

INITIATION AUX VIs POINT PAR POINT DE LABVIEW™

Contenu

Utilisation des bibliothèques de VIs point par point (PPP) de LabVIEW	2
Initialiser les VIs point par point	3
Les questions les plus fréquemment posées	6
Quelles sont les différences entre l'analyse point par point et l'analyse sur tableaux dans LabVIEW ?	6
Quelle est l'utilité de l'analyse point par point ?	7
Quelles sont les nouveautés apportées par l'analyse point par point ?	8
Quels sont les éléments familiers de l'analyse point par point ?	8
Comment est-il possible d'effectuer une analyse sans buffers de données ?	8
En quoi l'analyse point par point est-elle plus efficace dans les applications temps réel ?	9
Critères d'utilisation de l'analyse point par point	10
Quelle est l'importance à long terme de l'analyse point par point ?	10
Exemples	10
Filtre point par point et sur tableaux de points	10
Exemple de spectre d'amplitude temps réel	12
Histogramme mobile PPP (Moving Histogram PtByPt.vi)	13
Étude de cas de l'analyse point par point	13
Analyse point par point de roues de train	13
Aperçu de la solution point par point de LabVIEW	15
Caractéristiques d'un signal de roue de train	16
Paramètres de signaux dans le VI Roue de train PPP (Train Wheel PtByPt.vi)	17
Vitesse d'échantillonnage dans le VI Roue de train PPP (Train Wheel PtByPt.vi)	19
Exigences de filtrage du VI Roue de train PPP (Train Wheel PtByPt.vi)	19
Composantes utilisées pour l'analyse du VI Roue de train PPP (Train Wheel PtByPt.vi)	20
Composantes utilisées dans la gestion des événements du VI Roue de train PPP (Train Wheel PtByPt.vi)	20
Composantes utilisées pour le rapport du VI Roue de train PPP (Train Wheel PtByPt.vi)	21
Conclusion	22

LabVIEW™, National Instruments™, NI™ et ni.com™ sont des marques déposées de National Instruments Corporation. Les noms de produits et de sociétés cités sont des marques déposées par leurs propriétaires respectifs. Pour la liste des brevets protégeant les produits National Instruments, veuillez vous référer, selon le cas: à la rubrique **Aide»Brevets** de votre logiciel, au fichier `patents.txt` sur votre CD et/ou à ni.com/patents.

LabVIEW offre un ensemble de VIs pour l'analyse point par point. L'analyse point par point convient idéalement à l'acquisition de données temps réel. Lorsqu'un système d'acquisition de données requiert des performances déterministes temps réel, vous pouvez construire un programme qui utilise l'équivalent point par point des VIs d'analyse LabVIEW traditionnellement basés sur une analyse par tableaux de points.

Les performances temps réel sont aujourd'hui une réalité pour l'acquisition de données. Grâce à l'analyse point par point de LabVIEW, l'analyse de données peut, elle aussi, bénéficier de performances temps réel. Les étapes discrètes de l'analyse sur tableaux, telles que la préparation, l'analyse et la sortie de buffers, peuvent être trop lentes pour des systèmes temps réel, déterministes et très haute vitesse. Les systèmes temps réel nécessitent souvent une analyse continue effectuée pour chaque point de données, appelée point par point.

Ce document explique les concepts et certains détails de programmation de l'analyse de données point par point. Les VIs point par point de LabVIEW présentent les avantages suivants :

- Vous pouvez analyser des événements et y répondre en temps réel
- Le processus d'analyse est directement connecté au signal afin d'assurer une vitesse optimale et une perte minimale des données
- Les tâches de programmation sont facilitées puisque vous n'allouez pas de tableaux et que vous n'avez pas à vous soucier d'ajuster la vitesse d'échantillonnage
- L'analyse se synchronise automatiquement sur l'acquisition de données car vous travaillez instantanément avec un seul signal.

Utilisation des bibliothèques de VIs point par point (PPP) de LabVIEW

La structure des VIs point par point est un miroir des VIs d'analyse sur tableaux approprié à l'analyse de données en continu. Il existe cependant des différences de programmation. L'utilisation de VIs point par point permet généralement de limiter les tâches de programmation. Le tableau ci-après décrit les entrées et sorties caractéristiques d'un VI point par point dans LabVIEW.

Tableau 1. Entrées et sorties caractéristiques pour VIs point par point

Paramètre	Description
Entrée de données	Données entrantes
Sortie de données	Données sortantes, analysées

Tableau 1. Entrées et sorties caractéristiques pour VIs point par point

Paramètre	Description
Initialiser	Routine qui réinitialise l'état interne d'un VI
Longueur de l'échantillon	Paramètre définissant une portion de données représentative pour le système d'acquisition de données ou la fonction de calcul.

Reportez-vous à la section *Étude de cas de l'analyse point par point* pour étudier un exemple de système d'analyse point par point.

Initialiser les VIs point par point

Cette section explique dans quelles conditions utiliser le paramètre **initialiser** dans de nombreux VIs point par point. Cette section décrit également la fonction Premier appel ? de LabVIEW disponible dans LabVIEW 6.0 et dans les versions ultérieures.

Utilité de l'initialisation dans les VIs point par point

Grâce au paramètre **initialiser**, vous pouvez réinitialiser l'état interne des VIs sans interrompre le flux continu de données ou de calculs. Vous pouvez réinitialiser un VI pour répondre à différents événements, comme dans les cas suivants :

- Un utilisateur changeant la valeur d'un paramètre
- L'application générant un événement spécifique ou atteignant un seuil.

Par exemple, le VI Changement de valeur PPP situé dans la palette **Fonctions»Point par point»Autres fonctions PPP**, peut détecter des modifications sur une valeur d'entrée et agir de la manière suivante :

- Réception de la donnée d'entrée
- Détection du changement
- Génération d'une valeur VRAIE booléenne qui déclenche l'initialisation dans un autre VI
- Transfert de la donnée d'entrée vers un autre VI pour traitement.

La figure 1 montre le VI Changement de valeur PPP déclenchant l'initialisation dans un autre VI et transférant les données à ce VI. Dans ce cas précis, les données d'entrée sont un paramètre utilisé comme valeur pour le VI cible suivant.

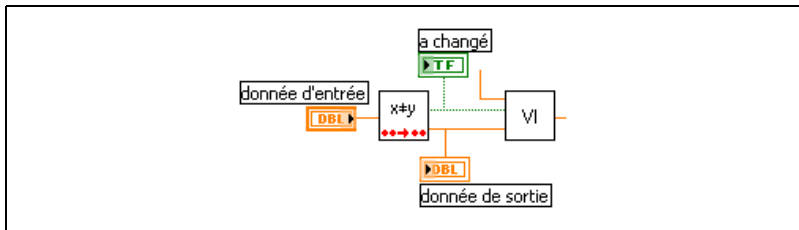


Figure 1. Rôle caractéristique du VI Changement de valeur PPP

Utilisation de la fonction Premier appel ?

De nombreuses applications point par point ne nécessitent pas l'utilisation du paramètre **initialiser** car l'initialisation se produit automatiquement à chaque fois qu'un opérateur quitte une application puis redémarre. Par exemple, certaines applications s'arrêtent toutes les 24 heures. La mise hors tension constitue une initialisation appropriée du système.

Les VIs point par point utilisent généralement la fonction Premier appel ? de LabVIEW. Dans un VI qui comprend la fonction Premier appel ?, l'état interne de ce VI est réinitialisé une seule fois, la première fois que vous l'appellez. La valeur de la sortie de la fonction Premier appel ? est toujours VRAI pour le premier appel du VI. La valeur passe à FAUX pour tous les appels ultérieurs du VI. Utilisez la fonction Premier appel ? située dans la palette **Fonctions»Avancé»Synchronisation** pour construire des VIs point par point. La figure 2 illustre une utilisation caractéristique de la fonction Premier appel ? avec une boucle While de LabVIEW.

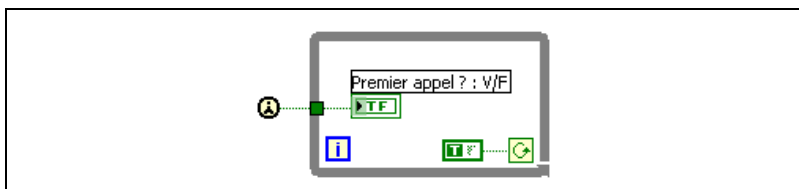


Figure 2. Utilisation de la fonction Premier appel ? dans une boucle While

Vérification d'erreur et initialisation

Les VIs point par point génèrent des codes d'erreur pour vous aider à identifier des défauts dans la configuration des applications que vous construisez. Plusieurs codes d'erreur point par point existent en plus des codes d'erreur LabVIEW standards.

Les codes d'erreur identifient généralement des paramètres et des configurations invalides. Pour une vérification d'erreur de niveau plus élevé, configurez votre programme afin de contrôler et de répondre à toute irrégularité dans l'acquisition de données et le calcul, et d'y remédier. Par exemple, une manière de contrôler les erreurs consiste à vérifier la gamme de vos données.

Un VI point par point ne générera un code d'erreur que lors de son premier appel ou de sa réinitialisation. De ce fait, il se comporte de façon optimale dans les applications déterministes temps réel.

Les VIs point par point génèrent un code d'erreur pour vous tenir informé de tout paramètre ou configuration invalide lorsqu'ils détectent une erreur durant le premier appel. Lors des appels ultérieurs, les VIs point par point mettent le code d'erreur à zéro et continuent à s'exécuter sans générer de code d'erreur. Vous pouvez programmer votre application pour choisir l'une des actions suivantes en réponse à la première erreur :

- Rapporter l'erreur et poursuivre l'exécution
- Rapporter l'erreur et arrêter
- Ignorer l'erreur et poursuivre l'exécution. Ceci est le comportement par défaut.

La séquence de programmation suivante décrit comment utiliser le VI Changement de valeur PPP afin de construire un mécanisme de vérification d'erreur point par point pour les VIs point par point disposant d'un paramètre **erreur**.

1. Sélectionnez un paramètre dont vous souhaitez surveiller de près les erreurs éventuelles
2. Connectez la valeur du paramètre en tant que **donnée d'entrée** au VI Changement de valeur PPP
3. Transférez le paramètre **donnée de sortie**, qui est toujours identique à **donnée d'entrée** vers le VI cible

4. Le VI Changement de valeur PPP retourne également une valeur VRAI chaque fois que la valeur du paramètre d'entrée change. Transmettez l'événement VRAI au VI cible afin de déclencher l'initialisation, comme l'illustre la figure 1
5. Lors du premier appel suivant cette initialisation, LabVIEW va contrôler l'état d'erreur. Cette boucle d'initialisation et de recherche d'erreur s'exécute à chaque fois que le paramètre d'entrée change.

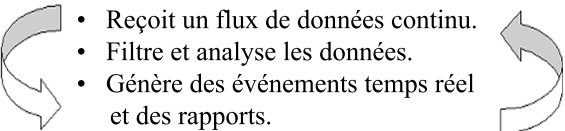
Les questions les plus fréquemment posées

Cette section répond aux questions les plus fréquemment posées sur l'analyse point par point.

Quelles sont les différences entre l'analyse point par point et l'analyse sur tableaux dans LabVIEW ?

Les tableaux 2 et 3 comparent l'analyse sur tableaux et l'analyse point par point sous des angles différents. Dans le tableau 2, les différences entre deux systèmes de distribution de combustible automobile, à carburateur ou à injection, soulignent les différences entre l'analyse sur tableaux et l'analyse point par point.

Tableau 2. Comparaison de méthodes traditionnelles et modernes

Méthode traditionnelle	Méthode moderne
Technologie automobile	
Carburateur <ul style="list-style-type: none"> • Le carburant s'accumule dans une cuve à niveau constant. • Le vide du moteur aspire le carburant à travers un ensemble unique de soupapes de dosage qui alimentent toutes les chambres de combustion. • Une combustion moyennement efficace se produit. 	Injection de carburant <ul style="list-style-type: none"> • Le carburant s'écoule du réservoir sans interruption. • Le carburant est vaporisé directement dans chaque chambre de combustion au moment de la combustion. • Une combustion précise et adaptée aux sollicitations du conducteur se produit.
Technologie d'analyse de données	
Analyse sur tableaux <ul style="list-style-type: none"> • Configurer un buffer. • Analyser les données. • Produire un buffer de données analysées. • Génère un rapport. 	Analyse point par point  <ul style="list-style-type: none"> • Reçoit un flux de données continu. • Filtre et analyse les données. • Génère des événements temps réel et des rapports.

Le tableau 3 présente d'autres points de comparaison entre l'analyse sur tableaux et l'analyse point par point.

Tableau 3. Comparaison entre l'analyse sur tableaux et l'analyse point par point

	Analyse sur tableaux	Acquisition de données et analyse avec les VIs point par point
Compatibilité	Compatibilité limitée avec les systèmes temps réel	Compatible avec les systèmes temps réel ; compatibilité descendante avec les systèmes sur tableaux
Types des données	Orienté tableau	Orienté scalaire
Interruptions	Interruptions critiques	Interruptions tolérées
Fonctionnement	Observation, hors ligne	Contrôle, en ligne
Performances et programmation	Compensation de la perte de données au démarrage (4–5 secondes) avec des structures de type “machines d'états” complexes	Pas de perte de données au démarrage ; initialise le système d'acquisition de données une fois et l'exécute en continu
Point de vue	Reflet d'un processus, tel un miroir	Flux naturel et direct d'un processus
Programmation	Spécifie un buffer	Pas de buffer explicite
Résultats	Génère un rapport	Génère un rapport et un événement en temps réel
Comportement de l'exécution	Traitement différé	Temps réel
Comportement de l'exécution	Arrêt	Poursuite
Comportement de l'exécution	Attente	Instantané
Mode de travail	Asynchrone	Synchrone

Quelle est l'utilité de l'analyse point par point ?

L'analyse point par point fonctionne parfaitement avec l'acquisition de données temps réel sur ordinateur. Dans l'analyse sur tableaux, le processus entrée-analyse-sortie est effectué sur des sous-ensembles d'un ensemble de données plus important. Dans l'analyse point par point, le processus entrée-analyse-sortie se déroule de manière continue et en temps réel.

Quelles sont les nouveautés apportées par l'analyse point par point ?

Lorsque vous effectuez une analyse point par point, gardez à l'esprit les concepts suivants :

- Initialisation : vous devez initialiser l'application d'analyse point par point pour éviter toute perturbation provenant des paramétrages que vous avez effectués lors de sessions précédentes d'analyse de données
- Exécution réentrante : vous devez activer l'exécution réentrante des VIs pour l'analyse point par point. L'exécution réentrante alloue une mémoire déterminée à un processus d'analyse unique et garantit ainsi que deux processus utilisant la même fonction d'analyse n'interféreront jamais les uns avec les autres



Remarque Si vous créez des VIs personnalisés pour les utiliser dans votre propre application point par point, assurez-vous d'activer l'exécution réentrante. L'exécution réentrante est activée par défaut dans presque tous les VIs point par point.

- Performances déterministes : l'analyse point par point va fréquemment de pair avec de nombreux systèmes déterministes, car elle s'intègre efficacement au flux d'un signal de données temps réel.

Quels sont les éléments familiers de l'analyse point par point ?

L'analyse point par point de LabVIEW est familière car l'approche pour la plupart des opérations d'analyse reste inchangée. Vous utilisez les filtres, l'intégration, les algorithmes de valeur moyenne, etc., dans les mêmes situations et pour les mêmes raisons que dans l'analyse de données sur tableaux. En revanche, le calcul de zéros dans les fonctions polynomiales n'est pas pertinent pour l'analyse point par point, et les versions point par point de ces VIs sur tableaux ne sont pas nécessaires.

Comment est-il possible d'effectuer une analyse sans buffers de données ?

Les fonctions d'analyse génèrent des solutions qui caractérisent le comportement d'un ensemble de données. Dans l'analyse et l'acquisition de données sur tableaux, il est possible d'analyser un vaste ensemble de données en répartissant ces dernières en dix buffers plus petits. L'analyse de ces dix ensembles de données engendre dix solutions. Vous pouvez en outre unifier ces dix solutions en une seule qui caractérise le comportement de l'ensemble des données tout entier.

Dans l'analyse point par point, vous analysez un ensemble de données complet en temps réel. Une unité d'échantillonnage d'une longueur spécifique remplace un buffer. L'unité d'échantillonnage point par point peut avoir une longueur qui correspond à celle d'un événement significatif

dans l'ensemble de données que vous analysez. Reportez-vous à la section [Étude de cas de l'analyse point par point](#) pour découvrir un exemple permettant d'acquérir plusieurs milliers d'échantillons par seconde afin de détecter des roues défectueuses. Dans ce cas, le signal provient d'un train qui roule à une vitesse de 60 à 70 km/h. La longueur de l'échantillon pour cette application correspond à la distance minimale entre les roues.

Une application d'analyse point par point standard analyse une longue série d'unités d'échantillonnage, mais il est probable que vous ne vous intéresserez qu'à quelques-unes de ces unités. Pour identifier ces échantillons d'un intérêt décisif, l'application se concentre sur les transitions, telles que la fin du signal pertinent.

Dans la section [Étude de cas de l'analyse point par point](#), une application permettant de détecter des roues de train défectueuses utilise la fin du signal afin d'identifier des échantillons d'un intérêt décisif. Au moment même où l'application identifie ce point de transition, elle capture la valeur de l'amplitude maximale de l'unité d'échantillonnage en question. Cette valeur particulière de l'amplitude correspond au signal complet pour la roue située sur le train dont le signal vient juste de terminer. Vous pouvez utiliser cette valeur de l'amplitude temps réel pour générer un événement ou un rapport sur la roue d'un train particulier.

En quoi l'analyse point par point est-elle plus efficace dans les applications temps réel ?

En règle générale, lorsque vous devez traiter un flux de données rapide et continu, l'analyse point par point est une solution adaptée. Par exemple, dans un contexte d'automatisation industrielle, les données de contrôle circulent sans interruption et les ordinateurs utilisent toute une variété de fonctions d'analyse et de transfert afin de contrôler un processus réel. L'analyse point par point peut s'effectuer en temps réel pour mener à bien ces tâches industrielles.

Certaines applications temps réel ne nécessitent pas d'acquisition ni d'analyse de données haute vitesse. Au lieu de cela, elles requièrent des programmes simples et fiables. L'analyse point par point offre cette simplicité et cette fiabilité, car vous n'allouez pas les tableaux de manière explicite et l'analyse de données se déroule naturellement et sans interruption.

Critères d'utilisation de l'analyse point par point

Vous pouvez continuer à travailler avec une analyse traditionnelle sur tableaux de points tant que vous pouvez contrôler votre processus sans recourir à une acquisition de données point par point déterministe et haute vitesse. Toutefois, si vous consacrez des ressources à une application d'acquisition de données temps réel, nous vous conseillons d'utiliser l'analyse point par point afin d'exploiter pleinement le potentiel de votre application. Lorsque vous augmentez la vitesse d'échantillonnage par dix, la nécessité d'utiliser l'analyse point par point s'accroît.

L'approche point par point simplifie la conception, la mise en œuvre et les processus de test car le flux de l'application coïncide étroitement avec le flux naturel des processus réels que vous désirez surveiller et contrôler.

Quelle est l'importance à long terme de l'analyse point par point ?

L'acquisition et l'analyse de données temps réel exigent sans cesse des applications plus stables et optimisées. L'analyse point par point est optimisée et stable, car elle s'insère directement dans le processus d'acquisition et d'analyse. L'analyse point par point optimisée et stable offre des performances similaires à celles des circuits FPGA (réseau de portes logiques programmable), des processeurs de signaux numériques, des contrôleurs embarqués, des unités centrales dédiées et des ASIC.

Exemples

L'analyse et le contrôle temps réel ne sont que quelques-uns des avantages du concept point par point. Les exemples suivants illustrent les avantages offerts par les fonctions et les VIs point par point.

Filtre point par point et sur tableaux de points

Le VI Filtre PPP et sur tableaux, illustré à la figure 3, génère un signal contenant un bruit parasite. Ce VI filtre le signal afin de supprimer le bruit et de fournir une meilleure représentation du signal actuel. Le filtre bloque les fréquences parasites dans le signal et permet le passage des fréquences importantes. Le VI utilise deux méthodes pour filtrer le bruit parasite du signal. La première supprime le bruit parasite à l'aide de la version point par point du filtre. La seconde utilise la fonction de filtrage sur tableaux.

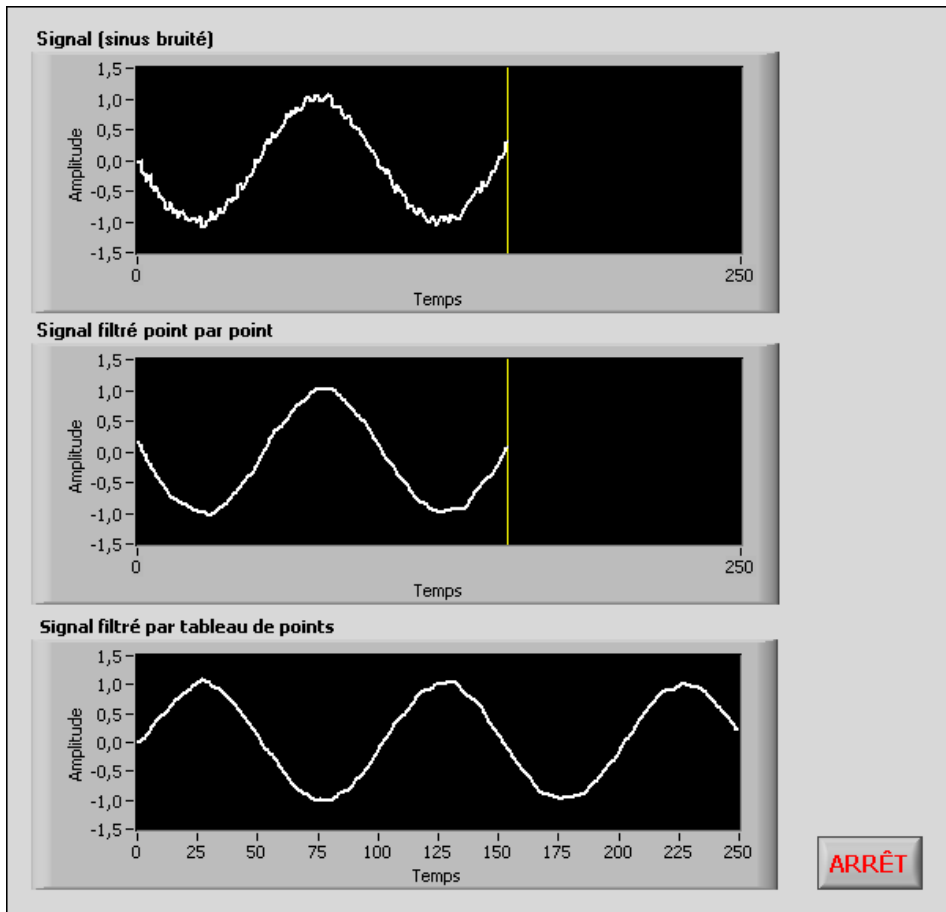


Figure 3. Comparaison de filtre point par point ou sur tableaux

Filtre point par point

Dans la programmation LabVIEW point par point, un VI acquiert un point de données unique et l'analyse, permettant ainsi une génération de rapports en temps réel. Au même moment, le VI acquiert un autre point de données unique et répète le processus point par point. Ainsi, le VI analyse chaque point du signal entrant et permet la génération d'un rapport ou d'une action en temps réel.

Filtre sur tableaux

Dans la programmation LabVIEW sur tableaux, un VI acquiert un ensemble de données, analyse ces données et génère un rapport. Dans le filtre sur tableaux, le VI acquiert et analyse un intervalle de signal. Après analyse, le filtre supprime les fréquences parasites de l'ensemble de données et le VI affiche un rapport de l'ensemble des données filtrées dans le graphe. Le graphe montre le signal filtré dans sa totalité. Le VI attend ensuite qu'une autre longueur de signal soit acquise puis répète le processus.

Exemple de spectre d'amplitude temps réel

Le VI Spectre d'amplitude temps réel (Realtime Amplitude Spectrum.vi), illustré à la figure 4, génère un signal d'échantillonnage et ajoute un bruit blanc. Le VI analyse le signal et affiche le spectre de puissance du signal sur un graphe.

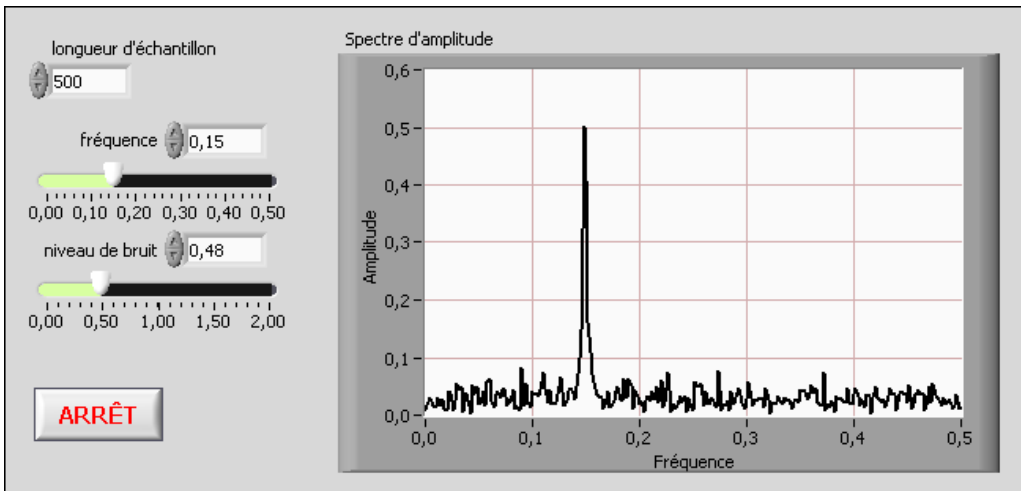


Figure 4. Analyse de spectre de puissance temps réel

Ce VI utilise des VIs point par point, vous permettant ainsi de modifier la fréquence, le niveau de bruit et la longueur de l'échantillon en temps réel. Ces changements se répercutent sur le spectre de puissance en temps réel. Le concept point par point est très utile dans les analyses et acquisitions de données qui exigent un niveau temps réel de détail.

Histogramme mobile PPP (Moving Histogram PtByPt.vi)

Le VI Histogramme mobile PPP (Moving Histogram PtByPt.vi), illustré à la figure 5, génère un signal, l'analyse, puis crée un histogramme. Vous pouvez régler les intervalles et les longueurs d'échantillons en temps réel.

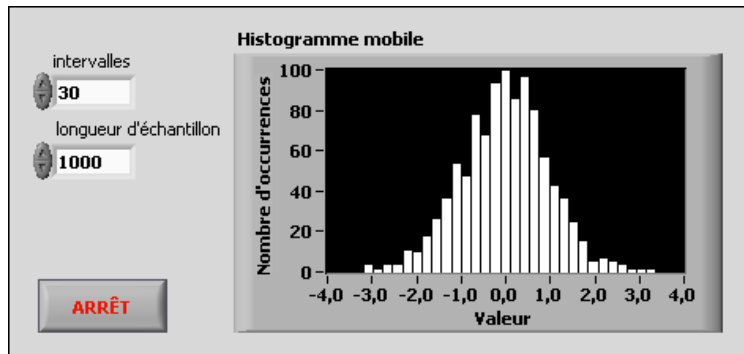


Figure 5. Histogramme mobile créé avec des VIs point par point

Il est possible, mais aussi beaucoup plus difficile, de créer un histogramme mobile en utilisant la méthode d'analyse sur tableaux. Le concept point par point vous permet de créer très facilement des VIs avec un comportement temps réel.

Étude de cas de l'analyse point par point

L'étude de cas de cette section montre une application complète d'analyse point par point construite avec LabVIEW. Une application d'acquisition de données temps réel qui détecte des roues de train défectueuses démontre la simplicité et la souplesse de l'analyse de données point par point. Le VI Roue de train PPP (Train Wheel PtByPt.vi) utilise les VIs point par point.

Analyse point par point de roues de train

Dans cet exemple, le personnel de maintenance d'un dépôt ferroviaire doit détecter des roues défectueuses. Un cheminot qui frappe sur une roue défectueuse à l'aide d'un marteau peut entendre une résonance différente permettant d'identifier un défaut. La surveillance automatisée doit remplacer le test manuel, car ce dernier est trop lent, trop rudimentaire pour détecter des vices cachés et comporte trop de risques d'erreurs. Une solution automatisée apporte également la puissance du test dynamique, car les roues de train peuvent être en mouvement durant le test, au lieu d'être immobiles.

La solution permettant d'identifier des roues de train potentiellement défectueuses doit également détecter les vices cachés avec rapidité et précision. L'application doit collecter les données pendant un trajet normal du train. Elle doit par ailleurs acquérir et analyser les données en temps réel pour simplifier la programmation et accroître la vitesse et la précision des résultats.

Le VI Roue de train PPP (Train Wheel PtByPt.vi) utilise les fonctionnalités d'analyse point par point de LabVIEW afin de construire une solution capable de détecter les roues de train défectueuses.

Les figures 6 et 7 montrent respectivement la face-avant et le diagramme du VI Roue de train PPP (Train Wheel PtByPt.vi). Vous pouvez télécharger ce VI, ainsi que d'autres VIs d'exemple, dans le répertoire `exemple\PtBYPt` de LabVIEW.

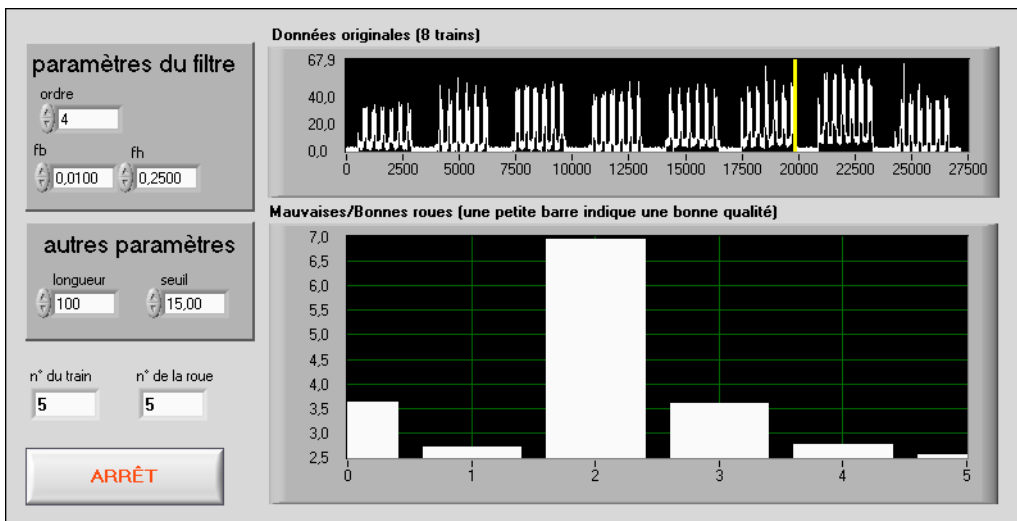


Figure 6. Face-avant du VI Roue de train PPP (Train Wheel PtByPt.vi)

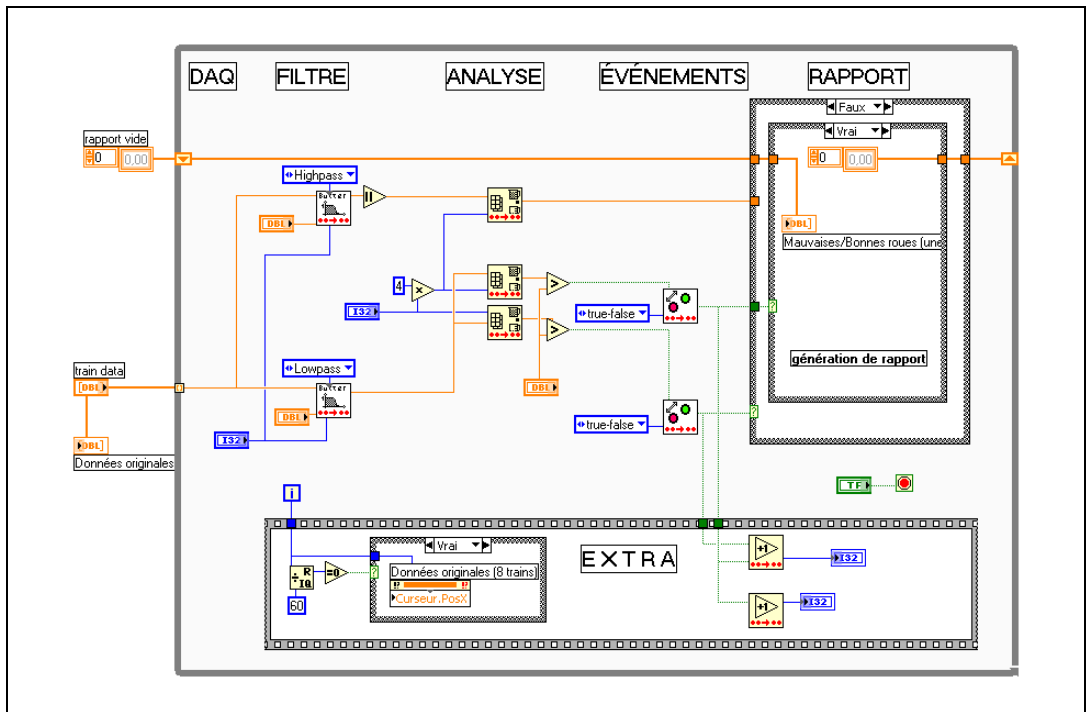


Figure 7. VI Roue de train PPP (Train Wheel PtByPt.vi)



Remarque Cet exemple traite de la mise en œuvre d’un programme d’analyse point par point sous LabVIEW. Les problèmes d’approche du conditionnement du signal et de périodes d’échantillonnage idéales dépassent le cadre de cet exemple.

Aperçu de la solution point par point de LabVIEW

Les données acquises par l’application de détection de roues circulent de manière continue à l’intérieur d’une boucle While. Le processus suivant décrit ce qui se passe dans la boucle :

1. DAQ : flux de données “waveform”
2. Filtre : séparation des composantes basse et haute fréquence
3. Analyse : détection de train, de roue et de niveau d’énergie du signal pour chaque roue
4. Événement : traitement de la détection de transitions sur les signaux correspondant aux différentes roues des trains
5. Rapport : enregistrement des trains, des roues et des trains susceptibles d’avoir des roues défectueuses.

Cette application de détection utilise des objets de programmation LabVIEW standards, tels que des structures Condition, des boucles While, des commandes numériques ainsi que des opérateurs numériques. Le programme du VI Roue de train PPP (Train Wheel PtByPt.vi) requiert également les VIs répertoriés dans les sections suivantes.

Caractéristiques d'un signal de roue de train

Le signal caractéristique émis par les roues de train détermine la façon d'analyser et de filtrer le signal point par point. Une roue de train en mouvement émet un signal qui contient des composantes hautes et basses fréquences. Si vous installez une jauge de contrainte sur une voie ferrée, vous détectez un signal bruité semblable à une courbe en cloche. La figure 8 illustre les composantes haute et basse fréquences de cette courbe.

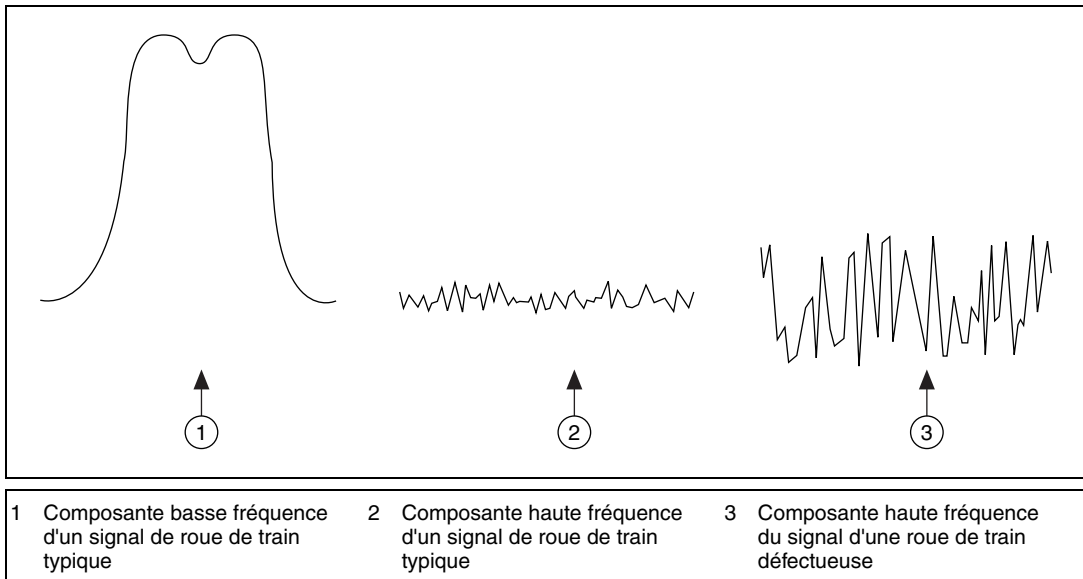


Figure 8. Composantes basse et haute fréquences d'un signal de roue de train

La composante basse fréquence du mouvement d'une roue de train représente le bruit de fonctionnement normal. Les roues normales et les roues défectueuses génèrent la même composante basse fréquence dans le signal. Le pic de la courbe représente le moment où la roue passe directement au-dessus de la jauge de contrainte. Les points les plus bas de la courbe en cloche représentent respectivement le début et la fin de la roue lors du passage de la roue au-dessus de la jauge de contrainte.

Le signal pour une roue de train contient également une composante haute fréquence qui reflète la qualité de la roue. En fonctionnement, une roue défectueuse génère davantage d'énergie qu'une roue normale. En d'autres

termes, la composante haute fréquence d'une roue défectueuse présente une amplitude plus importante.

Paramètres de signaux dans le VI Roue de train PPP (Train Wheel PtByPt.vi)

Le signal de toutes les roues, y compris celles qui sont défectueuses, est compris dans un intervalle prévisible. Ce comportement prévisible vous permet de choisir les paramètres qui figurent dans le tableau 4. Ces paramètres s'appliquent aux cinq étapes décrites dans la section [Aperçu de la solution point par point de LabVIEW](#).



Remarque Les paramètres du tableau peuvent vous aider à comprendre les détails de programmation avec l'approche point par point. Cependant, vous pouvez comprendre l'acquisition et l'analyse point par point sans étudier les paramètres du tableau. Rappelez-vous, vous devez régler les paramètres pour toute mise en œuvre du VI Roue de train PPP (Train Wheel PtByPt.vi), car les caractéristiques diffèrent d'un système d'acquisition de données à l'autre.

Tableau 4. Paramètres acceptables pour les fonctions utilisées dans le VI Roue de train PPP (Train Wheel PtByPt.vi)

Paramètre	Nom	Étape et VI concerné	But recherché dans le VI Roue de train PPP (Train Wheel PtByPt.vi)
Entrée de données	données d'entrée	Étape d'acquisition de données/DAQ In.VI	Source de données, périphérique, voie, échantillons par seconde et mise à l'échelle.
Résolution du filtre	ordre	Étape de filtrage/Filtre de Butterworth PPP	Manière dont le filtre coupe les fréquences hors de la bande sélectionnée. La valeur 2 est acceptable pour le VI Roue de train PPP (Train Wheel PtByPt.vi). L' ordre s'applique aux deux filtres de Butterworth dans le VI Roue de train PPP (Train Wheel PtByPt.vi).

Tableau 4. Paramètres acceptables pour les fonctions utilisées dans le VI Roue de train PPP (Train Wheel PtByPt.vi) (Suite)

Paramètre	Nom	Étape et VI concerné	But recherché dans le VI Roue de train PPP (Train Wheel PtByPt.vi)
Fréquence de coupure basse	fl	Étape de filtrage/Filtre de Butterworth PPP	Déterminer l'amplitude minimale du signal identifiant qu'une roue a quitté la jauge de contrainte. Pour cela, un filtre passe-bas est appliqué. Une fréquence (relative à la fréquence d'échantillonnage) de 0,01 est acceptable pour le VI Roue de train PPP (Train Wheel PtByPt.vi).
Fréquence de coupure haute	fh	Étape de filtrage/Filtre de Butterworth PPP	Un filtre passe-haut permet d'étudier les composantes haute fréquence générées par les vibrations de la roue et d'en déterminer la qualité. Une fréquence (relative à la fréquence d'échantillonnage) de 0,25 est acceptable pour le VI Roue de train PPP (Train Wheel PtByPt.vi).
Longueur du signal à analyser	longueur de l'échantillon	Étape d'analyse/Max. & Min. d'un tableau PPP	Taille de la portion du signal analysé par Roue de train PPP (Train Wheel PtByPt.vi). Pour calculer la longueur de l'échantillon idéale, il faut prendre en compte la vitesse du train, la distance minimale entre les roues et le nombre d'échantillons acquis par seconde. La valeur 100 est acceptable pour le VI Roue de train PPP (Train Wheel PtByPt.vi). Le VI Roue de train PPP (Train Wheel PtByPt.vi) utilise longueur d'échantillon afin de calculer les valeurs dans les trois appels du VI Max. & Min. d'un tableau PPP.
Initialisation	initialiser	Tous les VIs point par point	Routine qui réinitialise un VI pour une nouvelle session d'acquisition de données continue.

Tableau 4. Paramètres acceptables pour les fonctions utilisées dans le VI Roue de train PPP (Train Wheel PtByPt.vi) (Suite)

Paramètre	Nom	Étape et VI concerné	But recherché dans le VI Roue de train PPP (Train Wheel PtByPt.vi)
(Multiplicateur)	(aucun)	Étape d'analyse/Max. & Min. d'un tableau PPP	Permet de spécifier une portion de signal à analyser plus importante. Si cette portion plus importante ne parvient pas à détecter une activité au niveau des roues, le VI en déduit que le train complet est passé. La valeur 4 est acceptable pour le VI Roue de train PPP (Train Wheel PtByPt.vi).
(Seuil)	(aucun)	Étape d'analyse/Max. & Min. d'un tableau PPP	L'amplitude en sortie du filtre passe-haut est comparée à ce seuil point par point afin de détecter le début du passage d'une roue de train sur le capteur. Un seuil de 3 est acceptable pour le VI Roue de train PPP (Train Wheel PtByPt.vi).

Vitesse d'échantillonnage dans le VI Roue de train PPP (Train Wheel PtByPt.vi)

L'application de détection point par point opère sur un flux continu de données issues des roues d'un train en mouvement. Pour un train qui se déplace à une vitesse de 60 à 70 km/h, quelques centaines à quelques milliers d'échantillons par seconde peuvent certainement fournir les informations suffisantes pour détecter une roue défectueuse.

Exigences de filtrage du VI Roue de train PPP (Train Wheel PtByPt.vi)

L'application de détection point par point doit filtrer les composantes haute et basse fréquences d'un signal de roue de train. Deux filtres de Butterworth point par point accomplissent les tâches suivantes :

- Extraction des composantes basse fréquence du signal
- Extraction des composantes haute fréquence du signal.

Composantes utilisées pour l'analyse du VI Roue de train PPP (Train Wheel PtByPt.vi)

L'application de détection point par point doit analyser séparément les composantes haute et basse fréquences. Dans ce cas, la version point par point du VI Max. & Min. d'un tableau extrait les données qui révèlent le niveau d'énergie contenu dans les signaux de chaque roue, fin du train et fin de la roue.

Trois VIs Max. & Min. d'un tableau PPP distincts remplissent les tâches discrètes suivantes :

- Identification de l'amplitude maximale de la composante haute fréquence pour chaque roue
- Identification de la fin de chaque train
- Identification de la fin de chaque roue.



Remarque Le titre “VI Max. & Min. d'un tableau PPP” contient le mot “tableau” uniquement pour rappeler celui de la version sur tableaux du même VI. Il n'est pas nécessaire d'allouer des tableaux pour la version point par point de ce VI.

Une fois que l'étape d'analyse a identifié les valeurs maximales et minimales, une étape de gestion des événements détecte l'instant où ces valeurs franchissent un seuil donné.

Composantes utilisées dans la gestion des événements du VI Roue de train PPP (Train Wheel PtByPt.vi)

L'application de détection point par point enregistre chaque roue et chaque train qu'elle détecte. Le VI Basculement de booléen PPP génère un événement à chaque fois que le VI Max. & Min. d'un tableau PPP détecte la fin d'une roue de train. Le système d'acquisition de données génère également un événement à chaque fois que le VI Max. & Min. d'un tableau pt par pt détecte la fin d'un train. L'analyse du signal haute fréquence identifie, le cas échéant, quelles sont les roues défectueuses. Lorsque le VI est confronté à une roue potentiellement défectueuse, il transmet directement l'événement au rapport au moment même où le rapport reçoit l'événement de fin de la roue.

Deux VIs Basculement de booléen PPP remplissent les tâches suivantes :

- Détection du point de transition dans le signal indiquant la fin de la roue
- Détection du point de transition dans le signal indiquant la fin du train.

Le VI Basculement de booléen PPP réagit à des transitions. Lorsque l'amplitude d'un signal de roue unique tombe en dessous d'un certain seuil, la fin de la roue est arrivée au niveau de la jauge de contrainte. Pour ce système d'acquisition de données, 3 représente un bon seuil pour identifier la fin de la roue. Lorsque la longueur de signal tombe en dessous de ce niveau, le VI Basculement de booléen reconnaît un événement de transition et transmet cet événement à un rapport.

Composantes utilisées pour le rapport du VI Roue de train PPP (Train Wheel PtByPt.vi)

L'application de détection point par point effectue un rapport sur toutes les roues de tous les trains qui sont mesurées par le système d'acquisition de données. Le système effectue également un rapport sur toutes les roues susceptibles d'être défectueuses.

Chaque fois qu'une roue franchit la jauge de contrainte, l'application capture son signal, l'analyse et effectue un rapport sur l'événement. Le tableau 5 décrit les composantes d'un rapport sur une roue de train unique.

Tableau 5. Exemple de rapport sur une roue de train unique

Sources d'informations	Signification des résultats
Mécanisme de comptage des événements du signal	Première étape : la roue n°4 a franchi la jauge de contrainte.
Analyse des données en sortie du filtre passe-haut	Deuxième étape : la roue n°4 a franchi la jauge de contrainte et la roue est peut-être défectueuse.
Mécanisme de comptage des événements de fin du train	Troisième étape : la roue n°4 du train n°8 a franchi la jauge de contrainte et la roue est peut-être défectueuse.

Le VI Roue de train PPP (Train Wheel PtByPt.vi) utilise l'analyse point par point pour générer un rapport et non pour contrôler un processus industriel. Toutefois, ce système d'acquisition de données acquiert les données en temps réel, et vous pouvez modifier l'application afin de générer des réponses pour un contrôle en temps réel, afin par exemple de stopper le train lorsqu'elle rencontre une roue potentiellement défectueuse.

Conclusion

Lorsque l'on acquiert des données avec des performances temps réel, l'analyse point par point permet l'analyse des données en temps réel. L'analyse point par point est continue et instantanée. Pendant que vous acquérez des données, vous les filtrez et vous les analysez point par point afin d'extraire les informations dont vous avez besoin et de prendre certaines décisions en conséquence. Cette étude de cas démontre l'efficacité de l'approche point par point pour générer à la fois des événements et des rapports en temps réel.



370432A-01

Jan02