

Application n°2 (marine) Navire à cycle combiné

Charles MODIGUY (CS)
Nhan TRUONG-TRUNG (CS)
Pascal PICARD (SITIA)
David DELUCA (SITIA)

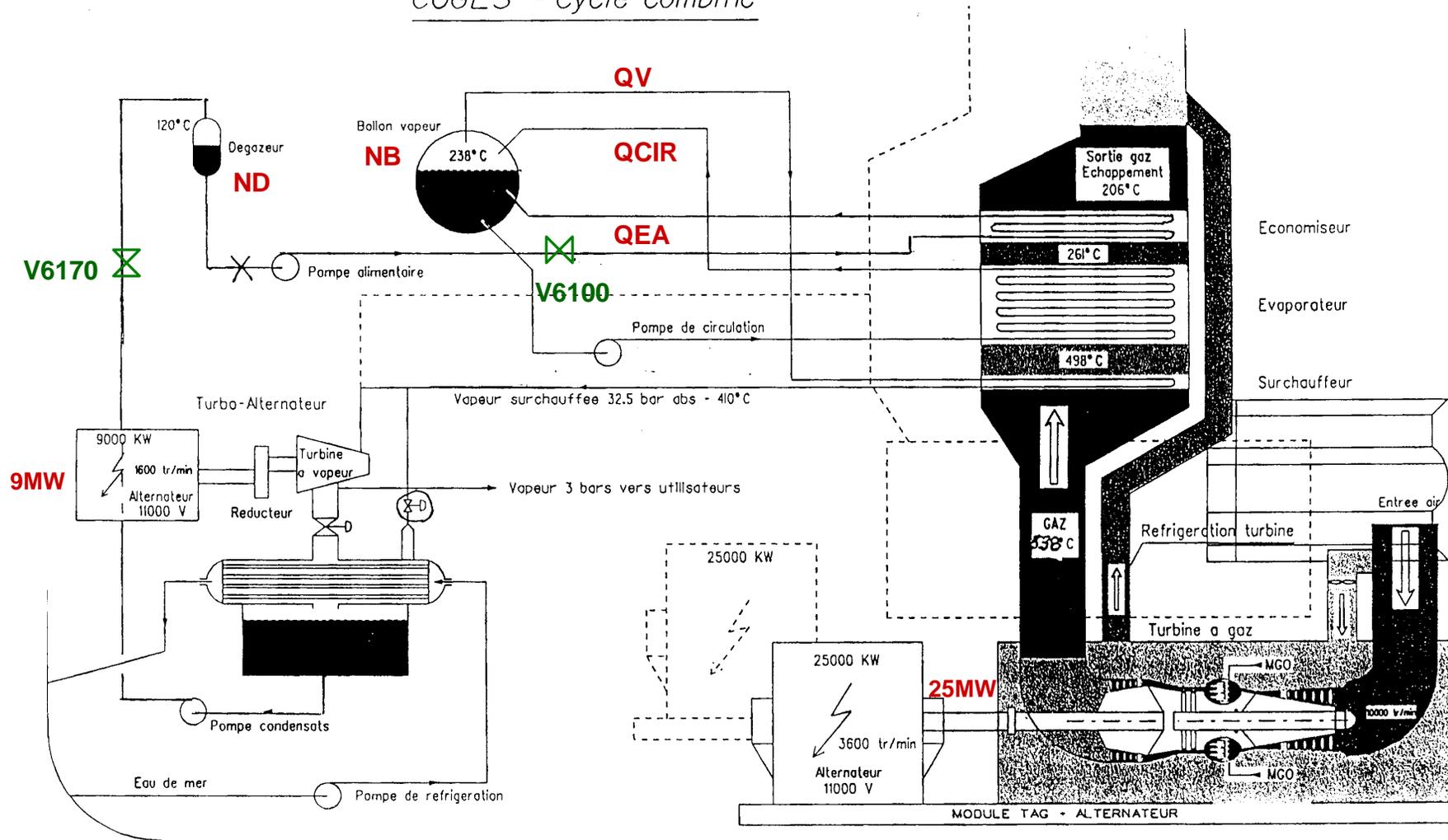


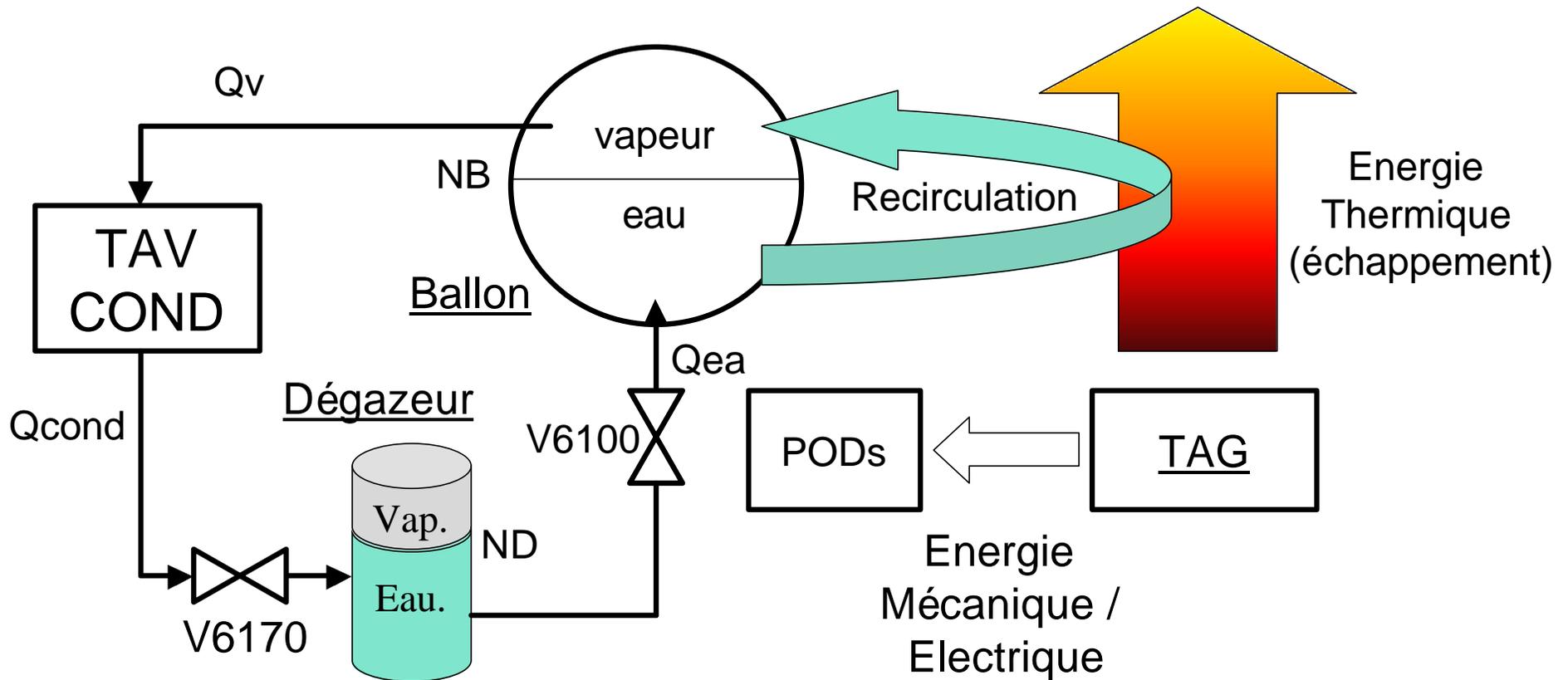
SOMMAIRE

Présentation générale de l'application marine
Développement UML Signal
Développement SynDEX

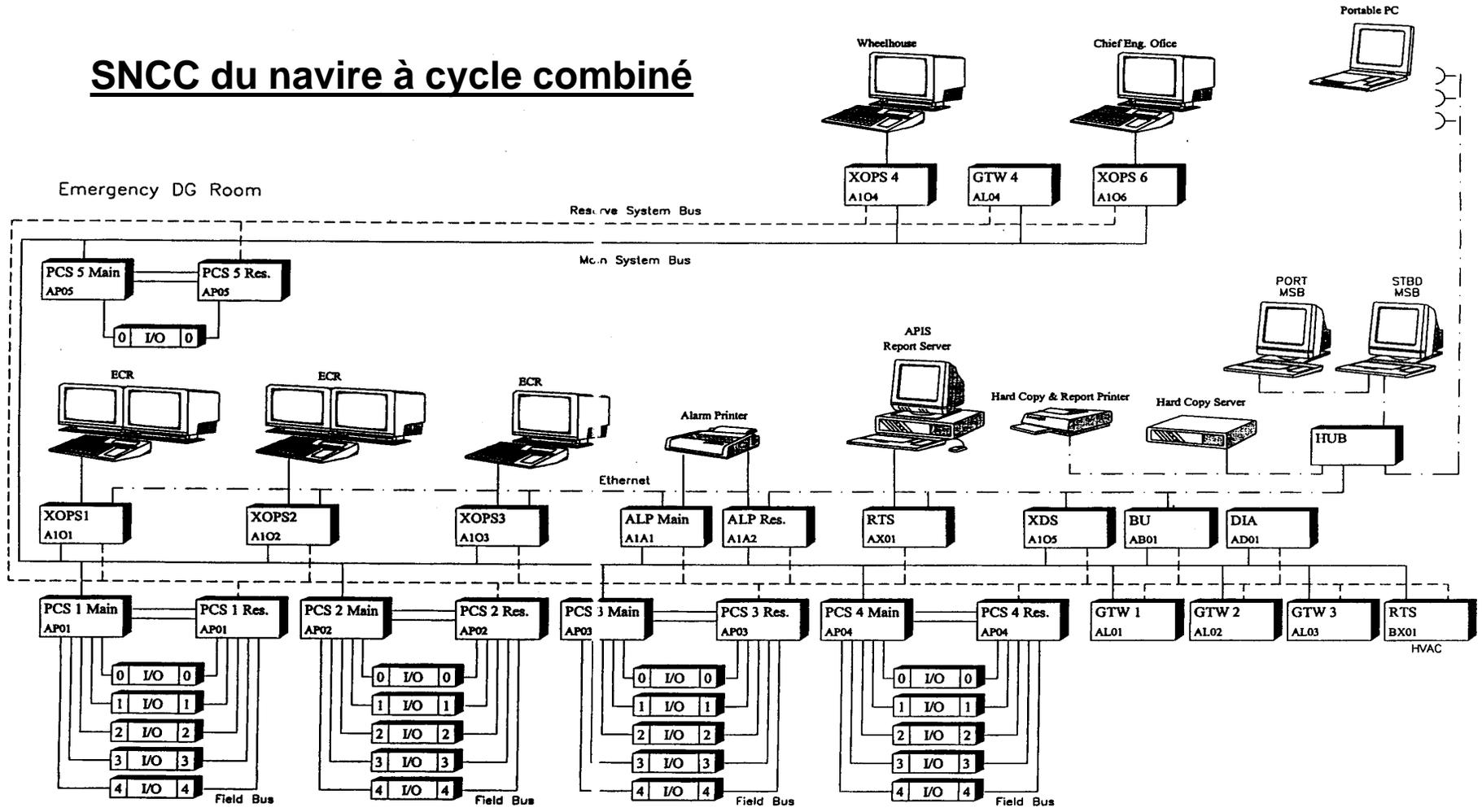
Présentation générale de l'application marine

COGES - Cycle combine



Cycle combiné : schéma général

SNCC du navire à cycle combiné



Fibre Optic Solution with .AOI and FBOI used PCS. 1-4
Number of I/O Racks are only indicative

Choix des processus étudiés

- Ballon vapeur et son contrôle commande (BAL, CCBAL)
- Dégazeur et son contrôle commande (DEG, CCDEG)
- Turbine à gaz (TAG, CCTAG)

Modélisation du ballon vapeur

- Processus : obtenu à partir de résultats expérimentaux
- Contrôle commande : donné par CAT

Modélisation du dégazeur

- Processus : modèle de connaissance + recalage à partir des essais en mer
- Contrôle commande : donné par CAT (du même type que pour le ballon vapeur)

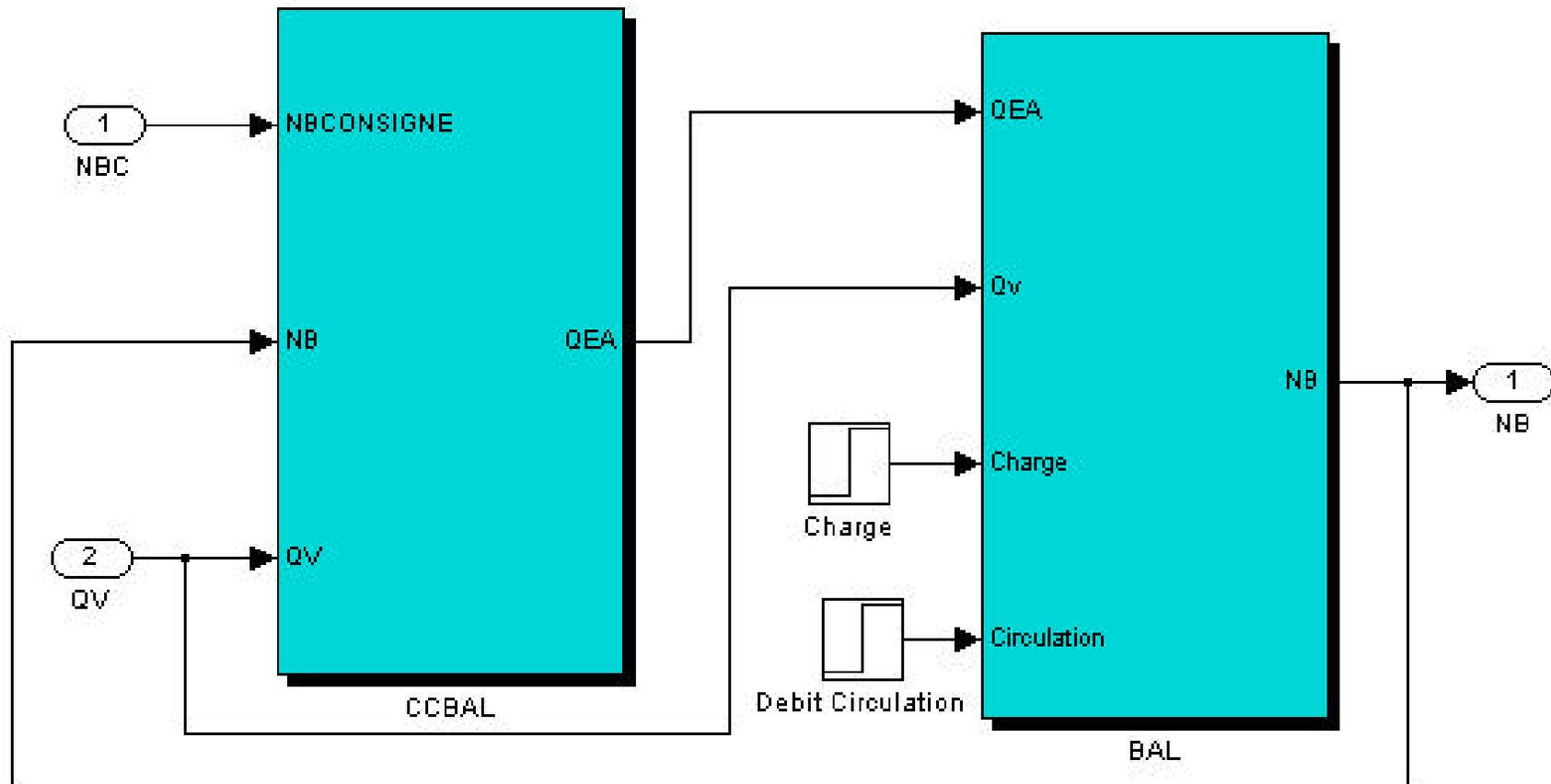
Modélisation de la Turbine à gaz

- Processus : modèle interne (bibliothèque SITIA + avis CAT)
- Contrôle commande : idem

Connaissance et validation des processus et leurs contrôle commande continus

- Développement et simulation sous Matlab Simulink
- Développement et simulation sous Scilab Scicos

Schéma bloc du processus Ballon avec son contrôle commande



Equations du système

Identification de la fonction de transfert NB/QEA

Celle-ci est obtenue à partir de résultats expérimentaux.

$$\frac{NB}{QEA}(p) = G(p) = \frac{Kg \cdot KcorG}{Gb_3 \cdot p^3 + Gb_2 \cdot p^2 + Gb_1 \cdot p}$$

où p désigne la variable de Laplace.

Identification de la fonction de transfert NB/QV

Celle-ci est obtenue à partir de résultats expérimentaux

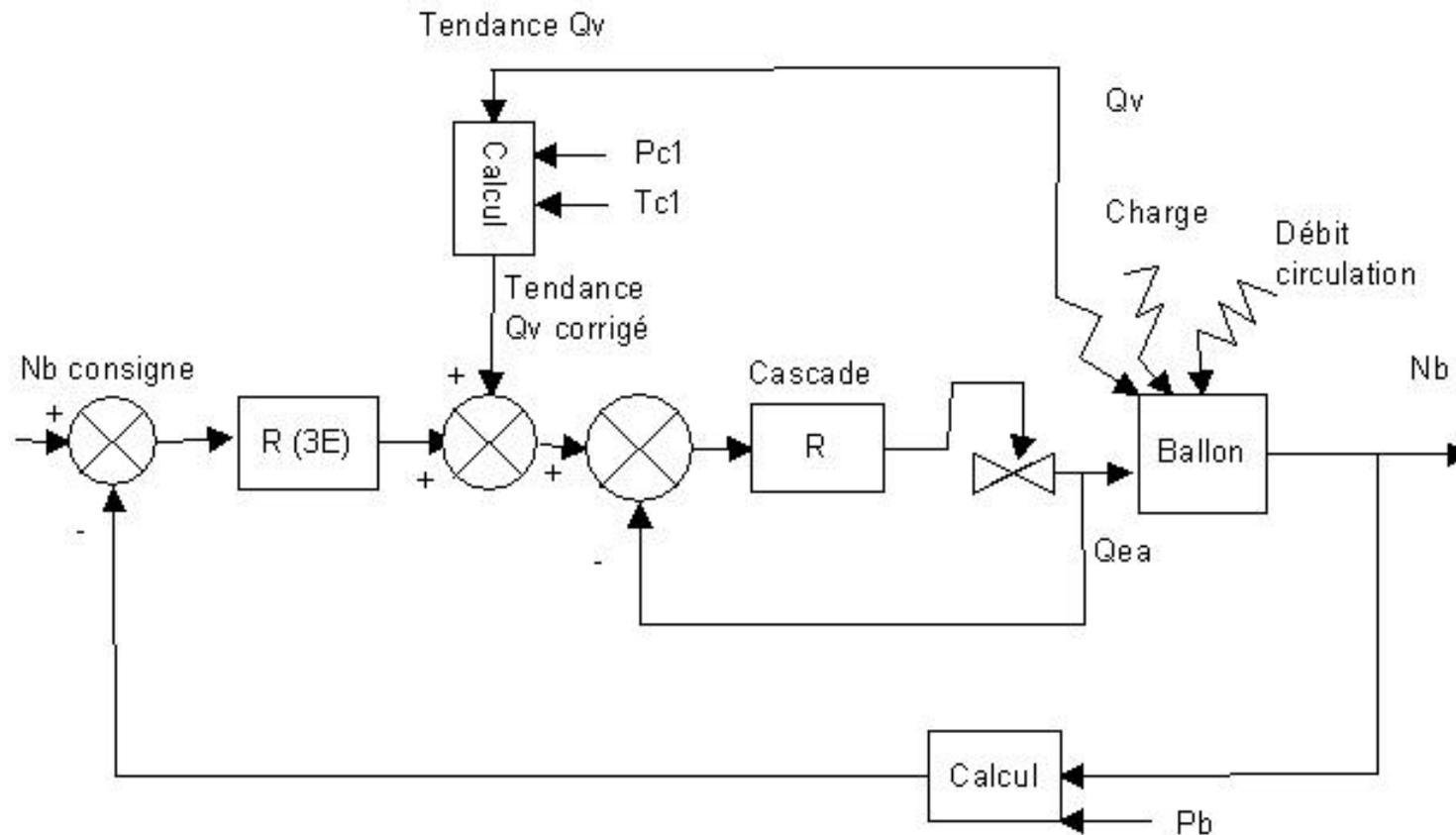
$$\frac{NB}{Qv}(p) = F(p) = \frac{Kf \cdot [1 - To \cdot p]}{Fb_3 \cdot p^3 + Fb_2 \cdot p^2 + Fb_1 \cdot p}$$

où p désigne la variable de Laplace.

Méthode d'identification utilisée

Méthode du maximum de vraisemblance sur modèle d'état avec conditions initiales (SITIA-ECN)

Contrôle commande du niveau ballon : régulations PI suivant les régimes Données : Chantiers de l'Atlantique



Loi de commande du régulateur

Régulation 3E ($Q_v > Q_v^{\text{Seuil}}$)

$$QEA = \frac{1}{K_{corG}} \left(Kp + \frac{1}{Ti \cdot p} \right) (NBC - NB) + Qv$$

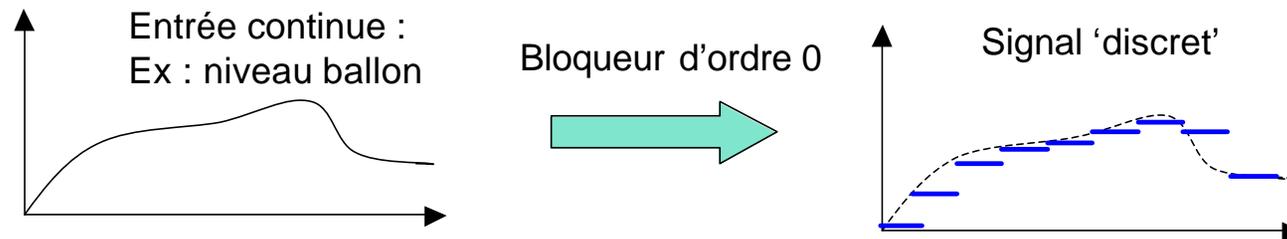
où K_{corG} , Kp et Ti sont des constantes

Régulation 1E ($Q_v < Q_v^{\text{Seuil}}$)

$$QEA = \frac{1}{K_{corG}} \left(Kp + \frac{1}{Ti \cdot p} \right) (NBC - NB)$$

Passage du contrôle / commande du process Ballon (continu) en Signal (discret)

Échantillonnage des entrées (continues) du contrôle commande



Discrétisation de la loi de commande

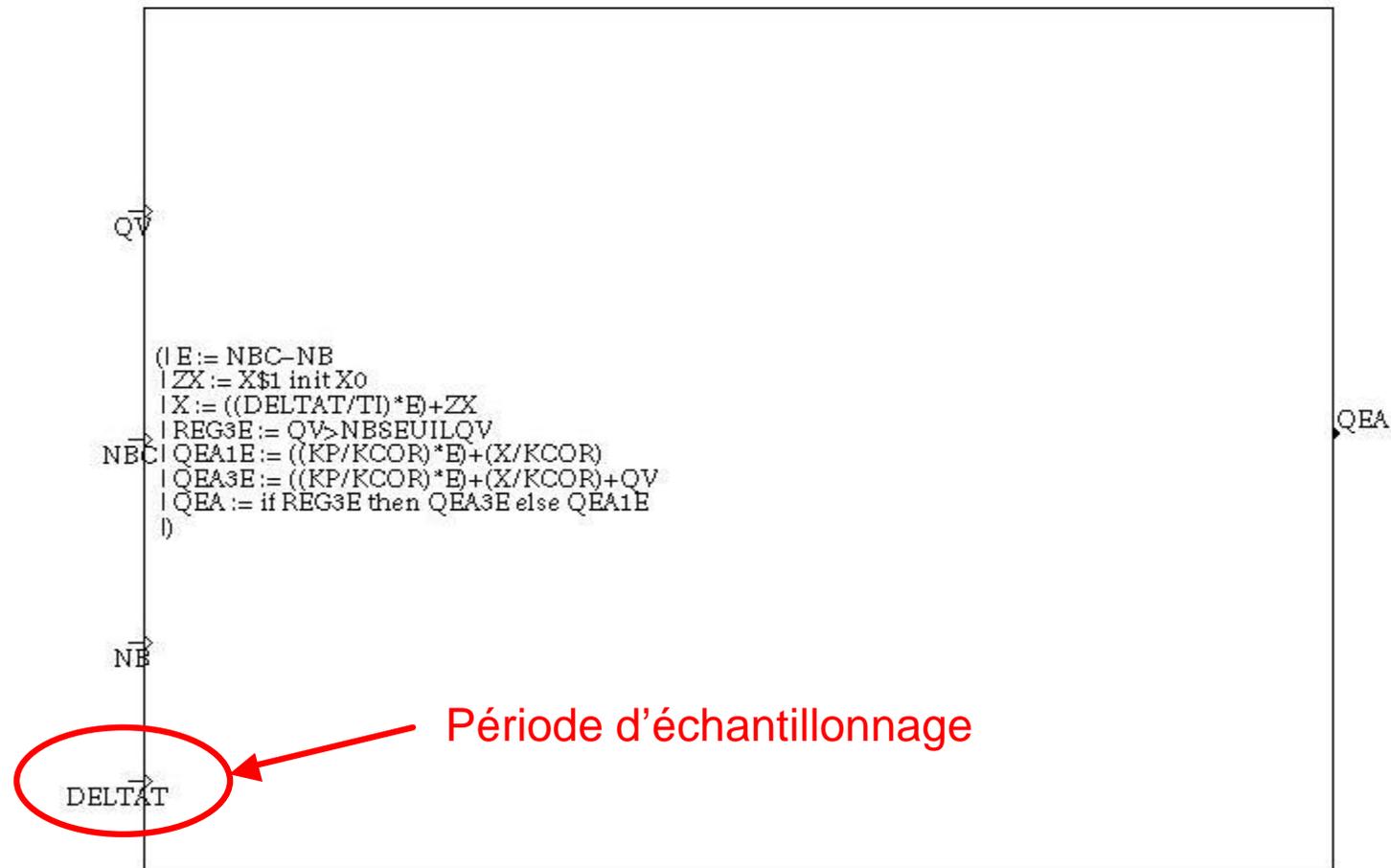
Ex : Le régulateur PI

$$U(t) = Kp \cdot E(t) + Ki \int E(t) dt$$

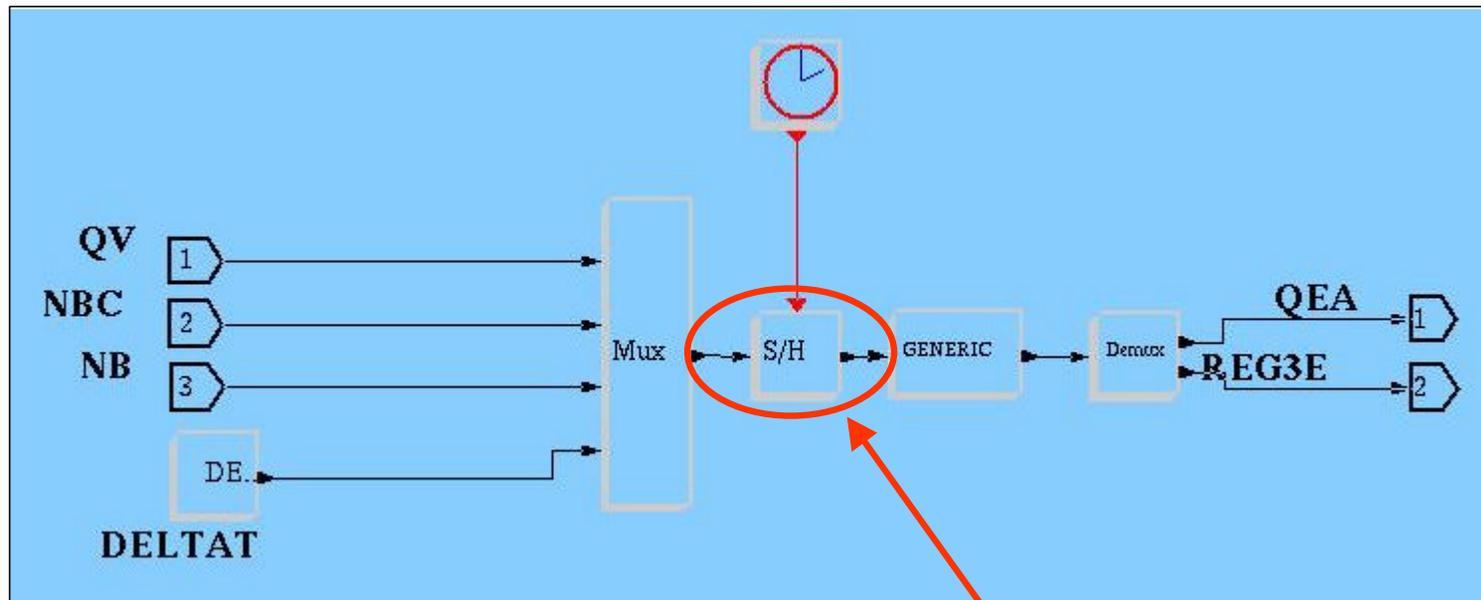
$$U(s) = \left[kp + \frac{ki}{s} \right] \cdot E(s) \text{ où } s \text{ est la variable de Laplace}$$

$$U(z) = \left[kp + ki \cdot \frac{Te}{1-z} \right] E(z) \text{ où } z^{-1} \text{ est l'opérateur retard}$$

Modélisation sous Polychrony du Contrôle Commande Ballon

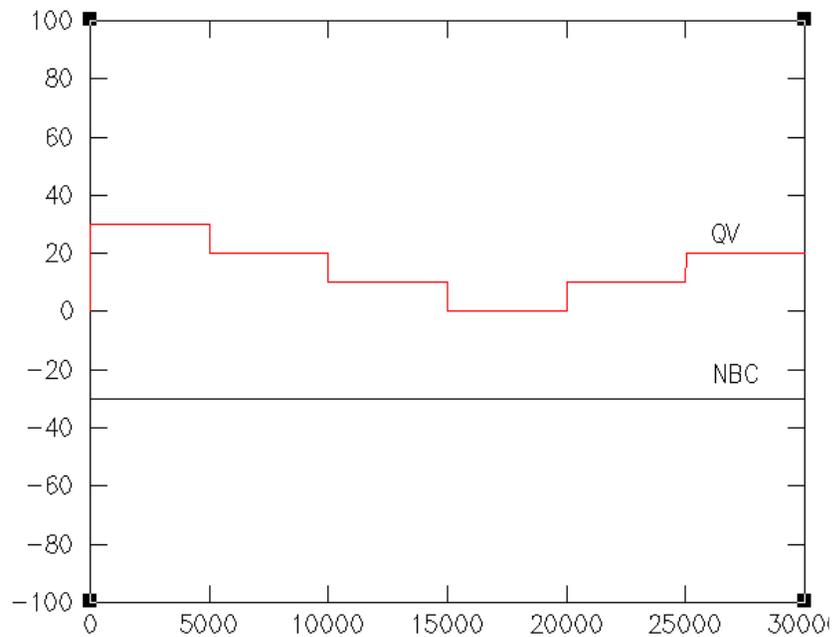


Génération automatique du code C Interfaçage avec Scicos

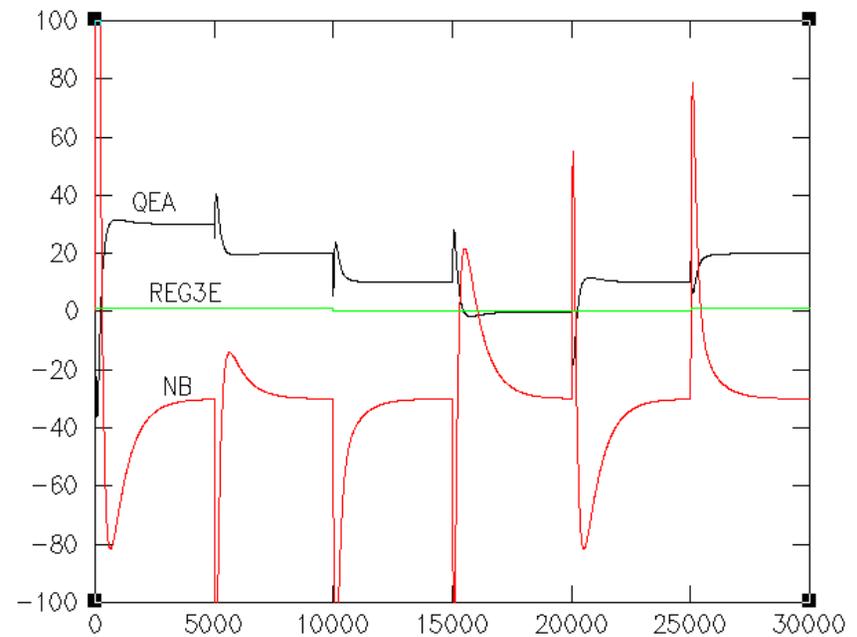


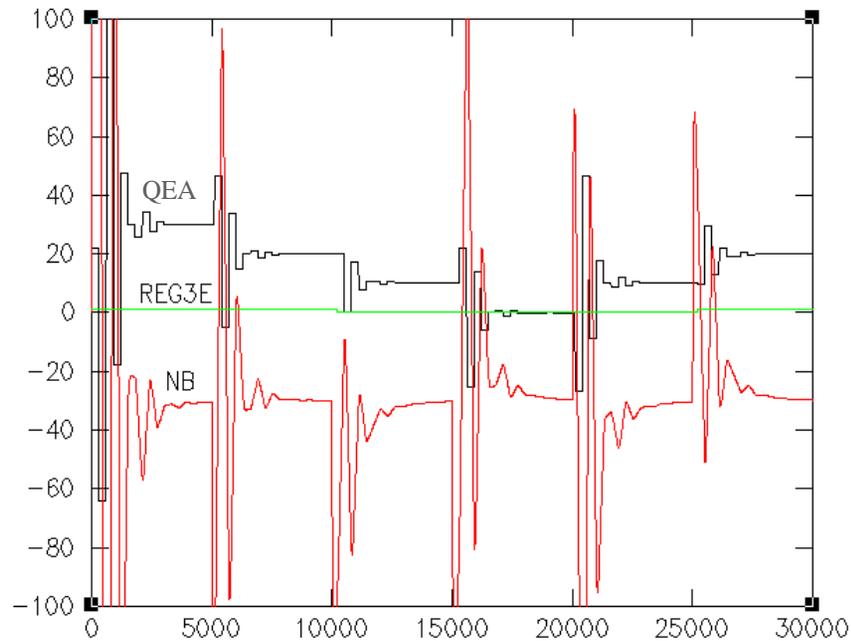
Bloqueur d'ordre 0

Simulation sous Scicos



Signal d'excitation

Simulation du système avec $T_e = 1s$

Simulation du système avec $T_e = 300s$ Effets de l'augmentation de T_e

- Ecart premier pic
- Apparition d'oscillations amorties
- Non convergence

Choix de la période d'échantillonnage

- Ce n'est pas (directement) le théorème de Shannon
- Pour contrôler un process continu en BF avec un régulateur discret, prendre une période d'échantillonnage « petite » devant le temps de réponse du process en BF :
 $0.01 * T_r < T_e < 0.1 * T_r$
Ici : Ballon ? temps de réponse en BF : $T_r \# 1500 \text{ sec.} = 25 \text{ min}$
? $T_e = 150\text{s}, 40\text{s}, 15\text{s}$
Dès que $T_e > 100 \text{ sec.}$, on observe des perturbations numériques

Développement UML Signal

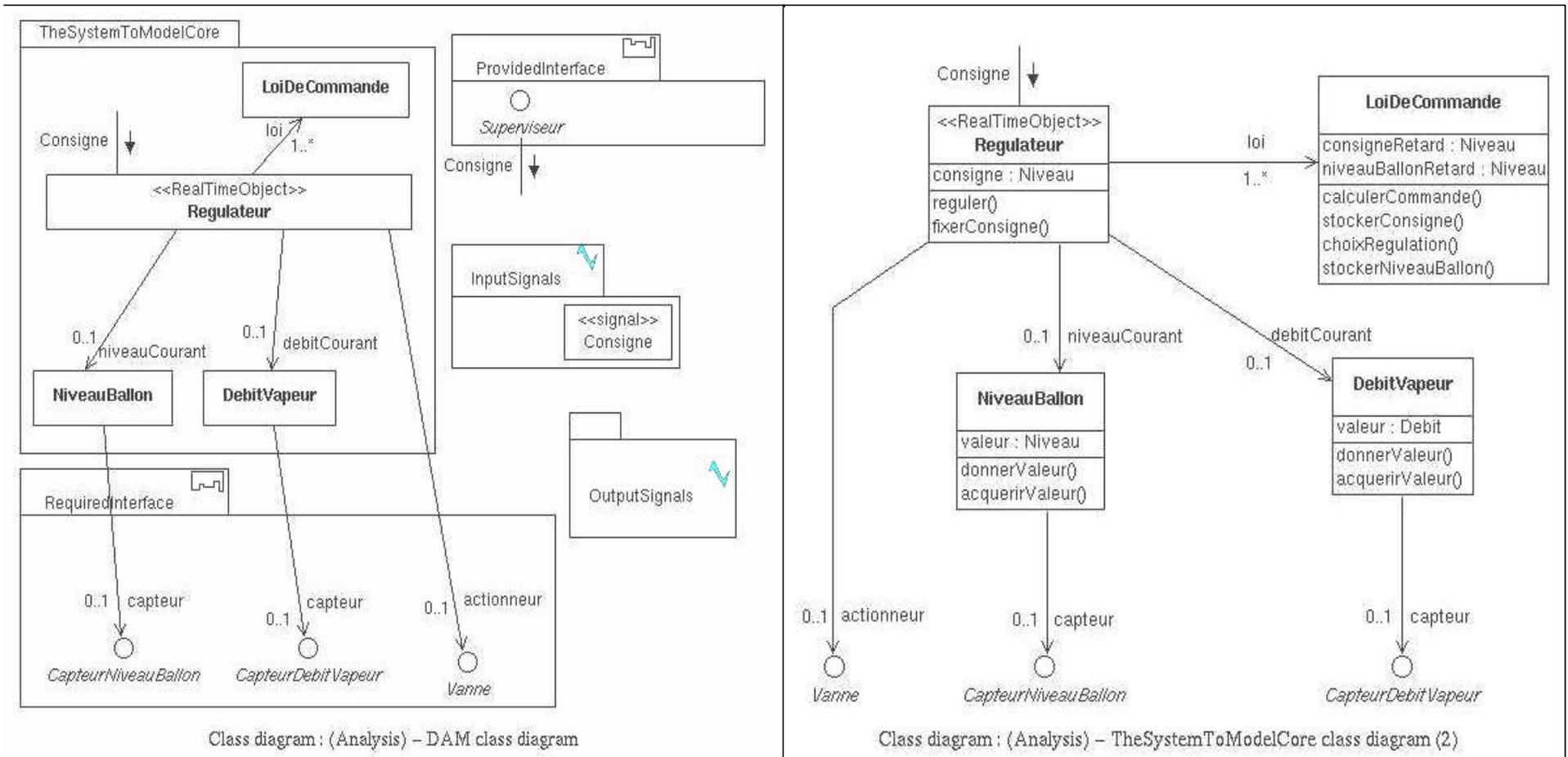
Modélisation suivant la méthodologie ACCORD-UML

- Utilisation de l'AGL Objecteering + modules ACCORD (CEA)

Principaux éléments du cahier des charges du système :

- Acquisition du niveau ballon (capteur niveau)
- Acquisition du débit de perturbation vapeur (capteur débit)
- Enregistrement de la consigne de niveau (superviseur)
- Calcul périodique de la commande (vanne)

Diagrammes structurels du Régulateur du niveau ballon



Scénario détaillé du cas d'utilisation « Reguler »

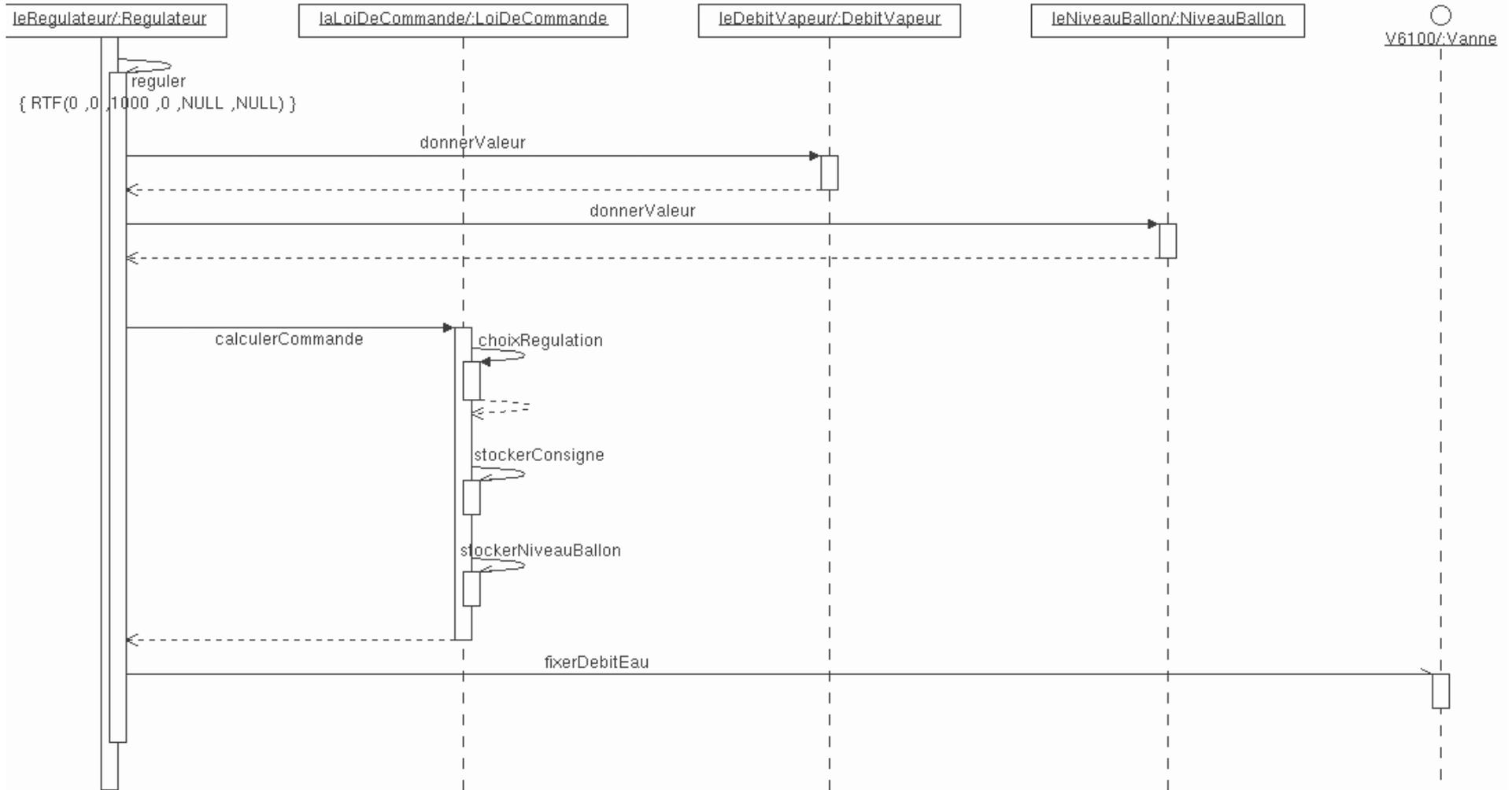


Diagramme de séquence : (DAM building) Reguler::Reguler

Scénario détaillé du cas d'utilisation « AcquerirNiveauBallon »

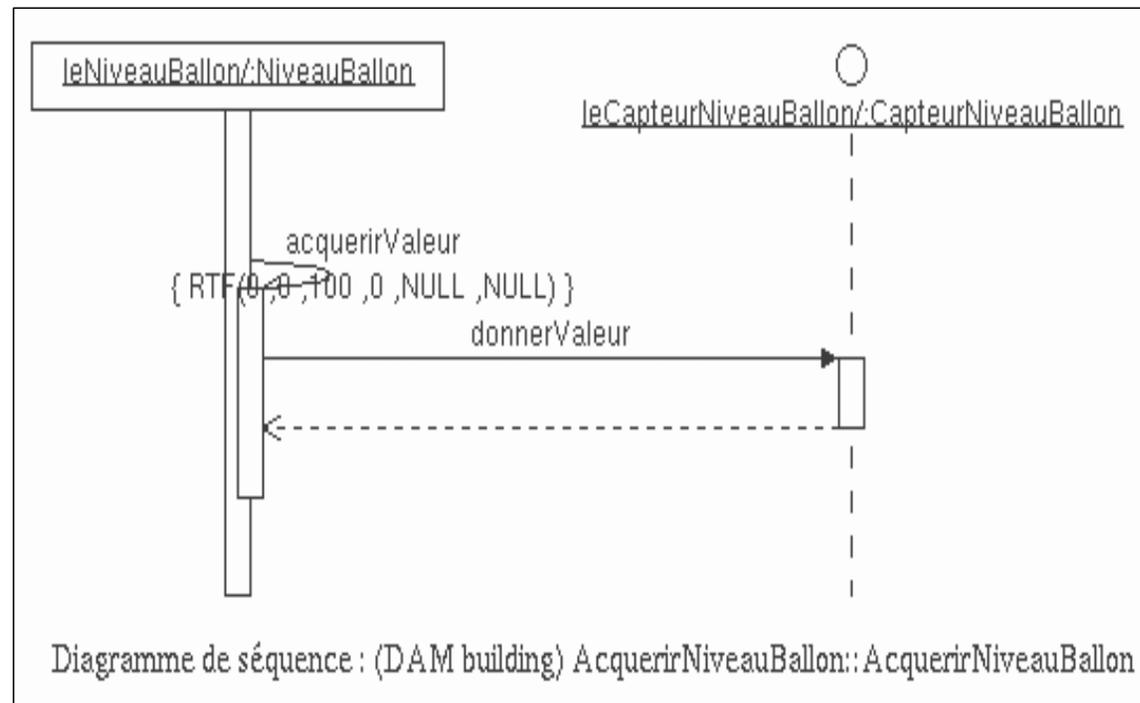
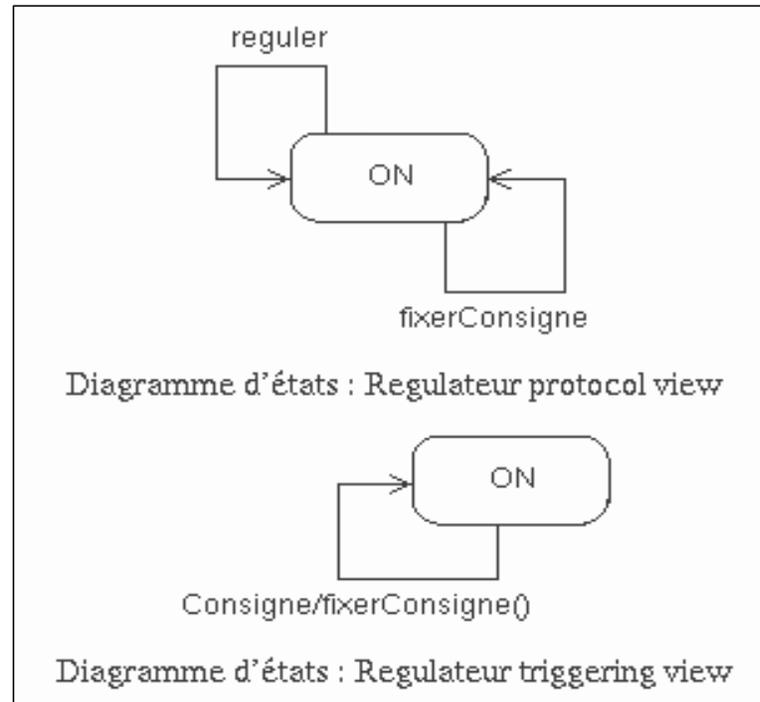


Diagramme comportement de la classe « Regulateur »



Modélisation en langage Signal « à la main » du contrôle commande du niveau ballon

- Etablissement de correspondances entre modèles UML et Signal

Architecture du programme signal

- Un module par classe
- Un process principal
- Structure d'un module :
 - Un process chart (si la classe a une machine à état) : modélisation du comportement
 - des process : correspondent aux différentes méthodes de la classe
- Structure du process principal :
 - un process env_nomDuSysteme
 - Interface avec l'environnement*
 - un sous process main_nomDuSysteme du process env _nomDuSysteme
 - Déclaration des statevar (variables d'état correspondant aux attributs)*
 - un sous-process sub_main_nomDuSysteme du process main_nomDuSysteme
 - Appels des process chart des modules des classes actives*

Ebauche de passerelle UML vers Signal (CS)

Objectifs

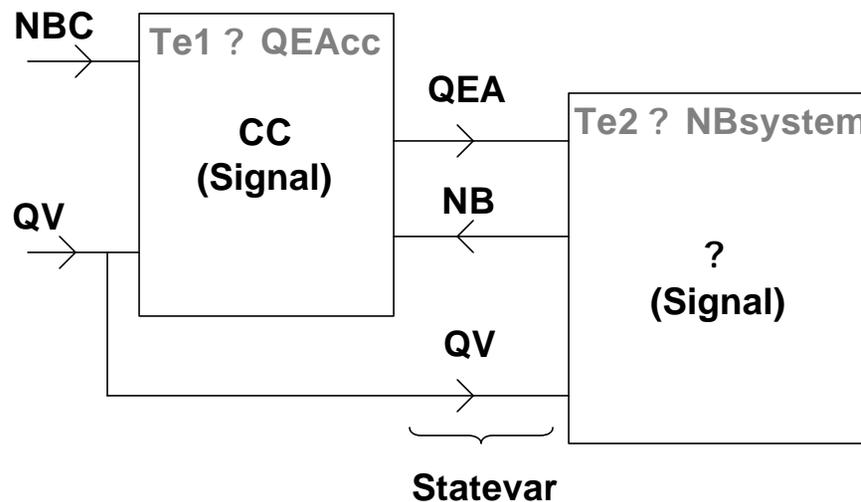
- Prise en main de la génération de code sous l'AGL « Objecteering »,
- Début d'exploration d'une voie non-conforme à la sémantique UML des diagrammes d'états-transitions.

Principaux éléments de la spécification de la passerelle

- Génération d'un programme Signal pour le paquetage UML « Core »
- Hypothèse d'une communication du système avec son environnement par signaux exclusivement
- Construction de la structure du programme (similaire à celui de contrôle commande ballon) par l'analyse des diagrammes structurels
- Construction automatique du corps des process « chart » par l'analyse des diagrammes d'états-transitions des classes actives.

Validation du régulateur ballon écrit en Signal à partir du modèle UML (IRISA SITIA)

- Discrétisation du processus ballon
 - Bloqueur d'ordre 0
 - Validation sous Matlab Simulink
 - Ecriture Signal
- Liaison entre le régulateur et le processus



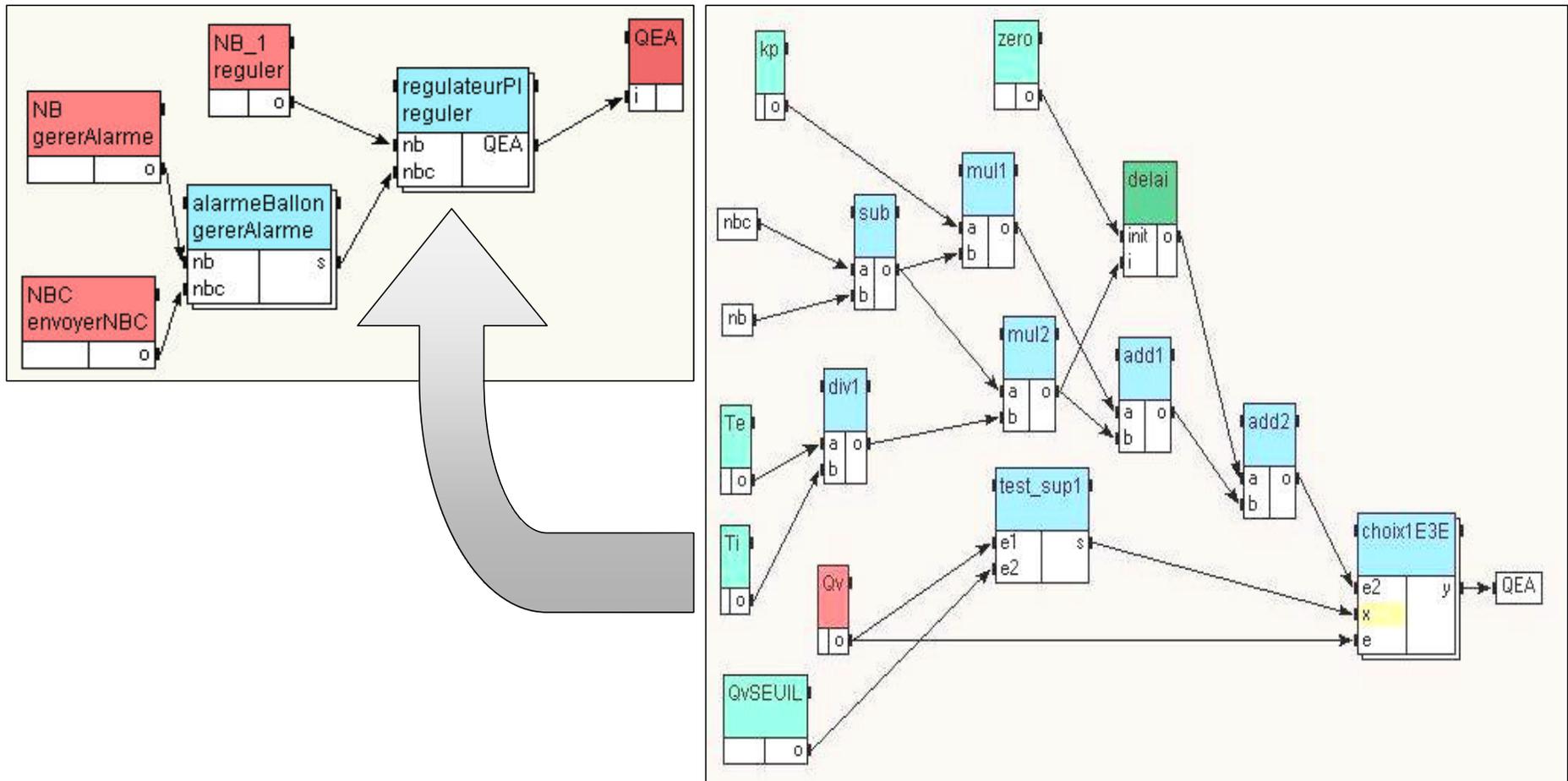
- Simulation graphique sous l'environnement de Signal (en cours)

Développement SynDEx

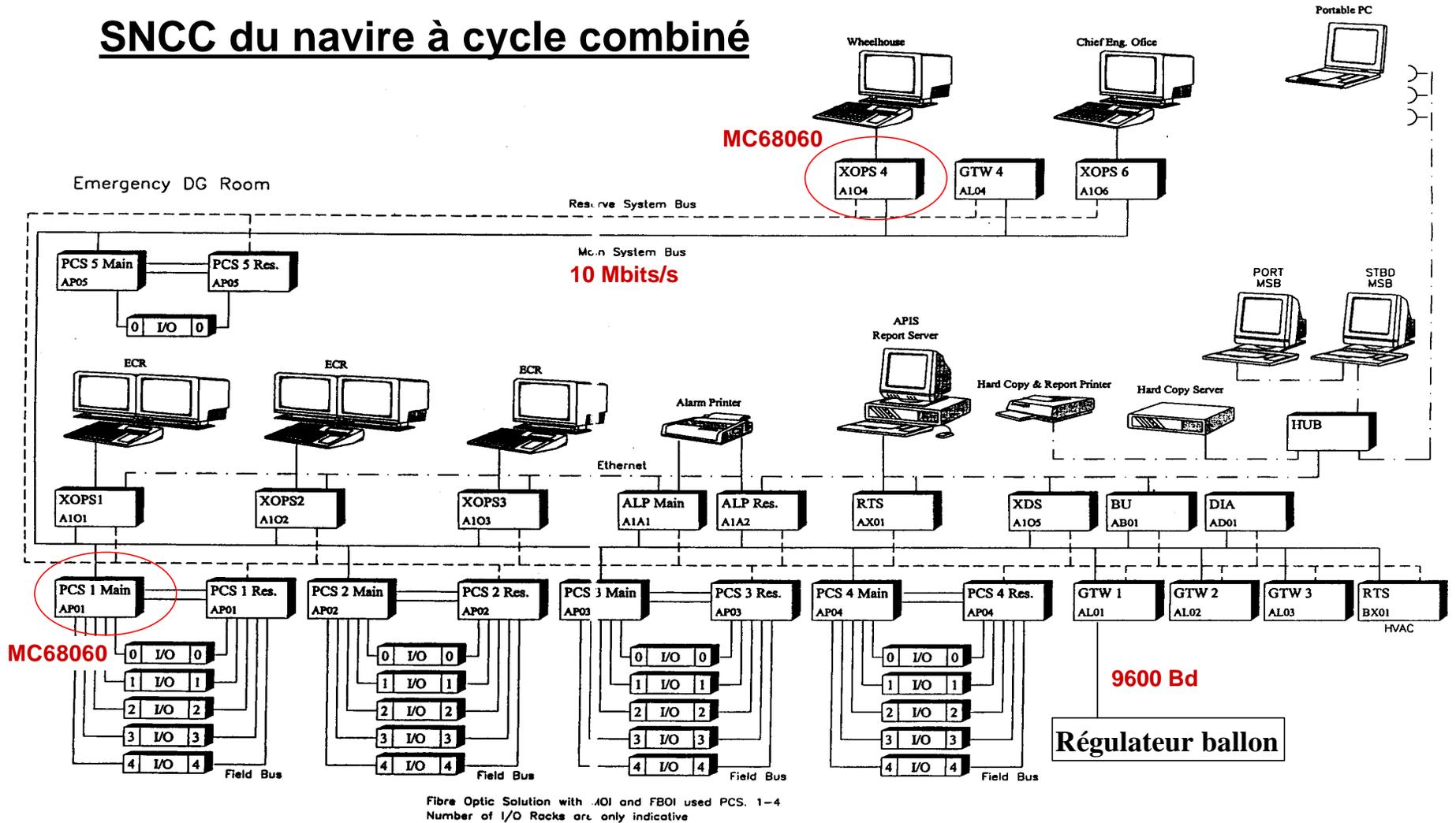
Structure du régulateur ballon sous SynDEx (en cours)

- But
 - Explorer les possibilités de SynDEx en vue de la description de l'application marine
 - Décrire l'algorithme de régulation et l'architecture existante
 - Retrouver un temps de cycle conforme à la réalité
- Algorithme
 - Régulateur PI
 - Alarme sur le niveau qui agit sur la consigne
- Architecture
 - SNCC : régulateur, automates, poste opérateur (XOPS)..
 - Temps entre une commande opérateur ? réponse en sortie < 1 sec.

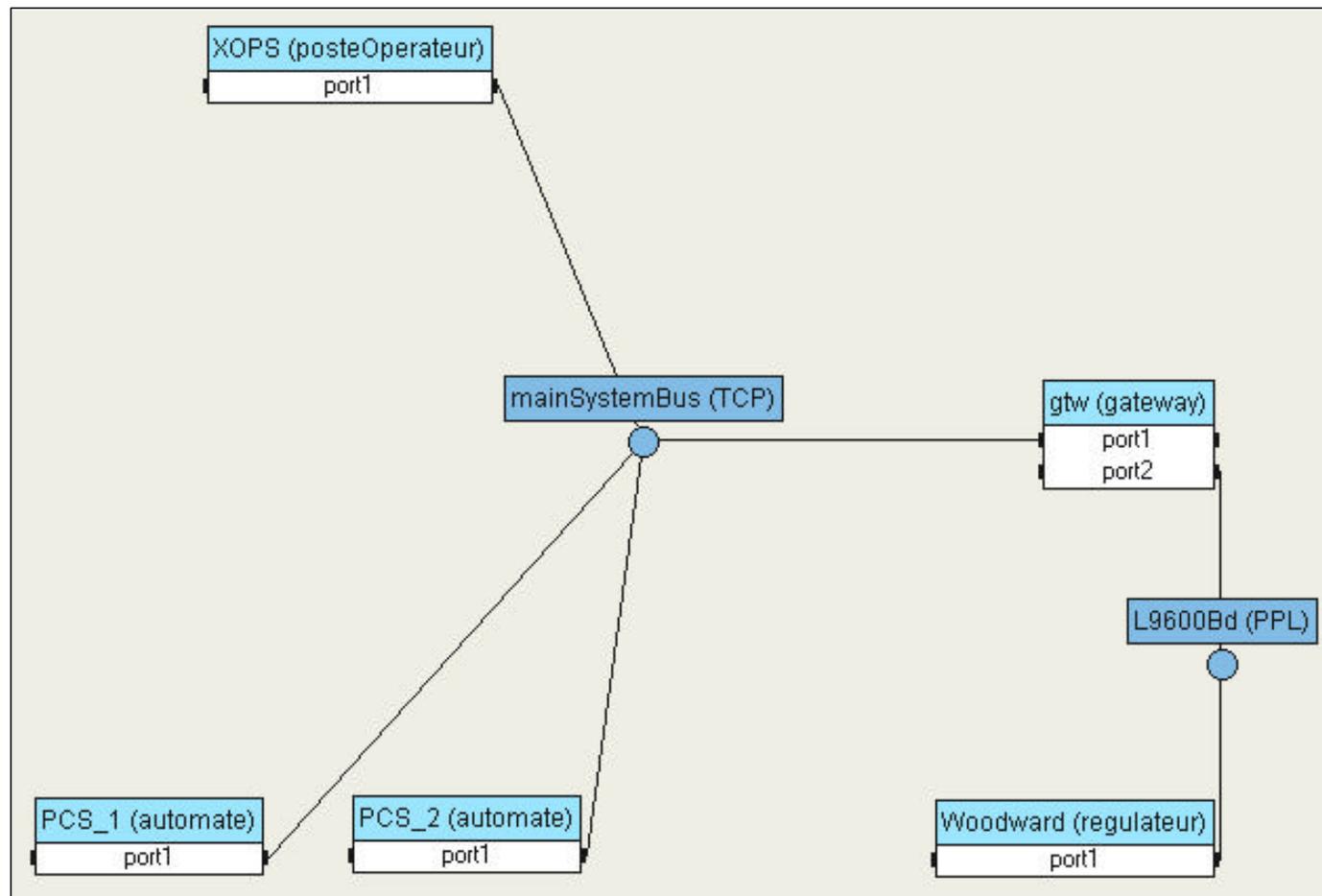
Algorithme de régulation sous SynDEx



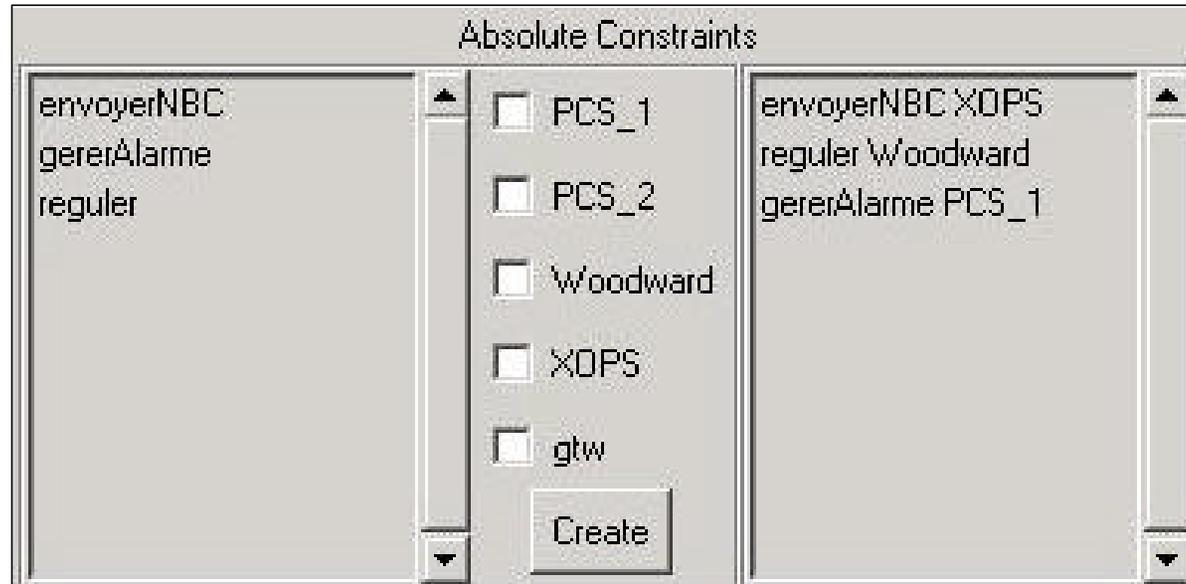
SNCC du navire à cycle combiné



Modélisation partielle de l'architecture



Ajout de contraintes



Manque d'informations

- Encombrement des bus
- Encombrement des opérateurs

Calcul d'adéquation

