PLATEFORME POUR L'ETUDE ET LA PROGRAMMATION DES ASSERVISSEMENTS ECHANTILLONNES

Eric DUQUENOY
IUP du Littoral
Université du Littoral Côte d'Opale
40 rue F.Buisson - BP649
62228 Calais - FRANCE
mailto:eric.duquenoy@iup-calais.univ-littoral.fr
http://eric.duquenoy.free.fr

Plan Général

- Cadre du projet
- Objectifs pédagogiques
- Solutions existantes
- Contraintes de développement
- Solution proposee
- Structure de l'application
- Résultats
- Performances
- Conclusion

CADRE DU PROJET

- **CADRE** : enseignement de l'**automatique linéaire** en licence *(GSI[1] en IUP[2] et EEA[3] à l'ULCO[4])*,
- **PROJET** :
  - application ouverte,
  - implantation rapide des algorithmes de régulation écrits en langage C
  - s'affranchir des contraintes de développement graphique sous Windows

[1] Génie des Systèmes Industriels

[2] `http://iupcalais.univ-littoral.fr`

[3] Electronique Electrotechnique Automatique

[4] `http://www.univ-littoral.fr`

OBJECTIFS PEDAGOGIQUES

- **Mise en œuvre** des principaux régulateurs propres aux asservissements linéaires échantillonnés :
  - correcteurs **industriels** classiques (proportionnel – intégral – dérivé)
  - correcteurs **adaptés** (réponse pile, placement de pôles)
- **Implantation** dans un langage évolué (ici, le langage c)

SOLUTIONS EXISTANTES

- **Solutions généralistes** (Simulink[1], Labview[2]...) **:**
  - Temps de prise en main important
  - Coût élevé
  - Nécessitent des cartes d'entrées-sorties propriétaires et coûteuses.

- **Solution spécialisée** (Acsyde[3]) **:**
  - prise en main rapide.
  - compatible avec un nombre important de cartes d'acquisition
  - **Mais ne permet pas la programmation en langage évolué**.

[1]Mathworks : http://www.mathworks.fr
[2]National Intruments : http://www.ni.com
[3]Ipsis : http://www.ipsis.com

CONTRAINTES DE DÉVELOPPEMENT (1)

- Utilisation d'un **langage connu** de l'étudiant et courant dans l'industrie.

- Langage compilé pour des problèmes de rapidité,
- **Adéquation** du logiciel avec le matériel
- **Faible encombrement** mémoire de l'application

CONTRAINTES DE DÉVELOPPEMENT (2)

- **Maîtrise** des différentes facettes de la régulation :
  - choix de la fréquence **d'échantillonnage**,
  - élaboration de la grandeur **d'erreur**,
  - génération de **consignes** en échelon ou en rampe,
  - utilisation d'une consigne **externe**,
  - problèmes liés à la **conversion** analogique-numérique :
    - résolution des convertisseurs
    - débordement numérique,...
  - implantation de **régulateurs** divers :
    - PID,
    - réponse pile,
    - tout-ou-rien, etc...

CONTRAINTES DE DÉVELOPPEMENT (3)

- **Récupération des mesures** vers un fichier pour exploitation

- **Nombre de séances de T.P. peu élevé** => nécessité pour l'étudiant de se concentrer essentiellement sur l'algorithme de régulation et non sur des problèmes purement informatiques de programmation graphique et d'interfaçage avec le matériel.

SOLUTION PROPOSEE (1)

- Nous proposons une **véritable application** et non un ensemble de fonctions qu'il faudrait assembler pour obtenir un programme fonctionnant correctement.

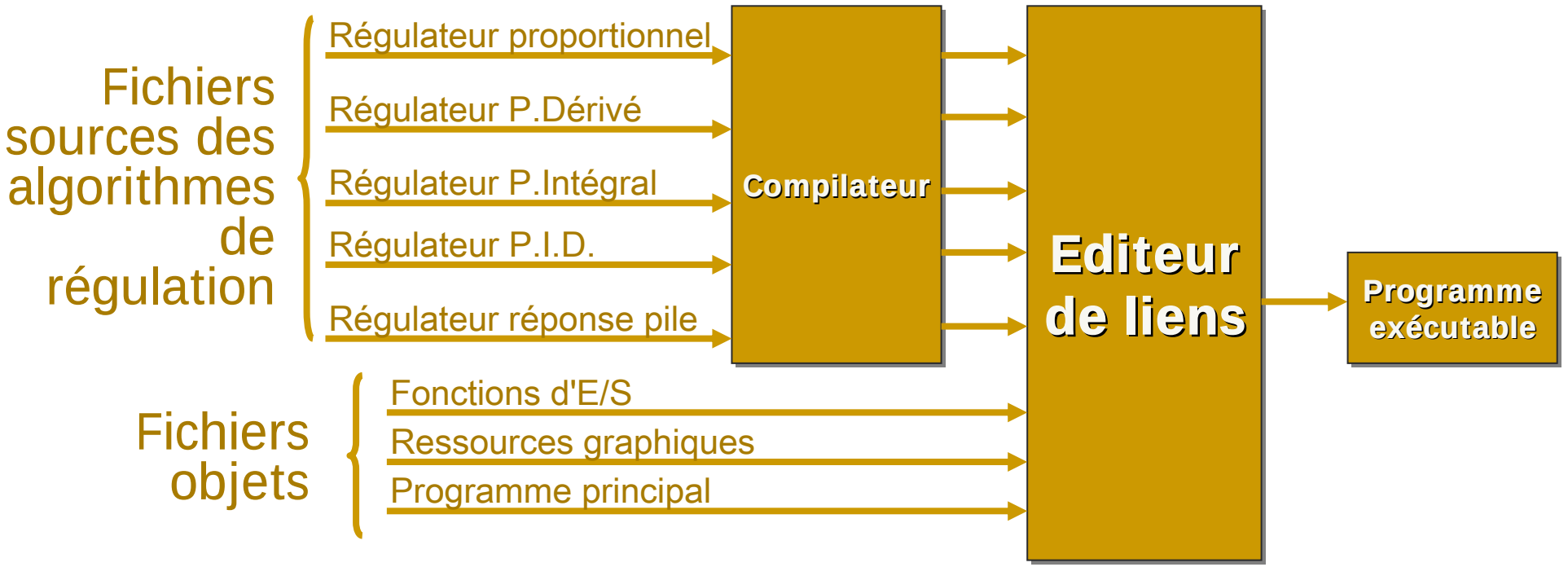
- Seules les **sources** des fonctions de régulation sont **modifiables**, le reste est fourni sous forme de modules objets.

SOLUTION PROPOSEE (2)

- Quatre **fonctions nouvelles** ont été créées pour gérer l'interface :
  - sortie vers un convertisseur,
  - lecture d'un convertisseur,
  - initialisation de l'horloge d'échantillonnage,
  - synchronisation sur cette horloge.

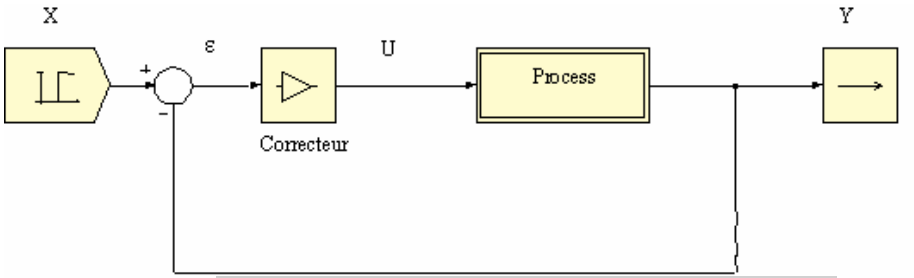
- Ce principe facilite grandement le travail de **déboguage** des travaux des étudiants par l'enseignant responsable de la séance

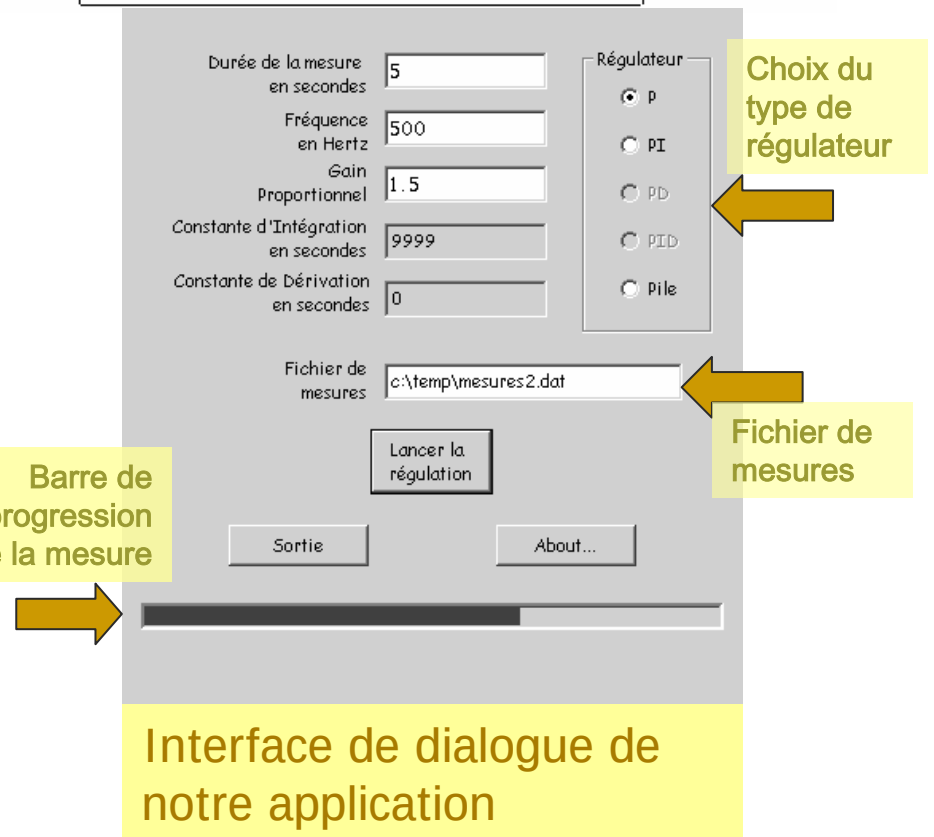
STRUCTURE DE L'APPLICATION



RÉSULTATS (1)

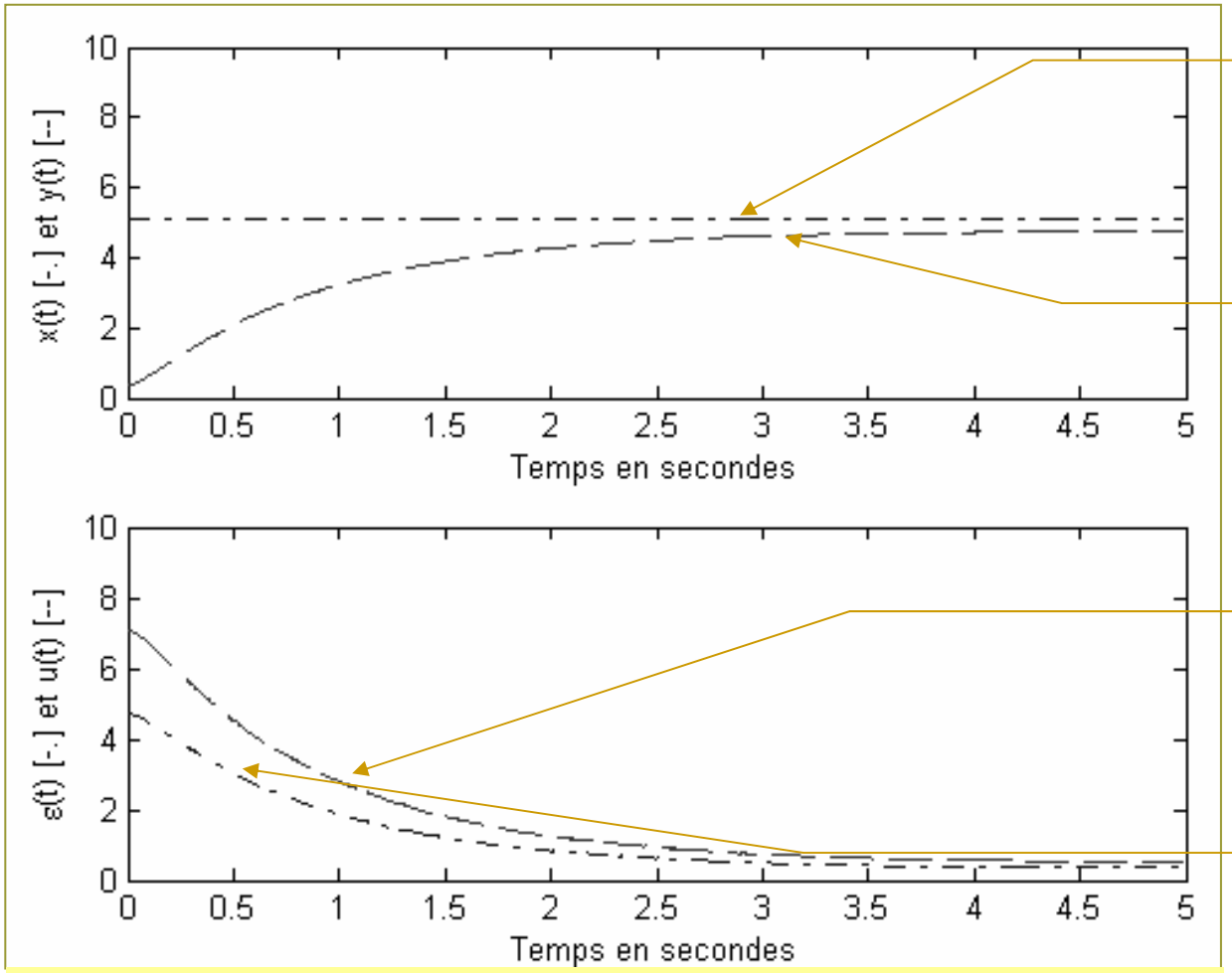
- **Processus :** premier ordre avec intégrateur, pour une constante de temps de l'ordre de 55ms et un gain statique de 0.7
- **Asservissement** de position muni d'un correcteur proportionnel
- **Réglages :**
  - gain proportionnel : 1.5
  - durée de la mesure :5s
  - fréquence d'échantillonnage :500Hz





Interface de dialogue de notre application

RÉSULTATS (2)



Échelon d'entrée

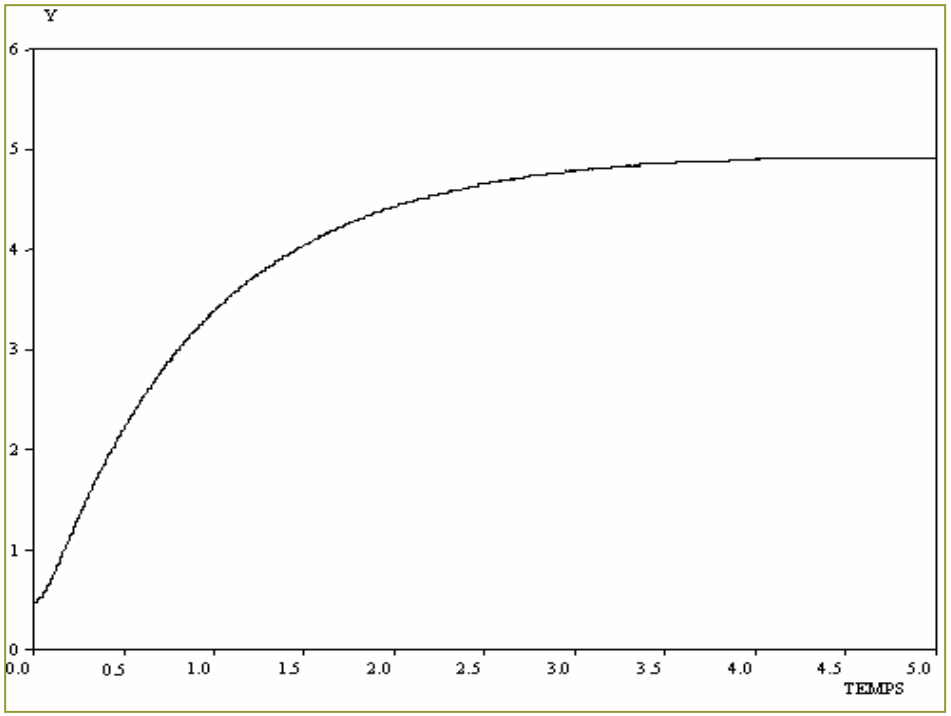
Signal de sortie

Signal de commande

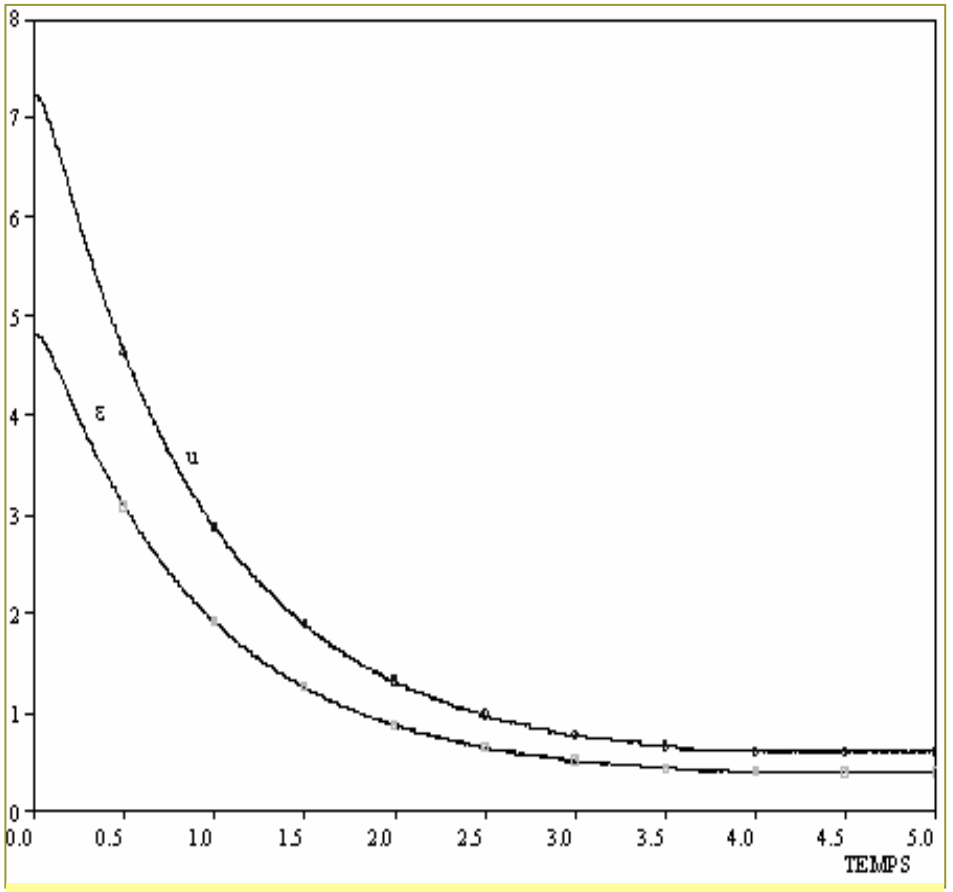
Signal d'erreur

Tracé graphique réalisé à partir du fichier de mesures généré par notre application

PERFORMANCES



Signal de sortie *y(t)* du même processus étudié avec ACSYDE.



Signaux d'erreur et de commande sous ACSYDE

Voici, pour comparaison, les résultats de mesures sur le même processus avec le logiciel ACSYDE

CONCLUSION

- **Avec notre outil**, les étudiants sont capables de programmer un asservissement en langage C, avec l'ensemble des régulateurs proposés (P, PI, PID et réponse pile) **en 6h environ**,
- **Notre plateforme** permet aux étudiants d'aborder les **différents aspects** de la programmation de bas niveau des asservissements numériques.
- **Plusieurs évolutions** sont envisagées:
  - adaptation à d'autres matériels,
  - extension au filtrage numérique,
  - commande à distance par réseau ethernet.

Nous remercions l'IUP du littoral pour nous avoir permis d'élaborer ce projet