HAL
open science

Réglage du routeur d'un sous réseau dédié au contrôle commande temps réel : Approche expérimentale avec apprentissage par réseau de neurones

Steve Limal, Bruno Denis, Ryad Zemouri, Jean-Jacques Lesage

To cite this version:

Steve Limal, Bruno Denis, Ryad Zemouri, Jean-Jacques Lesage. Réglage du routeur d'un sous réseau dédié au contrôle commande temps réel : Approche expérimentale avec apprentissage par réseau de neurones. 3ème Conférence Internationale sur la Productique (CIP'05), Dec 2005, Tlemcen, Algérie. paper #44, 7 p. hal-00424443

HAL Id: hal-00424443

https://hal.science/hal-00424443v1

Submitted on 15 Oct 2009

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.

Réglage du routeur d'un sous réseau dédié au contrôle commande temps réel Approche expérimentale avec apprentissage par réseau de neurones

Steve Limal[1], Ryad Zemouri[1,2], Bruno Denis[1], Jean-Jacques Lesage[1]

[1]LURPA/ENS de Cachan, 61 avenue du Président Wilson - 94235 Cachan CEDEX. Fax : +33 (0)1 47 40 22 20
{prénom.nom}@lurpa.ens-cachan.fr (http://www.lurpa.ens-cachan.fr)

[2] Laboratoire d'Automatisme du *Cnam*, 21 Rue Pinel, 75013 Paris. Fax : +33 (0)1 45 86 21 18
(www.**cnam**.fr)

Résumé

En dépit de leur qualité de déterminisme dédié au contrôle commande, les protocoles traditionnels de réseaux de terrains (tels que FIP ou PROFIBUS) se voient menacés par l'arrivée de solutions utilisant Ethernet. Cet article propose une démarche pour fournir à un utilisateur les valeurs de réglage d'un routeur sur un sous réseau de contrôle commande. Ces valeurs permettront de garantir la Qualité de Contrôle tout en maximisant les communications avec l'extérieur. Un « benchmark » est utilisé dans le but de valider la démarche utilisant un apprentissage par réseaux de neurones.

MOTS-CLÉS. : Réseau de terrain, Protocole TCP/IP, routeur, Réseau de neurones

1. Introduction

Les réseaux traditionnellement utilisés dans les applications de contrôle commande des systèmes automatisés industriels sont des réseaux dédiés. Leur qualité principale est leur aptitude à garantir les exigences de déterminisme des applications. En contrepartie, ils sont souvent « propriétaires », ce qui rend l'interopérabilité et l'interconnexion difficiles. De plus leurs débits sont restés faibles (de 0,2 Mbit/s à 2,5 Mbit/s) et leur coût ramené au point de connexion encore élevé.

L'utilisation des technologies Ethernet et TCP/IP dans le monde du réseau de terrain permet l'introduction de composants matériels ou logiciels ayant déjà largement fait leurs preuves dans le domaine de l'informatique et bénéficiant d'un énorme marché qui permet d'amortir rapidement les coûts de développement et donc d'offrir des coûts de connexion très intéressants. De plus les débits offerts (de 10 Mbit/s à 1 Gbit/s) sont nettement supérieurs aux réseaux de terrains traditionnels, et l'interconnexion des réseaux est simplifiée par l'intégration dans le protocole d'une couche IP de niveau 3 dans le modèle OSI de l'ISO. Néanmoins certaines caractéristiques d'Ethernet et TCP/IP peuvent rendre la solution rédhibitoire si aucune précaution n'est prise pour résoudre les problèmes engendrés.

Le premier problème est l'indéterminisme de l'accès au média Ethernet (CSMA/CD). Pour limiter les collisions, on utilise communément des commutateurs au prix d'un temps de réponse plus long qu'avec des concentrateurs. Des travaux sur les architectures commutées ont proposé des moyens d'estimer l'incidence de ces retards [4]-[10], et conclu sur leur aptitude dans les applications industrielles temps réel [6]-[12].

Le second problème est l'interconnexion des réseaux par la couche IP du protocole, donnant ainsi accès à tous les équipements connectés au réseau aux ressources du terrain, aux valeurs instantanées des E/S, aux variables des contrôleurs logiques, aux pages des serveurs WEB embarqués… Le problème ne vient pas de l'interconnexion elle-même qui est plutôt à créditer aux nombres des avantages du protocole, mais il vient des effets néfastes que cela peut entraîner sur la disponibilité des bandes passantes de chaque équipement.

Pour répondre à ce problème, on se propose au niveau de l'interconnexion entre le réseau de terrain et le réseau d'usine de mettre en place une politique de mise en forme du trafic. Dans la pratique cela consiste à régler les paramètres d'un routeur TCP/IP avec des fonctionnalités de qualité de service (QoS).

L'objectif de nos travaux est de fournir à l'utilisateur final les bonnes valeurs de réglage de la QoS du routeur qui empêcheront la perturbation du trafic de contrôle commande par saturation d'un équipement tout en maximisant la bande passante pour le trafic avec le réseau d'usine. L'idée est de déterminer ces valeurs avec un modèle de réglage élaboré à partir de mesures expérimentales. Nous avons retenu une approche basée sur les réseaux de neurones que nous validerons à l'aide d'un benchmark.

2. Contexte

Les architectures de commande retenues pour notre étude sont celles où les contrôleurs (automates programmables ou calculateurs industriels) utilisent des entrées et des sorties déportées au travers d'un réseau Ethernet commuté avec le protocole TCP/IP (ex : Modbus TCP/IP, Ethernet/IP, Fondation Fieldbus HSE). Tous les

contrôleurs et les module d'E/S déportés sont sur un réseau unique et dédié appelé « réseau de terrain ». Les fonctions de supervision, surveillance, diagnostic, maintenance, … sont situées sur un autre réseau dit « réseau d'usine » séparé du réseau de terrain par un routeur (figure 1). Ainsi, les flux de contrôle aux fortes exigences de déterminisme, sont isolés des autres flux aux exigences temporelles plus souples [13], et les interactions entre les deux flux peuvent être contrôlées au niveau du routeur. On notera T_{cc} le trafic issu des fonctions de contrôle commande et T_{hcc} le trafic des autres fonctions (hors contrôle commande).

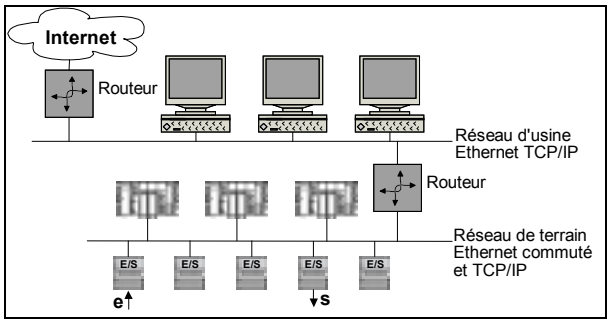

Figure 1. *Structure des architectures étudiées*

Pour évaluer l'impact du trafic dans le réseau sur les performances de la commande, il faut se munir de métriques. Ces performances attendues et les paramètres associés sont désignés par plusieurs auteurs par le concept de « Quality of Control » (QoC) [1]-[8]. Dans notre étude, nous avons retenu le temps de réponse entre un événement « e » sur une entrée et sa conséquence « s » sur une sortie.

Nous faisons l'hypothèse que les réseaux de terrain étudiés sont déjà suffisamment dimensionnés pour satisfaire aux contraintes du trafic T_{cc} seul, et donc que la qualité de contrôle (QoC) est garantie lorsque le trafic total T_t se limite au trafic de contrôle commande T_{cc}. Mais dès qu'un réseau de terrain est raccordé au réseau d'usine, le trafic total dévient : $T_t = T_{cc} + T_{hcc}$. Le rôle du routeur à Qualité de Service (QoS) sera de contraindre pour chaque équipement la valeur du trafic T_{hcc} qui maintienne la Qualité de Contrôle (QoC) de l'application.

Parmi les possibilités offertes par un routeur (filtrage, mise en file d'attente de priorité différente, mise en forme du trafic sortant), les services de mise en forme pour limiter le trafic venant de l'extérieur se sont donc imposés. Plus précisément nous avons choisi d'utiliser le gestionnaire de mise en file d'attente « Token Bucket Filter ». Celui ci ralenti les messages en fixant le délai mini entre trames transférées par le routeur à une valeur de réglage donnée [15].

Le problème de garantie de la QoC de l'application devient pour l'utilisateur final un problème de détermination des valeurs de réglage d'un routeur à QoS.

3. Objectifs des travaux

Les objectifs des travaux se situent à deux niveaux qui correspondent à :

- Aider l'utilisateur final en lui fournissant les valeurs du délai à régler sur le routeur sous forme d'un modèle ;
- Elaborer ce modèle en choisissant un outil adapté.

3.1. Aider l'utilisateur final

On définit l'utilisateur final comme étant l'ingénieur chargé de régler le routeur, c'est-à-dire, la valeur du délai mini entre trames venant du réseau usine en direction de chaque équipement du réseau de terrain. Celui-ci a besoin qu'on lui fournisse un moyen simple de déterminer ce réglage.

L'idée est de lui fournir ce moyen sous la forme d'un modèle paramétrique qui représente le comportement du délai de réglage recherché pour un couple (classe d'architecture ; équipement surveillé). La détermination de la valeur de réglage se résumera alors à une application numérique du modèle.

3.2. Elaboration du modèle

Avant tout, on peut facilement imaginer que le modèle destiné à l'utilisateur ne sera pas une fonction universelle. Il faudra permettre l'élaboration d'un modèle pour chaque classe d'architecture, pour chaque équipement dont on voudra éviter la saturation due au trafic. Même si le modèle sera différent pour un couple (classe d'architecture ; équipement considéré), le premier besoin qui a été fixé est celui d'un outil suffisamment générique, c'est-à-dire qui soit applicable quelles que soient la classe d'architecture étudiée.

Le second besoin fixé est celui d'un outil « économe ». En effet l'objectif est d'aider l'utilisateur final à réduire le coût du délai de mise au point sur site du réglage du routeur, et aussi de minimiser le coût du délai d'élaboration du modèle. La partie suivante traite de l'approche retenue pour répondre au mieux à ces deux besoins.

4. Approche retenue

L'approche retenue consiste en un choix de type de modélisation conformément aux besoins évoqués précédemment et en l'énonciation de critères de qualité de la modélisation retenue. L'utilisation d'un benchmark nous permettra d'évaluer l'outil choisi à partir de ces critères.

4.1. Choix d'un outil

Pour élaborer le modèle qui donne la valeur de réglage recherchée, il est possible de s'appuyer sur un modèle de connaissance en s'appuyant sur les lois physiques connues régissant son fonctionnement. Cette démarche implique d'élaborer un modèle spécifique à chaque équipement ; elle nous prive de la généricité de la démarche d'élaboration du modèle recherchée. De plus un modèle de connaissance nécessite de connaître des informations qui peuvent être difficilement disponibles lorsqu'on utilise des équipements propriétaires. Enfin si l'on décide de complexifier un modèle de connaissance pour qu'il devienne plus tolérant, il devient vite impossible de gérer le réglage de tous ses degrés de liberté.

C'est pourquoi nous avons choisi d'utiliser un modèle de représentation : les réseaux de neurones. On considèrera chaque système comme une boite noire ; des mesures expérimentales nous permettront d'établir une relation mathématique qui corresponde au mieux. S'il est impossible de donner un sens physique aux paramètres d'un réseau de neurones, cette méthode d'identification présente l'avantage de ne pas préjuger d'une forme à priori, ce qui va dans le sens de la généricité recherchée. De plus, cet outil se présente comme un outil d'identification possédant une méthode de fixation de ses degrés de liberté à partir de la donnée des points de mesure. Cette phase d'utilisation des réseaux de neurones est qualifiée de phase d'apprentissage.

4.2. Critères

Pour qualifier le modèle obtenu par apprentissage avec un réseau de neurones, il est important de disposer de critères d'évaluation. La base des critères retenus est la valeur absolue de l'écart entre le modèle et le réel pour une architecture donnée (c'est-à-dire pour une valeur fixée de chaque variable du modèle). Cette valeur de l'écart est un critère d'évaluation local du modèle. Pour le qualifier plus globalement, on étend ce critère sur un domaine de variation des variables en retenant la valeur absolue maximale des l'écarts entre le modèle et le réel sur le domaine.

Les deux domaines significatifs qui ont été retenus sont le domaine d'interpolation et le domaine d'extrapolation. Le premier est le plus petit polyèdre convexe dans l'espace des variables du modèle qui inclus tous les points de mesure lors de l'apprentissage, tandis que le second est l'ensemble du domaine de définition des variables moins le domaine d'interpolation.

A partir du critère d'écart absolu maximum en interpolation, on pourra juger de la capacité du modèle à donner de bons résultats à l'intérieur de la zone délimitée par les points de mesures utilisés pour fixer ses paramètres. Tandis qu'à partir du critère d'écart absolu maximum en extrapolation, on examinera la capacité d'un modèle à garder une tendance proche de la réalité lorsqu'on s'écartera de la zone délimitée par les points de mesures utilisés pour fixer ses paramètres.

De plus, la valeur de l'écart absolu maximum va nous offrir la valeur de la correction à appliquer pour prendre en compte les erreurs d'approximation de l'outil utilisé. Le fait de se baser sur le pire écart a pour conséquence de ne pas être optimum dans les zones où le modèle est proche de la réalité, mais le besoin principal est de garantir la qualité de contrôle en prévenant la saturation des équipements.

4.3. Benchmark

Il faut nous munir d'un moyen pour juger de la pertinence de notre modèle de représentation [16]. Nous avons choisi pour cela d'utiliser une classe d'architecture particulière comme benchmark. Cette classe d'architecture fait partie de la plate-forme expérimentale du LURPA. Dans le but de faciliter les relevés de valeurs numériques sur le benchmark, nous avons décidé de lui donner la forme d'une maquette numérique construite à partir d'un modèle de connaissance complété par des relevés expérimentaux. Si l'utilisation d'un modèle de connaissance ne répond pas à nos besoins comme il a été évoqué précédemment, il est approprié pour la réalisation d'un benchmark. Bien sur, le modèle de connaissance utilisé doit néanmoins être représentatif du comportement de la classe pour l'équipement considéré.

A cette condition, il va offrir un ensemble de données de référence qui vont nous permettre de calculer les valeurs des critères définis précédemment pour conclure sur la pertinence de l'approche et pour mettre en évidence les meilleures solutions d'apprentissage. Le calcul de critères à partir des données de références pour différents modèles nous fournira également le moyen de les classer entre eux.

La classe d'architecture que nous avons choisi d'étudier pour appliquer notre approche est une classe d'architecture ayant une topologie proche de celle illustrée par la figure 1 avec un sous réseau de terrain reliant des modules d'entrées sorties déportées et des contrôleurs. Les contrôleurs utilisés sont des Automates Programmable Industriels (API) qui utilisent le protocole Modbus TCP/IP. Une surcouche au protocole Modbus assure la scrutation périodique d'un groupe de modules d'entrées sorties déportés nécessaires. Les APIs peuvent combiner la scrutation de plusieurs groupes, avec une période spécifique pour chacun d'entre eux. Le tableau 1 liste les variables de cette classe, avec leur domaine de définition.

Variables du benchmark	Domaine de définition	Restriction pour le benchmark
Nombre de groupes de module d'E/S scrutés par l'API	$n_g \in \{1,2,3\}$	$n_g = 2$
Période de scrutation des groupes de module d'E/S en millisecondes	$T_1 = 10$ $T_2 \in \{15,\ 30,\ 45,\ ..,\ 240\}$	$T_1 = 10$ $T_2 = 60$
Nombre de modules d'E/S par groupe	$n_1 \in \{0,1,..,20\}$ $n_2 \in \{0,1,..,20\}$	$n_1 + n_2 \leq 20$

Tableau 1. *Liste des variables de la classe d'architecture du benchmark*

Le benchmark retenu est une sous classe d'architecture pour laquelle un API scrute 2 groupes de module d'E/S, le premier selon une période de 10ms et le second selon une période de 60ms. Le nombre de modules de chaque groupe reste un élément variable de la classe mais la somme totale des modules est limitée à 20 pour que la qualité de contrôle de l'architecture soit garantie.

La maquette numérique du benchmark fournie la valeur du délai D à régler sur le routeur pour prévenir la saturation du coupleur Ethernet TCP/IP de l'API en fonction des variables n_1 et n_2 (figure 2). On retrouve une tendance facilement explicable : plus il y a de modules à scruter plus il faudra limiter le trafic venant de l'extérieur en augmentant la valeur du délai dans le routeur. On retrouve également une limite du nombre de modules au dessus de laquelle le délai est infini (c'est-à-dire pas de routage) : le coupleur de l'API n'est alors même plus capable de gérer les seules communications de contrôle commande sans être saturé.

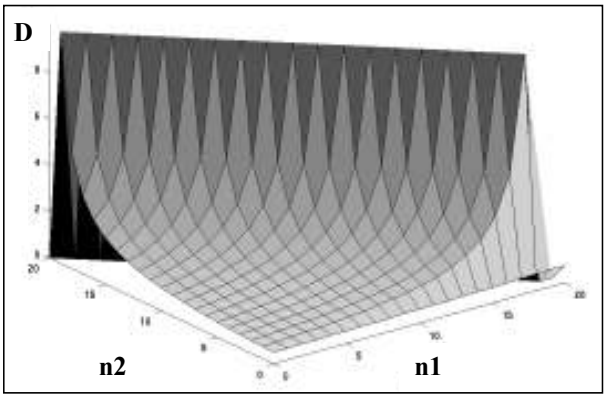

Figure 2. *Modèle de référence pour la classe d'architecture retenue pour le benchmark*

5. Les réseaux de neurones RFR

5.2. Le modèle

Les deux types de réseaux de neurones utilisés en approximation de fonction sont le Perceptron Multi Couche (PMC) [18] et les Réseaux à Fonctions de base Radiales (RFR) [3]-[9]-[11]-[17]. Le type de réseau que nous avons choisi est le réseau RFR car il présente l'avantage de converger rapidement vers un résultat avec très peu de points d'apprentissage. Un réseau RFR est constitué :

– d'une couche de m neurones d'entrées (un neurone par entrée),
– d'une couche cachée de n neurones gaussiens,
– d'une couche de p neurones de sortie ; pour la suite on ne considèrera que des RFR avec $p = 1$,

La sortie $\varphi_i(X)$ d'un neurone gaussien i est la réponse d'une gaussienne pour un vecteur d'entrée X. Chaque gaussienne est définie par les coordonnées de son centre C_i dans l'espace d'entrée, et son rayon $r_{i\theta}$ à hauteur θ donnée. On a :

$$\varphi_i(X) = e^{\frac{\|X - C_i\|^2}{2r_{i\theta}}} \quad \text{avec } X \in \mathbb{R}^m, C_i \in \mathbb{R}^m, r_{i\theta} \in \mathbb{R} \qquad [1]$$

Le neurone de la couche de sortie donne le résultat y d'une combinaison linéaire des sorties des n neurones gaussiens du réseau. On a donc :

$$y(X) = \sum_{i=1}^{n} \omega_i \cdot \varphi_i(X) \text{ avec } \omega_i \in \mathbb{R} \qquad [2]$$

Les paramètres qui définissent complètement un RFR sont donc le nombre de neurones gaussiens n, les coordonnées du centre des gaussiennes C_i, les rayons d'influence des gaussiennes $r_{i\theta}$, et les poids des connexions de sortie ω_i.

5.2. Mise en œuvre.

Deux aspects sont à mettre en œuvre pour l'utilisation des réseaux de neurones : l'apprentissage et l'exploitation.

L'apprentissage à pour but la détermination les paramètres du RFR à partir d'un ensemble de q points d'apprentissage $\{(X_j, Y_j)\},\ j \in \{1,..,q\}$ (figure 3).

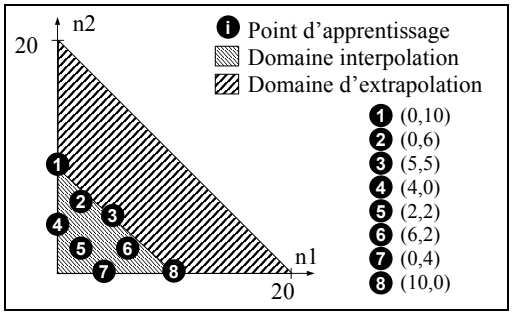

Figure 3. *Vue de la population d'apprentissage.*

L'algorithme d'apprentissage du RFR pour l'interpolation représente deux parties. La première non supervisée pour le calcul des coordonnées des centres des gaussiennes C_i à l'aide de la méthode des K-moyennes [7], et pour le calcul des rayons $r_{i\theta}$ par la technique RCE [5]. La deuxième supervisée consiste à calculer les pondérations ω_i par regression linéaire.

L'exploitation qui a pour but de fournir une réponse Y à une entrée X est la simple mise en œuvre d'une application numérique de la fonction (2) donnant la sortie à partir des entrées.

6. Validation du réglage d'un routeur sur un benchmark

6.1. Démarche de validation

L'objectif est de déterminer si les réseaux de neurones RFR permettent d'élaborer un modèle convenable du délai d'un routeur. La démarche retenue consiste à mener une campagne d'apprentissage sur le benchmark et d'examiner à l'aide des critères retenus pour la qualité des modèles si les apprentissages conduisent systématiquement à un résultat satisfaisant ou si des précautions doivent être prises lors de cette phase d'élaboration du modèle.

6.2. Déroulement de la campagne de validation

Ce qui peut varier d'un apprentissage à l'autre, c'est :

- La population d'apprentissage
  - Taille, $m \in \{8, 20\}$,
  - domaine couvert, $n_1 + n_2 \leq E$ avec $E \in \{10, 20\}$, soit 16% et 100% du domaine de définition,
  - répartition dans le domaine, équirépartis ou placés seulement sur la périphérie,
- La structure du réseau de neurones
  - le nombre de neurones gaussiens, $n \in \{m/4, m/2, 3m/4, m\}$
- L'algorithme d'apprentissage
  - La valeur de $\theta \in [0.8, 0.99]$, [14]

Le calcul des coordonnées des centres des gaussiennes dans l'algorithme d'apprentissage n'est pas déterministe à cause d'une initialisation aléatoire de la valeur des centres [7]. Une approche statistique dans l'évaluation des critères s'est révélée nécessaire. L'algorithme sera appliqué 30 fois sur chaque jeu de valeur pour avoir un échantillon représentatif. Pour chaque critère, la valeur finalement retenue est sa valeur maximale dans l'échantillon.

En tout, N = (96 jeux x 30 échantillons) = 2880 apprentissages ont été réalisés. Pour chaque modèle, les critères d'interpolation et d'extrapolation ont été élaborés à partir du calcul du critère de base local en P=180 points. La figure 4 représente la cartographie des écarts de la sortie du RFR après apprentissage par rapport au benchmark. Les conditions d'apprentissage étaient les suivantes.

- La population d'apprentissage (figure 3)
  - $m = 8$ points,
  - domaine couvert, $n_1 + n_2 \leq 10$,
  - répartition équirépartie dans le domaine,
- La structure du réseau de neurones
  - $n = 100\%$ de la taille de la population ($m = 8$)
- L'algorithme d'apprentissage
  - $\theta = 0.95$

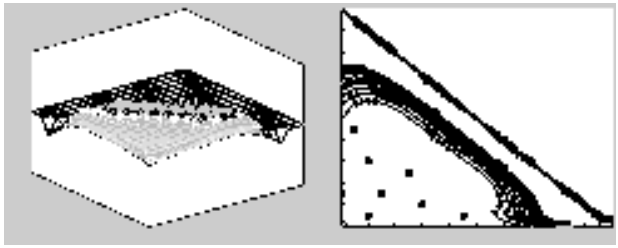

Figure 4. Résultat d'un des 2880 apprentissages de la campagne.

Si l'on compare cet exemple de modèle obtenu (figure 4) avec le benchmark de la figure 2, on remarque que le modèle n'approxime pas bien la tendance asymptotique verticale du benchmark qui fait partie de son domaine d'extrapolation. Par contre la cartographie des erreurs (figure 4) montre qu'en interpolation ($n_1 + n_2 \leq 10$) l'écart reste inférieur à 0,2ms. Les carrés de la figure 4 représentent la position des centres C_i des gaussiennes.

6.3. Interprétation des résultats

Les différents réseaux de neurones obtenus fournissent un modèle du délai à régler au niveau du routeur pour une classe d'architecture qu'est le benchmark. Selon l'instance de la classe d'architecture à

régler par l'utilisateur final le délai varie entre 0 à 10ms. On fixera à 0,5ms la limite supérieure pour un bon critère d'interpolation, et à 1,5ms la limite supérieure pour un bon critère d'extrapolation.

28 jeux de valeurs pour l'apprentissage sur les 96 de la campagne donnent un bon critère d'interpolation, tandis que 14 jeux ont un bon critère d'extrapolation sur les 48 pour lesquels il peut s'appliquer. En première remarque, on note l'existence de solutions satisfaisantes pour l'élaboration d'un modèle de délai par un réseau de neurones. En corollaire, il faut remarquer que la majorité des modèles obtenus ne satisfont pas les critères, la plupart d'entre eux dépassant les 10ms. Il faut donc analyser les points communs des bons jeux d'apprentissage.

L'intersection des jeux d'apprentissage convenables à la fois pour l'interpolation et pour l'extrapolation est de 13 éléments. Il est donc possible d'affirmer que l'interpolation et l'extrapolation ne sont pas des finalités antagonistes dans un modèle du délai. Dans ces conditions, analysons cet ensemble de 13 jeux d'apprentissages, 9 d'entre eux forment un domaine compact et cohérent : la population d'apprentissage à $m = 8$ points équirépartis avec $n_1 + n_2 \leq 10$, le réseau à m={2, 4, 6} neurones (soit de 25 à 75% de la taille de la population) et θ est indifférent.

La meilleure façon d'élaborer le modèle délai du routeur pour notre benchmark consiste à choisir le jeu d'apprentissage le plus économe qui réponde aux critères et le mieux centré dans le domaine des 9 jeux satisfaisants. Dans ces conditions il faut retenir un apprentissage avec 8 mesures sur des architectures avec 10 modules d'E/S équirépartis, et construire un réseau de 4 neurones avec θ=0,95. Pour l'utilisateur final, s'il souhaite obtenir le délai pour une architecture de 10 modules d'E/S ou moins, il doit majorer le délai de 0,5ms car le modèle est utilisé en interpolation. En revanche, s'il souhaite obtenir le délai pour une architecture de plus 10 modules d'E/S, alors il doit majorer le délai de 1,5ms car le modèle est utilisé en extrapolation. De cette façon la Qualité de Contrôle est garantie malgré les écarts entre le réel et le modèle.

7. Conclusions et perspectives

Notre objectif était de fournir à un ingénieur mettant en place une architecture de contrôle commande sur Ethernet TCP/IP, une manière simple de déterminer hors ligne le paramètre délai entre trames de Token Bucket d'un routeur (celui-ci sert de nœuds d'interconnexion entre le réseau de terrain mis en place et le réseau d'usine). On cherche ainsi à éviter la saturation d'un équipement tout en maximisant la communication avec l'extérieur. Pour y parvenir, nous avons choisi de lui fournir un modèle de représentation pour une classe d'architecture, pour un équipement, élaboré en utilisant les réseaux de neurones RFR. L'énonciation de critères de qualité et la confrontation de l'outil de modélisation à un Benchmark nous ont permis de juger de la validité de ce choix.

L'approche par réseau de neurones permet d'atteindre les objectifs fixés. La seule précaution à prendre est relative à la population d'apprentissage qui doit être représentative du réel, malgré la grande souplesse qu'offrent les réseaux de neurones. Le point fort de l'approche est d'obtenir un modèle de bonne qualité avec peu de mesures expérimentales (seulement 8) et sur de petites architectures mesurées (10 modules d'E/S au plus).

Nos travaux en cours tendent à finaliser la validation de la démarche : les résultats obtenus sur le benchmark nécessitent une validation sur architecture réelle pour pouvoir conclure de manière certaine sur la capacité de l'approche à atteindre l'objectif.

L'approche décrite consistait en un calcul hors ligne des paramètres. En perceptive on peut imaginer qu'une procédure en ligne, préliminaire à la mise en service, donnerait de meilleurs réglages. A terme, cette procédure pourrait naturellement aboutir à une auto configuration du réseau qui soulagerait complètement l'ingénieur de la configuration du routeur.

8. Bibliographie

[1] Årzén K.-E., Cervin A., Henriksson D., «Resource-Constrained Embedded Control Systems», Actes Co-design in Embedded Real-time Systems Workshop CERTS2003, Porto, Portugal, 2003.

[2] Brishoual M., Reconstruction de données. Application à la dosimétrie des radiotéléphones, Thèse de doctorat, LCST, INSA Rennes, Octobre 2001.

[3] Broomhead D.S., Lowe D., «Multivariable functional interpolation and adaptive networks», Complex Systems, vol. 2, 1988, p. 321-355.

[4] Georges J.-P., Rondeau E., Divoux T., «Evaluation de performances d'architectures Ethernet commuté», Techniques et Sciences Informatiques, vol. 22, n°5, 2003, p. 621-649.

[5] Hudak M.J., «RCE Classifiers: Theory and Practice», Cybernetics and systems, vol. 23 ,1992, p. 483-515.

[6] Lee K.-C., Lee S., «Performance evaluation of switched Ethernet for real-time industrial communications», Computer Standards & Interfaces, vol 24, n° 5, Novembre 2002, p. 411–423.

[7] MacQueen J., «Some methods for classification and analysis of multivariate observations», Actes de the Fifth Berkeley Symposium on Mathematical statistics and probability, vol. 1, 1967, p. 281-297, University of California, Berkeley, USA.

[8] Marti P., Fohler G., Ramamritham K., Fuertes J.M., «Improving Quality-of-Control using Flexible Timing

Constraints: Metric and Scheduling», Actes 23rd IEEE Real-time System Symposium, Austin, TX, USA, 2002.

[9] Moody J., Darken C., «Fast learning in networks of locally-tuned processing units», Neural Computation, vol. 1, 1989, p. 281-294.

[10] Qiong Song Y., Koubâa A., «Evaluation de performances d'Ethernet commuté pour des applications temps réel», Actes 10th International Conference on Real Time and Embedded Systems RTS'2002, Paris, France, 2002.

[11] Poggio T., Girosi F., A theory of networks for approximation and learning, A.I. Memo 1140, 1989, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

[12] Poulard G., Denis B., Faure J.-M., «Modélisation par réseau de Petri coloré des archtitectures de commande distribuées sur réseau de terrain Ethernet et TCP/IP», Actes 5e Conférence Francophone de MOdélisation et SIMulation MOSIM'04, Nantes, France, septembre 2004, p. 405-412.

[13] Simon E., Gressier-Soudan E., Berthelin J., «Avoid LAN Switches IP Routers provide a better alternative for a Real-Time Communication System», Actes Second International Workshop on Real-Time LANs in the Internet Age RTLIA'2003, Porto, Portugal, Juillet 2003.

[14] Zemouri R., Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à la e-maintenance, Université de Franche comté, Novembre 2003.

[15] Bert H., HOWTO du routage avancé et du contrôle de trafic sous Linux (traduction), http://www.linux-france.org/prj/inetdoc/guides/lartc/lartc.html, 2004.

[16] Dreyfus G., Réseaux de neurones : méthodologies et applications, Paris, Editions EYROLLES, 2002.

[17] Powell M.J.D., Radial basis functions for multivariable interpolation, Algorithms for Approximation, Oxford, Clarendon Press, 1987.

[18] Rumelhart D. E., Parallel Distributed Processing Explorations in the Microstructure of Cognition. Cambridge, MIT Press, 1986.