

Architecture Client-Serveur Ouverte pour Travaux Pratiques Distants 24h/24-7j/7 en Electronique

Franck Luthon, Dan Steriu

Université de Pau, Dept Informatique
IUT Château Neuf, Place Paul Bert, 64100 Bayonne, France.

Tel: +33(0)5.59.57.43.44

Email: Franck.Luthon@univ-pau.fr

Résumé

Nous présentons une architecture générique et globale (logiciel et matériel) pour la mise en œuvre à distance de travaux pratiques de laboratoire dans le domaine de l'électronique, de l'automatique ou du traitement de signal.

Les étudiants peuvent non seulement faire des simulations et de l'instrumentation virtuelle, mais ils peuvent surtout faire des expériences réelles en rapport avec la physique du phénomène étudié, par observation ou contrôle. Ceci est réalisé à distance grâce à une connexion TCP/IP au réseau internet.

L'architecture client-serveur est ouverte dans le sens où elle est basée sur le concept de module d'extension (ou plug-in), chaque plug-in correspondant à une expérience de laboratoire différente (par exemple le tracé automatique des courbes de Bode de systèmes linéaires, l'acquisition et le filtrage numérique de signaux).

Un prototype opérationnel est développé sous LabVIEW : pilotage à distance d'instruments de mesure par bus GPIB, de cartes d'acquisition de données (DAQ, PCI ou USB) et de matrices de commutations pour les connexions électriques et la mise sous tension "Marche/Arrêt". Il permet la gestion de connexions multi-utilisateurs, multi-sessions, multi-expériences.

L'avantage d'une telle architecture est la diminution des dépenses financières (grâce à la mise en réseau de dispositifs de mesure coûteux entre différentes institutions universitaires) et le partage des moyens et des contenus pédagogiques pour réaliser des travaux pratiques de laboratoires éloignés (et non pas seulement pour faire des cours magistraux, des séances d'exercices ou de travaux dirigés).

Thématique: T2: Systèmes Educatifs et Dispositifs de Formation

Mots-Clés: Enseignement à distance, Travaux Pratiques, LabVIEW, Electronique, Architecture

Abstract

We present a global architecture (software and hardware) for the remote implementation of practical laboratory works in the field of electronics, automation or signal processing.

The students can not only do simulations and virtual instrumentation, but they can also do real-world experiments linked with the physics of the phenomenon under study, through observation or control. This is done remotely with TCP/IP network connection.

The client-server architecture is open in the sense that it is based on the plug-in concept, each plug-in corresponding to a different lab experiment (e.g. automatic plot of Bode curves of linear systems, signal acquisition and numerical filtering...)

An operational prototype is implemented with LabVIEW: remote control of instruments (through the GPIB network), of data acquisition boards (DAQ, PCI or USB) and of switching matrices for electrical connections and power on/off, management of multi-user connection, multi-session, multi-experiment.

The advantage of such an architecture is the decrease of financial costs (sharing of expensive measurement devices) and the sharing of pedagogical means for real distant practical lab work (not only magistral courses nor tutorial exercises).

Theme Descriptor: T2: Educational systems, professional course and training systems

Keywords: Distant learning, Labwork, LabVIEW, EEE engineering, Cybernetics education

1. INTRODUCTION

1.1. Contexte Pédagogique

Ce travail décrit la conception et la mise en œuvre d'une architecture générique (logiciel et matériel) dédiée à l'enseignement pratique à distance en électronique, en automatique ou en traitement de signal. Le laboratoire (ou la salle de travaux pratiques TP) est accessible 24H sur 24 via le réseau internet avec le protocole TCP/IP. Il peut ou non y avoir planification des séances de TP distants: soit l'étudiant se connecte quand il veut (le soir, à midi, le week-end etc.), soit un emploi du temps semestriel lui indique les créneaux temporels de connexion à respecter pour la réalisation des TP distants.

L'étudiant, depuis son domicile, peut non seulement faire des simulations et de l'instrumentation virtuelle, comme c'est le cas avec l'outil Java-DSP (J-DSP) développé par l'université d'état d'Arizona [12, 13] (simulation en ligne interactive pour le traitement du signal numérique), mais il peut aussi réaliser de vraies expériences pratiques (vraies mesures) pour caractériser à distance des circuits ou des systèmes physiques.

Toutes les ressources technologiques présentes dans la nature (information, matériau et énergie) sont prises en compte dans les objectifs pédagogiques de la formation de l'ingénieur, ceci grâce au contrôle à distance ou à la simple observation, dite *monitoring* en anglais [4].

1.2. Architecture à base de plug-in

L'architecture client-serveur est basée sur le concept de plug-in, chaque plug-in correspondant à une expérience différente (un TP différent) disponible du côté du serveur de manipulations (Fig. 1).

Cinq types de modules clients (plug-ins) sont fournis:

- expériences de TP,
- simulations,
- pages web (démarré le web browser avec l'adresse URL voulue),
- démonstrations (pour illustration pendant un cours en ligne)
- autres (e.g. aide à l'édition de rapports et la rédaction de compte-rendus de TP, représentation et visualisation avancées de données, formulaire d'évaluation, autoévaluation, quizz etc.)

En ce qui concerne les plug-in côté serveur, un modèle (dit *template*, cf Fig. 2) est aussi disponible, qui permet à n'importe quel enseignant-tuteur autorisé de développer facilement une nouvelle manipulation et de la rajouter dans la liste du menu des expériences déjà disponibles.

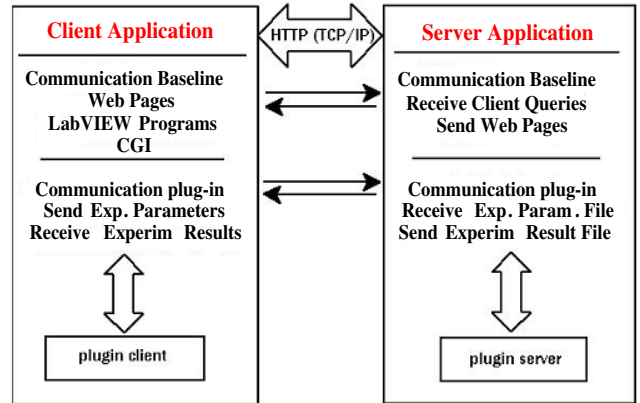


Fig. 1. Architecture client-serveur ouverte : le serveur gère la liste des expériences et les accès utilisateur, exécute les demandes des clients, ceci pour un usage optimisé du système matériel. Le client envoie les paramètres expérimentaux puis télécharge les résultats dès qu'ils sont disponibles.



Fig. 2. Modèle de Plug-in (template): avec gestion d'erreur, d'identifiant, et transmission de paramètres (data)

1.3. Applications

Les applications sont nombreuses : en travaux pratiques d'électronique [9] (voire même en bureau d'études ou projets), acquisition de données et laboratoire de traitement du signal, laboratoire de traitement d'image, automatique, asservissement et contrôle des systèmes, supervision de composants logiques programmables (PLD), interfaces de communication standards (ports série, parallèle, USB).

2. PRÉSENTATION DE L'ARCHITECTURE LOGICIELLE

2.1. Description de l'Application Serveur

La partie principale est le serveur Web qui travaille avec différents programmes comme montré sur la Fig. 3:

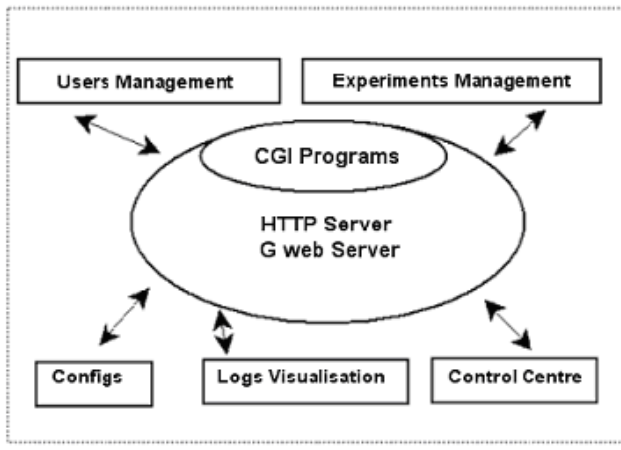


Fig. 3. Structure de l'application serveur: à base de scripts CGI

Les actions simplifiées du serveur sont présentées Fig. 4.

Le serveur HTTP est un G web serveur de LabVIEW. Il est utilisé avec le toolkit Internet Developer et la communication entre client et serveur se fait par CGI.

La gestion des accès utilisateurs (multi-session, multi-usager, multi-expérience) est assurée par le centre de contrôle du côté serveur, qui est situé physiquement près des vrais appareils de mesures, des systèmes et des sources d'alimentation. Le serveur gère la liste des demandes des clients (étudiants distants) de telle sorte que le système matériel est utilisé de manière optimale. Les principales fonctions du centre de contrôle sont les suivantes (Fig. 5):

- démarrer le serveur web,
- démarrer le serveur de face avant distante (remote panel),
- démarrer le chien de garde (watchdog),

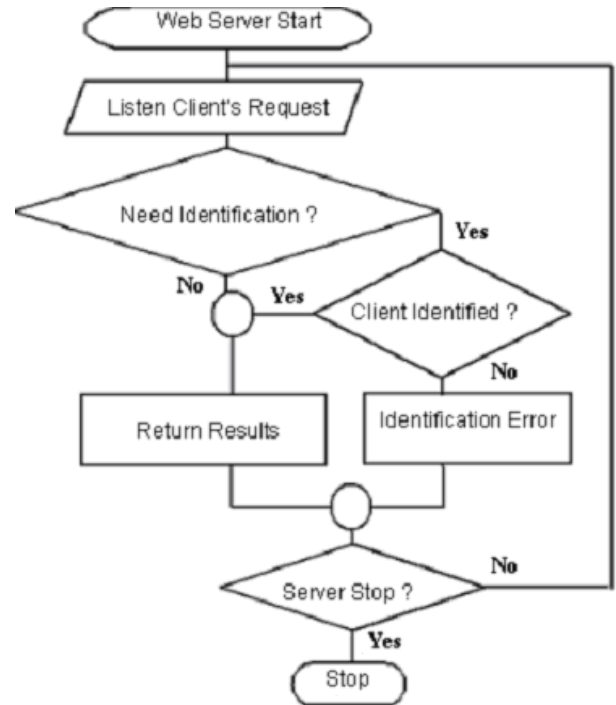


Fig. 4. Diagramme des tâches du serveur: démarrer serveur web, écouter requête client, identifier client, envoyer résultat, arrêter serveur

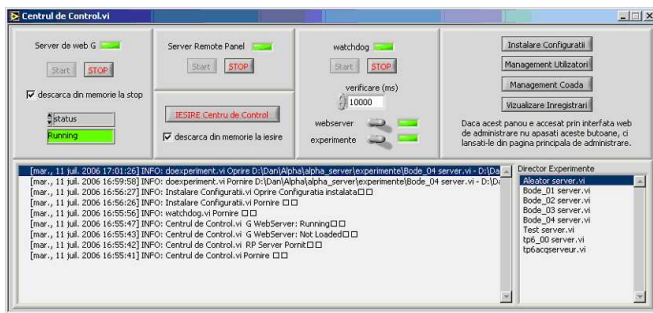
- installer la configuration des manipulations,
- gérer la liste des expériences connectées,
- gérer les utilisateurs (login, mot de passe),
- gérer la queue d'expériences en cours,
- visualiser les fichiers log, historique, statistiques d'utilisation,
- administration distante.

2.2. Description de l'Application Client

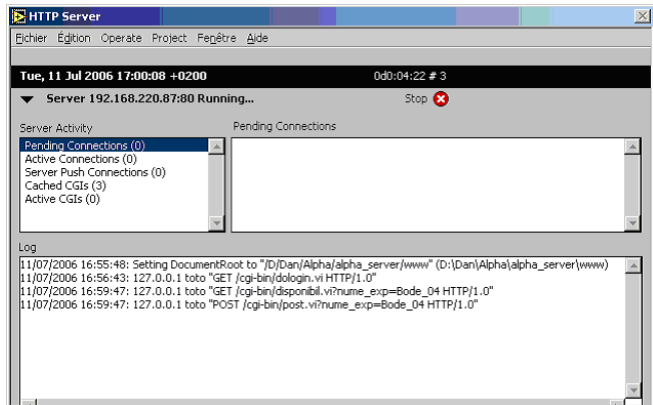
Un CDROM contenant tout le matériel pédagogique et la version runtime de l'application est fourni à chaque client (étudiant à domicile). L'application cliente ("Virtual Laboratory", voir Fig. 6) consiste en une interface IHM permettant la connexion et la sélection du mode de travail (simulation ou vraie expérience, en ligne ou hors ligne etc) et le réglage des paramètres.

Le diagramme d'activité de l'application cliente est montré à la Fig. 7.

Le client envoie au serveur ses choix relatifs à l'expérience, avec les paramètres de configuration (upload). Après exécution de l'expérience au niveau du serveur, le client distant télécharge les résultats de mesure (download). Ensuite



a)



b)

Fig. 5. Interface du centre de contrôle du côté serveur : a) A gauche: démarrage du serveur web; Au milieu: démarrage du watchdog ; A droite: configuration et gestion ; b) serveur HTTP en fonctionnement

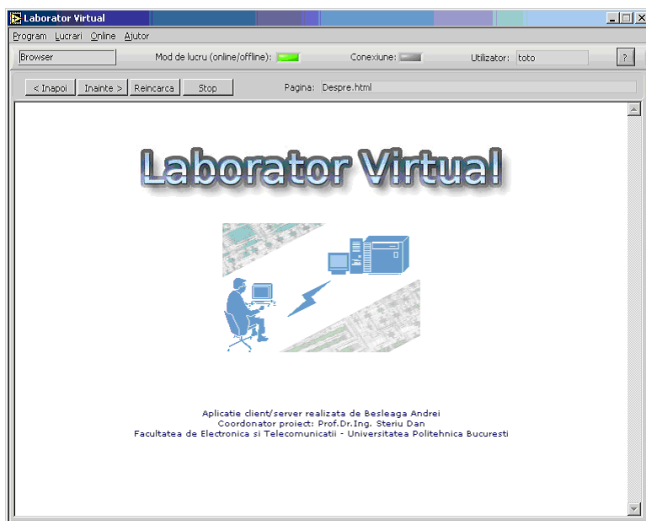


Fig. 6. Laboratoire Distant : Interface de l'application client

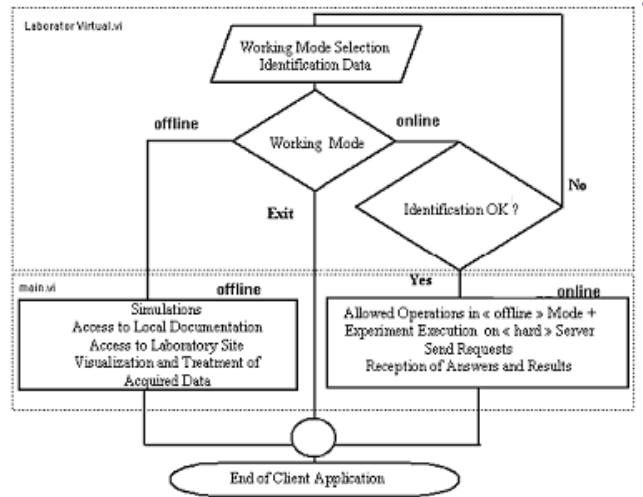


Fig. 7. Schéma des tâches de l'application client

l'étudiant peut faire l'analyse et l'interprétation des données expérimentales, en lien avec la théorie et les modèles appris, suivant les objectifs d'apprentissage fixés: tracé de courbes, validation de modèles théoriques, génération de compte-rendus qui seront évalués par l'enseignant.

2.3. Modules Plug-in Côté Serveur

Pour faciliter la programmation des modules d'extension du côté serveur, nous proposons une structure qui permet de travailler selon différents modes (sélectionnés par des commutateurs booléens):

- mode **Local** : les réglages pour la manipulation sont faits directement sur le serveur lui-même, cf. Fig. 8,
- mode **Client** : les réglages et paramétrages viennent alors d'un client distant (situation d'utilisation normale par un étudiant connecté sur le réseau), cf. Fig. 9.

Un commutateur permet la sélection du mode de fonctionnement côté serveur. En effet il est plus facile de développer, programmer et vérifier (débugger) un plug-in serveur en travaillant sans connexion client.

D'autre part, il est possible d'obtenir les résultats d'une vraie expérience (commutateur sur **Manip**) ou depuis un module de simulation (commutateur sur **Sim**).

Intr_Data_Local.vi est utilisé pour régler l'expérience localement, **Visu_result_Local.vi**, pour visualiser les résultats localement et **Manipulation.vi** pour commander les appareils.

La situation de travail normale (requête par le client → expérience réelle sur le serveur → résultats vers le client) est représentée sur la Fig. 9.

Il est également possible de répondre à la requête d'un client avec des résultats qui proviennent non pas d'une ma-

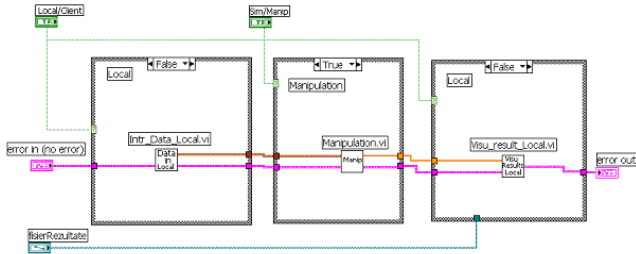


Fig. 8. Programme du plug-in serveur en phase de vérification d'une expérience (i.e. off-line, avant sa mise en service en ligne): ici Local/Client=False (i.e. mode Local) et Manip=True (i.e. une vraie expérience, et non pas une simulation)

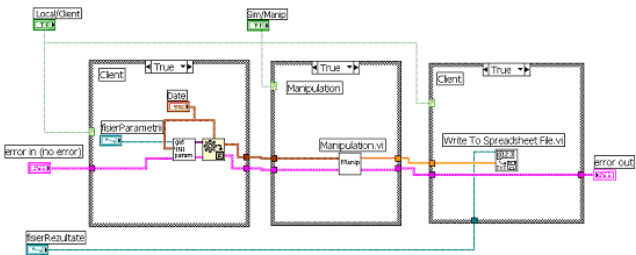


Fig. 9. Programme du module serveur en situation de travail normal, i.e. étudiant faisant ton TP à distance : commutateurs sur mode distant (Client=True) et mode Manipulation=True

nipulation mais d'une simulation, en utilisant des modèles mathématiques et physiques précis des appareils et systèmes à l'étude. Cette possibilité est très utile quand on doit opérer la maintenance des équipements de la manipulation: l'indisponibilité temporaire du matériel ne perturbe pas le client distant qui, au lieu de faire la vraie manipulation, fait une simulation à la place (notons qu'il peut même ne pas s'apercevoir de la différence).

3. IMPLANTATION DU PROTOTYPE

3.1. Logiciel LabVIEW

Le prototype est basé sur LabVIEW qui est un environnement de programmation graphique particulièrement dédié au domaine du test et de la mesure. Il permet une programmation aisée consistant à câbler ensemble des objets graphiques et une conception rapide d'interfaces utilisateur conviviales. L'emploi du langage graphique G autorise ainsi un cycle de prototypage rapide dans le développement d'applications [15].

3.2. Matériel GPIB

L'application serveur tourne sur un ordinateur PC avec système Windows, accès Internet, explorateur web et bus local GPIB (standard IEEE 488 pour l'instrumentation). Les appareils de mesure (générateur de signaux, alimentation programmable, oscilloscope, multimètre, carte d'acquisition de données avec E/S numériques et analogiques) et les matrices de commutateurs (multiplexage pour la configuration des connexions et la mise sous/hors tension des appareils) sont contrôlés et lancés à distance par l'étudiant connecté via internet au serveur qui est situé dans notre institut (Fig. 10).

4. EXEMPLE D'APPLICATION ET RÉSULTATS

4.1. Banc de Mesure: Serveur

Un prototype opérationnel multi-utilisateur et multi-expérience a été mis en œuvre et testé avec succès:

- réseau intranet de l'IUT de Bayonne,
- 5 à 30 connexions simultanées,
- modules d'extension validés pour différentes manipulations parmi lesquelles:
- l'acquisition de données avec traitement du signal (filtrage numérique RII) et,
- le tracé automatique des courbes de Bode (gain et phase) pour caractériser la fonction de transfert de systèmes linéaires.

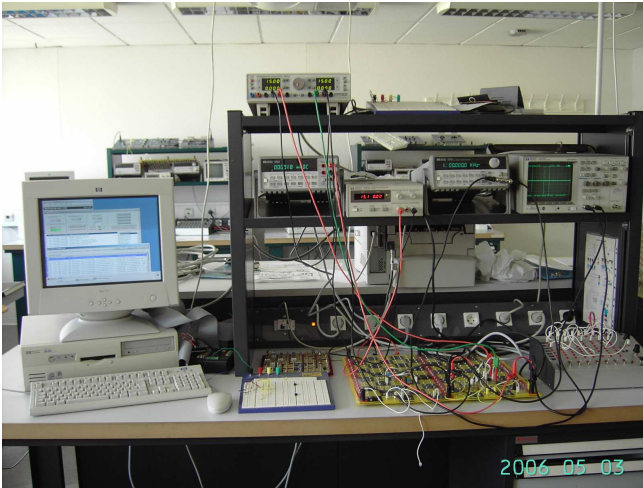


Fig. 10. Banc de mesure expérimental dans la salle serveur: PC serveur à gauche, instruments GPIB en haut et platine de connexions avec le composant sous test (DUT ou device under test) sur l'établi

Sur l'exemple présenté ici, il s'agit d'un filtre analogique réalisé par un circuit électronique comportant un AOP (ampli op.) et dont le schéma électrique est dessiné sur la Fig. 11.

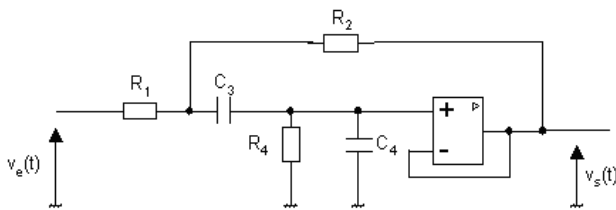


Fig. 11. Filtre actif de Sallen-Key à caractériser : filtre passe-bande du 2nd ordre (alimentation de l'AOP en $\pm 10VDC$)

Le banc de mesure pour le tracé des courbes de Bode est représenté Fig. 12.

4.2. Objectif Pédagogique

L'estimation de la phase repose sur l'utilisation d'un oscilloscope pour la mesure précise du décalage temporel entre les signaux d'entrée et de sortie du filtre. Notons qu'un tel instrument est bien plus complexe à programmer et à piloter qu'un simple multimètre. C'est une des raisons qui nous a poussés à mettre en œuvre cette manipulation, en plus du fait que l'étude de la fonction de transfert de filtres ou systèmes linéaires du 2nd ordre est d'une importance capitale pour les étudiants du domaine de l'EEA (Electricité, Electronique, Automatique) au niveau Bac+2 à Bac+4.

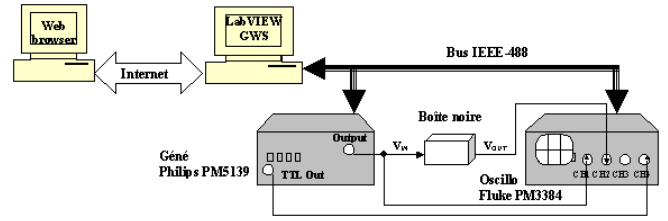
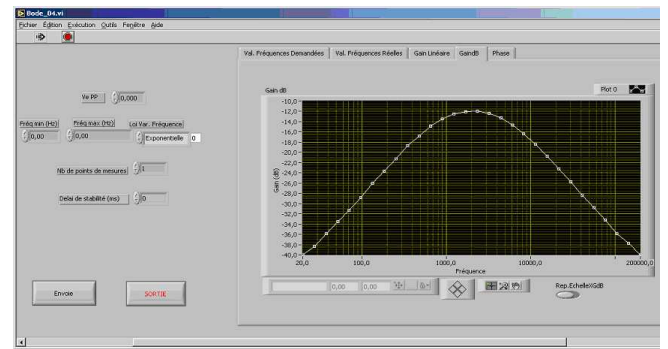


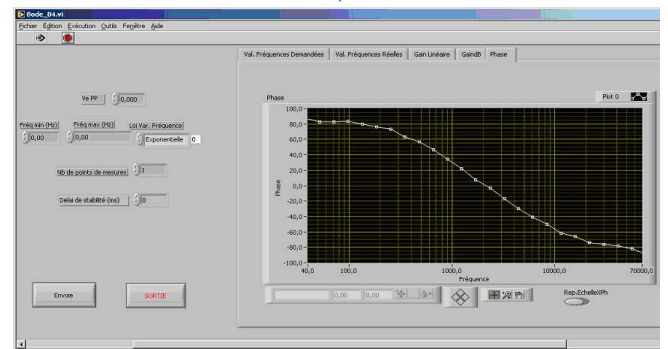
Fig. 12. Banc d'instruments pour le tracé des courbes de Bode (gain et phase)

4.3. Application Etudiant: Client

Après s'être connecté au serveur, l'étudiant choisit sa manipulation (e.g. tracé des courbes de Bode) et tous les réglages de paramètres (fréquences minimale f_{min} et maximale f_{max} , tension d'entrée V_{IN} , nombre de points de mesure N , mode d'affichage des graphiques de résultats (linéaire, dB, décade). Le chien de garde (watchdog) permet d'informer le client quand les données sont prêtes à être téléchargées (Fig. 13). Au vu des résultats, l'étudiant interprète les mesures et



a)



b)

Fig. 13. Téléchargement des résultats avec affichage des Courbes de Bode du côté client distant: a) Tracé du Gain en dB du filtre de Sallen-Key du 2nd ordre; b) Tracé des variations de la phase du filtre

redige un compte-rendu qui sera évalué par l'enseignant. Il déduit les caractéristiques essentielles du filtre, telles que sa

fréquence de résonance, sa bande passante à -3dB, sa marge de phase etc.

5. DISCUSSION ET PERSPECTIVE

5.1. Avantages

Cette architecture ouverte est extensible (grâce aux modules d'extension), et échelonnable (déploiement sur différents réseaux: LAN, WAN, etc). Elle permet la distribution libre de matériels et de contenus pédagogiques puisque n'importe quel enseignant-tuteur peut construire un nouveau module (plug-in) et l'ajouter dans la liste du menu des expériences.

Les avantages principaux d'une telle architecture sont les suivants: réduction des coûts financiers pour l'institution (on économise de l'argent car un seul exemplaire de chaque appareil coûteux dans le domaine de l'électronique ou de l'automatique peut être utilisé par de nombreux étudiants). Cela permet la mise en place d'un enseignement technique à distance qui englobe l'intégralité du processus d'apprentissage incluant non seulement les cours et les séances d'exercices, mais aussi les travaux pratiques, avec un accès possible aux manipulations 24 heures sur 24 et 7 jours sur 7.

Les autres avantages sont la rapidité des mesures, la réduction des tâches répétitives et des erreurs, ce qui laisse plus de temps à l'étudiant pour l'interprétation, l'accès non limité aux expériences, les démonstrations en ligne, le développement et la maintenance aisés par l'équipe technique, etc.

5.2. Perspectives

En plus de nos deux manipulations actuelles qui sont déjà disponibles dans le prototype (tracé de courbes de Bode et acquisition et filtrage numérique), nous avons l'intention d'inclure de nouvelles manipulations que nous avons déjà développées séparément [9, 14] :

- caractérisation d'un amplificateur audio BF de puissance (Fig. 14),
- caractérisation statique d'un transistor MOS (Fig. 15)

Actuellement, le prototype est validé d'un point de vue technique (vraies manipulations à distance et communication réseau) mais il nous reste encore à calibrer les objectifs d'apprentissage et à évaluer le prototype avec un plus grand nombre d'étudiants (pour établir les avantages et les inconvénients de l'expérimentation réelle par rapport à la simple simulation, ceci au regard des objectifs d'enseignement visés). Notre prochaine étape est de faire fonctionner ce prototype non seulement en Intranet, mais aussi sur Internet (connection entre l'IUT de Bayonne, France et l'Université Polytechnique de Bucarest, Roumanie).

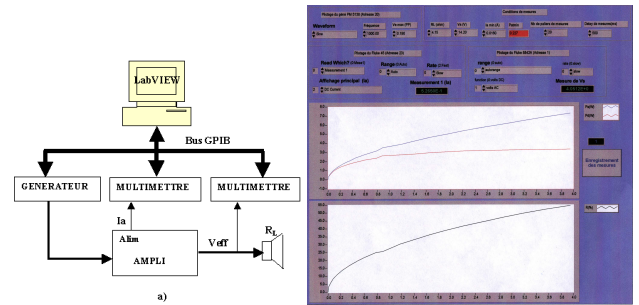


Fig. 14. Caractérisation d'un ampli audio: a) Banc de mesures ; b) Interface graphique utilisateur qui affiche les courbes de puissance et de rendement

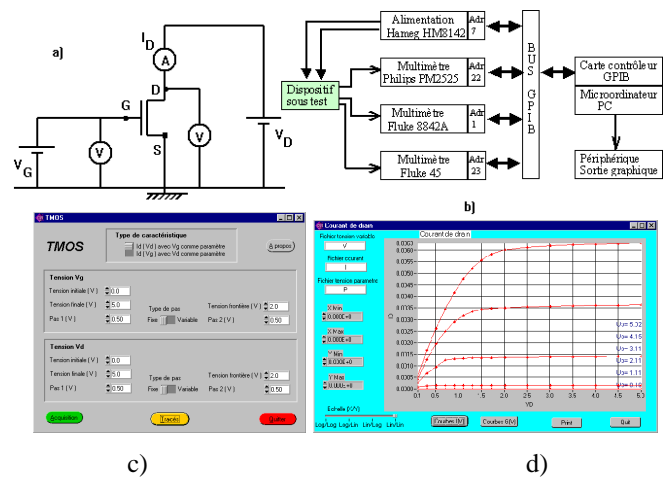


Fig. 15. Caractérisation d'un TMOS : a) Composant sous test ; b) Banc de mesure ; c) Panneau avant de réglage des paramètres ; d) Interface graphique utilisateur avec affichage des courbes du courant de drain en fonction de la tension

Pour les expériences nécessitant le temps-réel (par exemple besoin d'une vidéo en ligne montrant en temps-réel le phénomène physique en action ou en mouvement, e.g. cas de la programmation d'un robot mobile ou de l'asservissement d'un bras manipulateur), nous devons étudier quels choix devraient être changés dans l'architecture proposée qui repose actuellement sur le concept de téléchargement (upload et download), et donc sur le temps différé. Les points concernés sont notamment : contrôle Web avec CGI ou avec ActiveX ou avec Java ? DataSocket ou VI Server ou TCP VIs ? etc...

Nos travaux futurs porteront aussi sur le processus d'évaluation de ce type d'enseignement pratique à distance (formulaire d'évaluation de l'étudiant, quizz, approche constructiviste pour reproduire la notion de travail d'équipe ou en binôme à distance [6, 11], pratique de la pédagogie active [5] comme le jeu de piste, modélisation de l'interaction didactique et pédagogique [2, 10] etc), et sur l'emploi du nouveau standard rapide HS-488 à la place du standard GPIB classique. Une comparaison et un positionnement par rapport à d'autres méthodes et travaux sur les TP distants tels que [1,3] et d'autres architectures [7,8] est également en cours d'investigation¹.

6. REMERCIEMENTS

Les auteurs remercient Thierry Nodenot et Philippe Lopistéguy de l'Université de Pau, Bayonne, France, pour les discussions fructueuses et certaines références bibliographiques.

Ce travail a été supporté financièrement par l'Université de Pau, grâce à l'attribution d'un séjour de 3 mois de Professeur Invité au laboratoire d'Informatique LIUPPA, qui a permis l'accueil sur le site de Bayonne du Professeur Mihai-Dan Steriu de l'Université Polytechnique de Bucarest, Dept. Electronic Circuits & Devices, Roumanie.

7. REFERENCES

- [1] Bonivento, C., Gentili, L., Marconi, L., and Rappini, L. A web-based laboratory for control engineering education. In *Proceedings of 2nd International Workshop on Tele-education using Virtual Laboratories*. Sherbrooke, Canada, 2002.
- [2] Bouhadada, T., Laskri, M., Ghrib, T., and Chouaf, A. SAPIENCE: A simulator of interactive pedagogical activities for distance learning environment. In *Proc. Conf. INFORSID*. Hermès Science Pub., Hammamet, Tunisie, 2006, 1–16.
- [3] Chiculita, C. and Frangu, L. A web remote control laboratory. In *Proceedings of 6th Multiconference on Systemic, Cybernetics and Informatics*. Orlando, Florida, 2002.
- [4] Feisel, L. D. and Rosa, A. J. The role of the laboratory in undergraduate engineering education. *Journal of Engineering Education* (2005), 121–130.
- [5] Felder, R. and Brent, R. Active learning. In *IEEE DSP/Signal Proc. Education Workshop, USA*. Jackson Lake Lodge, Wyoming, 2006, 1–21.
- [6] Ferreira, J. and Mueller, D. The MARVEL EU project: A social constructivist approach to remote experimentation (2004). [Http://www.marvel.uni-bremen.de/](http://www.marvel.uni-bremen.de/).
- [7] Garcia-Zubia, J., de Ipiña, D. L., and Orduña, P. Evolving towards better architectures for remote laboratories: a practical case. *International Journal of Online Engineering (iJOE)*, 1, 2.
- [8] Garcia-Zubia, J., de Ipiña, D. L., and Orduña, P. Towards a canonical software architecture for multi-device weblabs. In *IECON*. 2005.
- [9] Luthon, F. Instrumentation programmée pour TP d'électronique distant. In *Actes de la Conférence FuturVIEW 2003*. Futuroscope, Poitiers, 2003, 263–268.
- [10] Masseux, N. Modélisation d'une interaction didactique distante individuelle synchrone (IDIS). *Revue Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (sticef.org)*, 10.
- [11] Müller, D. and Ferreira, J. M. Online labs and the MARVEL experience. *Int. J. Online Engineering*. [Http://www.i-joe.org](http://www.i-joe.org).
- [12] Spanias, A. J-DSP Java Digital Signal Processing and On-line Laboratory (2006). [Http://www.eas.asu.edu/~middle/jdsp/](http://www.eas.asu.edu/~middle/jdsp/).
- [13] Spanias, A. and Atti, V. Interactive online undergraduate laboratories using J-DSP. *IEEE Trans. Education*, 48, 4 (2005), 735–.
- [14] Steriu, M. and Luthon, F. Open architecture for signal processing lab distance learning. In *IEEE DSP Education Workshop*. Jackson Lake Lodge, Wyoming, USA, 2006.
- [15] Travis, J. *Distance Learning Solutions Guide*. Technical report, LabVIEW National Instruments (2000). [Http://ni.com/academic/distance_learning.htm](http://ni.com/academic/distance_learning.htm).

¹Une enquête réalisée en 2006 à l'Université Technologique de Sydney, Australie, et portant sur 25.000 étudiants, indique que 75% d'entre eux considèrent l'enseignement à distance simplement comme un complément des cours en présentiel, mais pas comme un substitut.