



HAL
open science

Conception et déploiement d'IHM avec Matlab pour l'acquisition et le traitement des signaux -Application aux communications acoustiques sous-marines

Thierry Le Gall

► To cite this version:

Thierry Le Gall. Conception et déploiement d'IHM avec Matlab pour l'acquisition et le traitement des signaux -Application aux communications acoustiques sous-marines. 3ème Ecole Technologique du Réseau des Electroniciens (RdE) du CNRS 2021, Nov 2021, Saint Malo, France. <hal-03790598>

HAL Id: hal-03790598

<https://hal.science/hal-03790598v1>

Submitted on 28 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Conception et déploiement d'IHM avec Matlab pour l'acquisition et le traitement des signaux - Application aux communications acoustiques sous-marines

Thierry LE GALL - Ingénieur R&D

UMR 6285 CNRS LabSTICC, pôle T2I3, équipe SI3

IMT Atlantique - Campus de Brest
Dpt. Mathematical & Electrical Engineering
Technopôle Brest-Iroise - CS 83818, Brest Cedex 03, France

3ème École Technologique du Réseau des Électroniciens (RdE) du CNRS
Saint-Malo, 22-26 novembre 2021

Résumé

Dans le cadre du regroupement des Laboratoires en Sciences et Techniques de l'Information et de la Communication (Lab-STICC, UMR 6285) le département *Mathematical and Electrical Engineering* d'IMT Atlantique met en œuvre une plate-forme expérimentale pour l'acquisition et le traitement des signaux en Transmission Acoustique Sous-Marine (TASM). Dans ce contexte, les algorithmes issus des travaux de recherche implémentés en langage C ou Matlab sont intégrés à des applications IHM développées dans l'environnement GUIDE (Graphical Users Interface Development Environment). Ces applications sont compilées et déployées pour les expérimentations en conditions réelles de propagation. Après une introduction du contexte, nous présentons les phases d'élaboration des applications IHM et nous concluons par des exemples de réalisation en communication acoustique sous-marine.

1 Introduction

Dans cette présentation effectuée en séminaire des retours d'expérience de la 3ème École Technologique du réseau des électroniciens du CNRS [1] nous traitons de la conception et du déploiement d'application IHM pour l'acquisition et le traitement des signaux en communication acoustique sous-marine.

Dans une première partie, nous présentons l'IMT Atlantique et le Département *Mathematical & Electrical Engineering* qui porte, notamment, la thématique d'axe de recherche en Transmission Acoustique Sous-Marine (TASM) dont nous évoquons l'historique et l'état de l'art de la plate-forme expérimentale.

Dans une deuxième partie, nous abordons la conception des applications IHM en présentant l'objectif, la stratégie et les moyens mis en œuvre, puis nous traitons de leur élaboration au niveau des phases de programmation, de compilation et de déploiement.

Dans une troisième partie, nous présentons quelques exemples de réalisation d'application en matière de virtualisation d'instrumentation et d'intégration de modem de communication acoustique sous-marine pour les essais en mer.

Enfin, nous concluons sur l'apport de la conception et du déploiement d'application IHM pour la recherche et l'expérimentation sur les transmissions acoustiques sous-marines en évoquant des perspectives d'évolution de notre stratégie de développement pour la plate-forme TASM.

2 Contexte

Notre activité de recherche et de développement en transmission acoustique sous-marine se situe au sein de l'établissement IMT Atlantique [2] et du département *Mathematical & Electrical Engineering* [3].

2.1 L'IMT Atlantique

L'Institut Mines-Télécom Bretagne Pays-de-la-Loire (IMT Atlantique) est un établissement d'Enseignement Supérieur et de Recherche (ESR) né de la fusion au 1er janvier 2017 de l'École Nationale Supérieure des Télécommunications de Bretagne (ENST-Bretagne) et de l'École des Mines de Nantes (EMN). Son ambition, en France et à l'international, est de constituer une référence dans le domaine des sciences du numérique, de l'énergie et de l'environnement. L'IMT Atlantique est implanté sur 3 campus : Brest, Nantes et Rennes (Figure 1).

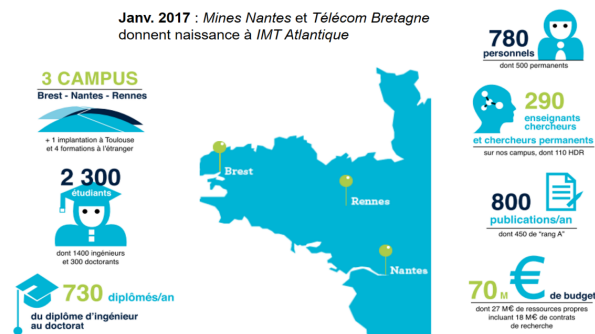


FIGURE 1 – L'IMT Atlantique en chiffres

2.1.1 Le Département MEE

Situé sur le campus de Brest (Figure 2), le département Mathematical & Electrical Engineering (MEE) (Figure 3) s'appuie sur l'expertise scientifique d'une centaine de membres, dont 37 permanents, réunis autour d'un projet fédérateur :

"De l'observation aux algorithmes : apprendre, communiquer et traiter l'information"

Les thématiques de recherche du département MEE sont : les communications numériques, la science des données, l'intelligence artificielle, la conception de circuits et les algorithmes embarqués, pour différents domaines d'application :

- les télécommunications (réseaux 5G, communications sous-marines, IoT),
- l'environnement (surveillance environnementale, télédétection satellitaire et aéroportée, océanographie, écologie marine),
- la défense (guerre électronique, sécurité de la couche physique, cybersécurité),
- la santé (aide au diagnostic, neurosciences cognitives, neuro-imagerie).



FIGURE 2 – L'IMT Atlantique (campus de Brest)



FIGURE 3 – Le Département Mathematical & Electrical Engineering d'IMT Atlantique (bât. K1)

Les travaux de recherche du département MEE sont effectués au sein de 7 équipes rattachées à l'UMR 6285 CNRS LabSTICC [4] :

- BRAIn : Better Representations for Artificial Intelligence,
- CODES : Codage et Décodage de l'information,
- CoSyDe : Communication System Design,
- MATRIX : Modèles et Algorithmes pour le TRaItement et l'eXtraction de l'information,
- OSE : Observations Signal & Environnement,
- SI3 : Sécurité, Intelligence et Intégrité de l'Information,
- 2AI : Algorithm Architecture Interaction.

2.2 L'axe de recherche TASM

L'axe de recherche Transmissions Acoustiques Sous-Marines (TASM) du département MEE s'applique à 3 domaines : télécommunications, environnement (mer) et défense. Il présente un historique d'activité sur une période remarquablement longue et jalonnée de défis scientifiques ayant permis l'acquisition d'une expertise de terrain en communication acoustique sous-marine et des moyens de mise en œuvre d'une plate-forme expérimentale.

2.2.1 Historique des activités

L'historique des activités de recherche et de développement en transmission acoustique sous-marine au département MEE s'étend sur plus de 30 ans (Figure 4). Vers la fin des années 80, les efforts de recherche en communication numérique à l'origine de l'émergence de la téléphonie mobile cellulaire s'intéressent également au canal sous-marin afin d'améliorer les faibles performances des systèmes analogiques existants. Les ondes électromagnétiques se propageant très difficilement dans l'eau, ce sont essentiellement les ondes acoustiques qui sont utilisées comme signal porteur d'information pour les communications sous-marines [5].

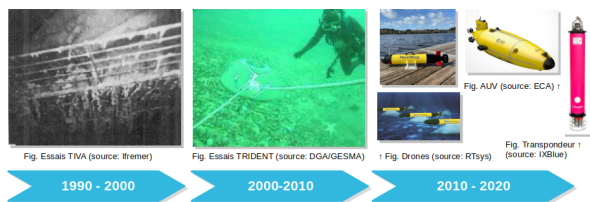


FIGURE 4 – 30 ans d'activités de R&D du département Mathematical & Electrical Engineering en Transmissions Acoustiques Sous-Marines

Au début des années 90, un partenariat est signé avec Ifremer et la Sté Orca Instrumentation pour la mise au point d'un système de Transmission d'Image par voie Acoustique (TIVA) expérimenté lors d'une campagne d'exploration du Titanic [6], [7]. Cette expérimentation démontre la faisabilité d'une transmission numérique d'information par onde acoustique en configuration verticale à faibles-débits et par grande profondeur. Elle permet à l'équipe de recherche d'acquérir une première expertise de terrain sur les transmissions ASM ainsi que d'intégration algorithmique temps-réel sur processeur de traitement numérique du signal Motorola 56001. Au milieu des années 90 et jusqu'au seuil des années 2000, cette expertise est renforcée par l'étude des transmissions sur canal horizontal et l'implémentation d'un système mono-capteur de téléphone acoustique sous-marin basé sur des algorithmes d'égalisation adaptative et de codage de la parole [8]-[14].

Au début des années 2000, un partenariat est signé avec la Délégation Générale de l'Armement (DGA) dans le cadre d'un Programme d'Études Amonts (PEA) du Groupe d'Études Sous-Marines de l'Atlantique (GESMA) visant à augmenter l'autonomie des véhicules sous-marins autonomes

(AUV : *Autonomous Underwater Vehicle*) utilisés en guerre des mines. Dans ce contexte, le projet TRIDENT permet de démontrer la faisabilité d'une transmission numérique d'information par onde acoustique en configuration horizontale à hauts-débits et par faible profondeur. Il permet à l'équipe de recherche de conforter son expertise scientifique en égalisation adaptative et synchronisation multi-capteurs ainsi que son savoir-faire d'intégration sur plate-forme de traitement du signal en temps-réel Texas Instruments TMS320C6201 [15]-[24]. Au milieu des années 2000, les expérimentations révèlent la nécessité de prendre davantage en considération les contraintes liées aux mouvements des véhicules sous-marins.

Au début des années 2010, l'équipe oriente ses recherches sur la compensation d'effet Doppler, les réseaux sous-marins et les communications sous-marines longues distances. Des multiples partenariats accélèrent le transfert de technologie [25]-[29]. L'évolution de la puissance de calcul des CPU au regard des fréquences utilisées en communication ASM, l'apparition de cartes d'acquisition multi-voies connectables aux environnements de simulation, permettent de valider les algorithmes en quasi-temps réel à partir d'une implémentation en langages de haut-niveau, sans recourir à une programmation en assembleur et l'intégration sur une carte spécifique. L'évolution des environnements de programmation permet d'intégrer les programmes de simulation sous la forme d'application IHM déployable sur une plate-forme d'expérimentation générique TASM.

2.2.2 Plate-forme expérimentale

La plate-forme d'expérimentation TASM du département MEE permet de valider les algorithmes de communication acoustique sous-marine en condition réelle de transmission, dans une cuve au laboratoire ou en mer lors des séances d'essai. La plate-forme s'articule autour de composants matériels et logiciels. Elle est constituée d'un poste d'émission et d'un poste de réception [32]-[33].

Le poste d'émission (Figure 5) est constitué d'un ordinateur durci pour l'utilisation en conditions extérieures [34] sur lequel s'exécutent les programmes de génération des formes d'onde porteuses d'information. L'ordinateur est connecté à une carte de conversion numérique/analogique du signal [37]. Un banc de filtres elliptiques paramétrables envoie le signal limité en bande à un amplificateur de puissance [35] muni d'une sortie auxiliaire permettant la mesure du signal à

l'oscilloscope [38]. Le signal analogique amplifié est envoyé au transducteur d'émission qui matérialise le signal électrique en onde acoustique [36].

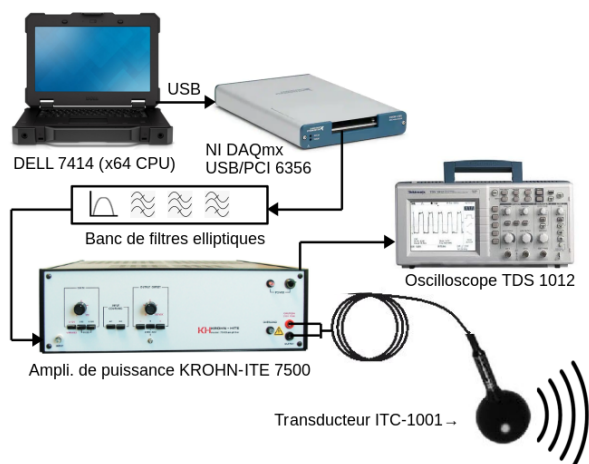


FIGURE 5 – Le poste d'émission de la plate-forme d'expérimentation pour les transmissions acoustiques sous-marines (TASM)

Le poste de réception (Figure 6) est constitué d'un réseau de 4 hydrophones pré-amplifiés réalisant la conversion de l'onde acoustique en signaux électriques [39]. Ces hydrophones sont reliés au boîtier amplificateur-conditionneur [40] qui distribue les signaux analogiques larges-bandes aux bancs de filtres elliptiques réalisant le filtrage anti-repliement en amont d'une carte de conversion analogique/numérique multi-voies [37]. Accessoirement, un oscilloscope [38] peut être connecté sur une voie afin d'afficher le signal reçu. La carte d'acquisition est reliée en USB à l'ordinateur [34] où s'exécutent les applications IHM permettant l'enregistrement et le traitement numérique du signal en quasi-temps réel.

Hormis l'aspect matériel, la plate-forme TASM est constitué de composants logiciels qui permettent l'intégration des algorithmes de communication acoustique sous-marine au sein d'application IHM déployées sur les postes d'émission et de réception.

3 Conception d' IHM

Dans cette section, nous abordons la conception d'application IHM pour l'expérimentation des algorithmes de communication acoustique sous-marine avec la plate-forme TASM. Dans un premier temps, nous présentons l'objectif d'intégration des algorithmes au sein d'application IHM ainsi que la stratégie et les moyens mis en œuvre pour les

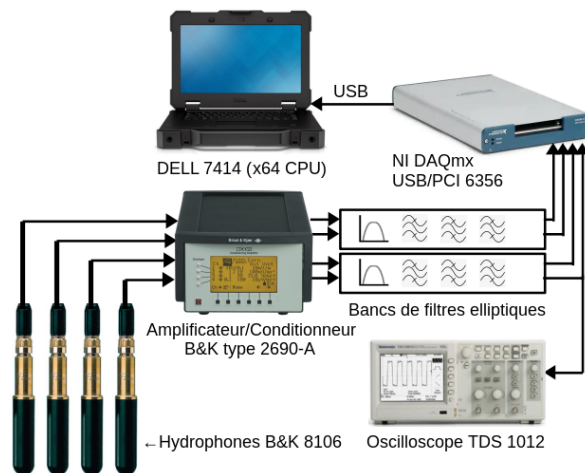


FIGURE 6 – Le poste de réception de la plate-forme d'expérimentation pour les transmissions acoustiques sous-marines (TASM)

développements logiciels. Dans un deuxième temps, nous traitons des phases d'élaboration d'application IHM au niveau de la programmation, de la compilation, et du déploiement.

3.1 Objectif, stratégie, moyens

Dans le contexte d'utilisation de la plate-forme TASM, l'objectif d'une application IHM est d'intégrer les algorithmes de communication acoustique sous-marine sous une forme permettant leur expérimentation in-situ lors des séances d'essai en mer. Les algorithmes peuvent ainsi être validés sur des signaux réels (Figure 7).

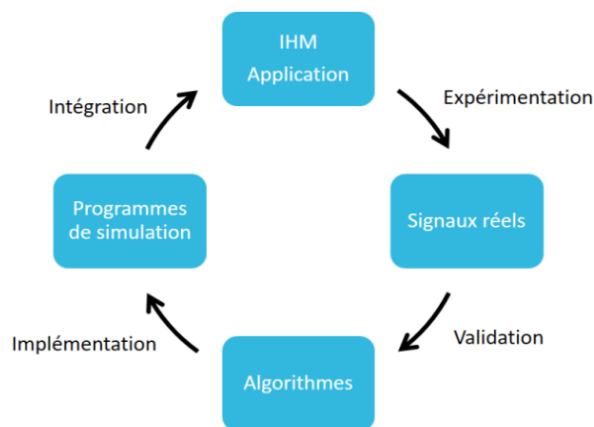


FIGURE 7 – Le cycle de développement des algorithmes de communication acoustique sous-marine

L'intégration des programmes de simulation au sein d'une application IHM munie d'un choix judicieux des paramètres de contrôle (API : *Application*

Protocol Interface) permet d’opérer les algorithmes en faisant abstraction de l’expertise nécessaire à la connaissance du code source. Un affichage pertinent des indicateurs de performance associé à une exécution des algorithmes en quasi-temps réel sur les signaux en cours d’enregistrement permet de valider leur bon fonctionnement ainsi que l’intégrité des données enregistrées sans attendre le retour au laboratoire.

En sortie du cycle de développement, l’application IHM peut être utilisée comme démonstrateur et le code source élaboré afin de répondre aux contraintes d’intégration peut faciliter le transfert de technologie.

Afin d’optimiser les efforts de développement, les applications IHM sont élaborées dans le même environnement de programmation que les simulations, en l’occurrence Matlab [42], par ailleurs très utilisé dans la communauté de la recherche. La conception d’IHM est réalisée par l’interface GUIDE (*Graphical User’s Interface Development Environment*) intégrée en standard à Matlab [43]. Le contrôle du pilote des cartes d’acquisition et de restitution des signaux s’effectue au moyen du module *Data Acquisition*. La compilation des IHM en exécutable applicatif est réalisée par le module *Compiler* invoquant le compilateur GCC MinGw-w64 [44]. L’exécution des livrables applicatifs hors de l’environnement de développement sur un poste d’exploitation ne disposant pas d’une licence Matlab est rendue possible au moyen du module *Component Runtime* (Figure 8).

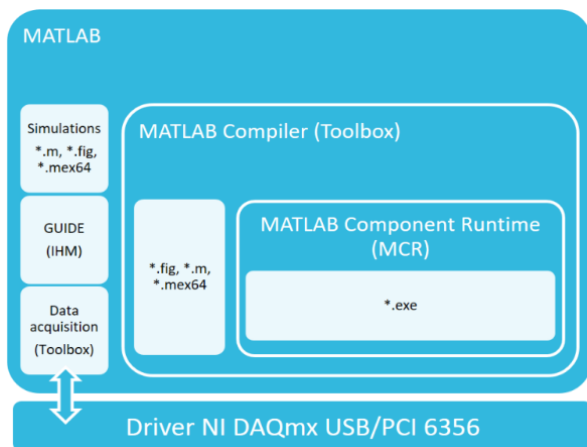


FIGURE 8 – L’environnement de développement d’application IHM de la plate-forme TASM

Enfin, les déploiements des applications IHM sur les postes d’exploitation sont effectués au moyen

des fichiers d’installation (*setup*) élaborés à partir du logiciel libre *Inno Setup* [45]. Préalablement, la conception d’application IHM implique différentes phases d’élaboration.

3.2 Phases d’élaboration

Dans cette section, nous abordons les différents phases d’élaboration des applications IHM au niveau de la programmation, de la compilation et du déploiement.

3.2.1 Programmation

La programmation des IHM s’effectue au moyen de l’interface GUIDE qui peut être invoquée depuis la fenêtre de commande (*Command Window*) dans Matlab avec l’instruction : `>>guide`. Une interface graphique de développement apparaît qui permet la conception interactive d’une IHM par ”glisser-déposer” d’objets sur une grille de positionnement à partir d’une bibliothèque de classes (Figure 9).

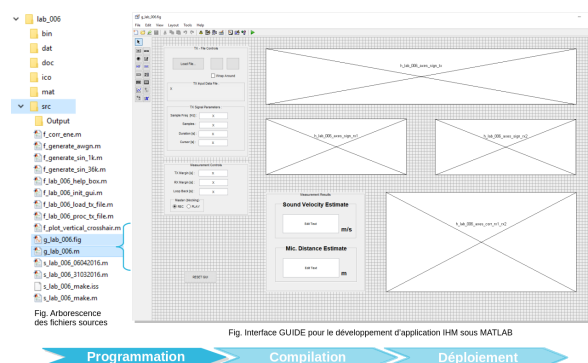


FIGURE 9 – L’interface de développement intégrée GUIDE pour la programmation d’application IHM sous Matlab

La conception interactive d’IHM par ”glisser-déposer” dispense le développeur d’effort de programmation des éléments graphiques de l’interface pour mieux se concentrer sur l’intégration du code de simulation au niveau de la logique d’application et des paramètres d’API. L’IHM en cours de développement peut être exécutée dans GUIDE et déboguée sur points d’arrêt dans l’environnement Matlab. A l’exécution du projet d’interface, GUIDE génère automatiquement 2 fichiers : *.fig contenant les informations graphiques et *.m permettant les appels (callback) des fonctions de traitement du signal (Figure 9).

3.2.2 Compilation

Une première étape de compilation permet de générer un exécutable applicatif à partir des fichiers sources *.fig et *.m. Pour cela, le module *Compiler* est utilisé par un script Matlab muni de l'instruction : `>>mcc` invoquant le compilateur GCC MinGw-w64 qui génère un fichier binaire exécutable (*.exe) compatible Windows 64-bits (Figure 10).

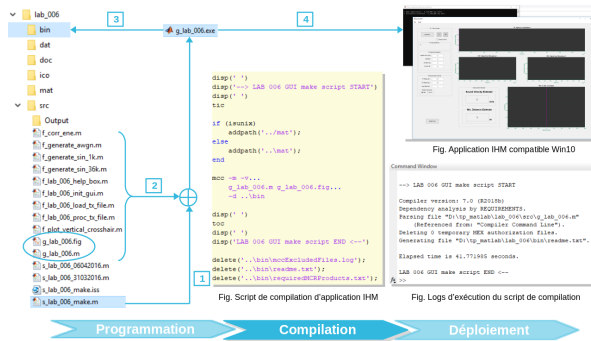


FIGURE 10 – La compilation d’IHM en exécutable applicatif Windows 64-bits sous Matlab sur un poste de développement

A ce stade de la compilation, le fichier binaire généré par *Compiler* est reconnu par le système d’exploitation du poste de développement comme une application 64-bits valide si bien que l’IHM est exécutable hors de l’environnement Matlab à partir de l’emplacement du fichier *.exe. Toutefois, le déplacement du binaire dans un autre répertoire peut générer des dysfonctionnements si des fichiers annexes (*.dat, *.pdf, etc...) utilisés par l’application relativement à l’exécutable ne sont pas repositionnés correctement. Par conséquent, l’exécutable applicatif est difficilement déployable en l’état c’est pourquoi une deuxième étape de compilation est mise en œuvre.

Une deuxième étape de compilation permet de générer un livrable applicatif à partir de l’exécutable et de ses fichiers annexes. Cette étape est réalisée au moyen d’un script édité, compilé et exécuté avec le logiciel libre Inno Setup [45]. Le deuxième fichier exécutable obtenu par la compilation avec Inno Setup est un fichier binaire d’installation (setup). Il contient l’ensemble des fichiers d’application et les instructions nécessaires à la création des répertoire, des sous-répertoires, au positionnement des fichiers ainsi que de création d’icône et de menu de lancement (Figure 11).

Le livrable applicatif obtenu à l’issu des deux

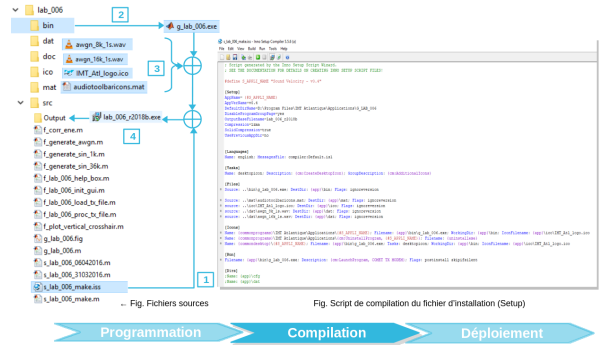


FIGURE 11 – L’édition d’un script de déploiement et la génération d’un fichier binaire d’installation avec Inno Setup sur un poste de développement

étapes de compilation peut être distribué sous la forme d’un unique fichier exécutable en vue du déploiement de l’application IHM sur un ou plusieurs postes d’exploitation.

3.2.3 Déploiement

L’objectif du déploiement est de permettre l’utilisation d’application IHM sur un poste d’exploitation qui ne dispose pas de licence Matlab. Pour cela, *Matlab Component Runtime* (MCR) est tout d’abord installé sur le poste puis le fichier binaire d’installation (setup) est exécuté en mode administrateur. L’installation déroule des menus de dialogue avec l’utilisateur permettant de sélectionner l’emplacement des répertoires et des fichiers d’application. A la fin de l’installation, l’application IHM peut être lancée au moyen d’une icône positionnée sur le bureau ou par sélection dans le menu ”Démarrer” du système d’exploitation (Figure 12).

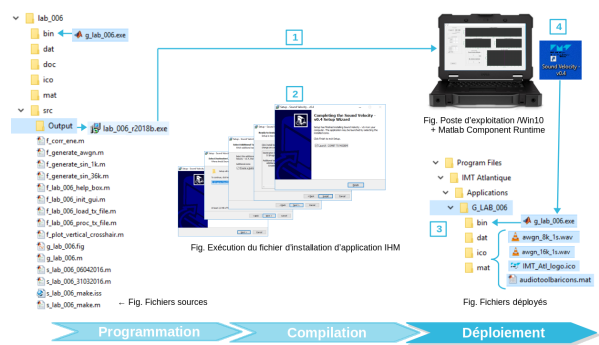


FIGURE 12 – Le déploiement d’application IHM à partir du fichier binaire d’installation sur un poste d’exploitation

A noter que le déploiement ne dépose aucun fichier de code source sur le poste d’exploitation mais seulement les fichiers annexes et l’exécutable appli-

catif dans une version chiffrée par *Matlab Compiler* qui nécessite la connaissance d'une clé pour la rétro-compilation. Cette fonctionnalité peu s'avérer utile du point de vue de la protection de propriété intellectuelle ou d'information sensible pour des expérimentations en dehors d'une Zone à Régime Restrictif (ZRR).

4 Exemples de réalisation

Dans cette section, nous présentons quelques exemples de réalisation d'application IHM en matière de virtualisation d'instrument de mesure et d'intégration de modem de communication acoustique sous-marines pour les expérimentations lors des séances d'essai en mer.

4.1 Virtualisation d'instrument

Au poste de réception de la plate-forme TASM un oscilloscope peut être connecté en amont de la carte d'acquisition afin de vérifier la présence d'un signal, de visualiser sa forme et de mesurer son amplitude (Figure 6). Cette méthode d'instrumentation présente l'inconvénient de diviser le signal uniquement sur l'une des voies cependant que les autres capteurs sont pas surveillés. Par ailleurs, la visualisation des signaux analogiques en amont de la carte d'acquisition ne révèle pas les éventuels problèmes de saturation des convertisseurs à l'origine des non-linéarités dans la représentation numérique des signaux. Au prix d'un investissement assez onéreux, un oscilloscope 4-voies peut être utilisé afin de surveiller simultanément les capteurs sans déséquilibrer les voies. Toutefois, cette solution ne permet pas de garantir l'absence de saturation au niveau des signaux enregistrés numériquement. L'utilisation d'une application IHM oscilloscope virtuel multivoies en aval de la carte d'acquisition permet de résoudre à moindre coût l'ensemble des inconvénients des mesures effectuées en amont (Figure 13).

L'application IHM oscilloscope multivoies déployée en réception sur un poste d'exploitation bénéficie des fonctionnalités du module *Data Acquisition* qui permettent de contrôler le driver de la carte de conversion multivoies NI-DAQmx au niveau des paramètres d'acquisition (fréquence d'échantillonnage, taille des buffers, niveau électrique de saturation des convertisseurs). L'application permet d'afficher et d'analyser simultanément les signaux des 4 capteurs dans le domaine numérique. La calibration de la chaîne de communication acoustique sous-marine peut ainsi

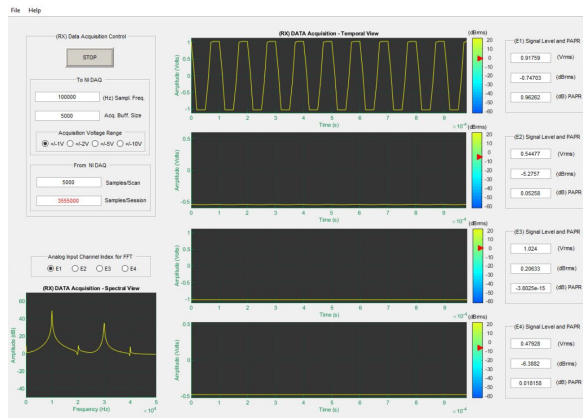


Fig. Exemple d'application IHM : oscilloscope virtuel multivoies (synchrones)

FIGURE 13 – L'application IHM oscilloscope 4-voies pour le contrôle d'intégrité des signaux enregistrés lors des séances d'essai en mer avec la plate-forme TASM

être effectuée en vérifiant la présence du signal et l'absence de saturation sur chacune des voies en réception, ce qui permet de fiabiliser la campagne d'enregistrement des signaux réels. L'application peut être aisément enrichie de nombreuses fonctionnalités d'analyse (FFT, dBBrms, PAPR, etc...) suivant les besoins expérimentaux.

4.2 Intégration de modem de communication ASM

L'intégration des programmes de simulation des modem de communication acoustique sous-marine en applications IHM facilite la mise en œuvre des algorithmes au niveau du réglage des paramètres de contrôle et la visualisation des performances en quasi-temps réel lors des séances d'essai en mer avec la plate-forme TASM (Figure 14).

A l'issu des séances d'essai en mer, les signaux enregistrés sur canal réel de propagation peuvent être exploités en laboratoire au niveau des programmes de simulation afin de perfectionner les algorithmes et de progresser dans la thématique de recherche. Les applications IHM intégrant les modems de communication acoustique sous-marine peuvent être utilisées comme démonstrateurs pour illustrer le savoir-faire de l'équipe de recherche et faciliter le transfert de technologie (Figure 15).

5 Conclusions & Perspectives

Nous avons présenté un retour d'expérience concernant la conception et le déploiement d'IHM avec Matlab pour les expérimentations en recherche sur les transmissions acoustiques sous-marines.

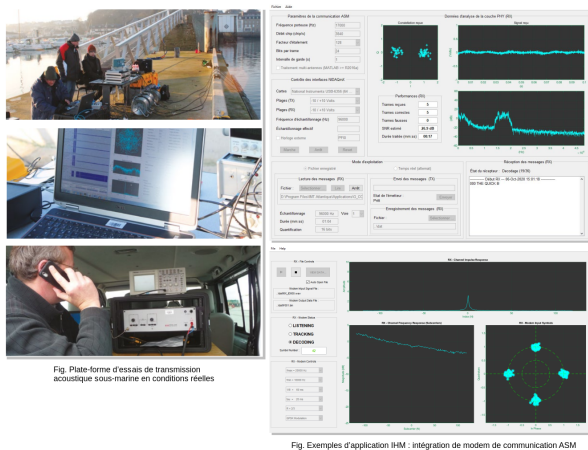


FIGURE 14 – Exemples d’application IHM pour la validation des algorithmes de communication acoustique sous-marine lors des séances d’essai en mer avec la plate-forme TASM

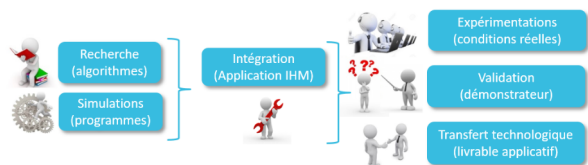


Fig. Apports de l’intégration algorithmique pour la R&D en communications acoustiques sous-marines

FIGURE 15 – Les apports d’intégration des algorithmes de communication acoustique sous-marine sous la forme d’application IHM

Notre approche de programmation nous permet d’élaborer des interfaces graphiques en ”glisser-déposer” dans le même environnement que les scénarios de simulation, ce qui permet d’optimiser les efforts d’intégration en se concentrant sur la logique d’application et les paramètres de contrôle des algorithmes.

La compilation des IHM en livrables applicatifs permet de déployer facilement les solutions intégrant les modems de communication acoustique sous-marine, sans licence supplémentaire et sans divulgation des codes sources de simulation des algorithmes, sur des postes d’exploitation.

La virtualisation d’instrument de mesure nous permet de fiabiliser à moindre coût les enregistrements des signaux cependant que les modem de communication acoustique sous-marine peuvent être aisément manipulés lors des séances d’essai en mer. A l’issu, les IHM applicatives intégrant les modems de communication acoustique sous-marine peuvent être utilisées comme démonstrateurs et

favoriser les transferts de technologie.

Cependant, l’environnement de développement sous licence Matlab s’avère peu adapté à la gestion des processus concurrents et l’outil GUIDE présente des limitations en matière de richesse d’objet. Sur ce dernier point, l’évolution récente de l’outil GUIDE vers App Designer [46] apporte une amélioration notable tout en imposant un effort conséquent de migration des applications existantes.

C’est pourquoi, en parallèle des efforts d’optimisation de notre architecture applicative sous la forme de briques logicielles réutilisables, nous explorons la conception et le déploiement d’application IHM en langage Python [47]-[50] en nous appuyant sur des version d’outils libres et cependant puissants tels que Eric6 [51] et QtDesigner [52] ainsi que sur les bibliothèques PyQt5 [53]-[60] et PyQtGraph [61], [62].

Références

- [1] <https://www.electroniciens.cnrs.fr/3eme-ecole-technologique/>
- [2] <https://www.imt-atlantique.fr/fr>
- [3] <https://www.imt-atlantique.fr/fr/1-ecole/departements-d-enseignement-recherche/mathematical-electrical-engineering>
- [4] <https://labsticc.fr/fr>
- [5] https://www.bruit.fr/images/acoustique_techniques/AT48-38-42_compressed.pdf
- [6] http://documents.irevues.inist.fr/bitstream/handle/2042/11763/AR6_19.pdf
- [7] G Ayela, M. Nicot and X Lurton, ”New Innovative Multimodulation Acoustic Communication System”, in *Proc. Oceans’94*, Brest, pp 292-295.
- [8] Gérard Lapiere, Nicolas Beuzeulin, Joël Labat, Joël Trubuil, André Goalic, et al.. ”1995-2005 : Ten years of active research on underwater acoustic communications in Brest”, *Oceans’05*, Jun 2005, Brest, France.
- [9] A. Goalic, C. Laot and S. Saoudi, ”Real Time Implementation of a Low Bit Rate Coder for an Acoustic Underwater Phone on a Fixed Point DSP Motorola 56001”, *International Conference on Signal Processing Applications and Technology, ICSPAT’92*, Boston, M.A., USA, 1992.
- [10] Joël Labat, ”Apport de l’égalisation auto-adaptative en communications acoustiques sous-marines”, *thèse de doctorat en Physique de l’Université de Bretagne Occidentale*, 1994.

- [11] J. Labat, "Real time underwater communications," *Proceedings of OCEANS'94*, Brest, France, 1994, pp. III/501-III/506 vol.3.
- [12] J. Labat, O. Macchi, C. Laot and N. Le Squin, "Is training of adaptive equalizers still useful?," *Proceedings of GLOBECOM'96. 1996 IEEE Global Telecommunications Conference*, London, UK, 1996, pp. 968-972 vol.2.
- [13] Christophe Laot, "Egalisation autodidacte et turbo-égalisation. Application aux canaux sélectifs en fréquences", *thèse de l'Université de Rennes 1*, 1997.
- [14] J. Labat, O. Macchi and C. Laot, "Adaptive decision feedback equalization : can you skip the training period?," in *IEEE Transactions on Communications*, vol. 46, no. 7, pp. 921-930, July 1998,
- [15] http://recherche.imt-atlantique.fr/trident_old/index.html
- [16] J. Trubuil, T. Le Gall, G. Lapiere and J. Labat, "Development of a Real-Time High Data Rate Acoustic Link", *IEEE Oceans 2001*, Hawaii, USA.
- [17] J. Labat and C. Laot, "Blind adaptive multiple-input decision-feedback equalizer with a self-optimized configuration," in *IEEE Transactions on Communications*, vol. 49, no. 4, pp. 646-654, April 2001.
- [18] G Lapiere, J Labat, J Trubuil, "Evaluation of high dated spleen acoustic link : contribution of blind spatio- temporal equalization", *Proc. ECUA 2000*, Lyon, France, Vol.1, pp. 549-554.
- [19] J. Trubuil, G. Lapiere, T. Le Gall and J. Labat, "Real-time high data rate acoustic link based on spatio-temporal blind equalization : the TRIDENT acoustic system," *OCEANS '02 MTS/IEEE*, Biloxi, MI, USA, 2002, pp. 2438-2443 vol.4.
- [20] Thierry Le Gall, *Étude et implémentation d'un égaliseur autodidacte, application au canal acoustique sous-marin*, mémoire d'ingénieur CNAM, ENST-Bretagne, mars 2002.
- [21] G. Lapiere, J. Labat, J. Trubuil, "Iterative equalization for underwater acoustic channel – Potentiality for the TRIDENT system", *Proc. OCEANS 2003*, San Diego, pp1547-1553.
- [22] Gérard Lapiere, "Contribution à l'égalisation adaptative autodidacte multi-dimensionnelle – Application au canal acoustique sous-marin ", thèse de l'Université de Rennes I, le 28 novembre 2003.
- [23] J. Trubuil, G. Lapiere and J. Labat, "Real time transmission of images and data through underwater acoustic channel the TRIDENT system," *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*, Anchorage, AK, 2004, pp. 1053-1056 vol.2.
- [24] A. Goalic, J. Trubuil, G. Lapiere and J. Labat, "Real time low bit rate speech transmission through underwater acoustic channel," *Europe Oceans 2005*, Brest, France, 2005, pp. 319-321 vol. 1.
- [25] Christophe Laot, "Récepteurs pour communications numériques sur canaux sélectifs en fréquences", *habilitation à diriger des recherches*, Université de Bretagne Occidentale, 2009.
- [26] François-Xavier Socheleau, "Communications acoustiques sous-marines sur canal fortement dispersif en temps et en fréquence : point de vue de la théorie de l'information", thèse de l'Université de Bretagne occidentale, 2011.
- [27] André Goalic, "Traitements temps réel en codage source et canal pour des Communications hertziennes et acoustiques sous marines", *habilitation à diriger des recherches*, Université de Bretagne Occidentale, 2012.
- [28] J. Trubuil, T. Le Gall, and T. Chonavel, "Doppler, synchronization and channel estimation for OFDM underwater acoustic communications", *IEEE Oceans 2014*, Taipei, Taiwan.
- [29] https://www.ixblue.com/sites/default/files/2020-03/Canopus_2020.pdf
- [30] <https://www.youtube.com/watch?v=gwxfMg5oRSA>
- [31] <https://www.ecagroup.com/en/solutions/alistar-3000-auv-autonomous-underwater-vehicle>
- [32] <https://www.imt-atlantique.fr/fr/recherche-et-innovation/plateformes-de-recherche/tasm>
- [33] <https://www.imt-atlantique.fr/sites/default/files/recherche/plateformes/tasm.pdf>
- [34] https://dl.dell.com/topicspdf/latitude-14-7414-laptop_owners_manual_en-us.pdf
- [35] <https://www.krohn-hite.com/7500Data.pdf>
- [36] https://www.gavial.com/_files/ugd/8aa48d_9dd2d9e2ca314cbe9de4d099fd2a8138.pdf
- [37] <https://www.ni.com/pdf/manuals/374452c.pdf>

- [38] <https://www.valuetronics.com/product/tds1012-tektronix-digital-oscilloscope-used>
- [39] <https://www.bksv.com/fr-FR/products/transducers/acoustic/microphones/hydrophones/8106>
- [40] <https://www.bksv.com/en/products/transducers/conditioning/microphone/2690-A-0F2>
- [41] <https://www.youtube.com/watch?v=BuZ7NJPH2GU>
- [42] https://fr.mathworks.com/products/matlab.html?s_tid=hp_products_matlab
- [43] <https://fr.mathworks.com/discovery/matlab-gui.html>
- [44] <https://www.mingw-w64.org/>
- [45] <https://jrsoftware.org/isinfo.php>
- [46] App Designer: Créer des applications web et desktop dans MATLAB - Site Web The Mathworks Inc.
- [47] The Python Programming Language - Official Website
- [48] Python Course - Numerical Programming - Online Tutorial
- [49] Matplotlib - Visualization with Python - Official Website
- [50] Matt HARRISON, *Illustrated Guide to Python 3*, Ebook, 2017.
- [51] The Eric6 Python IDE Project - Official Website
- [52] QtDesigner - Online Manual
- [53] The PyQt5 Project - Official Website
- [54] Pierre DENIS et Thibaut CUVELIER, *Créer des applications graphiques en Python avec PyQt5*, Editions D-BookeR, 2017. ISBN (EPUB) : 978-2-8227-0516-5.
- [55] B.M. HARWANI, *Qt5 Python GUI Programming - Cookbook*, Packt Publishing Ltd, 2018.
- [56] Martin Fitzpatrick, *Create Simple GUI Applications, with Python & Qt5*, Leanpub book, 4th Edition, 2019.
- [57] Robert JOHANSSON, *Numerical Python - Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib*, 2nd Edition, Apress, 2019.
- [58] Joshua M. WILLMAN, *Beginning PyQt - A Hands-on Approach to GUI Programming*, Apress Ed., 2020. ISBN-10 : 1484258568, ISBN-13 : 978-1484258569.
- [59] Martin FITZPATRICK, *Create GUI Applications with Python & Qt5 - The hands-on guide to making apps with Python, PyQt Edition*, Leanpub book, v0.4, 2020.
- [60] Joshua M. WILLMAN, *Modern PyQt - Create GUI Applications for Project Management, Computer Vision and Data Analysis*, Apress Ed., 2021, ISBN-13 : 978-1-4842-6603-8.
- [61] PyQtGraph - Official Website
- [62] What is PyQtGraph ? - Online Tutorial