

Interfaces Client Serveur Generiques Pour Travaux Pratiques Distants En Automatique

Résumé : Cette réalisation digitale s'inscrit dans le domaine des FAD (Formations À Distance) relative au développement et le fonctionnement d'une architecture générique et globale (matérielle et logicielle) permettant l'accès à distance aux instrumentations de laboratoire d'automatique. Cette méthode constitue un challenge et une contribution à la résolution de la problématique générale de la mise en ligne de travaux pratiques dans le domaine scientifique et technique. Dans notre cas, c'est la discipline de l'Automatique qui a suscité notre intérêt discipline transversale dans l'enseignement technique. La finalité du travail développé ici est de rendre accessible aux apprenants via le réseau Internet les équipements des laboratoires de Travaux Pratiques (TPs). L'idée de base a conduit à deux interfaces graphiques de type Homme-Machine testées individuellement avec succès dans différentes conditions notamment dans un réseau LAN et WAN. Ces applications peuvent être intégrées dans une plate-forme d'enseignement à distance comme un module d'enseignement complémentaire aux modules traditionnelles de type cours et TD interactifs dans le domaine de l'Automatique.

Mots clés : Client/Server, Réseau Internet, TCP/IP, Matlab, PID, GPC.

1. Motivation

Au cours de ces dernières années, le monde a connu de vastes transformations dans les systèmes technologiques du traitement de l'information et de la communication appelées communément les TIC (Technologie de l'Information et de la Communication). En effet, depuis leur apparition, on assiste à une révolution dans le domaine de l'enseignement et de la formation à distance. Les Technologies de l'Information et de la Communication dans l'Enseignement (TICE) recouvrent les outils et produits numériques pouvant être utilisés dans le cadre de l'enseignement, et regroupent un ensemble de moyens matériels et logiciels conçus et utilisés à des fins d'enseignement et de formation à distance. La stratégie de mettre en place des laboratoires d'expérimentation distants n'est plus à démontrer de nos jours, aussi bien pour le formateur que pour l'apprenant. Divers travaux existent dans le domaine de la formation à distance, mais sont peu nombreux à proposer des plateformes de formation en ligne intégrant des travaux pratiques. Bien que ces environnements de formation favorisent l'atteinte de plusieurs objectifs pédagogiques, la formation à caractère pratique, c'est-à-dire la réalisation effective des expériences de laboratoire à distance à travers le réseau internet n'est pas encore très répandue. Ce dernier point constitue un véritable handicap dans nombreux domaines, notamment en sciences expérimentales, où la formation pratique est complémentaire à tout enseignement théorique. Par ailleurs, le contrôle et la manipulation à distance en temps réel et une synchronisation de communication entre le serveur et le client représente actuellement un défi et une valeur ajoutée pour les plateformes d'expérimentation à distance. D'où un choix approprié pour le choix du logiciel de développement et le protocole de communications entre deux ordinateurs distants.

Les points qui ont motivé le développement de notre travail s'appuient sur les avantages qu'apporte la mise en place des plateformes à distance d'enseignement pratiques :

- Eviter la duplicité du matériel pédagogique surtout si son coût est trop élevé ;
- Eviter le déplacement du matériel d'expérimentation d'un laboratoire à un autre ;
- Simplifier la situation expérimentale puisque le temps alloué pour réaliser une séance de travaux pratiques à distance peut être modelée dans le temps en fonction de l'habileté des apprenants à progresser plus moins rapidement ;
- Eviter d'encourir à l'expérimentateur des risques corporels en cas de fausse manœuvre puisqu'il opère à distance ;
- Offrir à l'apprenant la possibilité d'archiver les résultats du travail accompli directement sur son ordinateur. Une analyse en temps différé de ces résultats peut être alors accomplie ;
- Si l'exploitation est suffisamment complexe, l'apprenant dispose d'une liberté d'investigation qui ne lui est que très rarement octroyée au cours de travaux pratiques en présentiel. En ce sens, Il peut envisager lui-même une organisation expérimentale : procéder par tâtonnements, recommencer un grand nombre de fois des tentatives similaires tant qu'il dispose de l'accès au laboratoire distant.

Notre travail s'inscrit dans le cadre du développement des outils pour la formation à distance à caractère expérimental. Nous proposons une plateforme générique de type laboratoire virtuel qui va déboucher sur deux interfaces graphiques pour la manipulation à distance dans la discipline de l'Automatique en se focalisant sur les problématiques sous-jacentes suivantes :

- Comment mettre en ligne des activités d'enseignement de type travaux pratiques tout en atteignant l'efficacité pédagogique des travaux pratiques en présentiel, voire plus ?
- Quel est le protocole de communication qui garantit une transmission fiable c'est-à-dire sans perte d'information entre le client (Apprenant) et le serveur (Formateur) ?
- Comment exploiter pleinement en temps réel le potentiel de calcul numérique par le logiciel Matlab ?

- Comment la solution appliquée à la discipline de l'Automatique peut être étendue le plus facilement possible à d'autres disciplines dans le domaine scientifique et technique ?

2. Architecture logicielle et matérielle d'accès au laboratoire distant d'automatique

Afin de remplir les objectifs cités précédemment, nous avons développé deux prototypes de travaux pratiques à distance pour l'enseignement. Ces prototypes sont basés sur le développement des interfaces graphiques Homme Machine élaborés à l'aide du logiciel MATLAB. L'architecture logicielle / matérielle est représentée par la Fig.1.

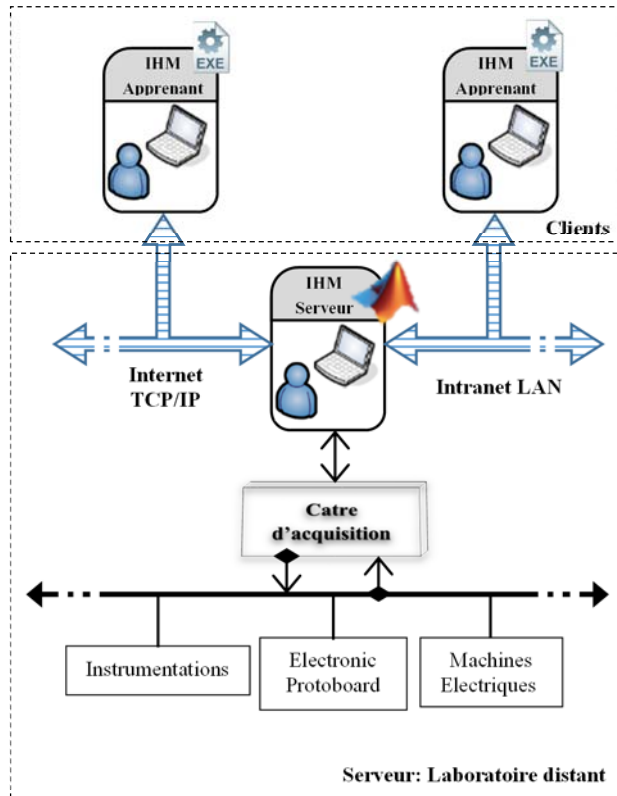


Fig. 1. Schéma synoptique de l'Architecture Client/Serveur du laboratoire distant.

Il s'agit d'une architecture de type client/serveur. Le serveur assure en temps réel la réception des ordres en provenance de client, pilote l'instrumentation et renvoie les résultats au client. Le client, c'est-à-dire l'apprenant effectue l'expérimentation comme en présentiel via les interfaces graphiques. Les fonctionnalités de ces interfaces seront décrites dans la section ?. Le serveur et les clients se connectent à cette architecture via le réseau Intranet/Internet en utilisant le protocole TCP/IP.

2.1 Logiciel d'implémentation et de supervision

L'outil informatique choisi pour le développement des prototypes d'expérimentation à distance est logiciel MATLAB/Simulink. Ce choix est dicté par le fait que Matlab dispose d'une grande capacité calculatoire pour traiter les problèmes liés à la discipline d'Automatique. Grâce à son outil le GUI, il est possible de concevoir des Interfaces Homme Machines (IHM) permettant d'assurer une supervision conviviale pour le pilotage du laboratoire distant côté client et un suivi et l'exécution des tâches côté serveur. L'intégration du Toolbox (Instrument Control Toolbox) permet une communication entre ordinateurs et appareils par la disponibilité du protocole TCP/IP.

Les interfaces graphiques développées pour l'application se base sur l'utilisation de l'outil graphique de Matlab GUI (User Interface graphique). La figure Fig.2

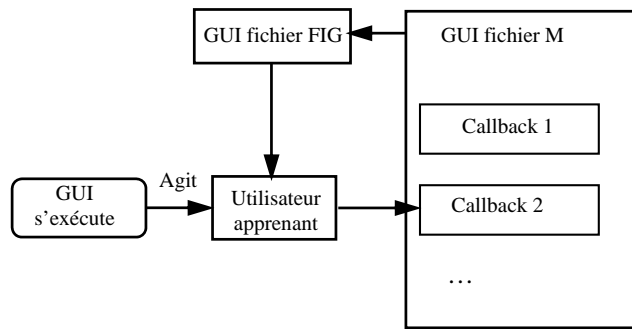


Fig. 2. Gestion de l'interface graphique GUI sous MATLAB.

Cet outil permet de développer des interfaces graphiques riches en fonctionnalité répondant à nos besoins en matière d'observation et de contrôle lors de la phase d'expérimentation.

Il est à noter que les interfaces graphiques client seront générées par des fichiers exécutables qui lui seront fournis au préalable. L'idée est d'exploiter à cet effet l'outil « MATLAB Compiler » qui permet de partager des applications telles qu'un exécutable ou une librairie. Ces interfaces sont conçues de manière qu'elles soient exploitées sur l'ordinateur client ne disposant pas du logiciel Matlab et sans prérequis de celui-ci de la part de l'apprenant. En effet, c'est une volonté pédagogique pour que l'apprenant se focalise sur les objectifs des travaux pratiques plutôt que le logiciel Matlab lui-même. Ces fichiers exécutables sont fournis individuellement aux utilisateurs ou télécharger à partir d'une plateforme d'enseignement à distance.

2.2 Protocole TCP/IP en Matlab

Le protocole TCP/IP pour la communication entre ordinateurs distants est implémenté dans Matlab. Il permettra d'assurer la transmission et la réception des différents types de données à travers le réseau Internet. La raison principale du choix de ce modèle de communication est la fiabilité qu'il assurera au niveau de l'échange des données entre l'application client et l'application Serveur.

La première tâche que doivent assurer l'application client et l'application Serveur est l'établissement d'un canal de communication via le réseau Internet/Intranet.

Afin de garantir des travaux pratiques à distance en temps réel, l'émission et la réception des données entre le serveur et le client seront cadencées de manière séquentielle selon un pas de **timeout** égale 1µs. MATLAB offre seulement la possibilité d'échanger des informations entre deux ordinateurs distants, ce qui impose de formuler une procédure spécifique permettant une synchronisation de communication entre le client et le serveur et ce sans pertes des informations échangées. La procédure proposée assurera en outre un échange de données en temps réel ce qui donne une dimension de la manipulation comme le déroulement en présentiel.

2.3 Matériel d'expérimentation et d'instrumentation

Notre architecture repose sur la disponibilité d'une carte d'acquisition et d'un procédé accessible par son entrée/sortie indépendamment de sa nature. L'équipement matériel utilisé pour réaliser les travaux pratiques à distance est donné par la fig. 3 suivante :

Le banc d'essai expérimental est constitué principalement des éléments suivants :



Fig. 3. Banc de mesure expérimental côté serveur à la salle des travaux.

- d'une maquette pour le contrôle de la température.
- D'une Carte d'acquisition pour l'acquisition des signaux de mesure provenance du procédé continu et l'émission des commandes. Cette carte doit être impérativement exploitable dans l'environnement de MATLAB. Dans notre cas nous avons utilisé une carte disposant plusieurs entrées/sortie analogiques et numériques reliée au serveur via le port USB.

Il est à noter que dans l'architecture générale, le principe du système de communication Serveur-Procédé est indépendant de la nature de la carte utilisée. De même, cette architecture est indépendante de la nature du procédé à contrôler du fait qu'il est supposé accessible par son entrée/sortie. La maquette de température utilisée ici peut être remplacée par tout procédé de laboratoire comme il est illustré par la Fig.4 représentant la synoptique du système de communication Serveur-Procédé.

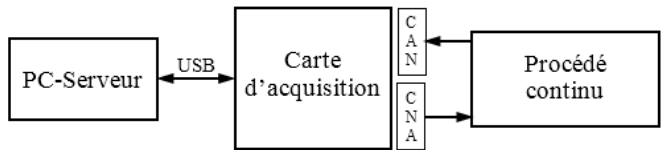


Fig.4. Synoptique du système de communication Serveur-Procédé

3. Partie Programmation

3.1 Description de l'application serveur

Le PC-Serveur est le cœur de l'application au niveau du service fourni au Client, il est équipé par un programme M-file développé dans la plateforme MATLAB. L'organigramme ci-dessus (fig. 8) traduit bien le principe du fonctionnement de nos applications ainsi que les différentes tâches du serveur.

Le programme au niveau de serveur est divisé en deux grandes parties : la première est de caractère invariable, elle est conçue à établir la connexion entre le client et le serveur, ainsi que la transmission, l'acquisition, traitement, l'enregistrement et l'affichage de différentes données. Alors que la seconde partie du programme serveur **Module générique** de la figure 5 offre à l'enseignant-responsable du TP la possibilité de rajouter sans limite des travaux pratiques différents et d'insérer facilement de nouvelles expériences dans le point de vue de répondre à des besoins rencontrés dans l'opération éducative. Pour cette raison, on dit que ces applications développées sont à caractères génériques et ouvertes.

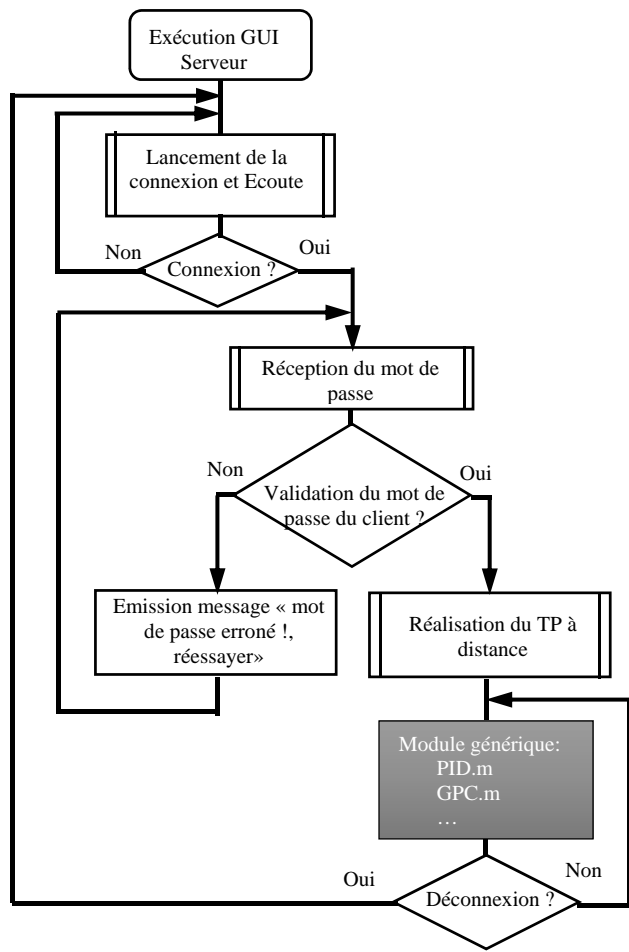


Fig. 5. Organigramme des tâches implémentées au niveau du PC-Serveur.

3.2. Description de l'application client

Les interfaces clientes développées sont sous forme des applications exécutables d'extensions .exe. Etant donné que l'utilisateur n'est pas sensé de disposer le logiciel MATLAB sur son ordinateur et aucune connaissance de ces logiciels n'est requise à l'apprenant Client. Le diagramme d'activité de l'application cliente est présenté ci-dessous (Fig.6).

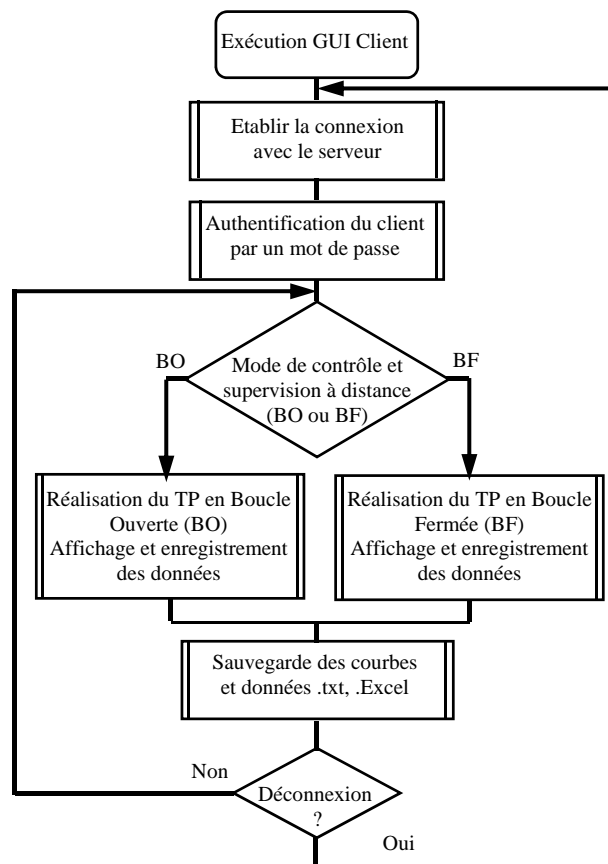


Fig. 6. Organigramme de différentes tâches implémenté au niveau du Client.

L'utilisateur apprenant dispose des programmes exécutables et d'une documentation nécessaire pour leur installation sur son ordinateur personnel. Ces programmes peuvent être fournis dans un support matériel (Cd-Room, Clé USB) ou téléchargés à partir d'une plateforme de formation à distance. L'utilisateur apprenant est demandé d'installer l'application des TP à distance en deux étapes (figure 7) :

- (i) la première étape consiste à installer « MCRInstaller » (1) : MCR (MATLAB Compiler Runtime), moteur de calcul qui fournit par MATLAB pour la distribution des applications exécutables sous utilisé matlab lui-même.
- (ii) Après l'installation du « MCRInstaller », on ouvre l'application cliente développée 'e-PIDMAT.exe' (2).

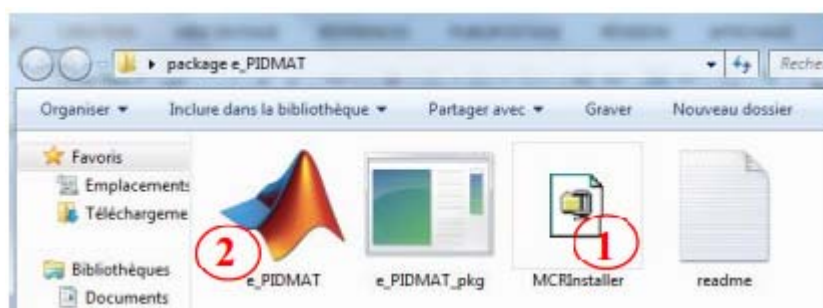


Fig. 7. Dossier d'installation de l'application e-PIDMAT.

L'exécutable e-PIDMAT.exe ou e-GPCMAT.exe, une fois installé sur le PC client, génère une interface graphique à caractère IHM pour effectuer la manipulation à distance. L'utilisateur envoie au serveur ses choix relatifs à l'expérience avec des paramètres de configuration. Suivant un scénario caractérise le déroulement du TP et à partir des objectifs d'apprentissage déjà fixés, l'apprenant peut donc faire : (i) Le traitement et la visualisation en temps réel des résultats expérimentaux de la régulation en modifiant ses

paramètres selon convenance. (ii) L'interprétation des résultats expérimentaux obtenus (validation des modèles théoriques) (iii) Le téléchargement de différents résultats de mesure et la rédaction du compte-rendu qui seront évalués par l'enseignant.

4. Les interfaces graphiques e-PIDMAT.exe et e-GPCMAT.exe

4.1. Présentation générale

Les techniques de régulation des procédés des systèmes sont certes nombreuses, mais nous avons retenu dans le cadre de ce travail deux techniques connues par leur efficacité et leur notoriété, à savoir, la commande PID implémenté dans l'application nommée e-PIDMAT et la commande GPC implémenté dans l'application nommée e-GPCMAT.

Les interfaces permettent d'étudier les systèmes dynamiques continus et le paramétrage d'un régulateur PID/GPC en vue de l'asservissement et de la régulation à distance et en temps réel. Elles doit notamment permettre à l'utilisateur d'afficher, de modifier, de comparer et d'exporter les mesures réalisées. Et ainsi de tracer les commandes ainsi les réponses mesurées du procédé commandé. La manipulation à distance du TP par l'utilisation de ces interfaces permettant à l'apprenant client de se familiariser, on développe son savoir dans le domaine de l'automatique en abordant quelques aspects de celui-ci (identification, contrôle, correction,...).

Selon les objectifs pédagogiques établis par l'enseignant administrateur du TP, les interfaces développées permettent d'assurer le déroulement de ces objectifs. On rappelle que les fonctionnalités de l'interfaces permet en d'autres de :

- Exploiter le bloc **commande en boucle ouverte** pour acquérir la réponse du procédé à une commande constante ou variable. Ce qui permet une analyse des caractéristiques statique et dynamique, ainsi l'identification d'un modèle pour le procédé.
- Exploiter le bloc **commande en boucle fermée** pour acquérir la réponse en boucle fermée selon le choix des paramètres du régulateur PID (e-PIDMAT) ou GPC (e-GPCMAT). Cela se fait selon une méthodologie spécifiée par l'enseignant. On précise que dans les deux fonctionnements, l'apprenant a la possibilité d'enregistre les signaux pour une exploitation ultérieures.

4.2. Application e-PIDMAT

Ci-dessous, un aperçu de la fenêtre principale de l'interface graphique pour le pilotage du procédé continu.

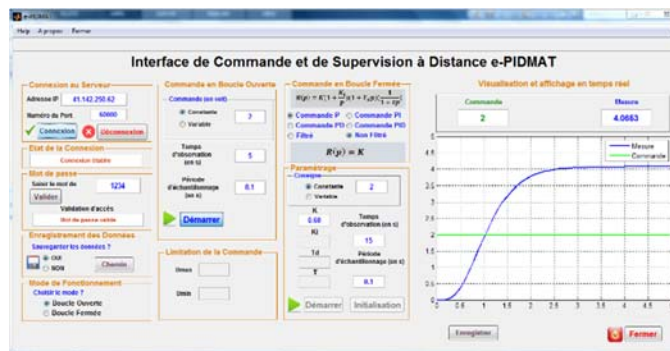


Fig. 8. Interface de l'application client, L'interface assure une connexion avec le PC-Serveur en spécifiant l'adresse IP du serveur et le port de communication.

L'application permet l'affichage de différentes réponses du système de commande en boucle ouverte et en boucle fermée. L'application est décomposée principalement de plusieurs blocs : (i) Bloc de connexion au serveur : Indiquant son adresse IP et le port de communication commun. (ii) authentification de l'utilisateur apprenant par un mot de passe donnée au préalable. (iii) choix de l'enregistrement ou non des réponses entrée/sortie. (iv) limitation de la commande générée par le régulateur PID. (v) le choix du régulateur P, PI et PID avec ou sans filtre passe bas, ainsi les paramètres des actions retenues, le régulateur implémenté est un PID filtré analogique de structure série Equation (1). (vi) l'affichage en temps réel des entrée/sortie.

$$R(p) = K \left(1 + \frac{K_i}{p} \right) (1 + T_d p) \left(\frac{1}{1 + \tau p} \right) \quad (1)$$

Les interfaces graphiques disposent un Help afin que l'utilisateur puisse à tout moment le consulter pour connaître leurs différentes fonctionnalités. L'enseignant ou le responsable du TP possèdent l'outil de superviser en temps réel afin qu'il puisse aussi à suivre le

déroulement du TP effectué par l'apprenant. La figure ci-dessous présente un exemple de l'interface serveur dans lequel la connexion et l'authentification sont valides et que l'apprenant a effectué un test sur le procédé.

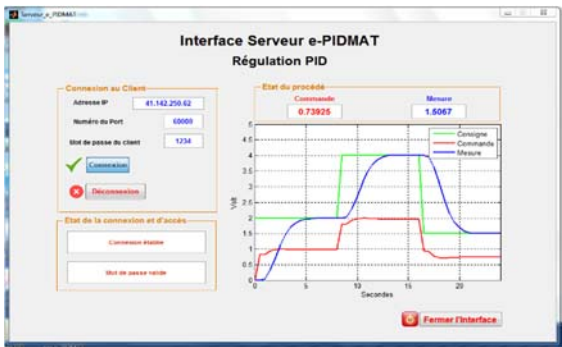


Fig. 9. Interface serveur e-PIDMAT de supervision à

4.3. Application e-GPCMAT

L'application e-GPCMAT permet une saisie facile de différentes grandeurs utiles de la commande du procédé continu par GPC. Sa structure est semblable à l'interface e-PIDMAT à l'exception du bloc relatif à la commande en boucle fermée. En effet, la loi de commande de type PID dans l'interface e-PIDMAT alors qu'il est ici de type GPC.

L'utilisation de la commande prédictive technique est particulièrement intéressante lorsque les systèmes possèdent des retards importants, des réponses inverses et de nombreuses perturbations. La mise en œuvre de la GPC s'effectue à partir du modèle représenté sous la forme CARIMA (Controlled Auto Regressive Integrated Moving Average) :

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t-1) + \frac{e(t)}{\Delta(q^{-1})} \tag{2}$$

La figure précédente illustre la réponse d'un système à une consigne variable dans laquelle, on notera le caractère anticipatif de cette réponse.

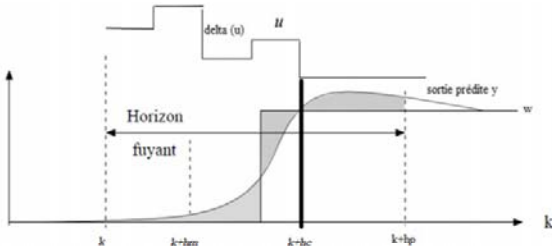


Fig. 10. Effet anticipatif de la commande prédictive.

La figure Fig. 11 est une capture d'écran de l'interface générée par l'interface client principale. Elle permet de saisir les paramètres du modèle discret du procédé à commander à distance. On précise que ces paramètres peut résulter d'une identification au préalable du procédé par l'utilisation de cet application ou fournis par l'enseignant via le manuel du TP.

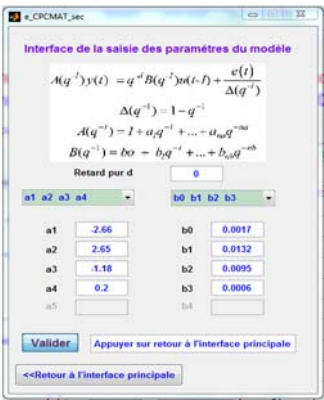


Figure 11. L'interface secondaire de la saisie des paramètres du modèle.

Quant au bloc **commande en boucle fermée**, l'apprenant doit saisir les paramètres du régulateur GPC et l'emplacement de la consigne variable. Les deux figures ci-dessous (Fig. 12) représentent respectivement les résultats d'un essai côté serveur et côté client dans lesquelles la connexion au serveur et l'authentification du client sont valides et que l'apprenant a effectué un test sur le procédé à distance.

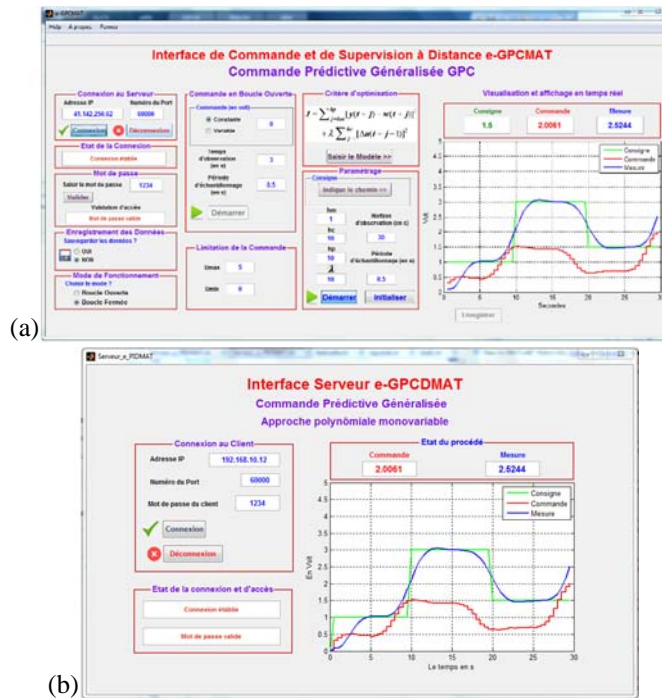


Fig. 12. Essai de commande à distance : a) L'application client e-GPCMAT, b) interface serveur de supervision à distance correspondante à cet essai.

4.4. Un scénario pédagogique

L'idée est de créer grâce au réseau informatique des TP délocalisés, constitués d'un apprenant muni de l'application exécutable sur son ordinateur, et une salle distante de travaux pratiques. Dans notre cas, c'est le laboratoire d'automatique de l'ENSET de Rabat. L'apprenant délocalisé peut être assisté ou non par un enseignant au niveau PC-Serveur.

Donc un apprenant peut à travers l'application cliente se connecter et récupérer en temps réel les données du poste serveur et donc réellement faire un TP mais à distance. Il peut donc visualiser en temps les résultats expérimentaux du régulateur en modifiant selon sa convenance.

Pour ce faire, l'utilisateur apprenant est demandé de passer par les cinq phases suivantes : (i) Téléchargement du manuel d'utilisation du TP choisi ainsi que l'application cliente correspondante. Ceci se fait soit à travers une plateforme d'enseignement (Cours, TD et TP), ou soit à travers une procédure spécifique à la réalisation du TP. (ii) installation de l'application Client. (iii) préparation du TP à partir d'une documentation fournie par l'administrateur serveur partie théorique. (iv) l'expérimentation à distance et la remise des données. (v) remise du compte rendu à partir des moyens précédents.

5. Conclusion

Dans ce papier, nous nous sommes intéressés à la conception et la réalisation de deux prototypes permettant de rendre accessible aux apprenants via le réseau Internet les équipements des laboratoires de Travaux Pratiques (TPs). Dans notre cas, c'est la discipline de l'Automatique qui a suscité notre intérêt.

Nous avons adopté dans le cadre de ce travail le protocole de communication TCP/IP comme un moyen de connecter à distance un PC-Serveur et un PC-Client dans une architecture client-serveur. Ces applications fournissent une accessibilité conviviale pour la supervision et le contrôle à distance de différents procédés de laboratoire.

Pratiquement tous les aspects relatifs à l'automatique sont pris en charge par ces interfaces dans le but d'effectuer des TP distants, ainsi que les fonctionnalités de ces interfaces sont à notre sens largement suffisantes pour substituer un TP en présentiel.

Les contributions apportées par ce travail peuvent être résumées dans les points suivants :

- Les interfaces client permettent un contrôle et une supervision totale en temps réel.
- Pas besoin d'un matériel de TP spécifique commercial adapté aux TPs à distance à l'exception d'une carte d'acquisition et deux ordinateurs dont un joue le rôle du PC-Serveur et l'autre du PC-Client.
- Les interfaces génériques sont indépendantes de la nature du procédé à commander, du moment qu'il est supposé accessible physiquement par son entrée/sortie.

- Comme leurs noms indiquent, les interfaces sont génériques et peuvent servir à la génération d'autres applications Client. Dans notre cas on est passé facilement de l'interface PID à l'interface GPC en changeant uniquement la loi de commande et ses paramètres.

Les applications ne nécessitent pas l'installation des logiciels de développement (MATLAB/Simulink) sur le PC-Client, du fait que ces applications sont installées sur le PC Client à partir des exécutables.

Les applications fonctionnent aussi bien en Intranet (réseau local) qu'en Internet (En dehors de réseau local. Ce dernier point donne bien entendu une dimension très large à l'accessibilité aux équipements distants.

Nous terminons enfin cette conclusion en donnant quelques perspectives à ce travail qui, par faute de temps, nous n'avons pas pu les explorer suffisamment afin de les présenter ici :

- Pour évaluer les performances des systèmes de régulation proposés, il est intéressant de donner les moyens à l'apprenant Client la possibilité d'injecter des perturbations et donc d'agir en définitive à distance sur le procédé.

- Pour établir une connexion internet entre le serveur et le Client, l'adresse publique de celui-ci doit être connue en avance par le serveur. Dans notre conception actuelle, cette adresse est supposée fournie par le PC client. Il est intéressant, voire impérative, de concevoir un moyen pour contourner cette contrainte. Nous pensons qu'une connexion avec l'outil Proxy peut être explorée.

- Tester ces interfaces dans leur version actuelle ou améliorée dans une plate-forme d'enseignement à distance disposant déjà des modules de type Cour et Travaux Dirigés.