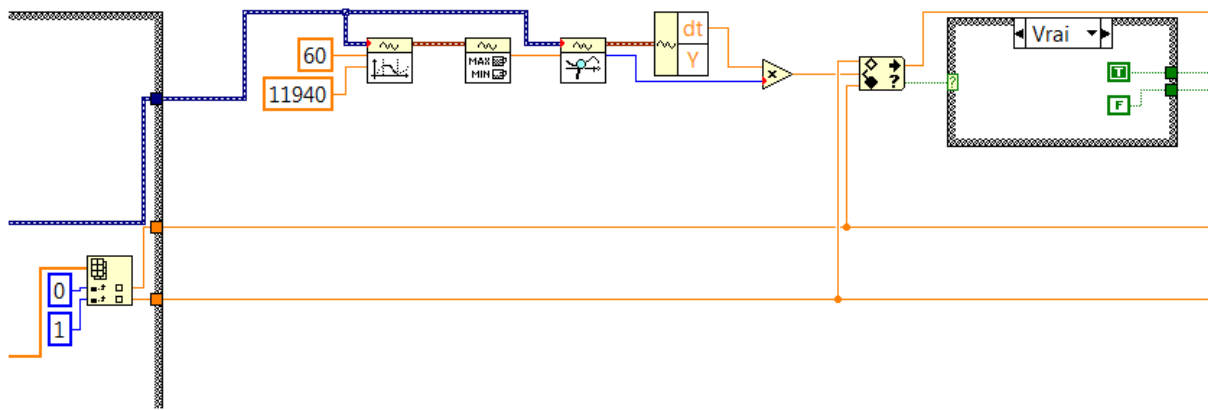


Figure 11 : Récupération de la fréquence propre

La comparaison avec les seuils de conformité s'effectue avec le VI "Dans la gamme". Si la valeur est entre les deux bornes définies, alors le VI renvoie la valeur en entrée ainsi qu'un booléen **VRAI**. Le test est alors conforme. Dans le cas contraire, la valeur en sortie est contrainte, c'est-à-dire que si la valeur en entrée n'est pas dans la gamme alors on récupère la valeur de la borne supérieure (respectivement inférieure) si on est au dessus (respectivement au dessous) de cette borne. Le VI renvoie alors **FAUX**. La distinction des cas se fait à l'aide d'une boucle condition.

Pour terminer on stocke tous ces résultats afin de pouvoir les afficher dans la suite de la procédure.

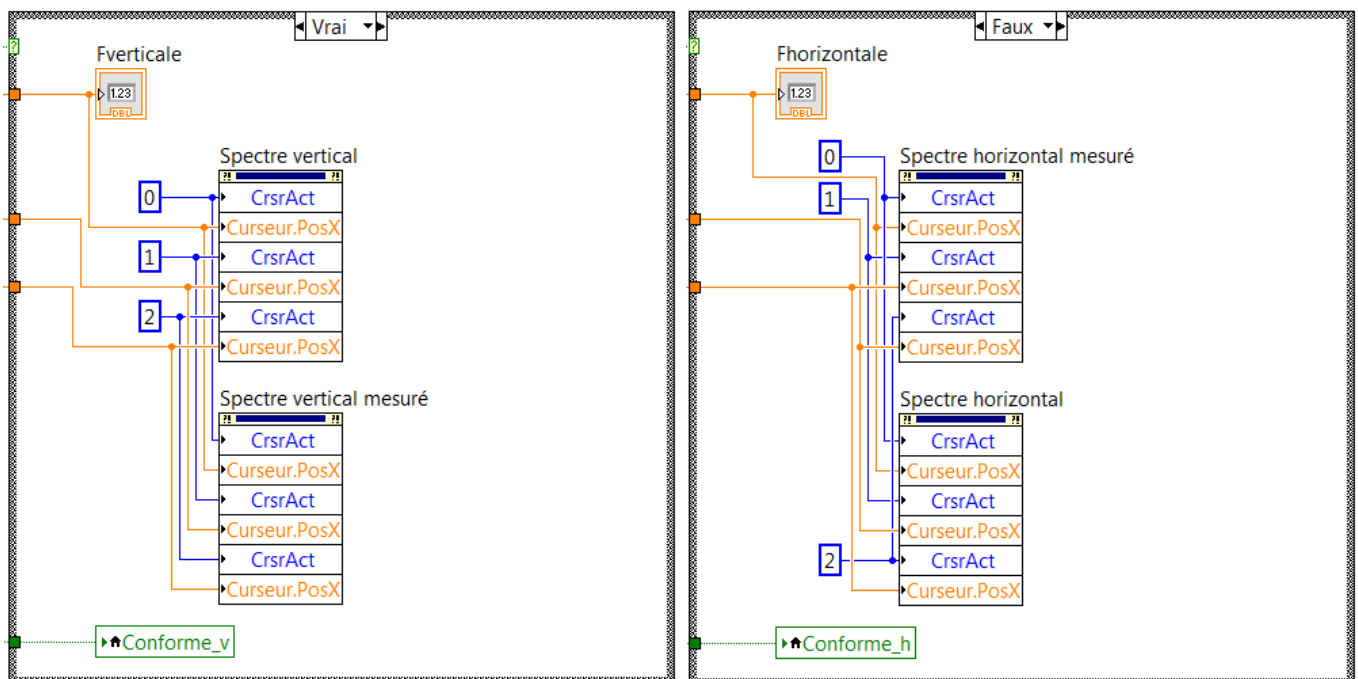


Figure 12 : Stockage des résultats dans les deux directions

Comme le montre la Figure 12, j'ai une nouvelle fois utilisé une structure condition afin de traiter indépendamment les deux directions. Dans chacune d'entre elles, je stocke la valeur de la fréquence propre dans un indicateur ainsi que le booléen de conformité dans une variable locale. Je positionne ensuite, à l'aide des nœuds de propriétés, les curseurs utiles sur les différents graphes. Il en existe 2 pour chaque direction : le spectre affiché lors de l'essai et celui enregistré dans le fichier de résultats.

En parallèle de ce traitement, le programme se met en attente de l'appui sur un bouton.

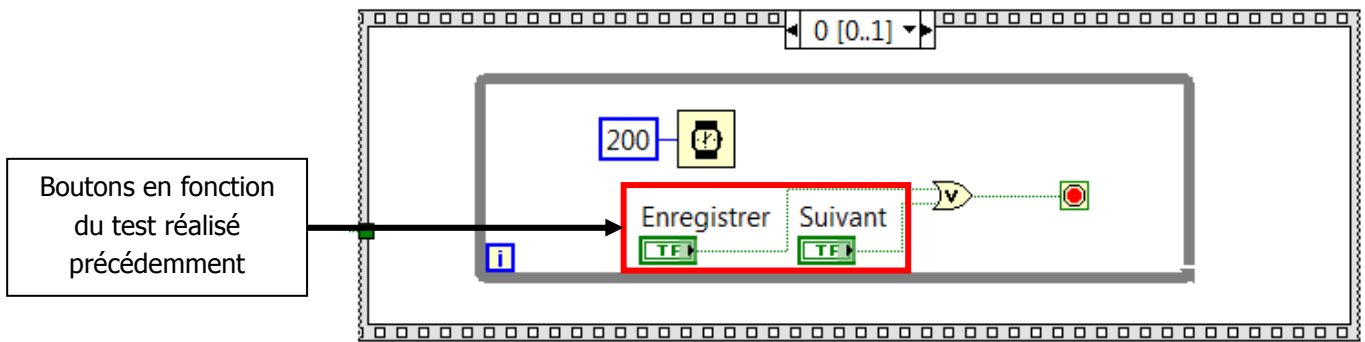


Figure 13 : Boucle d'attente d'action utilisateur

Comme le montre la Figure 13, lors de cette étape il y a deux possibilités :

1. Nous venons de faire les essais avec excitation verticale. Dans ce cas, un bouton "Etape suivante" est disponible sur l'onglet d'affichage. Lors de l'appui sur ce bouton un pop-up apparaît afin d'indiquer à l'opérateur qu'il doit changer le capteur de position avant de réaliser les essais horizontaux (voir Figure 14). Lorsque l'opérateur valide le repositionnement du capteur, le programme revient au début de l'étape 1 pour une nouvelle série d'essais.

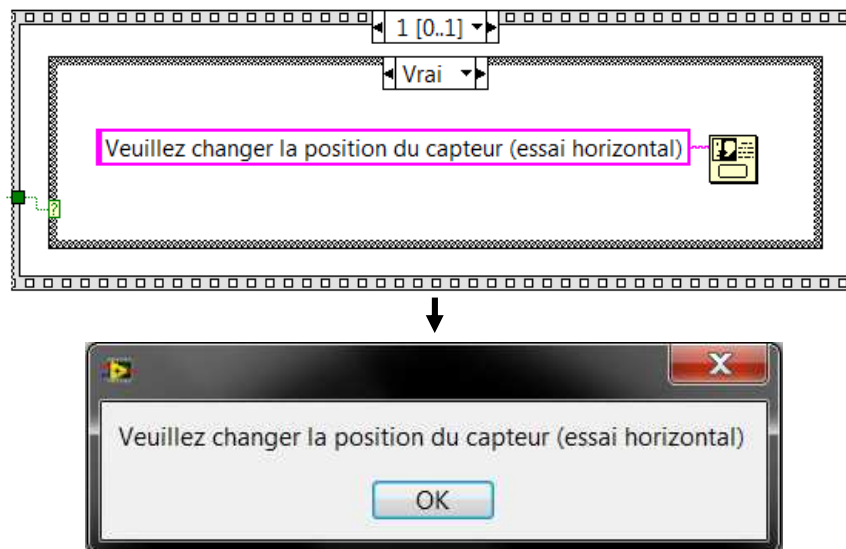


Figure 14 : Pop-up pour le repositionnement du capteur

2. Nous venons de faire les essais avec excitation horizontale. Dans ce cas, un bouton "Enregistrer" est disponible sur l'onglet d'affichage. Lors de l'appui sur ce bouton on passe à l'étape 3.

2.2.5 Etape 3 : sauvegarde des résultats

A la fin de l'étape 2 nous avons stocker les résultats dans différentes variables. Il nous reste donc maintenant à lire ces variables pour les mettre en forme et ainsi pouvoir les enregistrer selon le gabarit fourni par le client. Pour cela j'ai réalisé plusieurs procédures.

2.2.5.1 Mise en forme des données opérateur

Sur chaque rapport d'essai doivent figurer les informations suivantes :

- Le nom de l'opérateur,
- La date de l'essai,
- La référence du projet,
- Le commentaire de l'opérateur

Pour cela il suffit de récupérer les données dans le cluster afin de le mettre sous forme de tableau (voir Figure 15).

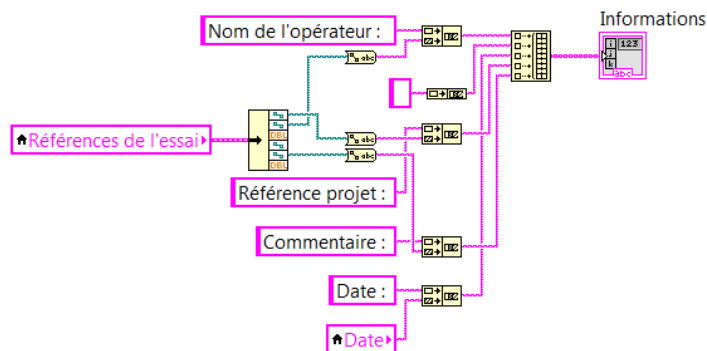


Figure 15 : Mise en forme des données du test

2.2.5.2 Mise en forme des résultats

De la même manière, les résultats doivent apparaître dans le rapport sous une forme spécifiée par le client. Il s'agit en fait d'un tableau qui récapitule les fréquences propres obtenues dans les deux direction, le seuil minimal de conformité ainsi que la conformité ou non de la pièce. Le tableau finale doit se présenter sous la forme de la Figure 16.

Résultats des essais		
Valeur	72.25	63
Spécification	≥ 52.00	≥ 52.00
Conforme	OUI	OUI

Figure 16 : Mise en forme souhaitée des résultats

Afin de réaliser cette mise en forme, j'ai dû générer un tableau 2D de chaînes de caractères (voir Figure 17). J'ai d'abord transformer les différentes variables nécessaires en chaînes de caractères. Pour les entier, c'est le VI "Nombre en chaîne fractionnaire" qui a été utilisé. On prend pour précision 2 chiffres après la virgule, ce qui est suffisant pour analyser le résultat. En ce qui concerne la conformité du test, j'ai utilisé le VI "Formater en chaîne" adapté aux booléens.

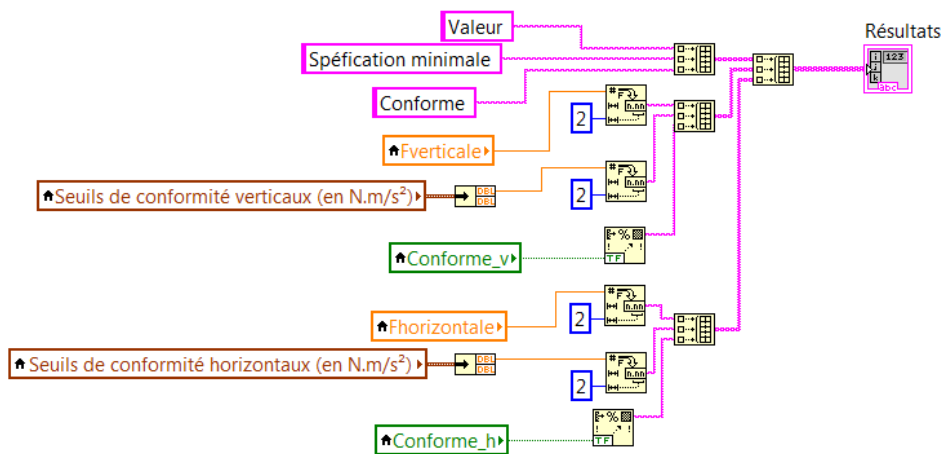


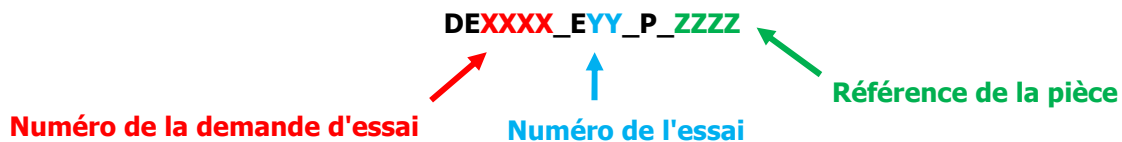
Figure 17 : Construction du tableau des résultats

Pour construire le tableau, je procède en deux étapes. Dans un premier temps je construis chaque ligne grâce au VI "Construire un tableau". La première ligne correspond au titre de chaque colonne. La deuxième ligne correspond aux résultats des essais verticaux. La dernière ligne correspond aux résultats des essais horizontaux. Je construis ensuite le tableau final en utilisant ce même VI mais en lui donnant pour arguments les trois lignes précédemment créées.

J'obtiens finalement bien le tableau souhaité.

2.2.5.3 Sauvegarde des courbes au format Word

Maintenant que nous avons toutes les informations nécessaires, il faut pouvoir les sauvegarder au format désiré c'est à dire sous un fichier Word. Pour commencer il faut définir le chemin de ce fichier selon les spécifications clients :



Ces données sont déjà présentes dans les données fournies par l'opérateur. Il suffit donc de les assembler correctement afin de définir le nom du fichier. Pour cela on utilise le VI qui concatène les chaînes. On lui fournit tout d'abord le chemin du rapport. On ajoute ensuite tous les éléments du nom du fichier comme sur la Figure 18.

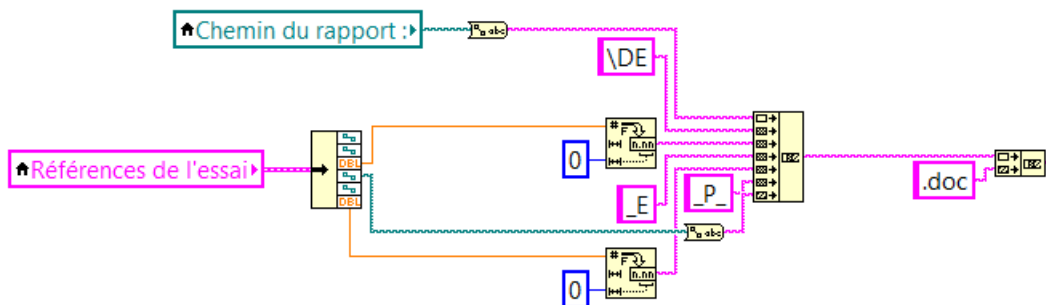


Figure 18 : Construction du nom du rapport

NB : Comme le fichier Word et le fichier Excel ne diffèrent que par leur extension, cette dernière est ajoutée séparément pour les deux fichiers.

Une fois le chemin du rapport clairement défini, il faut le remplir ! Pour ce faire j'ai opté pour la solution suivante : on affiche le tableau des informations, celui des résultats ainsi que les deux courbes et on cache la commande onglet. On réalise ensuite une impression d'écran du programme que l'on place dans le rapport. Finalement, on sauvegarde le rapport à l'endroit souhaité. Cette procédure est effectuée à l'aide de 3 VI présentés sur la Figure 19.

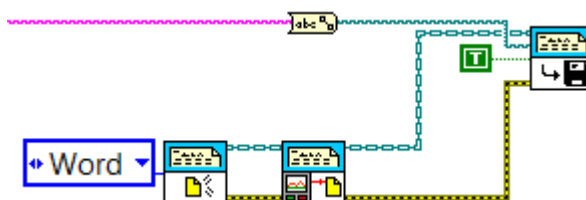





Figure 19 : Enregistrement du rapport au format Word

Ces trois VI sont des VI de la palette "Génération de rapport". Ils ont pour fonction :

1. "Nouveau rapport"  spécifier le type de fichier à enregistrer. Ici un fichier Word.
2. "Ajouter une image de la face-avant au rapport" 
3. "Enregistrer le rapport dans un fichier" 

La fonction d'enregistrement prend pour argument le chemin du fichier mais aussi un booléen. Ce dernier permet de prévenir l'opérateur qu'il risque d'écraser un fichier si celui-ci existe déjà. On évite ainsi la perte de données en cas de mauvaise manipulation.

Le rapport est maintenant enregistré au format Word.

2.2.5.4 Enregistrement des courbes au format Excel

Le client souhaite de plus que les valeurs des courbes soient enregistrées dans un fichier Excel afin de pouvoir les récupérer à posteriori. Ici la méthode est plus simple puisqu'il existe déjà un VI tout fait qui permet d'enregistrer des données au format tableur. Il s'agit du VI "Ecrire dans un fichier tableur" auquel il suffit de fournir :

- Le chemin du fichier
- Les données à écrire dans le fichier (tableau 1D et ou 2D)
- Un booléen qui permet de transposer les données
- Un booléen qui permet d'ajouter les données à un fichier déjà rempli

A cet instant nous disposons déjà de la valeur de chaque point pour les courbes. Cependant nous n'avons pas les abscisses correspondantes à toutes ces valeurs. Pour reconstruire le tableau des abscisses, on décompose la waveform du spectre. On obtient ainsi le pas de calcul Δt . Nous savons par ailleurs que pour une fréquence d'échantillonnage F_e donnée, LabVIEW ne trace le spectre d'amplitude que jusqu'à $F_e / 2$. Dans notre cas $F_e = 800$ Hz donc le spectre calculé s'étend de 0 à 400 Hz en abscisse. J'ai donc multiplié le Δt obtenu par le compteur d'une boucle `while` jusqu'à ce que le résultat de la multiplication soit supérieur à 400. En parallèle, chaque valeur est

ajoutée comme valeur d'indice i d'un tableau. Une fois cette procédure terminée, le vecteur des abscisse est reconstitué.

Le tableau obtenu possède un élément en trop puisque le dernier élément ajouté est celui pour lequel $dt \times (i+1)$ est supérieur à 400 Hz. Ce dernier élément a donc pour valeur $400 + dt$ Hz. Seulement cette fréquence n'existe pas dans le spectre et ne doit donc pas apparaître dans les résultats. J'ai donc utilisé le VI "Supprimer une portion d'un tableau" pour supprimer une portion de taille 1 et commençant au dernier élément.

Il ne reste plus qu'à reconstruire le tableau 2D à enregistrer dans le fichier Excel. Ici j'ai encore procédé en 2 étapes :

1. Construction du tableau des titres de chaque colonnes et création du fichier Excel.
2. Construction du tableau 2D avec les courbes et ajout au fichier Excel existant après transposition. Les 2 booléens du VI sont donc à l'état **VRAI**.

La procédure de sauvegarde est détaillée sur la Figure 20.

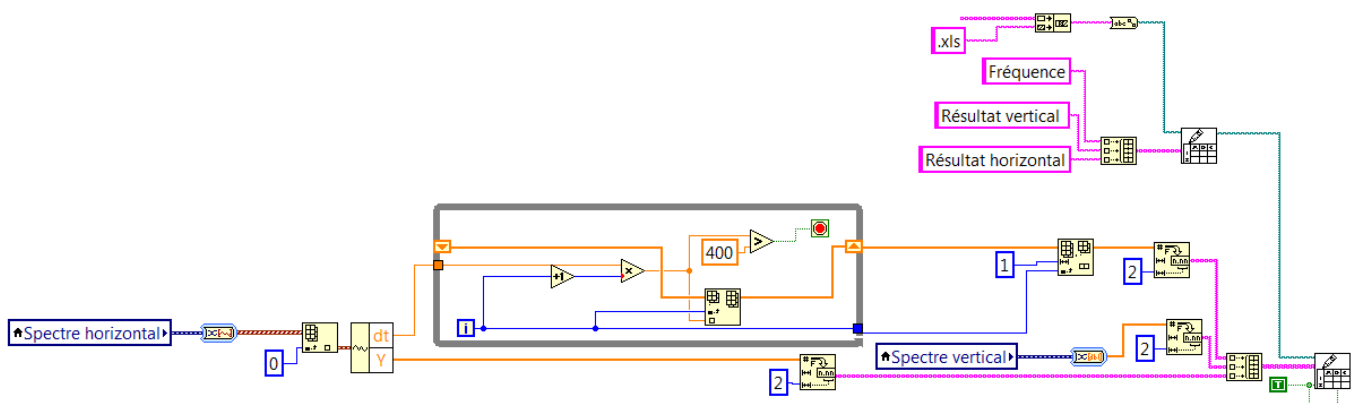


Figure 20 : Ecrition des courbes dans un fichier Excel

2.2.6 Etape 4 : Fin de l'essai

Toutes les étapes d'analyse ont été menées et la sauvegarde à bien eu lieu. Il reste maintenant une très courte étape qui consiste à demander à l'opérateur s'il souhaite faire un nouvel essai. En effet, l'ensemble du programme est placé dans une grande boucle **while** afin de pouvoir recommencer les essais autant de fois que nécessaire.

Dans cette dernière étape nous effectuons deux tâches :

1. Un pop-up apparaît sur l'écran (voir Figure 21) afin de demander à l'opérateur s'il souhaite arrêter les essais ici. S'il répond "Oui", alors le programme s'arrête. Autrement le système retourne à l'étape d'initialisation et tout recommence.

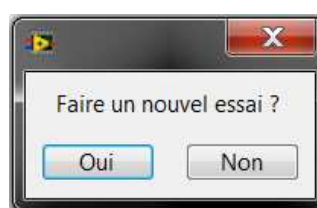


Figure 21 : Pop-up de fin de procédure

- Si un nouvel essai est demandé, toutes les données fournies par l'opérateur sont envoyées vers un registre à décalage pour que le test suivant soit initialisé avec les bonnes données directement. La valeur du numéro d'essai est incrémentée de 1.

La procédure complète est présentée sur la Figure 22.

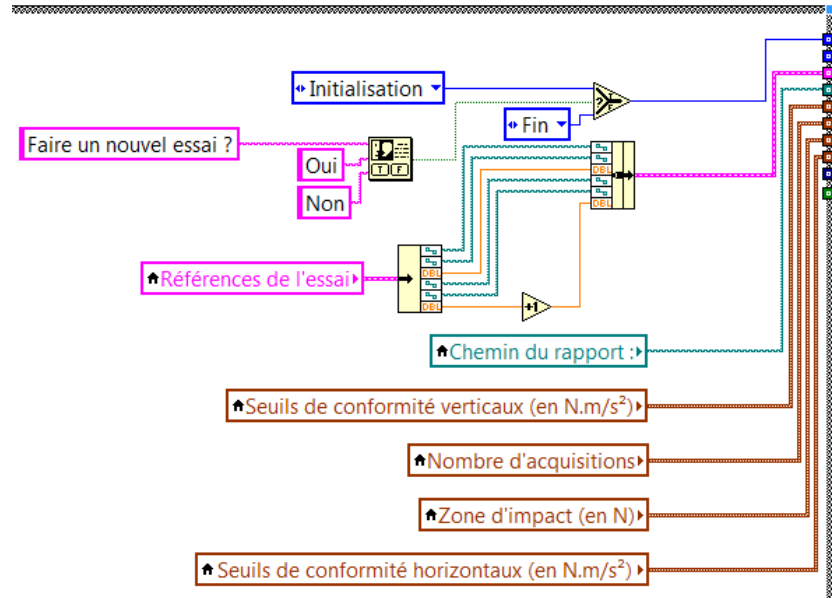


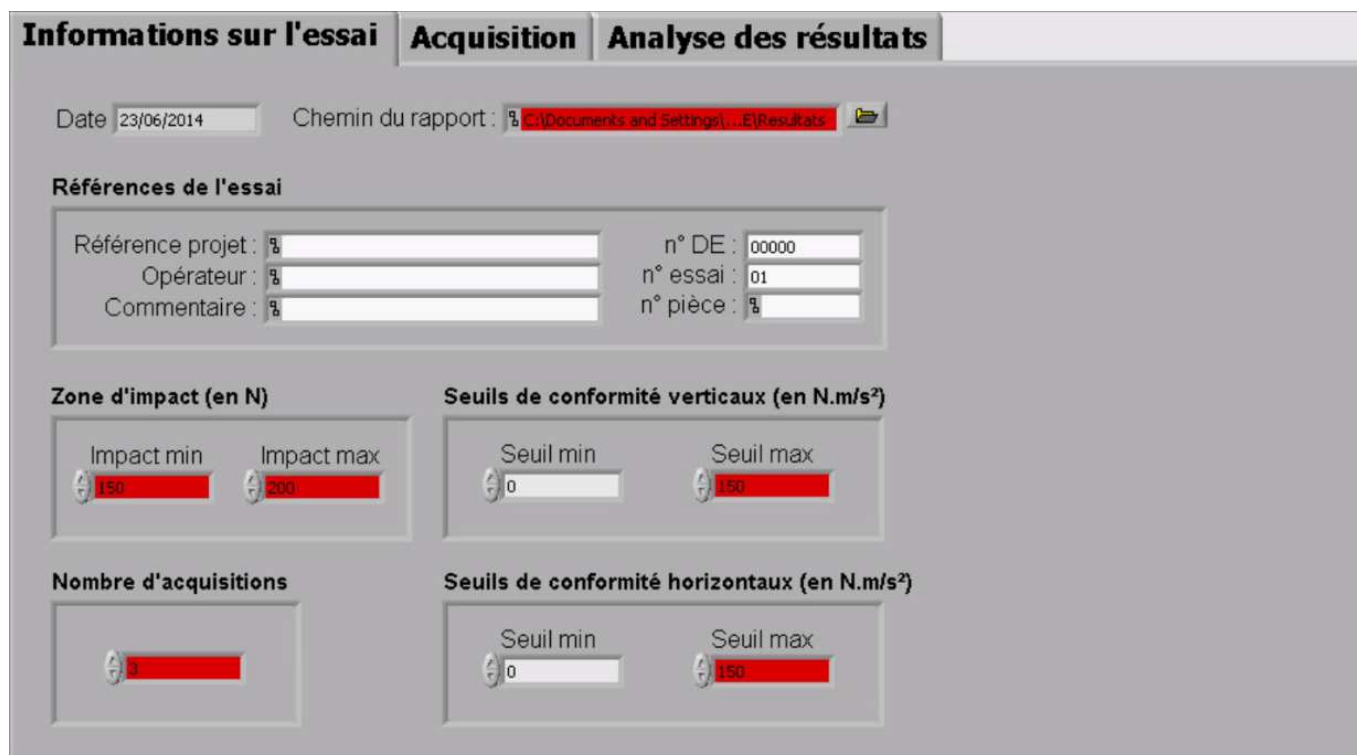
Figure 22 : Procédure de fin du programme

3 Le programme : manuel d'utilisation

Maintenant que le programme est codé et fonctionnel, il est intéressant de le tester et surtout d'expliquer comment il fonctionne. Ci-après est donc présentée une procédure simplifiée de l'utilisation du programme.

3.1.1 Onglet d'informations

L'onglet d'informations sur l'essai est l'onglet sur lequel l'opérateur va rentrer les données nécessaires au bon déroulement du test et à l'enregistrement du rapport de résultats. Comme nous l'avons vu plus haut, les données sont initialisées à une valeur particulière et il est impossible de lancer l'acquisition tant qu'il manque des données. L'onglet se présente donc comme sur la Figure 23.



The screenshot shows the 'Informations sur l'essai' tab with the following fields and values:

- Date: 23/06/2014
- Chemin du rapport: %1\Documents and Settings\...E\Results
- Références de l'essai:
 - Référence projet: %
 - Opérateur: %
 - Commentaire: %
 - n° DE: 00000
 - n° essai: 01
 - n° pièce: %
- Zone d'impact (en N):
 - Impact min: 150
 - Impact max: 200
- Seuils de conformité verticaux (en N.m/s²):
 - Seuil min: 0
 - Seuil max: 150
- Nombre d'acquisitions: 3
- Seuils de conformité horizontaux (en N.m/s²):
 - Seuil min: 0
 - Seuil max: 150

Figure 23 : Onglet d'information lors du lancement du programme

Nous pouvons voir que les champs de chaîne sont vides et que les champs numériques sont à 0 sauf le numéro de l'essai qui est géré par le programme. La date est aussi affichée automatiquement. Les champs en rouge représente les champs dont la valeur par défaut est définie par le client et ne doivent donc pas être modifiés avant le lancement du test. Une fois tous les champs correctement remplis, le bouton de démarrage du test apparaît comme nous le montre la Figure 24. Il suffit de cliquer sur le bouton pour passer à l'onglet d'acquisition.

Informations sur l'essai
Acquisition
Analyse des résultats

Date: Chemin du rapport:

Références de l'essai

Référence projet: <input type="text" value="Essai vidéo"/>	n° DE: <input type="text" value="12563"/>
Opérateur: <input type="text" value="Maxime LEFEBVRE"/>	n° essai: <input type="text" value="01"/>
Commentaire: <input type="text" value="Ceci est un test"/>	n° pièce: <input type="text" value="AE-253"/>

Zone d'impact (en N)

Impact min: <input type="text" value="150"/>	Impact max: <input type="text" value="200"/>
--	--

Seuils de conformité verticaux (en N.m/s²)

Seuil min: <input type="text" value="75"/>	Seuil max: <input type="text" value="150"/>
--	---

Nombre d'acquisitions

Seuils de conformité horizontaux (en N.m/s²)

Seuil min: <input type="text" value="75"/>	Seuil max: <input type="text" value="150"/>
--	---

Figure 24 : Onglet d'information avant le démarrage du test

3.1.2 Onglet acquisition

L'onglet acquisition est l'onglet au sur lequel vont être réalisés les mesures et leur moyenne. Comme nous l'avons vu, le système travaille en 3 phases :

1. Le programme commence par s'initialiser. Il ne faut surtout taper sur l'arbre pendant cette phase.
2. Ensuite le programme se met en attente du Dirac d'excitation. Il faut alors frapper une fois dans la bonne direction et avec une force comprise entre les deux valeurs définies dans le cluster "Zone d'impact". En cas d'échec à taper dans cette zone, il faut atténuer au maximum les vibrations résiduelles de l'arbre avant de frapper une deuxième fois avec le marteau.
3. Lorsque l'excitation du Dirac a été détectée, le programme passe dans sa dernière phase : l'acquisition de données. Le programme travaille jusqu'à avoir acquis les derniers échantillons du signal post-trigger. Après ce dernier échantillon, le système repasse en initialisation.

Ces trois phases sont affichées à l'aide d'indicateurs lumineux présentés sur la Figure 25. Un exemple d'utilisation de l'onglet d'acquisition est présenté sur la Figure 26 (les résultats ne sont pas des résultats réels).



Figure 25 : Les indicateurs d'état du système

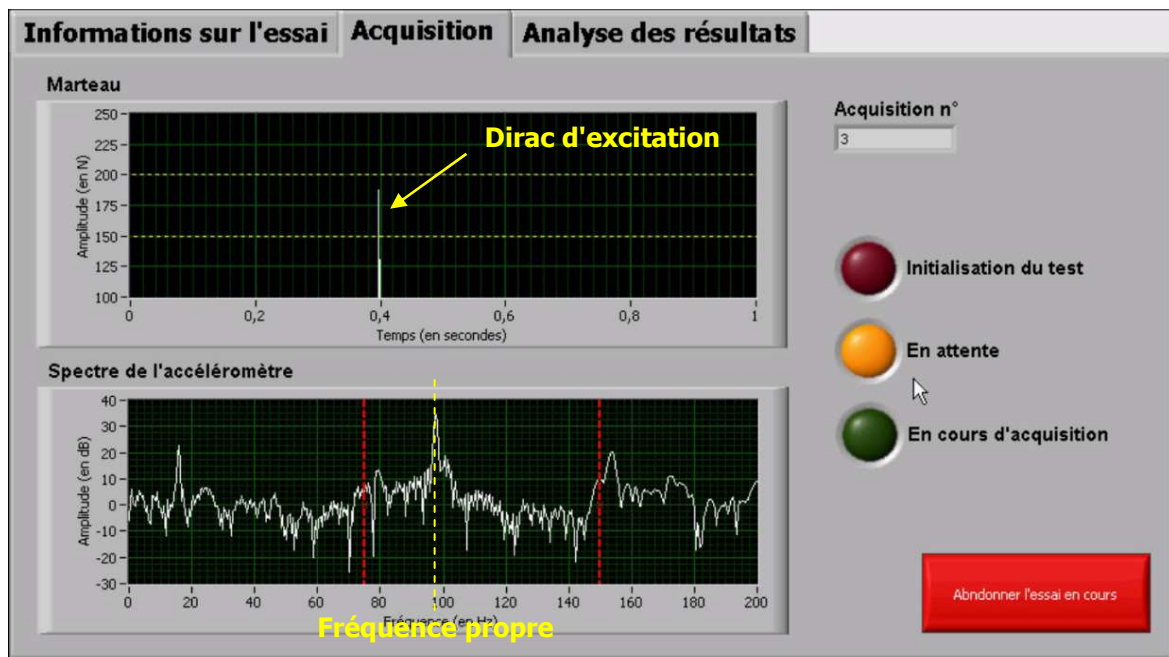


Figure 26 : Exemple d'utilisation

Il est quand même à noter que le programme passe en phase d'acquisition à l'itération de boucle suivant la détection de l'excitation. C'est pour cette raison que l'indicateur vert est encore à **FAUX** sur cette image.

Si on appuie sur le bouton d'abandon avant la fin du test, alors on retourne sur l'onglet d'informations et le numéro d'essai est incrémenté. Sinon, lorsque le nombre d'itérations pour les essais est atteint, le programme passe à l'onglet d'analyse des résultats.

3.1.3 Onglet analyse des résultats

C'est dans l'onglet d'analyse que l'on affiche le résultat des différents essais pour une direction donnée. Sur le graphe est affichée la fréquence propre. De plus un indicateur lumineux donne l'information sur la conformité du test. Un exemple est présenté sur la Figure 27.

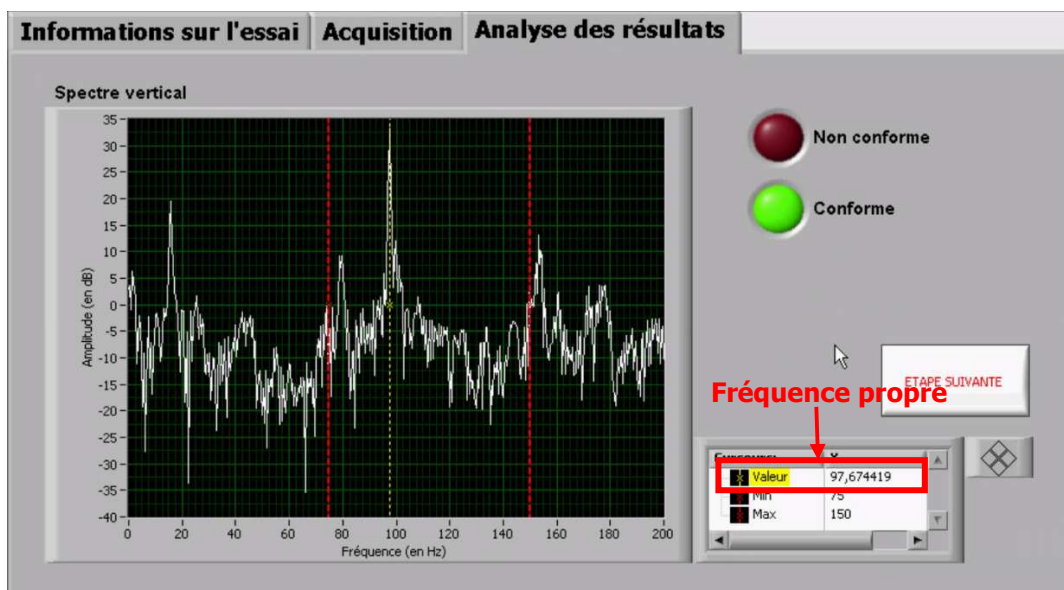


Figure 27 : Exemple de l'onglet d'analyse pour un test vertical conforme

Nous sommes ici à la fin des essais verticaux car le bouton "Etape suivante" est présent. A l'appui sur ce bouton, le pop-up de repositionnement du capteur va apparaître puis lors de la validation, le programme va retourner sur l'onglet d'acquisition afin de réaliser les mesures horizontales. Une fois ces mesures effectuées, l'onglet analyse s'affichera une nouvelle fois mais avec un bouton "Enregistrer". A l'appui, les fichiers de rapport vont être créés et le fichier Word va s'ouvrir (voir Figure 28). Le pop-up de fin de procédure apparaît alors. Si l'on appuie sur "Oui", le programme retourne à l'onglet d'informations et le numéro d'essai s'incrémente. Si non, le programme s'arrête.

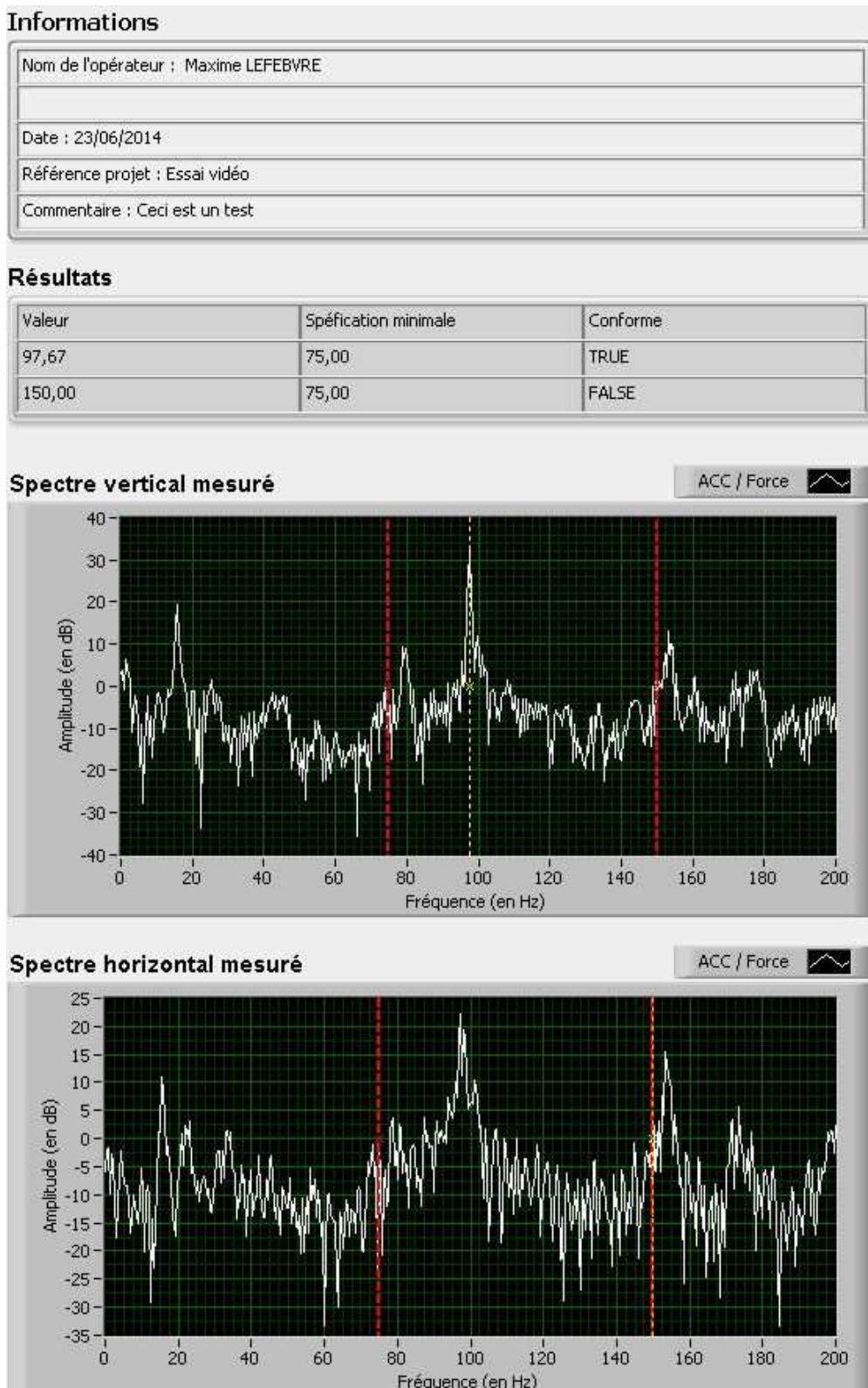


Figure 28 : Contenu du fichier Word en sortie du programme

4 Conclusion

Malgré quelques petites difficultés lors de la présentation chez le client, ce projet répond au cahier des charges tel que précisé en introduction. En effet, les tests effectués montrent que le programme d'acquisition reproduit fidèlement les résultats obtenus par le client avec son banc de référence. Au cours de cette présentation, j'ai tout de même procédé à quelques modifications notamment de la durée d'observation du signal (2 à 5 secondes). Il a juste suffi de modifier le nombre d'échantillons à acquérir (1600 à 4000).

Sur un plan personnel, ce projet m'a beaucoup apporté. En plus du fait de mettre en œuvre mes compétences sous le logiciel LabVIEW, il m'a notamment permis d'approfondir les connaissances acquises au cours de ce semestre passé à l'UTBM et particulièrement en traitement du signal. Je pense que ce semestre de projet m'aura donné envie de continuer à travailler sur ce genre de problématiques.

Par ailleurs, cette TW a été l'occasion de tester mes capacités à gérer un projet lié à la mécatronique. J'espère pouvoir mener dès le semestre prochain un projet de plus grande envergure car je reste persuadé que le travail effectué m'a permis de m'affirmer en tant que futur ingénieur.