

TP-4 Conception d'un programme LabVIEW pour la surveillance de la déformation des poteaux d'une presse

Par Luc Marchand, Ph.D., ing.

1. Introduction

Pour ce quatrième travail pratique, vous allez concevoir un programme LabVIEW qui aidera l'opérateur d'une presse à contrôler le niveau de déformation dans les poteaux de celle-ci. Cette presse sert au forgeage à chaud de grosses pièces d'acier qui sont à la fois comprimées et étirées (Fig. 1) dans le but d'obtenir des plaques qui ont une grande résistance mécanique. L'opération de forgeage applique deux types de chargement aux poteaux de la presse: une traction uniaxiale et une charge transversale qui crée de la flexion (Fig. 1). La combinaison de ces deux chargements engendre de grandes déformations à la base des poteaux qui sont encastrés dans le sol. Par conséquent, c'est dans cette zone que le niveau de contrainte sera le plus élevé.



The 5000T press

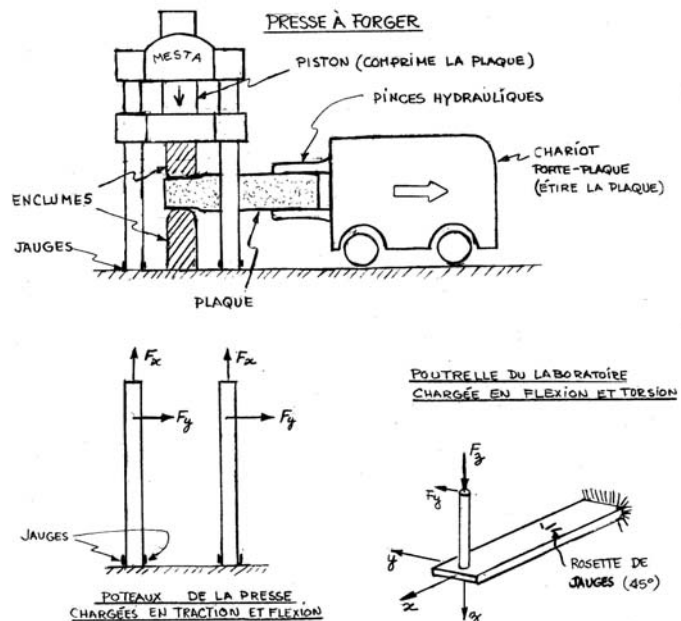


Figure 1 – À droite, la presse Mesta 5000 Tonnes de Slater Steel¹ à Sorel. À gauche, les modes de chargement des poteaux de la presse et de la poutrelle.

¹ Slater Steel de Sorel a fermé ses portes en 2003 suite à la faillite de sa maison mère aux États-Unis.

Les poteaux sont soumis à des chargements cycliques qui seront appliqués des millions de fois durant la vie de la presse. Pour s'assurer d'une grande longévité des poteaux, il faut éviter que leur niveau de contrainte ne dépasse la limite d'endurance du matériau (S_e)². Afin de mesurer la contrainte maximale dans les poteaux, des jauges de déformation sont collées à leur base. Le programme de surveillance à concevoir doit faire, en temps réel, la lecture des jauges de déformation et le calcul de la contrainte. Pour aider l'opérateur à éviter le dépassement de la limite d'endurance, la contrainte maximale est affichée dans un graphe déroulant, tandis que les charges de traction et de flexion, appliquées aux poteaux sont affichées dans des indicateurs individuels. Si la limite d'endurance est dépassée, le programme déclenche des alarmes visuelle et sonore pour avertir l'opérateur. Finalement, la plus haute contrainte maximale atteinte durant chaque cycle de forgeage est enregistrée dans un fichier³.

2. Simulation des poteaux avec une poutrelle encastrée

Comme vous vous en doutez bien, nous n'avons pas, à notre disposition, une presse de 5000 tonnes. Nous allons plutôt la remplacer par une simple poutrelle d'acier qui est encastrée sur le bord d'une table (Fig. 2). Bien que la poutrelle soit différente des poteaux de la presse, on peut cependant affirmer que les programmes de surveillance de la déformation seraient similaires dans les deux cas. En effet, la poutrelle subit des charges combinées de flexion et de torsion, ce qui ressemble au cas des poteaux qui subissent eux aussi deux chargements combinés, un en flexion et l'autre en traction. De plus, pour mesurer la contrainte maximale, on a recours à des jauges de déformation dans les deux cas. On peut donc conclure que le programme que vous allez concevoir pour la poutrelle, pourrait aussi servir dans le cas de la presse.

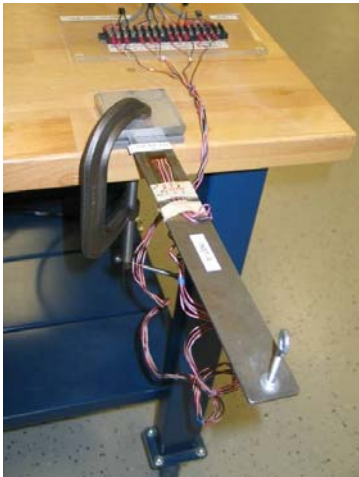


Figure 2 – Poutrelle fixée à la table

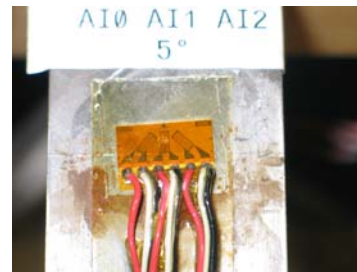
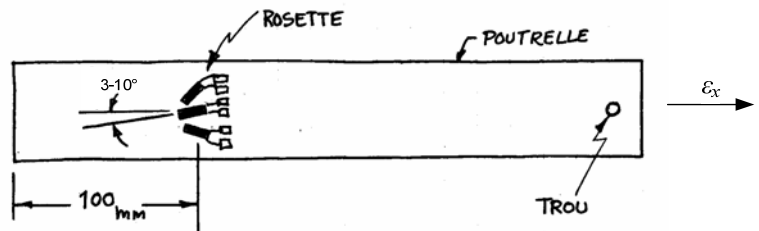


Figure 3 – Rosette 45° collée sur le dessus de la poutrelle

Poutrelle.

La poutrelle qui est utilisée pour le TP-4 est montrée dans la figure 2. Elle est encastrée à une extrémité et son chargement est appliqué manuellement à l'autre extrémité à l'aide d'une force verticale

² La limite d'endurance S_e , est la contrainte au-dessous de laquelle aucun dommage n'est fait au matériau car aucune amorce de fissure ne se crée.

³ Ce fichier pourrait servir aux ingénieurs à évaluer le dommage en fatigue des poteaux (ne fait pas partie du TP-4)

(flexion) et d'un couple (torsion). La poutrelle est fabriquée à partir d'une plaque d'acier dont les dimensions et les propriétés élastiques sont:

3,2 mm x 38 mm x 380 mm de long (dimensions exactes à mesurer au labo)
E = 200000 N/mm² Module de Young
v = 0,29 Coefficient de Poisson

Rosette

Une rosette 45° munie de trois jauges, est collée sur le dessus de la poutrelle à une distance d'environ 100 mm du bout qui est encastré (Fig. 3). Chaque jauge de la rosette est branchée dans un pont de Wheatstone où elle est montée en ¼ de pont. Pour les trois jauges, le voltage d'alimentation du pont et le facteur de jauge S_G sont:

V = 2,50 Volts
 $S_G = 2,10$

La jauge centrale de la rosette fait un petit angle (α) avec l'axe longitudinal de la poutrelle (axe x). Cet angle qui est indiqué sur la poutrelle (Fig. 3), sert au calcul des déformations ϵ_x et γ_{xy} dont on a besoin pour évaluer les moments de flexion et de torsion appliqués à poutrelle.

3. Cahier de charge du programme de surveillance de la poutrelle

Le programme LabVIEW que vous allez concevoir doit réaliser les opérations suivantes:

- a) Lire les signaux de sortie des trois ponts⁴ de jauges et les convertir en déformations. La vitesse de lecture et le nombre d'échantillons doivent être modifiables par l'utilisateur.
- b) Mettre en mémoire les "déformations initiales" des jauges à charge nulle afin de les soustraire des mesures subséquentes (équivalent à faire l'équilibrage initial du pont). La mesure des déformations initiales s'effectue à la demande de l'utilisateur.
- c) Calculer et afficher les déformations principales (ϵ_1, ϵ_2) et les contraintes principales (σ_1, σ_2) qui existent au niveau de la rosette.
- d) Calculer les moments de flexion et torsion appliqués à poutrelle et les afficher avec des indicateurs qui ont un grand impact visuel
- e) Afficher la contrainte maximale, σ_1 , dans un graphique déroulant et aussi dans un indicateur à grand impact visuel
- f) Répéter les mesures à un rythme qui peut être réglé par l'utilisateur (de 10 et 100 fois par seconde)
- g) Déclencher une alarme visuelle et sonore dès que la contrainte dépasse la limite d'endurance (S_e réglable). L'alarme doit s'arrêter automatiquement si la contrainte redescend sous la limite d'endurance ou si on arrête le programme. L'amplitude et le son de l'alarme sonore doivent être réglables par l'utilisateur. L'alarme sonore sera produite par un haut-parleur externe à partir d'un signal analogique de tension qui est généré par la carte d'acquisition de données.
- h) Enregistrer dans un fichier la plus haute contrainte atteinte durant chaque cycle de chargement de la poutrelle. La date et l'heure sont aussi enregistrées avec la contrainte.

⁴ Chacune des trois jauges de la rosette est montée en ¼ de pont.

Un exemple de face-avant pour un programme de surveillance est illustré à la figure 4.

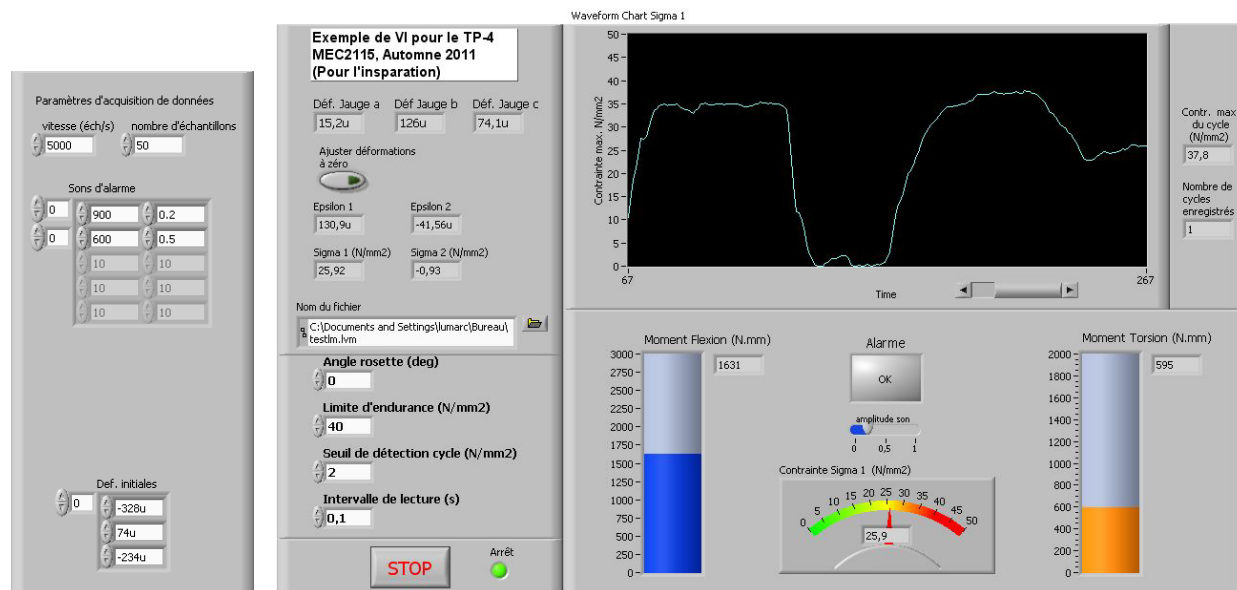


Figure 4 – Exemple de programme de surveillance des déformations de la poutrelle

4. Mesure et calcul des déformations des jauges

Chacune des jauges de la rosette est branchée dans un pont Wheatstone qui est situé dans le module de conditionnement pour jauges SC-2043-SG. Le signal de sortie des ponts est amplifié par un facteur de 10 par le module de conditionnement, avant d'être envoyé à la carte d'acquisition de données NI PCI-6221 qui se trouve dans l'ordinateur. Il faut donc tenir compte de cette amplification lors du calcul de la déformation de jauges.

Les ponts de Wheatstone sont reliés aux trois voies (canaux) suivantes de la carte d'acquisition de données:

- Jauge a: ai0
- Jauge b: ai1
- Jauge c: ai2



Pour faire la lecture des signaux de sortie des trois ponts, on fait appel au VI-Express "DAQ Assistant" qui est configuré pour acquérir des signaux de tension. Celui-ci utilise aussi une échelle personnalisée pour convertir les signaux des ponts en déformations. La création et la configuration du "DAQ Assistant" et de son échelle personnalisée sont expliquées en détails dans l'Annexe 1.

Au laboratoire, les déformations de la poutrelle vont varier lentement lorsqu'elle sera chargée manuellement pour simuler les opérations de forgeage. Pour réduire l'effet du bruit dans les signaux, nous allons recourir à une technique simple d'acquisition qui consiste à faire la moyenne de plusieurs échantillons mesurés dans un temps très court. Ainsi, dans l'Annexe 1, nous configurons le DAQ Assistant pour lire 50 échantillons par voie à une fréquence de 5000 Hz (éch./s), ce qui implique que la lecture des 50 échantillons s'effectue en 1/100 de seconde, un temps très court par rapport à la vitesse d'évolution des déformations de la poutrelle.

La figure 5 montre un exemple de sous-VI qui mesure et calcule la déformation moyenne de chacune des jauges de la rosette. La partie gauche du diagramme comprend le VI-Express "DAQ Assistant" qui fait la mesure des déformations (n échantillons), et le VI-Express "Statistics" qui calcule la déformation moyenne de chaque jauge. On note aussi que les données dynamiques sortant de "Statistics" sont converties en données de type tableau 1D avec le VI-Express "Convert from Dynamic Data". Cette conversion est nécessaire pour la suite des opérations du sous-VI qui sont expliquées plus bas.

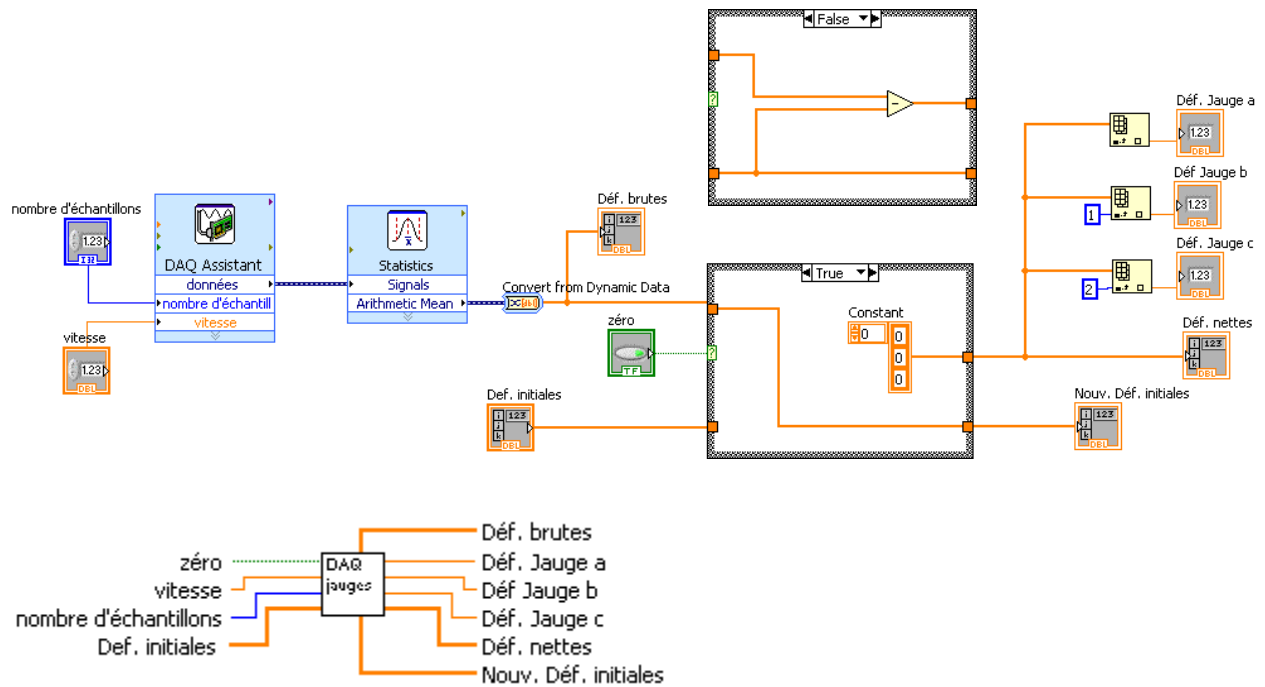


Figure 5 – Exemple d'un sous-VI qui mesure et calcule la déformation réelle moyenne de chaque jauge de la rosette. En bas, on peut voir les terminaux d'entrée et de sortie du sous-VI.

Mise en mémoire des déformations initiales et calcul des déformations nettes

Au départ, lorsque la poutrelle n'est pas chargée, les signaux de sortie des ponts de Wheatstone ne sont pas nuls, ce qui fait que l'on calcule une déformation initiale qui n'est pas réelle puisque la poutrelle n'est pas chargée. Pour éliminer cette déformation, nous allons faire appel à une astuce qui consiste à mettre en mémoire les déformations initiales (à charge nulle) pour ensuite les soustraire des déformations mesurées afin d'obtenir les déformations nettes (déformations réelles), celles qui sont engendrées par le chargement appliqué sur la poutrelle. Dans la figure 5, pour enregistrer les déformations initiales, on utilise une structure "Condition" qui peut soit mettre en mémoire les déformations initiales, ou bien les soustraire des déformations mesurées, selon la valeur de la commande booléenne à armement "zéro".

5. Calcul des déformations et des contraintes principales

Le calcul des déformations et des contraintes principales se fait à partir des formules de la rosette 45° que l'on retrouve dans le tableau 5.6 du recueil des notes de cours. Les deux valeurs initialement calculées sont celles du centre et du rayon du cercle de Mohr des déformations:

$$d = (\varepsilon_a + \varepsilon_c) / 2$$

$$r = [(\varepsilon_a - \varepsilon_b)^2 / 2 + (\varepsilon_b - \varepsilon_c)^2 / 2]^{0.5}$$

Avec d et r, on calcule les déformations principales

$$\varepsilon_1 = d + r$$

$$\varepsilon_2 = d - r$$

À partir des déformations principales, on calcule les contraintes principales:

$$\sigma_1 = E / (1 - \nu^2) (\varepsilon_1 + \nu \varepsilon_2) \quad , \text{ où } \sigma_1 \text{ est la } \underline{\text{contrainte maximale}} \text{ à la position de la rosette}$$

$$\sigma_2 = E / (1 - \nu^2) (\varepsilon_2 + \nu \varepsilon_1)$$

Pour programmer des équations à plusieurs termes, utilisez le VI-Express "Formula" ou un "Formula Node". Vous pouvez exécuter dans un seul sous-VI, tous les calculs précédents. N'oubliez pas de valider les calculs faits par votre sous-VI.

6. Calcul des moments de flexion et de torsion

Pour calculer le moment de flexion et le moment de torsion qui sont appliqués à la poutrelle, nous avons besoin de connaître la valeur des déformations: ε_x et γ_{xy} . Ces déformations sont calculées dans le repère cartésien x-y-z de la poutrelle (figure ci-contre).

$$\varepsilon_x = (\varepsilon_{x'} + \varepsilon_{y'}) / 2 + (\varepsilon_{x'} - \varepsilon_{y'}) / 2 * \cos[2(45^\circ - \alpha)] + (\gamma_{x'y'} / 2) * \sin[2(45^\circ - \alpha)]$$

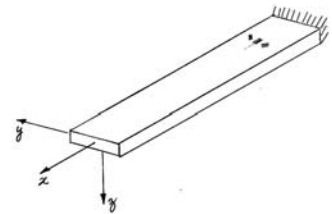
$$\gamma_{xy} = -(\varepsilon_{x'} - \varepsilon_{y'}) / 2 * \sin[2(45^\circ - \alpha)] + (\gamma_{x'y'} / 2) * \cos[2(45^\circ - \alpha)]$$

où $\varepsilon_{x'} = \varepsilon_a$

$\varepsilon_{y'} = \varepsilon_c$

$\gamma_{x'y'} / 2 = \varepsilon_b - (\varepsilon_a + \varepsilon_b) / 2$

α = angle entre la jauge centrale de la rosette (jauge b) et l'axe x de la poutrelle



L'angle α est marqué sur la poutrelle (Fig. 3). Le calcul des moments se fait à l'aide des équations suivantes:

$$\text{Moment de flexion}^5: \quad M_y = \varepsilon_x * (E * b * t^2) / 6$$

$$\text{Moment de torsion:} \quad T = (\gamma_{xy} / 2) * (E * b * t^2) / [3(1 + \nu)]$$

Vous pouvez regrouper dans un seul sous-VI, tous les calculs présentés dans cette section. L'Annexe 2 présente en détails le développement des équations précédentes.

⁵ Pour simplifier la présentation des résultats, nous ne tenons pas compte du signe algébrique de M_y .

7. Alarmes visuelle et sonore

Lorsque la contrainte maximale dans la poutrelle dépasse la limite d'endurance S_e , le programme de surveillance déclenche une alarme visuelle et une alarme sonore.

Alarme visuelle

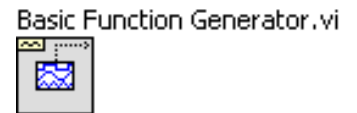
Pour l'alarme visuelle, on peut choisir un indicateur booléen (LED) qui sera grossi pour être bien visible. En plus, on peut faire clignoter le LED à l'aide d'un nœud de propriété en écriture.

Alarme sonore

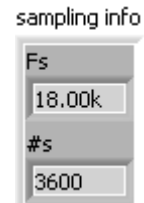
L'alarme sonore est produite avec un haut-parleur externe mais sans passer par la carte de son de l'ordinateur. Nous nous servons plutôt du générateur de signal de la carte d'acquisition de données NI PCI-6221. Plus précisément, la carte va générer une tension de sortie sur la voie "ao0" qui est branchée au haut-parleur.

Génération du son

La génération du son nécessite en premier lieu la création d'un signal numérique qui est ensuite envoyé au générateur de signal de la carte PCI-6221 qui le convertit en un signal analogique de tension qui est dirigé vers le haut-parleur. Le son de l'alarme peut comprendre une ou plusieurs notes. Chaque note est caractérisée par une fréquence, une durée et une intensité (amplitude). Une note est construite à l'aide d'une sinusoïde.



Pour générer le signal numérique d'une note⁶, utilisez le sous-VI "Basic Function Generator.vi" qui se trouve dans la sous-palette "Signal Processing" → "Wfm Generation". Vous devez fournir à ce sous-VI la fréquence et l'amplitude du son. Quant à la durée qui correspond au nombre de cycles de la sinusoïde, elle est définie par deux valeurs comprises dans le cluster "Sampling info" du sous-VI: la fréquence d'échantillonnage (Fs) et le nombre d'échantillons (#s). Par exemple, si je désire que chaque cycle de la sinusoïde soit composé de 20 points⁷, alors pour une note de fréquence 900 Hz et d'une durée de 0,2 seconde, nous aurons:



$$F_s = 20 \times 900 \text{ Hz} = 18 \text{ kHz}$$

$$\#s = 18 \text{ kHz} \times 0,2 \text{ s} = 3600 \text{ échantillons}$$

La figure 6 montre un exemple de sous-VI qui peut jouer une note sur un haut-parleur. On y remarque que l'amplitude, la fréquence et la durée de chaque note sont envoyées au sous-VI "Basic Function Generator.vi" qui produit un signal numérique de type "Waveform". Normalement, on dirige ensuite le signal numérique vers un VI-Express "DAQ Assistant" qui est configuré pour générer des signaux de sortie analogique en tension. Cependant, cette approche présente ici un inconvénient car la première fois que le VI-Express "DAQ Assistant" sera exécuté, il fixera de façon permanente les valeurs de "Fs" et de "#s" qui ne peuvent plus être changées par la suite, à moins d'arrêter le programme. Ceci nous empêche de générer correctement les signaux de notes différentes. Pour éliminer cette restriction, on transforme le VI-Express en un sous-VI auquel on apporte une modification mineure afin qu'il puisse accepter des valeurs différentes pour "Fs" et "#s". Ce sous-VI modifié apparaît sous le nom de "Sous-VI_DAQ_génération_variable.vi" dans la figure 6. La configuration du DAQ Assistant et sa transformation en sous-VI sont expliquées dans l'Annexe 3.

⁶ Pour générer les signaux d'une alarme à plusieurs notes, voir l'Annexe 3.

⁷ Pour une reproduction assez fidèle de la note, il est conseillé d'utiliser au moins 10 points par sinusoïde.

Pour vérifier le fonctionnement de votre alarme avec la carte de son d'un ordinateur, vous pouvez remplacer le sous-VI "DAQ_génération_variable" dans la figure 6, par le VI-Express "Play Waveform" qui se trouve dans la palette des Fonctions, sous-palette Programming → Graphics & Sounds → Sounds → Output



Durant le laboratoire, le volume des alarmes devra être réglé assez bas pour éviter une cacophonie dans la salle. Votre programme doit absolument contenir une commande qui contrôle l'amplitude du son et que l'utilisateur peut modifier en tout temps. Ne pas oublier que les alarmes visuelle et sonore doivent s'arrêter dès que la contrainte maximale descend sous la limite d'endurance.

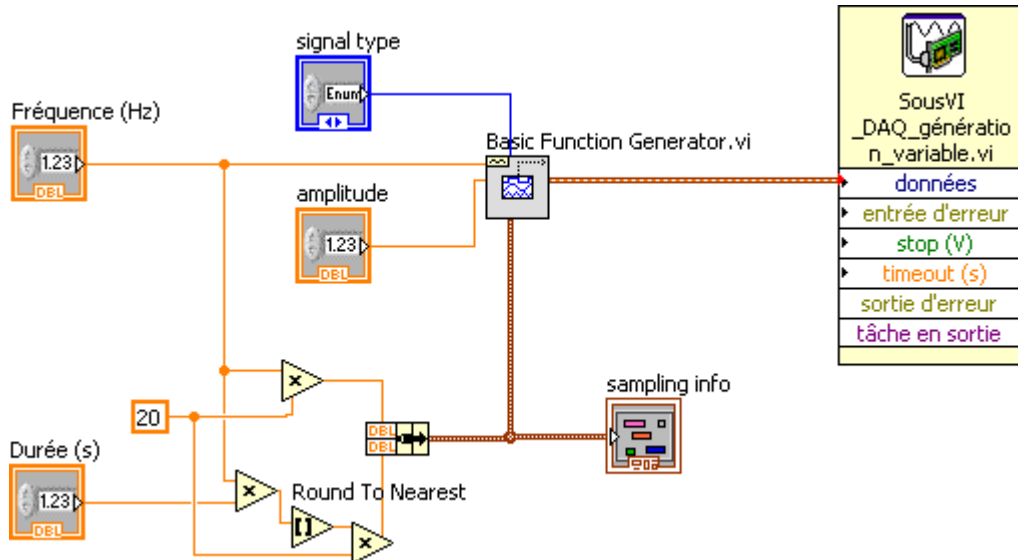


Figure 6 – Exemple d'un sous-VI qui génère le son d'une note avec une sinusoïde qui comprend 20 points. Voir le texte précédent pour une description de l'indicateur "sampling info".

8. Détection d'un cycle de chargement et enregistrement sur fichier

Le dernier item du cahier de charge du programme demande d'enregistrer dans un fichier texte la plus haute contrainte atteinte durant chaque cycle de chargement de la poutrelle. Pour y arriver, il faut mémoriser la plus haute contrainte d'un cycle et aussi détecter la fin d'un cycle de chargement.

La plus haute contrainte d'un cycle

On peut mémoriser la plus haute contrainte d'un cycle en se servant d'un registre à décalage dans la boucle While qui fait la lecture des jauges et le calcul de la contrainte maximale (voir Fig. 8). Durant l'exécution de la boucle, la valeur conservée dans le registre à décalage sera comparée à chaque nouvelle mesure de la contrainte maximale. Si celle-ci dépasse la valeur du registre, le registre est mis à jour avec la dernière mesure de contrainte. Lorsque la fin du cycle est détectée (voir ci-dessous) et qu'on sort de la boucle While, la valeur contenue dans le registre à décalage est alors la plus haute contrainte du cycle. Lorsqu'on débute un nouveau cycle de chargement, il faut remettre à zéro la valeur initiale du registre à décalage en branchant au tunnel d'entrée du registre, une constante égale à zéro.

Détection de la fin d'un cycle de chargement

Pour détecter la fin d'un cycle de chargement, on peut comparer la plus haute contrainte du cycle et la plus récente mesure de contrainte maximale, à une valeur de référence appelée "Seuil de détection" qui est une contrainte très basse⁸, par exemple 1 ou 2 N/mm². Cette comparaison se fait à l'intérieur de la boucle de lecture des jauges et de calcul de la contrainte. Elle peut être définie de la façon suivante:

[[Plus haute contrainte du cycle) > (Seuil de détection)] et [(Contrainte max.) < (Seuil de détection)]

Si vrai: le cycle est terminé et il faut sortir de la boucle de lecture pour enregistrer la plus haute contrainte du cycle dans un fichier

Si faux: continuer la lecture des jauges et le calcul de la contrainte maximale.

Enregistrement sur fichier

À la fin d'un cycle de chargement de la poutrelle, vous devez écrire sur une même ligne du fichier d'enregistrement, la date, l'heure et la plus haute contrainte du cycle. Une façon simple d'y parvenir consiste à utiliser la fonction "Get Date/Time String" pour obtenir la date et l'heure sous forme de texte (chaîne). Ensuite, il faut convertir la valeur numérique de la contrainte en texte (chaîne) avec la fonction "Number To Fractional String". Les trois chaînes obtenues sont combinées dans un tableau 1D qui est envoyé au sous-VI "Write To Spreadsheet File. vi". D'autres façons de réaliser l'enregistrement sur fichier sont aussi possibles. La figure 7 contient un exemple de sous-VI pour l'écriture sur fichier.

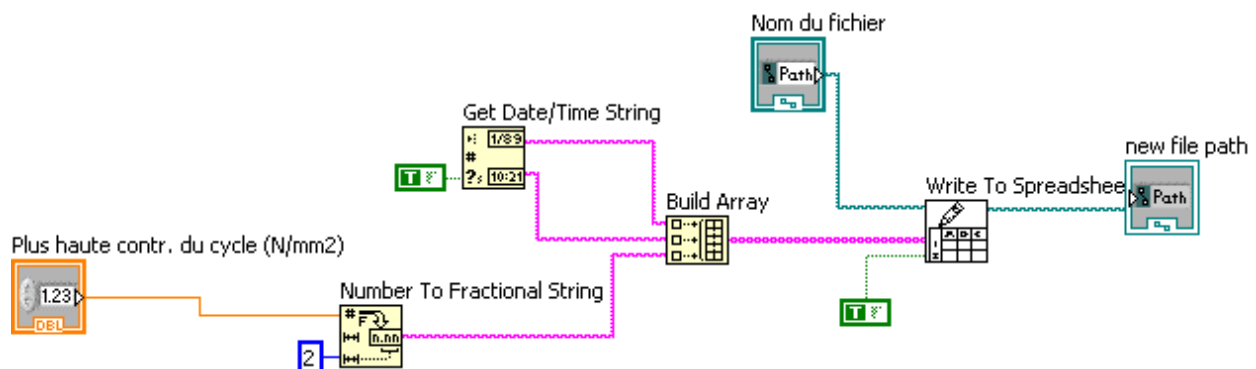


Figure 7 – Exemple de sous-VI pour l'écriture sur fichier

9. Structure générale du programme de surveillance

Ce qui suit présente une façon de réaliser le programme de surveillance. Ce n'est pas la seule façon de le faire et donc vous êtes entièrement libres de réaliser le programme à votre manière. L'important est qu'il réponde aux demandes du cahier de charge, qu'il soit votre création et qu'il fonctionne correctement.

Une façon simple de construire le diagramme du programme principal est d'avoir recours à **trois boucles While**. Une première boucle peut servir au contrôle l'alarme sonore. Les deux autres boucles While qui

⁸ Ne pas prendre zéro comme seuil de détection car il n'est pas certain que la contrainte descende jusqu'à ce niveau à la fin d'un cycle.

sont imbriquées l'une dans l'autre, forment le noyau du programme principal. La figure 8 présente l'arrangement général d'un tel diagramme.

Caractéristiques des boucles While

La boucle de contrôle de l'alarme sonore est indépendante du reste du programme car elle s'exécute en mode parallèle. Pour savoir quand démarrer ou arrêter l'alarme, la boucle fait continuellement la lecture de l'indicateur "Alarme" par le biais d'une variable locale. D'une manière similaire, la boucle lit aussi l'état de l'indicateur "arrêt" qui indique la fin de l'exécution du programme. La valeur de ces indicateurs est modifiée ailleurs dans les deux autres boucles.

La boucle de mesure des déformations des jauges exécute toutes les opérations importantes du programme:

- mesure des déformations des jauges,
- cadencement des mesures,
- calcul des déformations et des contraintes principales,
- calcul des moments de flexion et de torsion appliqués sur la poutrelle,
- affichage des résultats, incluant un graphique déroulant de la contrainte max.,
- lecture de toutes les commandes (sauf celles de l'alarme sonore)
- test de dépassement de la limite d'endurance,
- déclenchement ou arrêt de l'alarme,
- mise en mémoire de la plus haute contrainte d'un cycle,
- détection de la fin d'un cycle de chargement

La boucle d'enregistrement sur fichier englobe la boucle précédente (mesure des déformations). Sa fonction principale est d'enregistrer sur fichier la plus haute contrainte d'un cycle de chargement. Cette boucle sert également à remettre à zéro le registre à décalage de la boucle de mesure des déformations après chaque enregistrement. De plus, lorsqu'on arrête le programme avec le bouton stop, les valeurs de sortie de la boucle modifient l'état des indicateurs "Alarme" et "Arrêt" de façon à stopper correctement la boucle de contrôle de l'alarme sonore.

Dans la figure 8, à l'extrême gauche du diagramme, il y a une petite structure "Séquence" nommée Démarrage. Celle-ci a pour fonction d'éviter un arrêt accidentel de la boucle de l'alarme sonore lors du démarrage du programme. En ajustant à FAUX l'indicateur "Arrêt" avant que les boucles ne commencent à s'exécuter, on élimine cette possibilité.

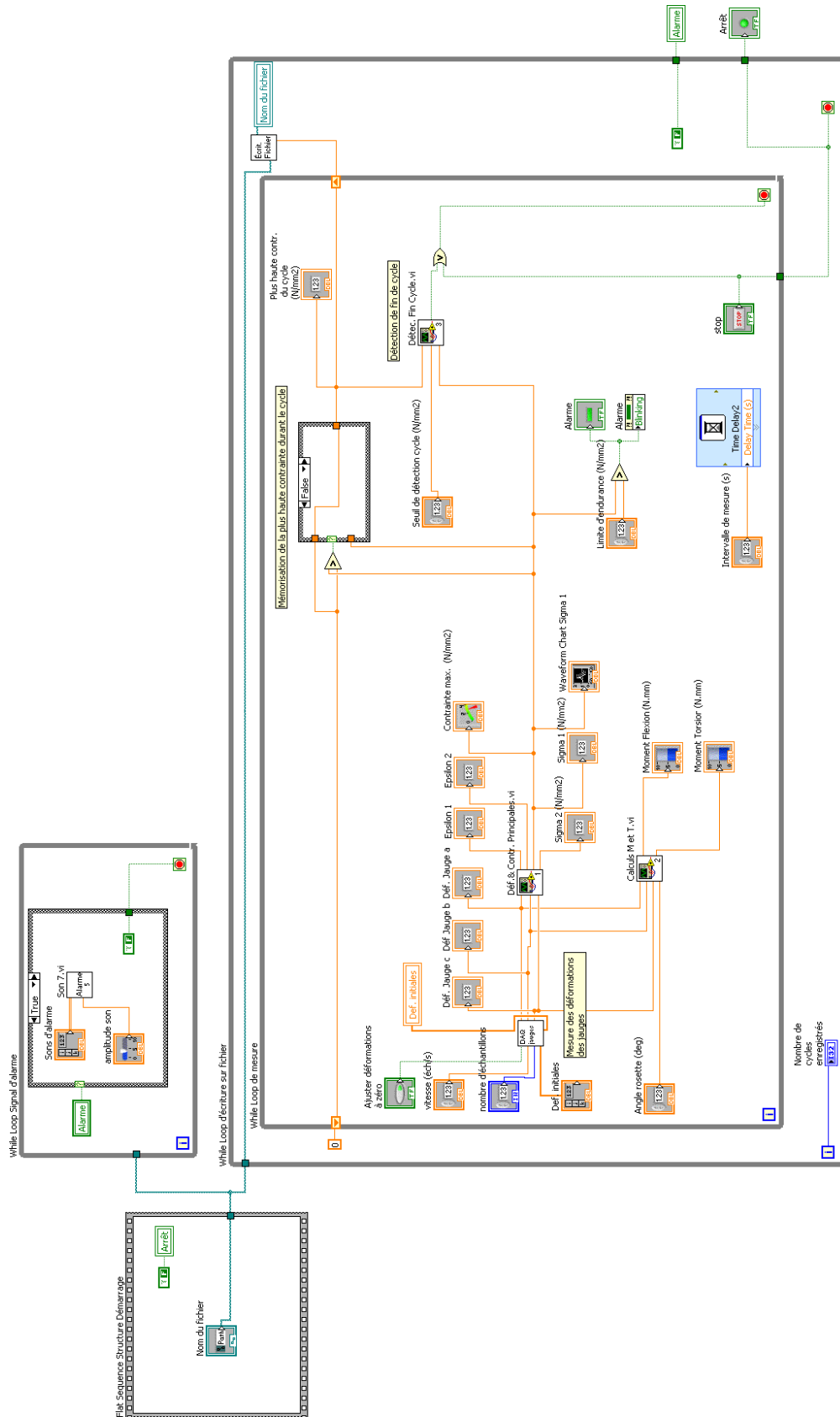


Figure 8 – Exemple de diagramme pour un programme de surveillance

10. Résumé du travail à faire

L'ensemble des fonctionnalités que doit contenir le programme LabVIEW de surveillance est résumé dans le cahier de charge présenté dans la section 3. Normalement, votre programme devrait contenir toutes ces fonctionnalités. Cependant, lors de la correction, seules celles qui fonctionnent pourront être évaluées. Il est donc important de suivre une méthode systématique pour développer votre programme.

Il est fortement conseillé de commencer le développement du programme en créant des sous-VI qui effectuent des tâches spécifiques, comme par exemple la mesure des déformations des jauges, le calcul des déformations principales, etc.. Plusieurs suggestions à cet effet ont été présentées dans les sections précédentes et quelques exercices vous ont été proposés à la fin des cours #2 et #3. Testez (et re-testez!) complètement chaque sous-VI pour vous assurer de son bon fonctionnement. Suivez aussi un ordre logique dans la mise au point des sous-VI. Par exemple, après le sous-VI qui calcule des déformations principales, vous pouvez concevoir le sous-VI qui fait le calcul des contraintes principales et le tester avec le précédent. Si tout fonctionne bien, vous pouvez même les combiner ensemble dans un seul sous-VI. Par ailleurs, l'utilisation de sous-VI performants dans le diagramme du programme principal va réduire considérablement sa complexité et accélérer sa mise au point.

Comment tester le programme sans la carte d'acquisition de données?

Même si l'ordinateur sur lequel vous allez préparer votre programme n'est pas équipé de la carte d'acquisition de données NI PCI-6221, vous pouvez quand même tester tout VI qui contient un VI-Express "DAQ Assistant", à l'aide de deux méthodes de simulation. Premièrement, vous pouvez créer au préalable un périphérique DAQmx simulé et configurer le "DAQ Assistant" avec ce périphérique. Si vous faites de l'acquisition de données (lecture de signaux), alors le périphérique simulé vous retourne un signal sinusoïdal dont l'amplitude est égale à la gamme d'entrée définie dans le "DAQ Assistant". Si vous faites de la génération de signal avec le périphérique simulé, alors il ne se passera rien. On peut aussi recourir à une deuxième méthode de simulation qui consiste à remplacer temporairement le VI-Express "DAQ Assistant". Par exemple, le "DAQ Assistant" qui fait la mesure des déformations des jauges peut être remplacé par le VI-Express "Simulate Signal" qui est capable de générer des échantillons numériques à la même fréquence que la carte d'acquisition de données. Dans le cas de l'alarme sonore, on peut remplacer le "DAQ Assistant" par le VI-Express "Play Waveform" qui va générer le son de l'alarme avec la carte de son de l'ordinateur (Voir section 7 p. 8).

11. Préparation et déroulement du laboratoire

Le temps de préparation du programme LabVIEW demandé dépasse très largement les 110 minutes que dure le laboratoire. Il est donc absolument nécessaire de vous préparer à l'avance et d'arriver au laboratoire avec un programme qui fonctionne déjà en mode simulation (sans carte d'acquisition de données).

Votre programme devra être présenté au professeur au début de la séance, une fois qu'il aura été installé sur l'ordinateur de votre poste de travail⁹. Le professeur pourra alors évaluer l'état de votre préparation. Par la suite, vous allez terminer la mise au point du programme en ajoutant et en configurant les VI-Express "DAQ Assistant" pour qu'ils puissent utiliser la "vraie" carte d'acquisition de données qui est montée à bord de l'ordinateur. Vous aurez environ 50 minutes compléter cette opération.

Vers la fin de la séance, chaque équipe fera une courte démonstration (2 minutes max.) du fonctionnement de son programme. Cette démonstration consistera à appliquer quelques cycles de chargement sur la poutrelle dont quelques-uns devront faire de déclencher l'alarme. Tous le monde pourra observer la démonstration. Cette démonstration complètera aussi l'évaluation de la préparation.

12. Évaluation et rapport

Le travail du TP-4 compte pour 9% de la note totale du cours. L'évaluation de la préparation qui se fera au laboratoire, comptera pour 3% de la note tandis que la correction du programme et du rapport comptera pour 6% de la note.

La correction du programme portera sur les points suivants: la face-avant (ergonomie, clarté, présence des éléments requis), l'organisation du diagramme et la structure générale du programme.

Rapport à faire

Chaque équipe écrira un bref rapport d'une longueur maximale de 4 pages, en excluant la page titre. Le rapport sera remis uniquement sous forme papier au professeur (Luc Marchand, A-103). Le rapport devra contenir les éléments suivants:

- La page titre disponible sur le site Moodle (format Word)
- Une description de la façon dont vous avez conçu la face-avant et structuré le programme principal
- Une description de la façon dont vous avez conçu et testé les sous-VI
- Une description des essais de simulation qui ont été faits pour valider tout le programme
- Un court commentaire sur de l'organisation initiale du travail d'équipe et sur la façon dont le travail a été réellement accompli (5 ou 6 lignes)
- Vos descriptions doivent s'appuyer sur des figures qui montrent en partie ou complètement, des faces-avant de VI, des diagrammes, ou tout autre élément pertinent comme, par exemple, une fenêtre de configuration. Ces figures peuvent être des captures d'écran.

Fait le 14 octobre 2011
Luc Marchand, Ph. D, ing.
Professeur

⁹ Le programme principal et tous ses sous-VI seront placés dans un seul répertoire qui sera localisé sur le bureau local de l'ordinateur. Le nom du répertoire doit être obligatoirement: {No. d'équipe}_ {Nom de famille d'un coéquipier}_ {no. de poste de travail}_ MEC2115. À la fin de la séance, on vous indiquera comment transférer ce répertoire au professeur.

Annexe 1 – Configuration du DAQ Assistant pour la lecture des jauges (¼ de pont)

Le VI-Express " DAQ Assistant" est initialement créé¹⁰ avec les paramètres suivants:

Acquérir des signaux → Entrée analogique → Tension.

Voies physiques supportées: Dev1(PCI-6221) → choisir ai0, ai1 et ai2

Par la suite, il est configuré de la façon suivante (Fig. 9):

Paramètres des voies

Renommer les trois voies (touche F2 du clavier): Jauge a, Jauge b et Jauge c

Configuration de la tension d'entrée

Configuration du terminal: NRSE

Paramètres de cadencement

Mode d'acquisition: N échantillons

Échantillons à lire : 50

Fréquence (Hz): 5000 (éch./s)¹¹

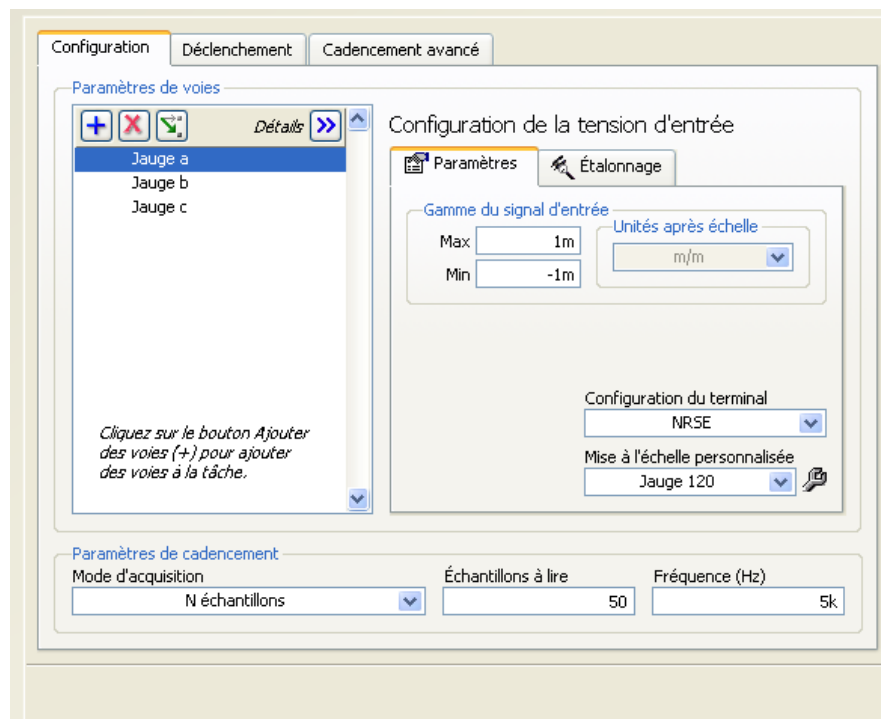


Figure 9 - Fenêtre de configuration du DAQ Assistant

¹⁰ Le VI-Express DAQ Assistant se trouve dans la palette de fonctions "Express", sous-palette "Input"

¹¹ En fait, puisqu'il y a trois voies à lire et que la carte d'acquisition ne possède qu'un seul convertisseur analogue-numérique, la fréquence d'échantillonnage réelle de la carte est ajustée automatiquement par LabVIEW à 3 x 5000 Hz, soit 15000 Hz. À cette fréquence, il est possible de lire les 50 échantillons des trois voies dans un temps total de 1/100 de seconde.

Création d'une échelle personnalisée pour obtenir directement les déformations

Cette échelle est basée sur l'équation du ¼ de pont :

$$\Delta E_m = V/4 * S_G * \varepsilon$$

Le voltage lu par la carte d'acquisition est le signal de sortie du pont amplifié par 10X. Si on nomme ce voltage $\Delta E_m'$, l'équation du pont devient:

$$\Delta E_m = \Delta E_m'/10 = V/4 * S_G * \varepsilon.$$

En réarrangeant,

$$\varepsilon = (\Delta E_m' * 4) / (10 * V * S_G)$$

La déformation est donc en relation linéaire avec le voltage lu ce qui permet d'utiliser une échelle linéaire pour convertir le voltage en déformation.

Pour créer une échelle personnalisée dans le DAQ Assistant, choisir l'option **Créer un nouvel objet** dans la commande **Mise à l'échelle personnalisée**. Nommez l'échelle "Jauge 120". Choisir ensuite une échelle linéaire dont la pente est égale à $4/(10 * V * S_G)$, où $V=2,50$ Volts et $S_G=2,10$. L'ordonnée à l'origine de la courbe est nulle (Fig. 10). Les unités après échelle sont des m/m. Une fois créé, cette échelle peut servir pour convertir en déformation les signaux des trois ponts de jauge lus par le DAQ Assistant.

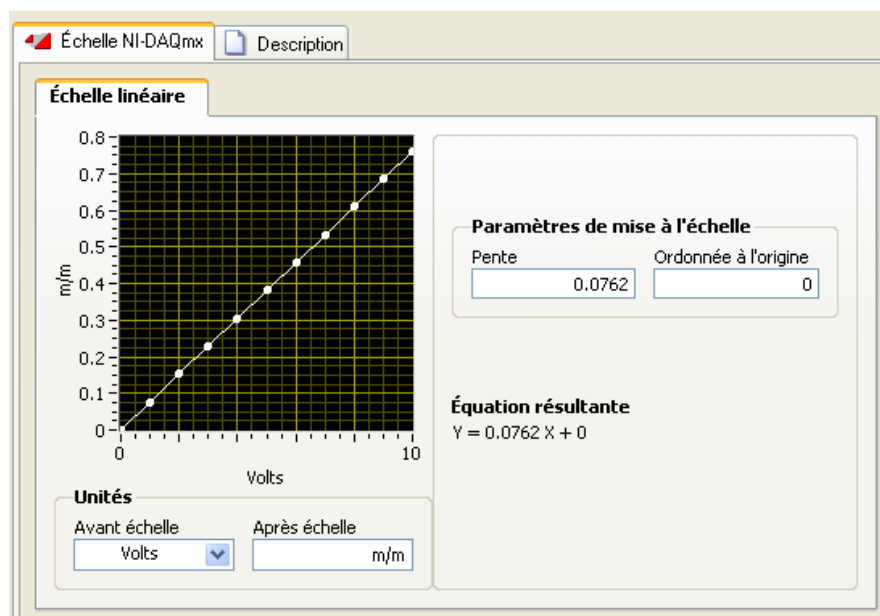


Figure 10 - Fenêtre de configuration de l'échelle linéaire

Une fois l'échelle personnalisée¹² sélectionnée dans la fenêtre de configuration du "DAQ Assistant", vous pouvez régler la "Gamme du signal d'entrée qui utilise les unités après échelle:

$$\text{Max} = 1000 \times 10^{-6} \text{ m/m}$$

$$\text{Min} = -1000 \times 10^{-6} \text{ m/m}$$

¹² L'échelle personnalisée de LabVIEW n'est pas intégrée dans le VI-Express mais est conservée dans l'environnement LabVIEW. Bien qu'on puisse faire migrer l'environnement d'un ordinateur vers un autre, il est plus simple et moins risqué de recréer l'échelle quand on change d'ordinateur.

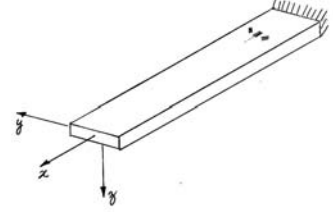
Annexe 2 – Équations pour le calcul des moments de flexion et de torsion dans la poutrelle

Dans ce qui suit, nous utiliserons le système d'axes cartésiens de la poutrelle (x-y-z). Lorsqu'une poutrelle est soumise à un moment de flexion M_y , la seule contrainte qui est générée est une contrainte axiale σ_x , toutes les autres contraintes sont nulles ($\sigma_y = \sigma_z = 0$). En se basant sur la loi de Hooke, on peut donc écrire que :

$$\varepsilon_x = (1/E)[\sigma_x - \nu(\sigma_y + \sigma_z)] = \sigma_x/E$$

et

$$\sigma_x = -M_y * c / I = 6M_y / (b * t^2) \quad \text{où } t \text{ est l'épaisseur et } b \text{ la largeur de la poutrelle}$$



En combinant les équations précédentes, on trouve que:

$$M_y = -\varepsilon_x * (E * b * t^2) / 6$$

Pour une poutrelle soumise à un moment de torsion T (M_x), la seule contrainte générée est une contrainte de cisaillement τ_{xy} qui est égale à:

$$\tau_{xy} = T * t / J = 3T / (b * t^2)$$

Sachant que

$$\tau_{xy} = \gamma_{xy} * G = (\gamma_{xy} * E) / [2(1+\nu)]$$

Donc,

$$T = (\gamma_{xy} / 2) * (E * b * t^2) / [3(1+\nu)]$$

Comme on vient de le démontrer, pour calculer le moment de flexion et le moment de torsion, il suffit de connaître les déformations ε_x et γ_{xy} . À partir des déformations mesurées avec la rosette et connaissant l'angle α que fait la jauge b avec l'axe x de la poutrelle, on peut calculer ε_x et γ_{xy} en se servant des équations générales concernant les déformations dans un plan. Pour cela, commençons par définir, pour la rosette, un système local d'axes cartésiens $x'-y'$ où :

$$\varepsilon_{x'} = \varepsilon_a \text{ et } \varepsilon_{y'} = \varepsilon_c$$

Pour la jauge ε_b placée à $+45^\circ$ de l'axe x' ,

$$\varepsilon_b = (\varepsilon_{x'} + \varepsilon_{y'}) / 2 + (\varepsilon_{x'} - \varepsilon_{y'}) / 2 * \cos(2 * 45^\circ) + \gamma_{x'y'} / 2 * \sin(2 * 45^\circ)$$

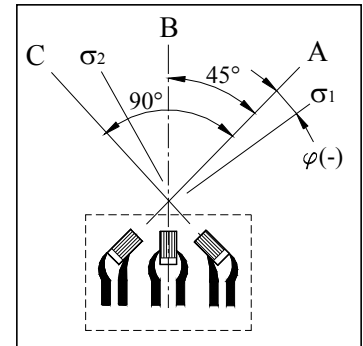
Après réarrangement et substitution, on trouve que

$$\gamma_{x'y'} / 2 = \varepsilon_b - (\varepsilon_a + \varepsilon_c) / 2$$

Connaissant $\varepsilon_{x'}$, $\varepsilon_{y'}$ et $\gamma_{x'y'}$, on peut maintenant obtenir les déformations ε_x et γ_{xy} dont on a besoin pour calculer M_y et T :

$$\varepsilon_x = (\varepsilon_{x'} + \varepsilon_{y'}) / 2 + (\varepsilon_{x'} - \varepsilon_{y'}) / 2 * \cos[2(45^\circ - \alpha)] + (\gamma_{x'y'} / 2) * \sin[2(45^\circ - \alpha)]$$

$$\gamma_{xy} = -(\varepsilon_{x'} - \varepsilon_{y'}) / 2 * \sin[2(45^\circ - \alpha)] + (\gamma_{x'y'} / 2) * \cos[2(45^\circ - \alpha)]$$



Annexe 3 – Configuration et modification du DAQ Assistant pour générer le signal de l'alarme

Pour produire le son de l'alarme avec un haut-parleur, nous générons un signal analogique de tension sur la voie ao0 de la carte NI PCI-6221. Pour y parvenir, il faut commencer par créer un VI-Express "DAQ Assistant" avec les paramètres suivants:

Générer des signaux → Sortie analogique → Tension
Voies physiques supportées: Dev1(PCI-6221) → ao0

La configuration par défaut qui apparaît dans la figure 11, doit être utilisée. Si vous le désirez, vous pouvez renommer la voie TensionSortie en appuyant sur la touche F2 du clavier.

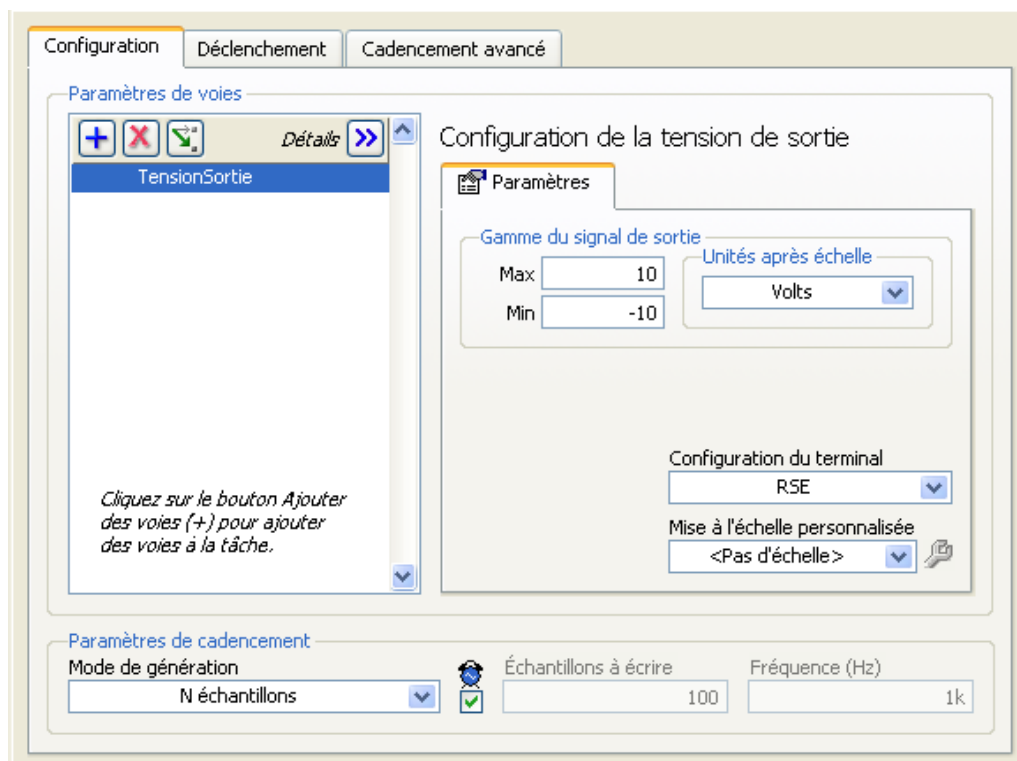


Figure 11 – Fenêtre de configuration d'un "DAQ Assistant" pour générer un signal (alarme)

Le VI-Express créé doit maintenant être transformé en sous-VI pour lui permettre de modifier la fréquence d'échantillonnage (F_s) et nombre d'échantillons (#s) des signaux qu'il génère sans avoir à arrêter le programme. Dans le menu local du VI-Express, choisir la commande "Open Front Panel", puis répondre positivement à la question que pose LabVIEW pour confirmer la conversion en sous-VI. Immédiatement, la couleur de fond du VI passe du bleu au jaune pour indiquer le changement. Ouvrir ensuite le diagramme du sous-VI et, en comptant à partir de la gauche du diagramme, repérer la cinquième structure condition (Voir figure 12). **Enlevez la fonction "First Call?" du terminal d'entrée et remplacez-la par une constante booléenne VRAI.** La figure 12, illustre la modification à faire. Sauvegardez votre sous-VI modifié sous un nom différent de celui qui est proposé par défaut (Untitled 1).

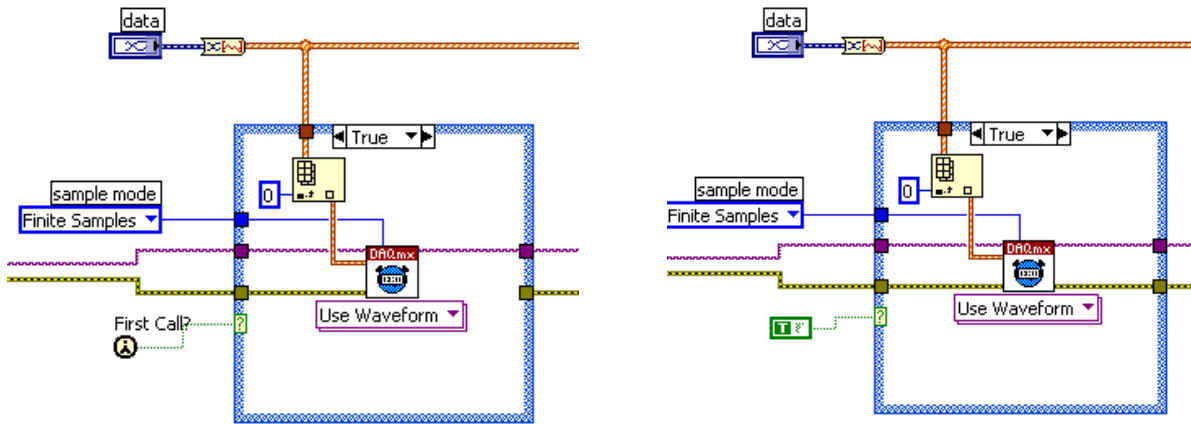


Figure 12 – Modification de la 5^{ème} structure condition du sous-VI " DAQ Assistant". À gauche, avant la modification et à droite après la modification. Qu'est ce qui a changé?

Faire jouer une alarme qui a plusieurs notes

Si vous utilisez plus d'une note dans votre alarme, il est conseillé de définir la fréquence et la durée de chaque note dans un tableau 2D (voir ci-contre). Chaque rangée du tableau sert à produire une note spécifique. Pour générer les signaux de ces notes, on commence par convertir le tableau 2D en deux tableaux 1D: un premier pour la fréquence et un deuxième pour la durée. On lit ensuite les éléments des deux tableaux avec une boucle FOR qui comprend une entrée autoindexée pour chaque tableau (voir Fig. 13). On peut aussi remplacer le tableau 2D par deux tableaux 1D, un pour la fréquence et un autre pour la durée. Cette deuxième méthode est plus facile à programmer, mais elle présente l'inconvénient d'avoir à placer deux tableaux sur la face-avant au lieu d'un seul.

Sons d'alarme		
0	900	0.2
0	600	0.5
	10	10
	10	10
	10	10

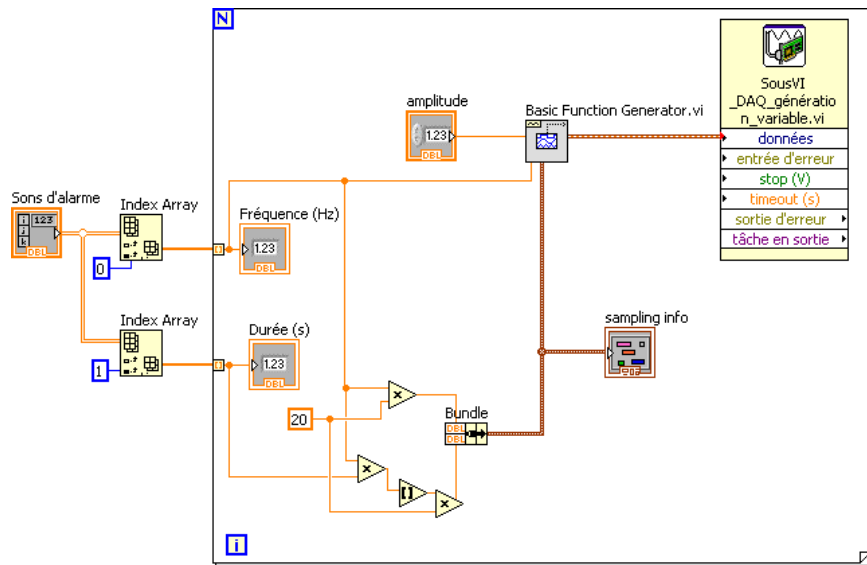


Figure 13 - Exemple d'un sous-VI qui génère le son de plusieurs notes