

Département de génie civil Faculté de génie

RAPPORT DE RECHERCHE

Centre de recherche en génie parasismique et en dynamique des structures

QUANTIFICATION DE L'ENDOMMAGEMENT MINIMAL DÉTECTABLE PAR DES MESURES DE VIBRATIONS DANS UN PONT ROUTIER EN SERVICE

Système de télésurveillance pour le suivi en continu des propriétés modales d'un pont

A. Loignon, X. Mongrain-Lalonde, O. Gauron et P. Paultre



CRGP

Rapport de projet de recherche exécuté par l'Université de Sherbrooke pour le Ministère des Transports du Québec :

Quantification de l'endommagement minimal détectable par des mesures de vibrations dans un pont routier en service

Système de télésurveillance pour le suivi en continu des propriétés modales d'un pont

Numéro de référence : MTQ-R732.1 Numéro de rapport : CRGP-2018/01

> Alex LOIGNON Xavier MONGRAIN-LALONDE Olivier GAURON Patrick PAULTRE

Université de Sherbrooke

Département de génie civil 2500 Boulevard Université J1K 2R1 Sherbrooke (Qc) Canada

31 janvier 2018

Résumé

Le présent document constitue le rapport final du volet 3 du projet de recherche R732.1 mené par l'Université de Sherbrooke (UdeS) pour le ministère des Transports du Québec (MTQ) portant sur la quantification de l'endommagement minimal détectable par des mesures de vibrations ambiantes sur un pont routier en service. Ce volet portait sur le déploiement sur un pont réel, le pont de la Rivière-aux-Mulets, d'un système de télésurveillance des propriétés modales de la structure.

Le projet a pour objectif principal le développement d'un prototype de système de télésurveillance de pont effectuant en continu et en temps réel l'extraction des propriétés modales et le suivi de leur évolution à partir de mesures de vibrations ambiantes. L'idée générale de fonctionnement du système est la suivante. D'abord, un système de matériels in situ mesure les vibrations ambiantes de la structure, qui est excitée par des charges de services régulières et par les conditions climatiques et sismiques environnantes. Ensuite, un algorithme spécialisé permet d'extraire les propriétés modales contenues dans les signaux enregistrés sur le site. Finalement, un portail web dédié présente les résultats de ces analyses en affichant, entre autre, l'évolution et le détail des propriétés modales. Plusieurs fonctionnalités spécifiques ont constitué le cahier des charges du système, combinées à des objectifs généraux de facilité d'emploi et de maintenance.

Le système développé est physiquement formé par deux matériel distincts. Le VI LabVIEW développé pour l'enregistrement de données en continu sur le CompactRIO et répondant à diverses situations constitue le premier élément majeur développé. Le serveur informatique, équipé d'une base de données, d'un serveur web HTTP et d'une liste de tâches permettant le pilotage et l'alimentation de la base en données ainsi que l'analyse modale automatique constitue le second élément majeur développé. Ces appareils sont connectés par le réseau internet et tous deux sont reliés au responsable de l'ouvrage indépendamment l'un de l'autre. Au niveau informatique, l'approche proposée consiste en un paradigme logiciel client/serveur. Donc, avec un simple navigateur web, l'utilisateur accède au serveur informatique du système de télésurveillance. Il peut consulter des enregistrements, apprécier l'évolution des propriétés modales, observer les déformées modales d'intérêt, observer les signaux en temps réel mesurés sur la structure, etc. En plus, un super-utilisateur peut reconfigurer le système de fond en comble et réaliser les opérations de maintenance usuelle. Le système développé limite les coûts en se basant d'une part sur des choix de logiciels ouverts et libres d'utilisations et d'autre part sur des logiciels actuellement utilisés par le Ministère.

En l'état actuel de développement, toutes les fonctions essentielles du prototype sont fonctionnelles, tant au niveau de l'acquisiteur qu'au niveau du serveur informatique et du portail web. Le système répond donc aux exigences du contrat de recherche R732.1. Les auteurs recommandent néanmoins de poursuivre le développement du prototype afin de le doter de fonctionalités additionelles qui seraient très précieuses aux utilisateurs du Ministère, dont plusieurs ont déjà été en grande partie développées dans le cadre du projet.

Table des matières

Re	ésumé	i
Τa	able des figures	\mathbf{iv}
Li	iste des tableaux	vi
1	Introduction 1.1 Mise en contexte	1 1 2 4 5 6
2	Architectures existantes de systèmes de télésurveillance 2.1 Architectures 2.2 Exemples notables de systèmes de télésurveillance de ponts	7 8 14
3	Approche de développement proposée du système de télésurveillance	17
3 4	Approche de développement proposée du système de télésurveillance Système d'acquisition 4.1 Matériel	117 21 21 24 27 46 46

6	Conclusion

A Fiche technique : NI cRIO-9074	84
B Fiche technique : NI-9239	93
C Fiche technique : NI PS-15	103
D Fichier de configuration	106
Bibliographie	108

Table des figures

1.1	Pourcentage de la valeur du parc de structures par décennie de construction	1
1.2	Structures à réparer en raison de leur état	2
1.3	Pont est de la Rivière-aux-Mulets	3
2.1	Schéma conceptuel d'un système de télésurveillance	8
2.2	Architecture d'un système manuel	9
2.3	Architecture d'un système automatique	9
2.4	Architecture du BCSIMS	10
2.5	Architecture multi-agent	11
2.6	Exemple de solution matérielle et logicielle offert par Wier-Jones	12
2.7	Interface web riche en information	13
2.8	Base de données distribuée sur plusieurs serveurs	14
2.9	Disposition des capteurs et systèmes d'acquisitions du Sutong Bridge	15
2.10	Disposition des capteurs et des systèmes d'acquisitions du Tsing Ma Bridge	16
2.11	Vue en élévation du Pitt River Bridge	16
3.1	Architecture du système de télésurveillance développé	18
4.1	Système d'acquisition utilisé pour le prototype	23
4.2	Architecture du CompactRIO	24
4.3	GRAFCET du programme d'acquistion dans le CompactRIO	25
4.4	Modèle suggéré par National Instruments pour les machine d'état	26
4.5	Étape $Init$	27
4.6	Programmation LabVIEW de l'état Init de la machine d'états	27
4.7	Étape $E10$	28
4.8	Programmation LabVIEW de l'état <i>E10</i> de la machine d'états	29
4.9	Étape $E20$	30
4.10	Programmation LabVIEW de l'état <i>E20</i> de la machine d'états	31
4.11	VI exécuté par la puce FPGA	31
4.12	Étape $E30$	32
4.13	Simplification de la programmation LabVIEW de l'état $E30$ de la machine d'états	33
4.14	Fonctions LabVIEW de gestion des occurrences	34
4.15	E30 : boucle de cueillette et empilement des données	35
4.16	<i>E30</i> : boucle <i>Répartition interne</i> de gestion des réceptivités	36

4.17	E30 : boucle Répartition externe de gestion des réceptivités	37
4.18	Étape $E31$	37
4.19	E30 : boucle Enregistrement E31 d'écriture sur événement	38
4.20	Étape $E32$	39
4.21	<i>E30</i> : boucle <i>Enregistrement E32</i> d'écriture sur horaire programmé	39
4.22	Étape $E33$	40
4.23	E30 : boucle Enregistrement E33 d'écriture à la demande	41
4.24	Étape $E34$	41
4.25	Étape $E35$	42
4.26	Composantes et liaisons impliquées dans la diffusion en temps réel des données	43
4.27	E30 : boucle Diffusion de préparation des données pour le streaming	45
۳ 1		40
5.1	Ordinateur portable utilise comme serveur informatique pour le prototype	49
5.2	Exemple d'une session MariaDB	53
53	Captures d'écran de l'onglet Accueil	60
0.0		00
5.3	Capture d'écran d'un onglet quelconque avant authentification	60
$5.3 \\ 5.4 \\ 5.5$	Capture d'écran de l'onglet Vibrations ambiantes	60 63
5.3 5.4 5.5 5.6	Capture d'écran de l'onglet Vibrations ambiantes	60 63 65
5.3 5.4 5.5 5.6 5.7	Capture d'écran de l'onglet <i>Vibrations ambiantes</i>	60 63 65 68
5.4 5.5 5.6 5.7 5.8	Capture d'écran de l'onglet Analyses modales - historique	60 63 65 68 70
5.3 5.4 5.5 5.6 5.7 5.8 5.9	Capture d'écran de l'onglet Analyses modales - détails	60 63 65 68 70 71
5.3 5.4 5.5 5.6 5.7 5.8 5.9 5.10	Capture d'écran de l'onglet Analyses modales - détails	60 63 65 68 70 71 72
5.3 5.4 5.5 5.6 5.7 5.8 5.9 5.10 5.11	Capture d'écran de l'onglet Analyses modales - détails	60 63 65 68 70 71 72 74
5.3 5.4 5.5 5.6 5.7 5.8 5.9 5.10 5.11 5.12	Capture d'écran de l'onglet Analyses modales - historique	$60 \\ 60 \\ 63 \\ 65 \\ 68 \\ 70 \\ 71 \\ 72 \\ 74 \\ 79$

Liste des tableaux

4.1	Spécifications requises d'un système d'acquisition pour un déploiement in situ permanent	22
4.2	Système d'acquisition NI cRIO recommandé dans le rapport CRGP-2014/04	22
4.3	Taille mémoire requise des files de données flottantes	30
4.4	Paramètres du fichier de configuration	46
5.1	Logiciels et langages de programmation utilisés sur le serveur informatique	50
5.2	Tables et attributs de la base de données <i>pont_riviere-aux-mulets</i>	54
5.3	Organisation de l'ensemble des fichiers composant le site web	58
5.4	Étapes de fonctionnement de l'onglet Analyses modales - détails	71
5.5	Liste des tâches de l'algorithme d'analyse modale automatique	77

Chapitre

1

Introduction

1.1 MISE EN CONTEXTE

Le présent document constitue le rapport final du volet 3 du projet de recherche R732.1 mené par l'Université de Sherbrooke (UdeS) pour le ministère des Transports du Québec (MTQ) portant sur la quantification de l'endommagement minimal détectable par des mesures de vibrations ambiantes sur un pont routier en service. La finalité de ce projet de recherche est la détection fine (localisation et quantification) de l'endommagement dans les structures de ponts par une méthode innovante jumelant des mesures expérimentales in situ et un modèle numérique des structures par éléments finis.

Le maintien en service et la maintenance des infrastructures routières sont des enjeux majeurs pour leurs propriétaires. L'inspection et la surveillance régulière de l'ensemble des ouvrages, particulièrement des ouvrages stratégiques, constituent ainsi un investissement important et nécessaire au Québec, où le réseau de ponts est vieillissant et fait l'objet de nombreuses réfections. Le réseau routier couvert par le ministère des Transports du Québec comprend 5307 ouvrages d'art, dont près de 70 % ont été construits entre 1960 et 1980 [24], comme l'illustre la figure 1.1.



Figure 1.1 – Pourcentage de la valeur du parc de structures par décennie de construction [24]

Les travaux de réparation majeurs sont habituellement nécessaires 30 ans après la construction des structures. À la figure 1.2, on constate qu'un effort particulier y est accordé depuis 2008, puisque le pourcentage des structures à réparer en raison de leur état diminue fortement d'année en année et a atteint 31,2 % en 2011 [24]. Cependant, 31,2 % des 5307 ouvrages d'art représentent encore 1655 structures devant faire face à des mesures de réfection plus ou moins importantes.



Figure 1.2 – Structures à réparer en raison de leur état [24]

Les méthodes de surveillance classiques de structures de pont (inspections visuelles, mesures acoustiques ou magnétiques, etc.) sont pertinentes, mais elles sont limitées, car elles ne permettent d'obtenir que des informations locales, et ne permettent pas d'inspection des zones inaccessibles de la structure. Dans ce contexte, l'utilisation d'une technique expérimentale in situ permettant d'augmenter le niveau de précision de l'inspection visuelle ou des informations issues d'instruments statiques très localisés devient un outil pertinent pour assurer une meilleure gestion du parc des ouvrages d'art.

Une série d'accéléromètres ou de capteurs de vitesse adéquatement disposés sur l'ensemble de l'ouvrage permet de faire un suivi global de la structure en déterminant ses propriétés dynamiques. Les variations soudaines ou progressives des fréquences naturelles et des modes de vibration sur une période donnée sont des indicateurs de changements dans les propriétés structurales d'un pont par rapport à son état initial, et donc d'endommagement [6, 21, 32]. Pour cette raison, la grande majorité des méthodes existantes de détection globale de l'endommagement font la comparaison des propriétés dynamiques obtenues lors d'une nouvelle analyse et lors de la précédente (ou avec les propriétés originales de la structure). C'est aussi sur ce principe que se base la méthode de détection de l'endommagement qui est testée dans le cadre du projet de recherche global mené par l'Université de Sherbrooke pour le ministère des Transports du Québec. Celle-ci est basée sur le recalage de modèles numériques à partir de résultats d'analyses modales expérimentales. La méthode consiste à adapter les propriétés méconnues d'un modèle d'éléments finis de manière à minimiser les écarts entre les données modales expérimentales et les prédictions du modèle d'éléments finis. La méthode utilisée est itérative et ajuste des paramètres physiques locaux qui actualisent par conséquent certaines matrices de rigidité élémentaires du modèle dans lesquelles ces paramètres figurent. Les lieux et intensités des ajustements effectués aux propriétés physiques locales peuvent être interprétés en termes de localisation et de quantification de dommages à la structure.

1.2 ANTÉCÉDENTS DU PROJET DE RECHERCHE R732.1

Le présent document est la suite directe du rapport CRGP-2014/04 [10] présenté au Ministère en mars 2014, qui portait sur l'analyse modale expérimentale du pont de la Rivière-aux-Mulets et la proposition d'une architecture d'instrumentation permanente à déployer sur le pont afin d'y appliquer ultérieurement les méthodes de détection d'endommagement développées. Le présent rapport conclut les travaux en lien avec l'objectif 3 du contrat de recherche R732.1 établi entre le Ministère et l'Université. Cet objectif y est énoncé comme suit :

Déployer, sur un pont du réseau du ministère des Transports, une instrumentation pour le suivi des propriétés dynamiques de l'ouvrage. Les données recueillies par ce dispositif seront accessibles à distance en tout temps par une connexion internet. L'acquisition des données pourra être effectuée régulièrement et automatiquement par programmation ou sur demande à distance. Le pont choisi conjointement par les parties pour l'étude est le pont est de la Rivière-aux-Mulets situé à Sainte-Adèle (Québec) sur l'autoroute des Laurentides (autoroute 15) en direction des Laurentides (fig. 1.3). En 1964, la construction de cet ouvrage unique en son genre en faisait une première en Amérique du Nord. Sa construction s'est effectuée en duo avec son jumeau orienté vers le sud qui a dû être démoli en 2006. Ces deux ponts ont été construits selon la technique de poutres-caissons (voussoirs) en béton mises en place par encorbellement (ou porte-à-faux successifs) et post-tensionnées.



Figure 1.3 – Pont est de la Rivière-aux-Mulets

Le pont à l'étude avait subi plusieurs réparations majeures, particulièrement l'ajout de postcontrainte interne et externe au caisson, et avait été instrumenté (cordes vibrantes, inclinomètres, LVDT, etc.) afin de suivre son évolution. Comme ce pont présentait des problèmes structuraux importants et qu'il s'endommagait continuellement, un suivi précis devait en être fait, ce qui en faisait un excellent candidat pour l'étude du projet de recherche. Cependant, au printemps 2016, le Ministère des Transports a informé l'Université de Sherbrooke que le remplacement du pont était planifié pour le début de l'année 2017. D'un commun accord entre les parties, il a donc été décidé de ne pas déployer de façon effective l'instrumentation envisagée, et que l'objectif 3 du projet serait donc tronqué de son application pratique. Cependant, l'essentiel du travail relié à l'objectif 3 ne réside pas tant dans l'installation effective in situ du système de télésurveillance que dans son développement. Ce travail de développement a donc été effectué en intégralité dans le but de l'utiliser concrètement sur un autre pont ultérieurement, et il est l'objet de ce rapport.

Une instrumentation permanente, alors adaptée au pont de la Rivière-aux-Mulets, a été définie dans le rapport CRGP-2014/04 [10], qui est pertinente pour une application des méthodes de détection d'endommagement par recalage des propriétés modales d'un modèle par éléments finis. Cette instrumentation consiste essentiellement en une série de dizaines d'accéléromètres piézoélectriques répondant à certains critères techniques et en un acquisiteur de type National Instrument cRIO doté d'un processeur capable d'effectuer des calculs en temps réel pour le traitement des données des accéléromètres et pour l'extraction des propriétés modales de la structure. C'est donc sur cette base matérielle que le système de télésurveillance décrit dans ce rapport a été développé. Le travail a consisté à programmer l'ensemble des composants logiciels permettant la collecte puis le traitement des données recueillies sur la structure. À terme, ces composants logiciels formeront un système de télésurveillance, en continu et en temps réel, pour la quantification de l'endommagement d'un pont routier en service par des mesures de vibrations ambiantes. Ce système de télésurveillance sera au service des ingénieurs du Ministère pour connaître l'état actuel de la structure directement depuis leur bureau d'étude. Le système développé a principalement pour but d'extraire automatiquement, en temps réel et en continu sur la durée de vie de l'ouvrage, les propriétés modales de la structure à partir d'accéléromètres déployés, à stocker ces propriétés pour en faire l'historique et à les présenter de façon ergonomique à l'utilisateur. L'application des algorithmes de détection d'endommagement par recalage de modèle numérique ne fait pas partie du cahier des charges fixé à ce stade du développement.

1.3 PROBLÉMATIQUE

Les méthodes de détection d'endommagement développées dans le projet de recherche R732.1 reposent sur la connaissance d'une évolution des fréquences propres et modes propres entre deux périodes de temps dans la vie de l'ouvrage. En théorie, ces méthodes pourraient donc être appliquées à partir du travail déjà effectué par les ingénieurs du Ministère, car parmi ses méthodes de surveillance régulières, le Ministère effectue des analyses modales sur ses ouvrages stratégiques. Cette technique permet en effet de collecter de l'information sur la structure à son échelle globale, même si pour le moment les ingénieurs du Ministère ne disposent pas d'outils d'analyses approfondies permettant de faire le lien entre ces propriétés et d'éventuels dommages à l'échelle locale. Le mode opératoire actuel pour mener les analyses modales implique : (1) de se déplacer sur le site du pont avec le matériel d'essai ; (2) de mesurer in situ les vibrations ambiantes du pont à partir de plusieurs configurations de capteurs (cette opération s'effectue la plupart du temps sur plusieurs journées); (3) de ramasser le matériel et de revenir au bureau d'étude pour la phase de post traitement des données; (4) d'analyser les données pour extraire les fréquences propres et modes propres de la structure à l'aide du logiciel ARTeMIS [33] et d'intégrer ces résultats à un fichier de suivi du pont.

Ce mode opératoire, en plus d'être fastidieux et d'impliquer des coûts de déplacement importants pour chaque mesure sur le pont, présente des obstacles pratiques et techniques à l'application appropriée des méthodes de détection d'endommagement :

- Le mode opératoire actuel fournit des données ponctuelles très espacées dans le temps, qui rendent impossible la connaissance en temps réel de l'état de la structure. Cela rend donc impossible la détection d'un endommagement soudain du pont suite à un événement majeur, qu'il soit localisé au périmètre de la structure (ex : collision d'un véhicule avec la structure) ou étendu sur le territoire (ex : séisme). Dans le premier cas, l'absence d'information immédiate sur l'état de la structure rend difficile le choix de l'intervention à exécuter sur l'ouvrage. Dans le deuxième cas, en plus des difficultés associées au choix de l'intervention, il est difficile de prioriser les interventions sur les différents ouvrages concernés. La surveillance en continu et en temps réel constitue l'un des objectifs majeur du projet R732.1.
- Les propriétés dynamiques d'une structure subissent de légères variations en raison des conditions de tempéature ou d'humidité environnantes. Dépendamment des matériaux, du gradient de température considéré, etc., ces variations peuvent être comparables à celles générées par des dommages légers ou modérés sur la structure. Le protocole actuel ne permet pas de caractériser précisément la dispersion des résultats d'analyse modale en raison des variations de conditions à l'intérieur d'une année ou même à l'échelle d'une seule journée. Le proto-

cole ne permet pas non plus de différencier les causes des évolutions modales observées, car les mesures sont très ponctuelles dans le temps et on ne peut jamais choisir exactement les conditions environnementales dans lesquelles elles seront effectuées. Une caractérisation de l'ouvrage à volonté et en continu est une solution à ces difficultés.

— Le mode opératoire actuel implique une caractérisation de l'ensemble du pont à partir de mesures sur une ou plusieurs journées complètes. Cela implique que tous les points d'un mode caractérisé l'ont été dans des conditions environnantes différentes (excitations différentes entre chaque configuration, variations de masse en fonction du trafic routier, possible évolution de température, ...). Ces variations génèrent des incertitudes expérimentales additionnelles dans les résultats, alors que celles-ci constituent la principale difficulté technique pour la fiabilité des algorithmes de détection d'endommagement utilisés. Une mesure simultanée de l'ensemble des points caractérisés est idéale.

De ces problématiques découle le besoin d'une installation permanente sur le site d'une instrumentation mesurant les vibrations ambiantes du pont. C'est un tel système qui a été développé et qui est décrit dans ce rapport. Les algorithmes de détection d'endommagement pourront opèrer par la suite sur la base de la connaissances des propriétés modales caractérisées de manière automatique en tout temps et dans des conditions environnantes documentées.

1.4 OBJECTIFS

Objectif général

Ce projet a pour objectif principal le développement d'un prototype de système de télésurveillance de pont effectuant en continu et en temps réel l'extraction des propriétés modales et le suivi de leur évolution à partir de mesures de vibrations ambiantes.

Objectifs spécifiques et cahier des charges du système

Les objectifs spécifiques du projet constituent le cahier des charges du système. Deux éléments principaux sont développés pour ce prototype qui assurent l'ensemble des fonctionnalités du système : un acquisiteur déployé in situ avec les capteurs, qui réalise l'ensemble des fonctions de mesure de vibrations, et un serveur informatique dans les locaux des utilisateurs, qui assure les fonctions de stockage et de traitement avancé des données.

L'acquisiteur doit effectuer les tâches suivantes :

- (1) Enregistrement flottant continu (ex : dix dernières minutes courantes) des données brutes de l'ensemble des accéléromètres déployés;
- (2) Sauvegarde des données temporelles des capteurs :
 - (a) à la demande,
 - (b) automatiquement (dates et heures programmables),
 - (c) au déclenchement d'un événement (dépassement de seuil programmable sur un ou plusieurs capteurs);
- (3) Envoi des données au serveur informatique;
- (4) Émission d'alertes par courriel en cas d'événement;
- (5) Écriture d'un registre des opérations.

6 Chapitre 1. Introduction

Le serveur doit effectuer les tâches suivantes :

- (6) Archivage des données brutes temporelles des accéléromètres dans une base de données;
- (7) Analyse modale automatique par méthode FDD (Frequency Domain Decomposition) [7] et archivage des résultats dans la base de données;
- (8) Envoi d'alertes en rapport à des évolutions anormales des résultats d'analyse modale;
- (9) Interface ergonomique pour les utilisateurs pour la configuration des paramètres ajustables et la présentation de la base de données, soit principalement :
 - (a) affichage en temps réel du flux de données des capteurs,
 - (b) enregistrement de données à la demande (avec ou sans analyse modale),
 - (c) consultation de la base des enregistrements,
 - (d) affichage des résultats des analyses modales et de l'évolution des propriétés dans le temps,
 - (e) reconfiguration des options du système de mesure.

Dans leur ensemble, ces objectifs spécifiques sont combinés à des objectifs généraux de facilité d'emploi et de maintenance. De plus, il est avantageux que le système puisse facilement s'adapter à d'autres ponts en minimisant la reprogrammation nécessaire.

1.5 PLAN DU RAPPORT

Incluant cette introduction, le présent rapport est organisé en six chapitres. Le chapitre 2 présente un état-de-l'art non exhaustif des diverses architectures utilisées dans des systèmes de télésurveillance actuellement déployés sur des structures réelles, notamment des systèmes de mesure des vibrations. L'architecture générale retenue pour le système développé est définie au chapitre 3. Le chapitre 4 détaille le développement du système d'acquisition in situ, basé sur un matériel National Instrument cRIO et sur le logiciel LabVIEW [26]. Le chapitre 5 détaille, quant à lui, l'architecture, les composantes et les fonctions concentrées sur le serveur informatique. Le bilan du développement du prototype et des recommandations pour la poursuite du développement du système intégré de télésurveillance et de détection d'endommagement de pont concluent le rapport au chapitre 6.

Chapitre

2

Architectures existantes de systèmes de télésurveillance

La quantification de l'endommagement d'un pont routier par des mesures de vibrations ambiantes est un des multiples sujets abordés par la discipline de la télésurveillance de l'état de santé des structures. Selon Zhou et al. [38], les premiers système de télésurveillance ont été installés dans les années 1990. On cherchait, pour les ponts d'une portée supérieure à 100 mètres, à connaître en permanence leur condition d'opération et leur état structural. Ce domaine nommé Structural Health Monitoring (SHM) dans la littérature anglophone est vaste. Il est d'ailleurs le sujet de plusieurs journaux scientifiques tel que *Journal of Structural Health Monitoring* (Éd. Sage), *Journal of Structural Control and Health Monitoring* (Éd. Wiley) ou encore *Journal of Civil Structural Health Monitoring* (Éd. Springer).

Plusieurs chercheurs à l'international s'intéressent à ce domaine d'étude. L'université de la Californie, au campus de San Diego, propose un cursus à la maîtrise spécifiquement orienté vers la télésurveillance des structures. Le SHM est mis en pratique sur les éoliennes, pipelines, ponts, barrages, bâtiments, stades, avions, trains, machines, etc. Pour les ponts, ces systèmes sont maintenant répandus partout à travers le monde, tel que rapporté par Inaudi [13].

D'une part, ce foisonnement s'explique par l'apport du système de télésurveillance sur la connaissance de l'état d'une structure. Les bénéfices possibles sont l'augmentation de la sécurité d'une structure en émettant des avertissements sur des pannes à venir, une gestion mieux ciblée des ressources pour l'entretien et la collecte d'informations qui mènent à des meilleures constructions [12, 18]. D'autre part, l'engouement autour de la discipline est justifié par le contexte particulier du vieillissement des infrastructures auquel la collectivité fait face [11, 24]. Les besoins justifiant les moyens, le SHM est un outil pertinent à mettre au service des ingénieurs.

Le travail décrit dans ce rapport est un projet de recherche appliquée, qui consiste à développer un prototype d'un système de logiciels répondant à la problématique d'extraction des propriétés modales d'un pont en temps réel et en continu. Précisons que le travail ne porte pas sur le choix des algorithmes de détection de l'endommagement, ni sur le choix des matériels d'acquisition de données, ni sur la disposition des capteurs à installer sur le pont. Ces éléments sont couverts par des activités de recherche distinctes à l'intérieur du projet R732.1. L'état-de-l'art qui est présenté dans ce chapitre est donc orienté vers l'étude des architectures formant les systèmes de télésurveillance existants. Spécifiquement, on cherche à identifier les constituants formant ces systèmes, c'est-à-dire les composants logiciels et les composants matériels. Ici, l'emphase sera portée sur les composants logiciels étant donné le contexte général de travail.

2.1 ARCHITECTURES

2.1.1 Principes

Les systèmes de télésurveillance de l'état de santé d'un pont sont constitués de trois parties fondamentales réalisant les fonctions d'acquisition de données, de stockage des données et de traitement des données. La figure 2.1 schématise un système de télésurveillance regroupant ces trois fonctions. Les données, mesurées dans cette figure pour deux structures distinctes, sont dirigées vers un système central d'archivage. En plus de disposer de cette capacité de stockage à long terme, celui-ci intègre tous les algorithmes et modules d'analyses nécessaires pour transformer les données brutes en information utile à la connaissance de l'état des structures concernées. La présence d'un portail permet de faire le suivi des structures et d'interfacer le système avec d'autres modules d'analyse.



Figure 2.1 – Schéma conceptuel d'un système de télésurveillance (adapté de [9])

Dans sa plus simple forme, le système de télésurveillance est dédié à une seule structure. Un système de télésurveillance nécessite l'intervention active d'un utilisateur, comme ceux présentés par Yang et al. [37] et par Lu [19], ou agit automatiquement comme celui présenté par Koo et al. [16]. La figure 2.2 montre le système développé par Yang et al. [37]. Pour celui-ci, l'utilisateur n'a accès qu'aux données brutes des capteurs stockés dans une base MySQL [28]. Cet accès passe par un navigateur web se connectant à un serveur informatique nommé Data Acquisition and Repository System (DARS). Ce système ne dispose pas d'algorithme de détection de l'endommagement, ni de script de calcul opérant sur les données contenues dans la base. L'utilisateur doit lui même télécharger et interpréter les données. La figure 2.3 montre le système développé par Koo et al. [16]. À la différence du système proposé par Yang et al. [37], celui-ci est équipé du logiciel Matlab [22] pour effectuer automatiquement des calculs sur les données enregistrées dans la base MySQL [28]. Une autre particularité du système est située au niveau de l'interaction entre l'utilisateur et le serveur informatique. En effet, celui-ci peut consulter des informations dans la base au travers de son navigateur web, en plus d'avoir un accès semi-direct aux données dans son environnement Matlab local. Cette conception est intéressante lorsque l'on veut développer de nouveaux traitements de données.



Figure 2.2 – Architecture d'un système manuel (adapté de [37])



Figure 2.3 – Architecture d'un système automatique (adapté de [16])

À l'autre bout du spectre de la complexité, le British Columbia Smart Infrastructure Monitoring System (BCSIMS) regroupe plus de dix ponts [14]. La figure 2.4 montre l'architecture du système de télésurveillance. Chacun de ces ponts est équipé de sa propre base de données et de son propre module d'analyse local. Tous les ponts sont dotés d'une connexion à une base de données globale et à un module d'analyse avancé sur lequel un deuxième ensemble d'algorithmes s'exécute hors ligne. Le BCSIMS comprend également une interconnexion avec 130 stations de télésurveillance du sol issues du Provincial Strong Motion Network, lui-même maintenu par le Pacific Geosciences Centre, ce qui lui permet d'intégrer des informations sur l'activité sismique de la région où se trouvent les ponts instrumentés.



Figure 2.4 – Architecture du BCSIMS (adapté de [14])

Pour les systèmes présentés jusqu'ici, l'architecture adopte la configuration client/serveur. Pour ce paradigme, il y a une distinction nette entre le demandeur et le producteur d'information. Le client, au travers d'un navigateur web, envoie des requêtes au serveur web, lui-même relié à une base de données et aux algorithmes spécialisés appropriés. Le client peut demander de l'information de nature générale, par exemple l'état d'un capteur, ou bien spécifique, par exemple le résultat d'un calcul d'endommagement, à propos de la structure.

Autrement, Smarsly et al. [31] proposent un système basé sur une configuration multi-agent. Cette architecture automatique, et apparemment autogérée, adopte le cadre logiciel JADE [2]. Pour ce système distribué, le dysfonctionnement d'un agent n'entraîne pas la faillite de l'ensemble. Cet avantage améliore grandement la robustesse d'un système de télésurveillance. L'architecture non orthodoxe proposés par Smarsly et al. [31] est montrée à la figure 2.5. L'architecture client/serveur et multi-agent sont comparables au niveau de l'acquisition et de l'archivage des données. La différence majeure concerne la mise en œuvre des activités de post traitement. Dans le cas du système multi-agent, une collection de petits logiciels répartie à travers un réseau participent conjointement aux objectifs encourus par le système de télésurveillance. Ces logiciels sont autonomes, réactifs et disposent d'habiletés sociales, dotant le système d'une forme d'intelligence artificielle. Pour l'architecture client/serveur, les logiciels sont confinés au sein du système d'acquisition et du serveur informatique. Une panne sur le matériel hébergeant les algorithmes de post traitement des données entraînera inévitablement une perte de fonctionnalité du système.



Figure 2.5 – Architecture multi-agent (adapté de [31])

2.1.2 Composants matériels

Sans exception, tous les systèmes présentés sont formés par deux matériels distincts, soit un serveur informatique et un système d'acquisition de données. Typiquement, ils sont reliés par un réseau internet.

Serveur informatique

Ce matériel collecte les données issues du système d'acquisition. Il dispose d'une capacité de mémoire dure très importante, puisque c'est sur ce matériel que les données sont stockées à long terme. Au niveau logiciel, le serveur informatique héberge un serveur web, une base de données et, selon le cas, des algorithmes de traitement de données. Le serveur informatique des systèmes présentés dans les références précédentes [14, 16, 19, 37] est hébergé sur leur campus universitaire respectif. Une autre avenue possible est l'hébergement commercial.

Système d'acquisition de données

Le système d'acquisition est localisé sur la structure surveillée. Ce matériel est très spécifique à la nature de la surveillance à exécuter. Il doit être choisi en fonction des types de capteurs installés sur le site. Le système in situ inclut les capteurs, le matériel de transmission (câbles ou antennes sans fils, par exemple), les dispositifs de conditionnement des signaux, l'unité d'acquisition et, en option, un ordinateur de bureau ou embarqué dans l'acquisiteur. Selon Celebi [5], le coût par canal est d'environ 4000 USD incluant le capteur, le câblage et l'installation. Un système de 12 à 18 canaux se chiffrerait à environ 50 000 USD selon cette estimation. Il est important de noter, cependant, que ces chiffres ne peuvent représenter qu'un ordre de grandeur très approximatif, puisqu'évidemment tout dépend des technologies de capteurs, de communication et d'acquisiteur

12 Chapitre 2. Architectures existantes de systèmes de télésurveillance

utilisées. Des unités d'acquisitions commerciales généralistes sont offertes par plusieurs entreprises telles que National Instruments, dSPACE, Labjack, Measurement Computing, Vishay, Omega, etc. Certaines entreprises spécialisées, telle que Wier-Jones & Terrascience Systems Ltd, offrent des solutions clé en main pour la réalisation complète d'un système de télésurveillance basé sur la mesure de vibrations ambiantes (fig. 2.6).



(a) Accéléromètre triaxial

(b) Unité d'acquisition

(c) Logiciel de surveillance

Figure 2.6 – Exemple de solution matérielle et logicielle offert par Wier-Jones

2.1.3 Composants logiciels

Les composants logiciels formant un système de télésurveillance sont répartis sur le serveur informatique et le système d'acquisition de données. Pour le système d'acquisition de données, les composants logiciels exécutent des tâches relative à la collecte des données en plus d'en faire une analyse sommaire (traitement du signal, dépassement d'un seuil, mesure aberrante, évolution prononcée d'une valeur, etc.). Pour le serveur informatique, les logiciels assurent le stockage à long terme et la présentation des données aux utilisateurs. S'il en est équipé, le serveur informatique procède à l'analyse des données par des logiciels spécifiques à cette tâche.

Ici, il n'est pas pertinent de faire une revue précise des logiciels équipant les systèmes d'acquisitions de données. En effet, ces éléments sont le plus souvent propres aux systèmes déployés, en plus d'être mal couverts par la littérature. De plus, le contexte général du projet fixe le matériel et le langage de programmation à utiliser pour le développement des logiciels du système d'acquisition, notamment le langage LabVIEW [26]. Les deux sous-sections suivantes se concentrent donc sur les éléments logiciels concernant l'interface web et la base de données présents dans le serveur informatique.

Interface web

Une manière efficace de présenter l'information issue d'un système de télésurveillance est de faire usage d'une interface web. Son efficacité tient du fait que seule l'information à présenter doit être gérée par le webmestre. Les utilisateurs accèdent à cette information via un navigateur web déjà présent sur leurs postes informatiques. Cette solution est celle qui a été déployée dans tous les systèmes présentés dans ce chapitre.

L'interface web est hébergée dans le serveur informatique et est prise en charge par un logiciel nommé *serveur HTTP*. La richesse de l'information présentée par l'interface web est variable et est propre au système de télésurveillance en question. La figure 2.7 présente un aperçu de l'interface web du réseau BCSIMS [14]. Plusieurs onglets répartissent logiquement l'information à destination de l'utilisateur. Notamment, on remarque la présence d'une représentation 3D pour simplifier l'identification et la sélection des capteurs. L'utilisateur peut tracer les valeurs brutes enregistrées par les capteurs et exécuter diverses analyses. Celui-ci peut également interroger le système sur le résultat d'analyses précédemment exécutées. Finalement, les données brutes et transformées sont disponibles au téléchargement si nécessaire. L'interface du réseau BCSIMS est probablement l'une des plus complètes et ergonomiques existant actuellement pour un système de télésurveillance de structures multiples. D'autres références présentent des interfaces web beaucoup plus courantes et simples, où sont essentiellement exposées les valeurs brutes enregistrées ou en temps réel de capteurs sélectionnés [16, 19, 31, 37, 38].

BRITISH COLUMBIA The Box Place on Earth	of rtation astructure	Civil Engin	neering ⁻	Natural Canada Ressou Canada	Resources rces naturelles				Earthquakes Canada <u>Séismes Canada About Us</u> <u>Help</u> Contact Us
Back to Main Page		Pitt Riv Station ID:	er Bridge 3233 DBID:	2 Longitude: W	V122.73 La	atitude: N49	25 Web Camera		
Station Events	Station History								
Select a time series below. You can then	recording in the list	Reports		Station Informa	tion Results				
- Preview data - Select channels fo	r display	Sensor Layou	ut	Sensor Statistics	Data Analysis	Results Dr	ift Analysis	Hysteresis	
 Select time intervi - View data series, Download the data 	Time ODS or Modes a recording								
2012-08-08 12:10:00	Scheduled								
2012-03-13 13:30:00	Scheduled								
2012-03-05 15:40:00	Scheduled								
2012-03-05 12:10:00	Scheduled							1 .	4
		Date		Top Channel Name	Bottom Channel Name	Drift	Threshold	liotes	
		08/08/2012 12	20-00 PM	IA-03	TA-03	1 18-07	0.011	Drift between two columns	
		05/03/2012 1:	40:00 PM	TA-03	TA-03	6E-08	0.011	Drift between two columns	
		05/03/2012 12	:10:00 FM	TA-03	TA-03	1.1E-07	0.011	Drift between two columns	
		05/03/2012 10	:00:00 AM	TA-03	TA-03	1.968E-05	0.011	Drift between two columns	

Figure 2.7 – Interface web riche en information du réseau BCSIMS (adapté de [14])

Base de données

Les systèmes de télésurveillance, fonctionnant en permanence, génèrent des quantités importantes de données. Par exemple, pour les ponts haubanés présentés par Zhou et al. [38], ce taux atteint 140 MB par heure. Le stockage des données devient donc un enjeu important. Soit on conserve simplement les données dans une multitude de fichiers indépendants. Soit on organise les données dans un ensemble unique et cohérent appelé base de données. Zhou et al. rappellent les avantages du stockage sur une base de données :

- le contrôle des données redondantes,
- l'amélioration de la cohérence des données,
- l'amélioration de l'intégrité des données,
- la possibilité de générer plus d'information avec la même quantité de données,
- le partage des données,
- l'amélioration de la rapidité et de l'accès aux données,
- l'amélioration de la productivité,

- la simplification de la maintenance grâce à l'indépendance des données,
- l'économie d'échelle.

Ces avantages combinés à la production massive de données rend évident le choix du stockage sur une base de données. Si la quantité de données devient trop importante, il est possible de distribuer la base sur plusieurs serveurs distincts, comme le propose Lu [19] avec l'organisation présentée à la figure 2.8. Avec une application à un viaduc, Lu précise que le système génère plus de 450 GB de données par année, justifiant par le fait même l'usage d'une base de données MySQL [28] distribuée. L'accès au système par l'utilisateur est géré par un navigateur web. Le flux de données et la gestion de la base sont totalement transparents pour l'utilisateur. Le système présenté par Lu est constitué des logiciels Apache [1] pour la gestion de l'interface web et des requêtes utilisateurs, MySQL [28] pour le stockage des données, et Matlab [22] pour la création de graphique et l'analyse des données. L'utilisateur doit lui-même gérer la mise à jour des données dans la base, mais l'auteur spécifie que cette tâche peut-être confiée à Windows sous la forme d'une tâche planifiée.



Figure 2.8 – Base de données distribuée sur plusieurs serveurs (adapté de [19])

2.2 EXEMPLES NOTABLES DE SYSTÈMES DE TÉLÉSURVEILLANCE DE PONTS

Dans la section précédente, le réseau de télésurveillance BCSIMS a été présenté. Avec une dizaine de structures de tailles variées connectées et une interface utilisateur très ergonomique, il s'agit probablement d'un des systèmes intégrés les plus avancés actuellement. Cette section présente brièvement trois exemples notables de déploiement d'instruments à des fins de télésurveillance sur des ponts importants :

- Sutong Bridge : pont haubané d'une travée principale de 1088 mètres construit dans la province de Jiangsu en Chine;
- Tsing Ma Bridge : pont suspendu d'une travée principale de 1377 mètres construit dans la province de Hong Kong en Chine;

 — Pitt River Bridge : pont haubané d'une travée principale de 190 mètres construit dans la province de la Colombie Britannique au Canada.

Les deux premiers exemples sont remarquables par l'étendu de leur système (nombre et variété des capteurs). Le troisième exemple, structure faisant partie du réseau BCSIMS, se rapproche du projet décrit dans ce rapport par sa dimension et sa finalité.

2.2.1 Sutong Bridge

La figure 2.9 montre un aperçu du système de télésurveillance équipant le Sutong Bridge. Celuici est constitué de plus de 1400 capteurs [15] et de sept unités principales d'acquisitions de données. Les unités d'acquisitions, *National Instruments - PXI*, sont relayées par une liaison à fibre optique à une station centrale externe au pont. Les capteurs sont de type anémomètres, thermomètres, jauges de déformation, accéléromètres, capteurs de déplacement, GPS, *weigh-in-motion*, capteurs de corrosion, capteurs élasto-magnétique, fibres optique, inclinomètres et hygromètres. Des caméras vidéos sont également présentes sur le pont.



Figure 2.9 – Disposition des capteurs et systèmes d'acquisitions du Sutong Bridge (adapté de [15])

2.2.2 Tsing Ma Bridge

La figure 2.10 montre un aperçu du système de télésurveillance équipant le Tsing Ma Bridge. Celui-ci est constitué de plus de 543 capteurs [36]. Ces capteurs sont de type anémomètres, thermomètres, jauges de déformation, accéléromètres, capteurs de déplacement, GPS, *weigh-in-motion* et inclinomètres. Des caméras vidéos sont également présentes sur le pont. Les capteurs sont connectés à six unités d'acquisitions de données basées sur une architecture du type ordinateur personnel. Ces unités sont interconnectées via un réseau à fibre optique.

2.2.3 Pitt River Bridge

La figure 2.11 montre une vue en élévation du Pitt River Bridge. Ce pont est équipé d'un système permanent pour la mesure des vibrations ambiantes. Dix-huit accéléromètres sont installés stratégiquement sur les câbles, les tours et le tablier, dont une partie est visible sur l'illustration de la figure 2.7. Un total de 46 canaux sont repartis sur deux appareils distincts, eux-même reliés à



Figure 2.10 – Disposition des capteurs et des systèmes d'acquisitions du Tsing Ma Bridge (adapté de [36])



Figure 2.11 – Vue en élévation du Pitt River Bridge (adapté de [35])

un serveur unique. Ce serveur permet de déclencher des alertes et autorise l'accès aux données en réponse à des requêtes issus du réseau BCSIMS [14]. Le système d'acquisition a été installé par l'entreprise Wier-Jones & Terrascience Systems Ltd de Vancouver. Les données sont quotidiennement téléchargées sur un serveur de stockage à long terme pour analyse future.

Chapitre

3

Approche de développement proposée du système de télésurveillance

À la lumière de l'état-de-l'art présenté au chapitre précédent, les systèmes de télésurveillance sont toujours formés de deux matériels distincts : le système d'acquisition pour la capture des données et le serveur informatique pour l'analyse et l'affichage de résultats. Excepté le système multi-agent de Smarsly et al. [31], les systèmes adoptent également tous le paradigme client/serveur. Tous les systèmes de télésurveillances présentés ci-dessus comptent sur le stockage de l'information dans des bases de données pour les raisons évoquées à la section 2.1.3.

L'architecture matérielle et logicielle du système mis en œuvre dans la présente étude se base donc sur ces mêmes fondements. Le système proposé est composé d'un matériel National Instruments CompactRIO (ou NI cRIO) pour l'acquisition des données sur le pont et d'un serveur informatique générique pour les fonctions de stockage et d'analyse des données, ainsi que de présentation et d'accès aux résultats via une interface web. La figure 3.1 illustre l'architecture matérielle et logicielle proposée pour la réalisation du système de télésurveillance, qui était initialement dédié au pont de la Rivière-aux-Mulets. Ce système répond aux objectifs énumérés à la section 1.4. Son usage, sa maintenance et sa programmation sont assurés par deux catégories d'utilisateurs. L'utilisateur classique se connecte à un site web via son navigateur local pour faire le suivi du pont sous surveillance et a un accès limité au site. Ce dernier lui donne la possibilité d'utiliser toutes les fonctions analytiques du système, mais pas de le reprogrammer :

- consulter l'information directement présentée par le site,
- demander de l'information spécifique contenue dans la base de données,
- déclencher un enregistrement de données à la volée à partir des données flottantes en temps réel contenues dans le système d'acquisition,
- observer les signaux en temps réel sur tous les canaux du système d'acquisition,
- modifier les horaires d'enregistrements automatiques et récurrents des données au niveau du système d'acquisition.

Par ailleurs, une catégorie de super-utilisateur a été créée, qui agit comme webmestre pour le serveur informatique et comme configurateur pour le système d'acquisition de données. Le superutilisateur a donc un accès total à toutes les composantes du système (modification de paramètres fondamentaux au niveau de la programmation de l'acquisiteur, retrait, ajout et mise à jour de données dans la base, etc.). Cet accès a été prévu pour établir un niveau de sécurité au niveau des fonctionnalités du système et devrait être réservé aux ingénieurs responsables de l'administration du système au Ministère.



Le serveur informatique est relié au réseau internet. Un ensemble de programmes y est exécuté pour répondre aux objectifs spécifiques propres à cette partie. Ces programmes sont :

- 1. Un serveur HTTP Apache [1] : Il répond aux requêtes de l'utilisateur issues de son navigateur web. Pour ce faire, le serveur héberge un site web écrit avec des balises statiques HTML et avec des commandes dynamiques PHP. Les balises HTML présentent invariablement l'information, par exemple le message de bienvenue de la page d'accueil. D'autre part, les commandes PHP permettent d'exécuter des actions retournant de l'information variable tel que :
 - des lectures et écritures dans la base de données,
 - l'exécution d'un calcul d'analyse modale,
 - des requêtes vers le système d'acquisition de données situé sur le pont.
- 2. La base de données MySQL [28] gérée à travers le logiciel MariaDB [20] : Huit tables sont stockées dans la base :
 - la table enregistrements : il s'agit des informations directement reliées aux enregistrements de données temporelles brutes, telles que la date et l'heure d'enregistrement ou le déclencheur de l'enregistrement.
 - la table analyses_modales : il s'agit des résultats des calculs d'analyses modales (fréquences, déformées, amortissements et degrés de complexité des modes) exécutés sur un enregistrement particulier.
 - les tables associées à des éléments de configuration (configurations, canaux, noeuds, capteurs) : il s'agit des détails du matériel d'acquisition in situ associés à un enregistrement donné, comme les fréquences d'échantillonnage, l'information sur les capteurs employés sur chaque canal d'acquisition (marque, sensibilité, no. série, etc.) et la disposition des capteurs sur le pont. Une nouvelle configuration peut être créée dans la base de donnée si une modification est apportée aux paramètres précédents, soit à distance par le super-utilisateur, soit matériellement in situ par le gestionnaire de l'ouvrage.
 - la table *journal* : il s'agit d'une table retraçant, par dates, toutes les modifications, événements particuliers ou commentaires des utilisateurs méritants d'être conservés pour l'historique du système.
 - la table *utilisateurs* : cette table contient la liste des utilisateurs autorisés à accéder au portail web du système, incluant les noms d'utilisateurs, mots de passe et niveau de privilèges (super-utilisateur ou non).

Dans l'approche développée, les données brutes sont stockées directement sur le système de fichier du serveur informatique. En effet, il n'y a pas lieu d'insérer dans la base de données ces grandes quantités de nombres qui n'ont que peu d'intérêt au format brut. Une telle démarche augmenterait sans raisons valables le volume de la base en n'apportant aucun bénéfice. Donc, seules les métadonnées associées aux enregistrements de données brutes sont écrites dans la base. Ces métadonnées discrètes ont des relations uniques avec les autres tables, enrichissant l'information stockées dans la base.

- 3. Un planificateur de tâches (planificateur Windows) : celui-ci exécute automatiquement différentes listes de tâches, qui sont définies en langage Python [29] et sont exécutées par l'interpréteur du même nom.
- 4. Le logiciel d'analyse modale Artemis Modal [33] : il permet d'extraire les propriétés modales à partir des enregistrements de données temporelles brutes. Le logiciel est piloté automatiquement et aidé dans son analyse par un script de commandes Python et Matlab [22] (interprétées par la librairie gratuite Matlab Runtime).

Le système d'acquisition de données à la figure 3.1 est constitué d'un matériel de type NI cRIO, présenté dans le rapport technique CRGP-2014/04 [10], des accéléromètres et des composants nécessaires à leur fonctionnement installés sur le site du pont. Globalement, le système d'acquisition de données est relié au réseau internet et collecte les vibrations ambiantes dans des fichiers d'enregistrements sauvegardés localement sur le CompactRIO. Comme le CompactRIO dispose d'un serveur FTP, ces enregistrements sont disponibles à tout client FTP externe. Le script Python s'exécutant sur le serveur informatique a la tâche de récupérer les enregistrements, puis de les supprimer du système d'acquisition pour éviter un débordement du disque local de l'appareil in situ.

Autant l'acquisiteur in situ que le serveur ont la capacité d'émettre des alertes courriels advenant un événement prédéfini. Par exemple, dans le cas d'une perte de connexion au réseau internet survenant sur le système d'acquisition, le serveur informatique émet un courriel aux utilisateurs pour les informer. Un événement sévère sur le pont (dépassement de seuil de capteurs) est également sujet à un message courriel, émis cette fois-ci par le système d'acquisition. Les éléments présentés cidessus permettent de constater que les deux principales composantes du système sont asynchrones et réactives. Elles s'exécutent parallèlement l'une de l'autre sans qu'il soit nécessaire pour l'une de connaître et de suivre l'état de l'autre.

Finalement, le système de télésurveillance développé répond aux objectifs de facilité d'usage et de maintenance. De fait, l'usage du système passe par un seul point d'entrée, à savoir le site web. Le flux des données est totalement transparent pour l'utilisateur. Concrètement, son interaction avec le système passe par un navigateur web et sa boîte de messagerie courriel. Pour la maintenance, un maximum de logiciels déjà adoptés par le MTQ sont employés. Il s'agit du logiciel d'analyse modale ARTeMIS Modal, du logiciel de programmation LabVIEW et du planificateur de tâches Windows. Les autres composants du système sont pris dans la collection des logiciels ouverts et sont donc totalement gratuits d'utilisation (Python, MySQL, Apache, Matlab Runtime). La duplication du système sur un nouveau pont entraîne donc des coûts supplémentaires seulement au niveau de l'instrumentation, car le serveur informatique reste le même, répondant ainsi à l'objectif de facilité d'adaptation à d'autres structures.

Chapitre

4

Système d'acquisition

Ce chapitre présente le prototype de système d'acquisition développé pour le système de télésurveillance de pont. Le matériel utilisé pour ce prototype est d'abord présenté et est comparé aux recommandations de matériel faites dans le rapport CRGP-2014/04 [10] pour un système qui serait effectivement déployé en permanence sur un pont. Les développements logiciels occupent ensuite la majeure partie de ce chapitre, notamment la description des fonctionnalités, de la programmation et de la mécanique de fonctionnement du VI (Virtual Instrument) développé sous LabVIEW [26].

4.1 MATÉRIEL

Le rapport CRGP-2014/04 [10] a établi les spécifications techniques minimales auxquelles doit répondre un système d'acquisition prévu pour un déploiement permanent sur le pont de la Rivièreaux-Mulets en vue de faire un suivi continu et en temps réel des propriétés dynamiques de l'ouvrage. Le tableau 4.1 rappelle ces spécifications minimum qui étaient essentiellement dictées par les besoins en rapidité de calcul pour l'application souhaitée afin de pouvoir traiter plusieurs tâches en temps réel en parallèle, par les besoins en capacités de stockage considérant la taille importante des fichiers générés et le besoin de stockage provisoire, par la précision requise sur les données et par les conditions environnementales exigeantes pour le fonctionnement en continu sur un pont. Considérant (i) les caractéristiques minimales du tableau 4.1, (ii) le choix qui avait été fait en 2014 de se tourner vers des accéléromètres de type piézoélectriques et vers une une plateforme programmable sous Labview et (iii) le matériel disponible à cette époque chez National Instrument, une recommandation d'achat pour l'équipement destiné au pont de la Rivière-aux-Mulets avait été émise. Celle-ci concernait un système NI CompactRIO, sur la base d'un contrôleur cRIO-9025 et d'un châssis cRIO-9112, dont les caractéristiques sont données au tableau 4.2.

Puisqu'il n'y a pas eu de déploiement sur le pont de la Rivière-aux-Mulets, aucun matériel n'a finalement été acheté. Le développement du prototype de système d'acquisition n'a donc pas été effectué directement sur le matériel recommandé, mais sur un matériel de substitution très similaire, disponible au laboratoire de structure de l'Université de Sherbrooke. Ce matériel est de type NI cRIO et se base sur un contrôleur-châssis cRIO-9074, dont l'architecture est en tout point identique au matériel recommandé. Les différences concernent essentiellement des performances moindres sur le matériel de développement, soit les vitesses de processeur, la capacité de stockage et la robustesse face à des conditions extrêmes. Cependant, comme la plateforme matérielle reste identique, tous les développements de VI effectués sur le matériel prototype et décrits dans ce chapitre pourront être directement portés sur le matériel de déploiement lorsque viendra le temps.

Ordinateur embarqué exécutant des programmes NI Labview				
Processeur	$\geq 400 \mathrm{MHz}$			
Mémoire vive	$\geq 256 \operatorname{Mo}$			
Capacité de stockage	$\geq 4 \mathrm{Go}$			
Connectivité TCP/IP via port Ethernet				
Nombre de canaux	44			
Conditionnement direct des signaux IEPE (si accéléromètres piézo)				
Compatibilité électronique complète avec les capteurs (alimentation et signaux d'entrée)				
Échantillonnage simultané de tous les canaux				
Fréquence d'échantillonnage	$\geq 200 \mathrm{Hz}$			
Résolution d'encodage	16 bits min.			
Températures opérationnelles	-40° C à $+50^{\circ}$ C			
Humidité ambiante	100% : considérer l'utilisation d'un boîtier ventilé			
Encombrement	modèle compact			

Tableau 4.1 – Spécifications requises d'un système d'acquisition pour un déploiement in situ permanent

Tableau 4.2 – Système d'acqu
sition NI c RIO recommandé dans le rapport CRGP-2014/04

Matériel	Caractéristiques				
cRIO-9025 : contrôleur temps réel embarqué					
Processeur	800 MHz				
Mémoire vive	$512\mathrm{Mo}$				
Capacité de stockage	$4\mathrm{Go}$				
cRIO-9112 : châssis avec micropro	cesseur $FPGA^{\dagger}$				
Emplacements pour modules	8				
Connectivité TCP/IP via por	t Ethernet				
NI 9144 : extension de châssis avec	c microprocesseur FPGA ^{\dagger}				
Emplacements pour modules	8				
Communication avec le châssi	s maître via port Ethernet				
NI 9234 (Qtés : 11) : module d'acc	quisition pour capteurs IEPE				
Canaux	4				
Fréquence d'échantillonnage	\leq 51.2 kHz				
Tension d'entrée	$\pm 5\mathrm{V}$				
Résolution	24 bits				
Échantillonnage simultané					
Filtres anti-repliement					
NI PS-15 : alimentation 24 VDC, 5	5A pour les châssis				
Températures opérationnelles	$-40^{\circ}C a + 70^{\circ}C$				
Humidité ambiante	85% max, sans condensation (boîtier ventilé nécessaire)				

[†] Microprocesseur FPGA (Field Programmable Gate Array) pour l'exécution de tâches en temps réel (élément de l'architecture NI cRIO).

L'ordinateur embarqué de tous les systèmes NI CompactRIO exécute le système d'exploitation en temps réel VxWorks[®] et compose la plateforme matérielle et logicielle du système d'acquisition développé dans ce projet. Le cRIO-9074 est alimenté en courant continu entre la plage 19 V à 30 V et consomme un maximum de 20 W (bloc d'alimentation NI PS-15). Deux ports permettent la



Figure 4.1 – Système d'acquisition utilisé pour le prototype

connectivité du cRIO-9074 au réseau Ethernet filaire. Le châssis loge un microprocesseur cadencé à 400 MHz pour l'exécution des programmes, et il est accompagné d'une puce FPGA (*Field Programmable Gate Array*) de 2 millions de portes logiques pour le prétraitement des signaux numériques. Le cRIO utilisé pour le développement dispose d'un disque de stockage de 256 Mo et d'une mémoire vive de 128 Mo. La figure 4.1 montre le matériel d'acquisition utilisé pour le prototype. Le châssis accepte jusqu'à huit périphériques d'entrées/sorties. La figure 4.1 montre le prototype équipé des trois périphériques NI-9239 disponibles au laboratoire de l'université de Sherbrooke. Les modules NI-9239 constituent une différence du prototype de développement par rapport au matériel recommandé pour installation in situ, car ils ne sont pas prévus pour conditionner des accéléromètres de type piézoélectrique. Ce sont des périphériques à quatre voies différentielles qui numérisent simultanément les signaux analogiques entre ± 10 V sur une précision de 24 bits. Au niveau logiciel, il sera très simple d'adapter le VI développé avec ce matériel pour le type précis d'accéléromètres qui seront installés sur le pont choisi. Tous les canaux des modules NI-9239 sont conditionnés par un filtre anti-repliement à la moitié de la fréquence d'échantillonnage. Cette dernière est programmable entre la plage de 1.613 kHz à 50 kHz selon l'équation :

$$f_e = \frac{12.8 \text{ MHz} \div 256}{n}$$
 (4.1)

où n est un entier compris entre 1 et 31. Les fiches techniques détaillées des principales composantes du système utilisé pour le développement du prototype sont fournies aux annexes A (cRIO-9074), B (cRIO-9239) et C (alimentation PS-15).

La figure 4.2 schématise l'architecture générale du CompactRIO. Les modules d'entrées/sorties sont reliés au microprocesseur temps réel via la puce FPGA. Pour utiliser le CompactRIO, il faut donc écrire deux programmes (VI), le premier pour le microprocesseur et le second pour la puce FPGA. National Instruments recommande la programmation du CompactRIO dans l'environnement LabVIEW [26]. LabVIEW est un environnement de programmation graphique, c'est-à-dire que le logiciel développé est un schéma constitué de plusieurs blocs reliés les uns aux autres par des liens transmettant les données. À la différence de la logique de programmation écrite linéaire et séquentielle habituelle, les schémas-bloc LabVIEW s'exécutent en suivant le principe du flux des données. Donc, un bloc représentant une fonction s'exécute seulement si toutes les données nécessaires sont disponibles en entrées et plusieurs blocs peuvent s'exécuter simultanément.

Le processus de développement de VI pour le CompactRIO comporte trois étapes. La première étape consiste à relier le CompactRIO à un ordinateur de bureau. Celui-ci doit être équipé de la suite



LabVIEW et des modules nécessaires au développement des routines à implémenter dans l'appareil. L'étape suivante consiste à programmer les routines à exécuter par le CompactRIO. Finalement, il suffit de compiler puis de téléverser les routines dans l'appareil. Au redémarrage, le CompactRIO exécute les routines de manière autonome. Alors, l'appareil peut être installé in situ sans nécessité de connexion à l'ordinateur de développement. Les logiciels s'y exécutant fonctionnent jusqu'à la mise hors-tension de l'appareil. Ils redémarrent automatiquement suite à la remise sous-tension.

4.2 CAHIER DES CHARGES ET GRAFCET DES LOGICIELS DÉVELOPPÉS

Tel que défini en introduction de ce rapport (section 1.4), le système d'acquisition du prototype développé doit répondre aux fonctions suivantes :

- (1) Enregistrement flottant continu des accéléromètres déployés;
- (2) Sauvegarde des données temporelles des capteurs :
 - (a) à la demande,
 - (b) automatiquement (dates et heures programmables),
 - (c) au déclenchement d'un événement;
- (3) Envoi des données au serveur informatique;
- (4) Émission d'alertes par courriel en cas d'événement;
- (5) Écriture d'un registre des opérations.

Les fonctions du système d'acquisition sont assurées par un programme principal (VI), un serveur de fichiers FTP et un fichier de configuration. Le VI a été l'objet de l'essentiel des efforts de programmation et sera décrit précisément aux sections 4.3.1 et 4.3.2. Le serveur de fichier FTP est une composante existante préprogrammée des CompactRIO et sera brièvement abordé à la section 4.4. Finalement, le fonctionnement du VI fait appel à un fichier de configuration qui sera présenté à la section 4.5.

Le VI répond aux fonctions (1), (2), (4) et (5) rappelées précédemment, tandis que la fonction (3) est en réalité assurée par le serveur informatique (chap. 5). Sa conception est basée sur le graphe fonctionnel de commande-étape-transition (GRAFCET) présenté à la figure 4.3. Pour rappel, "le



Figure 4.3 – GRAFCET du programme d'acquistion dans le CompactRIO

GRAFCET est un organigramme spécialisé, avec graphisme et règles particulières, utilisé pour décrire le cycle logique des systèmes automatisés..." [8]. De manière simplifié, le graphisme consiste en une séquence de boîtes reliées entre elles par des éléments de transition. De plus, une transition vers une nouvelle étape est franchie à la condition que la réceptivité soit vraie et que l'étape active soit celle immédiatement supérieure à la transition concernée. Par exemple, le GRAFCET du VI à la figure 4.3 montre que le passage à l'étape E10 est réalisé si la réceptivité *Init.fait()* est vraie et l'étape immédiatement active est *Init*, ou si la réceptivité *E34.fait()* est vraie et l'étape immédiatement active est *E34*. Certaines machines peuvent être directement programmées sur la base d'un GRAFCET, ce qui n'est pas le cas du CompactRIO. Pour cette étude, la présentation et les règles logiques du GRAFCET ont été utilisées à titre de support conceptuel pour les schémasbloc LabVIEW. Ce support et ses principes ont fortement influencé le développement du logiciel.

Pour une application à complexité modérée telle que celle présentée ici, on recommande de pré-concevoir les schémas-bloc à l'aide d'un state-diagram [25]. Un state-diagram représente graphiquement le fonctionnement d'une machine d'état. Il est l'équivalent dégénéré d'un GRAFCET. Ensuite, on recommande de décliner le state-diagram en machine d'état sur la base d'un modèle de conception d'application (application design pattern). Ce modèle est une architecture informatique permettant d'intégrer facilement les états de la machine et ses conditions évolutives. C'est selon ces recommandations qu'a été développé le logiciel de télésurveillance. Le modèle de conception suggéré est montré à la figure 4.4. Il s'agit d'une boucle tant que entourant une structure de conditions. La valeur de la condition pilote la structure et en détermine l'état actuel. À la fin des actions regroupées sous cet état particulier, la structure décide de la prochaine étape en fixant la nouvelle valeur de la condition. Cette dernière est passée à la prochaine itération via un registre à décalage. Cette progression cyclique, entrée-actions-sortie, règle tout le fonctionement du logiciel de télésurveillance.



Figure 4.4 – Modèle suggéré par National Instruments pour les machine d'état (adapté de [26])

Sur la figure 4.3, le double carré entourant l'étape *Init* indique le point de démarrage du logiciel. Suite à cet amorçage, le logiciel prend le contrôle du CompactRIO et réalise automatiquement les actions correspondant à chacune des huit étapes. La section suivante détaille l'ensemble des étapes du GRAFCET en montrant également le code LabVIEW associé. Les étapes *Init, E10* et *E20* sont étiquetées en tant que phases initiales, alors que l'état *E30* et les sous-états *E31, E32, E33, E34* et *E35* sont étiquetées comme fonctions principales avec déclencheurs.
4.3 DÉTAILS DE PROGRAMMATION DES VIS DÉVELOPPÉS

4.3.1 Phases initiales

Les étapes regroupées dans cette section servent à la mise en route de la machine d'état.

\mathbf{Init}

Sur toute la période de fonctionnement du logiciel de télésurveillance, cette étape n'est exécutée qu'une seule fois. Elle constitue le point d'entrée du programme. La figure 4.5 montre l'étape ainsi que la seule action qui y est rattachée, soit inscrire le démarrage du système dans le fichier journal du CompactRIO. Une fois fait, la machine d'état progresse automatiquement vers l'étape E10.



Figure 4.5 – Étape Init

La déclinaison LabVIEW de cette étape est montrée à la figure 4.6. Cette figure montre le détail exact du modèle de conception choisi, une boucle *tant que* entourant une structure de conditions. Différemment du modèle présenté à la figure 4.4, le passage d'une étape à l'autre n'est pas assuré par un registre à décalage, mais par une file d'attente. Cette dernière méthode est plus flexible et performante que le registre à décalage [3]. De plus, on remarque neuf registres à décalage placés dans la partie supérieure du bloc-diagramme. Ces registres permettent le passage des données d'une



Figure 4.6 – Programmation LabVIEW de l'état Init de la machine d'états

étape à l'autre. On remarque aussi que les actions à accomplir par la machine d'état à l'étape *Init* sont plus nombreuses que ce qui est proposé par le GRAFCET. Effectivement, le travail à accomplir dans l'environnement de la machine est plus détaillé que celui du graphe conceptuel. Notamment, la première action consiste à empiler dans la file d'attente la prochaine étape, qui sera *E10*. Ensuite, on note le démarrage du système et l'écriture du journal en accord avec le GRAFCET. Finalement, on exécute une réinitialisation de la puce FPGA.

E10

La figure 4.7 montre l'étape E10 qui consiste à lire le fichier de configuration défini par l'utilisateur, puis à assigner les valeurs qu'il contient aux variables internes du logiciel de télésurveillance. Deux conditions peuvent conduire à l'étape E10, soit la réceptivité Init.fait() ou E34.fait() avec, respectivement, l'étape active *Init* ou E34. Une reconfiguration des variables internes est donc opérée soit au démarrage initial du système, soit lorsqu'une modification des paramètres est explicitement demandée par l'utilisateur (cf description de l'état E34).



La déclinaison LabVIEW de l'étape E10 est montrée à la figure 4.8. Le prochain état de la machine est d'abord empilé dans la file d'attente, soit l'étape E20. Ensuite, comme l'indique le GRAFCET, la machine ouvre le fichier de configuration (section 4.5) et assigne les valeurs paramétrées tant par l'utilisateur que par le super-utilateur aux variables appropriées, notamment à l'étape E10 la fréquence d'échantillonnage et les durées d'enregistrement requises. Une fois les variables assignées, quatre files d'attente sont définies (buffers), qui réservent les blocs de mémoire vive nécessaires au cumul des données flottantes qui seront utilisées dans les états E31, E32, E33 et E35. Ces buffers sont indispensables pour assurer le transfert sans perte des données vers les processus respectifs, sachant que plusieurs des fonctions associées à ces quatre états font appel aux données de vibration du pont dans les minutes précédant leur déclenchement. Ce sont ces données qui sont stockées dans les files. Les tailles des quatre files de stockage et donc l'espace mémoire alloué sont différents selon la fonction à assurer. L'état E31 correspond à l'enregistrement de données suite à un événement sur le pont (dépassement de seuil par un capteur). Seules les quelques dernières minutes, voire les quelques dernières secondes, ont donc besoin d'être stockées pour récupérer les enregistrements immédiatement avant l'événement, alors que les données après l'événement pourront être traitées à la volée. L'état E32 correspond à l'enregistrement d'une certaine durée des données, programmé à partir d'une heure fixe. Comme les données nécessaires sont uniquement celles qui suivent le déclenchement, seul un très court buffer est nécessaire, par exemple 5 secondes, juste pour assurer qu'aucune donnée ne soit perdue par des retards de processus. Les données qui suivent le déclenchement de l'enregistrement peuvent être traitées la volée. L'état E33 correspond à un enregistrement de données déclenché manuellement par l'utilisateur. Comme 10 à 20 minutes de données sont généralement nécessaires à un utilisateur pour effectuer une analyse modale, le buffer requis doit être aussi de cette durée. Enfin, l'état E35 correspond à l'affichage en temps réel des



Figure 4.8 - Programmation LabVIEW de l'état E10 de la machine d'états

données de vibration sur un navigateur web. De façon à assurer un affichage fluide et sans perte des données qui transitent à travers le réseau, environ les 5 dernières secondes de données doivent être stockées de manière flottante. L'utilisation de plusieurs files en parallèle a pour effet de dupliquer les données des dernières minutes, mais cela s'avère être une solution très flexible et performante pour traiter en parallèle les boucles de réalisations des fonctions regroupées sous l'état E30.

L'étape E10 se finit sur deux actions (non représentées à la figure 4.8 pour des raisons de simplification du diagramme). D'abord, une écriture est effectuée au journal pour noter l'heure et la date de la reconfiguration du système. Puis, on assigne la valeur booléenne *faux* à la variable locale *surConfiguration*, qui correspond à la réceptivité chargée de déclencher la reconfiguration du système, comme on le verra à l'état E30.

Note : Pour rappel, le CompactRIO utilisé pour développer le prototype de système de télésurveillance dispose d'une mémoire vive de 128 Mo, qui est bien inférieure à celle du matériel qui doit être prévu pour un déploiement sur une structure réelle. Cette limitation a créé une contrainte sur les durées maximales possibles d'enregistrement des données flottantes sur le prototype de développement. Cela n'affecte cependant en rien la généralité de la programmation effectuée, puisque les tailles des files sont définies automatiquement à partir de durées spécifiées dans le fichier de configuration de l'utilisateur. L'estimation de la taille mémoire requise pour loger les données flottantes découle des paramètres d'acquisition. Ces paramètres sont la fréquence d'échantillonnage f_e , la durée requise de la mémoire flottante D, le nombre de canaux n_c et la taille mémoire de chaque échantillon b (en bits d'encodage). L'empreinte sur la mémoire vive M_f d'une file est :

$$M_f = f_e \times D \times n_c \times (b/8)$$

Le tableau 4.3 montre des tailles typiques de mémoire qui seraient nécessaires pour chacune des quatre files d'un système à déployer réellement. Évidemment pour un système déployé in situ, la

configuration de la taille des files doit être telle que leurs empreintes combinées est inférieure à la mémoire vive disponible.

	f_e (Hz)	D (s)	n_c	b (bits)	M_f (Mo)
surÉvénement $(E31)$	2000	60	40	24	14.4
surHoraire $(E32)$	2000	5	40	24	1.2
surUtilisateur $(E33)$	2000	1200	40	24	288.0
surStream $(E35)$	2000	5	40	24	1.2

Tableau 4.3 – Taille mémoire requise des files de données flottantes

E20

Lorsque la réceptivité E10.fait() est vraie et que l'étape active est E10, le logiciel active l'étape E20 montrée à la figure 4.9. L'action associée à cette étape consiste essentiellement à démarrer l'enregistrement flottant des données.



La déclinaison LabVIEW de cette étape est montrée à la figure 4.10. On empile d'abord le prochain état de la machine, soit l'étape *E30*. Ensuite et selon les variables définies à l'étape précédente, le logiciel d'acquisition des données est configuré et téléversé dans la puce FPGA. La configuration consiste à définir les canaux d'acquisition (nom, sensibilité et unité d'ingénierie dans lequel est exprimée la mesure), la fréquence d'échantillonnage réelle et la taille du buffer utilisé par le VI FPGA. La dernière action associée à l'état *E*20 consiste à démarrer le logiciel d'acquisition conformément au GRAFCET. Notons qu'il faut bien distinguer dans cette étape la fréquence d'échantillonnage réelle de la fréquence d'échantillonnage effective qui est celle obtenue dans les fichiers d'enregistrement à disposition des utilisateurs et qui est généralement de l'ordre de 200 Hz pour des analyses de ponts. La cadence d'échantillonnage minimale permise par le matériel FPGA est souvent beaucoup plus grande (1613 Hz dans le cas du matériel utilisé pour le prototype), ce qui oblige à procéder par la suite à un filtrage puis à un sous-échantillonnage des données avant de les enregistrer. Typiquement pour le pont de la Rivière-aux-Mulets, on utiliserait une fréquence d'échantillonnage réelle de 2000 Hz, puis on ne garderait qu'un point sur dix pour l'enregistrement des données à une fréquence effective de 200 Hz.

Le VI chargé dans la mémoire de la puce FPGA est illustré à la figure 4.11. La programmation FPGA est une technique moderne nécessitant des compétences pointues. Pour ce projet et selon les recommandations des développeurs LabVIEW, le VI du FPGA est basé sur la bibliothèque *NI CompactRIO waveform reference library* [27]. Pour le développement du prototype, aucune modification n'a été apportée au VI FPGA fourni par cette bibliothèque, qui permet de traiter 4 canaux. Une adaptation très simple de ce VI devra être effectuée pour l'application souhaitée à un pont



Figure 4.10 – Programmation LabVIEW de l'état E20 de la machine d'états



Figure 4.11 – VI exécuté par la puce FPGA

qui comptera davantage de canaux, qui consistera seulement à définir des blocs supplémentaires de canaux tels que ceux nommés Mod1/AI* au centre la figure 4.11 et à joindre leur sortie au même collecteur déjà présent. Plus de détails concernant le VI FPGA peuvent être obtenus en consultant la notice d'utilisation de la bibliothèque [27].

4.3.2 Fonctions principales et déclencheurs

Les étapes regroupées dans cette section correspondent aux fonctions principales réalisées par le système de télésurveillance. Elles sont constituées d'un état principal E30 qui mène à des sous-états E31 à E35 grâce à des déclencheurs.

E30

Lorsque la réceptivité E20.fait() est vraie et que l'étape active est E20, le logiciel de télésurveillance active l'état E30 montré à la figure 4.12. Cette étape consiste essentiellement à attendre la réalisation des réceptivités qui déclencheront les sous-états E31 à E35 et à gérer la réalisation de ces étapes, qui peuvent éventuellement s'exécuter simultanément. Au début de l'état E30, l'acquisition continue des données de vibrations par la puce FPGA du CompactRIO est un processus déjà en cours.



L'état E30 est l'état principal de fonctionnement du programme de télésurveillance, alors que tous les autres états ne sont que transitoires. Comme le montre la figure 4.3 du GRAFCET global, l'état E30 est en effet quasi perpétuel, car la réalisation des étapes E31, E32, E33 et E35 ramène automatiquement la machine en E30. L'unique chemin permettant de sortir de cet état est la réalisation de l'état E34, soit une demande de reconfiguration du système par l'utilisateur.

Contrairement à ce que suggère le GRAFCET, l'état E30 est en fait le plus complexe au niveau de la programmation LabVIEW. Une représentation simplifiée de l'ensemble de l'étape est montrée à la figure 4.13. La première action effectuée est d'empiler dans la file d'attente l'état suivant, soit E10 qui ne s'effectuera qui si une reconfiguration du système est demandée. Comme l'illustre la figure 4.13, l'étape E30 exécute ensuite de nombreuses boucles en parallèle, certaines de manière synchronisée, qui répondent aux fonctions principales du système de télésurveillance. L'état E30 a en fait pour fonction de gérer la réalisation de toutes ces boucles et de leurs fonctions internes, incluant leur déclenchement conditionnel, leur synchronisation (si nécessaire) et le transfert des données nécessaires entre les boucles. La figure 4.13 montre huit boucles tant que en parallèle :

- La boucle *cueillette et empilement des données* est chargée de stocker temporairement les données flottantes des capteurs dans les quatre files d'attentes créées à l'état *E10*.
- Les deux boucles Répartition interne et Répartition externe ont pour rôle de gérer la détection des réceptivités déclenchant les étapes E31 à E35. La première boucle détecte les événements déclencheurs liés à des événements internes au système d'acquisition (réceptivités surÉvénement et surHoraire), tandis que la seconde détecte les événements associés à des demandes externes venant de l'utilisateur (réceptivités surUtilisateur, surConfiguration et surStream).



Figure 4.13 – Simplification de la programmation LabVIEW de l'état E30 de la machine d'états

- Les trois boucles Enregistrement servent à réaliser les fonctions d'enregistrement des données des capteurs dans un fichier pour stockage permanent : enregistrement lié au dépassement d'un seuil sur les capteurs (étape E31, surÉvénement), enregistrement planifié (étape E32, surHoraire) ou enregistrement à la demande de l'utilisateur (étape E33, surUtilisateur).
- La boucle Serveur de données exécute toutes les fonctions impliquant une communication du CompactRIO soit avec le serveur de données, soit avec le navigateur web d'un utilisateur. C'est notamment cette boucle qui assure l'étape E35 pour la diffusion en temps réel des données sur le réseau en vue de leur affichage sur l'ordinateur distant d'un utilisateur. La boucle Serveur de données est assistée dans cette dernière fonction par la boucle Diffusion, chargée de conditionner adéquatement les données des capteurs avant leur diffusion sur le réseau.

Le fonctionnement de toutes les boucles est détaillé dans les prochains paragraphes. La figure 4.13 illustre de manière simplifiée les liens de synchronisation qui existent entre les boucles sans s'encombrer du réseau de fils LabVIEW de la programmation complète. Il existe différentes façon de gérer la synchronisation des processus dans Labview. Dans le VI développé, les synchronisations sont assurées exclusivement par les fonctions d'occurrences de LabVIEW. Celles-ci sont illustrées de façon conceptuelle à la figure 4.14. La logique d'utilisation des occurrences se fait en trois étapes. Une liaison d'occurrence est d'abord créée avec la fonction obtenir une occurrence (fig. 4.14a). Ensuite, lorsque le code rencontre la fonction attendre une occurrence (fig. 4.14c), il bloque l'exécution de toutes les fonctions suivantes jusqu'à ce que l'occurrence soit activée par la fonction générer une occurrence (fig. 4.14b) ou que le délai d'attente de l'occurence soit atteint. Ainsi, pour synchroniser deux boucles, il suffit de générer l'occurence au démarrage de la boucle maîtresse et de conditionner l'exécution des fonctions de l'autre boucle par l'attente de cette occurence. Les huit boucles de l'étape E30 montrées à la figure 4.13 sont synchronisées avec cinq liaisons d'occurrences distinctes. La cadence générale des boucles est maîtrisée par l'occurrence activée par la boucle cueillette et empilement des données, visible dans le code LabVIEW de la figure 4.15. Les autres occurrences seront explicitées dans les prochains paragraphes conjointement avec les étapes associées.



(c) Attendre une occurrence

Figure 4.14 – Fonctions LabVIEW de gestion des occurrences (adapté de [26])

Cueillette et empilement des données : La figure 4.15 montre la programmation LabVIEW de la boucle de cueillette et empilement des données à l'intérieur de l'état E30. Comme expliqué précédemment, on remarque à la figure que la boucle génère une occurence lorsqu'elle débute, qui



Figure 4.15 – Programmation de l'état E30 : boucle de cueillette et empilement des données

sert à synchroniser certaines des autres boucles de l'étape E30. Principalement, la boucle permet de transférer les données enregistrées par la puce FPGA du CompactRIO dans l'environnement en temps réel du VI de télésurveillance. Une fois dans cet environnement, les données sont empilées et dupliquées dans les quatre files indépendantes crées à l'étape E10 et qui correspondent aux quatre consommateurs de données, soit les processus nécessitant les données de vibrations ambiantes dans les étapes d'enregistrement E31, E32 et E33, et dans l'étape E35 de diffusion en temps réel des données vers le navigateur web de l'utilisateur. L'empilement des données dans les quatre files s'effectue par paquets de données correspondant à 0,5 secondes des signaux, si bien que la boucle est répétée deux fois par secondes. À la figure 4.15, on note que la boucle ne s'arrête que lorsque la réceptivité surConfiguration est détectée, donc lorsqu'un utilisateur demande la reconfiguration du système. Dans ce cas, à la sortie de la boucle, les quatre files de stockage flottant des données sont détruites et le processeur FPGA est arrêté.

Déclencheurs : Si on regarde le GRAFCET (fig. 4.3), un total de cinq déclencheurs ou réceptivités font évoluer la machine d'état momentanément hors de l'état E30 vers les sous-états E31 à E35. Les étapes E31, E32, E33 et E35 peuvent être simultanées et ramènent toutes la machine à l'état E30, alors que l'étape E34 s'exécute seulement une fois toutes les autres étapes inactives, et elle met fin à l'état E30. Pour permettre ce comportement et ne pas bloquer la détection des récep-

tivités pendant l'exécution des fonctions, la détection des réceptivités doit être traitée de manière indépendante dans des boucles dédiées. Il s'agit des deux boucles *Répartition interne* et *Répartition* externe, et leur programmation est montrée aux figures 4.16 et 4.17. Deux boucles différentes ont été créées pour des raisons purement organisationnelles du code, permettant de séparer le traitement des réceptivités surÉvénement et surHoraire, qui sont déclenchées automatiquement par le système lorsque certaines conditions sont réunies, et le traitement des réceptivités surUtilisateur, surConfiguration et surStream, déclenchées de l'extérieur par l'utilisateur. Ces boucles sont cadencées par l'occurence générée par la boucle cueillette et empilement des données si bien que l'état des réceptivités est vérifié 2 fois par seconde. Lorsqu'une réceptivité devient vraie une occurence est générée, ce qui active la boucle de réalisation de la fonction correspondant à l'état E31, E32, E33 ou E35, selon la réceptivité détectée. Les deux boucles de vérification des réceptivités sont interrompues par la détection de la réceptivité surConfiguration pour permettre la sortie de l'état E30. Les mécanismes de transition à l'état vrai de chacune des réceptivités sont détaillés dans les prochains paragraphes avec la description des fonctions auxquelles elles sont associées.



Figure 4.16 – Programmation de l'état E30 : boucle Répartition interne de gestion des réceptivités

E31

Au niveau du GRAFCET, la figure 4.18 montre que lorsque la réceptivité surÉvénement est vraie et que l'étape active est E30, le logiciel de télésurveillance active l'étape E31. La réceptivité surÉvénement correspond au dépassement d'un seuil d'accélération prédéfini dans le fichier de configuration sur un des canaux. Les actions associées à cette étape sont (1) de sauvegarder les données de la mémoire flottante associée dans un fichier d'enregistrement, correspondant à des durées de signaux avant et après détection de l'événement qui sont définies dans le fichier de configuration; (2) d'émettre une alerte courriel à la liste des utilisateurs sélectionnés; (3) de noter l'événement dans le fichier journal local du CompactRIO. Une fois toute ces étapes complétées, la réceptivité E31.fait() est vraie et la machine d'état retourne à l'état E30.

En langage LabVIEW, les trois actions énumérées ci-dessus se transposent dans le schéma-bloc de la figure 4.19. L'étape *E31* est synchronisée par la boucle gestion des réceptivités *Répartition interne* (fig. 4.16). Au dépassement d'un seuil, la variable *surEven* est passée à l'état *vrai* au niveau du VI FPGA, où s'effectue directement la surveillance du niveau des signaux. La figure 4.16 montre que cette variable pilote une structure conditionnelle, qui remet à zéro un compteur et génère une



Figure 4.17 – Programmation de l'état E30 : boucle Répartition externe de gestion des réceptivités



Figure 4.18 - Étape E31

occurrence. L'occurrence générée est détectée par la boucle de la figure 4.19, ce qui active les fonctions internes de cette boucle qui correspondent aux actions de l'étape *E31*.

Un nouveau fichier d'enregistrement est d'abord créé. Une boucle est ensuite chargée de désempiler les blocs de données de la file *surEven* (cela consiste à retirer un à un les plus vieux éléments de la file) et de les écrire dans le fichier. Le désempilement s'effectue bloc par bloc dans une boucle qui se poursuit tant que le nombre de blocs requis (correspondant à une durée définie dans le fichier de configuration) n'est pas atteint. Ainsi, les éléments déjà présents dans la file sont enregistrés, qui correspondent à une certaine durée de signaux avant le déclenchement de l'événement, ainsi que les nouveaux éléments de la file traités à la volée qui correspondent aux signaux après l'événement. Une action supplémentaire importante est effectuée sur chaque bloc entre le désempilement et l'écriture dans le fichier, soit le sous-échantillonnage des signaux. Tel qu'expliqué à la section 4.3.1, ceux-ci arrivent en effet du FPGA à la fréquence d'échantillonnage réelle f_e (par ex., 2000 Hz), qui est très supérieure à la fréquence d'échantillonnage effective utile choisie par l'utilisateur selon le pont étudié (par ex., 200 Hz). Pour ne pas générer des fichiers d'enregistrements de tailles déraisonnables



Figure 4.19 – Programmation de l'état E30 : boucle Enregistrement E31 d'écriture sur événement

à stocker ensuite de manière permanente dans la base de données, les signaux sont donc d'abord adéquatement filtrés (filtre passe-bas pour éviter les problèmes de repliement de fréquence), puis sous-échantillonnées de manière à obtenir des données échantillonnées à la fréquence utile définie par l'utilisateur dans le fichier de configuration.

Parallèlement à l'écriture du fichier précédemment décrite, la boucle de la figure 4.19 note l'événement dans le fichier journal du CompactRIO, puis une alerte courriel est envoyée à une liste d'utilisateurs. Finalement, le fichier créé est fermé, puis sauvegardé sur le disque local de l'appareil. Les figures 4.16 et 4.19 montrent qu'en cours d'enregistrement, la nouvelle détection d'un événement se répercute par la remise à zéro du compteur de blocs qui sert à arrêter l'écriture des données. Ce comportement permet de capturer de façon continue un événement s'étalant sur une durée indéterminée. La figure 4.19 montre que la boucle est interrompue lorsque la réceptivité *sur-Configuration* devient vraie, donc lorsqu'une reconfiguration du système est demandée. Cependant, la programmation est telle que le processus d'enregistrement en cours, s'il en existe un, sera d'abord mené à son terme et ne sera pas interrompu.

Notons que la programmation actuelle de la fonction entraîne l'activation de l'étape *E31* à chaque dépassement de seuil par un des capteurs. Ce comportement trivial n'est pas toujours souhaitable, car il peut parfois mener à l'émission de très nombreuses alertes inutiles aux utilisateurs. Le critère de déclenchement de la fonction pourra facilement être raffiné selon les besoins des utilisateurs du Ministère, par exemple en nécessitant que plusieurs capteurs franchissent simultanément leur seuil.

E32

La figure 4.20 montre que lorsque la réceptivité surHoraire est vraie et que l'étape active est E30, le logiciel de télésurveillance active l'étape E32. La réceptivité surHoraire correspond à la coïncidence entre un temps horaire programmé par l'utilisateur et la valeur de l'heure actuelle. Ainsi des enregistrements à horaires fixes ou réguliers peuvent être demandés par un utilisateur. La seule action associée à cette étape est de sauvegarder les données de la mémoire flottante associée dans un fichier d'enregistrement pour la durée définie par l'utilisateur. Une fois cette action complétée, la réceptivité E32.fait() est vraie et la machine d'état retourne à l'étape E30.



Figure 4.20 - Étape E32

En langage LabVIEW, l'état E32 se transpose dans le schéma-bloc de la figure 4.21. L'étape E32 est synchronisée par la boucle de gestion des réceptivités *Répartition interne* (fig. 4.16). La figure 4.16 montre que trois fonctions comparent l'heure actuelle du CompactRIO à celui de trois cases horaires différentes venant du fichier de configuration de l'utilisateur. S'il y a coïncidence entre le temps actuel et l'une ou l'autre de ces valeurs, l'occurence associée est générée, ce qui a pour effet d'activer les fonctions de l'état E32 dans la boucle de la figure 4.21. Sur cette figure, on voit que la fonction d'enregistrement des données dans un fichier suit exactement la même procédure que celle expliquée en détail pour l'étape E31: ouverture d'un fichier; désempilement bloc par bloc des données de la file *surHora* qui sont filtrées, sous-échantillonnées, puis enregistrées, tant que le nombre de bloc nécessaires pour obtenir un enregistrement de la durée souhaitée (définie dans le fichier de configuration) n'est pas atteint; fermeture du fichier puis stockage sur le disque



Figure 4.21 – Programmation de l'état E30 : boucle *Enregistrement E32* d'écriture sur horaire programmé

dur du CompactRIO. Notons que pour l'étape E32, la détection d'une nouvelle réceptivité durant un enregistrement en cours n'a aucun effet. À l'instar de l'état E31, la figure 4.21 montre que la boucle est interrompue lorsque la réceptivité *surConfiguration* devient vraie, mais attend que l'enregistrement en cours, s'il y en a un, soit achevé.

Les trois possibilités de programmation d'horaire offertes à l'utilisateur sont exprimées en code CRON à la figure 4.16. Ce code permet de synthétiser facilement l'information sur l'heure, le jour, le mois, l'année ou encore le jour de la semaine désirés pour l'enregistrement, tout en contenant l'information sur une éventuelle répétition horaire (section 5.4.6). Ainsi un horaire d'enregistrement régulier peut être facilement programmé (par exemple, chaque mercredi à midi).

E33

Lorsque la réceptivité sur Utilisateur est vraie et que l'étape active est E30, le logiciel de monitoring active l'étape E33, comme le montre la figure 4.22. La réceptivité sur Utilisateur correspond à la demande manuelle d'un utilisateur, via le site web d'interface du système (section 5.4.2), d'enregistrer sur le champ une certaine durée de données définie dans le fichier de configuration. Les actions associées à cette étape sont de sauvegarder les données de la mémoire flottante associée dans un fichier d'enregistrement et de noter l'événement dans le fichier journal local du CompactRIO. Une fois toute ces étapes complétées, la réceptivité E33.fait() est vraie et la machine d'état retourne à l'étape E30.



Figure 4.22 - Étape E33

En langage LabVIEW, les actions énumérées ci-dessus se transposent dans le schéma-bloc de la figure 4.23. L'étape E33 est synchronisée par la boucle de gestion des réceptivités *Répartition externe* (fig. 4.17). La figure 4.17 montre qu'au passage à l'état *vrai* de la variable *surUtil*, l'occurrence correspondante est générée, ce qui active les fonctions de l'état E33 dans la boucle de la figure 4.23. Le passage à l'état *vrai* de la variable *surUtil* s'effectue lors du décodage de la demande de l'utilisateur dans la boucle de communication détaillée plus loin (fig. 4.26). La fonction d'enregistrement des données fonctionne de manière identique à celle des états E31 et E32: ouverture d'un fichier; désempilement bloc par bloc des données de la file *surUtil* qui sont filtrées, sous-échantillonnées, puis enregistrées, tant que le nombre de bloc nécessaires pour obtenir un enregistrement de la durée souhaitée n'est pas atteint; fermeture du fichier puis stockage sur le disque dur du CompactRIO. È n parallèle, la boucle procède à l'écriture de la demande dans le journal du CompactRIO. À nouveau, la boucle est interrompue par la détection de la réceptivité *surConfiguration*, qui met fin au processus seulement lorsque l'enregistrement en cours est terminé.

Comme le montrera la section 5.4.2 au chapitre 5, il prévu de modifier le code de la figure 4.23 pour l'implémentation d'une option intéressante pour l'utilisateur. Plutôt que d'enregistrer des signaux correspondant à une durée fixe avant sa demande, celui-ci pourra ajuster la durée des signaux requis avant et après la demande à partir de l'interface web. Cette fonctionnalité n'est cependant



Figure 4.23 – Programmation de l'état E30 : boucle Enregistrement E33 d'écriture à la demande

pas encore implémentée en raison des importantes modifications requises au code LabVIEW. Entre autres, la fonction de désempilement devra commencer par détruire les données trop anciennes par rapport à la demande plutôt que de les écrire dans le fichier, et des informations supplémentaires devront compléter le message envoyé par le navigateur web à l'acquisiteur.

E34

La figure 4.24 montre que lorsque la réceptivité surConfiguration est vraie et que l'étape active est E30, le logiciel de télésurveillance active l'étape E34. La réceptivité surConfiguration correspond à la demande d'un utilisateur, via le site web, d'une reconfiguration de l'appareil d'acquisition selon de nouvelles valeurs de paramètres. Les actions associées à cette étape sont de mettre à jour le fichier de configuration selon les nouveaux paramètres transmis depuis le serveur par l'utilisateur, d'arrêter l'acquisition des données par le CompactRIO suite à la complétion des étapes E31, E32et E33 si elles étaient actives, et de noter la demande de reconfiguration dans le fichier journal local du CompactRIO. Une fois ces étapes complétées, la réceptivité E34.fait() devient vraie, et la machine d'état retourne à l'étape E10, où la reconfiguration des variables du système est réellement effectuée.



Figure 4.24 - Étape E34

42 Chapitre 4. Système d'acquisition

Le programme LabVIEW correspondant à l'étape E34 est implicite et très simple, puisque la seule action à exécuter consiste en fait à arrêter toutes les boucles sous-jacentes à l'étape E30. La figure 4.13 montre que lorsque la variable booléenne surConfiguration activée à la demande de l'utilisateur via le site web passe à l'état vrai, la condition d'arrêt des huit boucles parallèles de l'état E30 est réalisée. Comme expliqué précédemment, chacune des boucles mène à son terme l'action en cours avant de s'arrêter. De plus, à la sortie de la boucle de cueillette et d'empilement des données (fig. 4.15), les files de données flottantes sont détruites et le VI FPGA est arrêté. La fin de toutes les fonctions de la structure de condition de l'étape E30 entraîne la machine d'état hors de cet état et la ramène dans l'état suivant empilé, soit l'étape E10 (dans laquelle la boucle de répartition externe (fig. 4.17), mais elle est déclenchée dans la boucle de communication du CompactRIO (fig. 4.26) après décodage du message correspondant envoyé par le navigateur web de l'utilisateur. Préalablement à l'envoi de ce message, la mise à jour du fichier de configuration se fait via le serveur FTP local du CompactRIO, qui reçoit un nouveau fichier et qui écrase l'ancien. Cette opération de mise à jour de fichier ne dépend pas du VI LabVIEW de l'acquisiteur.

E35 et boucle de communication Serveur de données

La figure 4.25 montre que lorsque la réceptivité surStream est vraie et que l'étape active est E30, le logiciel de télésurveillance active l'étape E35. La réceptivité surStream correspond à la demande d'un utilisateur, via le site web, de visualiser sur son navigateur web les données mesurées en temps réel sur la structure (la diffusion en temps réel de données à travers un réseau s'appelle streaming en anglais). Quatre graphes pouvant afficher des capteurs différents sont à disposition pour cela sur l'interface web (section 5.4.2). Les actions associées à cette étape sont d'établir une connexion entre le navigateur web et le CompactRIO avec la demande appropriée, de conditionner adéquatement les signaux en temps réel réclamés par l'utilisateur, d'émettre les données vers le navigateur de l'utilisateur et de les afficher.



Figure 4.25 - Étape E35

L'étape E35 est d'un concept simple, mais d'une programmation pratique ardue. En effet, les composantes logicielles à programmer, tant au niveau de la page web que du système d'acquisition, sont nombreuses, le synchronisme et la réactivité doivent être maintenus entre les actions de l'utilisateur et le comportement conjoint du système d'acquisition et du site web. La programmation LabVIEW de l'étape E35 est constituée de la boucle de communication Serveur de données et de la boucle Diffusion. La boucle de communication assure également la transmission des demandes de l'utilisateur pour l'acquisition instantanée des données (étape E33) et pour la reconfiguration du système (étape E34). La figure 4.26 illustre l'ensemble des composantes et des liaisons impliquées dans le processus de communication, depuis l'intervention initiale de l'utilisateur jusqu'à l'affichage des données sur son navigateur web. La figure montre le cas d'une demande d'affichage en temps



Figure 4.26 – Illustration des composantes et liaisons impliquées dans la diffusion en temps réel des données

réel des signaux (étape E35), mais la plupart des éléments sont identiques pour traiter une demande de l'utilisateur d'enregistrement instantané des données ou de reconfiguration du système. Tel que présenté par la figure 4.26, huit phases majeures prennent places lors de la réalisation de l'étape E35:

- 1. À l'aide de son navigateur web, l'utilisateur charge la page du site de télésurveillance.
- 2. L'utilisateur effectue une demande de connexion au système d'acquisition à travers un bouton de la page web (2). Dans le cas d'une demande de reconfiguration, la connexion s'effectue automatiquement à l'appui du bouton Mise à jour du système dans l'onglet Configuration du site web (cf chap. 5, fig. 5.10). Pour une demande d'enregistrement instantané ou un affichage en temps réel des données, l'utilisateur appuit sur le bouton Connecter sous l'onglet Vibrations Ambiantes du site web (chap. 5, fig. 5.5). Pendant ce temps, le programme LabVIEW, qui se trouve dans l'état E30, exécute toutes les 250 ms la première phase de la boucle de communication (2.bis), soit la vérification d'une demande de connexion venant d'un navigateur web. Les deux éléments combinés activent, dans l'environnement JavaScript sous-jacent à la page web, une routine menant à la connexion entre le navigateur web et le CompactRIO. La liaison est du type Ethernet filaire via le protocole TCP/IP standard, mais un objet de communication innovant, nommé WebSocket, est le messager entre les deux appareils. Ce messager permet une communication bidirectionnelle et asynchrone. Le modèle de communication intégrant le WebSocket positionne les deux appareils au même niveau, c'est-à-dire qu'ils sont tous les deux à la fois réactifs et pro-actifs. De plus, le trafic de données est grandement réduit par la faculté de transmettre seulement l'information nouvelle et de manière compacte en utilisant des messages JavaScript Object Notation (JSON).
- 3. Le navigateur web envoit au CompactRIO le message correspondant à la demande de l'utilisateur : soit la demande de transmettre les données en temps réel collectées dans la file surStream accompagnée de la liste des quatre canaux requis (E35), soit la demande de procéder à un enregistrement immédiat des données de la file surUtil (E33), soit de reconfigurer le système (E34).
- 4. Le CompactRIO réceptionne le message issu du navigateur au format JSON et il le décode pour en extraire la nature et, selon le cas, les autres données pertinentes. Une structure de condition effectue alors l'action appropriée selon les trois types de messages possibles : activation de la réceptivité surUtilisateur, activation de la réceptivité surConfiguration ou activation de la réceptivité surStream avec attribution des variables spécifiant les canaux à diffuser. Dans les deux premiers cas, le reste du programme LabVIEW de l'étape E30 se charge d'effectuer les actions requises décrites dans les paragraphes précédents, et la boucle de communication ferme la connexion. Les prochaines phases décrites ci-dessous ne concernent donc que l'étape E35.
- 5. La figure 4.17 de la boucle de gestion des réceptivités *Répartition externe* montre que l'activation de la réceptivité surStream génère l'occurence correspondante, ce qui a pour effet d'activer les fonctions de la boucle *Diffusion* visible à la figure 4.27. Cette boucle conditionne adéquatement les signaux avant leur diffusion en désempilant les données flottantes accumulées dans la file surStream deux blocs par deux blocs (donc par paquets d'une seconde pour des raisons de fluidité d'affichage sur le navigateur) et en les sous-échantillonnant à 50 Hz. Les données étant destinées purement à de l'affichage, il est inutile ici de les filtrer avant le sous-échantillonnage. La fréquence de 50 Hz est très suffisante à cette fin tout en évitant d'encombrer inutilement la bande passante du réseau. Les signaux conditionnés sont ensuite empilés dans une nouvelle file réduite Stream d'une durée égale à la file originale surStream.



Figure 4.27 – Programmation de l'état E30 : boucle Diffusion de préparation des données pour le streaming

Notons que la durée de quelques secondes de ces files a pour effet que les données affichées sur le navigateur web ne sont pas à strictement parler en temps réel, mais accusent un retard de quelques secondes. Ce mécanisme participe également à résoudre les problèmes de retard dans les communications sans affecter l'affichage des données au niveau de la page web. La boucle *Diffusion* est programmée pour s'arrêter automatiquement après 5 minutes de diffusion de données dans le but d'éviter une utilisation abusive de la bande passante, lorsqu'un utilisateur, par distraction, laisse tourner le *streaming* sans l'utiliser réellement.

- 6. La boucle de communication *Serveur de données* procède alors au désempilement de la nouvelle file *Stream* bloc par bloc à l'intérieur d'une boucle infinie. Elle sélectionne uniquement les données des quatre graphes requis et les écrit dans un message au format JSON pour les envoyer vers le navigateur web via le WebSocket. Afin de préserver la bande passante, il est important de noter que si l'utilisateur ne spécifie aucun canal à afficher sur l'un des graphes, alors le message envoyé pour ce graphe reste vide.
- 7. Tout au long du processus, le WebSocket du navigateur de l'utilisateur est à l'écoute des messages entrants. Lorsque les données arrivent, une routine extrait les valeurs du message JSON et les répartit sur les graphiques appropriés de la page web.
- 8. Ces graphiques sont générés à l'aide de la librairie JavaScript Flot [17]. D'autres fonctions mettent à jour les données contenues dans les piles propres à chaque graphes (section 5.4.2). La page est rafraichie automatiquement sur la base d'une fonction spécialisée cadencée à la vitesse de réception des messages émis par le CompactRIO. Visuellement, cela permet de faire défiler les données en temps réel sur les graphes affichés sous l'onglet Vibrations Ambiantes du site web.

Remarque : LabVIEW ne supporte pas nativement les WebSockets. Pour les utiliser, il faut installer une API spécialisée. Plusieurs entreprises ont développé des versions propriétaires de cette API. Celle utilisée pour la programmation du système de télésurveillance est ouverte et développée par la société MediaMongrels [23]. La librairie WebSocket utilisée est construite sur la base de fonctions de communication réseau basiques de LabVIEW. Alors, elle peut être utilisée autant dans l'environnement de bureau Windows que dans l'environnement temps réel VxWorks de LabVIEW. L'installation de l'API se fait de manière standard via le gestionnaire de paquets LabVIEW VIPM.

4.4 SERVEUR DE FICHIER FTP

Le serveur FTP est une composante native des matériels CompactRIO et est un élément de réponse à l'objectif (3) du système d'acquisition, rappelé à la section 4.2, soit la transmission des données enregistrées au serveur informatique. Pour cela, le serveur met en œuvre le nécessaire pour gérer les connexions distantes et le transfert des fichiers par le protocole FTP. Ce protocole fonctionne selon le paradigme client/serveur. Donc, le transfert des fichiers s'opère à l'initiative d'une tâche de la liste qui sera présentée au chapitre 5, à la section 5.5. Dans le cadre du système de télé-surveillance développé, le fonctionnement du serveur est totalement transparent pour l'utilisateur puisqu'il répond de manière automatique aux requêtes de communication.

4.5 FICHIER DE CONFIGURATION

Le fichier de configuration qui a été largement évoqué dans les sections précédentes est présent directement dans le système de fichiers du CompactRIO et il regroupe l'ensemble des éléments paramétrables du système d'acquisition. Ce fichier, essentiel au bon fonctionnement du système de télésurveillance, touche directement ou indirectement tous les intervenants du VI LabVIEW, depuis la conversion des signaux en unités d'ingénierie jusqu'à la définition des horaires d'enregistrements programmés.

Le paramétrage du système est assuré par le super-utilisateur ou l'utilisateur selon l'importance de la nature de la variable. Par exemple, l'utilisateur peut changer des paramètres liés aux fonctions pratiques du système, comme les dates et horaires des enregistrements automatiques, alors que le super-utilisateur peut aussi régler les seuils de déclenchement des enregistrements sur événements ou bien les informations intrinsèques liées aux capteurs (sensibilité, unité, etc.) ou aux acquisitions (fréquences réelle et effective d'échantillonnage, etc.). Le paramétrage du système est accessible à travers des boîtes de texte de l'onglet *Configurations* du site web (fig. 5.10), qui doit ensuite générer automatiquement le nouveau fichier de configuration puis l'envoyer sur le CompactRIO

Variables organisé	es selon fichier .XM	ΛL	Unité	Description
Utilisateur	surHoraire	CRON_1 CRON_2 CRON_3 memFlot	syntaxe CRON syntaxe CRON syntaxe CRON durée	Horaire 1 pour les enregistrements planifiés Horaire 2 pour les enregistrements planifiés Horaire 3 pour les enregistrements planifiés Durée des enregistrements planifiés
	surUtilisateur	memFlot	durée	Durée des enregistrements sur demande
	surEvenement	preEven	durée	Durée d'enregistrement avant événement
		postEven	durée	Durée d'enregistrement après événement
		courriels	-	Liste des adresses email à qui envoyer des alertes
SuperUtilisateur	e chantillon nage	reel	Hz	Fréquence d'échantillonnage FPGA
		effect if	Hz	Fréquence d'échantillonnage des fichiers
	canal1	serieCapteur	-	Numéro de série du capteur du canal 1
		sensibilite	unité du capteur	Sensibilité du capteur du canal 1
		unite	-	Unité d'ingénierie du capteur du canal 1
		seuilMin	unité du capteur	Valeur min du seuil de déclenchement du cap- teur du canal 1
		seuilMax	unité du capteur	Valeur max du seuil de déclenchement du cap- teur du canal 1
	canal 2	serieCapteur	-	Numéro de série du capteur du canal 2
		sensibilite	unité du capteur	Sensibilité du capteur du canal 2
		•••		

Tableau 4.4 – Paramètres du fichier de configuration

lors de l'appui sur le bouton *Mise à jour du système* de la page. Si l'individu connecté dispose d'un compte super-utilisateur, celui-ci a un accès complet à la page *Configurations* et au fichier.

Tel que mentionné à la section 4.3.1, les paramètres du fichier de configuration sont chargés dans les variables appropriées à l'étape *E10*, soit lors des phases d'initialisation du système. Il s'agit d'un fichier au format .XML, ce qui simplifie l'organisation de l'information et son extraction dans l'environnement LabVIEW. Les balises déclarées doivent donc respecter le standard LabVIEW. L'annexe D montre un exemple de fichier de configuration, qui est cohérent avec les paramètres montrés à la figure 5.10 de la section 5.4.6. L'ensemble des champs disponibles et configurables du fichier, regroupés selon leur organisation au sein du fichier .XML, est présenté au tableau 4.4. On y distingue, grâce à la première colonne, les paramètres reliés aux niveaux d'autorisation de l'utilisateur et du super-utilisateur.

Chapitre

5

Serveur informatique

Dans l'architecture proposée du système de télésurveillance, le serveur informatique agit tel un centre de données. Le thème des données est l'essence même du serveur : il les collecte, les classe, les stocke, les traite, les analyse et les rend disponibles aux intervenants. Il permet aussi d'émettre des alertes et d'observer les données en temps réel mesurées sur le pont en fournissant les commandes Javascript nécessaires au navigateur web de l'utilisateur. Il remplit ses objectifs grâce à une collection de logiciels reposant sur un simple matériel de type *personal computer* (PC).

Ce chapitre présente le serveur informatique utilisé pour la création du prototype de système de télésurveillance des vibrations ambiantes d'un pont routier en service. Plus spécifiquement, on y présente d'abord le matériel utilisé ainsi que les logiciels qui y sont déployés, puis on présente de manière détaillée les trois fonctionnalités principales hébergées sur le serveur : la base de données, le portail web et l'exécution de tâches programmées incluant l'analyse avancée des données (analyses modales automatiques).

5.1 MATÉRIELS

La figure 5.1 montre le serveur informatique utilisé pour le prototype, soit un ordinateur portable Dell Precision M4500. Le même ordinateur a été utilisé pour développer l'ensemble du prototype de système de télésurveillance, et a donc joué le rôle de serveur informatique et autant que d'ordinateur hôte pour la programmation LabVIEW du CompactRIO (chap 4). L'ordinateur est équipé d'un processeur Intel Core i7 Q740 cadencé à 1.73 GHz avec une mémoire vive installée d'une taille de 8 Go. Le système d'exploitation exécuté est Windows 7 Entreprise 64 bits avec le Service Pack 1. Cet ordinateur ne dispose pas de caractéristiques de performances particulières à l'exception d'être en mesure d'exécuter les logiciels mentionnés à la section 5.2 avec une rapidité confortable pour les usagers. Cette absence de contrainte sévère sur la performance permet d'utiliser un matériel à prix réduit pour le déploiement effectif d'un tel système de télésurveillance (le coût de la machine présentée à la figure 5.1 étant inférieur à 1500 \$ à l'état neuf). Comme on le verra à la section suivante, la plupart des logiciels qui y sont exploités sont totalement gratuits d'usage et en architecture ouverte. Donc, le coût final du serveur une fois installé consistera essentiellement en un coût d'ingénieur nécessaire au développement des nouvelles routines, le cas échéant, et à la maintenance des logiciels. Pour le prototype développé de système de télésurveillance, un seul serveur est nécessaire. En effet, le trafic de données et le volume de demandes est très faible, ce qui ne justifie pas l'usage d'une grappe de serveurs contrairement à d'autres architectures plus complexes présentées dans la revue de littérature (chap. 2).



Figure 5.1 – Ordinateur portable utilisé comme serveur informatique pour le prototype

Pour le développement du serveur informatique, l'ordinateur a été pris en charge par le service informatique de l'Université de Sherbrooke, et réside donc en permanence sous le domaine usherbrooke.ca. Pour des raisons de sécurité du réseau informatique universitaire, il est nécessaire d'être connecté au réseau privé virtuel de l'Université de Sherbrooke pour que le serveur soit disponible depuis l'extérieur du site universitaire à l'adresse IP 10.43.102.114. Cette adresse fixe a été attribuée spécifiquement à cet ordinateur pour le développement du prototype et le serveur reste en ligne en permanence. Il peut répondre aux demandes des utilisateurs et des super-utilisateurs en tout temps. De la même manière, il peut émettre des demandes vers le système d'acquisition à tout moment.

5.2 LOGICIELS

Les logiciels déployés sur le serveur informatique sont listés au tableau 5.1. De plus, ce tableau présente les langages de programmation directement ou indirectement utilisés pour l'usage de ces logiciels dans le prototype de système de télésurveillance. Les fonctions et versions des logiciels et des langages sont également présentées.

La suite XAMPP (www.apachefriends.org) permet d'installer d'un même coup les logiciels MariaDB [20], Apache [1] et l'interpréteur PHP. On complète en installant l'interpréteur Python 2.7 [29], le logiciel Artemis Modal [33] et la librairie Matlab Runtime. Le planificateur de tâches Microsoft et les langages HTML, JavaScript et SQL n'ont pas besoin d'être installés, puisqu'ils sont implicites à d'autres logiciels. Ainsi, le planificateur est présent grâce au système d'exploitation Windows 7 Entreprise, les langages HTML et JavaScript sont interprétés directement par les navigateurs web, et le langage SQL est interprété par le serveur de la base de données. À l'exception d'Artemis Modal et du planificateur de tâches Microsoft, tous les logiciels sont en accès libre et gratuit d'utilisation.

La présence de XAMPP et de Python est justifiée en raison de leurs caractéristiques bénéfiques au développement et à la maintenance du système autant dans le court terme que le moyen et long terme. En effet, ce regroupement logiciel est *open source*, totalement gratuit d'usage, performant, animé par une communauté d'utilisateurs vaste et active, est commercialement indépendante, disponible pour tous les systèmes d'exploitations majeurs (Windows, Linux et OS X), et est répandu en hébergement commercial. La suite XAMPP rassemble également un écosystème informatique dont la cohabitation est assurée et intègre des outils satellites facilitant le développement (par ex.,

Logiciel	Version logiciel	Fonction pour prototype	Langage impliqué	Version langage	Fonction du langage
Apache	2.4.17	Servir la page web	PHP	5.6.17	Dynamiser l'information sur la page web
			HTML	5	Disposer l'information sur la page web
			JavaScript	6	Dynamiser la page web côté client
ArteMIS Modal	5.0	Calculer les pro- priétés modales	-	-	-
MariaDB	10.1.19	Servir la base de données	SQL	-	Interagir avec la base de données
Matlab Run- time	9.0	Piloter ArteMIS Modal	Matlab	R2016	-
Planificateur Windows	1.0	Déclencher les tâches Python	-	-	-
Python	2.7.11	Exécuter les tâches planifiées	Python	-	-

Tableau 5.1 – Logiciels et langages de programmation utilisés sur le serveur informatique

phpMyAdmin pour la gestion de la base de données). Le langage Python, lui, dispose de librairies variées et adaptées au domaine scientifique. La librairie Matlab Runtime permet, quant à elle, d'exécuter des programmes codés avec des fonctions basiques de Matlab, puis compilés sous forme de fichier exécutable.

Le choix du logiciel Artemis Modal pour l'extraction automatique des propriétés modales du pont est fait, car celui-ci a été adopté par le MTQ et l'université depuis plusieurs années. Les intervenants disposent donc des licences requises en plus d'avoir accès au support technique pour des opérations particulières. Notamment ici, l'exécution externe du logiciel (sans interface utilisateur) à partir de ses modules de calculs intrinsèques a requis le support du fournisseur qui commercialise le programme. Cette exécution externe des fonctions d'extraction modale est pilotée par un script Matlab [22].

Le gestionnaire des tâches est développé par Microsoft et est livré comme un accessoire au système d'exploitation Windows 7 Entreprise. À ce titre, il n'est pas gratuit, mais tout ordinateur équipé de Windows en dispose. Ici, il est opportun de préciser que l'environnement Windows a été adopté en raison de l'utilisation du logiciel Artemis Modal, qui ne peut être installé sur une distribution Linux.

Les sections suivantes fournissent les détails du serveur du prototype de système de télésurveillance : base de données, portail web et liste des tâches (incluant les analyses modales), qui sont pris en charge respectivement par le système de gestion de base de données relationnelles MariaDB, le serveur HTTP Apache, et des commandes Python (combinées à la librairie Matlab Runtime et au logiciel Artemis Modal).

5.3 BASE DE DONNÉES

L'usage d'une base de données permet l'augmentation des performances dans toutes les interactions avec celles-ci. La revue de la littérature à la section 2.1.3 cite chacun de ces avantages. Quelques-uns parmi ceux-ci sont particulièrement vrais pour le projet décrit dans ces pages :

- 1. Le contrôle des données redondantes :
 - Les données brutes de la base sont des métadonnées, appelées enregistrements (en anglais, *records*), qualifiant les signaux mesurés par le système d'acquisition ¹. Pour le prototype développé, les métadonnées sont archivées dans huit tables de la base, donc la nature des enregistrements varie selon la table dont il est question. Au niveau du serveur informatique, les métadonnées de la base de données permettent à l'utilisateur de retrouver le contexte lié à chaque série temporelle de signaux enregistrés. En général, pour beaucoup d'acquisitions, ce contexte reste le même. Ne pas utiliser ce système d'organisation des données impliquerait la définition d'un fichier conjoint au fichier contenant les signaux bruts pour l'enregistrement des attributs de ceux-ci. Cette option multiplie inutilement le nombre de fichiers et génère de la redondance. De plus, cette redondance est difficilement contrôlable puisqu'elle est associée à une multitude de fichiers. Ainsi, l'usage de tables dédiées dans une base de données pour enregistrer les attributs des signaux permet de contrôler la redondance.
- 2. La possibilité de générer plus d'information avec la même quantité de données :
 - Pour tracer l'évolution des propriétés modales à travers le temps, celles-ci sont stockées dans une table suite aux calculs effectués par Artemis Modal. La comparaison des résultats dans le temps permet d'émettre des hypothèses et de chercher à justifier la nature des changements, le cas échéant. Ce processus d'analyse d'ordre supérieur fera certainement appel à d'autres tables contenant d'autres attributs, qui finalement permettront de qualifier le degré de corrélation entre les phénomènes. Toute cette démarche théorique repose sur un stock initial unique de données. L'information supplémentaire est obtenue suite à des manipulations simples sur les attributs générés pour lesquelles la mise en œuvre d'une base agit comme un catalyseur.
- 3. Le partage des données :

Il est simple et rapide de copier-coller une base de données. Ensuite, on peut l'intégrer à d'autres bases soit pour le stockage, soit pour l'utiliser comme source additionnelle afin d'établir d'autres relations entre les variables.

4. L'amélioration de la rapidité, de la productivité et de l'accès aux données :

La base de données présentée à travers l'interface web du prototype développé permet d'accéder à l'information souhaitée sans avoir à faire des manipulations logicielles pour les importer, les filtrer, les croiser, les tracer, etc. Il suffit à l'utilisateur d'ouvrir son navigateur web et de se connecter sur le site de l'application web, qui est capable d'accéder rapidement à l'information désirée grâce à l'organisation systémique de la base de données.

5. L'économie d'échelle :

Les données du système de télésurveillance développé reposant sur une base de données, il devient possible d'interfacer un pont supplémentaire en ajoutant, avec peu d'effort, quelques composantes supplémentaires dans la base sans avoir à tout reprendre depuis le début.

Plusieurs interfaces permettent l'accès à la base de données : (1) en mode interactif avec un client logiciel tel que phpMyAdmin; (2) en mode interactif avec une invite de commandes; (3) en mode codage avec une interface de programmation applicative (API) disponible pour les langages populaires tels que C++, Matlab, Python et PHP pour ne citer que ceux-là. La figure 3.1 (chap. 3) illustre les différents modes d'accès à la base de données utilisés dans le prototype. Le mode interactif est privilégié lors de la mise en place d'une base de données et la définition des tables associées. Par la suite, l'usage de la base de données dans le serveur informatique repose sur l'API avec les

^{1.} Il ne faut pas confondre les métadonnées (ou enregistrements) et les données elles-mêmes qui, elles, sont les séries temporelles des signaux issus des accéléromètres installés sur le pont. Les signaux, eux, ne sont pas archivés dans la base, ils sont stockés sur le système de fichier du serveur informatique.

langages Python et PHP. Les deux prochains paragraphes présentent les flux de données mis en jeu dans le prototype pour ces deux langages.

Les flux de données en langage Python sont principalement issus des nouvelles mesures de vibrations ambiantes sous la forme de nouveaux enregistrements de métadonnées et des résultats de calculs des propriétés modales en sortie du logiciel Artemis Modal. Le listing 5.1 montre un extrait du code source, écrit en langage Python, pour l'ajout de nouvelles métadonnées à une table existante. Tout d'abord, l'appel à l'API est fait à la ligne 1. Ensuite, un objet *connection* est créé à l'aide de la fonction .connect. Ce dernier permet la création d'un autre objet, nommé *curseur*, afin d'exécuter une requête prenant la forme d'une chaîne de caractère. Les commandes se poursuivent jusqu'à ce que les métadonnées soient finalement ajoutées à la table *enregistrements*.

```
1 import MySQLdb as mdb
2 ...
3 try:
4 connection = mdb.connect(host=*, user=*, passwd=*, db="pont_riviere-aux-mulets")
5 curseur = connection.cursor()
6 curseur.execute("SELECT MAX(date) FROM enregistrements")
7 ...
```

Listing 5.1 – Extrait du code source Python pour l'ajout de nouvelles données à la base

Les flux de données en langage PHP concernent essentiellement l'accès au site web avec la vérification des paramètres de connexion (nom d'utilisateur et mot de passe) et les requêtes-réponses des pages *Liste d'enregistrements, Analyses modales - historique, Analyse modales - détails* et *Configurations* du portail web (section 5.4). Le listing 5.2 montre un extrait du code source, écrit en langage PHP, pour la connexion au site web. L'API utilisé avec le langage PHP apparaît aux lignes 3, 4, 6 et 7 avec les mots-clés ayant pour racine mysqli_. De façon similaire au listing 5.1, l'usage de l'API consiste à créer des objets sur lesquels on applique des fonctions. Ainsi, l'objet **\$con** (le connecteur) permet l'exécution de la requête de la ligne 6 avec la fonction mysqli_query(). Les résultats sont par la suite analysés afin de déterminer s'il s'agit d'un utilisateur autorisé à accéder au site web.

```
<?php
1
2
    . . .
    $con = mysqli_connect(*,*,*);
3
   mysqli_select_db($con, 'pont_riviere-aux-mulets');
4
    $sql="SELECT * FROM utilisateurs WHERE nom util = '".$nom."':
5
    $result = mysqli_query($con,$sql);
6
\overline{7}
    $row = mysqli_fetch_array($result);
8
    . . .
9
    2>
```



On remarque dans les listings 5.1 et 5.2 la présence du langage de requête structurée SQL, respectivement aux lignes 6 et 4. Les listings présentés opèrent donc sur la base de formulations préprogrammées en SQL. Ces formulations sont figées et restitueront toujours le même type de données à chaque exécution des scripts. Cela est différent du travail effectué par un super-utilisateur qui interagirait directement en SQL avec la base de données pour y faire toutes sortes de tâches

tombant dans les catégories de création, de configuration ou de maintenance. À cette fin, le superutilisateur utilise un logiciel avec interface graphique comme phpMyAdmin, ou bien il manipule directement le SQL en ligne de commande avec une invite tel que MS-DOS. La figure 5.2 montre l'exemple d'une session MariaDB où le super-utilisateur interroge la table *utilisateurs*.

🖬 Administrateur : XAMPP for Windows - mysql -u root -p	
MariaDB [pont_riviere-aux-mulets]> desc utilisateurs;	^
Field Type Null Key Default Extra	
id int(11) NO 0 nom_uti1 varchar(20) YES NULL mot_passe varchar(20) YES NULL Evolution text YES NULL	III
4 rows in set (0.04 sec) MariaDB [pont_riviere-aux-mulets]> select × from utilisateurs where id=1; ++++	
id nom_util mot_passe Evolution +	
1 ing1 abc123 210.218.585.591.594.597.599.602.606	
1 row in set (0.00 sec)	
MariaDB [pont_riviere-aux-mulets]>	Ŧ

Figure 5.2 – Exemple d'une session MariaDB : description de la table *utilisateurs*, puis sélection de tous les attributs de l'utilisateur correspondant à "id = 1"

Hormis les données brutes des signaux de vibrations ambiantes qui sont stockées sur le système de fichiers du serveur informatique, toutes les données sont stockées dans une base de données MariaDB nommée *pont_riviere-aux-mulets*. Cette base contient huit tables : *analyses_modales*, *canaux, capteurs, configurations, enregistrements, journal, noeuds* et *utilisateurs*. Le tableau 5.2 liste l'ensemble de leurs attributs, de même que le type de données associé à ceux-ci. La dernière colonne permet de noter d'autres renseignements, à savoir si l'attribut est une clef primaire et s'il doit être incrémenté automatiquement, s'il y a une valeur par défaut associée à l'attribut ou, finalement, s'il est possible d'omettre une valeur dans la consignation de l'enregistrement. Pour information, au sens d'une table (peu importe laquelle), la clef primaire est un identifiant unique permettant de retracer un enregistrement de la table. Cette clef peut être associée à un seul attribut ou formée d'une combinaison d'attributs. Les prochains paragraphes décrivent les tables constituant la base de données, présentées dans un ordre facilitant la compréhension des liens entre elles.

Table capteurs

La table consigne les informations relatives aux instruments déployés (actuellement ou par le passé) sur la structure surveillée. La table a pour clef primaire l'attribut *capteur_id*. Cet attribut s'incrémente automatiquement à toute nouvelle entrée dans la table. Pour chaque capteur identifié, les informations sont l'identifiant du capteur (entier de 10 caractères maximum), le type du capteur (mot énuméré dans une liste ne contenant pour l'instant que le terme 'accelerometre'), son numéro de série (mot à caractères variables de longueur 100 au maximum), sa sensibilité (nombre à virgule

Tables	Attributs	Types	Note
analyses_modales	enregistrement_id	int(10)	clef primaire
	mode	int(10)	clef primaire
	frequence	float	
	amortissement	float	
	complexite	float	
	list_noeuds	text	
	$amplitude_x$	text	
	amplitude_y	text	
	amplitude_z	text	
	phase_x	text	
	phase_y	text	
	phase_z	text	
canaux	canal_id	int(10)	clef primaire, auto-incrément
	capteur_id	int(10)	
	noeud_id	int(10)	
	orientation_x	float	
	orientation_y	float	
	orientation_z	float	
	seuil_min	float	
	seuil_max	float	
capteurs	capteur_id	int(10)	clef primaire, auto-incrément
	type	enum('accelerometre')	$\mathrm{accelerometre}^\dagger$
	no_serie	varchar(100)	
	sensibilite	float	
	unite	$enum('(mV)/(m.s^{-2})')$	$(mV)/(m.s^{-2})^{\dagger}$
configurations	config_id	int(10)	clef primaire, auto-incrément
	utilisateur_id	int(10)	
	mise-en-service	datetime	$CURRENT_TIMESTAMP^{\dagger}$
	$echantillonnage_reel$	decimal(10,0)	2000^{\dagger}
	$echantillonnage_effectif$	decimal(10,0)	200^{+}
	canal_1	int(10)	
	canal_2	int(10)	
	canal_3	int(10)	
	canal_4	int(10)	
enregistrements	enregistrement_id	int(10)	clef primaire, auto-incrément
	date	date	
	heure	time	
	type	enum('surEvenement',	$surHoraire^{\dagger}$
		'surHoraire', 'surUtilisateur')	
journal	utilisateur_id	int(10)	clef primaire
	date_heure	datetime	clef primaire, CURRENT_TIMESTAMP [†]
	information	text	
noeuds	noeud_id	int(10)	clef primaire, auto-incrément
	position_x	float	
	position_y	float	
	position_z	float	
utilisateurs	utilisateur_id	int(10)	clef primaire, auto-incrément
	nom	varchar(20)	
	mot_passe	varchar(20)	
	super_utilisateur	$tiny_int(1)$	0^{\dagger}
	evolution	text	NULL [†] [‡]

Tableau 5.2 – Tables et attributs de la base de données *pont_riviere-aux-mulets*

[†] Valeur par défaut. [‡] Absence de valeur acceptée.

flottante), ainsi que l'unité dans laquelle doit être exprimée la mesure (mot énuméré dans une liste ne contenant pour l'instant que le terme '(mV)/(m.s⁻²)'). Les attributs *sensibilite* et *unite* contiennent les valeurs utilisées par le VI LabVIEW lors de l'acquisition et de l'enregistrement des données (chap 4). Les attributs *type* et *unite* ont les valeurs par défaut 'accelerometre' et '(mV)/(m.s⁻²)', si bien qu'à l'occasion d'une nouvelle entrée, ces valeurs n'ont pas besoin d'être spécifiées.

Table noeuds

La table établit une liste des points de mesure physiques où sont disposés des capteurs sur la structure instrumentée. La table a pour clef primaire l'attribut *noeuds_id*, qui s'incrémente automatiquement à toute nouvelle entrée dans la table. Les trois autres attributs sont simplement les coordonnées du point de mesure dans chacune des trois directions principales, x, y et z, encodées comme des nombres à virgule flottante.

Table canaux

La table consigne les informations relatives à la définition d'un canal d'acquisition sur le matériel du CompactRIO. La table a pour clef primaire l'attribut *canal_id*, qui s'auto-incrémente à toute nouvelle entrée dans la table. Pour le prototype de système télésurveillance, un canal est défini par l'identifiant du capteur branché, l'identifiant du nœud du capteur, son orientation (via un vecteur directeur défini par ses trois cordonnées x, y et z) et les seuils minimal et maximal engendrant un événement au niveau de l'acquisiteur pour le capteur considéré (chap 4). À l'exception des attributs d'identifiants (encodés sur des entiers à 10 caractères), les autres attributs de la table *canaux* sont des nombres à virgules flottantes.

Table configurations

La table *configurations* permet de colliger l'information pertinente définissant entièrement les caractéristiques du système de mesure installé sur le pont instrumenté : définition complète de chacun des canaux dans leur ordre de branchement à l'acquisiteur (qui correspond à l'ordre des colonnes du fichier d'enregistrement des signaux bruts) et les fréquences d'échantillonnage réelle et effective. Un identifiant unique est associé à une configuration sous l'attribut *config_id*, qui forme la clef primaire de la table et est auto-incrémenté à toute nouvelle configuration. Les informations sur les canaux sont obtenues de la table *canaux*, et par extension des tables *noeuds* et *capteurs*, à travers les attributs *canal_** dont les valeurs correspondent aux identifiants de la table *canaux*. Les valeurs des fréquences d'échantillonnage réelle et effective sont celles utilisées par le VI LabVIEW de l'acquisiteur (chap. 4) et ont respectivement des valeurs par défaut de 2000 Hz et 200 Hz. La table consigne aussi l'identifiant de l'utilisateur auteur de la configuration, ainsi que la date et l'heure de la mise-en-service. La mise-en-service a pour valeur par défaut le moment où la nouvelle configuration a été consignée (*CURRENT_TIMESTAMP*).

Table *enregistrements*

La table *enregistrements* archive, sous des attributs portant des noms éponymes, la date, l'heure et le type de déclencheur à l'origine de chacun des enregistrements des signaux des capteurs effectués in situ par le système d'acquisition. L'attribut *enregistrement_id* forme la clef primaire et est autoincrémenté à toute nouvelle entrée de la table. En considérant les liens entre les tables de la base de données, toutes les métadonnées liées à un enregistrement sont retraçables. En effet, la date et l'heure de l'enregistrement combinées à la date de mise-en-service de la configuration permettent de faire le lien avec les conditions générales de la collecte des données brutes. Notons que l'attribut *type* prend une des valeurs d'une énumération ayant pour choix les éléments de la liste 'surEvenement', 'surHoraire' et 'surUtilisateur' (la valeur par défaut est 'surHoraire'). Les données elles-mêmes des enregistrements (signaux des capteurs) ne sont donc pas archivées dans la base. Elles sont stockées dans un fichier du système de fichiers du serveur dont le nom est tel qu'on peut retrouver son chemin grâce aux seules métadonnées de la table *enregistrements* (section 5.5.1).

Table analyses_modales

La table consigne les résultats issus d'un calcul des propriétés modales d'un enregistrement donné. La clef primaire de la table est la combinaison des attributs enregistrement_id et mode, ce dernier correspondant au numéro usuel du mode identifié sur la structure. La table permet de consigner la fréquence propre, l'amortissement et la complexité d'un mode (section 5.5.2) à l'aide de nombres à virgule flottante. Le vecteur propre du mode est consigné à l'aide de l'attribut list_noeuds, qui établit la liste des identifiants des nœuds où est définie la déformée modale (format texte d'entiers séparés par une virgule), et de six autres attributs également au format texte. Il s'agit de l'amplitude de déplacement et de la phase de celui-ci dans chacune des trois directions principales x, y et z pour chacun des nœuds de la liste (en respectant l'ordre de la liste). Par exemple, si le système est construit à l'aide de 4 canaux d'accéléromètres installés en duo selon les directions x et y de deux nœuds sur la structure, les attributs amplitude_* et phase_* seront des vecteurs unidimensionnels de 2 valeurs séparées des virgules. L'attribut enregistrement_id, commun aux tables analyses_modales et enregistrements, est suffisant pour remonter à la configuration exacte du système à laquelle est liée le calcul des propriétés modales présentées.

Table journal

La table *journal* offre au système un lieu de stockage général pour les annotations pertinentes effectuées automatiquement par le CompactRIO (journal du cRIO) et pour des commentaires qu'un utilisateur désire consigner dans le système à partir de l'interface web (section 5.4). La table contient l'identifiant unique de l'utilisateur à l'origine des commentaires, la date et l'heure au moment de la consignation, et le texte d'information lui-même. Les attributs *utilisateur_id* et *date_heure* forment la clef primaire de la table permettant d'identifier de manière unique un commentaire du journal.

Table utilisateurs

La table *utilisateurs* permet d'enregistrer les informations relatives à la description des utilisateurs du système de télésurveillance. Cette table contient d'abord un identifiant unique avec l'attribut *utilisateur_id* (clef primaire de la table qui s'auto-incrémente pour chaque nouvelle entrée dans la table), le nom de l'utilisateur et son mot de passe pour l'accès au site web, avec les attributs *nom* et *mot_passe* (chaînes de 20 caractères maximum). Un attribut *super_utilisateur* permet d'identifier le niveau d'autorisation permis à l'utilisateur. Il est typé mini-entier, prenant soit la valeur '0' soit la valeur '1' selon que l'utilisateur est ou non un super-utilisateur. Enfin, l'attribut *evolution* est formé d'une liste des *enregistrement_id* consultés par l'utilisateur en question lors de sa dernière session sous l'onglet *Analyses modales - historique* du site web (section 5.4). Avec ce dernier attribut au format texte, la base de données permet au prototype de réafficher l'évolution des fréquences propres des modes de la structure pour les enregistrements consultées à la dernière visite du site web. Cette caractéristique permet de qualifier sans ambiguïté le système comme étant une application web dynamique. L'attribut *evolution* peut ne pas contenir de valeur.

5.4 PORTAIL WEB ET SERVEUR HTTP

Le serveur HTTP constitue la porte d'entrée pour l'utilisation du prototype. En effet, le serveur expose à l'utilisateur le site web dans lequel les fonctionnalités accessibles sont classées dans sept onglets distincts. La liste suivante présente ces onglets et spécifie leur fonctionnalité principale :

- 1. Accueil : connexion et déconnexion au site (fig. 5.3);
- 2. Vibrations ambiantes : consultation des vibrations ambiantes en temps réel et enregistrement à la demande (fig. 5.5);
- 3. Liste d'enregistrements : accès aux enregistrements des signaux (fig. 5.6);
- 4. Analyses modales historique : évolution des fréquences propres dans le temps (fig. 5.7);
- 5. Analyses modales détails : détail des modes propres identifiés, incluant les déformées (fig. 5.8);
- 6. Configurations : détails de la configuration du système avec possibilité de mise à jour (fig. 5.10);
- 7. Journal de bord : consignation et consultation de faits pertinents.

Le prototype utilise le serveur web Apache [1], tandis qu'un utilisateur utilise son navigateur web habituel pour interagir avec le système. L'architecture client/serveur entre l'utilisateur et le site web utilise le modèle de communication requête/réponse. Ceci est vrai à l'exception de l'affichage des données en temps réel (fig. 5.5), où une communication bidirectionnelle est utilisée via un objet spécialisé, le *WebSocket*, déjà mentionné à la section 4.3.2. Dans ce dernier cas, le site web agit comme un conteneur, et les données proviennent directement du système d'acquisition sans passer par le serveur informatique (fig. 4.26).

Le site web est une partie majeure des travaux de programmation réalisés dans ce projet. Il est constitué de 34 fichiers, dont 27 ont été écrits dans le cadre des travaux de ce rapport. Ces derniers représentent plus de 3300 lignes de code. Les sept autres fichiers ont été écrits par des tiers et sont en accès libre. Le tableau 5.3 présente l'organisation de ces fichiers ainsi que leur fonction principale dans le site web en partant du fichier racine *index.html*.

Du côté de l'aspect visuel, de la logique de navigation par onglet, de même que de la gestion des boutons, tout est pris en charge par la bibliothèque jQuery User Interface (listing 5.3, lignes 5 et 8). Elle-même repose sur les fonctions de la bibliothèque jQuery (listing 5.3, ligne 7). Le widget *Tabs* de jQuery User Interface, appelé à la déclaration d'une division identifiée tabs (listing 5.3, ligne 19), gère les onglets depuis la page racine *index.html*. Dans cette page, les onglets sont définis à l'aide d'une liste (listing 5.3, ligne 20) et incorpore les hyperliens des fichiers associés à chacun de ceux-ci. Les fichier ayant l'extension .*php* dans le listing 5.3 sont activés par l'utilisateur au clic sur un onglet.

Les sous-sections suivantes décrivent un par un les onglets du site web en affichant une copie d'écran de ceux-ci, en expliquant leurs mécaniques de fonctionnement lorsque nécessaire et en présentant des éléments de code informatique lorsque cela est pertinent.

5.4.1 Accueil

Le site web est accessible à l'adresse IP 10.43.102.114/pont_riviere-aux-mulets. Cette adresse fonctionne depuis l'intérieur du campus de l'université de Sherbrooke, où est hébergé le serveur, mais peut être accédée depuis l'extérieur à l'aide du réseau privé virtuel de l'université, puisque les ports sur lesquels le système repose ont été débloqués par le service informatique de l'institution. Cette adresse pourra évidemment être modifiée lorsque le système sera déployé sur un pont réel, et il y aura également la possibilité d'acheter un nom de domaine plus évocateur qu'une simple adresse IP, par exemple www.tele_pont-riviere-aux-mulets.org.

$\mathrm{Fichiers}^{\dagger}$	Fichiers appelés †	Fonctions
Racine index.html (237)	Écrits par des tiers jquery-ui.css jquery-1.12.1.js jquery-ui.js jquery.flot.js jquery.flot.time.js three.js	Fournir le style du site web Simplifier la programmation Fournir l'interface par onglet du site web Fournir les conteneurs des graphiques Gérer l'axe temporelle des graphiques Fournir le conteneur 3D d'affichage du pont Permettre la manipulation du pont
	TrackballControls.js	dans le conteneur 3D
	des travaux	
	accueil.php (108)	derer l'acces au site web, gerer la presentation et le fonctionnement de l'onglet Accueil
	vibrations_ambiantes.php (9) listes_enregistrements.php (9)	Gérer l'accès à l'onglet Vibrations ambiantes Gérer l'accès à l'onglet Listes d'enregistrements Cérer l'accès à l'onglet
	analyse_historique.php (9)	Analyse modales - historique
	analyse_detail.php (9)	Gérer l'accès à l'onglet Analyse modales - détails
Opplata	configurations.php (20 - à voir) journal.php (9)	Gérer l'accès à l'onglet Configurations Gérer l'accès à l'onglet Journal de bord
accueil.php	login.php (26)	Tester l'authenticité d'un utilisateur
	RetirerEvolution.php (21)	Extraire la liste des enregistrements consultés à la dernière visite
	logout.php (5)	Déconnecter l'utilisateur du site web
$vibrations_ambiantes.php$	vibrations_ambiantes.html (154)	Gèrer la présentation de l'onglet Vibrations ambiantes
vibrations_ambiantes.html	vibrations_ambiantes.js (420)	Gérer le fonctionnement de l'onglet Vibrations ambiantes
$listes_enregistrements.php$	listes_enregistrements.html (62)	Gèrer la présentation de l'onglet Listes d'enregistrements
$listes_enregistrements.html$	listes_enregistrements.js (312)	Gérer le fonctionnement de l'onglet Listes d'enregistrements
listes_enregistrements.js	supprimerEnregistrement.php (48)	Permettre la suppression d'un fichier d'enregistrement
	requete_donnees.php (141)	Exécuter une requête des fichier correspondant aux critères de recherche le l'utilisateur
$analyse_historique.php$	analyse_historique_v1.html (128)	Gerer la presentation de l'onglet Analyses modales - historique
analyse_historique_v1.html	analyse_historique_v1.js (441)	Gérer le fonctionnement de l'onglet Analyses modales - historique
analyse_historique_v1.js	EvolutionAuto_v1.php (115)	Exécuter une requête d'évolution modale selon des critères automatiques
	Affichage_historique_v1.php (72)	Exécuter une requête d'évolution modale selon les enregistrements choisis par l'utilisateur
analyse_detail.php	analyse_detail.html (52)	Gérer la présentation de l'onglet Analyses modales - détail
$analyse_detail.html$	analyse_detail_v4.js (678)	Gérer le fonctionnement de l'onglet Analyses modales - détail
analyse_detail_v4.js	Affichage_detail_al.php (79)	Retirer les détails de l'analyse modale du fichier sélectionné
configurations.php journal.php	DetailNomEnregistrement.php (48) configurations_v1.html (181) journal.html (1)	Retirer le nom de l'enregistrement sélectionné Gérer la présentation de l'onglet Configurations Gérer la présentation de l'onglet Journal de bord

Tableau 5.3 – Organisation de l'ensemble des fichiers composant le site web

[†] Les nombres entre parenthèses indiquent le nombre de lignes constituant le fichier.

1	<html></html>	
2	<head></head>	
3	<title>STSUS</title>	
4		
5	<pre><link href="jquery-ui.css" rel="stylesheet" type="text/css"/></pre>	
6		
$\overline{7}$	<script src="jquery-1.12.1.js"></script>	
8	<script src="jquery-ui.js"></script>	
9	<script src="jquery.flot.js"></script>	
10	<pre><script src="jquery.flot.time.js"></script></pre>	
11	<script src="three.js"></script>	
12	<pre><script src="TrackballControls.js"></script><td></td></pre>	
13		
14	<pre><script type="text/javascript"></pre></td><td></td></tr><tr><td>15</td><td>// Variables et fonctions générales communes à plusieurs onglets</td><td></td></tr><tr><td>16</td><td></script></pre>	
17		
18	<body></body>	
19	<div id="tabs"></div>	
20		
21	<li id="accueil">Accueil	
22	<li id="vibrations_ambiantes">	
23	Vibrations ambiantes	
24	id="liste_enregistrement">	
25	Listes d'enregistrements	
26	<li id="analyses_historique">	
27	Analyses modales - historique	
28	<li id="analyse_details">	
29	Analyses modales - détails	
30	id="configurations">	
31	Configurations	
32	<li id="journal">Journal de bord	
33		
34		
35		
36		

Listing 5.3 – Extrait du code source de la page racine *index.html* du site web

L'accès au site est protégé par un nom d'utilisateur et un mot de passe, comme le montre la figure 5.3a. L'authentification est réussie si l'utilisateur insère ses informations de manière identique à celles contenues dans la table *utilisateurs* de la base de données. Une authentification réussie fait apparaître la fenêtre correspondant à la figure 5.3b en montrant le message 'Connexion réussie !', en affichant le nom de l'utilisateur et en lui fournissant un moyen de sortie via l'hyperlien *Déconnexion*. Une connexion réussie ouvre aussi l'accès aux autres onglets du site. Autrement, une tentative d'accès aux onglets autres que *Accueil* affiche un hyperlien renvoyant au formulaire d'authentification, comme le montre la figure 5.4. Le système d'authentification est évidemment sensible à la casse. Un échec à l'authentification fait apparaître le message 'Mauvais utilisateur et/ou mot de passe' en rouge.

Toutes ces fonctionnalités sont prises en charge par le code source du fichier *accueil.php* présenté par le listing 5.4. Celui-ci est divisé en quatre parties : (1) l'activation de l'environnement de session à la ligne 1, (2) un script pour la détection d'un clic sur le bouton *Connexion* et pour la vérification des

STSUS	×	and the second se			_	
← → C ③ Non	sécurisé 10.43.102.114/pont_rivie	ere-aux-mulets/				☆ 🗵 👂 🗄
	<i>a</i>	A				
	Systèn	ne de télésurveilla	nce des structures de	l'Université de She	rbrooke	
Accueil	Vibrations ambiantes	Listes d'enregistrements	Analyses modales - historique	Analyses modales - détails	Configurations	Journal de bord
		ldentifiez-vo	ous pour accéder aux or	nglets ci-dessus		
		Utilisateur:	Mot de passe:	Connexion		
		(a) avant authentificat	tion		
_						
STSUS	× 10.42102114/seat disi	en europeiete (_		
	securise 10.43.102.114/pont_rivie	ere-aux-mulets/				94 X M 12 :
	Svstèn	ne de télésurveilla	nce des structures de	l'Université de She	rbrooke	
-	•					
Accueil	Vibrations ambiantes	Listes d'enregistrements	Analyses modales - historique	Analyses modales - détails	Configurations	Journal de bord
	•					
			Connexion réussie!			
Utilizatore in	a 1					
LITUSATEUR' 10	S I					
Déconnexion	•					

Figure 5.3 – Captures d'écran de l'onglet Accueil

	🕁 💹 🔎 🗄
Système de télésurveillance des structures de l'Université de Sherbr	rooke
Accueil Vibrations ambiantes Listes d'enregistrements Analyses modales - historique Analyses modales - détails Co	onfigurations Journal de bord
<u>Identification</u> requise	

Figure 5.4 – Capture d'écran d'un onglet quelconque avant authentification

informations renseignées par l'utilisateur en comparaison à ce qui est archivé dans la base de données (lignes 3 à 14), (3) les fonctions d'affichage selon que l'utilisateur est authentifié ou non (lignes 16 à 24), (4) le paragraphe pour l'affichage du nom d'usager de l'utilisateur et l'hyperlien permettant la déconnexion (lignes 26 et 27). À titre d'illustration de certains mécanismes de programmation récurrents à travers le site web, les listings 5.5 et 5.6 montrent les scripts satellites *login.php* et *logout.php* appelés par le listing 5.4.

Le listing 5.5 illustre l'usage de l'API mysqli pour la communication avec la base de données. Le code source du listing démarre avec l'activation de l'environnement de session à la ligne 2. Ensuite, les variables correspondant aux champs renseignés par l'utilisateur à la page d'accueil sont récupérées, i.e. le nom d'usager et son mot de passe aux lignes 4 et 5. Les lignes 7 à 10 permettent la connexion à la base de données, alors que les lignes 12 à 16 permettent la formulation, l'exécution et l'extraction d'une requête permettant de retirer le nom et le mot de passe de l'usager. Le listing 5.5 se termine avec une structure conditionnelle entre les lignes 18 et 25. D'abord à la ligne 18, s'il y a correspondance du nom et du mot de passe fourni par l'utilisateur avec ceux extraits de la base de données, le script affecte ces valeurs aux variables de session utilisateur et mot_passe . L'émission

```
1
     <?php session_start();?>
2
3
    <script type="text/javascript">
4
        $(document).ready(function(){
5
            $("#Connexion").click(function(){
6
                 \\ Appel à la base de données avec les infos de l'utilisateur
7
                 \\ via un $.post au script login.php
            }
 8
9
        });
10
11
        function appuiEnter(event){
             \\ Activer la fonction "Connexion.click()"
12
13
        }
    </script>
14
15
    <?php
16
    if (! isset ( $_SESSION ['utilisateur'] )) {
17
        echo "\\ Affichage avant la connexion, ou
18
               \\ Affichage après une tentative de connexion échouée";
19
    }
20
    else {
21
22
        echo "\\ Affichage après une tentative de connexion réussie";
    }
23
    2>
^{24}
25
    26
27
    <a id="deConnexion" style="color:blue;" href="logout.php"></a>
```



de l'écho 1 permettra au listing 5.4 de considérer l'utilisateur comme étant authentifié. A contrario, l'absence de correspondance du nom et du mot de passe à la ligne 18 entraîne l'émission de l'écho θ , sans effet sur les variables de session. Le listing 5.4 considèrera la tentative d'authentification comme un échec.

Lorsqu'un utilisateur authentifié clique sur l'hyperlien *Déconnexion* montré à la figure 5.3b, cela exécute le listing 5.6. Le code de ce listing consiste à activer l'environnement de session. Il est ensuite possible de détruire les variables de session, et finalement recharger la page racine du site web.

Si la variable de session *utilisateur* est renseignée, la page *accueil.php* fait finalement appel au script *RetirerEvolution.php*, qui permet de récupérer dans une variable de session les *enregistre-ment_id* consultés par l'utilisateur à sa dernière visite sur le site web et consignés dans l'attribut *evolution* de la table *utilisateurs* de la base de données. Les lignes de code étant fortement similaires au listing 5.5, cela n'est pas pertinent de présenter ces dernières.

5.4.2 Vibrations ambiantes

Un clic sur l'onglet Vibration ambiantes (fig. 5.5) active le listing 5.7. Ce code source permet l'exécution des fonctionnalités de l'onglet si la variable de session *utilisateur* est renseignée en chargeant la page *vibrations_ambiantes.html* (ligne 5). Autrement, l'hyperlien *Identification* s'affichera et permettra à l'utilisateur de retourner à la page d'accueil, tel qu'expliqué à la section précédente. Cette fonction est gérée par les lignes 7 et 8 du listing 5.7. À l'exception du fichier *accueil.php*, la

```
1
     <?php
\mathbf{2}
     session_start();
3
4
     $nom = $_POST['nom'];
5
     $mot_passe = $_POST['mot_passe'];
6
     $con = mysqli_connect('localhost', 'root', '*******');
7
8
     if (!$con) {
             die('Could not connect: ' . mysqli_error($con));
9
    }
10
11
12
    mysqli_select_db($con, 'pont_riviere-aux-mulets');
     $sql="SELECT * FROM utilisateurs WHERE nom = '".$nom."';
13
    $result = mysqli_query($con,$sql);
14
15
     $row = mysqli_fetch_array($result);
    mysqli_close($con);
16
17
     if(($row['nom'] == $nom) and ($row['mot_passe'] == $mot_passe)){
18
             $_SESSION['utilisateur'] = $nom;
19
             $_SESSION['mot_passe'] = $mot_passe;
20
21
             echo 1;
    }
22
     else {
23
^{24}
             echo 0;
    }
25
     ?>
26
```

Listing 5.5 – Code source login.php (illustration de la communication avec la base de données)

```
1 <?php
2 session_start();
3 session_destroy();
4 header("Location: index.html");
5 ?>
```

Listing 5.6 – Code source logout.php

structure du code *vibrations_ambiantes.php* montrée par le listing 5.7 est identique pour tous les fichiers gérant l'accès aux onglets, c'est-à-dire *listes_enregistrements.php*, *analyses_historique.php*, *analyses_detail.php*, *configurations.php* et *journal.php*. La seule différence entre ceux-ci se trouve à la ligne 5, où le fichier inclus correspond à la page à afficher.

L'onglet Vibrations ambiantes remplit principalement deux fonctions. Il permet d'abord à l'utilisateur d'observer en temps réel, sur un maximum de quatre graphiques, l'évolution des signaux de vibrations ambiantes des capteurs de son choix (*streaming*). L'utilisateur peut sélectionner les capteurs intéressants avec le menu déroulant associé à chaque graphique. Il permet ensuite à l'utilisateur de déclencher à la demande un enregistrement instantané. Comme le montre la figure 5.5, l'interface prévoit déjà la possibilité de spécifier des durées de signaux requis avant et après l'envoi de la requête, bien que cette option n'est pour le moment pas fonctionnelle au niveau du VI Lab-VIEW de l'acquisiteur. Ces fonctions ont déjà été abordées largement au chapitre 4 du point de vue de l'acquisiteur et de la programmation LabVIEW nécessaire (section 4.3.2), et le lecteur est
STSUS	×		_	_	_						
	n sécurisé 10.43.102.114/pont_riv	viere-aux-mulets/				ማ 🕁 🗾 🔎					
	Systèn	ne de télésurveilla	nce des structures de	l'Université de She	rbrooke						
Accueil	Vibrations ambiantes	Listes d'enregistrements	Analyses modales - historique	Analyses modales - détails	Configurations	Journal de bord					
Signaux:	Signaux: Temps réel Temps réel Temps réel										
Graph 1: Al	1 •		Graph 2: Al6 🔹								
1.0			1.0 - Al1								
0.5			AI2 0.5 AI3								
			AI4 AI5								
0.0			0.0 AI6								
-0.5			AI8 AI9								
			All0 All1								
-1.0			-1.0								
Graph 3: Al.	3 •		Graph 4: Al4 •								
0.5			0.5								
0.0			0.0								
-0.5			-0.5								
-1.0			-1.0								
Connecter	Deconnecter										
Acquisti	on manuelle:										
Analyses mos	s avant, 120 s apres										
Démarrer	Arrêter										
			Drôt								
			Figure								

Figure 5.5 – Capture d'écran de l'onglet Vibrations ambiantes

invité à consulter la figure 4.26 détaillant le processus informatique qui implique simultanément le VI LabVIEW de l'acquisiteur et le serveur HTTP. Notamment, lorsque l'utilisateur clique sur le bouton *Connecter*, un objet *Websocket* est généré et tente d'établir la connexion directe entre le navigateur web de l'utilisateur et le CompactRIO. S'il y arrive, le bouton *Connecter* passe au jaune sur la page *Vibrations ambiantes* du site. Une requête peut alors être exécutée par le CompactRIO, soit un enregistrement sur demande en utilisant la section *Acquisition manuelle* de la page, ou la mise à disposition des données pour l'affichage graphique en sélectionnant des canaux à afficher dans les menus déroulants des graphes de la section *Signaux* de la page.

Dans le cas d'une demande d'enregistrement, tout le travail est effectué par le système d'acquisition, et les données qui en résultent pourront être récupérées sous l'onglet *Listes d'enregistrements*. La difficulté de programmation de la page *Vibrations ambiantes* réside dans la fonction de *streaming*. Étant donné son ampleur, le fonctionnement détaillé du script *vibrations_ambiantes.js* correspondant ne peut pas être présenté dans ce rapport. En effet, ce fichier de 420 lignes est constitué de 41 variables, fonctions et objets. Notons simplement qu'à l'initiative de l'utilisateur ayant cliqué sur le bouton *Connecter*, la fonction *init* est appelée. Celle-ci exécute un certain nombre d'actions permettant l'acheminement des valeurs des signaux mesurés par le CompactRIO vers les graphiques de

```
1
    <?php
\mathbf{2}
        session_start();
3
4
        if(isset($_SESSION['utilisateur'])){
5
             include 'vibrations_ambiantes.html';
6
        }
        else echo "<h3 style='text-align: center'><a style='color:blue;' href='index.html'>
7
        Identification</a> requise</h3>";
8
    2>
9
```

Listing 5.7 – Code source vibrations_ambiantes.php

l'onglet Vibrations ambiantes. Notons que le CompactRIO ne diffuse aucune donnée sur le réseau. Il les met seulement à disposition et c'est le navigateur web qui les cueille au fur et à mesure. Par conséquent, si l'utilisateur quitte l'onglet Vibrations ambiantes alors que le streaming est en cours, le flux de données est interrompu, ce qui préserve la bande passante (la connexion au CompactRIO est aussi interrompue, comme si l'utilisateur avait cliqué sur le bouton Déconnecter). De la même manière, si aucun canal n'est sélectionné par l'utilisateur dans les menus déroulants d'un des graphiques, aucune donnée n'est inutilement diffusée sur le réseau et la page web affiche une simple ligne de zéros sur le graphique. Au niveau de la page web, la difficulté majeure de la fonction de streaming est de parvenir à afficher les données sur les graphiques pour qu'elles respectent la base de temps à laquelle elles correspondent réellement. En effet, seules les valeurs des signaux demandés sont communiquées par paquets au navigateur web, sans la base de temps, et celui-ci doit ensuite afficher les paquets consécutifs de données à un rythme reproduisant leur durée réelle. Si l'affichage est trop rapide par rapport à la durée réelle des signaux, la file de données finit par se vider, ce qui finit par créer un délai d'attente apparent à l'affichage. Si l'affichage est trop lent, les données s'accumulent et un retard excessif de l'affichage se crée par rapport à la réalité. Pour limiter ce risque, mais surtout pour faire face de façon générale à un retard potentiel dans la communication avec l'acquisiteur, l'affichage est programmé pour se réinitialiser si plus de 5 secondes de données sont accumulées dans les files d'affichage de la page web.

Il est prévu qu'ultérieurement l'onglet Vibrations ambiantes offre une troisième fonction encore non implémentée. Dans le haut de l'onglet, on observe à la figure 5.5 les options Temps réel ou Enregistrement antérieur. Elles permettent de choisir d'afficher dans les graphiques de la page soit les données mesurées en temps réel par le CompactRIO, ou les données provenant d'un fichier d'enregistrement archivé sur le serveur (celui qui est sélectionné dans la page Listes d'enregistrements). Seule l'option Temps réel est fonctionnelle actuellement. L'autre option nécessitera en fait la programmation d'une toute nouvelle page (bien que l'affichage restera sous l'onglet Vibrations ambiantes), car les fonctions à exécuter sont intrinsèquement différentes entre les deux options.

5.4.3 Listes d'enregistrement

L'onglet Listes d'enregistrements (fig. 5.6) a pour fonction de présenter de manière pratique la liste complète des enregistrements de signaux archivés dans la base de données du système de télésurveillance, d'en présenter les principales caractéristiques et d'en permettre l'accès aux données brutes à l'utilisateur. Grâce à certaines autres fonctionnalités disponibles, l'onglet est aussi une porte d'entrée vers les onglets Analyses modales - historique et Analyses modales - détails. Au niveau de la programmation, l'accès à l'onglet se fait à travers le fichier listes_enregistrements.php,

STSUS	×	the second	Accession and			_		_	÷ _ 0 ×
	sécurisé 10.43.10	2.114/pont_riv	/iere-aux-mulets/						야 ☆ 💹 👂 🗄
		Systèr	me de télé	surveilla	ince des str	uctures de	l'Université de She	rbrooke	
Accueil	Vibrations a	mbiantes	Listes d'enre	egistrements	Analyses moda	ales - historique	Analyses modales - détails	Configurations	Journal de bord
Filtros									
Filles.									
Dates	Heures		Type Anal	lyses modales	Afficher				
Du 2010-11-	11 de 00:00		Horaire 🛛 🖻 D	isponible	Cochés				
au 2015-07-	17 à 15:59	✓ surl	Utilisateur 🗆 N	on-disponible	✓ Non-cochés				
		₹ curi	Evenement						
Appliquer		⊡ sui i	Lvenement						
Appriquer									
Liste des	données	disponi	bles: Téléch	arger les coche	és Tout cocher /	décocher Suppri	mer la sélection de la base de do	nnées	
Pécultat : 2 e	pregistrements								
Resultat . 2 e	megisciements								
4			_		_	Analyse			
Evolution	Selection	ID	Date	Heure	Туре	Modale	Nom		
		20	2014-11-24	12:00:00	surEvenement	<u>Oui</u>	pont_riviere-aux-mulets_2	2014-11-24_12-00-	
							<u>oo_surevenem</u>		
۲	0	210	2015-07-17	12:00:00	surHoraire	<u>Oui</u>	<u></u>	e.txt	

Figure 5.6 – Capture d'écran de l'onglet Listes d'enregistrements

dont le script fonctionne de la même manière que celui du fichier *vibrations_ambiantes.php* décrit à la section 5.4.2 (listing 5.7).

L'onglet se présente en deux sections, soit *Filtres* pour la sélection des paramètres de recherche optimisée proposés à l'utilisateur, et *Liste des données disponibles* pour l'affichage des enregistrements correspondant aux paramètres de recherche. Les paramètres de recherche accessibles à l'utilisateur pour la section *Filtres* sont (1) les dates et (2) les heures encadrant la recherche, (3) le type de déclencheur ayant initié l'enregistrement des signaux au niveau de l'acquisiteur, (4) la disponibilité ou non de résultats d'analyse modale pour ces enregistrements (une acquisition de données ne donne pas nécessairement lieu à une analyse modale, notamment les enregistrements sur demande de l'utilisateur) et finalement (5) l'affichage des enregistrements pour lesquels l'utilisateur a précédemment coché ou non la case permettant de les inclure dans l'attribut *evolution* de la table *utilisateurs* de la base de données (pour affichage de l'évolution des fréquences propres à l'onglet *Analyses modales - historique*). Une fois les paramètres de filtres définis, l'utilisateur soumet sa requête à la base de données en cliquant sur le bouton *Appliquer*.

L'interrogation de la base de données, notamment des tables *enregistrements*, *analyses_modales* et *utilisateurs*, qui contiennent l'ensemble des informations reliées aux critères de recherche, permet de générer la liste d'enregistrements correspondant à ces critères. Pour chaque enregistrement présenté, cette liste indique le numéro d'identifiant correspondant (attribut *enregistrement_id*), la date et l'heure de l'enregistrement (attributs *date* et *heure*), la raison du déclenchement de l'enregistrement (attribut *type*), l'existence ou non d'un résultat d'analyse modale pour l'enregistrement et le nom du fichier des données brutes.

À partir de la liste présentée, il est possible pour l'utilisateur de télécharger les fichiers d'enregistrement, soit un à un en cliquant sur l'hyperlien correspondant au nom du fichier, soit en groupe à l'aide de la sous-liste des enregistrements ayant leur case Évolution cochée et du bouton Télécharger les cochés. Le bouton Tout cocher / décocher permet, comme son nom l'indique, de cocher ou de décocher l'ensemble des enregistrements de la liste présentée afin de faciliter la tâche

```
$(document).ready(function()
1
\mathbf{2}
    {
3
         // Préparation des données avant l'appel à la fonction
         // de mise-à-jour de la liste d'enregistrement.
4
\mathbf{5}
\mathbf{6}
         updateListeEnregistrement();
    });
\overline{7}
8
9
     // 11 fonctions pour la sauvegarde des paramètres du filtre liés à la session en cours
10
    // Gestion de l'action lorsque l'utilisateur clic sur l'hyperlien "Dui".
11
    function TableAnalyseDetail(enregistrement_id)
12
13
    {
         // Diverses actions
14
    }
15
16
    // Gestion de l'action associée au clic sur le bouton "Appliquer"
17
    $("#soumettre").click(function()
18
19
    {
20
         updateListeEnregistrement();
21
    });
22
    // Gestion du bouton de téléchargement des enregistrement_id cochés.
23
    $("#Telecharge").click(function()
24
    {
25
26
         // Diverses actions
27
    });
28
    // Gestion du bouton de suppression d'un enregistrement_id.
29
    $("#supprimerEnregistrement").click(function()
30
31
    {
32
         // Diverses actions
33
    });
34
    function updateListeEnregistrement()
35
36
     {
37
         // Étape 1: Diverses actions pour la préparation de la requête
38
         // Étape 2: Requête à la base de données
39
         $.post("requete_donnees.php",
40
41
             // Diverses actions
42
         );
43
    }
```

Listing 5.8 – Extrait du code source *listes_enregistrements.js*

de l'utilisateur. Finalement, le bouton Supprimer la sélection de la base de données permet de supprimer définitivement de la base l'enregistrement pour lequel le bouton de la colonne Sélection est actif.

Si l'utilisateur clique sur l'hyperlien Évolution en tête de la liste présentée, il sera redirigé vers l'onglet Analyses modales - historique qui présentera l'évolution des fréquences propres de la structure à partir de la seule liste des enregistrements que l'utilisateur a cochés. Si l'utilisateur active l'option de la colonne Sélection pour un enregistrement (une seule sélection possible) et qu'il clique sur l'hyperlien portant le même nom, le site web sera redirigé vers l'onglet Vibrations ambiantes pour la consultation des signaux contenus dans le fichier sélectionné. La programmation de cette fonctionnalité n'étant toutefois pas complète, tel que souligné à la section 5.4.2, rien ne se passe pour le moment. Enfin, en cliquant sur l'hyperlien Oui de la colonne Analyse modale d'un enregistrement de la liste, l'utilisateur accède à l'onglet Analyses modales - détails avec les résultats d'analyse modale de l'enregistrement correspondant. Bien sûr, cela est conditionnel à ce que l'analyse modale ait été préalablement exécutée sur l'enregistrement en question, sans quoi la liste afficherait le simple texte Non à la place de l'hyperlien Oui.

Toutes ces fonctionnalités sont l'affaire du couple de fichiers listes enregistrements.html et listes enregistrements.js avec la participation conjointe des fichiers satellites requete donnees.php et supprimerEnregistrement.php. Ces fichiers totalisant 563 lignes de code ne peuvent pas être présentés de façon détaillés dans ce mémoire. Néanmoins, le listing 5.8 présente une version simplifiée du code source listes enregistrements js en y préservant sa structure. Ce code source est central aux fonctions de l'onglet traité à cette section. Lorsque l'utilisateur clique sur l'onglet Listes d'enregistrements et après le chargement complet de la page web, le navigateur web actionne automatiquement le code compris entre les lignes 1 et 7 du listing 5.8. Cette portion se termine par un appel à la fonction updateListeEnregistrement (lignes 35 à 43). Cette dernière compte en réalité pour le tier du fichier listes enregistrement. js; elle est donc majeure au fonctionnement de celui-ci. En dépit de son ampleur, la fonction est simplement la succession de deux étapes, soit préparer les données nécessaires à la formulation de la requête à la base de données, puis faire un appel au fichier requete donnees.php à l'aide de ces données. Le fichier requete donnees.php nécessite en entrée la date de début et de fin, l'heure de début et de fin, un vecteur contenant le type de déclencheur souhaité, et finalement un vecteur contenant la disponibilité ou non d'une analyse modale. Il fournit en sortie la liste des enregistrements correspondant aux paramètres demandés.

Les autres lignes du listing 5.8 concernent la gestion des actions à engager lorsque l'utilisateur clique sur un bouton (lignes 18, 24 et 30) ou sur l'hyperlien *Oui* (ligne 12). À la ligne 9, onze fonctions permettent la sauvegarde des paramètres du filtre liés à la session en cours, si bien que l'utilisateur peut naviguer sur les autres onglets du site sans perdre ses critères de sélection lorsqu'il reviendra sur l'onglet *Listes d'enregistrements*. À la déconnexion de l'utilisateur, ces paramètres ne seront cependant pas sauvegardés pour une prochaine session, contrairement à la liste des enregistrements cochés qui est directement archivée dans l'attribut *evolution* de la table *utilisateurs* de la base de données.

5.4.4 Analyses modales - historique

L'accès à l'onglet *Analyses modales - historique* se fait à travers le fichier *analyse_historique_v1.php*. Le fonctionnement de ce script est le même que celui de *vibrations_ambiantes.php* décrit à la section 5.4.2.

L'onglet est illustré à la figure 5.7 et permet la visualisation de l'évolution de toutes fréquences propres identifiées pour la structure (huit modes illustrés avec des données quelconque à la figure 5.7). L'onglet est constitué de trois sections : (1) *Filtres* pour le paramétrage d'une sous-liste



Figure 5.7 – Capture d'écran de l'onglet Analyses modales - historique

des enregistrements archivés dont on veut afficher l'évolution des résultats sur la page, (2) Graphiques d'évolutions pour l'observation graphique de l'évolution des fréquences propres obtenues des enregistrements sélectionnés, (3) Tableau d'évolution pour l'affichage des mêmes résultats sous la forme d'un tableau.

Pour la section Filtre, l'utilisateur a le choix d'afficher les résultats des enregistrements d'une liste qu'il a lui-même constituée en cochant les cases correspondantes sous l'onglet *Listes d'enregistrements*, tel que décrit à la section précédente. Dans ce cas, la liste des enregistrements présentés est constituée à partir des valeurs de l'attribut *evolution* de la table *utilisateurs*. Chaque utilisateur du système peut ainsi se constituer sa propre liste d'enregistrements à visualiser et qui s'affiche dés qu'il consulte l'onglet *Analyses modales - historique*. Sinon, un outil de filtre similaire à celui décrit pour l'onglet *Listes d'enregistrements* est mis à disposition de l'utilisateur pour constituer automatiquement une autre sélection des résultats d'analyse à visualiser. Les paramètres de recherche accessibles à l'utilisateur sont les dates et heures encadrant les enregistrements, l'inclusion de tous les enregistrements ou bien leur inclusion selon une certaine périodicité, et finalement le type de déclencheur ayant initié la formation du fichier d'enregistrement. La valeur ajoutée du filtre de cette page est la possibilité de retenir automatiquement les enregistrements en respectant au mieux possible une certaine période de temps (par ex., toutes les 2 semaines) en fonction des dates d'enregistrements disponibles. Une fois les paramètres de filtre définis, l'utilisateur soumet sa requête à la base de données en cliquant sur le bouton *Appliquer*.

La réponse à la requête de l'utilisateur permet de générer la liste d'enregistrements correspondant à ses critères de recherche. Les résultats sont affichés sous forme de graphes temporels avec échelle automatique et de tableau présentant les valeurs des fréquences pour chacun des modes. Le tableau fournit la liste des résultats sous la forme d'une ligne pour chaque enregistrement, contenant non seulement les valeurs des fréquences propres identifiées pour tous les modes, mais aussi l'identifiant de l'enregistrement correspondant (attribut *enregistrement_id*), la date et l'heure de l'enregistrement et le type de déclencheur. Il est possible d'exporter le tableau d'évolution présenté en cliquant sur le bouton *Exporter au format .txt*.

Toutes ces fonctionnalités sont gérées par analyse_historique_v1.html et analyse_historique_v1.js avec la participation des fichiers satellites EvolutionAuto_v1.php et Affichage_historique_v1.php. Ces fichiers totalisant 756 lignes de code ne peuvent pas être présentés de façon détaillée dans ce rapport. Le code source analyse_historique_v1.js est central aux fonctions de l'onglet traité à cette section. Sa structure est toutefois très semblable à celle du fichier listes_enregistrements.js présentée de façon simplifiée à la section précédente (listing 5.8), si bien qu'il n'est pas nécessaire de la reproduire ici. Notons simplement que la requête à la base de données peut provenir, comme indiqué précédemment, de la sous-liste des enregistrements cochés par l'utilisateur à l'onglet Listes d'enregistrements ou de l'application des paramètres du filtre de sélection automatique. Dans les deux cas, un appel au fichier Affichage_historique_v1.php est fait avec un vecteur contenant les attributs enregistrement_id d'intérêt pour l'utilisateur. La sortie de ce fichier est une matrice au format JSON contenant les informations affichées dans le tableau d'évolution. Une fois réacheminées au navigateur web de l'utilisateur, les informations contenues dans la matrice sont utilisées pour le tracé des graphiques et le remplissage du tableau d'évolution.

5.4.5 Analyses modales - détails

L'onglet Analyses modales - détails est illustré à la figure 5.8 et permet à l'utilisateur de visualiser les résultats détaillés de l'analyse modale d'un enregistrement particulier. Les informations sont directement puisées dans les attributs de la table *analyses_modales* de la base de données, euxmêmes issus de l'analyse effectuée par le logiciel Artemis Modal.

STSUS	×	to a second and the second second				_	- 0 <mark>- x</mark>
\leftrightarrow \rightarrow C \odot N	on sécurisé 10.43.102.	114/pont_riviere-aux-mulet	s/				야 ☆ 💹 👂 :
		Système de t	élésurveillai	nce des structures de l	l'Université de She	rbrooke	
Accueil	Vibrations an	nbiantes Listes d	l'enregistrements	Analyses modales - historique	Analyses modales - détails	Configurations	Journal de bord
* Enregistre ID : 584 (20 Résulta	ment selectionné c 15-07-18 20:00:00 ts d'identific	lans la liste des enreg	istrements:				
No. Mode	Fréquence (Hz)	Amortissement (%)	Complexité (%)				x r
Mode 1	1.0000	0.0010	0.0030				
Mode 2	2.0000	0.0010	0.0030		8	~	
Mode 3	3.0000	0.0010	0.0030			1000	
Mode 4	4.0000	0.0010	0.0030		l l		
Mode 5	5.0000	0.0010	0.0030		A		
Mode 6	6.0000	0.0010	0.0030		- Se		Mode 1
Mode 7	7.0000	0.0010	0.0030				Mode 2
Mode 8	8.0000	0.0010	0.0030				Mode 3 Mode 4
Mode 9	9.0000	0.0010	0.0030			Anima	Mode 5 Mode 6 Mode 7 Mode 8 Aucun tion Arrêt Mode 1 •

Figure 5.8 - Capture d'écran de l'onglet Analyses modales - détails

L'onglet présente dans la partie de gauche de la page un tableau avec les fréquences, amortissements et complexités de chacun des modes identifiés pour la structure. Dans la partie droite, un géométrie filaire dynamique du pont est illustrée et permet l'affichage et l'animation du mode désiré. La géométrie du pont peut être manipulée par l'utilisateur à l'aide de sa souris en modifiant l'orientation 3D et en zoomant. Un système d'axes aide l'utilisateur à la tâche en affichant l'orientation de la géométrie à la suite des manipulations. Les boutons *Animation* et *Arrêt*, ainsi que le sélecteur de mode permettent respectivement le démarrage de l'animation et son arrêt, ainsi que le choix du mode (huit modes dans l'exemple de la figure 5.8). Si aucun mode n'est sélectionné, la géométrie originale du pont est présentée.

Toutes ces fonctions s'appliquent pour l'enregistrement dont le bouton *Sélection* a été activé à l'onglet *Listes d'enregistrements* ou à l'enregistrement dont l'utilisateur a cliqué sur l'hyperlien *Oui* dans ce même onglet. L'enregistrement sélectionné et dont les résultats d'analyse modale sont affichés est indiqué en haut à gauche de l'onglet *Analyses modales - détails* avec son identifiant, sa date et son heure. En cliquant sur l'hyperlien correspondant à cette indication, le site web renvoie l'utilisateur à l'onglet *Listes d'enregistrements*, où il aura le loisir de modifier son choix.

Toutes ces fonctionnalités sont l'affaire du couple analyse_detail.html et analyse_detail_v4.js conjointement avec les fichiers satellites Affichage_detail_al.php et DetailNomEnregistrement.php. Ces fichiers totalisant 857 lignes de code ne peuvent pas être présentés de façon détaillée dans ce rapport. Néanmoins, un nombre limité d'étapes intervient lors de la consultation de l'onglet. Celles-ci sont décrites au tableau 5.4 et leur organisation logique est illustrée à la figure 5.9. L'onglet fonctionne à deux chemins possibles : (1) l'utilisateur accède à l'onglet Analyses modales - détails en ayant sélectionné un enregistrement valide (i.e. avec une analyse modale disponible) ou sans aucune sélection (dans ce cas, l'enregistrement le plus récent est utilisé), puis les informations extraites de la base de données se retrouvent affichées dans le tableau et le conteneur 3D; (2) l'utilisateur accède à l'onglet Analyses modales - détails avec un enregistrement non-valide (sans analyse modale), auquel cas la page invite l'utilisateur à en choisir un autre.

Étapes	Description
A0	L'utilisateur accède à l'onglet sans avoir sélectionné d'enregistrement
A1	L'utilisateur accède à l'onglet en ayant sélectionné un enregistrement pour lequel un résultat d'analyse modale n'est pas disponible
A2	L'utilisateur accède à l'onglet en ayant sélectionné un enregistrement pour lequel un résultat d'analyse modale est disponible
B0	Le script extrait les informations correspondant à l'enregistrement le plus récent
B1	L'onglet invite l'utilisateur à choisir un autre enregistrement
B2	Le script extrait les informations correspondant à l'enregistrement choisi par l'utilisateur
C0	L'onglet reçoit les informations du serveur et les arrange dans la section $R\acute{e}sultats \ d'identification$
D0	L'onglet initialise le conteneur 3D pour la production du rendu du pont

Tableau 5.4 – Étapes de fonctionnement de l'onglet Analyses modales - détails



Figure 5.9 - Organisation logique des étapes de l'onglet Analyses modales - détails

La partie la plus délicate à programmer de l'onglet Analyses modales - détails est sans conteste l'affichage dynamique 3D des modes avec leur animation. Pour cela, la bibliothèque Three.js (https://threejs.org/), en distribution libre, a été utilisée. Sans entrer dans les détails du code, l'affichage fonctionne essentiellement en récupérant les coordonnées x, y et z des nœuds inclus dans la déformée modale à afficher (table noeuds et attribut list_noeuds de la table analyses_modales) pour afficher la géométrie du pont. L'allure des déformées est ensuite obtenue en ajoutant à ces coordonnées le vecteur de déformée modale selon les trois directions x, y et z, constitué à partir des attributs amplitude_* et phase_* de la table analyses_modales. L'animation est effectuée simplement par une succession d'images de la géométrie à intervalles de temps réguliers en multipliant à chaque image l'amplitude de la déformée ajoutée par un facteur oscillant entre -1 et 1.

5.4.6 Configurations

L'onglet *Configurations*, illustré à la figure 5.10, permet à l'utilisateur et au super-utilisateur de mettre à jour ou de consulter les paramètres configurables du système de télésurveillance, no-tamment toutes les variables utilisées par le VI LabVIEW décrites au chapitre 4 pour le fichier de configuration (section 4.5) et certaines informations intrinsèques aux capteurs déployés archivées dans la base de données dans les tables *canaux*, *capteurs* et *noeuds*.

L'onglet *Configurations* distingue les utilisateurs des super-utilisateurs à partir de l'attribut super_utilisateur contenu dans la table utilisateurs de la base de données. Un simple utilisateur aura ainsi uniquement accès à une version réduite de la page présentée à la figure 5.10, soit seulement la partie supérieure de la page (en gris). L'utilisateur a accès aux seuls paramètres d'utilisation du système, soit les variables définissant le comportement du VI LabVIEW pour les déclencheurs d'enregistrement surHoraire, surUtilisateur et surÉvenement. Les paramètres configurables de la

TSUS	×											
C 🛈 Non s	sécurisé 10.43.102.114/pont_rivi	ere-aux-mulets/			야 ☆ 🗾							
	Stration	o do télécumoillor	aa daa atmuatuwaa da	UUniversité de She	ubuo o Iro							
	System	ie de telesurveillar	ice des structures de	I Universite de Sne	rdrooke							
Accueil	Vibrations ambiantes	Listes d'enregistrements	Analyses modales - historique	Analyses modales - détails	Configurations Journal de bord							
					Mise à jour du système							
arametra	age des enregistr	ements "surHoraire	•									
ON 1 : 15,30	0 11-13 * *											
ON 2 : 0 0 2	46											
irée de la mé	émoire flottante (ex.: 5 se	condes, 3 minutes,) : 5	secondes •									
	ana dan anyanisty	ana anta "auni lettiaati										
arametra	age des enregistr	ements surutilisate	eur :									
rée de la mé	émoire flottante (ex.: 5 se	condes, 3 minutes,) : 3	minutes 🔹									
aramètr	age des enregistr	ements "surFvenem	ent" ·									
	age des ennegiser	enteries sargrenen										
rée de la mé	émoire flottante enregistre	ée avant l'événement : 5	secondes 🔹									
irée de la mé	émoire flottante enregistre	ée après l'événement : 2	minutes •	Durée de la ménoire foutante energistre avait tévrientent : 2 ménoire se la ménoire foutante energistre avait tévrientent : 2 ménoire energistre energistre avait tévrientent : 2 ménoire energistre								
Duree de la memoine routaine entregistree après revenement : 2 minutes •												
n cas d'événei	ment, envoyer un message	à:										
1 cas d'événe ex.loignon@l	ment, envoyer un message usherbrooke.ca	èà:										
n cas d'événe ex.loignon@l	ment, envoyer un message usherbrooke.ca	à :										
। cas d'événe. ex.loignon@।	ment, envoyer un message usherbrooke.ca	- à :										
cas d'événe ex.loignon@i	ment, envoyer un message usherbrooke.ca	à: Ilonnage et des cana	aux d'acquicition :									
i cas d'événe ex.loignon@i onfigura	ment, envoyer un message usherbrooke.ca	llonnage et des cana	aux d'acquisition :									
cas d'événe ex.loignon@u onfigura hantillonnag	ment, envoyer un message usherbrooke.ca Ition de l'échanti e réel : 2000.00 • (Hz) e effectif (annés décimati	llonnage et des cana	aux d'acquisition :									
cas d'événe ex.loignon@(onfigura nantillonnag nantillonnag	ment, envoyer un message usherbrooke.ca tion de l'échanti e réel : 2000.00 • (Hz) e effectif (après décimatio	llonnage et des cana n) : 200	aux d'acquisition :									
cas d'événe ex.loignon@i onfigura hantillonnag hantillonnag o. Capteur	ment, envoyer un message usherbrooke.ca Ition de l'échanti e réel : 2000.00 • (Hz) e effectif (après décimatio No. de série	e à : Ilonnage et des cana on) : 200 (Sensibilité	aux d'acquisition : Hz) Unité									
cas d'événe ex.loignon@ onfigura hantillonnag hantillonnag o. Capteur	ment, envoyer un message usherbrooke.ca Ition de l'échanti e réel : 2000.00 • (Hz) e effectif (après décimation No. de série 20076	a : Ilonnage et des cana on) : 200 (Sensibilité 254.84	aux d'acquisition : Hz) Unité (mV)/(r	n.s^-2)								
cas d'événe ex.loignon@ onfigura hantillonnag hantillonnag o. Capteur	ment, envoyer un message usherbrooke.ca Ition de l'échanti e réet : 2000.00 • (Hz) e effectif (après décimation No. de série 20076 AC-61-H-01	e à : Ilonnage et des cana on) : 200 (Sensibilité 254.84 339.78	Hz) (mY)/(r (mY)/(r	n.s^-2) n.s^-2)								
cas d'événe ex.loignon® onfigura hantillonnag hantillonnag io. Capteur	ment, envoyer un message usherbrooke.ca Ition de l'échanti e réet : 2000.00 • (Hz) e effectif (après décimation No. de série 20076 AC-61-H-01 ES-U2-01	a à : Ilonnage et des cana on) : 200 (Sensibilité 254.84 339.78 1019.3	Aux d'acquisition : Hz) Unité (mV)/(r (mV)/(r (mV)/(r	n.s^-2) n.s^-2) n.s^-2)								
cas d'événe ex.loignon® onfigura hantillonnag hantillonnag o. Capteur	ment, envoyer un message usherbrooke.ca Ition de l'échanti e réel : 2000.00 • (Hz) e effectif (après décimation No. de série 20076 AC-61-H-01 ES-U2-01 W2-AC-DC	a à : Ilonnage et des cana on) : 200 (Sensibilité 254.84 339.78 1019.3 551.9	aux d'acquisition : Hz) Unité (mV)/(i (mV)/(i (mV)/(i (mV)/(i	n.s^-2) n.s^-2) n.s^-2) n.s^-2)								
cas d'événe ex.loignon@i onfigura hantillonnag hantillonnag o. Capteur o. Capal	ment, envoyer un message usherbrooke.ca Ition de l'échanti te réel : 2000.00 • (Hz) te effectif (après décimation No. de série 20076 AC-61-H-01 ES-U2-01 W2-AC-DC No. Capteur	e à : Ilonnage et des cana on) : 200 (Sensibilité 254.84 339.78 1019.3 551.9 Position (x, y, z)	Aux d'acquisition : Hz) Unité (mV)/(r (mV)/(r (mV)/(r (mV)/(r (mV)/(r (mV)/(r (mV)/(r (mV)/(r (mV)/(r)	n.s°-2) n.s°-2) n.s^-2) n.s^-2) Seuil minimum	Seuil maximum							
cas d'événe ex.loignon@i onfigura hantillonnag hantillonnag o. Capteur	ment, envoyer un message usherbrooke.ca Ition de l'échanti e réel: 2000.00 • (Hz) e effectif (après décimation No. de série 20076 AC-61-H-01 ES-U2-01 W2-AC-DC No. Capteur 1	a) : 200 (Sensibilité 254.84 339.78 1019.3 551.9 Position (x, y, z) (0, 0, 0)	Aux d'acquisition : Hz) Unité (mV)/(r (mV)/(r (mV)/(r (mV)/(r Orientation (x, y, z) (1, 0, 0)	n.s^-2) n.s^-2) n.s^-2) Seuil minimum -0.025484	Seuil maximum 0.025484							
cas d'événe ex.loignon@i onfigura hantillonnag o. Capteur o. Capteur	ment, envoyer un message usherbrooke.ca tion de l'échanti e réel : 2000.00 • (Hz) e effectif (après décimation No. de série 20076 AC-61-H-01 ES-U2-01 W2-AC-DC No. Capteur 1 2	tà : illonnage et des cana on) : 200 (Sensibilité 254.84 339.78 1019.3 551.9 Position (x, y, z) (0, 0, 0) (0, 0, 0)	Aux d'acquisition : Hz) Unité (mV)/(i)))))))))))))))))))))))))))))))))))	n.s^-2) n.s^-2) n.s^-2) Seuil minimum -0.025484 -0.033978	Seuti maximum 0.025484 0.033978							
I cas d'événe ex.loignon©i onfigura hantillonnag hantillonnag o. Capteur	ment, envoyer un message usherbrooke.ca Ition de l'échanti e réel : 2000.00 • (Hz) e effectif (après décimation No. de série 20076 AC-61-H-01 ES-U2-01 W2-AC-DC No. Capteur 1 2 3	E à : Illonnage et des cana on) : 200 (Sensibilité 254.84 339.78 1019.3 551.9 Position (X, y, z) (0, 0, 0) (0, 0, 0) (0, 0, 0)	Aux d'acquisition : Hz) Unité (mV)/(r (mV)/(r (mV)/(r Orientation (x, y, z) (1, 0, 0) (0, 1, 0) (0, 0, 1)	n.s^-2) n.s^-2) n.s^-2) Seuti minimum -0.025484 -0.033978 -0.0193	Seuti maximum 0.025484 0.033978 0.10193							
n cas d'événe lex. Loignon©i onfigura chantillonnag chantillonnag ko. Capteur lo. Captal	ment, envoyer un message usherbrooke.ca Ition de l'échanti e réel : 2000.00 • (H2) e effectif (après décimation No. de série 20076 AC-61-H-01 ES-U2-01 W2-AC-DC No. Capteur 1 2 3 4	a): 200 () Sensibilité 254.84 339.78 1019.3 551.9 Position (x, y, z) (0, 0, 0) (0, 0, 0) (0, 0, 0) (0, 0, 0) (20, 0, 0)	Aux d'acquisition : Hz) Unité (mV)/(t (mV)/(t (mV)/(t Orientation (x, y, z) (1, 0, 0) (0, 1, 0) (0, 0, 1) (1, 0, 0)	n.s ^{°-} 2) n.s ^{°-} 2) n.s ^{°-} 2) Seuil minimum -0.025484 -0.033978 -0.010193	Seuil maximum 0.025484 0.033978 0.10193							

Figure 5.10 – Capture d'écran de l'onglet Configurations

section *surHoraire* concernent les trois plages horaires possibles d'enregistrements préprogrammées ainsi que la durée de ces enregistrements. Le paramètre sous la section *surUtilisateur* concerne la durée de la mémoire flottante dans le système d'acquisition, correspondant à la durée de la file *surUtil* (fig. 4.8) et donc à la durée d'enregistrement, lorsqu'un enregistrement des signaux est demandé manuellement par l'utilisateur sous l'onglet *Vibrations ambiantes*. Finalement, les trois paramètres de la section *surEvenement* concernent, dans l'ordre, la durée d'enregistrement requise avant le dépassement d'un seuil (correspondant à la durée de la mémoire flottante dans le système d'acquisition et donc à la durée de la file *surEven* à la figure 4.8), la durée requise d'enregistrement après le dépassement d'un seuil, ainsi qu'une liste d'adresses courriel à contacter à la suite d'un évènement.

La façon de noter les heures d'enregistrement préprogrammées dans la section *surHoraire* n'est pour le moment pas intuitive. Elle est cependant très versatile et fonctionne selon la logique d'utilisation du logiciel CRON dont le format est compris par le VI LabVIEW (fig. 4.16). Ainsi, le premier paramètre dans la notation CRON concerne la notion de minutes, le deuxième la notion d'heures, le troisième la notion de jour du mois et finalement le dernier la notion de mois. Dans ce code, le séparateur des paramètres est le caractère *espace*. La virgule permet de définir deux valeurs pour le même paramètre, et il ne doit pas y avoir d'espace après celle-ci. Le tiret permet de définir une plage de valeurs, encore une fois sans espace après celui-ci. L'étoile permet de considérer tous les créneaux horaires existants pour le paramètre en question. À des fins d'illustration, voici le décodage des deux premiers exemples fournis à la figure 5.10 :

- La première heure programmée sur la figure 5.10, correspondant à la chaîne de caractère 15,30 11-13 * *, signifie qu'il y aura enregistrement des données lorsque l'horaire correspond : (1) aux quinzième et trentième minutes; (2) entre onze heure et treize heure inclusivement; (3) pour tous les jours du mois; (4) pour tous les mois de l'année. On a donc des enregistrements tous les jours à 11h15, 11h30, 12h15, 12h30, 13h15 et 13h30.
- La deuxième heure programmée sur la figure 5.10, correspondant à la chaîne de caractère 0 0 24 6, signifie qu'il y aura enregistrement des données lorsque l'horaire correspond : (1) à la minute zéro; (2) à l'heure zéro; (3) pour le vingt-quatrième jour du mois; (4) au sixième mois de l'année. Il y aura donc un enregistrement à minuit tous les 24 juin.

En plus d'avoir accès aux paramètres de l'utilisateur, le super-utilisateur a un accès réservé à la partie inférieure de la page (fig. 5.10), en jaune. On y trouve des paramètres intrinsèques à la définition du système d'acquisition déployé in situ sur la structure. On comprend donc que ces éléments ont un certain degré de protection. Le super-utilisateur peut y définir les fréquences d'échantillonnage réelle (prise dans les choix offerts par le CompactRIO) et effective. De plus, il peut définir les capteurs avec leur numéro de série, leur sensibilité et l'unité dans laquelle est exprimée la valeur qu'ils mesurent. Les canaux du système d'acquisition sont ensuite paramétrables par le super-utilisateur en associant à chaque canal du CompactRIO un capteur, la position de ce capteur, son orientation (vecteur directeur à trois coordonnées x, y et z) et les seuils minimum et maximum de déclenchement d'un événement pour le capteur (exprimé dans son unité d'ingénierie).

Au clic sur le bouton *Mise à jour du système*, un script envoie à la base de données toutes les informations insérées dans les champs réservés au super-utilisateur en les distribuant à travers les tables *canaux*, *capteurs* et *configurations*. Ensuite, une liste de tâches (section 5.5.3) génère un nouveau fichier de configuration et ce fichier est acheminé via FTP au CompactRIO. Une fois le transfert complété, le navigateur web envoie un message *Websocket* à la machine d'état du CompactRIO pour entamer le processus de reconfiguration (section 4.3.2). En l'état actuel du prototype, le fichier de configuration doit encore être construit et téléversé manuellement pour que la mise à jour s'effectue correctement.

5.4.7 Journal de bord

L'onglet Journal de bord a pour objectif la consignation et la consultation par les utilisateurs de notes pertinentes archivées dans la table journal de la base de données et dans le fichier journal.txt du CompactRIO. Ces notes peuvent être insérées manuellement par les utilisateurs ou bien peuvent être générées automatiquement par le CompactRIO lors de certains états de la machine d'état (chap. 4). Cet onglet, d'une programmation qui sera assez simple, est encore en développement au stade actuel du prototype.

5.5 LISTES DE TÂCHES

Le fonctionnement général du prototype de système de télésurveillance repose sur trois listes de tâches. Celles-ci ont pour mission : (1) le transfert périodique et l'archivage adéquat des nouveaux fichiers de données disponibles enregistrés sur le CompactRIO; (2) le déclenchement d'une analyse modale automatique à partir d'un enregistrement donné; (3) la génération et le téléversement sur le CompactRIO d'un nouveau fichier de configuration. Ces tâches sont brièvement présentées dans les paragraphes qui suivent.

5.5.1 Transfert des données du CompactRIO vers le serveur informatique

Le transfert et l'archivage des fichiers de données du CompactRIO vers le serveur informatique est certainement la tâche la plus importante, car c'est elle qui nourrit la base avec les données recueillies sur le pont. Cette tâche est mise en route à l'initiative du planificateur de tâches Windows, illustré à la figure 5.11, qui la réexécute à un intervalle de temps défini par l'administrateur du système de télésurveillance. La figure 5.11 montre l'exemple de l'exécution toutes les cinq minutes du fichier transfert_fichiers_cRIO_vers_serveur.py. L'intervalle de temps peut être choisi à volonté et pourrait notamment être réduit, par exemple à 30 secondes, si l'utilisateur souhaite minimiser le délai entre une demande manuelle d'enregistrement faite à l'onglet Vibration ambiantes et la mise à disposition de l'enregistrement correspondant. Il est important de comprendre que le CompactRIO n'envoie pas les fichiers d'enregistrement des signaux après les avoir générés, mais que c'est le serveur qui vérifie la disponibilité de nouveaux fichiers à travers la présente tâche.

Planificateur de tâches	of Sheet was	
Fichier Action Affichage ?		
💠 🔿 🖄 📰 📓 📰		
Fichier Action Affichage ?	Nom Statut Déclencheurs Prochaine exécution © copier Do: Préf A 1655 louz les jours - Après le déclenchement, recommencer tous les 5 minutes indefiniment. 2017-01-12 1283511 © Coopier Do: Préf A 1655 louz les jours - Après le déclenchement, recommencer tous les 1 heure pendant 1 jour. 2017-01-12 1283511 © Coopier Do: Préf A 1835 tous les jours - Après le déclenchement, recommencer tous les 1 heure pendant 1 jour. 2017-01-12 150000 © Külupdate. Prét A 15000 tous les jours - Après le déclenchement, recommencer tous les 1 heure pendant 12 be 2017-01-12 150000 © Külupdate. Prét A 1000 tous les jours - Après le déclenchement, recommencer tous les 1 heure pendant 12 be 2017-01-12 160000 © KÜlupdate. Prét A 1000 tous les session d'un utilisateur Sult 70000 © NUlupdate. Prét A 1000 tous les jours - Après le déclenchement, recommencer tous les 1 heure pendant 12 be 2017-01-12 160000 © NUlupdate. Prét A 1000 tous les jours - Après le déclenchement, recommence tous les 1 heure pendant 12 be 2017-01-12 160000 © NUlupdate. Prét A 1000 tous les jours - Après le déclenchement, recommence tous les 1 heure pendant 12 be 2017-01-12 160000 <td< td=""><td>Actions Bibliothèque du Planificateur de tâches Importer une tâche Importer une tâche Importer une tâche Afficher toutes les tâches actives Désactiver Thistorique de toutes les tâc Nouveau dossier Affichar toutes les tâches actives dictage Actualiser Aide Elément sélectionné Escuter Propriétés Supprimer Aide</td></td<>	Actions Bibliothèque du Planificateur de tâches Importer une tâche Importer une tâche Importer une tâche Afficher toutes les tâches actives Désactiver Thistorique de toutes les tâc Nouveau dossier Affichar toutes les tâches actives dictage Actualiser Aide Elément sélectionné Escuter Propriétés Supprimer Aide
<	Général Détails Le Planificateur de tâches a terminé l'instance « (6b76d91b-6c4b-4c04-8633-740cc2757daa) » de la tâche « ProuberDonneetCRIO » pour l'utilicateur « ISHFRBROOKFJoial 501 ». Journal · Microsoft-Windows-TackScheduler/Dinécationnal	

Figure 5.11 – Planificateur de tâches Windows

Le fichier $transfert_fichiers_cRIO_vers_serveur.py$ est une liste de commandes Python organisées en six étapes :

- 1. Vérification du dernier transfert de fichiers entre le CompactRIO et le serveur : une date trop ancienne (à définir par l'utilisateur, par ex. supérieure à une semaine) indique un probable problème de communication avec le CompactRio et une alerte est émise par courriel pour effectuer une vérification. Les tâches suivantes ne sont alors pas exécutées;
- 2. Téléchargement des nouveaux fichiers de données du CompactRIO sur le système de fichiers du serveur informatique;

- 3. Suppression des fichiers téléchargés sur le CompactRIO afin de libérer l'espace mémoire limité du système d'acquisition;
- 4. Extraction des métadonnées incluses dans le nom du fichier de données généré par le CompactRIO lors de l'enregistrement;
- 5. Insertion des nouvelles métadonnées dans la base de données;
- 6. Déplacer les fichiers de données dans le répertoire de référence sur le serveur de données et en créer une copie sous le nom *signaux_enregistrements.dat* dans le dossier d'analyse modale, le cas échéant.

Il convient de préciser l'étape d'extraction des métadonnées nécessaires à la base, soit la date et l'heure d'enregistrement ainsi que le type de déclencheur impliqué (respectivement pour les attributs *date, heure* et *type* de la table *enregistrements*). Ces informations sont directement obtenues du nom du fichier téléchargé, formaté de manière systématique par le CompactRIO lors de l'enregistrement. Le format du nom de fichier est du type pont_nom-du-pont_date_heure_déclencheur.txt (par ex., pont_riviere-aux-mulets_2014-11-24_12-00-00_surEvenement.txt), ce qui permet un transfert aisé des informations requises en décodant le nom du fichier de manière séquentielle par la recherche des caractères "__".

5.5.2 Analyse modale automatique

L'objectif principal du prototype de système de télésurveillance développé dans ce projet est d'effectuer un suivi automatisé, continu et en temps réel des propriétés modales du pont instrumenté. À cette fin, il est indispensable qu'une liste de tâches automatisées soit chargée de mener l'extraction des propriétés modales à partir des fichiers d'enregistrement des signaux collectés in situ par le CompactRIO. Actuellement dans le prototype, la tâche est prévue pour s'exécuter systématiquement chaque fois qu'un nouveau fichier de données est transféré du CompactRIO au serveur informatique, donc immédiatement après la liste des tâches décrite à la section précédente. Ultérieurement, il est envisagé de rendre la tâche d'analyse modale optionnelle dans le cas d'enregistrements venant d'une demande de l'utilisateur à l'onglet *Vibrations ambiantes* pour éviter un encombrement inutile de la base de données lorsque cette demande d'enregistrement d'utilisateur a un autre but que le besoin de données modales.

Plusieurs méthodes d'analyses modales à partir de données brutes de vibrations ambiantes de structures existent dans la littérature [7]. Les deux principales sont les méthodes Stochastic Subspace Identification (SSI) et Frequency Domain Decomposition (FDD). La méthode SSI offre l'avantage d'être en soi une méthode automatique qui ne requiert aucune intervention de l'utilisateur entre l'entrée des données et l'obtention de modes identifiés. Au regard des auteurs, cette méthode manque cependant de robustesse pour un système de télésurveillance qui aspire à détecter de l'endommagement, car elle est extrêmement sensible au rapport signal sur bruit des données, ne garantit pas la possibilité d'extraire tous les modes d'intérêt et n'offre aucune possibilité de forcer l'identification d'un mode manquant même si si celui-ci devait être de moindre précision. Le choix s'est donc porté sur la méthode FDD qui est plus robuste, plus précise et qui offre des possibilités d'intervention supplémentaires à l'utilisateur. La contrepartie du dernier argument est évidemment que la méthode n'est pas automatique et qu'il faut donc rationaliser et programmer les choix habituellement faits par un utilisateur pour l'automatiser. Les détails sur les méthodes SSI et FDD sortent du cadre de ce rapport et le lecteur intéressé pourra, au besoin, consulter la référence [7] ou encore le rapport CRGP-2015/04 [30] remis au MTQ dans le cadre du projet de recherche global R732.1. Pour des questions de compréhension de l'algorithme décrit dans les prochains paragraphes, il convient néanmoins de résumer les étapes de calcul effectués par la méthode d'analyse FDD :

- Les données temporelles brutes subissent d'abord plusieurs traitement préliminaires et subissent une transformation de Fourier, qui permet d'obtenir le contenu fréquentiel de chacun des signaux.
- Les transformées de Fourier des signaux sont combinées entre elles pour obtenir la matrice spectrale des densités spectrales de puissance, qui est ensuite décomposée en valeurs singulière (SVD) selon trois matrices, dont une qui contient les spectres des valeurs singulières.
- L'utilisateur sélectionne les pics des valeurs singulières dont la plupart ont lieu à des fréquences qui correspondent aux modes propres de la structure. On obtient ainsi la fréquence et les déformées des modes (venant, eux, d'une autre matrice de la décomposition SVD).
- L'utilisateur doit contrôler que les résultats obtenus correspondent bien à des modes propres en inspectant les déformées.

Une amélioration de la méthode FDD est connue sous le nom *Enhanced FDD* (EFDD) qui posttraite automatiquement les spectres de valeurs singulières localement au niveau des pics identifiés par l'utilisateur. Il en ressort une estimation améliorée de la fréquence propre identifiée et de la déformée modale, ainsi qu'une estimation de l'amortissement modal (inaccessible avec la FDD).

Considérant que la méthode EFDD est implémentée dans le logiciel Artemis Modal [33] qui est reconnu pour sa fiabilité, et qu'autant l'université de Sherbrooke que le Ministère disposent d'une licence payante pour ce logiciel, il a été décidé de s'appuyer dessus pour le développement de la tâche d'analyse modale du prototype plutôt que de reprogrammer entièrement la méthode. Artemis Modal requiert habituellement d'être utilisé à travers son interface graphique. Il est cependant possible de piloter ses fonctions de calcul fondamentales par un script extérieur en utilisant les interfaces COM de Windows. La documentation du fournisseur sur le sujet est malheureusement inexistante et le fournisseur s'est avéré très hésitant à fournir de l'aide sur le sujet, car cela s'approche du cœur du fonctionnement du logiciel. Cela explique qu'actuellement la tâche d'analyse modale automatique n'est pas encore complétée. Un fichier d'exemple de pilotage par interface COM, programmé en Matlab, a bien été donné par le fournisseur, mais celui-ci utilisait uniquement la méthode SSI. Le travail d'adaptation pour la méthode EFDD est donc encore en cours. Le principal problème est que les méthodes COM pour la méthode EFDD doivent être identifiées et leur syntaxe trouvée (le fournisseur ne souhaitant pas répondre sur le sujet, bien qu'il ait fourni la licence d'utilisation appropriée nécessaire). Une difficulté supplémentaire s'est ajoutée du fait que les distributions actuelles 64-bit de Matlab ne supportent pas le type d'interface COM requis par les méthodes d'Artemis Modal. La programmation de la tâche doit donc être effectuée avec une distribution 32-bit de Matlab, ce qui implique aussi d'utiliser une distribution 32-bit du logiciel Artemis Modal. Seule la syntaxe d'une des méthodes COM nécessaires d'Artemis Modal reste à identifier. Les auteurs ont pu faire fonctionner les autres méthodes requises et piloter avec succès le logiciel par script. Une méthodologie pour automatiser la méthode EFDD a aussi été mise au point, programmée sous Matlab et validée (en effectuant manuellement sous l'interface d'Artemis Modal les opérations à effectuer à travers l'interface COM). Notons que le logiciel Artemis Modal propose une option d'identification automatique des modes selon la méthode FDD et EFDD. L'expérience montre cependant que cette option n'est pas fiable, ni robuste et qu'elle ne peut pas être envisagée dans le cadre du projet.

Trois logiciels sont impliqués dans la liste de tâches d'analyse modale, soit Python, Matlab et Artemis Modal. Cependant Matlab n'a pas besoin d'être installé sur le serveur informatique et aucune licence n'est donc requise. Le code Matlab est en effet compilé sous forme de fichier exécutable et ne fait appel qu'à des fonctions libres de droit incluses dans la librairie Matlab Runtime qui doit être installée sur le serveur. Puisque plusieurs logiciels interviennent avec de trop nombreuses lignes de codes, les scripts de la tâche d'analyse modale automatisée ne sont pas présentés dans ce rapport. Le tableau 5.5 résume néanmoins les principales étapes de la tâche en identifiant à chaque fois les logiciels mis en jeu et les fonctions clé appelées.

Tableau 5.5 – Liste des tâches	de l'algorithme	d'analyse mod	ale automatique
--------------------------------	-----------------	---------------	-----------------

No	Étape	Logiciel	Fonctions ou fichiers principaux
1	Exécution du code Matlab compilé au format	Python	$analyse_modale_Matlab.exe$
2	Ouverture de deux serveurs d'interface COM pour Artemis Modal	Matlab	classes $SVSLicMgr.dll$ et $SVSDataModel.dll$
3	Vérification de la licence Artemis Modal	Artemis Modal	méthode COM AddConsumer
4	Configuration de l'analyse et importation des signaux bruts	Artemis Modal	méthode COM CreateFromFile et fichier de configuration *.cfg
5	Traitement des signaux, calcul de la matrice spectrale et SVD	Artemis Modal	méthode COM StartProcessing
6	Exportation du spectre de la première valeur singulière	Artemis Modal	méthode COM GetFunction
7	Recherche automatique optimisée des pics du spectre	Matlab	-
8	Identification EFDD des fréquences identifiées	Artemis Modal	méthodes COM ISVSModalEstimatorEFDD et CreateMode
9	Exportation des modes identifiés	Artemis Modal	méthodes COM GetMode, ISVSBaseMode, GetFrequency, GetDamping, GetRawModeShape
10	Fermeture des serveurs d'interface COM	Matlab	-
11	Élimination des erreurs d'identification auto- matique	Matlab	matrice des MAC
12	Calcul de la complexité des modes	Matlab	fonctions DelaunayTri et convexHull
13	Écriture des résultats dans un fichier temporaire	Matlab	-
14	Archivage des résultats dans la base de données	Pyhton	API mysqli

La tâche débute avec le logiciel Python qui, à la fin de sa tâche de transfert des fichiers d'enregistrements (section 5.5.1), demande l'exécution du programme Matlab (*analyse_modale_Matlab.exe*) pour effectuer automatiquement l'analyse modale (étape 1). Le code Matlab débute avec la création de deux sessions d'interface COM (étape 2), l'une permettant d'abord la vérification d'une licence valide par Artemis Modal (étape 3) et l'autre donnant accès aux fonctions de calcul qui sont le cœur du logiciel. Parmi ces fonctions, le premier bloc appelé permet à Artemis Modal de charger le fichier de configuration (fichier ASCII avec l'extension .cfg) que l'utilisateur doit avoir préalablement préparé en suivant rigoureusement la syntaxe requise par Artemis (étape 4). Ce fichier est nécessaire même pour une utilisation du logiciel à travers son interface graphique et ne requiert donc pas de travail ni de compétences additionnels pour l'utilisateur. Une section supplémentaire doit cependant être ajoutée au fichier *.cfg habituel pour définir des paramètres de calcul qui sont autrement entrés par l'utilisateur sous l'interface graphique (listing 5.9). Le chargement du fichier *.cfg implique également l'importation du fichier contenant les signaux bruts des capteurs à traiter. Pour cela, la tâche de transfert d'un nouveau fichier de données (section 5.5.1) a copié le fichier

Processing	
Detrend 1	<pre>// Detrending (Enable: 1[0n]/0[0ff])</pre>
Decimate 1	// Decimation (Degree: 1,2,3,4,5,10,20,30,40,50,100)
Filter 0	<pre>// Filtering (Enable: 1[0n]/0[0ff])</pre>
Projection 1 0	<pre>// Projection Channels (Enable: 1[0n]/0[0ff]), (Number of Channels: 0 [Auto], 1,2,3,4)</pre>
Harmonics O	<pre>// Harmonic Detection (Enable: 1[On]/0[Off])</pre>
Spectrum 1024	<pre>// Spectral Density: (Number of Frequency Lines: 2,4,8,16,32,64,128,256,512,1024,2048,)</pre>
SSI 1 100 20 20 100 1	// SSI Configuration: (Enable: 1[On]/0[Off]),
Estimators 1 0 0	
Stability 1 0.2048 5	0.05 1 0.05 5

78 Chapitre 5. Serveur informatique

Listing 5.9 – Code de la section additionnelle du fichier de configuration *.cfg d'Artemis Modal

dans un répertoire dédié à l'analyse modale (celui contenant aussi le fichier *.cfg) et le renomme de manière fixe en signaux enregistrements. dat pour que le même fichier *.cfg reste valide d'un enregistrement à l'autre. Par la suite, les commandes COM nécessaires sont exécutées pour qu'Artemis Modal mène tous les traitements nécessaires sur les signaux jusqu'à obtenir les matrices spectrales décomposées en valeurs singulières (étape 5). La méthode COM GetFunction est ensuite utilisée pour exporter dans l'environnement Matlab le vecteur contenant le spectre fréquentiel de la première valeur singulière, généralement suffisant pour l'identification des modes d'un pont (étape 6). Un script Matlab se charge ensuite de trouver les pics pertinents de ce spectre (étape 7). Cette étape est rendue complexe par la nature bruitée des signaux traités qui empêche l'obtention d'un spectre lisse lorsque la résolution fréquentielle utilisée est raffinée et donc par la présence de petits pics secondaires sans rapport avec les fréquences naturelles de la structure. Une détection des pics principaux est donc effectuée par le script Matlab par un balayage du spectre qui ne détecte que les pics dont l'acuité est suffisante, i.e. ceux qui sont définis par un nombre minimum de pentes positives (montées) immédiatement suivies d'un nombre minimum de pentes négatives (descentes) (par ex., 3 montées suivies de 3 descentes). Le bruit présent sur le spectre entraîne la détection de nombreux pics de cette manière (fig. 5.12a). Le problème est résolu en appliquant la méthode à un spectre décimé (en ne gardant, par ex., que 1 point sur 5). On obtient un spectre lissé, tel que présenté à la figure 5.12b dans lequel un nombre très réduit de pics locaux est détecté lors du balayage du spectre. Cette méthode permet d'extraire un ensemble réduit de fréquences qui contient au moins les fréquences des modes recherchés. La liste de fréquences obtenue est ensuite utilisée dans Artemis Modal pour effectuer sur chacune un calcul EFDD (étape 8) et les résultats d'identification (fréquences, amortissement et déformées) sont alors exportés dans l'environnement Matlab (étape 9). À partir de ce point, le reste des calculs est opéré par Matlab et les serveurs d'interface COM peuvent donc être fermés (étape 10). Puisque plusieurs des fréquences trouvées à l'étape 7 peuvent ne pas correspondre à des modes, il convient d'éliminer les résultats d'identification erronés (étape 11). Pour cela, Matlab importe un fichier de référence qui contient, au format texte, les déformées des modes recherchés issues d'une analyse modale manuelle antérieure. L'ensemble des déformées modales nouvellement identifiées sont alors comparées à celles du fichier de référence en générant la matrice des MAC (Modal Assurance Criterion, indicateur de colinéarité entre deux vecteurs dont la valeur est comprise entre 0 et 1). Cette matrice effectue le calcul d'un MAC pour chaque combinaison possible des nouveaux vecteurs avec ceux de référence, ce qui permet d'identifier quelle nouvelle déformée correspond à quel mode recherché. Les résultats non associés à un vecteur de référence sont donc éliminés des résultats. Le script Matlab se poursuit avec le calcul de la complexité des déformées modales identifiées (étape 12). Ce calcul de complexité ne sera pas détaillé dans cette section et a été défini dans plusieurs rapports liés au projet R732.1 remis au Ministère dans le passé [4, 30, 34]. À cet instant, tous les calculs d'identification



Figure 5.12 – Détection automatique par Matlab des pics du spectre de la première valeur singulière

modale sont terminés. Le code Matlab exporte alors les résultats de façon ordonnée dans un fichier texte temporaire (13), ce qui met fin au programme *analyse_modale_Matlab.exe*. De retour dans l'environnement Python, le fichier est lu et Python archive les résultats en les entrant dans la table *analyses_modales* de la base de données, attribut par attribut (étape 14).

5.5.3 Génération et téléversement d'un nouveau fichier de configuration

La tâche traitée dans ce paragraphe a déjà été évoquée à la section 5.4.6 pour la description de l'onglet *Configurations* du site web. Elle consiste à reformuler automatiquement un fichier de configuration au format .XML pouvant être lu par le système d'acquisition, tel que présenté à la section 4.5 et à l'annexe D, à partir des données fournies par l'utilisateur dans les boîtes de texte de l'onglet *Configurations* du site web. La tâche suivante consiste à téléverser ce nouveau fichier de configuration sur le CompactRIO via FTP. Cette liste de tâches est uniquement exécutée à l'initiative d'un utilisateur ayant apporté des changements à partir de l'onglet *Configurations* du site web et lorsque celui-ci clique sur le bouton *Mise à jour du système*. Comme indiqué à la section 5.4.6, cette liste de tâches n'est pas encore fonctionnelle pour le prototype développé.

Il est aussi prévu, dans un développement ultérieur du prototype, que le fichier *.cfg utilisé en entrée par Artemis Modal pour l'exécution des analyses modales automatiques (section 5.5.2) soit généré automatiquement lors de la tâche de reconfiguration du système, puisque toutes les informations nécessaires sont en réalité déjà contenues dans la base de données.

Chapitre

6

Conclusion

Le présent document constitue le rapport final du volet 3 du projet de recherche R732.1 mené par l'Université de Sherbrooke (UdeS) pour le ministère des Transports du Québec (MTQ) portant sur la quantification de l'endommagement minimal détectable par des mesures de vibrations ambiantes sur un pont routier en service. Cet volet portait sur le déploiement sur un pont réel, le pont de la Rivière-aux-Mulets, d'un système de télésurveillance des propriétés modales de la structure. En raison de la planification du remplacement prochain du pont par le Ministère, il n'a pas été possible de concrétiser le déploiement du système in situ. Cependant, l'essentiel du travail relié à l'objectif 3 ne réside pas tant dans l'installation effective du système de télésurveillance que dans son développement. Ce travail de développement a été mené à bien par l'université de Sherbrooke et a été présenté dans ce rapport.

Étant donné l'enjeu de maintien en service du vaste réseau routier québécois combiné à son vieillissement manifeste, les requis en inspection et maintenance nécessaire induisent une charge de travail importante au MTQ. Ce travail prend la forme de déplacements in-situ, de collecte de données, de post-traitement et d'analyse. La prise de décision suit après tous ces efforts et les délais occasionnés par ce mode opératoire peuvent s'avérer problématique. Dans ce contexte, l'utilisation d'une technique expérimentale appliquée à distance via un système de télésurveillance devient un outil pertinent pour assurer une meilleure gestion du parc des ouvrages d'art. C'est ce qui est proposé dans le projet de recherche R732.1 par l'analyse de l'évolution des propriétés modales de la structure. Le suivi de ces évolutions est déjà effectué par le Ministère, mais d'une manière qui ne permet pas son utilisation pour les méthodes utilisant un recalage de modèles numériques par éléments finis tel que proposé dans le projet R732.1. Le prototype développé et décrit dans le présent rapport répond essentiellement à ces problématique en permettant le suivi en continu et en mesurant les propriétés dynamiques aussi souvent que nécessaire.

Le projet a mené au développement d'un prototype de système de télésurveillance pour le suivi en continu des propriétés modales d'un pont. Celui-ci est essentiellement formé par un système d'acquisition combiné à un serveur informatique. Le système d'acquisition utilisé pour le développement du prototype est basé sur le matériel CompactRIO de la société National Instruments et son logiciel opérationnel a été developpé en langage LabVIEW. Le serveur informatique, lui, repose sur une simple architecture de type PC (*Personal computer*) sans caractéristiques de performance particulières. Grâce au système proposé et développé dans le cadre de ce travail, le responsable de l'ouvrage peut focaliser son attention sur la prise de décisions, car les données lui parviennent sans déplacement nécessaire en plus d'être post-traitées et analysées automatiquement. Le cRIO et le logiciel développé pour l'enregistrement de données en continu et répondant à diverses situations constituent le premier élément majeur développé et présenté dans ce rapport. Le serveur informatique, équipé d'une base de données, d'un serveur web HTTP et d'une liste de tâches permettant le pilotage, l'alimentation de la base en données et l'analyse modale automatique constitue le second élément majeur développé et présenté.

Des avantages significatifs sont amenés par ce système novateur. En effet, le classement très structuré des données dans une base et l'interface web permettent aux utilisateurs une facilité de consultation hors pair. L'enregistrement des signaux pour des horaires définis par l'utilisateur, à la demande ou bien à la suite d'un événement est une possibilité nouvelle par rapport à la méthode actuelle de mesure des propriétés modales du Ministère. Au fil du temps, le cumul des données permettra une meilleure compréhension des ouvrages. Qui plus est, le système proposé met en œuvre les composantes informatiques nécessaires à l'observation en temps réel des signaux mesurés sur l'ouvrage, à la consultation à volonté de l'historique et de l'évolution des propriétés de l'ouvrage, et à la reconfiguration à distance des composantes numériques du système d'acquisition. Le système est également en mesure d'émettre des alertes advenant un événement générant des signaux dépassant des seuils préétablis.

L'architecture distribuée adoptée pour le développement facilitera l'accueil ultérieur d'autres ponts dans ce qui pourra devenir un réseau de télésurveillance. Cela limitera essentiellement les coûts induit par la connexion du nouveau matériel sur un autre pont. L'usage de logiciels commerciaux déjà détenus par le Ministère (Windows, Artemis Modal) permet une prise en main simplifiée du système tout en évitant les surcharges pour l'achat de licences. D'autre part, l'usage maximum de logiciels libres (APACHE, MySQL, Python, Matlab Runtime, etc.) permet la réduction des coûts en plus de bénéficier d'une communauté de développeurs active, très pédagogue et à l'avant-garde de l'industrie.

Il est important de comprendre que le système développé par l'université de Sherbrooke est un prototype et non un produit commercial final. En tant que tel, comme tout logiciel ou suite de logiciels nouvellement développés, il nécessite des améliorations fonctionnelles et un débogage qui ne pourront être faits qu'à l'usage. Il est important de mentionner aussi que le mandat initial des fonctionnalités décrites en introduction de ce rapport était très ambitieux en regard d'un travail parti de rien. Il convient donc d'accepter que les fonctionnalités primaires essentielles, qui répondent aux exigences définies au contrat de recherche R732.1, ont toutes été programmées et sont fonctionnelles. D'autres, que les auteurs ont choisi de s'imposer en début de projet pour rendre le système plus efficace et intéressant, nécessitent d'être finalisées pour être pleinement opérationnelles ou pourraient être améliorées pour les rendre plus ergonomiques. D'autres fonctionnalités pratiques, encore optionnelles, n'ont pas pu être abordées et sont des améliorations en devenir. En reprenant la liste des fonctionnalités du cahier des charges défini en introduction, voici clairement l'avancement du prototype en regard des fonctionnalités initialement définies :

- (1) Enregistrement flottant continu des données brutes des accéléromètres déployés : **complété et fonctionnel**.
- (2) Sauvegarde des données temporelles des capteurs :
 - (a) à la demande : fonctionnel, amélioration possible.

Une amélioration, déjà prévue à l'interface web à l'onglet *Vibrations ambiantes* reste à programmer au niveau de l'acquisiteur, soit la possibilité de définir la plage de temps que l'utilisateur souhaite enregistrer avant et après sa requête. Actuellement, le système enregistre la durée complète de la file de données accumulées définie à l'onglet *Configurations*.

(b) automatiquement (dates et heures programmables) : fonctionnel, amélioration possible. Une amélioration ergonomique devra être faite à l'interface web à l'onglet *Configurations* pour rendre la définition des plages horaires d'enregistrement plus intuitive. Dans ce cas, elles devront être traduites en code CRON lors de la génération automatique du fichier de configuration de l'acquisiteur à l'appui sur le bouton *Mise à jour du système*.

- (c) au déclenchement d'un événement (dépassement de seuil programmable sur un ou plusieurs capteurs) : fonctionnel, amélioration possible.
 La fonction réagit actuellement à chaque dépassement sur n'importe quel capteur, ce qui peut générer une quantité déraisonnable d'enregistrements et d'alertes. Il faudra programmer un critère de déclenchement plus sévère en exigeant que le dépassement ait lieu sur plusieurs capteurs simulanément.
- (3) Envoi des données au serveur informatique : complété et fonctionnel.
- (4) Émission d'alertes par courriel en cas d'événement : complété et fonctionnel.
- (5) Écriture d'un registre des opérations : fonctionnel, amélioration possible. Le registre s'écrit de façon parfaitement automatique sur le CompactRIO, tel que l'exige la fonction, mais il n'y a pas encore de possibilité de consulter le registre à partir de l'interface web (il faut récupérer le fichier texte sur le CompactRIO), ni d'y consigner des commentaires personnels. C'est le rôle de l'onglet *Journal de bord* qui n'a pas été développé.
- (6) Archivage des données brutes temporelles des accéléromètres dans une base de données : **complété et fonctionnel**.
- (7) Analyse modale automatique par méthode FDD et archivage des résultats dans la base de données : **partiellement complété et non fonctionnel**.
 - La méthodologie et l'algorithme pour effectuer cette fonction ont été élaborés. L'algorithme Matlab permettant d'automatiser l'extraction des modes à partir de la méthode EFDD est complété et a été validé. La routine d'ajout des résultats de l'analyse sur la base de données dans la table *analyses_modales* existe déjà. La routine de pilotage d'Artemis Modal à partir de Matlab via une interface COM est partiellement complétée, car certaines méthodes COM d'Artemis Modal restent à trouver. En raison de ce dernier point, l'analyse modale automatique n'est pas encore fonctionelle et les auteurs recommandent de poursuivre le développement du prototype pour compléter cette fonctionnalité importante pour le gestionnaire d'ouvrage. Notons que cette fonctionalité ne figurait pas au rang des exigences minimales du système pour le contrat R732.1.
- (8) Envoi d'alertes en rapport à des évolutions anormales des résultats d'analyse modale : **non développé**.

Cette fonctionnalité très intéressante du point de vue du gestionnaire d'ouvrage n'a pas encore été abordée, mais ne figurait pas au rang des exigences minimales du système.

- (9) Interface ergonomique pour les utilisateurs pour la configuration des paramètres ajustables et la présentation de la base de données, soit principalement :
 - (a) affichage en temps réel du flux de données des capteurs : complété et fonctionnel.
 - (b) enregistrement de données à la demande (avec ou sans analyse modale) : **partiellement** fonctionnel.

L'enregistrement de données à la demande est fonctionnel avec amélioration possible, tel qu'expliqué précédemment. L'option de demander ou non l'analyse modale automatique suite à l'enregistrement n'est pas développée. Pour le moment le prototype réagit de telle sorte qu'une analyse modale est effectuée systématiquement après la réception de nouveaux fichiers de données. L'option sera facilement programmable en modifiant le nom des fichiers de données et en adaptant la liste de tâches de transfert des données.

- (c) consultation de la base des enregistrements : complété et fonctionnel.
- (d) affichage des résultats des analyses modales et de l'évolution des propriétés dans le temps : **complété et fonctionnel**.
- (e) reconfiguration des options du système de mesure : partiellement fonctionnel. Côté système d'acquisition, la machine d'état est déjà totalement fonctionnelle pour la reconfiguration lorsqu'elle reçoit le message Websocket émis par le navigateur web lors de l'appui sur le bouton Mise à jour du système. En revanche, pour fonctionner l'utilisateur doit téléverser au préalable le nouveau fichier de configuration (préparé manuellement) sur le CompactRIO. L'objectif sera complété en dotant le script de l'onglet Configurations du site web d'une routine d'ajustement des paramètres contenues dans la table configurations de la base de données en fonction des modifications apportées par un utilisateur ou un super-utilisateur. Le script devra aussi générer automatiquement le nouveau fichier de configuration au format .XML. Ensuite, la liste de tâches Python devra téléverser sur le système d'acquisition le nouveau fichier de configuration.

Outre certaines des fonctionnalités précédentes qui nécessitent encore soit des améliorations, soit d'être programmées, plusieurs autres fonctionalités pourraient être développées dans le futur, qui n'étaient pas spécifiées au cahier des charges initial : possibilité d'afficher dans l'onglet *Vibrations ambiantes* la trace temporelle d'enregistrements archivés ; possibilité d'afficher dans l'onglet *Configurations* les paramètres d'une ancienne configuration ; archiver et mettre un lien vers les fichiers de configuration d'Artemis Modal correspondant (*.cfg) ; génération automatique du fichier *.cfg ; calcul automatique, archivage et consultation de statistiques sur les signaux de vibrations ambiantes du pont, . . . Ces idées ne sont évidemment qu'un aperçu des optimisations possibles et qu'un utilisateur pourra demander au fil du temps. C'est pourquoi les auteurs de ce rapport recommandent fortement de poursuivre le développement du prototype fourni au delà du contrat de recherche R732.1. L'éventail de possibilités additionnelles abordé ici constitue l'un des avantages principaux du système développé dans un projet de recherche plutôt que par un fournisseur commercial et développé à partir de langages très versatiles, comme LabVIEW.

Lorsque les derniers scripts seront complets, notamment celui d'analyse modale automatique, il sera possible de passer à un déploiement d'essai en laboratoire, puis in situ sur une structure. À ce moment, des cas-tests seront imposés au système afin de garantir le bon fonctionnement global. Déjà, les objectifs complétés fonctionnent sans problème avec des capteurs ou des jeux de données fictives pour fin de développement. Le déploiement test ira plus loin en corroborant la validité générale du système. Cette étape d'essai et débogage est nécessaire à tout système avant une utilisation in situ sur un pont.

Par la suite, dans l'idée du projet de recherche R732.1, l'étape suivante sera celle de la détection automatique de l'endommagement permise par les résultats du système de télésurveillance. Celle-ci ira plus loin en intégrant des étapes d'analyses évoluées additionnelles, où un modèle par éléments finis sera mis à jour avec les propriétés dynamiques issues du prototype développé dans le but d'évaluer de possibles changements structuraux pouvant être corrélés à de l'endomagement. ANNEXE

A

Fiche technique : NI cRIO-9074



(866) 531-6285 orders@ni.com

Requirements and Compatibility | Ordering Information | Detailed Specifications | Pinouts/Front Panel Connections ser manuals and dimensional drawings, visit the product page resources tab on ni.com

Last Revised: 2014-11-06 07:14:18.0

CompactRIO Integrated Systems with Real-Time Controller and Reconfigurable Chassis NI cRIO-907x





- Integrated CompactRIO systems with a reconfigurable FPGA chassis and embedded
 Up to 400 MHz real-time processor real-time controller
- · Lower-cost systems for high-volume OEM applications
- Up to 2M gate reconfigurable FPGA
- 4 or 8 slots for C Series I/O modules

- Up to two 10/100BASE-TX Ethernet ports with built-in FTP/HTTP servers and LabVIEW remote panel Web server
- Up to 256 MB DRAM memory, 512 MB of nonvolatile storage
- RS232 serial port and available USB port for peripheral devices

Overview

NI cRIO-907x integrated systems combine an industrial real-time controller and reconfigurable field-programmable gate array (FPGA) chassis for industrial machine control and monitoring applications. The NI cRIO-9074 integrated system features an industrial 400 MHz real-time processor and an eight-slot chassis with an embedded, reconfigurable 2M gate FPGA chip. The new NI cRIO-9076 integrated system contains a 400 MHz real-time processor, a four-slot chassis with an embedded, reconfigurable LX45 FPGA chip, and a high-speed USB port. Both systems feature built-in nonvolatile memory and a fault tolerant file system. The new four-slot NI cRIO-9075 and NI cRIO-9076 systems provide a cost-optimized solution for high volume deployments and OEM applications.

Back to Top

Requirements and Compatibility

OS Information

VxWorks

Driver Information

NI-RIO

Software Compatibility

- LabVIEW
- LabVIEW FPGA Module
- LabVIEW Professional Development System
- LabVIEW Real-Time Module

Back to Top

Comparison Tables

Product	Module Slots	Processor Speed (MHz)	FPGA	DRAM (MB)	Internal Nonvolatile Storage (MB)	10/100BASE-TX Ethernet Port	RS232 Serial Port	Power Supply Input Range	USB Port
NI cRIO-9072	8	266	Spartan-3 1M	64	128	yes	yes	19 to 30 VDC	no
NI cRIO-9073	8	266	Spartan-3 2M	64	128	yes	yes	19 to 30 VDC	no
NI cRIO-9074	8	400	Spartan-3 2M	128	256	yes (Dual)	yes	19 to 30 VDC	no
NI cRIO-9075	4	400	Spartan-6 LX25	128	256	yes	yes	9 to 30 VDC	no

Product	Module Slots	Processor Speed (MHz)	FPGA	DRAM (MB)	Internal Nonvolatile Storage (MB)	10/100BASE-TX Ethernet Port	RS232 Serial Port	Power Supply Input Range	USB Port
NI cRIO-9076	4	400	Spartan-6 LX45	256	512	yes	yes	9 to 30 VDC	yes
								E	Back to Top

Application and Technology

System Configuration

These NI CompactRIO real-time controllers combine a four- or eight-slot reconfigurable chassis into an integrated system. The user-defined FPGA circuitry in the chassis controls each I/O module and passes data to the controller through a local PCI bus using built-in communication functions.

Product	FPGA	Logic Cells	Multipliers	RAM (Kb)
NI cRIO-9073	Spartan-3 2M	46080	40	720
NI cRIO-9074	Spartan-3 2M	46080	40	720
NI cRIO-9075	Spartan-6 LX25	24051	38	936
NI cRIO-9076	Spartan-6 LX45	43661	58	2088

FPGA Resource Comparison

These systems also accept up to eight NI C Series I/O modules. A variety of I/O modules are available including voltage, current, thermocouple, RTD, accelerometer, and strain gage inputs; up to ±60 V simultaneous sampling analog I/O; 12, 24, and 48 V industrial digital I/O; 5 V/TTL digital I/O; counter/timers; pulse generation; and high voltage/current relays.

The 10/100 Mbits/s Ethernet port allows for programmatic communication over the network and the cRIO-9074 features dual Ethernet ports, which allows for the use of one port for network communication to a host PC or enterprise system and the other port for expansion I/O (easily connect another CompactRIO system or another Ethernet-based device for additional I/O). The new cRIO-9076 also features a USB 2.0 port for data storage and connection to peripheral devices.

NI CompactRIOs have the ability to by synchronized with an SNTP time server on a network and the cRIO-9072, cRIO-9073, and cRIO-9074 also feature a built-in backup battery to maintain operation for the Real-Time Clock when external power is removed. The cRIO-9075 and cRIO-9076 do not contain a backup battery for the Real-Time Clock.

Embedded Software

You can synchronize embedded code execution to an FPGA-generated interrupt request (IRQ) or an internal millisecond real-time clock source. The LabVIEW Real-Time ETS OS provides reliability and simplifies the development of complete embedded applications that include time-critical control and acquisition loops in addition to lower-priority loops for postprocessing, data logging, and Ethernet/serial communication. Built-in elemental I/O functions such as the FPGA Read/Write function provide a communication interface to the highly optimized reconfigurable FPGA circuitry. Data values are read from the FPGA in integer format and are then converted to scaled engineering units in the controller.



CompactRIO Software Architecture

Note: NI Scan Engine is not supported on the cRIO-9075 and cRIO-9076.

Built-In Servers

In addition to programmatic communication via TCP/IP, UDP, Modbus/TCP, IrDA, and serial protocols, the CompactRIO controllers include built-in servers for Virtual Instrument Software Architecture (VISA), HTTP, and FTP. The VISA server provides remote download and communication access to the reconfigurable I/O (RIO) FPGA over Ethernet. The HTTP server provides a Web browser user interface to HTML pages, files, and the user interface of embedded LabVIEW applications through a Web browser plug-in. The FTP server provides access to logged data or configuration files.

Ordering Information

For a complete list of accessories, visit the product page on ni.com.

Products	Part Number	Recommended Accessories	Part Number
NI cRIO-9072			
cRIO-9072 8-Slot Integrated 266 MHz Real-Time Ctrlr, 1M Gate FPGA Requires:	779998-01	Connector Block: Screw Terminal - NI 9978 4-pos screw terminal power supply plugs (quantity 5)	196938-01
NI cRIO-9073			
cRIO-9073 8-Slot Integrated 266 MHz Real-Time Ctrlr, 2M Gate FPGA Requires:	780471-01	Connector Block: Shielded - NI PS-15 Power Supply, 24 VDC, 5 A, 100-120/200-240 VAC Input **Also Available: Screw Terminal	781093-01
		Connector Block: Screw Terminal - NI 9979 Strain relief kit for 4-pos power connector	196939-01
NI cRIO-9076			
cRIO-9076 4-Slot Integrated 400 MHz Real-Time Ctrlr, LX45 FPGA Requires: 1 Connectivity Accessory	781716-01	Connectivity Accessory: Shielded - NI PS-15 Power Supply, 24 VDC, 5 A, 100-120/200-240 VAC Input	781093-01
NI cRIO-9075			
cRIO-9075 4-Slot Integrated 400 MHz Real-Time Ctrlr, LX25 FPGA Requires: 1 Connectivity Accessory	781715-01	Connectivity Accessory: Shielded - NI PS-15 Power Supply, 24 VDC, 5 A, 100-120/200-240 VAC Input	781093-01
NI cRIO-9074			
cRIO-9074 8-Slot Integrated 400 MHz Real-Time Ctrir, 2M Gate FPGA Requires: 1 Connectivity Accessory	779999-01	Connectivity Accessory: Shielded - NI PS-15 Power Supply, 24 VDC, 5 A, 100-120/200-240 VAC Input	781093-01

Back to Top

Software Recommendations

LabVIEW Professional Development System for Windows

Anders Frank	2000 (Marco 1)	80
	00- 30- 30-	20- 70-
Ů.". Ů."	1013- 201- 1012- 102-	
	003- 404- 203- 203-	

NI LabVIEW Real-Time Module



- Advanced software tools for large project development
- Automatic code generation using DAQ Assistant and Instrument I/O Assistant
- Tight integration with a wide range of hardware
- Advanced measurement analysis and digital signal processing
- Open connectivity with DLLs, ActiveX, and .NET objects
- Capability to build DLLs, executables, and MSI installers

- Design deterministic real-time applications

Take advantage of built-in PID control, signal processing, and analysis functions
Automatically take advantage of multicore CPUs or set processor affinity manually
Includes real-time OS, development and debugging support, and board support
Purchase individually or as part of a

with LabVIEW graphical programming Download to dedicated NI or third-party hardware for reliable execution and a wide

selection of I/O

LabVIEW suite

NI LabVIEW FPGA Module





- Design FPGA applications for NI reconfigurable I/O (RIO) hardware targets
- Program with the same graphical environment used for desktop and real-time applications
- Execute control algorithms with loop rates up to 300 MHz
- Implement custom timing and triggering logic, digital protocols, and DSP algorithms
- Incorporate existing HDL code and third-party IP including Xilinx CORE Generator functions
- Included in the LabVIEW Embedded Control and Monitoring Suite

3/8

Support and Services

System Assurance Programs

NI system assurance programs are designed to make it even easier for you to own an NI system. These programs include configuration and deployment services for your NI PXI, CompactRIO, or Compact FieldPoint system. The NI Basic System Assurance Program provides a simple integration test and ensures that your system is delivered completely assembled in one box. When you configure your system with the NI Standard System Assurance Program, you can select from available NI system driver sets and application development environments to create customized, reorderable software configurations. Your system arrives fully assembled and tested in one box with your software preinstalled. When you order your system with the standard program, you also receive system-specific documentation including a bill of materials, an integration test report, a recommended maintenance plan, and frequently asked question documents. Finally, the standard program reduces the total cost of owning an NI system by providing three years of warranty coverage and calibration service. Use the online product advisors at ni.com/advisor to find a system assurance program to meet your needs.

Calibration

NI measurement hardware is calibrated to ensure measurement accuracy and verify that the device meets its published specifications. To ensure the ongoing accuracy of your measurement hardware, NI offers basic or detailed recalibration service that provides ongoing ISO 9001 audit compliance and confidence in your measurements. To learn more about NI calibration services or to locate a qualified service center near you, contact your local sales office or visit ni.com/calibration.

Technical Support

Get answers to your technical questions using the following National Instruments resources.

- Support Visit ni.com/support to access the NI KnowledgeBase, example programs, and tutorials or to contact our applications engineers who are located in NI sales offices around the world and speak the local language.
- Discussion Forums Visit forums.ni.com for a diverse set of discussion boards on topics you care about.
- Online Community Visit community.ni.com to find, contribute, or collaborate on customer-contributed technical content with users like you.

Repair

While you may never need your hardware repaired, NI understands that unexpected events may lead to necessary repairs. NI offers repair services performed by highly trained technicians who quickly return your device with the guarantee that it will perform to factory specifications. For more information, visit ni.com/repair.

Training and Certifications

The NI training and certification program delivers the fastest, most certain route to increased proficiency and productivity using NI software and hardware. Training builds the skills to more efficiently develop robust, maintainable applications, while certification validates your knowledge and ability.

- Classroom training in cities worldwide the most comprehensive hands-on training taught by engineers.
- On-site training at your facility an excellent option to train multiple employees at the same time.
- Online instructor-led training lower-cost, remote training if classroom or on-site courses are not possible.
- · Course kits lowest-cost, self-paced training that you can use as reference guides.
- Training memberships and training credits to buy now and schedule training later.

Visit ni.com/training for more information.

Extended Warranty

NI offers options for extending the standard product warranty to meet the life-cycle requirements of your project. In addition, because NI understands that your requirements may change, the extended warranty is flexible in length and easily renewed. For more information, visit ni.com/warranty.

OEM

NI offers design-in consulting and product integration assistance if you need NI products for OEM applications. For information about special pricing and services for OEM customers, visit ni.com/oem.

Alliance

Our Professional Services Team is comprised of NI applications engineers, NI Consulting Services, and a worldwide National Instruments Alliance Partner program of more than 700 independent consultants and integrators. Services range from start-up assistance to turnkey system integration. Visit ni.com/alliance.

Detailed Specifications

•					
The following specifications are typical for the – 20 to 55 °C operating temperature range unless otherwise noted.					
Network					
Network interface	10BaseT and 100BaseTX Ethernet				
Compatibility	IEEE 802.3				
Communication rates	10 Mbps, 100 Mbps, auto-negotiated				
Maximum cabling distance	100 m/segment				
RS-232 Serial Port					

Maximum baud rate

115,200 bps

	5 6 7 8
Data bits	1.0
Stop bits	I, Z
Parity	Udd, Even, Mark, Space
Flow control	RIS/CIS, XON/XOFF, DIR/DSR
SMB Connector (cRIO-9074 Only)	
Output Characteristics	
Minimum high-level output voltage	
With –100 µA output current	2.9 V
With –16 mA output current	2.4 V
With -24 mA output current	2.3 V
Maximum low-level output voltage	
With 100 µA output current	0.10 V
With 16 mA output current	0.40 V
With 24 mA output current	0.55 V
Driver type	CMOS
Maximum sink/source current	±24 mA
Maximum 3-state output leakage current	±5 μΑ
Input Characteristics	
Minimum input voltage	0 V
Minimum low-level input voltage	0.94 V
Maximum high-level input voltage	2.43 V
Maximum input voltage	5.5 V
Typical input capacitance	2.5 pF
Typical resistive strapping	1 kΩ to 3.3 V
Memory	
cRIO-9072, cRIO-9073	
Nonvolatile	128 MB
System memory	64 MB
cRIO-9074	
Nonvolatile	256 MB
System memory	128 MB
Reconfigurable FPGA	
cRIO-9072	
Number of logic cells	17,280
Available embedded RAM	432 kbits
cRIO-9073, cRIO-9074	
Number of logic cells	46,080
Available embedded RAM	720 kbits
Internal Real-Time Clock	
Accuracy	200 ppm; 35 ppm at 25 °C
Power Requirements	
Caution You must use a National Electric Code (NEC) UL Listed Class 2 power supply w	48 W 24 VDC
Recommended power supply	20 W maximum
Power consumption	

Power supply input range	19 to 30 V
Physical Characteristics	
If you need to clean the controller, wipe it with a dry towel.	
Screw-terminal wiring	0.5 to 2.5 mm 2 (24 to 12 AWG) copper conductor wire with 10 mm (0.39 in.) of insulation stripped from the end
Torque for screw terminals	0.5 to 0.6 N · m (4.4 to 5.3 lb · in.)
Weight	929 g (32.7 oz)
Safety Voltages	
Connect only voltages that are within these limits	

35 V max. Measurement Category I

V terminal to C terminal

Measurement Category I is for measurements performed on circuits not directly connected to the electrical distribution system referred to as MAINS voltage. MAINS is a hazardous live electrical supply system that powers equipment. This category is for measurements of voltages from specially protected secondary circuits. Such voltage measurements include signal levels, special equipment, limited-energy parts of equipment, circuits powered by regulated low-voltage sources, and electronics.

Caution Do not connect the system to signals or use for measurements within Measurement Categories II, III, or IV.

Safety Standards

This product is designed to meet the requirements of the following standards of safety for electrical equipment for measurement, control, and laboratory use:

- IEC 61010-1, EN 61010-1
- UL 61010-1, CSA 61010-1

Note For UL and other safety certifications, refer to the product label or the Online Product Certification section.

Electromagnetic Compatibility

This product meets the requirements of the following EMC standards for electrical equipment for measurement, control, and laboratory use:

- EN 61326 (IEC 61326): Class A emissions; Industrial Immunity
- EN 55011 (CISPR 11): Group 1, Class A emissions
- AS/NZS CISPR 11: Group 1, Class A emissions
- FCC 47 CFR Part 15B: Class A emissions
- ICES-001: Class A emissions

Note For the standards applied to assess the EMC of this product, refer to the Online Product Certification section.

Note For EMC compliance, operate this product according to the documentation.

CE Compliance

This product meets the essential requirements of applicable European Directives, as amended for CE marking, as follows:

- 2006/95/EC; Low-Voltage Directive (safety)
- 2004/108/EC; Electromagnetic Compatibility Directive (EMC)

Online Product Certification

Refer to the product Declaration of Conformity (DoC) for additional regulatory compliance information. To obtain product certifications and the DoC for this product, visit ni.com/certification, search by module number or product line, and click the appropriate link in the Certification column.

Environmental Management

National Instruments is committed to designing and manufacturing products in an environmentally responsible manner. NI recognizes that eliminating certain hazardous substances from our products is beneficial not only to the environment but also to NI customers.

For additional environmental information, refer to the *NI and the Environment* Web page at ni.com/environment. This page contains the environmental regulations and directives with which NI complies, as well as other environmental information not included in this document.

Waste Electrical and Electronic Equipment (WEEE)



EU Customers At the end of the product life cycle, all products *must* be sent to a WEEE recycling center. For more information about WEEE recycling centers, National Instruments WEEE initiatives, and compliance with WEEE Directive 2002/96/EC on Waste Electrical and Electronic Equipment, visit ni.com/environment/weee.htm.

Battery Replacement and Disposal

This device contains a long-life coin cell battery. If you need to replace it, use the Return Material Authorization (RMA) process or contact an authorized National Instruments sevice representative.

After replacement, recycle the old battery. For additional information, visit ni.com/environment.

Hazardous Locations

U.S. (UL)

Canada (C-UL)

Class I, Division 2, Groups A, B, C, D, T4; Class I, Zone 2, AEx nL IIC T4

Class I, Division 2, Groups A, B, C, D, T4; Class I, Zone 2, Ex nL IIC T4

Environmental

The cRIO-9072/3/4 is intended for indoor use only, but it may be used outdoors if mounted in a suitably rated enclosure.					
Operating temperature (IEC 60068-2-1, IEC 60068-2-2)	– 20 to 55 °C				
Note To meet this operating temperature range, follow the guidelines in the installation instructions for your CompactRIO system.					
Storage temperature (IEC 60068-2-1, IEC 60068-2-2)	– 40 to 85 °C				
Ingress protection	IP 40				
Operating humidity (IEC 60068-2-56)	10 to 90% RH, noncondensing				
Storage humidity (IEC 60068-2-56)	5 to 95% RH, noncondensing				
Maximum altitude	2,000 m				
Pollution Degree (IEC 60664)	2				
Shock and Vibration					
To meet these specifications, you must panel mount the CompactRIO system and affix ferrules to the ends of the power terminal wires.					
Operating shock (IEC 60068-2-27)	30 g, 11 ms half sine 50 g, 3 ms half sine, 18 shocks at 6 orientations				
Operating vibration, random (IEC 60068-2-64)	5 g _{rms} , 10 to 500 Hz				
Operating vibration, sinusoidal (IEC 60068-2-6)	5 g, 10 to 500 Hz				

Cabling

The following table shows the standard Ethernet cable wiring connections for both normal and crossover cables.

	Ethernet Cable Wiring Connections					
Pin	Connector 1	Connector 2 (Normal)	Connector 2 (Crossover)			
1	white/orange	white/orange	white/green			
2	orange	orange	green			
3	white/green	white/green	white/orange			
4	blue	blue	blue			
5	white/blue	white/blue	white/blue			
6	green	green	orange			
7	white/brown	white/brown	white/brown			
8	brown	brown	brown			

Back to Top

Pinouts/Front Panel Connections



Back to Top

©2010 National Instruments. All rights reserved. CompactRIO, FieldPoint, LabVIEW, National Instruments, National Instruments Alliance Partner, NI, and ni.com are trademarks of National Instruments. Other product and company names listed are trademarks or trade names of their respective companies. A National Instruments Alliance Partner is a business entity independent from National Instruments and has no agency, partnership, or joint-venture relationship with National Instruments.

My Profile | RSS | Privacy | Legal | Contact NI © 2014 National Instruments Corporation. All rights reserved.

ANNEXE

В

Fiche technique : NI-9239



Technical Sales

(866) 531-6285 orders@ni.com

Ordering Information | Detailed Specifications | Pinouts/Front Panel Connections For user manuals and dimensional drawings, visit the product page resources tab on ni.com.

Last Revised: 2014-11-06 07:14:18.0

NI 9229, NI 9239

±10 V or ±60 V, Simultaneous Analog Input, 50 kS/s, 4 Ch Modules



- 4 differential channels, 50 kS/s per channel sample rate
- ± 10 V (NI 9239) or ± 60 V (NI 9229) measurement range, 24-bit resolution
- Screw-terminal or BNC connectivity
 - -40 °C to 70 °C operating, 5 g vibration, 50 g shock

· 250 Vrms ch-ch, CAT II (screw terminal), or 60 VDC ch-ch, CAT I (BNC) isolation

Overview

Antialias filter

The NI 9229 is a 4-channel, 24-bit C Series analog input module for use in any NI CompactDAQ or CompactRIO chassis. The NI 9239 is similar to the NI 9229 in functionality except for the input range. You can find detailed specifications for the NI 9239 and NI 9239 in the same manual for comparison.

With channel-to-channel isolation, your entire system, including the device under test, is protected from harmful voltage spikes up to the isolation rating. In addition to safety, isolation eliminates measurement errors caused by ground loops because the front end of the module is floating. **EMC Performance** To ensure EMC compliance for BNC, you must use a ferrite bead, such as NI part number 782801-01.

Recommended Accessories

-NI 9971 strain relief and operator protection (for screw-terminal variant) -EMI suppression ferrite for NI 9229/9239 BNC (for BNC variant)

Box Contents

- -1 NI 9229 or NI 9239 C Series module
- -1 NI 9229/9239 Operating Instructions and Specifications manual
- -4 NI 9976 two-position screw-terminal connectors (for screw-terminal variant)

Back to Top

Comparison Tables

Product Name	Signal Ranges	Channels	Sample Rate	Simultaneous	Resolution	Isolation	Connectivity
NI 9201	±10 V	8 Single-Ended	500 kS/s/ch	No	12-Bit	250 Vrms Ch-Earth (Screw Terminal), 60 VDC Ch-Earth (D-SUB)	Screw Terminal, 25-Pin D-SUB
NI 9205	±200 mV, ±1 V, ±5 V, ±10 V	32 Single-Ended, 16 Differential	250 kS/s	No	16-Bit	250 Vrms Ch-Earth (Spring Terminal), 60 VDC Ch-Earth (D-SUB))	Spring Terminal, 37-Pin D-SUB
NI 9206	±200 mV, ±1 V, ±5 V, ±10 V	32 Single-Ended, 16 Differential	250 kS/s	No	16-Bit	600 VDC Ch-Earth	Spring Terminal
NI 9215	±10 V	4 Differential	100 kS/s/ch	Yes	16-Bit	250 Vrms Ch-Earth (Screw Terminal), 60 VDC Ch-Earth (BNC)	Screw Terminal, BNC
NI 9220	±10 V	16 Differential	100 kS/s/ch	Yes	16-Bit	250 Vrms Ch-Earth (Spring Terminal), 60 VDC Ch-Earth (D-SUB)	Spring Terminal, 37-Pin D-SUB
NI 9221	±60 V	8 Single-Ended	800 kS/s	No	12-Bit	250 Vrms Ch-Earth (Screw Terminal), 60 VDC Ch-Earth (D-SUB)	Screw Terminal, 25-Pin D-SUB

Product Name	Signal Ranges	Channels	Sample Rate	Simultaneous	Resolution	Isolation	Connectivity
NI 9222	±10 V	4 Differential	500 kS/s/ch	Yes	16-Bit	60 VDC Ch-Ch	Screw Terminal
NI 9223	±10 V	4 Differential	1 MS/s/ch	Yes	16-Bit	60 VDC Ch-Ch	Screw Terminal
NI 9229	±60 V	4 Differential	50 kS/s/ch	Yes	24-Bit	250 Vrms Ch-Ch (Screw Terminal), 60 VDC Ch-Ch (BNC)	Screw Terminal, BNC
NI 9239	±10 V	4 Differential	50 kS/s/ch	Yes	24-Bit	250 Vrms Ch-Ch (Screw Terminal), 60 VDC Ch-Ch (BNC)	Screw Terminal, BNC

Back to Top

Application and Technology

NI C Series Overview



NI C Series modules are engineered to provide high-accuracy measurements to meet the demands of advanced DAQ and control applications. Each module contains measurement-specific signal conditioning to connect to an array of sensors and signals, bank and channel-to-channel isolation options, and support for wide temperature ranges to meet a variety of application and environmental needs all in a single rugged package. You can choose from more than 100 C Series modules for measurement, control, and communication to connect your applications to any sensor on any bus.

Most C Series I/O modules work with both the NI CompactDAQ and NI CompactRIO platforms. The modules are identical, and you can move them from one platform to the other with no modification.

NI CompactRIO Platform



Powered by the NI LabVIEW reconfigurable I/O (RIO) architecture, NI CompactRIO combines an open embedded architecture with small size, extreme ruggedness, and hot-swappable industrial I/O modules. Each system contains an FPGA for custom timing, triggering, and processing with a wide array of modular I/O to meet any embedded application requirement.

Configure Your Complete NI CompactRIO System

NI CompactDAQ Platform



NI CompactDAQ is a portable, rugged data acquisition platform that integrates connectivity and signal conditioning into modular I/O to directly interface with any sensor or signal. Using NI CompactDAQ with LabVIEW, you can easily customize how you acquire, analyze, present, and manage your measurement data. From research to development to validation, NI provides programmable software, high-accuracy measurements, and local technical support to help ensure you meet your exact measurement application requirements.

Configure Your Complete NI CompactDAQ System

Back to Top

Ordering Information

For a complete list of accessories, visit the product page on ni.com.

Products	Part Number	Recommended Accessories	Part Number
		No accessories required.	
		No accessories required.	

Back to Top

Software Recommendations

LabVIEW Professional Development System for Windows



NI LabVIEW Real-Time Module



- Advanced software tools for large project development
 Automatic code generation using DAQ
- Assistant and Instrument I/O Assistant Tight integration with a wide range of
- hardwareAdvanced measurement analysis and digital signal processing
- Open connectivity with DLLs, ActiveX, and .NET objects
- Capability to build DLLs, executables, and MSI installers
- Design deterministic real-time applications with LabVIEW graphical programming
- Download to dedicated NI or third-party hardware for reliable execution and a wide selection of I/O
- Take advantage of built-in PID control, signal processing, and analysis functions
- Automatically take advantage of multicore CPUs or set processor affinity manually
- Includes real-time OS, development and debugging support, and board support
- Purchase individually or as part of a LabVIEW suite

NI LabVIEW FPGA Module





- environment used for desktop and real-time applications
- Execute control algorithms with loop rates up to 300 MHz
- Implement custom timing and triggering logic, digital protocols, and DSP algorithms
- Incorporate existing HDL code and third-party IP including Xilinx CORE Generator functions
- Included in the LabVIEW Embedded Control and Monitoring Suite



Support and Services

System Assurance Programs

NI system assurance programs are designed to make it even easier for you to own an NI system. These programs include configuration and deployment services for your NI PXI, CompactRIO, or Compact FieldPoint system. The NI Basic System Assurance Program provides a simple integration test and ensures that your system is delivered completely assembled in one box. When you configure your system with the NI Standard System Assurance Program, you can select from available NI system driver sets and application development environments to create customized, reorderable software configurations. Your system arrives fully assembled and tested in one box with your software preinstalled. When you order your system with the standard program, you also receive system-specific documentation including a bill of materials, an integration test report, a recommended maintenance plan, and frequently asked question documents. Finally, the standard program reduces the total cost of owning an NI system by providing three years of warranty coverage and calibration service. Use the online product advisors at ni.com/advisor to find a system assurance program to meet your needs.

Calibration

NI measurement hardware is calibrated to ensure measurement accuracy and verify that the device meets its published specifications. To ensure the ongoing accuracy of your measurement hardware, NI offers basic or detailed recalibration service that provides ongoing ISO 9001 audit compliance and confidence in your measurements. To learn more about NI calibration services or to locate a qualified service center near you, contact your local sales office or visit ni.com/calibration.

Technical Support

Get answers to your technical questions using the following National Instruments resources.

- Support Visit ni.com/support to access the NI KnowledgeBase, example programs, and tutorials or to contact our applications engineers who are located in NI sales offices around the world and speak the local language.
- Discussion Forums Visit forums.ni.com for a diverse set of discussion boards on topics you care about.
- Online Community Visit community.ni.com to find, contribute, or collaborate on customer-contributed technical content with users like you.

Repair

While you may never need your hardware repaired, NI understands that unexpected events may lead to necessary repairs. NI offers repair services performed by highly trained technicians who quickly return your device with the guarantee that it will perform to factory specifications. For more information, visit ni.com/repair.

Training and Certifications

The NI training and certification program delivers the fastest, most certain route to increased proficiency and productivity using NI software and hardware. Training builds the skills to more efficiently develop robust, maintainable applications, while certification validates your knowledge and ability.

- Classroom training in cities worldwide the most comprehensive hands-on training taught by engineers.
- On-site training at your facility an excellent option to train multiple employees at the same time.
- · Online instructor-led training lower-cost, remote training if classroom or on-site courses are not possible.
- Course kits lowest-cost, self-paced training that you can use as reference guides.
- Training memberships and training credits to buy now and schedule training later.

Visit ni.com/training for more information.

Extended Warranty

NI offers options for extending the standard product warranty to meet the life-cycle requirements of your project. In addition, because NI understands that your requirements may change, the extended warranty is flexible in length and easily renewed. For more information, visit ni.com/warranty.

OEM

NI offers design-in consulting and product integration assistance if you need NI products for OEM applications. For information about special pricing and services for OEM customers, visit ni.com/oem.

Alliance

Our Professional Services Team is comprised of NI applications engineers, NI Consulting Services, and a worldwide National Instruments Alliance Partner program of more than 700 independent consultants and integrators. Services range from start-up assistance to turnkey system integration. Visit ni.com/alliance.

Back to Top

Detailed Specifications

The following specifications are typical for the range -40 to 70 °C unless otherwise noted. All voltages are relative to the AI- signal on each channel unless otherwise noted. The specifications are the same for the NI 9229 and the NI 9239 unless otherwise noted.

Input Characteristics	
Number of channels	4 analog input channels
ADC resolution	24 bits
Type of ADC	Delta-Sigma (with analog prefiltering)
Sampling mode	Simultaneous
Internal master timebase (f_M)	

4/9

Accuracy	±100 ppm max
Data rate range (f_s) using internal master timebase	
Minimum	1.613 kS/s
Maximum	50 kS/s
Data rate range (\boldsymbol{f}_{s}) using external master timebase	
Minimum	390.625 S/s
Maximum	51.2 kS/s
Data rates ¹	$\frac{f_M + 256}{n}, n = 1, 2,, 31$

Input voltage ranges (AI+ to AI–) ²				
Module	Nominal (V)	Typical (V)	Minimum (V)	
NI 9229	±60	±62.64	±61.5	
NI 9239	±10	±10.52	±10.3	

Overvoltage protection	±100 V
Input coupling	DC
Input impedance (AI+ to AI-)	1 ΜΩ

Input impedance (AI+ to AI-)

Accuracy, NI 9229					
Measurement Conditions	Percent of Reading (Gain Error)	Percent of Range ³ (Offset Error)			
Calibrated max (–40 to 70 °C)	±0.13%	±0.05%			
Calibrated typ (25 °C, ±5 °C)	±0.03%	±0.008%			
Uncalibrated max (-40 to 70 °C)	±1.2%	±0.55%			
Uncalibrated typ (25 °C, ±5 °C)	±0.3%	±0.11%			

Accuracy, NI 9239				
Measurement Conditions	Percent of Reading (Gain Error)	Percent of Range ⁴ (Offset Error)		
Calibrated max (-40 to 70 °C)	±0.13%	±0.06%		
Calibrated typ (25 °C, ±5 °C)	±0.03%	±0.008%		
Uncalibrated max (-40 to 70 °C)	±1.4	±0.70%		
Uncalibrated typ (25 °C, ±5 °C)	±0.3%	±0.11%		

Input noise	
NI 9229	320 µV _{rms}
NI 9239	70 μV _{rms}
Stability	
Gain drift	±5 ppm/°C
Offset drift	
NI 9229	±150 μV/°C
NI 9239	±26 μV/°C
Post calibration gain match (ch-to-ch, 20 kHz)	0.22 dB max
Crosstalk (1 kHz)	-130 dB
Phase mismatch (ch-to-ch)	
NI 9229	0.045°/kHz max
NI 9239	0.075°/kHz max

Phase mismatch (module-to-module, max)
NI 9229	0.045°/kHz max +360° $\cdot f_{in}/f_M$
NI 9239	0.075°/kHz max +360° $\cdot f_{irr}/f_M$
Phase nonlinearity ($f_s = 50 \text{ kS/s}$)	0.11° max
Input delay	
NI 9229	38.4/ <i>f</i> _s + 2.6 μs
NI 9239	38.4/ <i>f_s</i> + 3 μs
Passband	
Frequency	0.453 · <i>f</i> _s
Flatness (f _s = 50 kS/s)	±100 mdB max
Stopband	
Frequency	0.547 · <i>f</i> _s
Rejection	100 dB
Alias-free bandwidth	0.453 · <i>f</i> _s
-3 dB prefilter bandwidth ($f_s = 50$ kS/s)	24.56 kHz
CMRR (<i>f_{in}</i> = 60 Hz)	
NI 9229	116 dB
NI 9239	126 dB
SFDR (1 kHz,60 dBFS)	-128 dBFS
Total Harmonic Distortion (THD)	
1 kHz, –1 dBFS	-99 dB
1 kHz, -20 dBFS	–105 dB
MTBF	662,484 hours at 25 °C; Bellcore Issue 2, Method 1, Case 3, Limited Part Stress Method
Note Contact NI for Bellcore MTBF specifications at other temperatures	s or for MIL-HDBK-217F specifications.
Power Requirements	
Power consumption from chassis	
Active mode	740 mW max
Sleep mode	25 μW max
Thermal dissipation (at 70 °C)	
Active mode	760 mW max
Sleep mode	16 mW max
Physical Characteristics	
Screw-terminal wiring	16 to 28 AWG copper conductor wire with 7 mm (0.28 in.) of insulation stripped from the end
Torque for screw terminals	0.22 to 0.25 N · m (1.95 to 2.21 lb · in.)
Ferrules	$0.25 \text{ mm}^2 \text{ to } 0.5 \text{ mm}^2$
Weight	147 g (5.2 oz)
Safety	
If you need to clean the module, wipe it with a dry towel. Safety Voltages	
Connect only voltages that are within the following limits.	
	250 V Meggurement Category II
	2 300 V verified by a 5 s dialectric withstand test
Vitti Statiu	
Continuous	250 V Measurement Category II
Withstand	1,390 V_{ms} , verified by a 5 s dielectric withstand test
	1110 *

Withstand

Maximum voltage (channel-to-earth ground and channel-to-channel)

60 VDC, Measurement Category I

Measurement Category I is for measurements performed on circuits not directly connected to the electrical distribution system referred to as MAINS ⁵ voltage. This category is for measurements of voltages from specially protected secondary circuits. Such voltage measurements include signal levels, special equipment, limited-energy parts of equipment, circuits powered by regulated low-voltage sources, and electronics.

Caution Do not connect to signals or use for measurements within Measurement Categories II, III, or IV.

Measurement Category II is for measurements performed on circuits directly connected to the electrical distribution system. This category refers to local-level electrical distribution, such as that provided by a standard wall outlet (e.g., 115 V for U.S. or 230 V for Europe). Examples of Measurement Category II are measurements performed on household appliances, portable tools, and similar products.

Caution Do not connect to signals or use for measurements within Measurement Categories III or IV.

Safety Standards

⚠

This product is designed to meet the requirements of the following standards of safety for electrical equipment for measurement, control, and laboratory use:

- IEC 61010-1, EN 61010-1
- UL 61010-1, CSA 61010-1

Note For UL and other safety certifications, refer to the product label or the Online Product Certification section.

Hazardous Locations	
U.S. (UL)	Class I, Division 2, Groups A, B, C, D, T4; Class I, Zone 2, AEx nA IIC T4
Canada (C-UL)	Class I, Division 2, Groups A, B, C, D, T4; Class I, Zone 2, Ex nA IIC T4
Europe (DEMKO)	Ex nA IIC T4

Environmental

National Instruments C Series modules are intended for indoor use only but may be used outdoors if installed in a suitable enclosure. Refer to the manual for the chassis you are using for more information about meeting these specifications.

Operating temperature (IEC 60068-2-1, IEC 60068-2-2)	–40 to 70 °C
Storage temperature (IEC 60068-2-1, IEC 60068-2-2)	-40 to 85 °C
Ingress protection	IP 40
Operating humidity (IEC 60068-2-56)	10 to 90% RH, noncondensing
Storage humidity (IEC 60068-2-56)	5 to 95% RH, noncondensing
Maximum altitude	2,000 m
Pollution Degree (IEC 60664)	2

Shock and Vibration

To meet these specifications, you must panel mount the system and either affix ferrules to the ends of the terminal wires or use the NI 9971 backshell kit to protect the connections.

Operating vibration				
Random (IEC 60068-2-64)	5 g _{rms} , 10 to 500 Hz			
Sinusoidal (IEC 60068-2-6)	5 g, 10 to 500 Hz			
Operating shock (IEC 60068-2-27)	30 g, 11 ms half sine, 50 g, 3 ms half sine, 18 shocks at 6 orientations			

Electromagnetic Compatibility

This product is designed to meet the requirements of the following standards of EMC for electrical equipment for measurement, control, and laboratory use:

- EN 61326 EMC requirements; Industrial Immunity
- EN 55011 Emissions; Group 1, Class A
- . CE, C-Tick, ICES, and FCC Part 15 Emissions; Class A

Note For EMC compliance, operate this device with shielded cables.

CE Compliance

This product meets the essential requirements of applicable European Directives, as amended for CE marking, as follows:

- 2006/95/EC; Low-Voltage Directive (safety)
- 2004/108/EC; Electromagnetic Compatibility Directive (EMC)

Note For the standards applied to assess the EMC of this product, refer to the Online Product Certification section.

Online Product Certification

Refer to the product Declaration of Conformity (DoC) for additional regulatory compliance information. To obtain product certifications and the DoC for this product, visit ni.com/certification, search by module number or product line, and click the appropriate link in the Certification column.

Environmental Management

National Instruments is committed to designing and manufacturing products in an environmentally responsible manner. NI recognizes that eliminating certain hazardous substances from our products is beneficial not only to the environment but also to NI customers.

For additional environmental information, refer to the *NI and the Environment* Web page at ni.com/environment. This page contains the environmental regulations and directives with which NI complex, as well as other environmental information not included in this document.

Waste Electrical and Electronic Equipment (WEEE)



EU Customers At the end of their life cycle, all products *must* be sent to a WEEE recycling center. For more information about WEEE recycling centers and National Instruments WEEE initiatives, visit ni.com/environment/weee.htm.

电子信息产品污染控制管理办法 (中国 RoHS)

中国客户 National Instruments 符合中国电子信息产品中限制使用某些有害物质指令 (RoHS)。
 关于 National Instruments 中国 RoHS 合规性信息, 诸登录 ni.com/environment/rohs_china。
 (For Information about China RoHS compliance, go to ni.com/environment/rohs_china.)

Calibration

You can obtain the calibration certificate for this device at ni.com/calibration.

Calibration interval

1 year

¹ The data rate must remain within the appropriate data rate range. Refer to the Understanding NI 9229/9239 Data Rates section of the NI 9229/9239 Operating Instructions and Specifications for more information.

² Refer to the Safety Guidelines section for more information about safe operating voltages.

- ³ Range equals 62.64 V
- ⁴ Range equals 10.52 V

⁵ MAINS is defined as the (hazardous live) electrical supply system to which equipment is designed to be connected for the purpose of powering the equipment. Suitably rated measuring circuits may be connected to the MAINS for measuring purposes.



Back to Top

©2014 National Instruments. All rights reserved. CompactRIO, FieldPoint, LabVIEW, National Instruments, NI, ni.com, and NI CompactDAQ are trademarks of National Instruments. Other product and company names listed are trademarks or trade names of their respective companies. A National Instruments Alliance Partner is a business entity independent from National Instruments and has no agency, partnership, or joint-venture relationship with National Instruments.

My Profile | RSS | Privacy | Legal | Contact NI © 2014 National Instruments Corporation. All rights reserved.

ANNEXE

С

Fiche technique : NI PS-15



(866) 531-6285 orders@ni.com

For user manuals and dimensional drawings, visit the product page resources tab on ni.com.

Last Revised: 2014-11-06 07:14:59.0

Industrial Power Supplies NI PS-14, NI PS-15, NI PS-16, NI PS-17



- 24 VDC power supplies for CompactRIO, NI CompactDAQ, NI Single-Board RIO, NI
 20 or 50 percent power reserves for dynamic loads (NI PS-15 and NI PS-16) Smart Cameras, and NI touch panel computers
- Full output power between -25 °C and +60 °C
- 115 V/230 V autoselect input
- Efficiencies up to more than 90 percent
- Low input rush current

- Toolless spring-clamp terminals for easy field connectivity
- DIN-rail mounting included and panel/side mount accessories available
- -25 °C to 70 °C operating temperature range

Overview

The NI PS-14, PS-15, PS-16, and PS-17 industrial power supplies feature long expected life, generous power reserves, and compact size. The 24 V supplies are ideal for powering any NI CompactRIO, NI Single-Board RIO, NI Smart Camera, NI touch panel computer, NI CompactDAQ, or NI Compact FieldPoint system, as well as heavy accessory loads such as DC motor drives. The DIN-rail system and spring clamp terminals do not require tools and make installation fast and easy. The wide range and autoselect input voltages help you avoid user errors. The wide operating temperature range and extraordinary EMI immunity enables trouble-free operation even under harsh conditions. Depending on the device, NI industrial power supplies guarantee power reserves of 20 to 50 percent. For the PS-15 and PS-16, you can use this extra current continuously for ambient temperatures under 45 °C. With these power reserves, you may not need to oversize for dynamic loads, but you can simply choose a unit that meets the operating requirements. In some cases, you can select a smaller unit to save money and space.

Back to Top

Comparison Tables

Product	Output Voltage (V)	Output Current (A)	Output Power (W)	Input Voltage	Temperature Range (°C)	Size WxHxD (mm)	Weight (g)	Part Number
PS-14	24 to 28	3.3	80	100 to 240 VAC/110 to 375 VDC	-25 to +70	32 x 124 x 102	430	783167-01
PS-15	24 to 28	5	120	90 to 132/180 to 264 VAC	-25 to +70	32 x 124 x 117	500	781093-01
PS-16	24 to 28	10	240	90 to 132/180 to 264 VAC	-25 to +70	60 x 124 x 117	700	781094-01
PS-17	24 to 28	20	480	85 to 276 VAC/110 to 150 VDC	-25 to +70	82 x 124 x 127	1200	781095-01

Back to Top

Application and Technology

Mounting Options

The PS-14/15/16/17 power supplies include clips for DIN-rail mounting. National Instruments provides optional accessories for wall, panel, and side mounting to minimize installation depth. You can couple the included DIN-rail mounting bracket with the side mount kit for a low-profile DIN-rail installation.

Part Number	Product	Installed Width (mm)	Installed Height (mm)	Installed Depth (mm)	Bracket Weight (g)
	Panel Mount Kit for				

199432-01	PS-14/15/16/17	Equal to PS Width	143	PS Depth +7	50
199429-01	Side Mount Kit for PS-14/15	118	145	38 (Panel)/44 (DIN)	140
199430-01	Side Mount Kit for PS-16	118	145	66 (Panel)/72 (DIN)	180
199431-01	Side Mount Kit for PS-17	138	145	88 (Panel)/94 (DIN)	205

Back to Top

Support and Services

System Assurance Programs

NI system assurance programs are designed to make it even easier for you to own an NI system. These programs include configuration and deployment services for your NI PXI, CompactRIO, or Compact FieldPoint system. The NI Basic System Assurance Program provides a simple integration test and ensures that your system is delivered completely assembled in one box. When you configure your system with the NI Standard System Assurance Program, you can select from available NI system driver sets and application development environments to create customized, reorderable software configurations. Your system arrives fully assembled and tested in one box with your software preinstalled. When you order your system with the standard program, you also receive system-specific documentation including a bill of materials, an integration test report, a recommended maintenance plan, and frequently asked question documents. Finally, the standard program reduces the total cost of owning an NI system by providing three years of warranty coverage and calibration service. Use the online product advisors at ni.com/advisor to find a system assurance program to meet your needs.

Calibration

NI measurement hardware is calibrated to ensure measurement accuracy and verify that the device meets its published specifications. To ensure the ongoing accuracy of your measurement hardware, NI offers basic or detailed recalibration service that provides ongoing ISO 9001 audit compliance and confidence in your measurements. To learn more about NI calibration services or to locate a qualified service center near you, contact your local sales office or visit ni.com/calibration.

Technical Support

Get answers to your technical questions using the following National Instruments resources.

- Support Visit ni.com/support to access the NI KnowledgeBase, example programs, and tutorials or to contact our applications engineers who are located in NI sales offices around the world and speak the local language.
- Discussion Forums Visit forums.ni.com for a diverse set of discussion boards on topics you care about.
- Online Community Visit community.ni.com to find, contribute, or collaborate on customer-contributed technical content with users like you.

Repair

While you may never need your hardware repaired, NI understands that unexpected events may lead to necessary repairs. NI offers repair services performed by highly trained technicians who quickly return your device with the guarantee that it will perform to factory specifications. For more information, visit ni.com/repair.

Training and Certifications

The NI training and certification program delivers the fastest, most certain route to increased proficiency and productivity using NI software and hardware. Training builds the skills to more efficiently develop robust, maintainable applications, while certification validates your knowledge and ability.

- Classroom training in cities worldwide the most comprehensive hands-on training taught by engineers.
- On-site training at your facility an excellent option to train multiple employees at the same time.
- Online instructor-led training lower-cost, remote training if classroom or on-site courses are not possible.
- · Course kits lowest-cost, self-paced training that you can use as reference guides.
- Training memberships and training credits to buy now and schedule training later.

Visit ni.com/training for more information.

Extended Warranty

NI offers options for extending the standard product warranty to meet the life-cycle requirements of your project. In addition, because NI understands that your requirements may change, the extended warranty is flexible in length and easily renewed. For more information, visit ni.com/warranty.

OEM

NI offers design-in consulting and product integration assistance if you need NI products for OEM applications. For information about special pricing and services for OEM customers, visit ni.com/oem.

Alliance

Our Professional Services Team is comprised of NI applications engineers, NI Consulting Services, and a worldwide National Instruments Alliance Partner program of more than 700 independent consultants and integrators. Services range from start-up assistance to turnkey system integration. Visit ni.com/alliance.

Back to Top

©2013 National Instruments. All rights reserved. CompactRIO, FieldPoint, National Instruments, NI, ni.com, and NI CompactDAQ are trademarks of National Instruments. Other product and company names listed are trademarks or trade names of their respective companies. A National Instruments Alliance Partner is a business entity independent from National Instruments and has no agency, partnership, or joint-venture relationship with National Instruments.

ANNEXE

Fichier de configuration

Cette annexe présente le script du fichier de configuration utilisé par le VI LabVIEW du compactRio, avec des valeurs typiques de variables.

ī.

1	xml version='1.0' standalone='yes'?	36	
2	<lvdata xmlns=""></lvdata>	37	
3	<version>12.0.1f5</version>	38	<cluster></cluster>
4	<cluster></cluster>	39	<name>surEvenement</name>
5	<name>ConfigDaq</name>	40	<numelts>3</numelts>
6	<numelts>2</numelts>	41	<string></string>
7	<cluster></cluster>	42	<name>preEven</name>
8	<name>Utilisateur</name>	43	<val>5_secondes</val>
9	<numelts>3</numelts>	44	
10	<cluster></cluster>	45	<string></string>
11	<name>surHoraire</name>	46	<name>postEven</name>
12	<numelts>4</numelts>	47	<val>2_minutes</val>
13	<string></string>	48	
14	<name>CRON_1</name>	49	<string></string>
15	<val>15,30 11-13 * *</val>	50	<name>courriels</name>
16		51	<val>alex.loignon@usherbrooke.ca</val>
17	<string></string>	52	
18	<name>CRON_2</name>	53	
19	<val>*/3 14-15 * *</val>	54	
20		55	<cluster></cluster>
21	<string></string>	56	<name>SuperUtilisateur</name>
22	<name>CRON_3</name>	57	<numelts>5</numelts>
23	<val>0 0 1 7</val>	58	<cluster></cluster>
24		59	<name>echantillonnage</name>
25	<string></string>	60	<numelts>2</numelts>
26	<name>memFlot</name>	61	<dbl></dbl>
27	<val>5_secondes</val>	62	<name>reel</name>
28		63	<val>2000</val>
29		64	
30	<cluster></cluster>	65	<dbl></dbl>
31	<name>surUtilisateur</name>	66	<name>effectif</name>
32	<numelts>1</numelts>	67	<val>200</val>
33	<string></string>	68	
34	<name>memFlot</name>	69	
35	<val>3_minutes</val>	70	<cluster></cluster>

71	<name>canal1</name>	111	<val>-0.033978</val>
72	<numelts>5</numelts>	112	
73	<string></string>	113	<dbl></dbl>
74	<name>serieCapteur</name>	114	<name>seuilMax</name>
75	<val>20076</val>	115	<val>0.033978</val>
76		116	
77	<dbl></dbl>	117	
78	<name>sensibilite</name>	118	<cluster></cluster>
79	<val>1000</val>	119	<name>canal3</name>
80		120	<numelts>5</numelts>
81	<string></string>	121	<string></string>
82	<name>unite</name>	122	<name>serieCapteur</name>
83	<val>(mV)/(m.s^-2)</val>	123	<val>ES-U2-01</val>
84		124	
85	<dbl></dbl>	125	<dbl></dbl>
86	<name>seuilMin</name>	126	<name>sensibilite</name>
87	<val>-0.025484</val>	127	<val>1019.3</val>
88		128	
89	<dbl></dbl>	129	<string></string>
90	<name>seuilMax</name>	130	<name>unite</name>
91	<val>0.025484</val>	131	<val>(mV)/(m.s^-2)</val>
92		132	
93		133	<dbl></dbl>
94	<cluster></cluster>	134	<name>seuilMin</name>
95	<name>canal2</name>	135	<val>-0.10193</val>
96	<numelts>5</numelts>	136	
97	<string></string>	137	<dbl></dbl>
98	<name>serieCapteur</name>	138	<name>seuilMax</name>
99	<val>AC-61-H-01</val>	139	<val>0.10193</val>
100		140	
101	<dbl></dbl>	141	
102	<name>sensibilite</name>	142	<cluster></cluster>
103	<val>339.78</val>	143	<name>canal4</name>
104		144	<numelts>5</numelts>
105	<string></string>	145	
106	<name>unite</name>	146	
107	<val>(mV)/(m.s^-2)</val>	147	
108		148	
109	<dbl></dbl>	149	
110	<name>seuilMin</name>		

Bibliographie

- [1] APACHE SOFTWARE FOUNDATION, Apache v2.4.17, Wakefield, MA, 2015. www.apache.org.
- [2] F. BELLIFEMINE, G. CAIRE, A. POGGI et G. RIMASSA, Jade : A white paper. *EXP in search of innovation*, 3(3):6–19, 2003.
- [3] P. A. BLUME, *The LabVIEW style book*. Prentice Hall, Pearson Education, Upper Saddle River, NJ, 2007.
- [4] K. BOULERICE, O. GAURON et P. PAULTRE, Quantification de l'endommagement minimal détectable par des mesures de vibrations dans un pont routier en service - Détection d'endommagement sur des poutres de pont routier en béton précontraint. Rap. tech. CRGP-2017/01, Centre de recherche en génie parasismique et en dynamique des structures, Université de Sherbrooke, Décembre 2017.
- [5] M. CELEBI, Seismic instrumentation of buildings (with emphasis on federal buildings). Rap. tech., 0-7460-68170, United States Geological Survey, Menlo Park, CA, 2002.
- [6] S. DOEBLING, C. FARRAR, M. PRIME et D. SHEVITZ, Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics : A literature review. Rap. tech. LA-13070-MS, Los Alamos National Laboratory, 1996.
- [7] D. EWINS, *Modal Testing : Theory, Practice and Application*. Taylor & Francis, London, 2nd édn, 2000.
- [8] J.-L. FANCHON, Guide des sciences et technologies industrielles. Nathan-Afnor, France, 2007.
- [9] M. FRASER, A. ELGAMAL, J. P. CONTE, S. MASRI, T. FOUNTAIN, A. GUPTA, M. TRIVEDI et M. EL ZARKI, Elements of an integrated health monitoring framework. In Smart Nondestructive Evaluation and Health Monitoring of Structural and Biological Systems II. SPIE -The International Society for Optical Engineering, San Diego, CA, 2003.
- [10] O. GAURON, A. SAIDOU, S. AMBROISE, N. TURCOTTE, C.-P. LAMARCHE et P. PAULTRE, Quantification de l'endommagement minimal détectable par des mesures de vibrations dans un pont routier en service - Proposition d'une instrumentation permanente pour le pont de la Rivière-aux-Mulets. Rap. tech. CRGP-2014/04, Centre de recherche en génie parasismique et en dynamique des structures, Université de Sherbrooke, Mars 2014.
- [11] W. R. HABEL, SHM systems supporting extension of the structures service life. Journal of Civil Structural Health Monitoring, 3(4):225–226, 2013.

- [12] D. HUSTON, Structural sensing, health monitoring, and performance evaluation. CRC Press, Taylor & Francis, Boca Raton, FL, 2010.
- [13] D. INAUDI, Overview of 40 bridge structural health monitoring projects. In 26th Annual International Bridge Conference. Engineers Society of Western Pennsylvania, Pittsburgh, PA, 2009.
- [14] Y. KAYA, C. VENTURA, M. TUREK et S. HUFFMAN, British Columbia Smart Infrastructure Monitoring System. In 7th European Workshop on Structural Health Monitoring. Nantes, France, 2014.
- [15] J. Ko et Y. NI, Technology developments in structural health monitoring of large-scale bridges. Engineering structures, 27(12):1715–1725, 2005.
- [16] K. Y. KOO, N. de BATTISTA et J. M. W. BROWNJOHN, SHM data management system using MySQL database with MATLAB and Web interfaces. In 5th International Conference on Structural Health Monitoring of Intelligent Infrastructure. International Society for Structural Health Monitoring of Intelligent Infrastructure, Winnipeg, Manitoba, Canada, 2011.
- [17] O. LAURSEN, Flot v0.8.3, 2017. www.flotcharts.org.
- [18] H. LI et J. OU, The state of the art in structural health monitoring of cable-stayed bridges. Journal of Civil Structural Health Monitoring, 6(1):1–25, 2016.
- [19] P. LU, Scalable web-based data management system in long term structure health monitoring. Rap. tech., Iowa State University, Ames, IA, 2004.
- [20] MARIADB FOUNDATION, MariaDB v10.1.19, 2015. https://mariadb.org/.
- [21] T. MARWALA, Finite Element Model Updating Using Computational Intelligence Techniques : Applications to Structural Dynamics. Springer, 2010.
- [22] MATHWORKS, Matlab, Natick, MA, 2015. www.mathworks.com.
- [23] MEDIAMONGRELS, Labview websocket API, London, United Kingdom, 2017. www.mediamongrels.com/websockets-library.
- [24] MINISTÈRE DES TRANSPORTS DU QUÉBEC, Bilan de l'état des structures du réseau routier supérieur québécois 2011. Rap. tech., Ministère des Transports du Québec, Québec, QC, Canada, 2012.
- [25] NATIONAL INSTRUMENTS, NI LabVIEW for CompactRIO Developper's Guide Recommended LabVIEW Architectures and Development Practices for Control and Monitoring Applications. Rap. tech., National Instruments, Austin, TX, 2014.
- [26] NATIONAL INSTRUMENTS, Labview, Austin, TX, 2015. www.ni.com.
- [27] NATIONAL INSTRUMENTS, Ni compactrio waveform reference library, Austin, TX, 2015. http://sine.ni.com/nips/cds/view/p/lang/fr/nid/209114.
- [28] ORACLE CORPORATION, Mysql, France, 2015. www.mysql.fr.
- [29] PYTHON SOFTWARE FOUNDATION, Python v2.7.11, Wilmington, Delaware, USA, 2015. www.python.org.

- [30] M. SAIDOU, O. GAURON, C.-P. LAMARCHE et P. PAULTRE, Quantification de l'endommagement minimal détectable par des mesures de vibrations dans un pont routier en service -Détection d'endommagement dans des poutres de pont en béton armé testées en laboratoire - Rapport d'étape. Rap. tech. CRGP-2015/04, Centre de recherche en génie parasismique et en dynamique des structures, Université de Sherbrooke, Août 2015.
- [31] K. SMARSLY, K. H. LAW et D. HARTMANN, Multiagent-Based Collaborative Framework for a Self-Managing Structural Health Monitoring System. *Journal of Computing in Civil Engineering*, 26(1):76–89, 2012.
- [32] H. SOHN, C. R. FARRAR, F. M. HEMEZ, D. D. SHUNK, D. W. STINEMATES, B. R. NADLER et J. J. CZARNECKI, A review of structural health monitoring literature : 1996-2001. Rap. tech. LA-13976-MS, Los Alamos National Laboratory Report, 2004.
- [33] STRUCTURAL VIBRATION SOLUTION, Artemis modal 5.0, Aalborg, Danemark, 2017. http://www.svibs.com/.
- [34] N. TURCOTTE, O. GAURON, C.-P. LAMARCHE et P. PAULTRE, Quantification de l'endommagement minimal détectable par des mesures de vibrations dans un pont routier en service -Étude du pont de la rivière-aux-mulets. Rap. tech. CRGP-2015/05, Centre de recherche en génie parasismique et en dynamique des structures, Université de Sherbrooke, Octobre 2015.
- [35] C. VENTURA, P. ANDERSEN, L. MEVEL et M. DÖHLER, Structural Health Monitoring of the Pitt River Bridge in British Columbia, Canada. In 6th World Conference on Structural Control and Monitoring. European Association for the Control of Structures, Pavia, Italia, 2014.
- [36] K.-Y. WONG, Instrumentation and health monitoring of cable-supported bridges. Structural control and health monitoring, 11(2):91–124, 2004.
- [37] C. O. YANG, M. YEN et N. WENG, The development of an on-line structural health monitoring system based on wireless sensor networks. *Electronic Journal of Structural Engineering -Special Issue : Wireless Sensor Networks and Practical Applications*, p. 45–54, 2010.
- [38] H. ZHOU, T. CHAN, J. WANG et Y. NI, A structural health monitoring data management system for instrumented cable-supported bridges. Rap. tech., Hong Kong Polytechnic University, Hung Hom, Hong Kong, 2006.