Contribution a la téléoperation de robots en présence de délais de transmission variables

Arnaud Lelevé

ACADEMIE DE MONTPELLIER

UNIVERSITE MONTPELLIER II

– Sciences et Techniques du Languedoc –

THESE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE DE MONTPELLIER II

Discipline : **Génie Informatique, Automatique et Traitement du signal**

Formation doctorale : **Système Automatiques et Microélectroniques**

Ecole doctorale : **Information, Structures, Systèmes**

présentée et soutenue publiquement

par

Arnaud LELEVÉ

Le 12 décembre 2000

Titre :

Contribution à la téléopération de robots en présence de délais de transmission variables

JURY

M. Philippe COIFFET	Directeur de recherche CNRS au LRP	Rapporteur
M. Thierry DIVOUX	Professeur au CRAN (Nancy)	Rapporteur
M. Pierre DAUCHEZ	Chargé de recherche CNRS au LIRMM	Directeur de thèse
M. Etienne DOMBRE	Directeur de recherche CNRS au LIRMM	Examinateur
M. Alain FOURNIER	Professeur au LIRMM	Président
M. Philippe FRAISSE	Maître de conférences au LIRMM	Examinateur
M. Alain MICAELLI	Ingénieur au CEA	Examinateur

A Laure et à toute ma famille,

Remerciements

Attention, cette page est la dernière page de ce manuscrit pouvant contenir des informations non scientifiques, donc lisibles par le commun des mortels. Si vous ne comprenez rien à l'automatique ni aux réseaux informatiques, vous pourrez reposer cet exemplaire à la fin de cette page. Si vous êtes motivé, vous pouvez également pousser jusqu'à l'introduction, voire même le début de l'état de l'art. Après, c'est du chinois.

Il s'agit également de la seule page qui a échappé à la censure, hormis la dédicace. Heureusement, celle-ci (la censure, pas la dédicace) n'a pas été trop sévère ou n'a pas trouvé de prétexte pour exercer ses foudres. Elle a même laissé passer une référence bibliographique fantaisiste ... une seule, je vous rassure.

Il est temps de passer en revue les nombreuses personnes à remercier. Etant limité en place, je risque d'omettre certaines personnes qui auraient aimé passer à la postérité; que ces personnes veuillent bien m'excuser. Si elles se manifestent, je tacherai de les citer dans les remerciements de mon rapport d'Habilitation à Diriger des Recherches dans quelques années.

En premier lieu, je tiens à remercier Gaston Cambon, directeur du *LIRMM* de m'avoir accueilli au sein de son laboratoire réputé internationalement, notamment pour son taux d'ensoleillement annuel.

Je remercie vivement Philippe Fraisse, qui m'a encadré pendant le DEA et la thèse. Il m'a également chaleureusement accueilli au sein de son département *GTR* à l'*IUT* de Béziers. Mes remerciements vont également à Pierre Dauchez, mon directeur de thèse constamment à l'écoute des problèmes des thésards.

M. Philippe Coiffet, directeur de recherche *CNRS* au *LRP* et M. Thierry Divoux, professeur au *CRAN*, ont eu la gentillesse d'accepter d'être rapporteurs de cette thèse. Etienne Dombre, Alain Fournier et Alain Micaelli ont eu la courtoisie d'accepter de tenir les rôles d'examinateurs. Je leur renouvelle mes remerciements.

En second lieu, ces remerciements sont distribués à l'ensemble du département *Robotique*: David Andreu, François Pierrot toujours prêt à motiver les troupes et à repousser les murs, Jérome Vaganay qui s'est sacrifié en prenant le lit de camp à *Anchorage*, Marie-Jo Aldon et André Crosnier toujours de bonne humeur, Olivier Strauss et René Zapata (l'inséparable *compadre* de Bruno Jouvencel) jamais en panne d'inspiration en matière de blagues et Freddy pour sa générosité, sa cuisine et ses farces récurrentes. Ils vont aussi à l'ensemble de l'équipe enseignante du département

GTR de l'*IUT* de Béziers.

Un très grand merci à Corine pour avoir eu la patience de relire mon rapport et qui a été d'une aide précieuse pour maintenir la motivation des troupes.

Il va sans dire que je n'aurais jamais pu commencer ni finir cette thèse sans l'aide administrative de Josette. Je serais également resté cloîtré au laboratoire et je n'aurais jamais connu le froid arctique de l'Alaska sans l'aide de Nadine. Merci également à Nicole pour sa bonne humeur constante, sans oublier les autres personnes du labo que j'ai pu cotoyer depuis 4 ans.

Plein de mercis à tous ceux qui ont partagé mon quotidien, thésards et ex-DEA, pour ne citer que Clotilde, Muriel, Christine (une chtite poire?), Tiphaine (cherchez l'intruse) et Sandrine, pour la gente féminine et, par ordre d'ancienneté, Lionel et ses pensées Zen, Manu, Denis M., Erwann (pekee, pekee, pekee...), Olivier C. (Dézo), Philou, les 2 Fréd. C., Philippe B., Geovany B., Jérome A., Vincent C., Julien A., Gilles D., Benoit T., Laurent T., Philippe L., Abraham S. et Seb. K. pour la gente masculine.

Plein de mercis également à l'ensemble des thésards et permanents qui ont gravité autour des journées jeunes chercheurs, notamment François Berry, Olivier Carmona, Alain Gourdon et Bruno Marhic, ainsi que Xavier Clady qui a repris le flambeau du site web de l'*AJCR*[1].

N'oublions pas Laure et Stéphanie qui ont activement participé au pot et toutes les personnes (la liste est malheureusement trop longue pour figurer ici; cf. tome II) qui ont alimenté ma réserve d'images abracadabrantes et de blagues et qui ont entretenu une correspondance quotidienne électronique (B13s, B4s, le LEM, les parisiens, le néo-belge...).

Pleins de bisous à toute ma famille qui a assisté de près et de loin à l'avancement de mes travaux. Un big big Thank-you avec des bisous à Laure et à Phébus pour m'avoir soutenu pendant les périodes critiques et pour avoir enduré mon absence prolongée durant les nombreuses soirées et week-ends passés au labo à rédiger le rapport et à préparer la soutenance.

Une pensée à Bill Gates pour avoir égaillé notre quotidien de ses bogues artistiques ainsi qu'à Linus Torvald pour avoir inspiré Corine dans son œuvre intitulée *Arnux*.

Bon, j'espère que j'ai oublié personne... Sur ce, je vous souhaite une agréable lecture.

1. Association des Jeunes Chercheurs en Robotique (http://www.ajcr.free.fr/)

Table des matières

Table des figures

Liste des tableaux

Introduction

Certains corps de métier ont parfois besoin de recourir à des manipulations à distance notamment lorsque des objets dangereux doivent être transportés et/ou quand l'environnement est trop agressif pour les humains (milieux sous-marins en grande profondeur, milieux explosifs, ...). Le poste de travail doit alors se situer de préférence le plus près de la zone opératoire mais accessible à l'opérateur humain.

Un exemple typique est la collecte de données sous-marines au moyen d'un véhicule submersible plongé au milieu de l'océan atlantique [SAY 96] et téléopéré depuis la surface. Ce type d'expérimentation requiert un équipement conséquent (bateau superviseur, submersible, équipements de collecte de données, ...), une équipe technique expérimentée et de nombreuses heures de travail in-situ.

Ces manipulations sont loin d'être aisées à mettre en œuvre car n'importe quelle entreprise ne dispose pas des moyens nécessaires à de telles expérimentations grandeur nature. Il y a trois principales raisons à cela :

- financières : coût de l'expérimentation, des transports (humain et matériel), de l'infrastructure à mettre en place et salaires des équipes présentes sur le site opératoire,
- humaines : éloignement du lieu d'origine, besoin d'une équipe technique spécialisée, environnement parfois inhospitalier (par exemple, plates-formes en pleine mer),
- techniques : distance entre le site et le lieu de travail habituel, matériel coûteux et difficile à transporter, ...

Afin de limiter les diverses conséquences évoquées ci-dessus, il est préférable de limiter les déplacements du personnel pour chaque expérimentation. Autrement dit, un minimum de personnes compétentes doit être présent in situ (pour reprendre l'exemple précédent, sur le bateau superviseur) mais contrôle les expérimentations depuis son lieu de travail usuel (sur terre). Cela sous-entend de passer d'une téléopération à courte distance (bateau – sous-marin) à une téléopération à longue distance (bureau – bateau – sous-marin).

D'autre part, les entreprises ont tout intérêt à mettre en commun des ressources dans le cadre de projets de partenariat. Il est en effet intéressant de pouvoir partager

un même site expérimental : les coûts s'en trouvent approximativement divisés par le nombre d'utilisateurs et l'infrastructure de téléopération à longue distance n'en est pas pour autant bouleversée. Ainsi, chaque partenaire pourrait utiliser la même plate-forme d'expérimentation, chacun récupérant les données qui l'intéressent. L'équipe locale n'aurait qu'à se charger des questions purement techniques propres à chaque sortie du submersible.

Du point de vue technologique, la téléopération d'un système, partagée entre différentes équipes disséminées dans le monde et aux équipements informatiques diversifiés, nous a amené à envisager un poste de contrôle à distance sous forme d'une application informatique, exécutable sur n'importe quel type d'ordinateur. La technologie *JAVA* nous a semblé adaptée à ce concept. Malheureusement, elle n'était pas encore assez stable à l'époque du début de ce travail.

Par ailleurs, les cas de téléopération à longue distance impliquent généralement l'utilisation de différents moyens de communication. Pour reprendre l'exemple précédent de téléopération sous-marine, les données doivent parcourir :

- le réseau local de l'entreprise pilote (*Ethernet*, *Token-Ring*, ...),
- le médium longue distance dont le choix dépend notamment des besoins en débit,
- une transmission radio entre la terre et le bateau,
- et enfin, une transmission acoustique du bateau au sous-marin.

En ce qui concerne le médium longue distance, s'il est possible de louer des lignes spécialisées entre une entreprise et un site d'expérimentation, cela devient difficile lorsque ce site est partagé entre plusieurs entités et coûteux lorsque les distances sont importantes. Le moyen de communication informatique longue distance (commun au site et à ces entreprises) qui s'impose alors est l'*Internet*. L'avantage de son utilisation est sa « gratuité » dans le cadre expérimental ; tous les laboratoires de recherche et une bonne partie des entreprises modernes possèdent en général une liaison fixe à l'*Internet*. Il est toujours possible de se connecter à un fournisseur d'accès le temps d'effectuer ce type d'expérimentations. De plus, la structure en forme de maillage de l'*Internet* procure une sécurité de fonctionnement indispensable pour une application telle que la téléopération d'un robot.

Toutefois, l'infrastructure nécessaire à la mise en œuvre d'une téléopération à longue distance va engendrer de nouvelles contraintes auxquelles il faudra se plier. En effet, la mise en chaîne de différents média de communication engendre une bande passante limitée par le médium ayant la plus faible bande passante, ce qui pénalise l'ensemble de la communication.

En outre, nous devrons faire face non seulement à des temps de transmission variables dans le temps (car nous nous plaçons dans des cas où la distance entre le robot et l'opérateur est telle que la vitesse de propagation des signaux ainsi que les délais imposés par les nombreux organes de commutation ne sont plus négligeables) mais aussi à des débits variables (lorsque les media sont partagés entre plusieurs utilisateurs, comme dans le cas de l'*Internet*).

Dans le cas de la commande d'un robot bouclée à distance, ces délais variables, non déterministes, interviennent à l'aller et au retour (de façon généralement asymétrique) et sont un facteur principal d'instabilité de ce type de commande. Par exemple, lors d'un mouvement d'approche, dès que le contact est perçu par l'opérateur, celui-ci donne l'ordre d'immobiliser le téléopérateur. Cependant ce dernier continue d'avancer tant que l'ordre d'arrêt ne lui est pas parvenu, d'où une possibilité de détérioration du téléopérateur et/ou d'une partie de l'environnement. En fait, ces problèmes interviennent dès lors qu'il ne s'agit plus d'un mouvement libre du téléopérateur sans risque de collision. Pour un véhicule, les phases d'évitement d'obstacle sont tout aussi critiques si l'opérateur ne perçoit pas l'obstacle assez rapidement pour pouvoir intervenir à temps. Il est à noter que des délais de transmission de moins de une seconde peuvent déstabiliser un téléopération directe.

Enfin, l'opérateur ne peut avoir qu'une représentation partielle du site opératoire par exemple sous forme de flux vidéo retardés. Ces contraintes techniques rendent la télémanipulation difficile voire incertaine pour l'opérateur.

Vu l'intérêt général et le nombre d'applications de la téléopération à longue distance, vu les contraintes technologiques que cela sous-entend, notre intention est de mettre en œuvre une architecture de téléopération « bas-niveau » d'un robot via un médium de télécommunication numérique présentant les défauts évoqués ci-dessus. Cette architecture permettra ultérieurement de tester des méthodes de téléopération « haut-niveau » comme nous pouvons en rencontrer dans la littérature scientifique actuelle.

Nous allons, dans un premier temps, recenser les différentes théories et technologies employées en vue de téléopérations à courte et à longue distances.

Nous étudierons ensuite un schéma minimum de téléopération à courte distance. Nous analyserons les résultats d'une première expérimentation et nous les utiliserons dans le but d'élaborer un modèle mathématique d'une simple boucle de téléopération. Nous analyserons également le comportement du réseau informatique utilisé comme moyen de transmission dans la boucle de téléopération et nous incorporerons ces résultats dans notre modèle en vue de simuler une téléopération à longue distance.

Nous utiliserons le modèle ainsi développé pour effectuer des simulations à courte et à longue distance ; celles-ci nous permettront de déterminer les défauts à palier pour améliorer le schéma de téléopération à longue distance.

Nous analyserons ensuite les problèmes de stabilité de ces systèmes téléopérés et nous proposerons des améliorations au schéma initial de téléopération fondées sur les conclusions des précédentes simulations. Nous validerons ensuite ces améliorations sur la plate-forme de téléopération mise au point.

Nous proposerons enfin un schéma complet de téléopération mettant en œuvre l'ensemble des travaux étudiés dans cet ouvrage dans le cadre de télémanipulations terrestres en milieu hostile.

Enfin, nous évoquerons les améliorations à apporter à cette étude et nous proposerons des thèmes de recherche propres à compléter ce travail.

Chapitre 1

État de l'art

1.1 Les origines de la téléopération

1.1.1 Les débuts de la robotique

C'est en 1960 que le premier robot industriel fut introduit dans une usine. A cette époque, la robotique se divisait en deux clans ; d'une part celui des industriels qui cherchaient à augmenter leurs cadences en automatisant les tâches les plus fastidieuses et, d'autre part, celui des roboticiens qui, poussés par les progrès en intelligence artificielle, cherchaient la panacée en terme de robot : « le robot universel », autrement dit, l'homme artificiel.

La robotique industrielle a ainsi connu un essor formidable pendant 20 ans, portée par une très forte demande des gros industriels (le secteur de l'automobile en particulier et l'industrie produisant en général en grandes séries) cherchant à diminuer leurs coûts et délais de fabrication en automatisant au maximum les chaînes de production.

Devant l'immensité de la tâche, le second clan a dû se résoudre à une solution intermédiaire : plutôt que de tenter vainement de remplacer l'homme par le robot, il s'est orienté vers la conception d'un robot assistant l'homme. En effet, aujourd'hui encore, certaines tâches complexes nécessitent la présence ou l'intervention de l'homme au cours du processus semi-automatisé.

La robotique industrielle a elle aussi dû s'adapter aux réalités industrielles ; les études de gestion de production et de personnel ont également amené les industriels à réintégrer l'homme dans certains processus de fabrication.

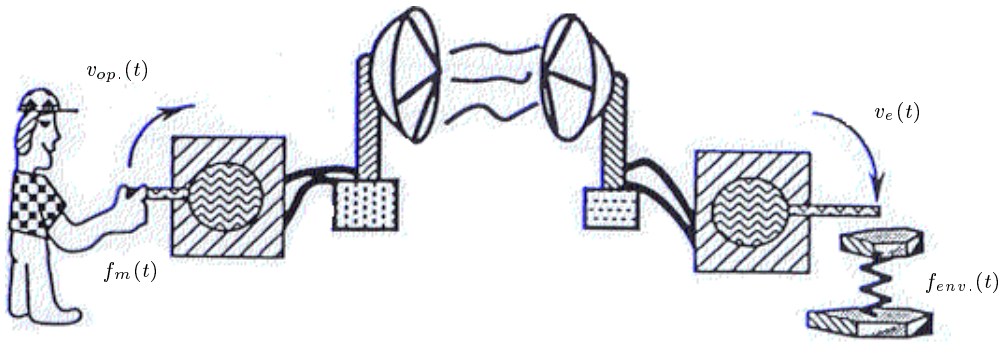
La téléopération apparaît ainsi au carrefour de la robotique industrielle et de la robotique plus « scientifique ». Elle intervient là où l'homme seul est soit inefficace, soit dans l'impossibilité d'agir (environnement hostile, faible accessibilité, forte variabilité, ...) et là où le robot autonome n'existe pas encore, faute d'intelligence suffisante

pour faire face aux contraintes de commande et de sécurité. Il s'agit d'un juste milieu où l'humain met à profit ses facultés intellectuelles nettement supérieures à celles du robot (analyse, savoir-faire, adaptabilité, prise de décision en temps réel, ...) et où le robot ajoute ses qualités de précision (résolution, justesse, ...), de force, et « d'infatigabilité ». Il soulage ainsi l'opérateur de contraintes opérationnelles et décisionnelles de bas niveau et lui permet de se concentrer sur la réalisation de la tâche au plus haut niveau.

1.1.2 Petit historique de la téléopération

La structure qui s'est vite imposée en terme de téléopération est la structure « *maître–esclave* » (voir figure 1.1). Elle est constituée de deux parties qui interagissent :

- l'opérateur commande son robot via une interface de commande, le *maître*, qui lui restitue le plus fidèlement possible les événements liés au robot,
- la partie opérative du robot, l'*esclave*, répond aux ordres du maître et lui transmet en retour les données qu'il a captées.



(a) Illustration (d'après [AND 92])

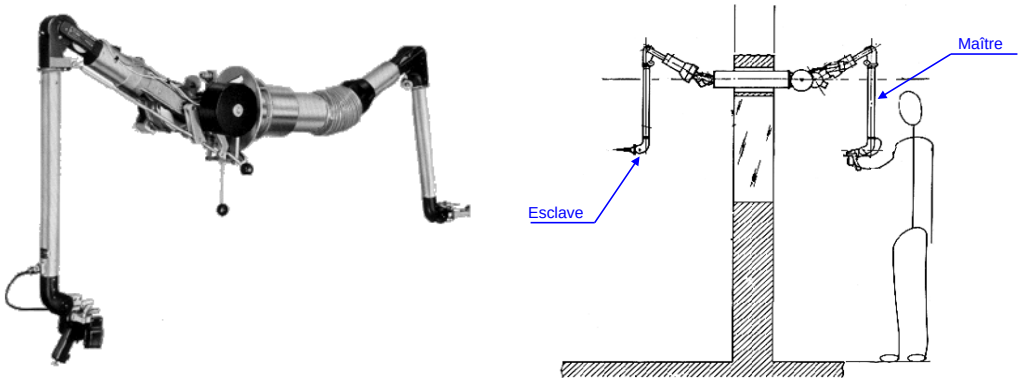
(b) Schéma bloc

Figure 1.1 – *Schéma d'une structure de téléopération maître–esclave classique*

Le maître n'est pas forcément une réplique de l'esclave à l'échelle près. A faible distance, ces dispositifs permettent, entre autres exemples, de manipuler des produits dangereux enfermés dans une caisse blindée (comme celui de la figure 1.2), ou de

commander un bras manipulateur pour déplacer des objets lourds. Dans ce dernier cas, l'échelle en effort est modifiée de manière à ce que n'importe quel opérateur humain puisse intervenir.

Ce type de structure permet un retour des efforts que doit développer l'esclave vers l'opérateur, ce qui lui offre une plus grande précision dans ses gestes. Ces efforts peuvent être de surcroît remis à l'échelle de l'opérateur lorsqu'il doit agir, par exemple, sur des micro-systèmes.



(a) Vue d'ensemble (b) Schéma

Figure 1.2 – *Exemple de télémanipulateur mécanique pour le domaine nucléaire*

La transmission entre le maître et l'esclave a d'abord été mécanique (engrenages, câbles, ...) : en 1948, GOERTZ et son équipe de l'Argonne National Laboratory ont créé un manipulateur pour l'expérimentation nucléaire en laboratoire [GOE 64]. L'opérateur possédait une vision directe sur la zone de travail à travers une paroi transparente. Les inconvénients propres à son système étaient son échelle limitée à 1:1 et le poids du manipulateur maître qui induisait une fatigue physique de l'opérateur. D'autre part, les distances maître–esclave étaient limitées par la technologie et la visibilité directe du poste de travail. Plus tard, des télémanipulateurs à transmission hydraulique ont été créés pour les esclaves devant exercer des efforts surhumains, comme les engins de chantier, par exemple.

En 1954, le même GOERTZ est passé à la télémanipulation électrique prenant la forme d'un asservissement bilatéral position–position. Le passage à une transmission électrique (filaire ou radio, analogique ou, plus récemment, numérique) ouvrait alors le champ à de nombreuses améliorations. Dès lors, les échelles géométriques et en efforts entre le maître et l'esclave ne restent plus limitées à 1:1, le rayon d'action s'accroît et l'électronique puis l'informatique viennent offrir de nouveaux horizons dans la commande de tels télémanipulateurs. Il devient alors possible de diminuer la fatigue physique de l'opérateur en compensant les effets de la gravité sur le dispositif maître par un algorithme de commande adéquat. De plus, l'opérateur a désormais la possibilité d'enregistrer des séquences de manipulations que le robot répétera automatiquement. Cependant, l'augmentation de la distance maître–esclave a pour inconvénient de sup-

primer la visualisation directe des objets à manipuler. L'opérateur doit alors avoir recours à un système de vidéo-surveillance induisant une fatigue psychique.

La distance entre le maître et l'esclave va imposer la technologie de transmission : plus cette distance grandit, plus une solution électrique puis numérique apparaissent les plus adéquates pour des questions de coût, de complexité et d'immunité au bruit du signal avec la distance. Malheureusement, en augmentant les distances, les délais de propagation rendent la téléopération de plus en plus difficile pour l'opérateur car il doit mentalement tenir compte des délais de transmission.

En 1965, Ferrel a démontré l'instabilité d'un télémanipulateur opéré via une commande bilatérale (reprenant le schéma de celle représentée en figure 1.1) en présence de délais de transmission de l'ordre de 0,1 s [FER 65].

En 1981, Vertut réussit à stabiliser un téléopérateur du type [FER 65] par une diminution de la bande passante des signaux échangés entre le maître et l'esclave au prix de vitesses limitées à 10 cm/s [VER 81]. En filtrant ces signaux, il évitait ainsi que le système ne s'accroche sur des fréquences de résonance rendant le système instable.

Il n'est pas rare de rencontrer des systèmes où les délais de transmission dépassent le dixième de seconde, rendant ainsi les téléopérateurs contemporains instables. En outre, le problème de la stabilisation d'un asservissement bilatéral en présence de retards, qui plus est parfois variables, est loin d'être évident à résoudre. C'est pourquoi, au cours des années 80, les roboticiens laissent un peu de côté cette quête mathématique de la stabilisation. Ils s'orientent plutôt vers des solutions donnant d'une part un peu d'autonomie à l'esclave et améliorent d'autre part le maître à l'aide de modules de prédiction afin de compenser les effets des retards sur les retours d'informations. Ainsi, en 1986, Sheridan fut parmi les premiers à proposer des « *affichages prédictifs* » et un « *contrôle superviseur* » [SHE 86] (cf. §1.3.4, page 28).

En 1988, Anderson et Spong [AND 88] se penchent alors sur les problèmes de stabilité dans un asservissement bilatéral en présence de délais de transmission. Ils modélisent la boucle de téléopération à l'aide des réseaux de Hilbert non linéaires (cf. annexe A). Ils obtiennent ainsi un réseau de quadripôles en série, à l'image des quadripôles d'un système électronique. En faisant appel à la théorie des lignes de transmission et de la stabilité développée par Lyapunov, ils démontrent qu'il est possible de rendre le système stable quels que soient les délais de transmission. Cela suppose toutefois que le maître et l'esclave sont initialement passifs et qu'il faut les compléter de telle sorte que leur norme $\mathcal{H}_\infty$ soit strictement inférieure à 1. Cependant ces auteurs ne donnent pas de méthode pour construire physiquement un maître et un esclave passifs, d'autant plus qu'un système obtenu par discrétisation d'un système continu via un échantillonneur bloqueur d'un système passif n'est pas automatiquement passif [LEU 92]. Ainsi cette méthode ne sera pas directement applicable pour des contrôleurs numériques et/ou via des canaux de transmission numériques.

En 1989, les mêmes roboticiens améliorent leur schéma de téléopérateur bilatéral continu en le rendant stable pour un retard constant donné [AND2 89]. Ils ont validé expérimentalement leur concept sur un système linéaire à un degré de liberté. Le système

était initialement instable dès 40 ms de retard ; une fois amélioré, le retard maximum supporté a atteint 200 ms. Cependant il faudra attendre 1992 pour démontrer la stabilité asymptotique des signaux de vitesse du maître et de l'esclave. Ils étendent alors leurs résultats à un système non linéaire à n degrés de liberté et démontrent la possibilité de changer l'échelle des puissances entre le maître et l'esclave sans altérer la passivité du maître et de l'esclave.

En 1990, NIEMEYER et SLOTINE [NIE 90] reprennent le travail précédent en utilisant un formalisme habituellement dédié aux guides d'ondes (cf. §1.3.1, page 19). Ils introduisent le concept d'impédance caractéristique et démontrent que les délais de transmission agissent sur les impédances vues respectivement depuis le maître et depuis l'esclave. Ils précisent comment ces derniers doivent être élaborés afin d'obtenir des caractéristiques de réflexion et de transmission adéquates.

C'est en 1993 que la première téléopération spatiale sol–navette a été réalisée avec le robot *ROTEX* [HIR 94]. Les délais de transmission aller–retour atteignaient 7 s car les signaux devaient transiter via des satellites pour obtenir une communication continue. La solution adoptée dans ce cas a fait appel à une interface en 3 dimensions où se superposaient des images vidéo en stéréo et un affichage virtuel permettant à l'opérateur d'effectuer des présimulations de ses manœuvres (cf. figure 1.3).

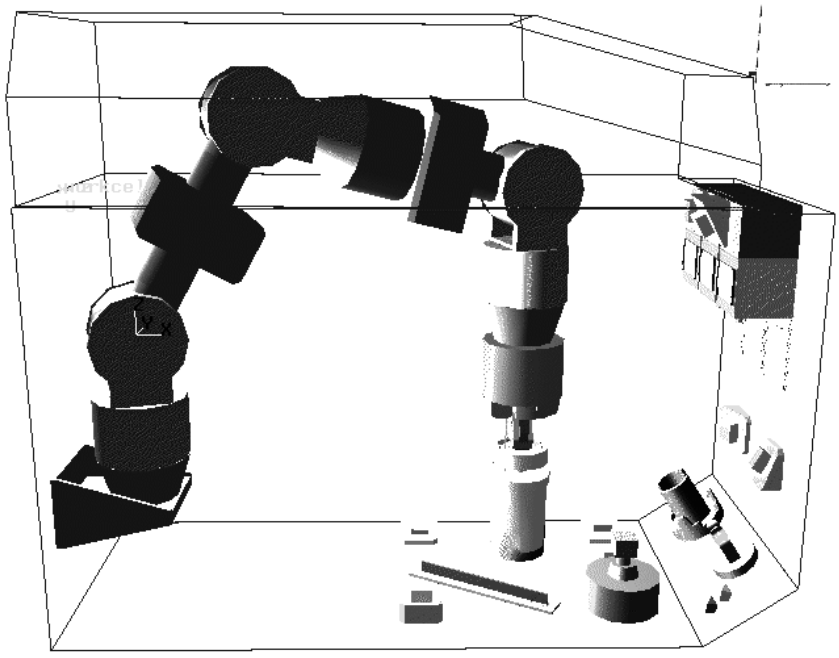
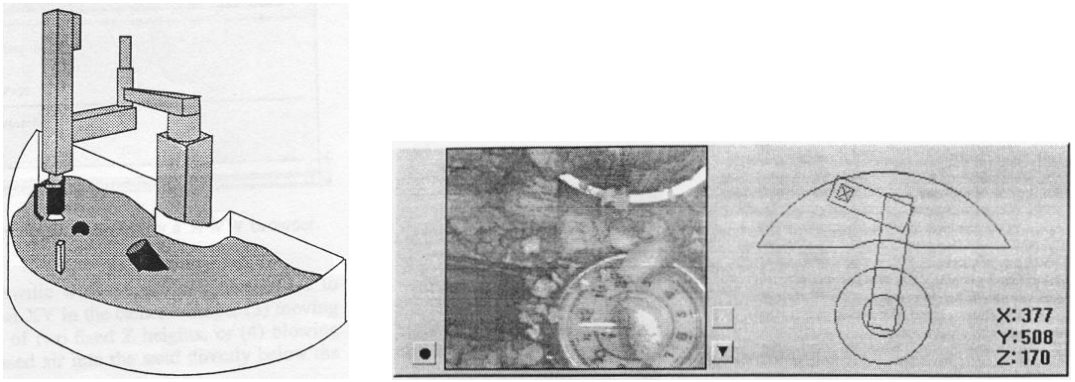


Figure 1.3 – *Représentation virtuelle du robot* ROTEX *[HIR 94]*

Dès 1995 fleurissent des expérimentations de téléopération de robots via l'*Internet* notamment aux Etats-Unis [GOL 95], [PAU 96] et en Australie [TAY 95]. Dans le cas du projet *Mercury* [GOL 95], l'expérimentation consiste en un jeu accessible au commun des internautes. Il s'agissait de résoudre une énigme en fouillant dans un bac à sable à

l'aide d'un robot de type *SCARA* équipé d'une caméra et d'une buse d'air comprimé (cf. figure 1.4) . L'interface était assez rudimentaire et peu ergonomique, limitée par les fonctionnalités des serveurs *WWW*. Il faudra attendre 1997 pour voir apparaître des interfaces sous *JAVA*, notamment dans [DEP 97] puis [GRA 00].



(a) Le robot (b) L'interface *WWW*

Figure 1.4 – *Environnement de « jeu » pour le robot de type SCARA [GOL 95]*

Simultanément, les techniques de téléprogrammation s'améliorent et atteignent le monde sous-marin comme dans [MAD 96] et [SAY 96]. Certains transforment les retours d'efforts en informations visuelles et sonores [MIT 95], d'autres font appel à des casques de réalité virtuelle pour une vision stéréo et des capteurs de mouvement de différentes parties du corps de l'opérateur afin d'envoyer des consignes plus évoluées à l'esclave [POO 95]. Le principe de surimpression de mouvements simulés du manipulateur sur une image vidéo en 3 dimensions [RAS 96] se généralise (cf. figure 1.5).

En 1996, Tarn et Brady ont proposé un contrôleur intéressant, s'adressant à des retards variables bornés et dont les variations sont également bornées [TAR 96]. Ils s'appuient sur un observateur développé dans [WAT 81].

En 1996 également, Kheddar, Tzafestas et Coiffet innovent en matière de téléprogrammation en téléopérant simultanément depuis le LRP[1] quatre robots différents localisés à Poitiers, Grenoble, Nantes et Tsukuba au Japon. La tâche consistait à assembler un puzzle de quatre pièces en utilisant une interface ayant un niveau d'abstraction élevé ; le concept de « robot caché » [KH2 97] a permis à l'opérateur d'accomplir la tâche en agissant dans un environnement simulé directement avec ses mains d'une manière naturelle.

En 1997, Leung [LEU 97] propose une méthode pour développer un contrôleur dans le cadre d'une boucle de téléopération bilatérale avec des retards plafonnés. Cette méthode fait appel à la commande $\mathcal{H}_\infty$ [DEL 93] et une méthode nommée μ-analyse. Cette méthode ne suppose pas la passivité du maître ou de l'esclave et est applicable

1. Laboratoire de Robotique de Paris (http://www.ccr.jussieu.fr/lab/p6/ufr923/lab7/e.html)

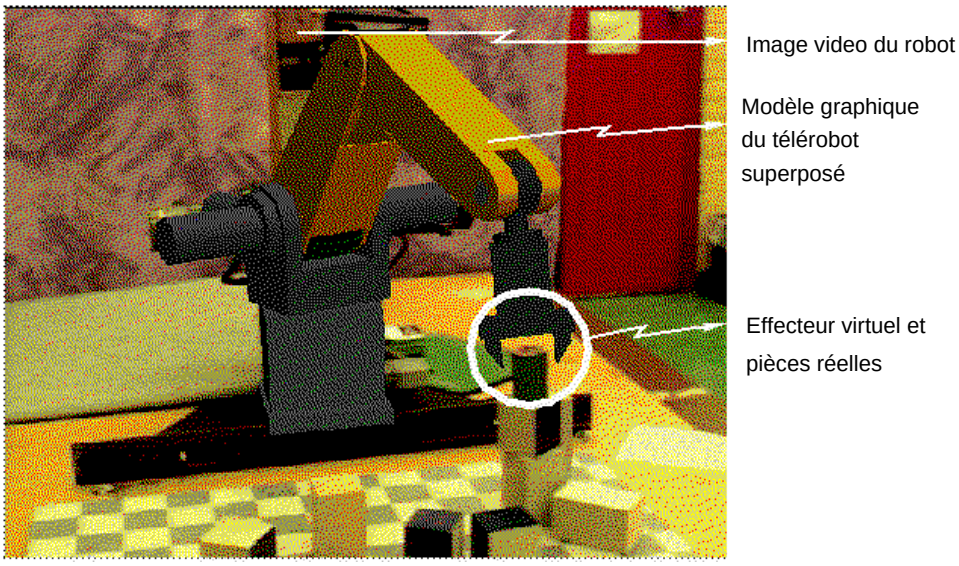


Figure 1.5 – *Superposition du modèle virtuel sur une image vidéo stéréoscopique [RAS 96]*

aux mouvements libres d'un manipulateur ainsi que pour les mouvements en interaction avec l'environnement

Jusqu'à nos jours, la recherche en téléopération s'est concentrée sur deux aspects complémentaires, d'une part l'amélioration des commandes bilatérales (par exemple [KOS 97], [NIE2 97] et [YOK 00]) et, d'autre part l'amélioration des commandes par téléprogrammation (pour ne citer que [TSU 97], [TUR 97] et [BAL 98]). Toutefois, l'équipe d'Ando s'est penchée en 1999 sur les problèmes de perception des retards par l'opérateur dans le cadre de téléopérations [AND 99]. Elle note trois modes de fonctionnement de l'opérateur en fonction de la taille des délais de transmission.

- le mode opératoire temps–réel lorsque les délais sont inférieurs à quelques millisecondes ; l'opérateur ne se rend pas compte de la présence de retards,
- le mode opératoire avec perception du retard ; ce dernier se situe alors entre quelques millisecondes et plusieurs dixièmes de secondes. L'opérateur prédit les conséquences de ses actions au fur et à mesure sans arrêter la téléopération,
- le mode opératoire avec estimation du retard ; au-delà de la seconde, l'opérateur utilise une technique de mouvement puis d'attente de la perception des conséquences de ce mouvement : scénario « *move and wait* ».

Cependant, ces expériences n'ont pas abouti à des résultats permettant d'améliorer la réalisation d'interfaces homme–machine plus adaptées au profil psychologique d'un opérateur humain.

Notons également l'apparition de la télémédecine dans les thèmes de recherche liés à la téléopération dès 1998 et l'expérience *SHISHA98* [GOU 99] dans le cadre du projet

SYRTEC[2] (cf. figure 1.6).

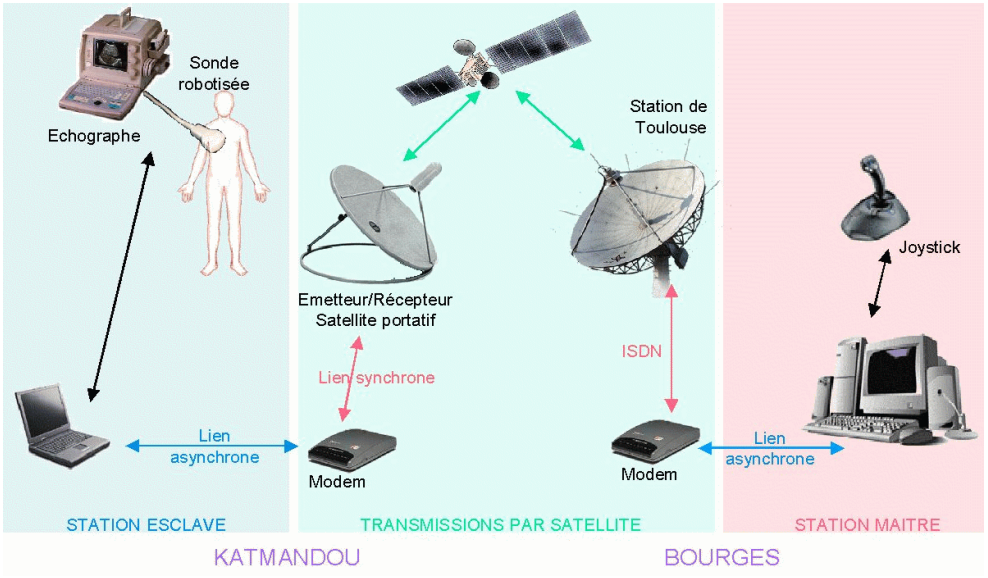


Figure 1.6 – *Contexte de téléopération pour l'expérience* SHISHA98 *[GOU 99]*

1.2 Applications

Il existe trois grands domaines particulièrement concernés en matière de téléopération.

– **Le domaine spatial** : la NASA a commencé à développer la téléopération pour explorer des planètes lointaines. *Rocky 7*, le «*rover*[3] » qui a exploré la planète Mars (figure 1.7(a)) est sans doute le système téléopéré le plus célèbre à ce jour. Il s'agit de la téléopération à longue distance la plus spectaculaire et la plus difficile à résoudre. La NASA a aussi effectué la téléopération depuis la terre d'un bras manipulateur situé sur la navette *Columbia* [HIR 94]. La navette étant en orbite basse (300 à 500 km de la surface de la terre), le temps de trajet aller d'un message est de 0,25 s mais l'émetteur–récepteur terrestre ne voit la navette que pendant 10 à 20 minutes toutes les 2 heures. L'utilisation de relais par satellites a permis de garantir la liaison radio quelle que soit la position de la navette. Hélas, elle a fait grimper le temps de transmission moyen à 6 s avec des variations rapides autour de cette moyenne de l'ordre de quelques centaines de millisecondes et des variations lentes de l'ordre de la seconde. Plus récemment, les chercheurs du JPL[4] ont téléopéré un rover sur Mars lors de la mission « *Mars Polar Lander* » [BAC 00].

2. Tele-Scanning Robot System
3. issu de *ROV* : « Remotely Operated Vehicle », autrement dit un véhicule téléopéré
4. *Jet Propulsion Laboratory*: laboratoire de recherche lié à la NASA à Pasadena en Californie.

- **Le monde sous-marin** : la recherche de nouveaux gisements pétrolifères implique l'exploration des hauts-fonds. Une introduction assez complète sur les problèmes liés à la téléopération sous-marine est développée dans [MAD 96]. Si, dans le cas spatial la distance est le principal facteur de retard, ici, c'est le milieu aquatique lui-même qui se prête mal aux transmissions. L'équipe de SAYERS a effectué des expérimentations grandeur nature consistant à téléopérer un véhicule immergé à une profondeur de 7 m depuis un laboratoire situé à 500 km par l'intermédiaire d'un bateau relais situé en surface non loin du submersible [SAY 96]. Ses modems acoustiques ont permis une communication entre le bateau et le sous-marin à un débit de l'ordre de 10 kbits/s avec un délai de transmission aller–retour global de l'ordre de 10 s. L'IFREMER possède plusieurs *ROV* pour l'exploration des fonds sous-marins. La figure 1.7(b) présente l'un d'entre eux : *Victor 6000*. Il s'agit d'un engin à câble piloté à partir d'un navire support. Il a été conçu pour faire de l'investigation optique et effectuer des missions locales incluant l'imagerie, la mise en œuvre d'instrumentation ainsi que des prélèvements d'eau, de sédiments ou de roches.
- **Le domaine nucléaire** s'intéresse de près à la téléopération afin d'effectuer des manipulations dans des endroits exposés aux radiations. Ainsi le CEA a développé *Sherpa* (visible en figure 1.7(c)), un hexapode pour ambiance hostile, capable de se faufiler à l'intérieur d'une installation nucléaire par des chemins initialement prévus pour l'homme, de soulever et d'emporter des charges lors d'interventions, d'amener des bras manipulateurs en position de travail pour réparer ou remplacer du matériel en panne.

(a) Rocky 7 (b) Victor 6000 (c) Sherpa

Figure 1.7 – *Exemple de robots téléopérés en milieu hostile*

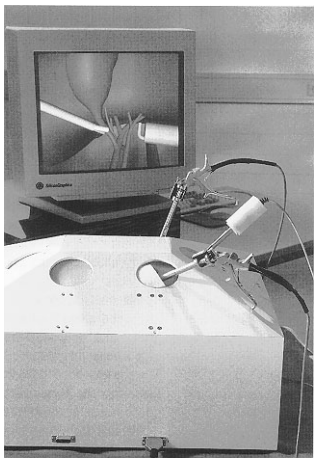
En dehors de ces trois domaines, particulièrement importants car correspondant à un environnement peu accessible à l'homme, d'autres applications apparaissent dans des domaines plus variés :

- La télémédecine[5] : en 1998, par exemple, une téléopération par satellite a permis à des médecins situés à Bourges d'effectuer des relevés médicaux par ultra-sons sur

5. http://www.acl.lanl.gov/TeleMed/

un alpiniste escaladant le mont Shisha (8000 m d'altitude) au Népal [GOU 99]. D'autre part, la figure 1.8 illustre l'intérêt de la téléopération dans les nouvelles disciplines chirurgicales en montrant un simulateur pour une opération endoscopique téléopérée et une vue d'artiste d'une salle d'opération futuriste. Dans ce domaine, des équipes de recherche telles que celle de TANIMOTO [TAN 98] se concentrent sur la sécurité, indispensable lors d'opérations neurochirurgicales téléopérées.

- La télémaintenance : entre autres exemples, Hydro-Québec (l'E.D.F. du Québec) étudie cet outil afin de surveiller et de réparer ses installations électriques réparties sur son immense territoire.
- La gestion d'un parc automobile en libre-service, en faisant appel à la téléopération des véhicules vides afin de les amener à des points de rendez-vous.



(a) Simulateur pour endoscopie

(b) Salle d'opération du futur

Figure 1.8 – *La téléchirurgie : une branche récente pour la téléopération*

Si nous sortons du cadre de la robotique, nous nous apercevons que la téléopération a d'autres acceptions :

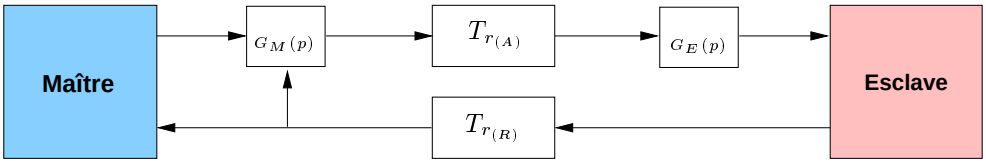
- l'assistance en ligne aux clients («*help desk*» ou «*hot line*»),
- la commande d'enquêtes et de sondages à distance, le plus couramment par téléphone.

1.3 Les différentes familles

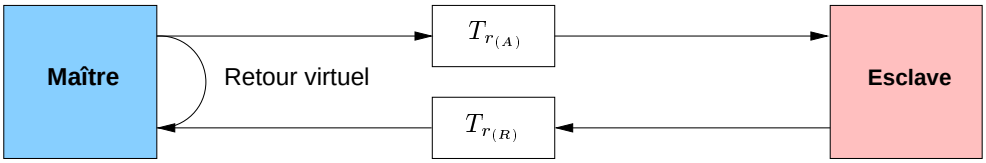
Dans leur article [PAU 92], PAUL et al. proposent quatre grandes approches des problèmes liés au retard en téléopération :

- implantation d'un asservissement dont la boucle inclut le maître, le médium de transmission et l'esclave (cf. figure 1.9(a)) ; valable en général pour des petits

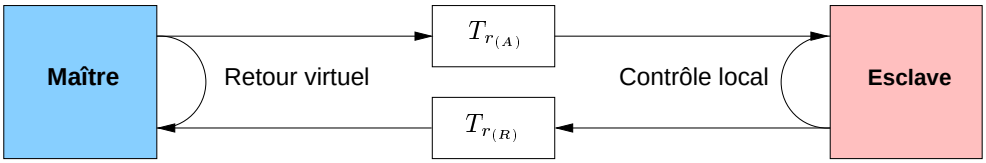
retards peu variables,
- prédiction des mouvements de l'esclave et affichage prédictif via une interface homme–machine améliorée (cf. figure 1.9(b)),
- contrôle partagé : l'esclave gagne en autonomie en étant capable d'effectuer des manœuvres simples en contact et en mouvement d'approche avec son environnement (cf. figure 1.9(c)),
- contrôle superviseur : ce ne sont plus des consignes qui sont transmises mais des ordres symboliques, par exemple, « visser écrou », « atteindre telle position », ... (cf. figure 1.9(d)).



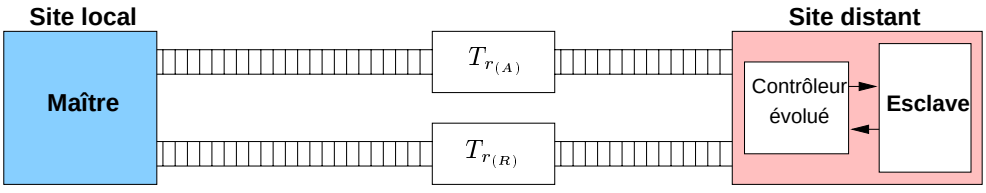
(a) Commande à boucle globale



(b) Affichage prédictif et Interface Homme–Machine avancée



(c) Contrôle partagé (shared control)



(d) Contrôle superviseur (supervisory control)

Figure 1.9 – *Quatre grandes approches de la téléopération à longue distance (d'après [STE94])*

1.3.1 Commandes à boucle globale

Ce type de commande maintient une boucle d'asservissement intégrant le maître, l'esclave et les délais de transmission.

Étude en automatique classique

En automatique classique, un retard pur τ peut se modéliser par une fonction de transfert $H_r(p)$ dans le domaine de LAPLACE (cf. équation 1.1).

$$\mathcal{L}\left[f(t-\tau)\right] = e^{-p.\tau}.F(p) \quad \Rightarrow \quad H_r(p) = e^{-p.\tau} \tag{1.1}$$

Cette fonction de transfert entraîne un déphasage $\Delta\Phi$ (équation 1.2) qui diminue très rapidement la marge de phase qui devient difficile à déterminer pour des systèmes d'ordre supérieur à 1.

$$\Delta\Phi = \omega.\tau \tag{1.2}$$

Si nous souhaitions compenser parfaitement ce retard pur, il faudrait ajouter un prédicteur pur $C(p) = e^{+p.\tau}$, ce qui n'existe malheureusement pas (non causal).

Prédicteur de Smith

Une solution proposée dans [DEL 93] utilise un « *prédicteur de* SMITH » (représenté figure 1.10). Il s'agit d'une méthode qui s'apparente au *modèle interne* [DEL 93]. Le prédicteur doit connaître les retards aller et retour supposés constants (ou tout au moins lentement variables), en l'occurrence τ, ainsi qu'un modèle (ne serait-ce qu'approché) $E^*(p)$ du processus à asservir $E(p)$. Le contrôleur doit posséder un correcteur capable d'asservir le processus en l'absence de retard.

$Y(p)$ est le signal utilisé par le contrôleur pour asservir le processus.

$$Y(p) = \left[E^*(p) + e^{-2.\tau.p}(E(p) - E^*(p))\right].U(p)$$

Si $E^*(p)$ modélise parfaitement le processus $E(p)$, alors:

$$Y(p) = E^*(p).U(p) = E(p).U(p)$$

Le contrôleur peut ainsi agir comme s'il n'y avait aucun délai de transmission. Un simple correcteur de type *P.I.D.* peut suffire à contrôler correctement le processus.

Cependant, $E^*(p)$ n'est, dans la réalité, jamais parfait ; d'autre part le prédicteur doit connaître parfaitement le retard τ, ce qui augmente nettement la marge d'imprécision de l'estimation.

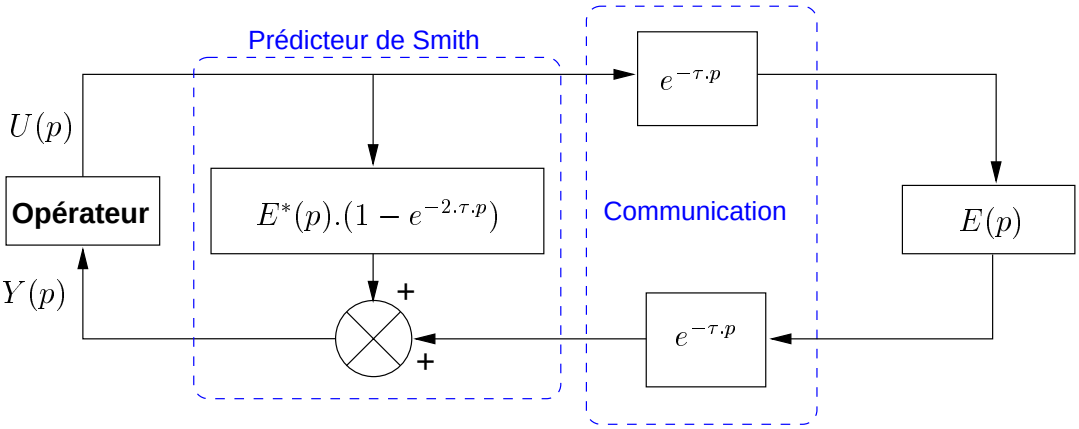


Figure 1.10 – *Prédicteur de* Smith

Commande bilatérale

En général, le fait d'asservir la position ou la vitesse du téléopérateur ne suffit pas pour commander un système à distance. Si ce type d'asservissement peut se révéler suffisant dans le cas de mouvements libres du système téléopéré, il n'est pas assez évolué pour traiter des mouvements d'approche ou en contact avec l'environnement.

Une *commande bilatérale* est un asservissement qui couple une partie des variables d'état du dispositif maître à une partie correspondante des variables d'état de l'esclave.

Il existe quatre combinaisons dans le cadre de la commande de robots :

- le maître est commandé en position, l'esclave aussi,
- le maître est commandé en position, l'esclave en force,
- le maître est commandé en force, l'esclave en position,
- le maître et l'esclave sont commandés en force.

Les tâches réservées à la téléopération nécessitent souvent que le manipulateur interagisse avec son environnement, en entrant en contact physique avec celui-ci (par exemple, suivi de surface et assemblage : opérations représentées en figure 1.11). La commande bilatérale permet ainsi de commander en effort (ou en position) et en position (ou en effort) le manipulateur distant et apporte ainsi un confort à l'opérateur rapidement devenu indispensable. Elle est donc un « classique » en téléopération.

L'inconvénient de cette commande est qu'elle est très sensible aux délais de transmission.

Pour pouvoir étudier les problèmes posés par les retards dans une commande bilatérale, il a d'abord fallu trouver une description adaptée à l'ensemble des composants d'une boucle de téléopération. Pour cela, le maître et l'esclave sont modélisés d'un point de vue mécanique à l'aide d'une masse, d'un ressort et d'un amortisseur (cf. figure 1.12).

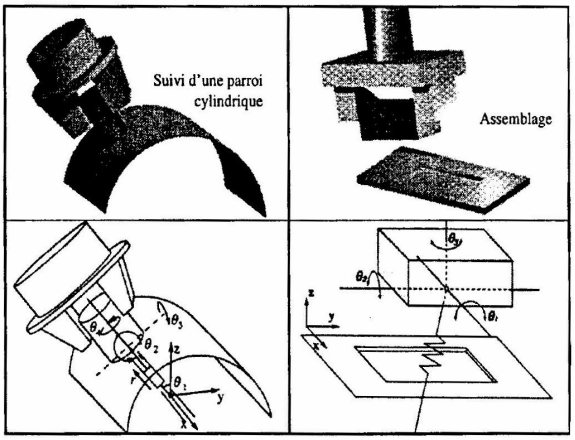


Figure 1.11 – *Représentations virtuelles d'opérations en interaction avec l'environnement (d'après [KH2 97])*

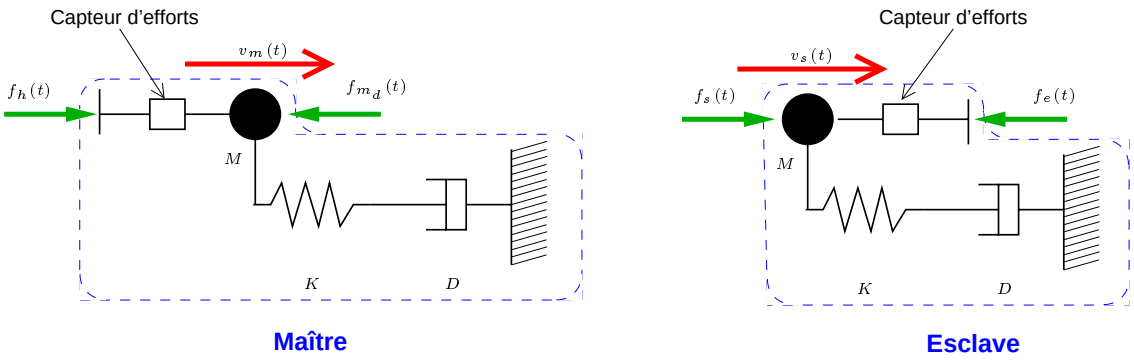


Figure 1.12 – *Modèles du maître et de l'esclave avec un réseau de* Hilbert

Anderson et Spong [AND 92] ont songé à une certaine catégorie de réseaux de Hilbert non linéaires, les réseaux de type *PHIDE* (cf. annexe A). Il leur a ainsi été possible de modéliser entièrement la chaîne de téléopération pour un retard T_r constant et de rendre le maître et l'esclave passifs vis-à-vis du bloc de transmissions qui est lui, déjà passif par nature (cf. figure 1.13). En opérant ainsi, le système est stable pour toute valeur de T_r.

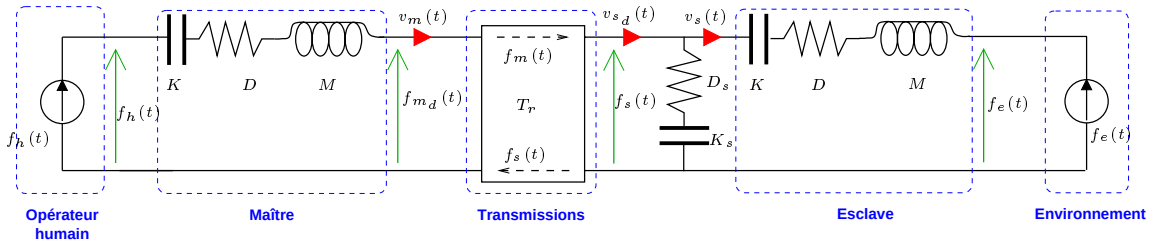


Figure 1.13 – *Chaîne de téléopération modélisée en réseau de* Hilbert

Ce modèle, dans lequel le maître et l'esclave possèdent la même dynamique modélisée par les équations (1.3) et (1.4), est représenté en figure 1.12.

$$M.\dot{v}_m(t) + D.v_m(t) + K.\int_0^t v_m(\tau).\mathrm{d}\tau \quad = f_h(t) - f_{m_d}(t) \tag{1.3}$$

$$M.\dot{v}_s(t) + D.v_s(t) + K.\int_0^t v_s(\tau).\mathrm{d}\tau \quad = f_s(t) - f_e(t) \tag{1.4}$$

$v_m(t)$ et $v_s(t)$ sont les vitesses respectives du maître et de l'esclave. M, D et K correspondent à leur masse, leur coefficient de viscosité et leur raideur. $f_h(t)$ est la force appliquée au maître par l'opérateur, $f_e(t)$ la force exercée par l'environnement sur l'esclave. $f_s(t)$ est la force qui permet au maître et à l'esclave de se suivre mutuellement :

$$f_s(t) = K_s.\int_0^t \left(v_{s_d}(\tau) - v_s(\tau)\right).\mathrm{d}\tau + D_s.\left(v_{s_d}(t) - v_s(t)\right) \tag{1.5}$$

avec $v_{s_d}(t)$ l'information de vitesse du maître que l'esclave reçoit.

Cette étude ne s'adresse qu'aux systèmes continus. Cependant, en utilisant la transformation bilinéaire (cf. annexe A), il est possible de modéliser un système échantillonné.

Ce type de commande a été testé avec succès par les auteurs de [KOS 97] sur un système à un degré de liberté avec une période d'échantillonnage de 2 ms.

Notion d'outil virtuel ou contrôle en impédance

La présence de retards de transmission impose des limitations fondamentales des performances d'un télérobot, indépendamment de l'aspect technique. En particulier, la

réaction à une perturbation inconnue ou à un contact avec l'environnement ne peut pas avoir d'effet avant le temps de parcours aller–retour de l'information. Ainsi la bande passante de la boucle sera limitée par cet aspect.

Cependant, il est très important de présenter à l'opérateur un système simple et prévisible. Chaque comportement inattendu risque de compliquer singulièrement le travail de l'opérateur qui aurait pu se concentrer exclusivement sur la tâche à accomplir. L'idéal serait de pouvoir comparer le système à un jeu vidéo, simple à utiliser mais possédant assez de fonctions pour atteindre le but escompté.

Niemeyer et Slotine ont ainsi proposé de créer un *outil virtuel* que l'opérateur manipule depuis son pupitre de contrôle. Ce nouvel outil transforme le télérobot en un système aux dynamiques simples et prévisibles [NIE2 97].

Cet outil virtuel possède des caractéristiques inertielles et de raideur qu'il est possible d'adapter aux délais de transmission. Par exemple, un faible délai donne un outil virtuel équivalent à un tournevis: léger et rigide. Au fur et à mesure que le retard augmente, l'outil devient de plus en plus lourd et/ou mou à l'image d'une perceuse et/ou d'une éponge. D'autre part, pour un mouvement libre, il est possible de paramétrer l'outil de manière à obtenir un mouvement rapide et « mou ». A l'inverse, lors d'une phase de contact, il est plus intéressant de manipuler un outil lent mais rigide de façon à bien contrôler les efforts de contact.

Il est important en général de reconnaître les limitations induites par les délais de transmission et de réduire les performance de l'outil virtuel en conséquence.

L'approche par variables d'ondes (cf. annexe B) permet justement un ajustement automatique. Elle complète l'approche précédente car elle permet non seulement de créer des systèmes robustes aux retards, mais encore de placer des éléments en cascade sans les problèmes classiques de causalité liés aux admittances/impédances. Cette approche se fonde uniquement sur des concepts de puissance et d'énergie et est applicable à des systèmes non linéaires et peut gérer des modèles inconnus ou incertains. De ce fait, elle convient bien aux interactions avec un environnement physique réel.

Considérons un esclave incluant un contrôleur Proportionnel–Dérivé (B et K sont des matrices de gain, symétriques définies positives et constantes) afin que $\dot{x}_s(t)$ et $x_s(t)$ suivent respectivement les consignes $\dot{x}_{s\,d}(t)$ et $x_{s\,d}(t)$ (cf. figure 1.14). L'opérateur impose $\dot{x}_m(t)$ et reçoit en retour une représentation $f_m(t)$ de l'effort que doit fournir l'esclave pour accomplir le mouvement.

$$f_s(t) = -B.\big(\dot{x}_s(t) - \dot{x}_{s\,d}(t)\big) - K.\big(x_s(t) - x_{s_d}(t)\big) \tag{1.6}$$

La figure 1.14 illustre les différents trajets des ondes ; trois modes y sont représentés :

- A chaque mouvement induit par l'opérateur, correspond un retour direct sous la forme d'un amortissement $b.\dot{x}_m(t)$.
- Au niveau du bloc de transmissions, une partie de chaque onde incidente $v_m(t)$ et $u_s(t)$ est réfléchie. Ces ondes réfléchies s'ajoutent aux ondes $u_s(t)$ et $v_m(t)$

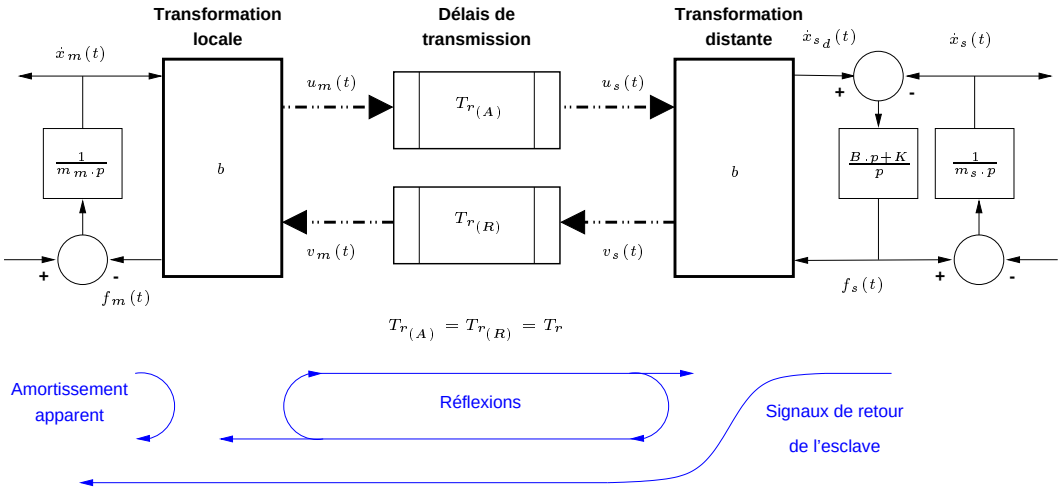


Figure 1.14 – *Modélisation d'une boucle de téléopération simple à l'aide des variables d'ondes*

qui émettent à leur tour des ondes réfléchies à leur arrivée aux extrémités du bloc de transmission. Ces réflexions ne contiennent pas d'information intéressante et peuvent mettre plusieurs cycles avant de « mourir ». Elles créent ainsi des perturbations qui peuvent dégrader les performances des divers asservissements.

– Enfin, le dernier trajet correspond au retour des informations de l'esclave vers le maître : vitesse réelle de l'esclave $\dot{x}_s(t)$ et efforts $f_s(t)$ fournis pour réaliser le mouvement.

Tous les éléments de ce système sont passifs, il est donc stable indépendamment du délai constant T_r. Cependant, sous cette forme, les données transmises contiennent une combinaison des vitesses et des forces. Autrement dit, aucune mesure de position n'est transmise directement. Théoriquement, l'asservissement en position est faisable mais du fait que ce sont des vitesses et non des positions qui sont véhiculées, le cumul des erreurs numériques lors des diverses transformations et de l'intégration des vitesses peut engendrer une dérive progressive entre la position du maître et celle de l'esclave.

Il existe deux solutions à ce problème. La première solution a été proposée par Niemeyer et Slotine dans [NIE1 97] et consiste à transmettre une version intégrée des signaux d'ondes $u_m(t)$, $u_s(t)$, $v_m(t)$, $v_s(t)$ en plus des signaux d'ondes eux-mêmes. En effet ces signaux possèdent l'information de position et peuvent être construits directement sans nécessiter d'intégration.

La seconde solution a été présentée par les mêmes auteurs dans [NIE2 97]. Elle consiste en une correction du signal $u_m(t)$ en fonction de la dérive mesurée entre $x_{s_d}(t)$ et $x_m(t)$.

Les diverses réflexions parasites représentées sur la figure 1.14 sont dues à une inadéquation des impédances du maître et de l'esclave par rapport au bloc de transmissions. Ce dernier possède une impédance b que l'opérateur peut modifier si besoin est. Afin d'éviter des réflexions parasites, il faut adapter les impédances du maître et de l'esclave

à celle du bloc de transmission. Pour ce faire, il suffit de paramétrer correctement deux correcteurs P.D., un au niveau du maître et l'autre au niveau de l'esclave (cf. figure 1.15)

Du point de vue de l'opérateur, le système peut être caractérisé uniquement par sa masse M_{app}, sa raideur K_{app} et son amortissement B_{app} qui sont donnés dans ce cas par les relations :

$$M_{app} = M_m + b.T_r + M_s \tag{1.7}$$

$$B_{app} = 2.b \tag{1.8}$$

$$K_{app} = {K_m}^{-1} + \left(\frac{b}{T_r}\right)^{-1} + {K_s}^{-1} \tag{1.9}$$

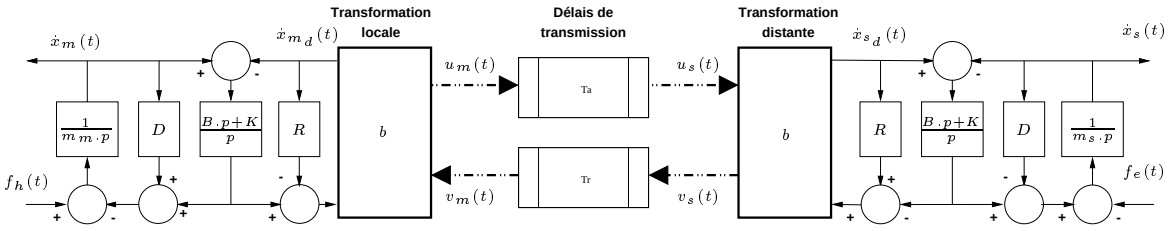


Figure 1.15 – *Adaptation d'impédance à l'aide de deux correcteurs P.D.*

Ces équations permettent de comprendre l'effet de l'impédance caractéristique b et du retard T_r sur le comportement apparent du système. L'impédance b est un compromis entre une grande inertie et une grande rigidité. Le retard détériore ces deux aspects de manière identique en augmentant la masse apparente et en réduisant la raideur.

Cas des retards variables

Ces méthodes de stabilisation ont pour hypothèse la constance du retard. Or les moyens de transmission informatique habituels ne garantissent pas cette constance. D'autre part des systèmes stables en présence de retards constants peuvent devenir instables pour de très faibles variations de ces retards. Qui plus est, les temps de transmission aller–retour ne sont pas identiques.

Pour palier ce problème, l'équipe de KOSUGE propose dans [KOS 96] une compensation des variations de retards semblable à celle développée au chapitre 3. Cette compensation passe par une première phase d'audit du réseau pour déterminer le retard mesuré le plus élevé, noté $T_{r_{max}}$. Ensuite, les trames de données sont émises à période constante multiple de la période d'échantillonnage du maître et de l'esclave.

Quand une trame est émise, par exemple, par le maître à l'instant t_0, elle est réceptionnée par l'esclave après un délai de $\Delta t = t_0 - t_r(t_0)$. Si Δt est inférieur à $T_{r_{max}}$, la trame est conservée pendant une durée $T_{r_{max}} - \Delta t$ dans une file d'attente. Ainsi le retard global de chaque trame est égal à $T_{r_{max}}$. En réalité, le choix de $T_{r_{max}}$ est fixé à

une valeur légèrement inférieure au retard maximum observé pendant la phase d'audit du réseau car la probabilité qu'un tel retard survienne est faible. Si un retard supérieur à $T_{r_{max}}$ advient, alors les valeurs de la trame précédente sont fournies et la trame en retard éliminée. Nous supposons que le récepteur est capable, avant de réguler les trames incidentes, de les remettre dans l'ordre d'émission ; ce détail n'est pas spécifié dans l'article. Les auteurs prévoient une adaptation de $T_{r_{max}}$ en fonction de l'évolution des délais de transmission afin de moduler $T_{r_{max}}$ en fonction des conditions de transmission sur une longue période d'expérimentation. Pour rendre le bloc de transmission passif (cf. annexe A), ils imposent un temps de vol aller $T_{r_{max_{(A)}}}$ identique au temps de vol retour $T_{r_{max_{(R)}}}$: $T_{r_{max_{(A)}}} = T_{r_{max_{(R)}}} = T_{r_{max}}$

Des expérimentations leur ont permis d'observer des délais de transmission maître vers esclave variant entre 0 et 705 ms avec 95% des retards mesurés inférieurs à 495 ms, d'où un choix de $T_{r_{max}} = 495$ ms, en estimant un taux de dépassement à 5%.

Notons que cette méthode de calcul de $T_{r_{max}}$ nécessite de pouvoir calculer le temps de vol d'une trame entre le maître et l'esclave ; or, si ceux-ci sont éloignés, il est difficile de synchroniser leur horloge à la milliseconde près. D'autre part, cette méthode ne tient pas compte des variations des retards, uniquement du maximum de ces retards ; ainsi si, par malchance, lors de la phase d'audit du réseau, un pic nettement supérieur au mode principal des retards est observé, $T_{r_{max}}$ sera exagéré par rapport à la globalité des retards observés. L'adaptation de $T_{r_{max}}$ aux conditions de trafic peut limiter les effets d'une telle sur-estimation mais l'article ne stipule que des adaptations destinées à contrer des hausses de trafic, pas de baisse. Le fait d'imposer le même retard global à l'aller et au retour peut pénaliser une liaison asymétrique ; cette condition est indispensable pour la passivité du bloc de transmissions mais ne l'est plus pour des commandes ne faisant pas intervenir la notion de passivité. Enfin, il n'est pas toujours acceptable de perdre des informations, même 5% ; tout dépend de la nature de la tâche que l'esclave doit effectuer.

Les auteurs de [KOS 97] effectuent un retour en arrière en se contentant d'un échantillonneur bloqueur d'ordre 0 paramétré à la période d'émission à la place du régulateur de retards. Ils ne démontrent pas la stabilité de leur système mais fournissent des résultats d'expérimentations satisfaisants.

L'équipe de YOKOKOHJI a proposé son propre compensateur de retards dans [YOK 00]. Sa méthode est représentée figure 1.16 et est développée dans le cas d'une émission du maître vers l'esclave.

Quand une onde est émise par un des émetteurs (le maître ou l'esclave), elle est accompagnée de sa date de départ enregistrée dans le signal $t_m^{last}(t)$. En considérant les primitives du signal émis $u_m(t)$ et du signal retardé $(\tilde{u}_s(t))$ puis compensé $\breve{u}_s(t)$, les auteurs montrent qu'il est possible de restaurer le signal initial $u_m(t)$.

Il suffit d'effectuer un asservissement proportionnel sur l'erreur entre les intégrales de $u_s(t)$ et $\breve{u}_s(t)$ (K est une matrice de gain diagonale et définie positive) :

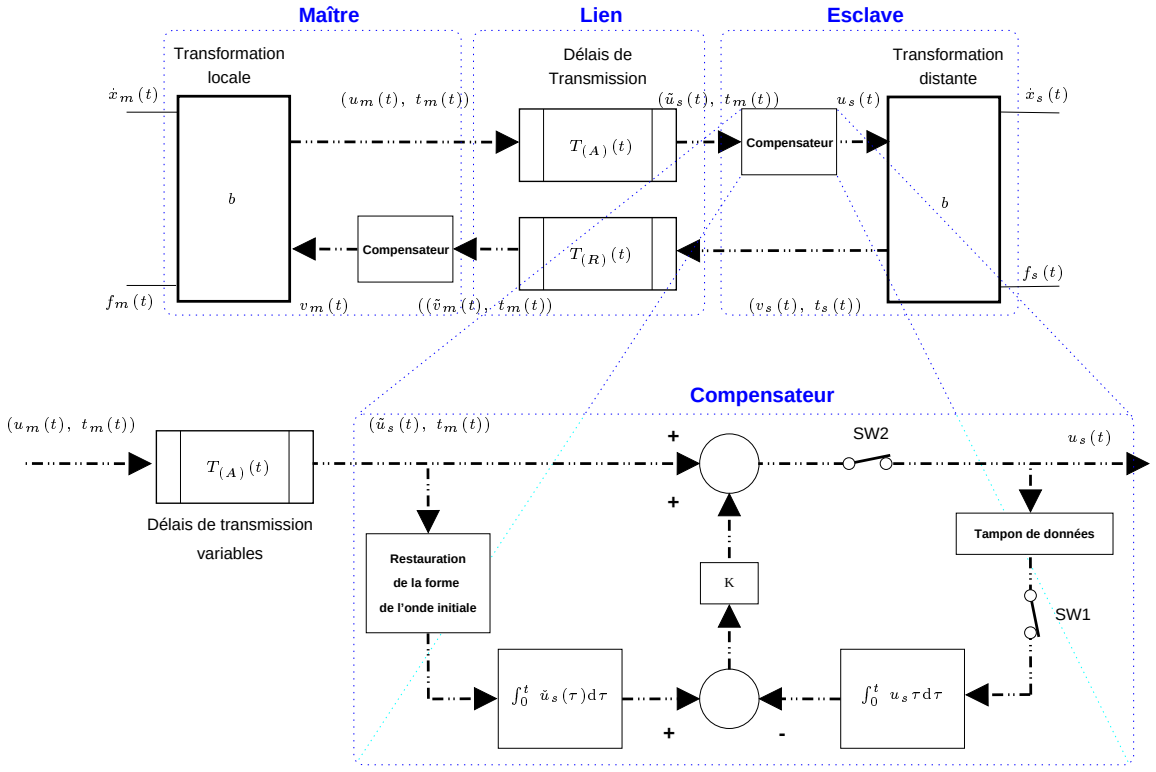


Figure 1.16 – *Structure globale de compensation des retards proposée par* YOKOKOHJI

$$u_s(t) = \tilde{u}_s(t) + K.\left(\int_0^t \check{u}_s(\tau).\mathrm{d}\tau - \int_0^t u_s(\tau).\mathrm{d}\tau\right) \tag{1.10}$$

Un gain K élevé permet au système de converger rapidement mais le système devient très sensible aux variations de retards. Le gain K est donc à déterminer selon ce compromis.

Il est toutefois possible de limiter cette sensibilité aux fluctuations des retards en introduisant un second retard $\overline{T_{r_{(A)}}(t)}$ tel que :

$$u_s(t) = \tilde{u}_s(t) + K.\left(\int_0^t \check{u}_s(\tau).\mathrm{d}\tau - \int_0^{t_s^{lim}(t)} u_s(\tau).\mathrm{d}\tau\right) \tag{1.11}$$

avec $t_s^{lim}(t)$ défini par :

$$t_s^{lim}(t) = \begin{cases} t & \text{si } t \leqslant t_m^{last}(t) + \overline{T_{r_{(A)}}(t)}, \\ t_m^{last}(t) + \overline{T_{r_{(A)}}(t)} & \text{sinon.} \end{cases} \tag{1.12}$$

Dans l'équation (1.11) $u_s(t)$ est intégré jusqu'à l'instant $t_m^{last}(t)$ à la différence de l'équation (1.10) où $u_s(t)$ est intégré jusqu'à t même si la transmission est arrêtée. Cette méthode consiste à ajouter l'interrupteur $SW1$ sur la figure 1.16. Il est possible de déterminer $T_{r_{(A)}}(t)$ en effectuant une moyenne à horizon fini de manière à adapter cette compensation en fonction des variations à long terme des retards du lien de transmissions.

D'autre part les auteurs proposent un contrôle de la puissance consommée par le système afin d'assurer la passivité du système dans toutes les circonstances et de limiter les conséquences d'une instabilité passagère notamment quand les transmissions sont momentanément coupées. Ces travaux ont été simulés mais pas encore expérimentés. Ils sont parus très récemment (en avril 2000) et proposent un compensateur de retard sous la forme d'une modification du signal reçu $\tilde{u}_s(t)$ afin qu'il ressemble au mieux au signal émis retardé par un retard constant.

1.3.2 Commandes à retour virtuel

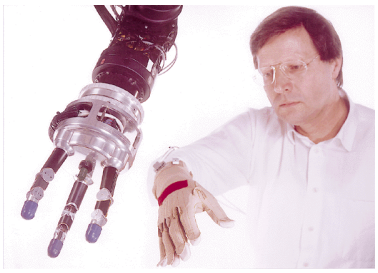
Dans le cadre de téléopérations à longue distance, les délais de transmissions varient de manière trop importante pour être dispensés de toute compensation. Mais une éventuelle compensation risque de pénaliser une commande de type bilatérale en maintenant un délai global constant mais nettement plus élevé que la moyenne des retards dus simplement au réseau. Une partie des roboticiens s'est donc orientée vers une solution tendant à améliorer l'ergonomie du poste de travail à l'aide de divers artifices. Dans tous les cas de manipulation d'objets à distance, il est primordial de garder un retour en effort afin que l'opérateur puisse opérer le plus finement possible.

Les commandes à retour virtuel font appel à des « affichages prédictifs » et des interfaces homme–machine évoluées pour fournir un retour d'information exempt de retard à l'opérateur. Ce dernier peut ainsi continuer le cours de manipulations ne faisant pas appel à une précision particulière sans avoir à attendre le retour réel des informations.

Les progrès en informatique, surtout du point de vue de la puissance des processeurs, ont permis de mettre en œuvre des traitements d'images vidéo en temps réel. Ainsi le premier roboticien a avoir proposé de superposer une image virtuelle en deux dimensions du télémanipulateur sur un retour vidéo fut Bejczy dans [BEJ 90]. L'opérateur manipule le robot virtuel en temps réel ; l'affichage prédictif de celui-ci crée un retour virtuel d'informations. Des améliorations ont notamment été apportées par [RAS 96] en passant en trois dimensions (cf. figure 1.5, page 11). Cette évolution a apporté dans un premier temps une meilleure compréhension de la réalité et donc de meilleures réactions de la part de l'opérateur. Les photographies de la figure 1.17 illustrent l'équipement de télémanipulation choisi lors d'un travail faisant suite à l'expérience Rotex [HIR 94] pour l'ESA[6].

Notons que les systèmes utilisant ce principe de superposition en deux ou trois

6. Agence Spatiale Européenne



(a) Main à 7 axes

(b) Visualisation 3D

(c) Kit Réalité virtuelle

Figure 1.17 – *Exemples d'interfaces utilisateurs améliorées*

dimensions nécessitent un étalonnage de la caméra. Ils sont inefficaces dans le cas de mouvements dans la direction de l'axe de l'objectif et ne sont pas adaptés aux mouvements fins car l'opérateur n'a qu'une vision globale de la scène. Il faudrait pouvoir déplacer la caméra (en la plaçant au bout d'un bras manipulateur ?) ou tout au moins pouvoir l'orienter et agrandir l'image de façon optique depuis le pupitre de commande.

L'ennemi redouté des systèmes de visualisation est, plus que les retards de transmission, la limitation de la bande passante. Plutôt que d'envoyer une image entière gourmande en bande passante, [BAL 98] propose justement d'utiliser un filtre de Kalman pour prévoir quelle partie de l'image va être modifiée et sera émise.

Il faut noter que l'état du robot virtuel doit être régulièrement resynchronisé avec celui de l'esclave pour limiter les dérives dues à la boucle ouverte ainsi créée. Cependant si le robot vient à entrer en contact avec l'environnement de manière inattendue, par exemple parce que l'environnement a évolué, l'opérateur aura l'information trop tard. Les limitations de ce type de commande ont conduit les roboticiens à privilégier une méthode de contrôle partagé.

1.3.3 Contrôle partagé

Rentrent dans cette catégorie des commandes telles que le contrôle de compliance partagée [KIM 92] ou encore le contrôle en impédance [BAC 92]. Il s'agit en fait de commandes bilatérales améliorées ; ainsi nous aurions également pu classer la commande bilatérale avec notion d'outil virtuel dans cette catégorie.

Il s'agit désormais d'asservir le robot en force et/ou en position localement. L'opérateur y gagne en lisibilité car le retour virtuel lui transmet immédiatement une estimation, plus exactement, une prédiction de l'action du robot distant.

Dans l'arsenal des solutions proposées pour améliorer l'Interface Homme–Machine, Mitsuishi et son équipe ont proposé d'implémenter un retour d'effort un peu particulier lorsqu'un retour d'effort via une manette n'est plus assez ergonomique pour cause de retards trop importants. Le retour a lieu sous forme sonore ; ainsi ce retour d'effort n'agit pas directement sur le mouvement de la main de l'opérateur.

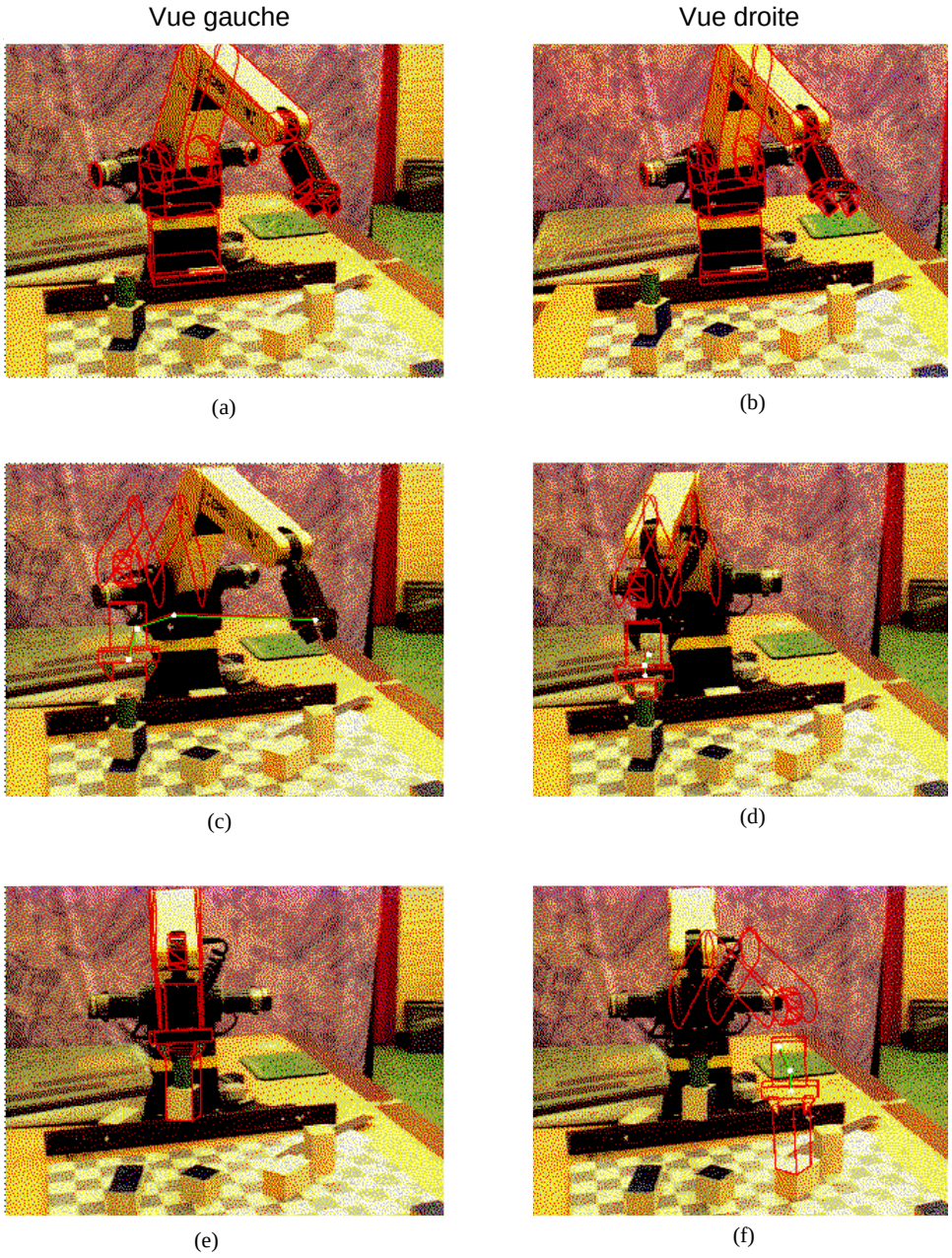


Figure 1.18 – *Illustration d'une manœuvre de déplacement [RAS 96]*

Un autre exemple publié dans [TAR 96] est donné par TARN et son équipe qui proposent une solution très avancée : ils utilisent un observateur prédicteur dédié aux systèmes à retards multiples [WAT 81]. Ils encapsulent ce prédicteur dans un contrôleur. Le prédicteur anticipe l'affichage de l'état du système téléopéré (avant même que les ordres n'arrivent à celui-ci) :

- les délais sont variables autour d'une moyenne de l'ordre de 1 s ;
- la période de transmission est de 200 ms ;
- le protocole de communication est *UDP*[7] via une connexion *PPP*[8].

D'autre part, TSUMAKI et UCHIYAMA préfèrent utiliser un modèle à tolérance d'erreurs géométriques (qui résultent en erreurs de position) et dynamiques (qui résultent en erreurs d'efforts et de vitesses) [TSU 97]. Pour ce faire, ils imposent une relation d'impédance entre l'esclave virtuel et l'esclave réel. Le modèle interne au prédicteur permet de créer des affichages prédictifs et des efforts virtuels afin de donner les moyens à l'opérateur de travailler dans de bonnes conditions.

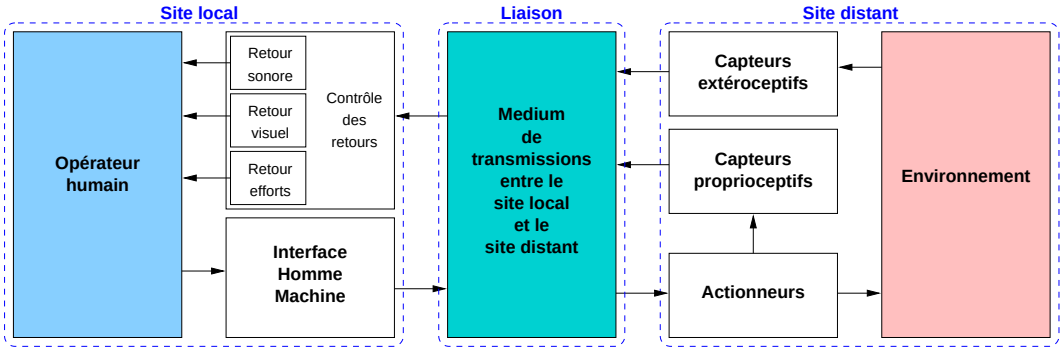


Figure 1.19 – *Schéma général d'une boucle de téléprogrammation*

L'inconvénient majeur des méthodes entrant dans cette catégorie est que les retards limitent la complexité et la fidélité des tâches à accomplir.

1.3.4 Contrôle superviseur ou téléprogrammation

Quand les retards sont tels (du point de vue de leur grandeur et de leurs variations) que les méthodes de téléopération précédentes sont inopérantes, une solution consiste à rompre la boucle globale présente dans les asservissements de type maître-esclave en deux boucles indépendantes telles que représentées en figure 1.20

Cette rupture engendre un degré d'abstraction supérieur des commandes envoyées par l'opérateur vers le robot téléopéré. Le manipulateur est asservi localement en force

7. cf. annexe C
8. Protocole permettant de transmettre des paquets *IP* sur une liaison téléphonique classique

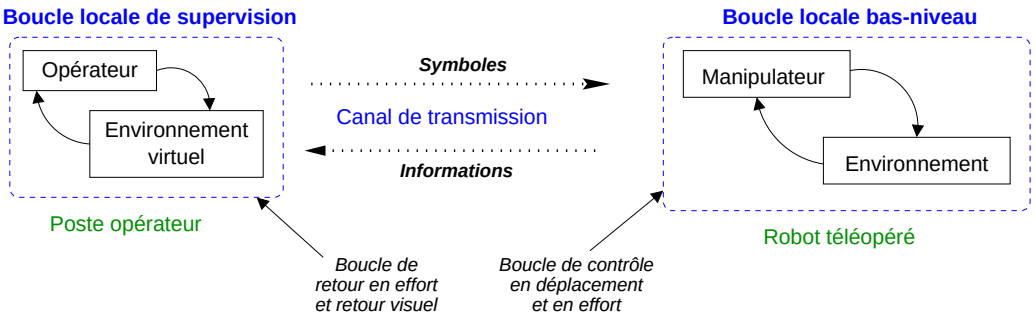


Figure 1.20 – *Concept de la téléprogrammation*

et/ou en position et doit être capable de reproduire des séquences de mouvements connus.

Il existe plusieurs niveaux d'abstraction en fonctions des auteurs. Cela va d'ordres tels que « *atteindre tel point avec telle orientation* », à des ordres plus subjectifs tels que « *attraper la poignée* », « *saisir l'objet* », ... Un tel degré de complexité, s'il facilite grandement la tâche pour l'opérateur, engendre le besoin de pré-programmer des tâches et d'utiliser de nombreux capteurs afin d'adapter ces programmes à l'environnement. Cela dit, chaque tâche, aussi complexe soit–elle, peut toujours être décomposée en mouvements élémentaires classiques en robotique.

Une autre difficulté se situant au niveau de la boucle de supervision consiste à tenir compte des incertitudes géométriques, cinématiques et dynamiques et à utiliser des méthodes de recouvrement en cas d'erreur grave (mouvement sorti des tolérances).

Parmi les avantages qu'apporte ce type de commande, la réduction des besoins en bande passante est intéressante. En effet, il n'est plus indispensable d'envoyer des consignes de façon continue. L'information symbolique peut se réduire à quelques échanges avant et après chaque manœuvre téléprogrammée. Il est toutefois préférable de garder un maximum de retours pour surveiller le déroulement des opérations.

Le terme de « téléprogrammation » décrit le concept de génération automatique d'un programme symbolique pour transmission puis interprétation et exécution par le télémanipulateur. Il a été proposé par Funda et Paul respectivement dans [FOU 91] et [PAU 93] comme nouvelle approche de la téléopération à grands retards éventuellement variables. Certains auteurs comme Pook dans [POO 95] et Ferrell dans [FER 67] préfèrent parler respectivement de « télé-assistance » et de « contrôle superviseur ». Cette approche fait appel à un retour d'informations à l'opérateur (« *feedback* ») assisté par ordinateur au niveau du système local et à une autonomie limitée du système distant. Elle a été validée expérimentalement mais il restait de nombreux points à élucider concernant le diagnostic d'erreurs de manipulation, les conditions de recouvrement, l'amélioration des retours d'information vers l'opérateur et une approche systématique de codage des tâches complexes en interaction avec l'environnement.

En 1994, Stein propose des améliorations à cette architecture de téléprogramma-

tion dans [STE94]. Il intègre notamment des techniques de diagnostic et de recouvrement d'erreur. Ses études portent sur des systèmes à retard variant de 5 à 10 s ayant une bande passante limitée sans retour vidéo et dont le manipulateur et son environnement sont en contact.

Le retour virtuel en 3 dimensions peut être complété par des syntaxeurs eux-aussi en 3 dimensions permettant de suivre les mouvements du corps ou tout au moins d'une partie de celui-ci (tête, bras, poignets et doigts). Lorsque l'opérateur est muni d'un casque de vision stéréographique, l'ensemble crée ainsi un système de *réalité virtuelle*[9]. Il est alors possible d'utiliser des gestes pour exécuter des tâches pré-programmées (« *aller à un point* », « *attraper une poignée* », ...). Ce type d'interface permet d'obtenir de bons résultats malgré des délais de communication de l'ordre de 4 s dans le cas de [POO 95].

Cette approche nécessite un équipement coûteux mais apporte à l'opérateur un confort propre à améliorer la qualité et la rapidité de son travail. Afin de faciliter l'utilisation générique de ce type d'appareillage, certains laboratoires ont étudié la façon de les intégrer dans les boucles de téléopération et ce, quel que soit le type de bras manipulateur utilisé [KH2 97]. Ils ont notamment été conduits à une téléopération à longue distance simultanée de plusieurs robots différents avec la même interface opérateur (cf. [KH3 97] et [KH1 97]).

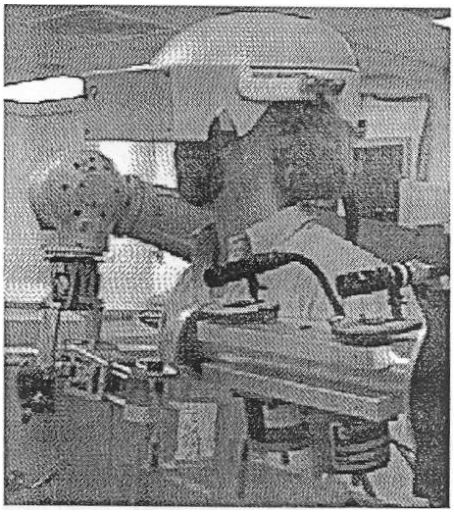


Figure 1.21 – *Equipement d'immersion en réalité virtuelle utilisé pour [POO 95]*

L'opérateur a également la possibilité de pré–simuler à l'écran une tâche à effectuer. Il peut ainsi, assez aisément, peaufiner la manœuvre à exécuter ultérieurement dans le cadre d'une téléprogrammation et vérifier que le robot n'entrera pas en collision avec son environnement [RAS 96]. La possession du modèle cinématique ou, dans le meilleur des cas, dynamique est alors nécessaire. Il est également possible d'utiliser un modèle dynamique adaptatif réagissant a fortiori aux derniers relevés de mouvements

9. cf. glossaire page 165

réceptionnés par la *base*. Le second intérêt de cette superposition est de pouvoir pré-simuler une manœuvre et ainsi de pouvoir détecter, a priori, des risques de collision.

Le monde de la téléopération sous-marine (cf. figure 1.22) a toujours été porté vers ce type de commande probablement à cause d'une bande passante assez réduite et des difficultés de manœuvres sous l'eau [SAY 96]. La robotique spatiale privilégie aussi la téléprogrammation lorsqu'il s'agit de téléopérer des rovers sur des planètes lointaines [BAC 92].

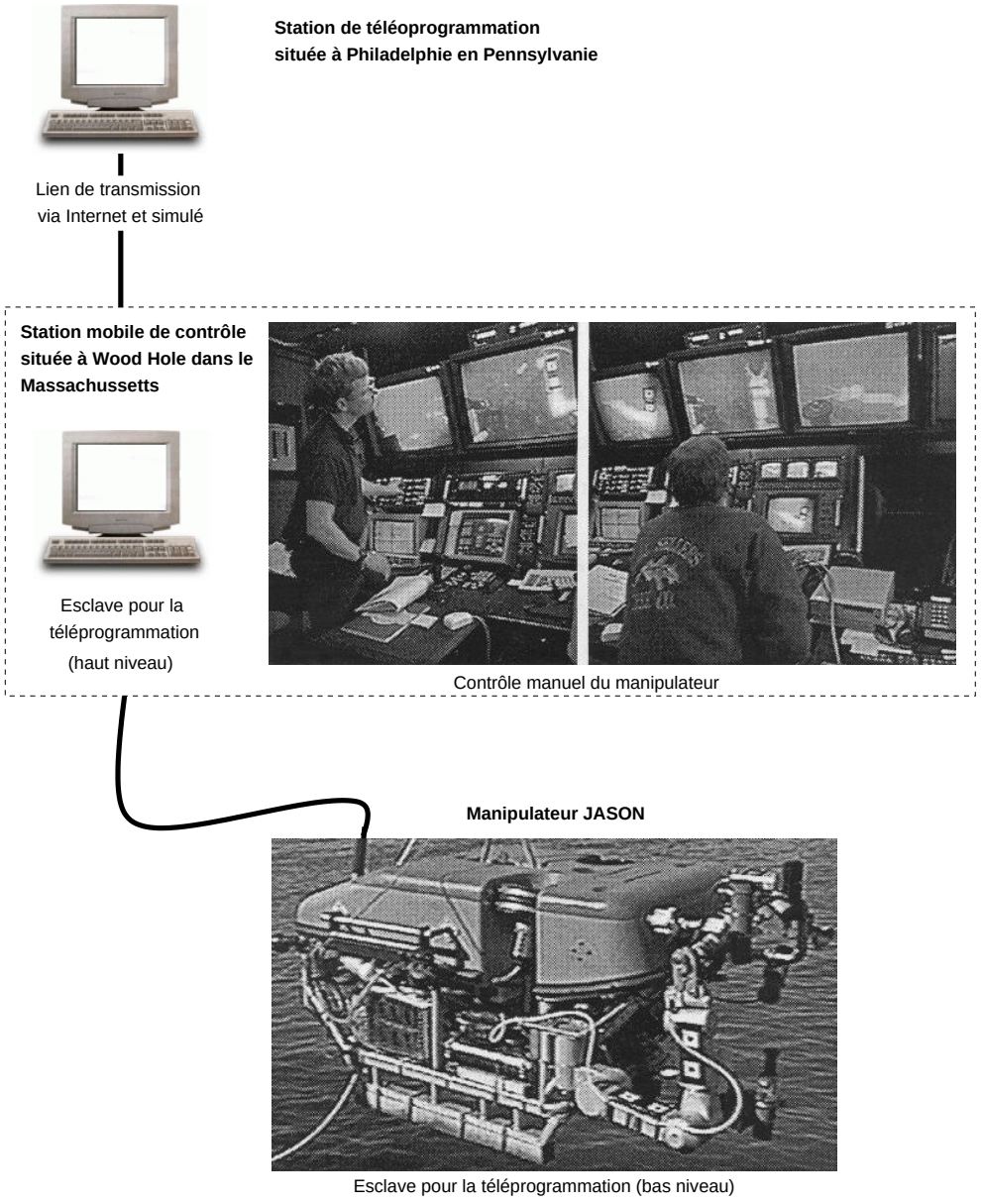


Figure 1.22 – *Boucle de téléopération sous-marine [SAY 96]*

1.4 Téléopération via l'*Internet*

Depuis une dizaine d'années, l'*Internet* est devenu une vitrine technologique pour la robotique. Ainsi, dans le cadre de recherches, certains laboratoires ont mis à la disposition des internautes des robots (en général des bras manipulateurs) qu'ils peuvent téléopérer. Certains de ces laboratoires le présentent comme un jeu où il s'agit, par exemple, de résoudre une énigme en découvrant des objets dans le sable (cf. [GOL 95] et [LUO 97]) ou plus simplement d'empiler des cubes en un minimum de mouvements [TAY 97]. Dans tous les cas, il s'agit de téléprogrammation.

Le but de ces recherches est d'étudier :

- les problèmes engendrés par la téléopération via l'*Internet* (cf. [SAY 96]),
- l'aptitude d'une personne à s'adapter à une interface initialement assez rudimentaire,
- la conception d'interfaces ergonomiques [GRA 00],
- les problèmes de sûreté de fonctionnement notamment pour des robots qui travaillent 24 heures sur 24, 365 jours par an ([OBO 97]),
- l'amélioration, via l'interface, des temps d'activité de l'opérateur,
- le partage entre laboratoires de la commande de sites d'expérimentation distants [STE 97] et [GOL 00],
- les architectures permettant d'interfacer un robot avec un serveur web ([TAY 95]).

Les applications sont, entre autres :

- des jeux : pour le commun des mortels, piloter un bras manipulateur dans le but d'empiler des cubes ou de résoudre une énigme [GOL 95] relève du jeu d'adresse et de réflexion,
- la télémanufacture : par exemple, Cybercut[10] est une entreprise qui réalise des découpes sur papier, bois, métaux directement depuis une interface *JAVA*,
- l'entraînement à distance pour un centre de formation, par exemple, qui ne peut se permettre de posséder autant de systèmes que d'élèves (chacun y accède à tour de rôle depuis son poste de travail).
- la commande d'expérimentations (sous-marines par exemple) à distance depuis n'importe quel ordinateur relié à l'*Internet* pour des scientifiques de laboratoires disséminés dans le monde [PAU 96].

La structure du *Web* n'est pas adaptée à une interaction entre le serveur et le client aussi poussée. En effet, il s'agit, à l'origine, d'un système de documentation à usage général initié par le CERN[11] en 1992. Le client est censé venir lire des pages uniquement en cliquant sur des liens hypertextes (« *Point and Click* ») reliant les pages entre elles.

Les actions possibles de l'utilisateur sont :

- cliquer sur un lien à l'aide d'un pointeur (souris, trackball, ...),

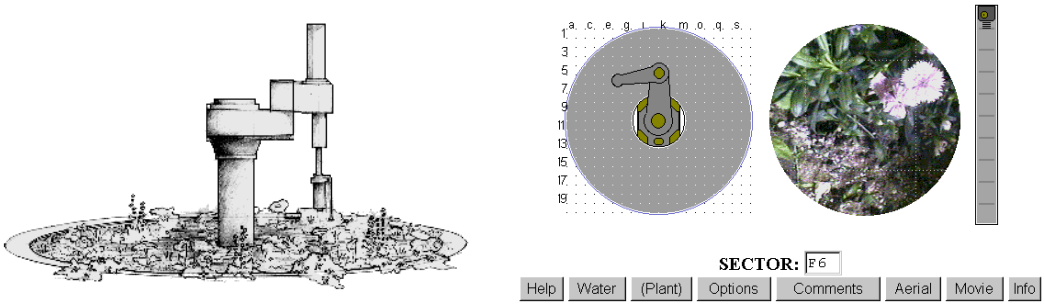
10. http://cybercut.berkeley.edu
11. Organisation Européenne pour la Recherche Nucléaire, cf http://www.cern.ch

- cliquer sur une partie d'une image,
- presser des boutons (poussoir, radio, à sélection),
- effectuer un choix entre plusieurs propositions (combo),
- entrer des renseignements dans des cases d'édition.

Lorsqu'il s'agit d'interagir avec une base de données, ce type d'actions est suffisant. Toutes ces actions envoient des renseignements à des programmes logés sur les serveurs (scripts *CGI* [12]). Ceux-ci envoient à leur tour des données résultant de la requête de l'utilisateur. L'exemple le plus parlant est un moteur de recherche tel que Yahoo [13] ou Altavista [14], pour ne citer qu'eux.

Par contre, lorsqu'il s'agit de téléopérer un robot, il est un peu limité. En outre, le rafraîchissement automatique limité à une période supérieure ou égale à 1 s grève le plus l'ergonomie de l'interface. Le programmeur était obligé d'avoir recours à des astuces de programmation assez lourdes dès lors qu'il cherchât un peu plus de dynamisme et d'interactivité.

Un exemple de téléopération via *Internet* illustrant l'utilisation de scripts *CGI* est proposé depuis août 1995 à l'Université de Californie du Sud. Il s'agit d'un « *téléjardin* » [15] (représenté figure 1.23) qu'il est possible d'arroser, auquel l'internaute peut ajouter des plantes et dont il peut suivre la croissance. L'utilisateur visualise l'ensemble à l'aide d'un schéma du téléopérateur vu de dessus, d'une photo prise par une caméra fixée au poignet du bras manipulateur et de différents boutons de contrôle (zoom, arrosage, plantation, ...).



(a) Vue d'ensemble (b) Interface WWW

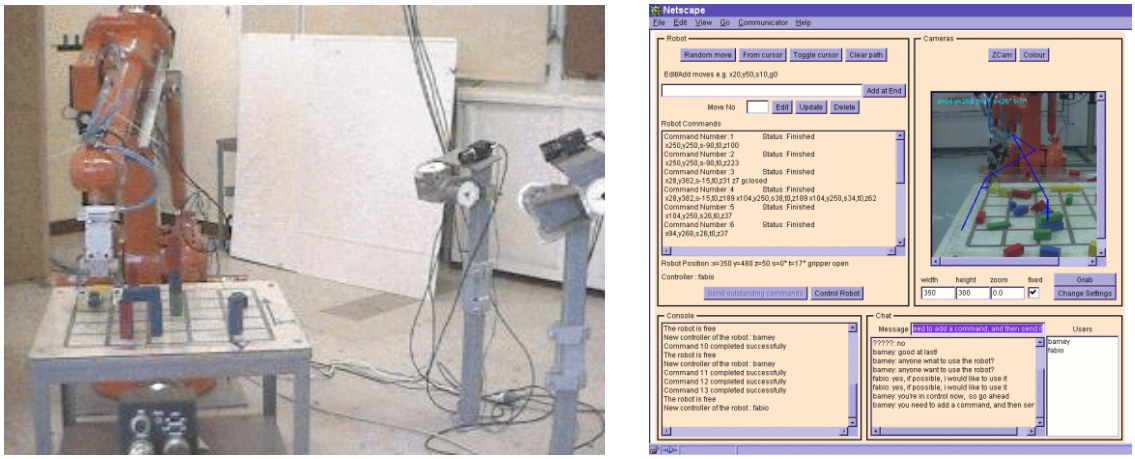
Figure 1.23 – *Le téléjardin de l'USC*

Aujourd'hui, la solution la plus proche de l'idéal passe par l'utilisation d' « *applets* » *JAVA* : petits programmes téléchargés puis exécutés sur la machine de l'utilisateur et ayant des permissions réduites afin d'assurer la sécurité des données stockées dans

12. *Common Gateway Interface*; cf. http://hoohoo.ncsa.uiuc.edu/cgi/
13. http://www.yahoo.fr
14. http://www.altavista.fr
15. http://www.usc.edu/dept/garden/

l'ordinateur hôte. Il est maintenant assez aisé de programmer une interface opérateur en *JAVA* capable de dialoguer avec un serveur lié au télémanipulateur. On retrouve ce type d'interface chez [DEP 97].

Un second exemple fait appel à la technologie *JAVA* pour proposer une interface plus dynamique (représentée sur la figure 1.24). Il s'agit du « *Télérobot* »[16] de l' « University of West Australia ». Ici, l'utilisateur pilote un bras manipulateur afin de déplacer des objets sur une table. Il dispose de trois vues différentes et commutables de la scène [TAY 97].



(a) Vue d'ensemble (b) Interface *JAVA*

Figure 1.24 – *Le télérobot de l'UWA*

Ce type d'interface est également utilisé par la NASA pour préparer et visualiser les résultats des missions de ses rovers envoyés sur d'autres planètes : c'est le projet *WITS*[17] [BAC 00]. Il s'agit d'une application *JAVA* cliente représentée en figure 1.25, que plusieurs scientifiques participant aux expérimentations et disséminés sur le territoire américain, peuvent connecter au serveur du JPL. Ils utilisent une interface partagée via l'*Internet*. Pour des raisons évidentes de sécurité, les données sont cryptées (principe de la clef publique et de la clef privée) et sont envoyées via une couche de type *SSL : Secured Sockets Layer*.

Il y a quelques années, est apparue une nouvelle technologie permettant de distribuer une représentation en 3 dimensions d' objets (molécule, robot, ...) sur le *Web* : le langage *VRML*[18] qui permet de représenter des scènes simples en trois dimensions. N'importe quel navigateur capable de visualiser ce format permet à l'utilisateur de naviguer en temps réel dans la scène. Ce langage a également gagné en interactivité en rendant possible le couplage d'une applet *JAVA* avec une scène. Jörg VOGEL de

16. http://telerobot.mech.uwa.edu.au/
17. Web Interface for Telescience: http://robotics.jpl.nasa.gov/tasks/wits/homepage.html
18. *Virtual Reality Modeling Language* cf. http://www.vrml.org

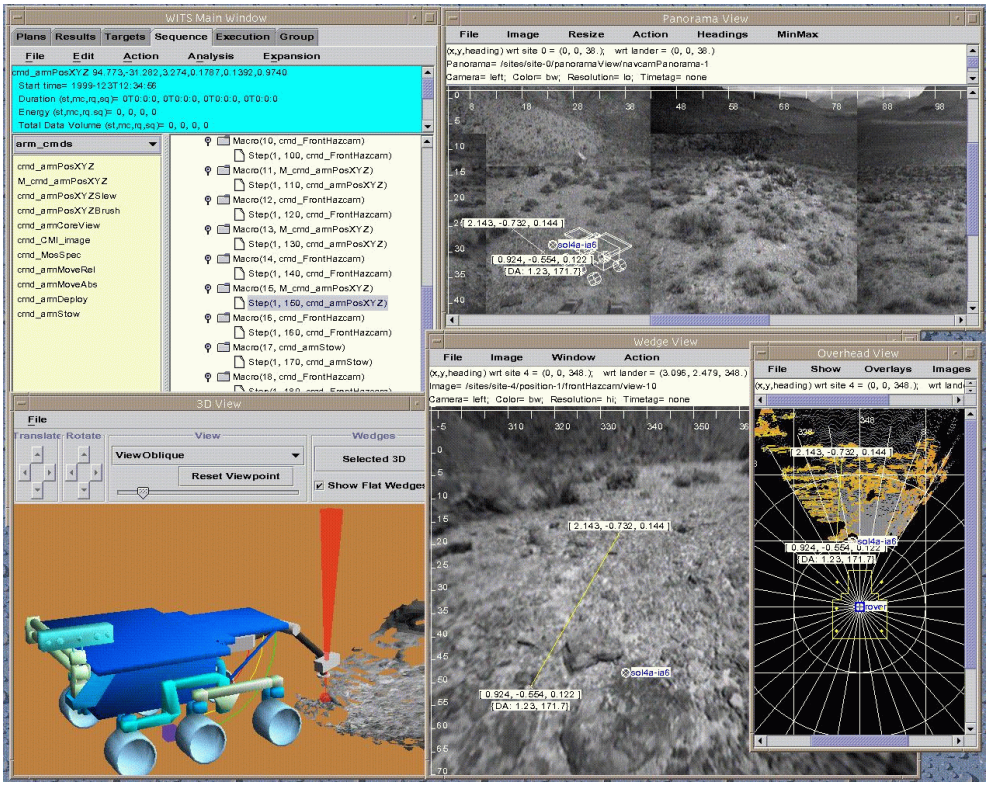


Figure 1.25 – *Interface* JAVA *du projet* WITS *[BAC 00]*

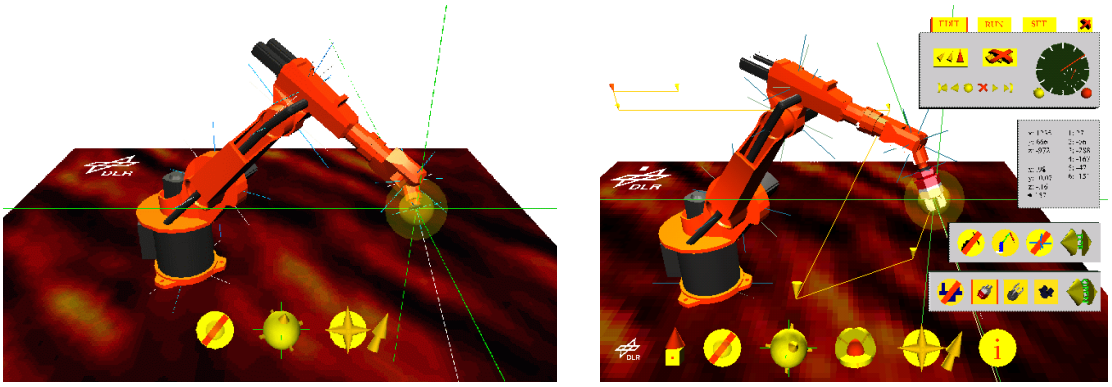
l' « *Institute of Robotics and System Dynamics* » en Allemagne a développé des interfaces de téléopération[19] fondées sur ce principe [VOG 99]. La figure 1.26 présente l'une d'elles.

1.5 Conclusion

Les techniques de télécommunications évoluent sans cesse. Partie de transmissions mécaniques, la téléopération a évolué au fil du temps vers des transmissions électroniques, dans un premier temps analogiques puis numériques. Avec cette évolution, les distances ont pu augmenter de quelques dizaines de centimètres initialement à des milliers de kilomètres dans le cadre d'expérimentations terrestres et nettement plus lorsqu'il s'agit de téléopation spatiale. Cet éloignement a pour principal avantage de multiplier le nombre d'applications potentielles à cette technique largement reconnue dans le domaine de la robotique. Il a pour principal inconvénient d'apporter des limitations en terme de retards de transmission et de limitation de bande passante, fonctions des distances et des moyens mis en œuvre.

Cet état de l'art fait ressortir des architectures de téléopération dédiées à des téléopérations dont les retards de transmission (constants) supportés atteignent quelques

19. http://www.robotic.dlr.de/Joerg.Vogel/Vrml/index.html



(a) Vue d'ensemble (b) Outils d'aide

Figure 1.26 – *Exemple d'interface* VRML

secondes, hormis celles classées sous la bannière de la téléprogrammation. Quasiment toutes cherchent à implanter une commande à retour d'effort en conservant le schéma classique du maître et de l'esclave.

Le problème de la variabilité des retards est plus récent; en effet, il est beaucoup plus présent dans le cas où il est fait appel à des moyens de transmissions numériques partagés tels que le réseau *Internet*.

Ce dernier a ouvert une branche de recherche en matière d'interface web homme-machine pour la téléopération via *Internet*. Ici aussi les techniques se sont affinées avec les progrès réalisés au niveau grand public avec l'apparition de technologies telles que *JAVA* et *VRML*.

Chapitre 2

Modélisation

2.1 Introduction

Le but de l'étude qui suit est de proposer un modèle mathématique d'un système générique de téléopération complet comprenant :

- le système à téléopérer,
- le médium avec ses défauts : délais de transmission variables sans perte de données et
- le poste de commande.

Nous détaillerons tout d'abord le schéma global de téléopération sur lequel nous avons fondé tous nos travaux.

Nous présenterons ensuite la plate-forme d'expérimentations, constituée d'un manipulateur mobile terrestre (décrit dans § 2.3, page 39), plate-forme que nous avons utilisée dans un premier temps pour diverses identifications (cf. § 2.4, page 51) venant compléter le modèle mathématique.

Pour des raisons de commodité, nous avons adopté comme médium une liaison de type *IP*. Cela a pour intérêt de pouvoir utiliser avec le même protocole notre réseau local pour de faibles distances et l'*Internet* pour de longues distances. Ainsi, nous décrirons dans § 2.5, page 60, la modélisation du médium de transmissions.

2.2 Schéma initial de téléopération

Par la suite, nous utiliserons les termes « *base* » et « *système distant* » pour désigner respectivement l'ensemble téléopérateur et le système à téléopérer. Cette architecture ressemble ainsi à un système bouclé (cf. figure 2.1).

Pour simplifier le raisonnement, nous ne considérerons qu'un système monodimensionnel — celui-ci est pensé de telle sorte qu'il puisse être étendu à plusieurs dimensions aisément.

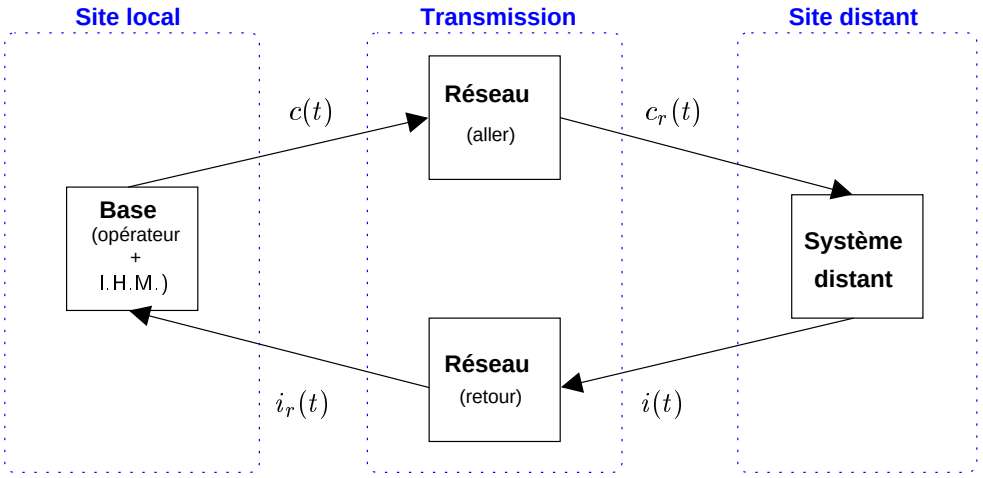


Figure 2.1 – *Schéma de téléopération basique*

Les quatre principaux signaux intervenant dans les transmissions entre *base*, bloc de transmission et *système distant* sont :

- $c(t)$ signal de consignes,
- $c_r(t)$ signal de consignes $c(t)$ retardé aléatoirement,
- $i(t)$ signal de retour d'informations,
- $i_r(t)$ signal de retour d'informations $i(t)$ retardé dans le temps.

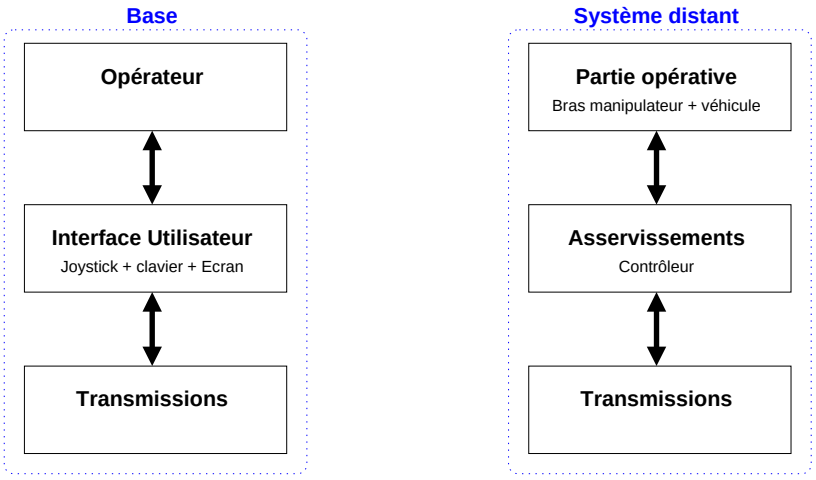
2.2.1 La *base*

La *base* (représentée en figure 2.2(a)) communique avec le *système distant* grâce au même lien full-duplex[1]. L'Interface Homme-Machine (I.H.M.) envoie périodiquement (à la même période T_t) les consignes de l'opérateur au *système distant* et affiche des données concernant l'état de la liaison et du *système distant* au fur et à mesure de la téléopération.

2.2.2 Le *système distant*

Le *système distant* (représenté en figure 2.2(b)) inclut un organe de transmission full-duplex (non représenté en figure 2.2) avec la *base* ainsi que l'asservissement local de ses variables d'état (la vitesse, la direction ou l'un des 6 axes du manipulateur) au niveau du bloc contrôleur. Il émet périodiquement (à la période T_t) l'état de ses capteurs à la *base*.

1. lien bidirectionnel avec émission et réception simultanées



(a) Base (b) Système distant

Figure 2.2 – *Détails des blocs* base *et* système distant

2.2.3 Transmissions

Afin d'obtenir un système global modélisable en un système échantillonné, les transmissions de données entre la *base* et le *système distant* se font sous forme de paquets de données (consignes et informations en retour) émises périodiquement à la période T_t. Celle-ci dépend de la quantité de données à émettre par rapport à la bande passante globale du médium. Dans toute notre étude, cette période est constante. Il n'est pas exclu d'améliorer ultérieurement les performances de notre architecture en faisant appel à une période de communication T_t variable par paliers et ainsi d'optimiser la bande passante du médium utilisé. Nous avons donc arbitrairement fixé cette période T_t à 200 ms. Il s'agit d'une limite technologique imposée par la vitesse d'exécution de l'application de téléopération utilisée au cours de nos toutes premières expérimentations.

Les transmissions entre la *base* et le *système distant* ne sont jamais parfaites. Elles font intervenir des délais de transmission variables dus, d'une part, à la distance à parcourir lorsque les deux protagonistes sont particulièrement éloignés et, d'autre part, aux organes de commutation qui peuvent être nombreux s'il s'agit d'un réseau informatique fortement maillé tel que l'*Internet*.

2.3 Site expérimental

Afin de construire notre modèle de simulation générique, nous avons effectué des expérimentations préalables en faisant appel à un manipulateur mobile terrestre à notre disposition. Une description plus détaillée de son architecture est disponible dans [LEL 98]. La figure 2.3 est une vue « d'artiste » de notre plate-forme de téléopération.

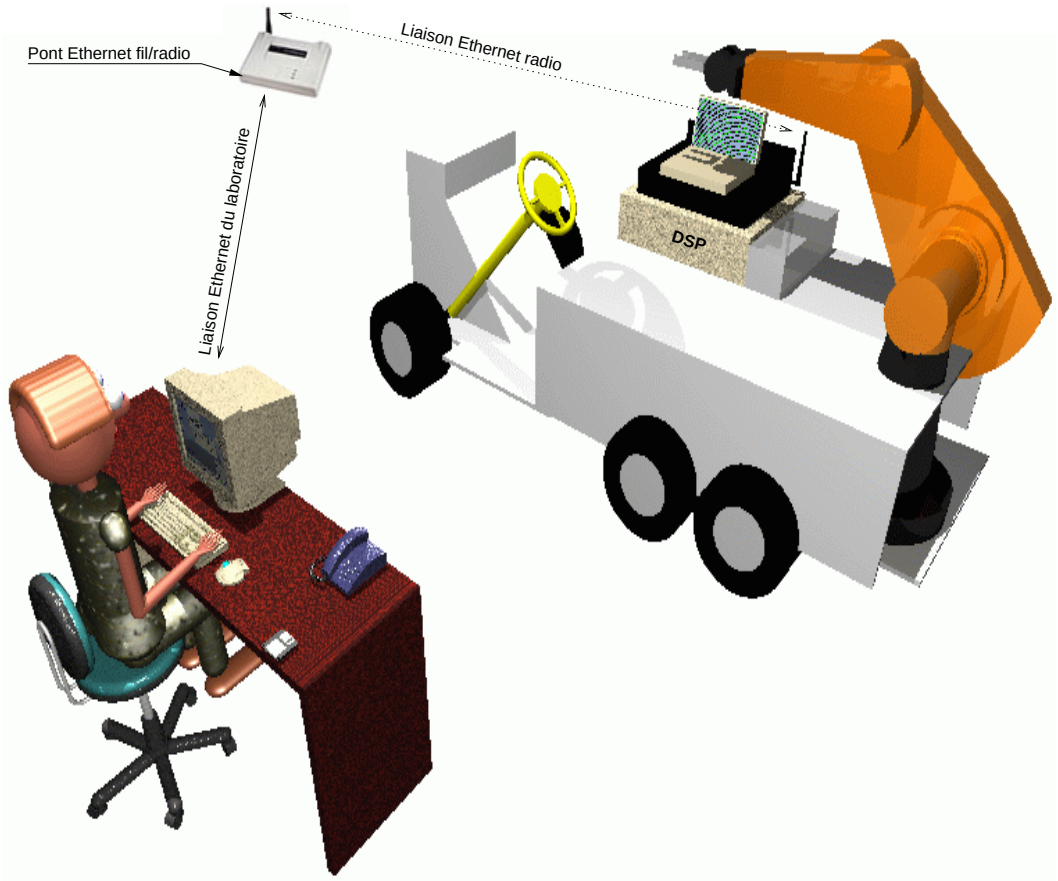


Figure 2.3 – *Plate-forme de téléopération*

2.3.1 La *base*

La *base* consiste en un ordinateur de type *PC* équipé d'un joystick. Pour effectuer une téléopération, un programme se charge de se connecter au *PC* superviseur, de lui transmettre les consignes de l'opérateur et de réceptionner en retour les informations nécessaires à la téléopération. L'opérateur se contente alors de manipuler son joystick pour manœuvrer soit le véhicule, soit le bras manipulateur, soit les deux dans le cadre de manœuvres particulières.

Pour tester le manipulateur mobile dans différentes situations, nous avons créé un script simulant une action coordonnée du bras manipulateur et du véhicule. Ce script est exécuté par la *base* et enregistre tous les retours d'information émis par le manipulateur mobile en les datant afin de pouvoir étudier ultérieurement le comportement du système global de téléopération.

2.3.2 Le *système distant*

Le *système distant* consiste en un véhicule terrestre de type 6×6 pourvu d'un bras manipulateur *PUMA* visible figure 2.4. Un *DSP*[2] asservit localement cet ensemble, en position pour les 6 axes du bras et la position du volant, et en vitesse pour le véhicule. Pour des raisons de sécurité, un filtre passe-bas échantillonné du second ordre limite les accélérations des axes du bras, du volant et du véhicule.

Un ordinateur de type *PC* connecté au *DSP* (*PC* Supervision) et doté d'une carte *Ethernet* sans fil reçoit les consignes par radio et les transmet au *DSP* chargé des divers asservissements. Il récupère les résultats de mesures générés par le *DSP* et les envoie à la *base*.

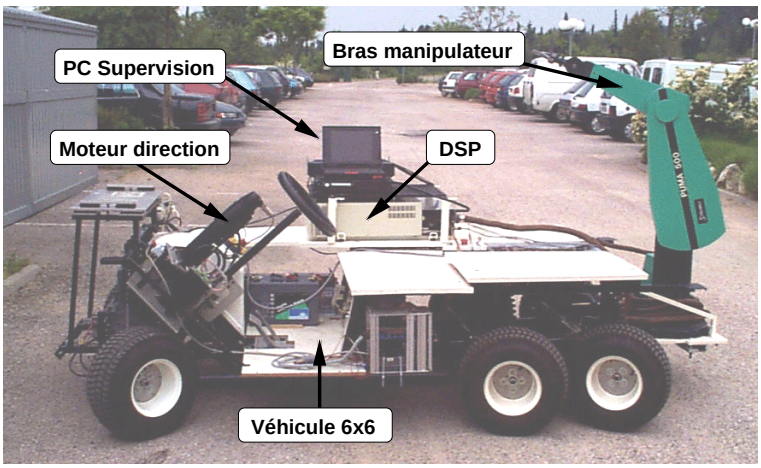


Figure 2.4 – *Manipulateur mobile*

2. *Digital Signal Processor* : microprocesseur spécialisé dans les calculs de nombres flottants

2.3.3 Architecture logicielle

Afin d'effectuer les identifications nécessaires à l'élaboration d'un modèle de boucle générique de téléopération, nous avons développé un environnement logiciel de téléopération.

Cet environnement est constitué d'un programme *serveur TCP*[3] situé sur le *PC* lié au système distant (ici notre manipulateur mobile) que nous avons baptisé « MANIMOB » et d'un programme *client TCP* « BASE » exécuté au niveau du poste opérateur.

Pour effectuer une téléopération basique, le client BASE (dont l'interface est en figure 2.6) établit une connexion avec le serveur MANIMOB. Ceux-ci s'échangent ensuite consignes et informations de retour.

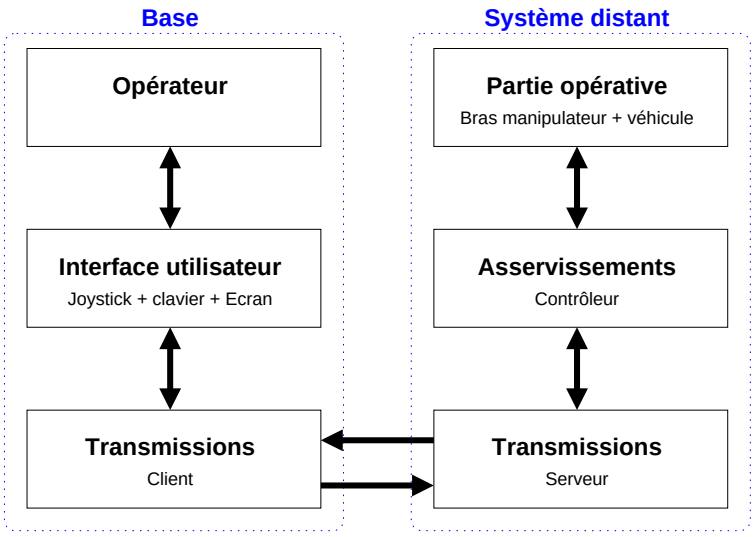


Figure 2.5 – *Architecture logicielle de téléopération*

Choix du protocole de transmission

Pour des raisons pratiques, nous avons choisi d'utiliser des protocoles de la famille *TCP/IP*. Il existe deux protocoles couramment utilisés pour connecter deux programmes reliés par un réseau informatique sur lequel est implanté l'ensemble des protocoles *TCP/IP* tel que l'*Internet* : *TCP* (*Transport Control Protocol*) et *UDP*[4] (*User Datagram Protocol*).

Nous avons préféré *TCP* à *UDP* pour les raisons suivantes :

1. *TCP* est orienté connexion : on peut s'arranger pour qu'un seul poste téléopère le *système distant* à tout instant et le protocole est capable de détecter les ruptures de connexions,

3. cf. annexe C, page 175
4. cf. annexe C, page 175

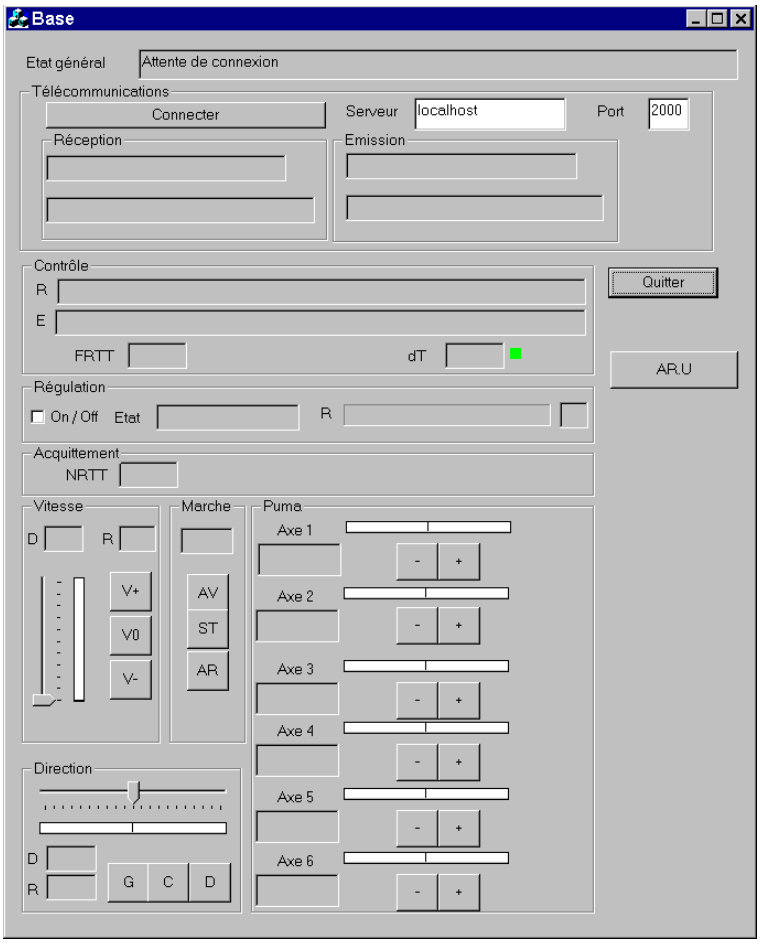


Figure 2.6 – *Interface logicielle pour BASE*

2. *TCP* est un protocole fiable : il s'assure que la livraison de données a lieu dans une séquence correcte (*UDP* ne garantit ni la livraison ni l'ordre d'arrivée des flots de donnée),

3. *TCP* (contrairement à *UDP*) contrôle dynamiquement le flux de données ; ainsi si le récepteur est trop lent pour récupérer ses données, l'émetteur ralentira automatiquement ses émissions.

Les inconvénients de *TCP* par rapport à *UDP* pour notre application sont :

1. le temps de transmission des données légèrement plus long, du fait des traitements de reséquençage en réception (remise des données dans l'ordre de départ),

2. en cas de perte d'une trame, les données suivantes ne seront accessibles que lorsque cette trame aura été réémise et correctement réceptionnée ; cela peut être gênant si les données suivantes contiennent des données urgentes de type arrêt d'urgence,

3. le contrôle de flux qui peut, en cas de baisse brutale de la bande passante, ralentir l'émission de données initialement périodique. Il est important de bien dimensionner la bande passante avant d'effectuer la téléopération et d'adapter le flot de données à émettre — autrement dit, la période d'émission des messages — au débit maximum possible pour éviter ce type de situation.

En fait, une analogie courante associe le protocole *TCP* à un service de conversation téléphonique et *UDP* à un service de livraison postale (dont les « facteurs » peuvent éventuellement perdre du courrier en route sans s'en rendre compte...).

A l'instar de chaque couche de protocole située entre la couche physique et la couche application (cf figure C.1, page 175), nous avons ajouté un contrôle supplémentaire des trames échangées afin d'augmenter la sécurité des transmissions. Il s'agit d'une vérification par somme de contrôle de la validité du contenu des données. En effet, nous ne pouvons nous permettre d'envoyer des consignes erronées à un robot en mouvement.

D'autre part, lorsque la transmission est coupée momentanément, nous nous sommes aperçu que les *sockets*[5] *TCP* ne nous le signalaient pas assez rapidement. Or nous ne pouvons laisser un robot en mouvement sans consigne supplémentaire : il risque d'effectuer des manoeuvres dangereuses pour lui et son environnement. C'est pourquoi, nous avons ajouté un système d'acquittement supplémentaire. Ce système permet d'une part de détecter une rupture de communication non détectée par *TCP* et, dans ce cas là, d'effectuer un arrêt d'urgence adéquat du système distant. Il permet, d'autre part, d'effectuer des mesures de « *temps de trajet aller–retour* » des données en temps réel.

5. cf. annexe C, page 175

Fonctionnement détaillé

Nous avons opté pour la construction la plus modulaire possible des applications MANIMOB et BASE. Ainsi, nous retrouvons dans les deux programmes les modules suivants (disposés selon la figure 2.7) :

- un *noyau* gérant l'interaction entre les parties suivantes,
- un bloc *transmissions* pour le dialogue entre les deux applications,
- une *interface utilisateur*,
- un *dialogue avec le DSP* (uniquement pour MANIMOB) : interface avec le programme d'asservissement exécuté sur le *DSP*.

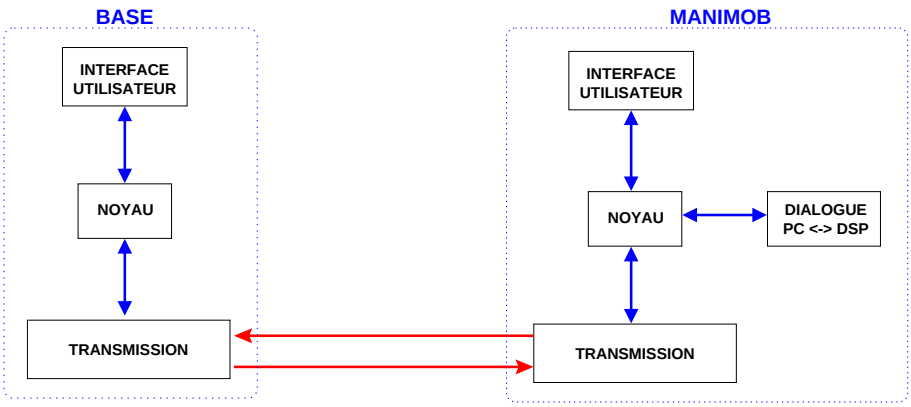


Figure 2.7 – *Conception modulaire des applications BASE et MANIMOB*

Par ailleurs, la partie transmission est elle-même constituée de deux principaux blocs : le récepteur et l'émetteur (cf. figure 2.8). Chacun d'eux correspond à une connexion *TCP* avec, respectivement, l'émetteur et le récepteur de l'application à contacter. Cela permet notamment de séparer les données à acheminer sur le réseau de leurs acquittements. Un bloc superviseur synchronise l'émetteur et le récepteur lors des phases de connexion et de déconnexion de l'application.

Chacun de ces blocs émetteur et récepteur est constitué d'un empilement de couches dédiées chacune à une tâche particulière dans la chaîne de transmission (cf. figure 2.9). La fonction de ces couches est, dans l'ordre ascendant :

- transmissions bas-niveau : gestion de la connexion de type client ou serveur, accès aux *sockets*, tampons d'émission et de réception,
- sécurisation des données : ajout (à l'émission) et vérification (à la réception) d'une somme de contrôle,
- protocole d'acquittement pour le renforcement de la sûreté des données et les calculs de temps de vol.

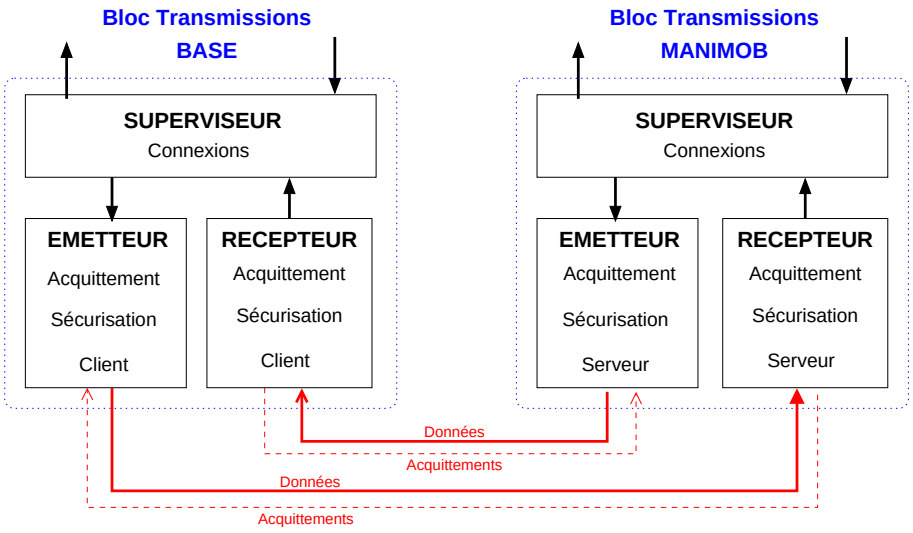


Figure 2.8 – *Séparation des fonctions d'émission et de réception*

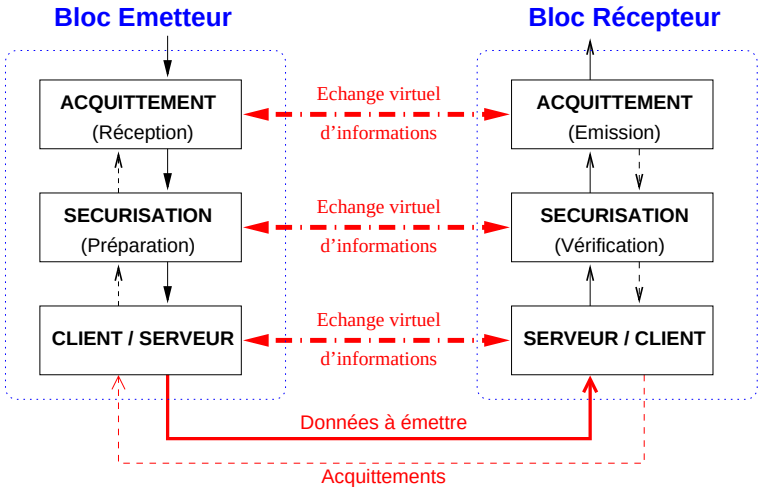


Figure 2.9 – *Conception stratifiée des blocs de transmissions*

La nature relativement symétrique des transmissions des données nous a permis de mettre en commun une grande partie du code source d'émission et de réception de BASE et MANIMOB. La partie transmission de données est entièrement découplée de la partie connexion. Ainsi, un récepteur ou un émetteur peuvent être aussi bien serveur que client. Tous les cas de figures sont représentés dans ces deux programmes.

Aspect temps réel

L'application MANIMOB est exécutée sur le *PC* superviseur (cf figure 2.4) dont le système d'exploitation est *Windows* 95 (imposé pour des questions de compatibilité avec l'équipement de gestion et d'interface du *DSP*). L'application BASE est également exécutée sur un *PC* mais sous *Windows NT4*. Ces deux systèmes d'exploitation ne sont pas des systèmes de type « *temps réel* ».

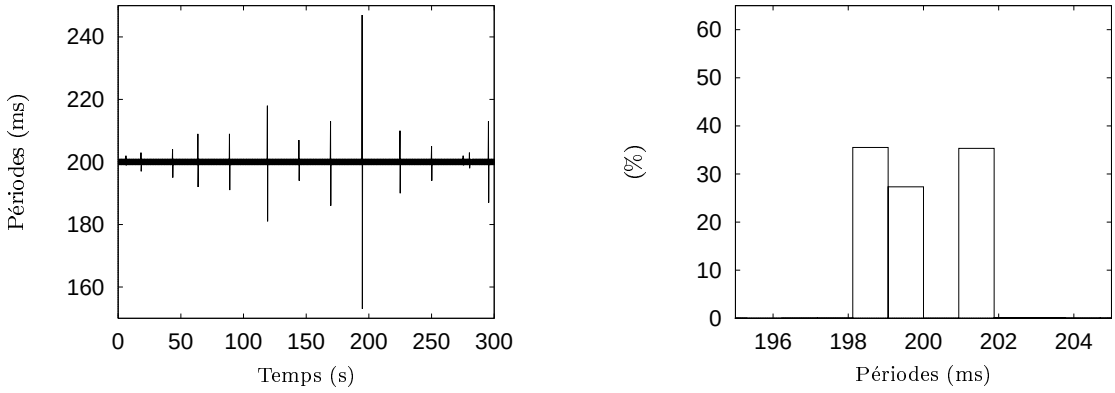
Les routines que nous avons développées s'appuient sur une fonctionnalité offerte par l'environnement de programmation Visual C^{++} nommée « *Timer* ». Il s'agit d'un événement que nous programmons pour qu'il appelle une routine particulière après un laps de temps donné. Ce laps de temps est supérieur ou égal à 1 ms. Il est également possible de programmer le timer de manière à ce que cet événement soit périodique. Cependant, si le système d'exploitation est occupé à une tâche de priorité supérieure à celle de notre processus, il faudra attendre que cette tâche se termine pour que l'événement se produise, d'où certaines imprécisions parfois critiques pour une application de robotique. Il n'est actuellement pas possible de spécifier un niveau de priorité pour nos événements dans cet environnement logiciel. Il faut prendre en compte ces limites avant de conclure sur la qualité des algorithmes qui s'appuient sur cette gestion temporelle.

Enfin, il est fortement conseillé d'éviter tout accès aux ressources du *PC* hôte simultanément à l'exécution de nos applications.

Pour illustrer la qualité de l'horloge obtenue pour les deux applications BASE et MANIMOB, nous avons effectué quelques mesures temporelles. Nous avons relevé les instants d'appel des routines périodiques afin de caractériser d'une part la périodicité de l'horloge ainsi générée et, d'autre part, de vérifier la présence d'une dérive temporelle éventuelle.

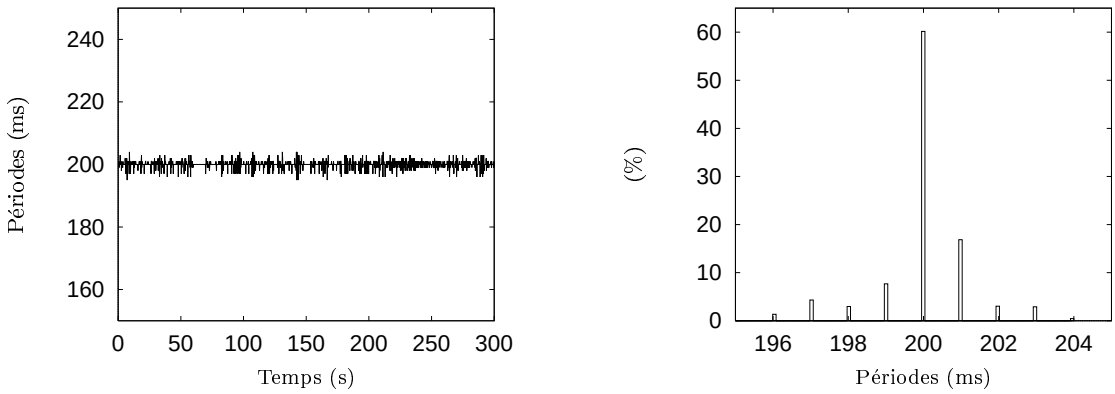
Les figures 2.10 et 2.11 représentent respectivement la qualité des périodes pour BASE et MANIMOB. Dans les deux cas, la moyenne est bien centrée sur $T_t = 200$ ms. Cependant, il apparaît que la périodicité est nettement plus fiable pour MANIMOB que pour BASE. Cela tient au fait que le *PC* sur lequel est exécuté MANIMOB (sous *Windows* 95) possède nettement moins de processus actifs en tâches de fond que le *PC* sur lequel est exécuté BASE (sous *Windows NT4*). Ainsi l'écart-type pour MANIMOB est de 1,5 ms contre 2,3 ms pour BASE. L'amplitude des variations des périodes autour de la moyenne ne dépasse fort heureusement pas les 200 ms : localement 10 ms pour MANIMOB et 95 ms pour BASE, ce qui n'est pas négligeable.

Dans le cas de BASE, nous pouvons aisément constater une symétrie autour de la période moyenne (elle existe également pour MANIMOB mais elle est nettement



(a) Evolution (b) Distribution

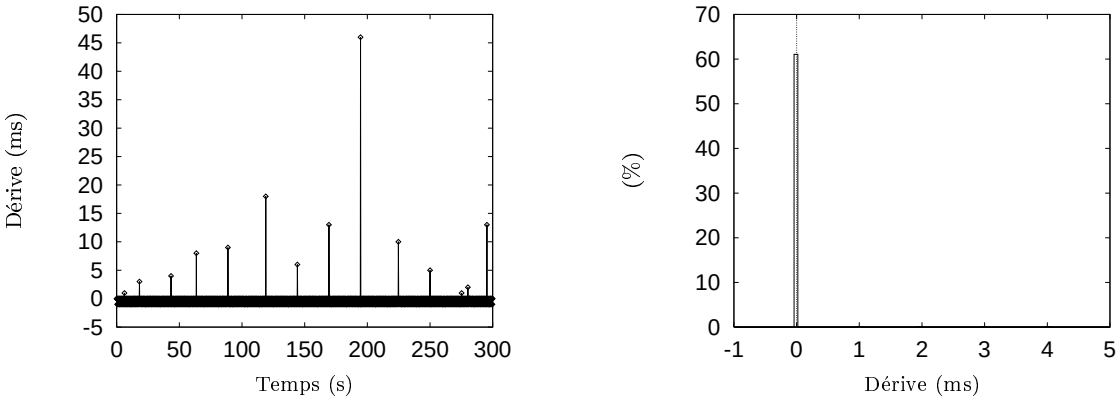
Figure 2.10 – *Qualité des périodes d'horloge pour BASE*



(a) Evolution (b) Distribution

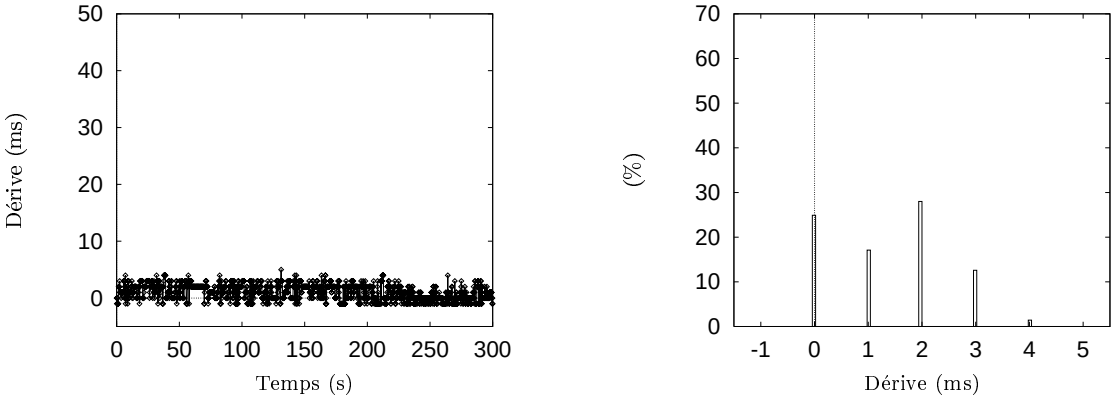
Figure 2.11 – *Qualité des périodes d'horloge pour MANIMOB*

moins visible ici). Elle est due au fait que les tops d'horloge sont générés de manière à éviter toute dérive : si un top arrive plus tard que prévu, le suivant arrivera à l'heure prévue, quitte à obtenir localement une période plus faible que celle désirée. Ainsi si une période dépasse les 200 ms de x ms, la période suivante sera amputée de x ms, d'où cette symétrie autour de la moyenne.



(a) Evolution (b) Distribution

Figure 2.12 – *Dérive de l'horloge de BASE*

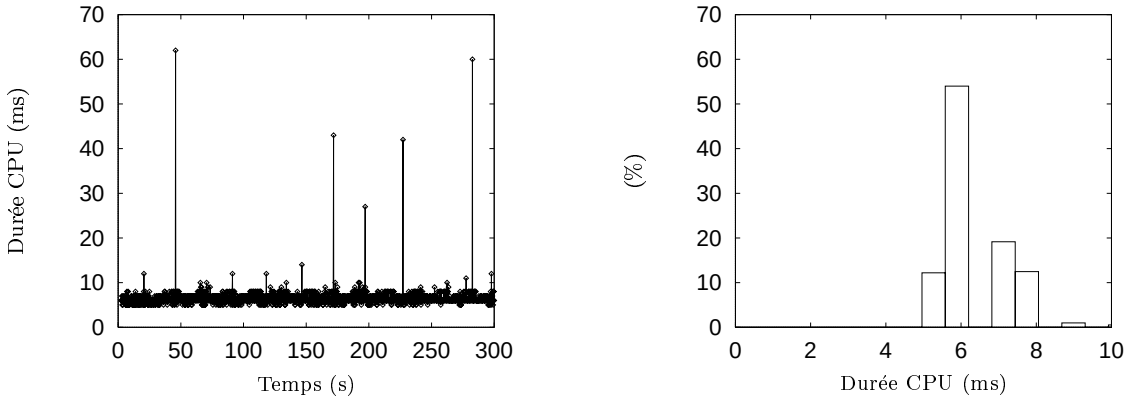


(a) Evolution (b) Distribution

Figure 2.13 – *Dérive de l'horloge de MANIMOB*

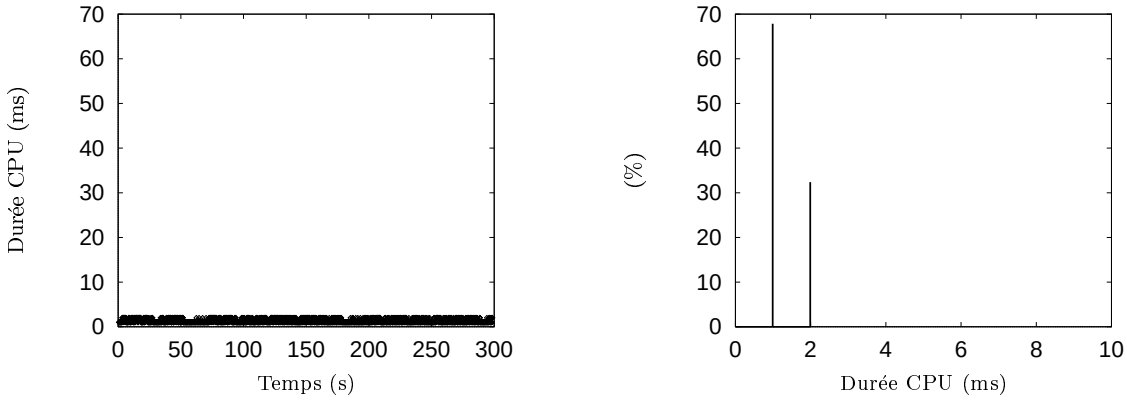
Par ailleurs, nous pouvons constater sur les figures 2.12 et 2.13 que notre signal d'horloge ne dérive pas par rapport à l'horloge du PC; nous retrouvons les mêmes dispersions qu'au niveau des mesures de période.

Nous avons également observé les temps de travail nécessaires aux divers calculs effectués à chaque période d'horloge (figures 2.14 et 2.15). Nous pouvons constater qu'actuellement les temps de calcul sont nettement inférieurs à la période d'horloge. D'après ces relevés, nous pourrions être tenté de diminuer la période de moitié. Or il



(a) Evolution (b) Distribution

Figure 2.14 – *Temps CPU pour* BASE



(a) Evolution (b) Distribution

Figure 2.15 – *Temps CPU pour* MANIMOB

faut également tenir compte de la bande passante du canal de transmission par rapport au volume de données échangées entre les deux applications. Sans compter que ce temps de calcul sera amené à augmenter dans un usage futur.

Comme nous le constaterons ultérieurement dans cette étude, il sera indispensable, à l'avenir, de porter ces applications sur une plate-forme temps réel. Le plus simple étant d'adopter dans ce cas, l'extension temps réel de *Windows NT4*, *RTX*, qui permet d'obtenir une résolution d'horloge de 100 μs.

2.4 Identifications

Ces expérimentations préalables ont eu lieu « à faible distance » pour des raisons pratiques. Elle nous ont permis, dans un premier temps, de nous consacrer aux problèmes dus à la téléopération, en général. L'identification du comportement du réseau à longue distance a été effectuée indépendamment.

2.4.1 Modèle générique du système distant

Nous avons réalisé quelques essais indiciels sur l'ensemble des axes du manipulateur mobile. Nous avons généré un signal de consigne $c(t)$ formé de créneaux de différentes amplitudes. La figure 2.16 reprend la réponse indicielle qui correspond à la direction du véhicule. Le signal $c_{r_f}(t)$ correspond au signal $c_r(t)$ après passage par le filtre passe-bas cité dans § 2.3.2, page 41.

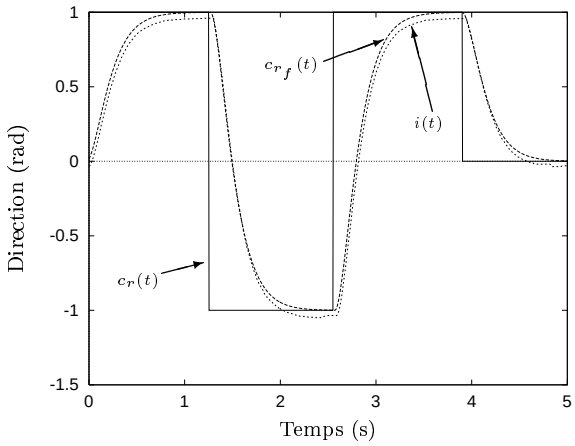


Figure 2.16 – *Réponse indicielle pour la direction*

Grâce à ce filtre d'entrée, tant que le contrôleur local fonctionne correctement, nous pouvons modéliser le système distant en un filtre discret du second ordre $H_r(z)$ sans faire de simplification abusive (cf. équation (2.1)).

$$H_r(z) = \frac{z^{-1}.(a_1 + a_2.z^{-1})}{1 - b_1.z^{-1} - b_2.z^{-2}} \quad (T_{DSP} = 3\ ms) \tag{2.1}$$

En pratique, nous retrouvons la forme de réponse d'un filtre passe-bas échantillonné du second ordre dont les deux pôles sont confondus et correspondent à une constante de temps de l'ordre de 400 ms.

2.4.2 Modèle de la transmission

Méthodes

La modélisation de la transmission repose sur les deux méthodes décrites ci-dessous.

Pour ces deux méthodes, les sites cibles choisis ont été :

- une machine interne au laboratoire : très faible distance,
- une machine externe au laboratoire : à l'IUT de Béziers (environ 75 km),
- une machine située à Paris : moyenne distance (850 km),
- une machine située dans le New-Jersey aux États-Unis (environ 6.000 km) : très grande distance.

Première méthode

Elle utilise le protocole *ICMP*[6] afin de déterminer le temps aller–retour de trames *IP* entre deux machines. Ce type de mesure est classique pour effectuer une mesure brute du temps de réponse (aller–retour) $T_{r_{(A+R)}}(t)$ des trames entre deux machines, indépendamment du protocole de transport (*TCP* ou *UDP*) et de la couche application. Le programme `PING`, présent sur toute machine reliée à un réseau de type *TCP/IP*, est fondé sur ce principe. Connaissant la taille de la trame envoyée S_f, il est également possible d'obtenir un ordre de grandeur du débit D_r entre deux hôtes en effectuant le calcul (2.2).

$$D_r = 2 \times \frac{S_f}{T_{r_{(A+R)}}} \tag{2.2}$$

La longueur de la trame inclut l'en-tête *IP* de 20 octets, l'en-tête *ICMP* de 8 octets et les données supplémentaires éventuellement ajoutées. Dans notre cas, nous avons $20 + 8 + 32 = 60$ octets. Il convient de noter le résultat en kbits/s ou en Mbits/s sachant que 1 kbit = 1.024 bits et 1 Mbit = 1.024 kbits = 1.048.576 bits.

Le déroulement de la mesure consiste en :

- l'envoi de trames de type *ICMP écho* toutes les 100 ms pendant 1 minute,
- la récupération des trames de réponse et calcul du temps d'aller–retour,
- l'étude de l'évolution de ces temps d'aller–retour selon l'éloignement de diverses cibles et en fonction du temps.

6. cf. annexe C, page page 175

```
tracert 194.199.229.110

 Trace l'itineraire vers Salsa.IUTbeziers.Univ-Montp2.FR
 [194.199.229.110] avec un maximum de 30 troncons :
 1 <10 ms  <10 ms  <10 ms  193.49.108.1
 2 <10 ms  <10 ms  <10 ms  192.168.15.2
 3 <10 ms  <10 ms  <10 ms  193.48.168.181
 4 <10 ms  <10 ms  <10 ms 193.48.168.251
 5  30 ms  <10 ms   80 ms  193.48.170.22
 6  10 ms  <10 ms   10 ms 193.50.61.209
 7  10 ms   10 ms   20 ms iut-beziers.r3lr.ft.net [193.50.61.70]
 8  11 ms   10 ms   10 ms router-iut.iutbeziers.univ-montp2.fr [194.199.227.2]
 9  10 ms   10 ms   10 ms Salsa.IUTbeziers.Univ-Montp2.FR [194.199.229.110]
```

Figure 2.17 – *Itinéraire des trames entre deux hôtes du LIRMM et de l'IUT de Béziers*

Cible	**Locale**	**Béziers**	**Paris**	**New Jersey**	
Distance (km)	< 1	75	850	6000	
Horaires	$\emptyset$	$\emptyset$	$\emptyset$	AM	PM
Moyenne (ms)	13,5	22,9	26,7	290	815
Ecart-type (ms)	4,8	7,9	15,5	74	51
Débit moyen (kbits/s)	70	40	35	3,2	1,15

Tableau 2.1 – *Résultats des différentes mesures utilisant* ICMP

D'autre part, il est possible de connaître le chemin qu'empruntent les trames grâce à la commande `TRACEROUTE`. Cette commande affiche, en outre, des renseignements intéressants sur les délais aller–retour entre notre hôte et chacun des intermédiaires tout au long du trajet, ce qui permet de calculer les temps de réponse et débits sur chaque tronçon.

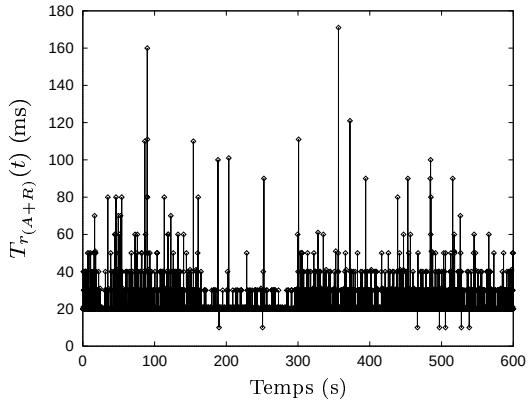
Pour information, l'itinéraire emprunté par les trames entre notre PC et l'hôte situé à l'IUT de Béziers est présenté en figure 2.17. Les trois mesures de temps correspondent à trois sondes simultanées pouvant passer par des chemins différents.

Résultats de cette première méthode

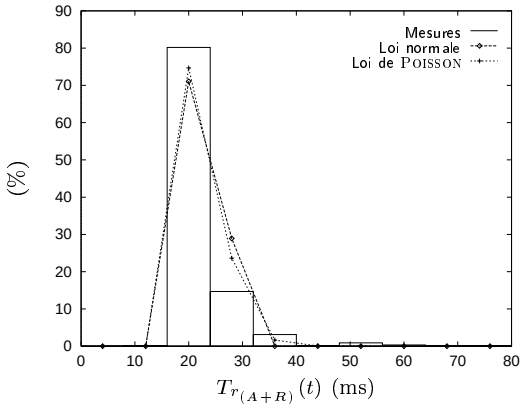
Le tableau 2.1 rassemble les résultats numériques obtenus pour nos différentes cibles.

Le test de communication avec Béziers est celui qui comprend le plus de relevés. C'est donc le plus significatif pour un modèle à « petite distance » (moins de 100 km). La figure 2.18(a) représente l'évolution temporelle des retards aller–retour mesurés. On y constate une nette marge inférieure de l'ordre de 20 ms, très rarement rognée.

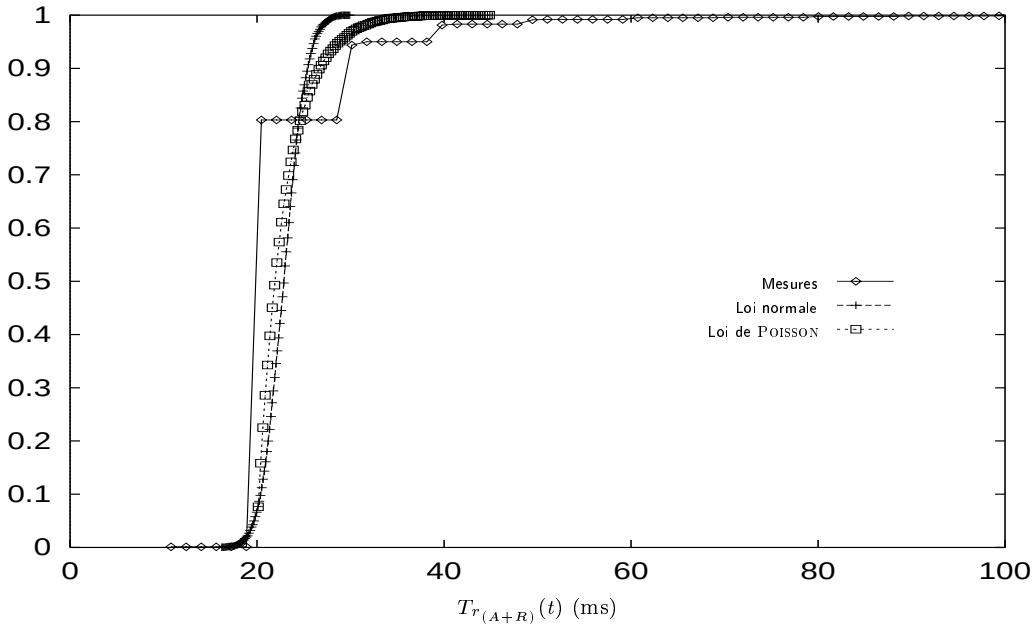
Nous avons estimé que cette valeur de 20 ms correspond à peu près au temps minimum nécessaire pour le transport des trames entre les deux hôtes, compte tenu de la distance et des nombreux routeurs. La distribution normalisée de ces temps de



(a) Délais aller–retour



(b) Distribution



(c) Répartition

Figure 2.18 – *Résultats des mesures entre Montpellier et Béziers avec* ICMP

trajet aller–retour est proposée en figure 2.18(b). Cette distribution est superposée à celle d'une loi gaussienne de moyenne et d'écart-type identiques à ceux des mesures, ainsi qu'à celle d'une loi de POISSON décalée de 20 ms et de moyenne égale à la moyenne des mesures diminuée de 20 ms. Sont également visibles en figure 2.18(c), les fonctions de répartition normalisées de nos mesures et de ces deux lois. Ces deux figures devraient nous permettre de trouver la loi la plus proche. Cependant, il semble difficile de pencher pour l'une ou pour l'autre étant donnée la faible résolution de nos mesures au niveau du coude situé autour de 20 ms. Sur une étude similaire, [OBO 97] a penché pour une exponentielle inverse, ce qui se rapproche plus d'une loi de POISSON que d'une gaussienne.

REMARQUE 1 *En ce qui concerne les hôtes situés à Paris et au New-Jersey, nous n'avons pas émis de séquence continue de* `PING`*s mais des séries de 2 s. En effet, l'émission en continu de ce type de trames sur un hôte peut être interprétée comme une attaque du site par tentative de saturation du réseau cible. Nous n'avons pas eu l'accord des administrateurs réseau ciblés pour effectuer le test complet. Il est à noter toutefois que la bande passante entre la France et les États-Unis est très variable et chute énormément l'après-midi en raison du décalage horaire. C'est la raison pour laquelle nous avons distingué ces deux plages horaires. En outre, pour les États-Unis, de nombreuses trames de réponse (19% le matin comme l'après-midi) n'ont pas été reçues dans le temps limite (fixé arbitrairement à 10 s) et n'entrent pas en compte dans ces calculs. Il convient donc de considérer que ces résultats sont en-dessous de la réalité.*

Seconde méthode

Cette méthode utilise le protocole *TCP* pour se connecter au port écho d'une machine distante. Des trames sont envoyées toutes les 100 ms pendant une minute et nous notons leur temps d'aller–retour à leur réception (le serveur sur lequel nous nous connectons se contente de réémettre les données qu'il reçoit).

Cette méthode est plus significative des temps d'aller–retour pour une application faisant appel à *TCP* car ce délai reflète vraiment ce que nous obtiendrions lors d'une téléopération utilisant ce même protocole de transport.

Résultats de cette seconde méthode

Nous avons étudié deux cas de figures représentatifs. Les résultats des mesures avec l'hôte situé à l'IUT de Béziers sont compilés dans la figure 2.19. Ceux avec l'hôte situé à l'université Rutgers située dans le New-Jersey aux États-Unis le sont dans les figures 2.20 pour le matin et 2.21 pour l'après-midi.

Le tableau 2.2 rassemble les valeurs numériques obtenues pour nos différentes cibles.

Les sous-figures (a) représentent l'évolution des dates d'arrivée. Si l'ensemble des délais aller–retour était nul ou très faible comparé à la période d'émission, nous devrions

Cible	**Locale**	**Béziers**	**Paris**	**New Jersey**	
Distance (km)	< 1	75	850	6000	
Horaires	$\emptyset$	$\emptyset$	$\emptyset$	AM	PM
Moyenne (ms)	<5	9,1	23	1.550	2.550
Ecart-type (ms)	<5	7,2	16	1.180	1.730

Tableau 2.2 – *Résultats des différentes mesures avec* TCP

obtenir la droite de pente 1/1 (en pointillés) superposée aux résultats expérimentaux. Cette représentation permet de détecter des arrivées de données par paquets observées dans les deux cas de liaison Rutgers–Montpellier. Ce phénomène se traduit par des parties plates qui reflètent une arrivée quasi-simultanée d'un groupe de trames initialement émises à période constante. Diverses causes peuvent expliquer ce phénomène. La plus probable est que chaque trame de début de « *bouchon* » a mis plus de temps que les trames suivantes à faire son aller–retour (il se peut qu'elles aient été endommagées durant leur transport, ce qui a obligé l'émetteur à les ré-émettre). Comme *TCP* garantit l'ordre d'arrivée des données, la couche transport du protocole *TCP* a dû stocker toutes les trames suivantes jusqu'à recevoir la trame incriminée et a pu, alors, transmettre l'ensemble des données récupérées d'un seul tenant.

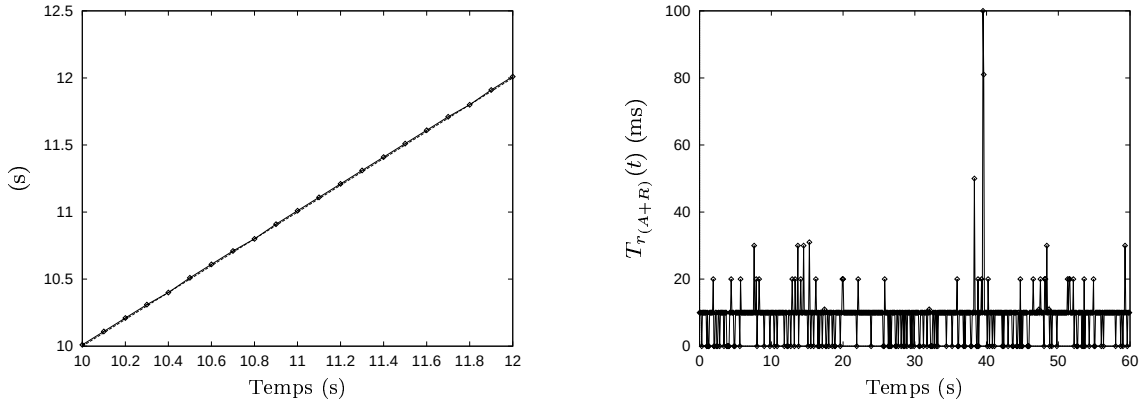
Les sous-figures (b) représentent l'évolution des temps de trajet aller–retour $T_{r_{(A+R)}}(t)$. Nous constatons dans le cas de Béziers qu'en temps normal, $T_{r_{(A+R)}}(t)$ oscille en dessous de 20 ms. Cependant quelques pics dégradent ces performances en montant, par exemple jusqu'à 100 ms. Dans le cas de Rutgers, nous retrouvons les effets de bouchon précédents : nous observons un comportement en « *dents de scie* » qui corrobore les explications précédentes. En effet, lors d'un bouchon, plus les trames sont émises tard après le début du bouchon, moins elles passent de temps bloquées à l'intérieur. Cette diminution est quasi-linéaire car les trames sont émises à période constante et reçues quasiment simultanément.

Les sous-figures (c) représentent la distribution de ces temps de trajet. La résolution de la mesure du temps d'aller–retour est de l'ordre de 10 ms. Ceci est dû aux fonctions de mesures temporelles utilisées lors du développement de cette application. Nous avons appris ultérieurement comment descendre cette résolution jusqu'à 1 ms en faisant appel à d'autres fonctions. Ceci explique la faible qualité des mesures pour les cibles proches (Béziers par exemple) où les valeurs de $T_{r_{(A+R)}}(t)$ sont assez proches de cette résolution.

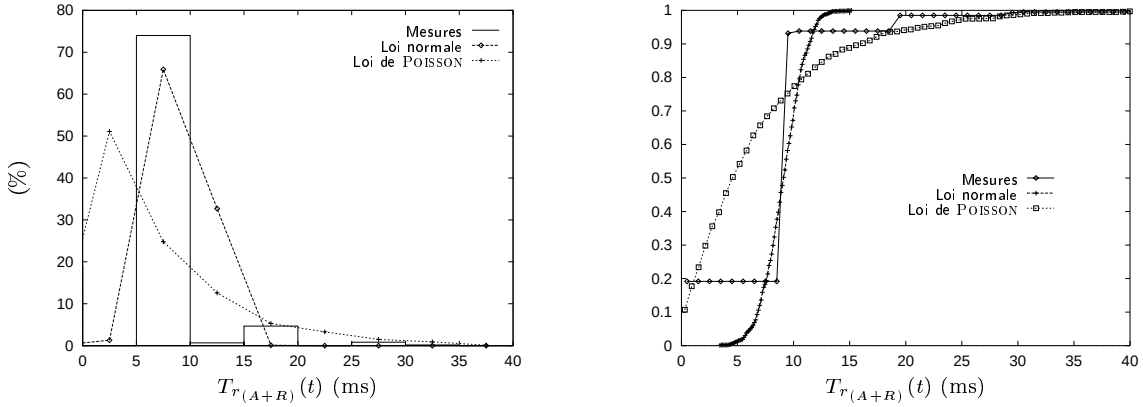
Les sous-figures (d) représentent la fonction de répartition de nos mesures.

Dans ces deux dernières sous-figures, les mesures sont superposées à une loi gaussienne et une loi de Poisson dont les paramètres ont été ajustés pour se rapprocher au mieux (visuellement) les fonctions de distribution normalisée et de répartition.

Dans tous les cas, nous identifions ici une distribution de Poisson, distribution typique des files d'attente. Il faut cependant noter que dans chaque cas, il existe une marge minimale (comme pour *ICMP*) absente pour Béziers (due probablement à la

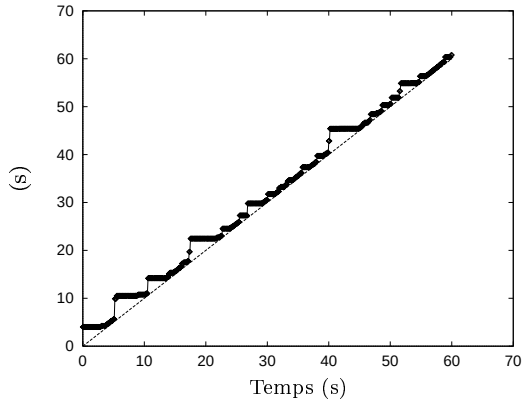


(a) Dates d'arrivée (b) Evolution

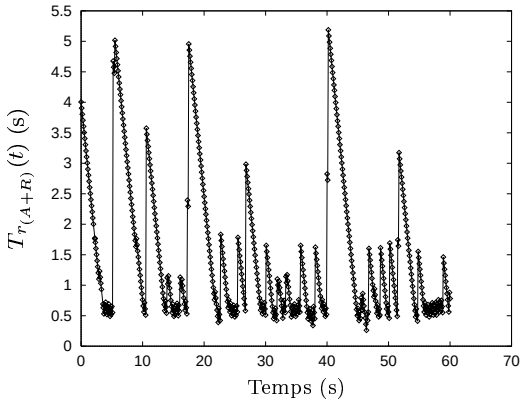


(c) Distribution (d) Répartition

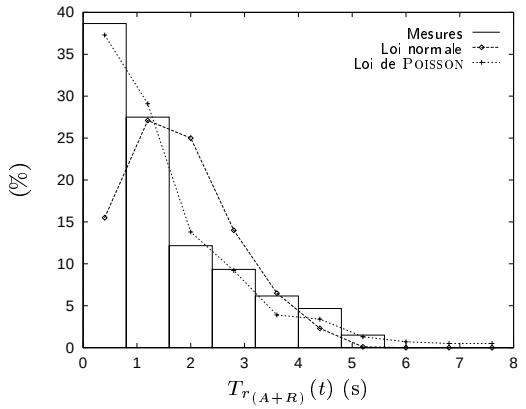
Figure 2.19 – *Résultats des mesures de délais entre Montpellier et Béziers avec* TCP



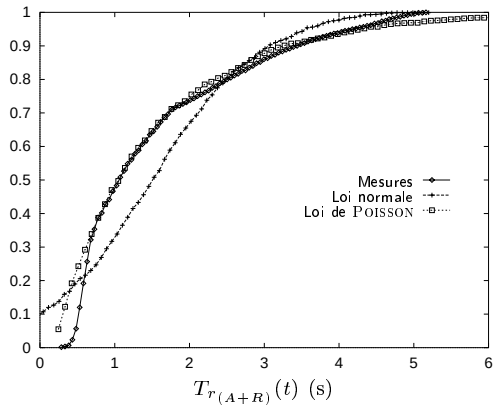
(a) Dates d'arrivée



(b) Evolution



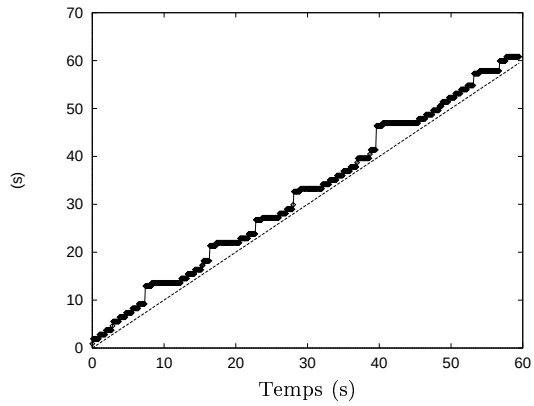
(c) Distribution



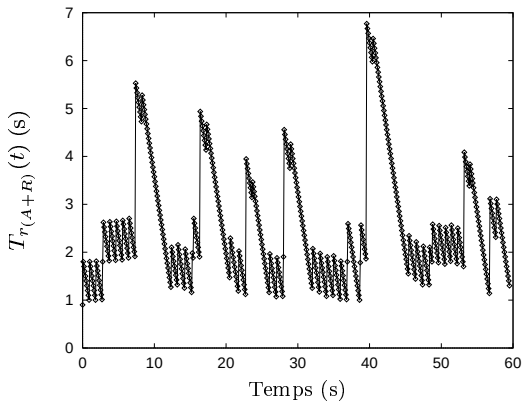
(d) Répartition

Figure 2.20 – *Délais Montpellier — New-Jersey avec* TCP *le matin*

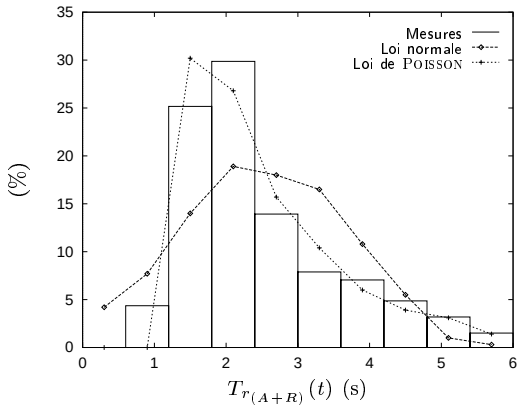
faible résolution de ces mesures et au fait que le laboratoire est relié à l'IUT par un lien direct à 2 Mbits/s), de l'ordre de 200 ms pour Rutgers le matin et 1,3 s l'après-midi.



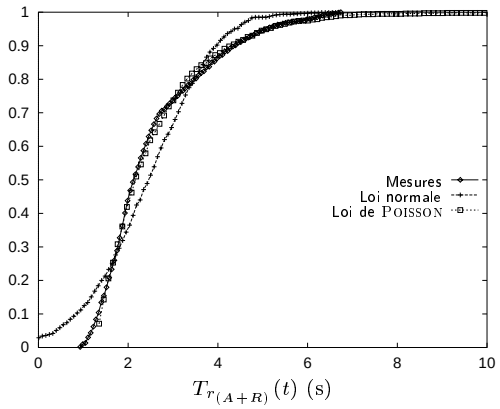
(a) Dates d'arrivée



(b) Evolution



(c) Distribution



(d) Répartition

Figure 2.21 – *Délais Montpellier — New-Jersey avec* TCP *l'après-midi*

Conclusions relatives à ces mesures

Globalement, les mesures des moyennes des temps de trajet aller–retour $T_{r_{(A+R)}}(t)$ de trames de longueur déterminée et constante donnent une idée de la qualité de la transmission en matière de temps de réponse et de bande passante sur une longue période. Ainsi, nous notons une différence notable entre le matin et l'après-midi pour la liaison transatlantique.

L'écart-type semble, lui, mieux représenter la qualité du réseau sur une faible période. En effet, nous pouvons constater au niveau de la liaison avec Rutgers que l'après-midi, l'écart-type n'est pas forcément plus grand que le matin.

Notons que cet écart-type permet d'aider à dimensionner un régulateur de retards comme nous le verrons dans les sections suivantes. En utilisant l'amplitude maximale

des variations de délais sur une période d'écoute, il est possible de dimensionner la file d'attente d'un tel régulateur. Il est toujours préférable d'ajouter un coefficient de sécurité qui peut être justement déduit de cet écart-type.

Les lois de comportement des délais du réseau déduites de l'étude des trames *ICMP* sont une première approche de la modélisation du comportement du réseau. Cependant cette modélisation ne tient pas compte des couches de protocoles situées au-dessus de *IP* qui influent de façon non négligeable sur le comportement des retards.

En ce qui concerne les lois de distribution étudiées avec *TCP*, le fait que nous nous trouvions en présence de lois de POISSON est dû à la couche transport de ce protocole qui gère les trames réceptionnées comme dans une file d'attente.

Les effets de bouchon observés avec ce protocole sont très nuisibles pour une application de téléopération, puisque nous avons fortement intérêt à ce que les trames arrivent au plus vite à leur destination. Cependant si *TCP* n'avait pas remis les données dans l'ordre de départ et récupéré les trames détériorées, nous pouvons imaginer aisément les conséquences que cela pourrait avoir sur le contenu d'un message à sa réception et éventuellement sur la commande d'un robot. Il convient dès lors de déterminer quel avantage est prioritaire : la sécurité des données (au risque d'allonger le temps de transmission) ou la rapidité de transmission (quitte à valider les données réceptionnées ensuite). Notons d'autre part que les tests de validité des trames par sommes de contrôle effectuées dans les différentes couches réseau ne garantissent pas la validité des données à 100% ; chaque couche effectue son propre test à l'aide d'une somme de contrôle (et d'un acquittement pour *TCP*), ce qui limite les erreurs mais ne dispense pas d'effectuer soi-même son propre test au niveau de la couche application quand l'intégrité des données est critique.

Il est encore possible d'affiner le modèle en effectuant les mesures directement avec les applications utilisées sur la plate-forme d'expérimentations, BASE et MANIMOB. Ainsi la couche application sera comprise dans le modèle et le simulateur sera encore plus réaliste. Cependant, le modèle sera à revoir à chaque évolution de ces logiciels.

2.5 Modélisation pour simulations

2.5.1 Organisation

En s'appuyant sur les résultats de § 2.4, page 51, nous avons conçu un système monodimensionnel avec une période de transmission $T_t = 200$ ms. Globalement, le modèle visible sur la figure 2.22 reprend l'architecture en boucle présentée sur la figure 2.1.

Nous retrouvons les signaux $c(t)$, $c_r(t)$, $i(t)$ et $i_r(t)$ introduits au cours de § 2.2, page 37.

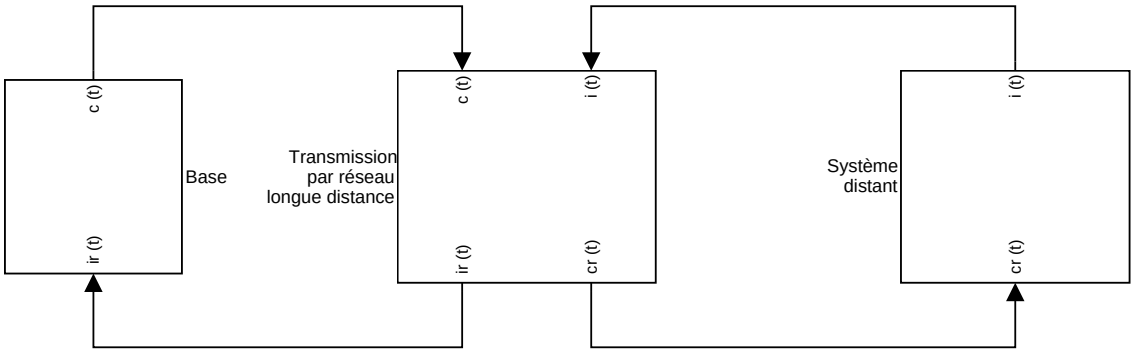


Figure 2.22 – *Organisation générale de la simulation*

2.5.2 Système hybride

Les principales difficultés rencontrées lors de la conception de ce modèle ont été :

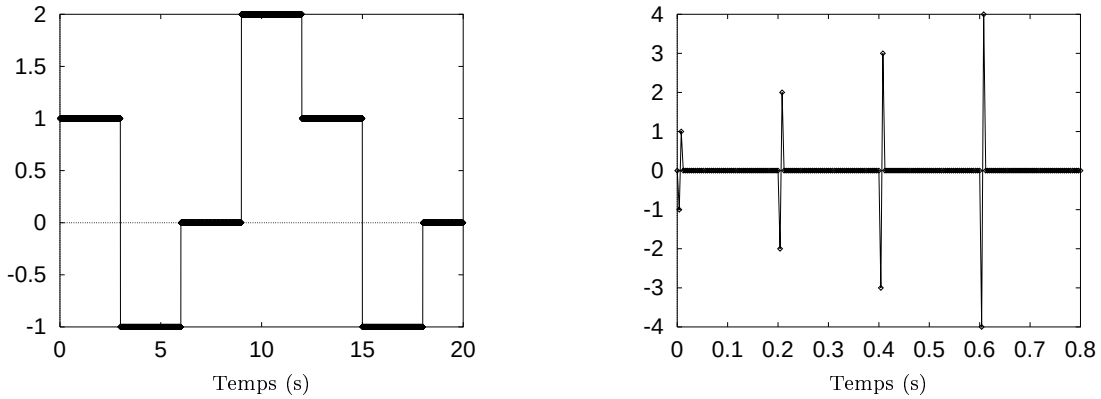
- la présence simultanée de signaux périodiques (notamment $c(t)$ et $i(t)$) et de signaux apériodiques (par exemple $c_r(t)$ et $i_r(t)$),
- la présence de différentes périodes d'échantillonnage (3 ms pour le *DSP*, 200 ms pour les transmissions),
- le manque de fonctions intégrées à Simulink pour gérer des événements discrets dans un tel système hybride.

Nous avons résolu l'ensemble de ces problèmes en dédoublant tous les signaux en :

- un signal de données pseudo-continu (plus exactement, échantillonné au pas de simulation), cf figure 2.23(a) et
- un signal de synchronisation dont la forme est visible figure 2.23(b).

Le signal de synchronisation fait office de signal d'horloge ; le signal de données est à observer à chaque pic du signal de synchronisation. Dans le cas d'un signal périodique (resp. apériodique), le signal de synchronisation est lui-même périodique (resp. apériodique).

Remarque 2 *Le signal de synchronisation sert également à identifier les signaux tout au long de leur parcours à travers les différents blocs ; en effet chaque échantillon j est associé à un pic de hauteur j du signal de synchronisation. Ceci permet notamment de déterminer le temps d'aller–retour des signaux entre la* base *et le* système distant.



(a) Signal de données

(b) Signal de synchronisation

Figure 2.23 – *Exemple de signaux de données et de synchronisation*

2.5.3 Description des modèles initiaux

Le système distant

Nous avons utilisé un modèle générique dérivé de celui présenté en § 2.4, page 51, qui reprend l'équation (2.1).

Afin de rendre ce modèle un peu plus réaliste, nous avons ajouté un signal aléatoire qui fait office de bruit sur la sortie des capteurs du système distant (visible Figure 2.24).

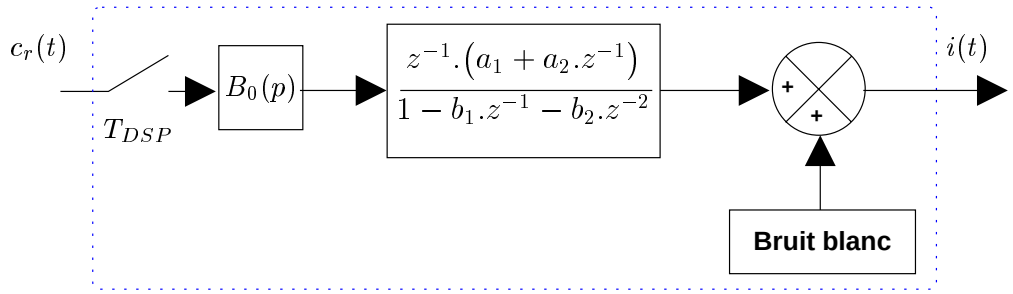


Figure 2.24 – *Modèle de simulation du système distant*

$c_r[k]$ et $i[k]$ correspondent respectivement au signaux $c_r(t)$ et $i(t)$ échantillonnés à la période du *DSP*.

$$i[k] = b_1.i[k-1] + b_2.i[k-2] + a_1.c_r[k-1] + a_2.c_r[k-2] + \omega_k \qquad \forall k \in \mathbb{Z} \qquad (2.3)$$

où ω_k est un bruit blanc dont l'amplitude a été fixée arbitrairement à environ 3% de l'amplitude de $c_r[k]$, ceci afin de simuler un ensemble de bruits électriques et numériques

dégradant légèrement le signal.

La réponse du modèle du système distant à des stimuli en forme de créneaux est présentée figure 2.25.

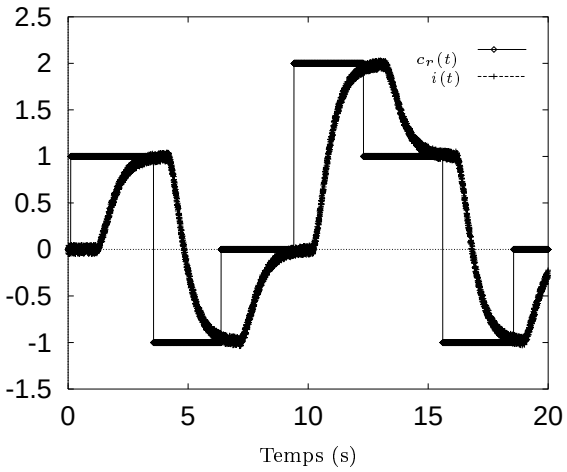


Figure 2.25 – *Réponse du* système distant *simulé à des échelons*

Remarque 3 *Au niveau du modèle de simulation, nous considérons des signaux de transmissions ($c(t)$, $c_r(t)$, $i(t)$ et $i_r(t)$) continus. Pour refléter la réalité où les blocs* base *et* système distant *récupèrent des signaux échantillonnés à la période de transmission T_t, nous avons placé à l'entrée de chacun de ces blocs un échantillonneur bloqueur d'ordre 0.*

La *base*

D'une part, la *base* simule l'action de l'opérateur en envoyant des séquences périodiques de valeurs de consigne. Ces valeurs sont échantillonnées à la période de transmission T_t du système global : ce qui donne le signal $c(t)$. D'autre part, elle récupère les valeurs renvoyées par le système distant sous la forme du signal $i_r(t)$. La figure 2.26 illustre les séquences périodiques émises et les données reçues par la *base*.

Le canal de transmission

Deux blocs identiques (dont un est visible figure 2.27) retardent les signaux en provenance de et en partance pour la *base*.

Pour chaque échantillon entrant dans un bloc de transmission (aller et retour), un retard spécifique $T_{r_{(A|R)}}[k]$ est généré par le sous-bloc « *Générateur de retards aléatoires* » suivant une loi statistique choisie arbitrairement en fonction des cas étudiés en § 2.4.2, page 55.

Les deux blocs sont paramétrés avec le même type de loi statistique avec les mêmes paramètres : nous simulons ainsi un canal de transmission globalement symétrique :

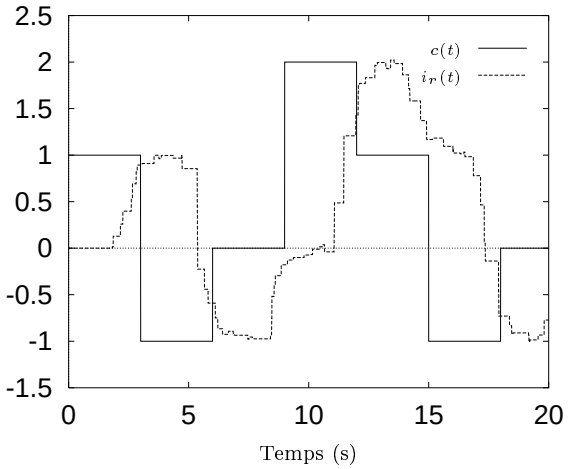


Figure 2.26 – *Exemple de signaux désiré et retour*

$$\overline{T_{r_{(A)}}} = \overline{T_{r_{(R)}}} = \overline{T_r} \quad \text{et} \quad \sigma_{T_{r_{(A)}}} = \sigma_{T_{r_{(R)}}} = \sigma_{T_r}$$

Les échantillons qui entrent dans le « *Programmateur d'échantillons* » en sortent dans le même ordre car le protocole de communication *TCP* garantit l'arrivée des paquets de données dans le bon ordre. Il s'agit d'une file (*First-In, First-Out*) qui empile les échantillons à leur entrée et les désempile ensuite en fonction de leur date de sortie (calculée d'après leur date d'entrée et leur retard affecté) et de leur numéro d'identification fourni par le signal de synchronisation.

La remise en ordre des échantillons a pour effet de modifier la distribution des retards réellement affectés aux données. Faute d'avoir un modèle plus précis des effets du protocole *TCP* sur les données manipulées, nous nous contenterons de ces résultats dans cette étude.

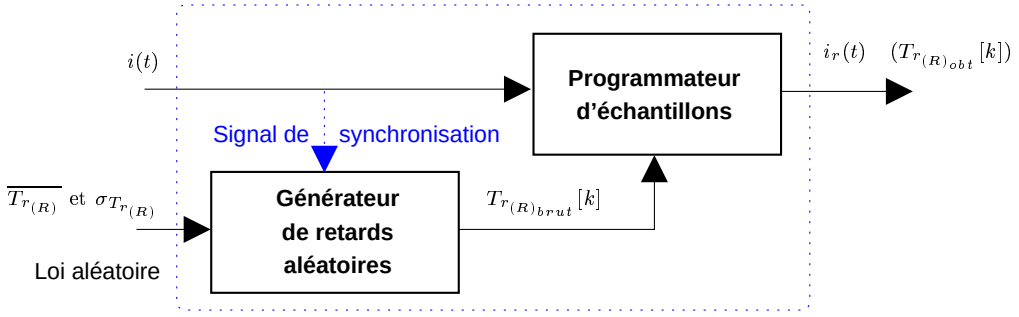


Figure 2.27 – *Modèle de simulation du lien de transmission (ici pour le signal de retour)*

Nous avons testé ce bloc avec les différentes valeurs obtenues en § 2.4.2, page 55, avec le protocole *TCP*. Nous avons paramétré trois séries de valeurs correspondant aux mesures de retard avec Béziers et Rutgers le matin puis l'après-midi. Ces valeurs

Cibles	**Mesuré** (ms)		**Brut** (ms)		**Obtenu** (ms)	
	$\overline{T_{r_m}}$	$\sigma_{T_{r_m}}$	$\overline{T_{r_b}}$	$\sigma_{T_{r_b}}$	$\overline{T_{r_o}}$	$\sigma_{T_{r_o}}$
Béziers	4,55	7,19	5,02	4,03	4,97	4,50
New Jersey AM	770	590	520	420	820	630
New Jersey PM	1270	610	950	490	1330	610

Tableau 2.3 – *Résultats des différentes simulations (en ms)*

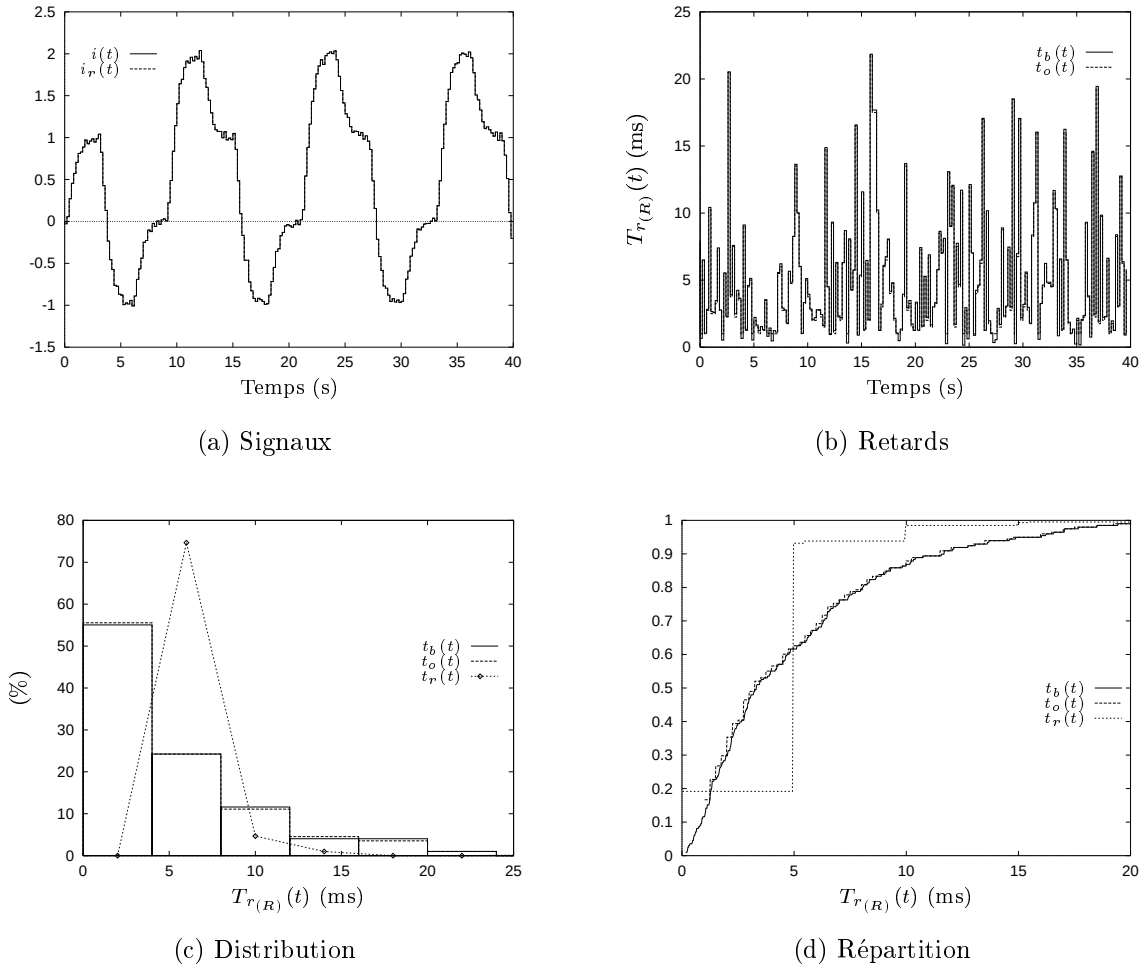
sont extraites du tableau 2.2, page 56, après avoir été divisées par 2. En effet, ces mesures correspondent à des temps de trajet aller–retour tandis que nous avons observé uniquement le bloc « retour » au niveau de la simulation. Le tableau 2.3 présente ces valeurs dans la colonne « *mesuré* » ($\overline{T_{r_{mes}}}$ et $\sigma_{T_{r_{mes}}}$). Les moyennes et écarts-types des retards générés par le bloc générateur de retards aléatoires figurent dans la colonne « *brut* » ($\overline{T_{r_{brut}}}$ et $\sigma_{T_{r_{brut}}}$). La colonne « obtenu » correspond aux mesures au niveau de la sortie du bloc programmateur d'échantillons du lien de transmission au retour : $\overline{T_{r_{obt}}}$ et $\sigma_{T_{r_{obt}}}$.

Le but de la simulation était d'obtenir des lois aléatoires des retards les plus proches de celles observées lors des modélisations précédentes. Ainsi nous avons visuellement réglé les paramètres de génération des valeurs (brutes) de retard $T_{r_{brut}}(t)$ de manière à faire coïncider le mieux possible les courbes de distribution normalisée et de répartition entre les valeurs mesurées et les valeurs obtenues. Ainsi, nous obtenons les conditions de simulation les plus proches des conditions expérimentales observées précédemment.

Nous montrons ici trois cas correspondant aux mesures de § 2.4.2, pages 55 et suivantes.

Les sous-figures 2.28(a), 2.29(a) et 2.30(a) représentent l'effet du canal de transmission sur l'allure des données émises par le *système distant*. $i(t)$ représente les données issues du *système distant*. Ce signal est échantillonné à la période $T_t = 200$ ms. $i_r(t)$ est le signal en sortie du bloc réseau retour. Nous pouvons aisément remarquer pour le cas à longue distance que ce dernier signal n'est pas synchrone et ne correspond pas à une version régulièrement retardée du précédent signal.

Les sous-figures (b),(c) et (d) affichent respectivement l'évolution temporelle, la distribution normalisée et la répartition des retards dans les trois cas.



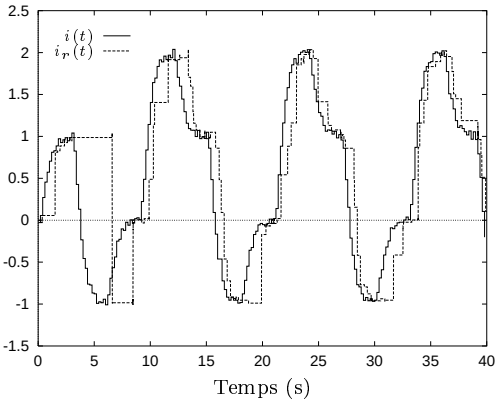
(a) Signaux

(b) Retards

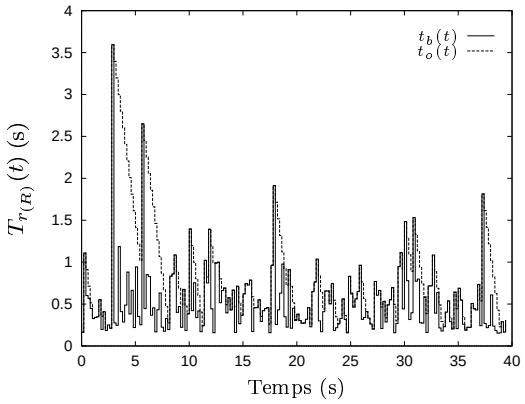
(c) Distribution

(d) Répartition

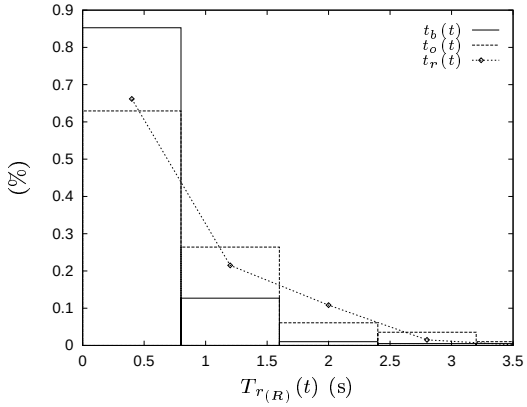
Figure 2.28 – *Signaux et retards obtenus pour une simulation de transmission avec Béziers*



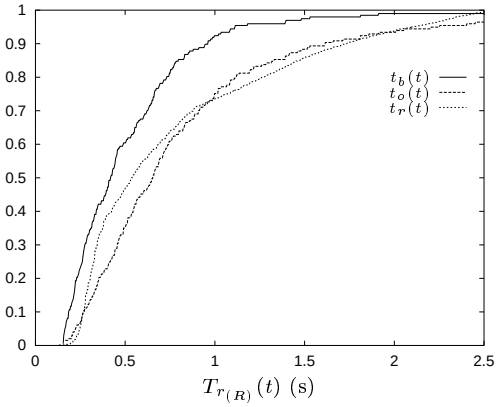
(a) Signaux



(b) Retards

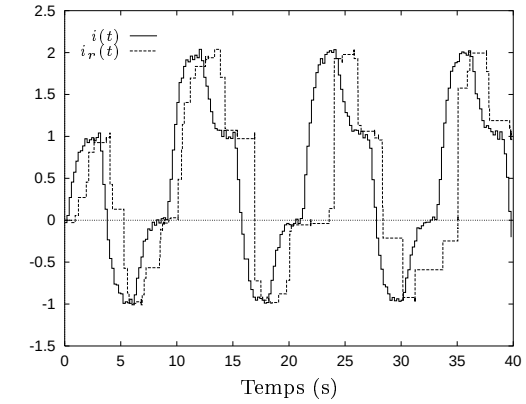


(c) Distribution

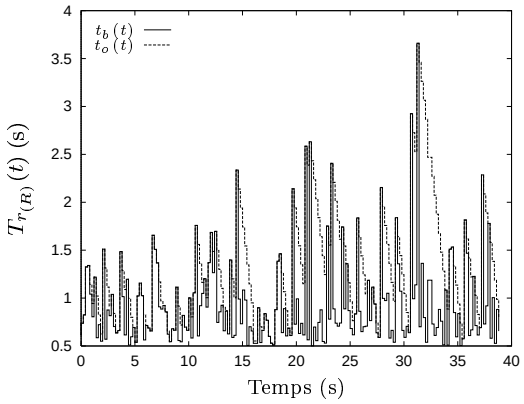


(d) Répartition

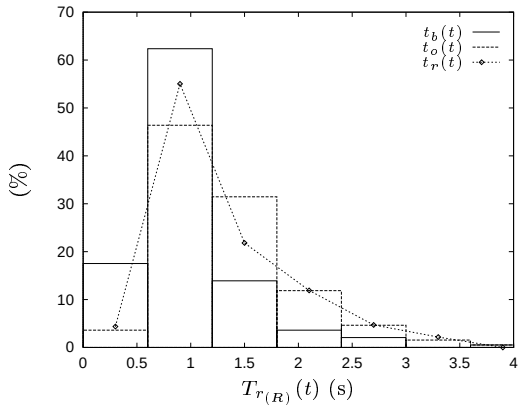
Figure 2.29 – *Signaux et retards obtenus pour une simulation de transmission avec* Rutgers *le matin*



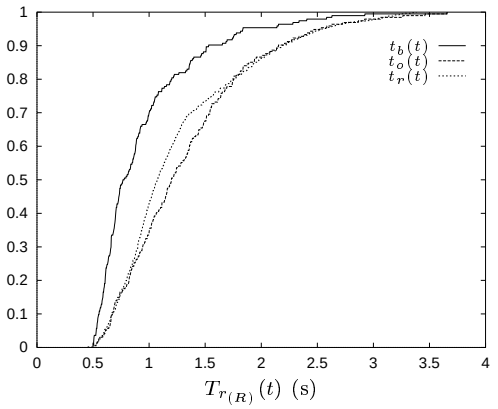
(a) Signaux



(b) Retards



(c) Distribution



(d) Répartition

Figure 2.30 – *Signaux et retards obtenus pour une simulation de transmission avec* Rutgers *l'après-midi*

2.6 Récapitulatif

En partant de mesures et de constatations relatives à notre plate-forme terrestre de téléopération, nous avons abouti à un modèle mathématique d'une boucle de téléopération monodimensionnelle générique.

Pour ce qui concerne les transmissions, celles-ci sont modélisées par un retard pur variable dans le temps selon une loi stochastique dépendant du protocole de communication employé et de la distance entre la *base* et le *système distant*. Dans notre étude, nous avons étudié les protocoles *ICMP* et *TCP* couplés à des distances allant de 0 à 6000 km. Les moyennes et écarts-types des mesures obtenues présentées dans les tableaux 2.1 pour *ICMP* et 2.2 pour *TCP* sont rappelés ci-dessous. Dans l'étude des retards avec le protocole *ICMP*, nos mesures ne nous ont pas permis d'identifier clairement une loi statistique de type normale ou POISSON. Au cours de l'étude des retards avec le protocole *TCP*, nous avons pu identifier la loi statistique à une loi de POISSON associée à un seuil minimum.

ICMP	**Locale**	**Béziers**	**Paris**	**New Jersey**	
Distance (km)	< 1	75	850	6000	
Horaires	$\emptyset$	$\emptyset$	$\emptyset$	AM	PM
Moyenne (ms)	13,5	22,9	26,7	290	815
Ecart-type (ms)	4,8	7,9	15,5	74	51
Débit moyen (kbits/s)	70	40	35	3,2	1,15

TCP	**Locale**	**Béziers**	**Paris**	**New Jersey**	
Distance (km)	< 1	75	850	6000	
Horaires	$\emptyset$	$\emptyset$	$\emptyset$	AM	PM
Moyenne (ms)	<5	9,1	23	1.550	2.550
Ecart-type (ms)	<5	7,2	16	1.180	1.730

Nous avons également identifié le comportement du *système distant*. Nous avons adopté, comme modèle, un filtre échantillonné du second ordre (équation (2.1) rappelée ci-dessous) auquel s'ajoute un bruit blanc dont l'amplitude a été fixée arbitrairement à 3% de l'amplitude du signal non bruité. Dans le cas de notre manipulateur, les deux pôles sont confondus et correspondent à un filtre passe-base dont la constante de temps est de l'ordre de 400 ms.

$$H_r(z) = \frac{z^{-1}.(a_1 + a_2.z^{-1})}{1 - b_1.z^{-1} - b_2.z^{-2}} \quad (T_{DSP} = 3\ ms)$$

Nous avons utilisé ce modèle en simulation afin de mettre en évidence les problèmes à résoudre pour effectuer une téléopération à longue distance. La figure 2.31 présente l'ensemble de la boucle de téléopération modélisée sous Matlab–Simulink [LEL1 99].

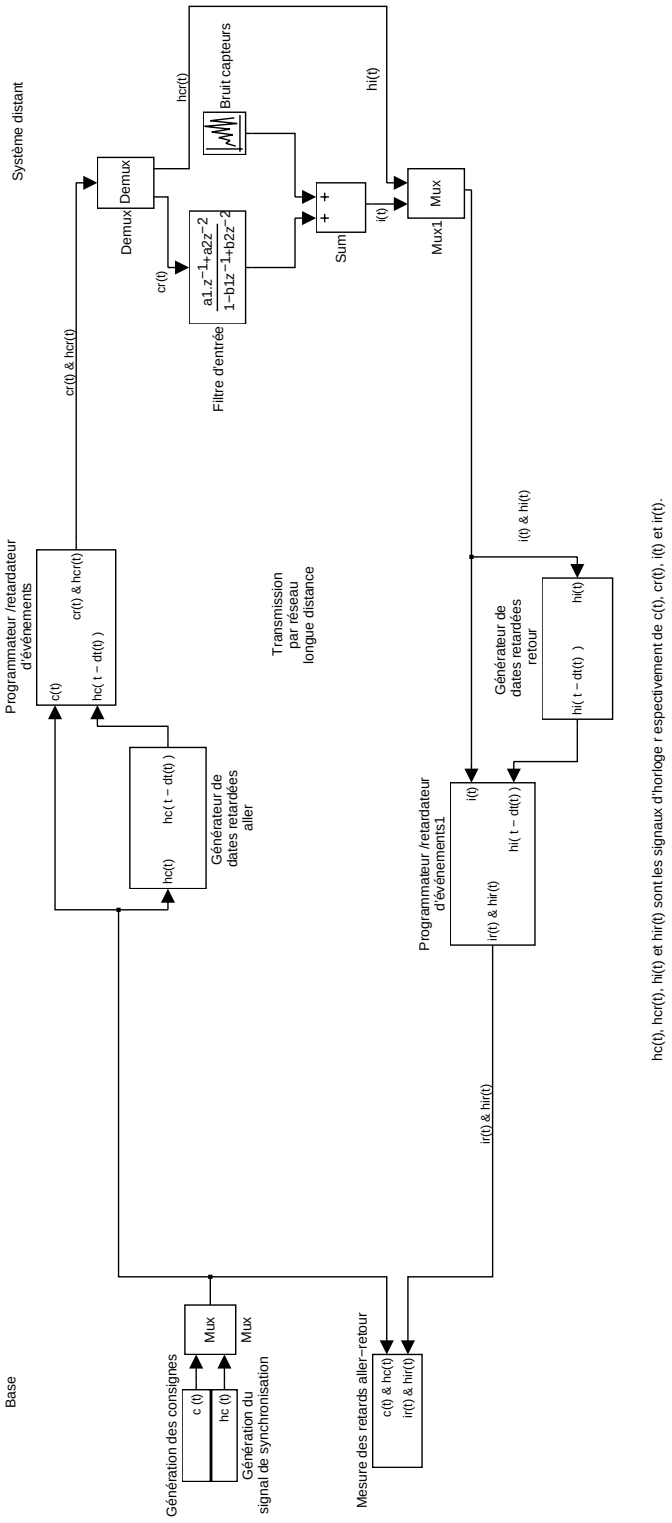


Figure 2.31 – *Modèle global de la boucle de téléopération initiale*

2.7 Conclusion

A partir d'expérimentations préliminaires effectuées sur notre manipulateur mobile terrestre, nous avons élaboré un schéma simple de téléopération générique comprenant le poste opérateur (la *base*), le médium de transmissions bidirectionnel et le système à téléopérer (le *système distant*).

Pour ce faire, nous avons dû développer un ensemble logiciel sous forme d'un client (BASE) et d'un serveur (MANIMOB). Afin de tester ces applications localement, nous avons également développé un programme relais s'intercalant entre BASE et MANIMOB et dont le rôle consiste à infliger des retards aux messages échangés afin de simuler des conditions de travail équivalentes à celles d'une liaison longue distance, sur l'*Internet* par exemple.

En tenant compte des protocoles utilisés, une observation des retards de transmission entre plusieurs hôtes situés à des distances différentes nous a permis d'étudier les caractéristiques de ces retards afin d'en tenir compte et de pouvoir les reproduire le plus fidèlement possible dans nos travaux.

Nous avons ensuite reproduit le schéma de téléopération proposé initialement, sous forme d'un environnement de simulation apte à intégrer de nouvelles fonctionnalités susceptibles d'améliorer ce schéma de téléopération basique.

Ce modèle nous a permis de déterminer les problèmes intrinsèques à la téléopération à longue distance. Le plus important est la variation des retards qui déforment les signaux lors de leurs passages par le lien de transmission à l'aller et au retour. Ces signaux ne sont donc pas aisément exploitables tels quels.

De plus, le fait que l'opérateur obtienne une réponse à ses ordres plus d'une seconde après l'envoi de la consigne, détériore nettement la maniabilité du *système distant*. Comme nous avons pu le constater dans l'état de l'art, il est fortement conseillé de réaliser une prédiction, ou tout ou moins une estimation de l'état du *système distant* pour améliorer l'ergonomie du téléopérateur.

Le chapitre suivant propose des améliorations destinées à palier les principaux problèmes évoqués dans ce chapitre.

Chapitre 3

Études

3.1 Introduction

Ce chapitre aborde différentes propositions pour résoudre les problèmes engendrés par les retards variables de transmission.

En premier lieu, nous mènerons une étude sur l'effet de ces retards sur la stabilité d'un système, du premier ordre d'abord, du second ordre ensuite.

Nous proposons ensuite de réguler les retards de transmission de telle sorte que, globalement, les retards soient constants. C'est l'objet du paragraphe 3.3. Nous pourrons dès lors envisager de transmettre des signaux échantillonnés à une période compatible avec la bande passante du médium de transmissions et obtenir, ainsi, un système isochrone[1].

Enfin, dans la section 3.4, nous proposerons une méthode générique pour estimer l'état du *système distant* depuis la base en temps réel. Cette méthode permet également de prédire le comportement du *système distant* dès l'émission des consignes par l'opérateur. Cette prédiction peut être utilisée ensuite par l'interface homme machine de la *base* afin de réaliser des affichages prédictifs.

Ces travaux ont été testés en simulation dans l'environnement Matlab–Simulink que nous avons développé à cette fin et une partie d'entre eux a été validée expérimentalement sur notre manipulateur mobile terrestre. Ils constitueront les fondements d'un système de téléopération plus évolué faisant appel à des travaux tels que ceux de [CAB 2000]. Ils nous permettront également de tester des lois de commandes en téléopération plus évoluées.

1. Dans notre cas, un système possédant la même période de transmission pour tous les liens de transmission

3.2 Analyse

3.2.1 Système étudié

Pour cette étude, nous nous sommes basés sur les résultats publiés par [HIR 80].

Le système étudié est représenté figure 3.1. Il reprend la structure du schéma de téléopération classique présenté figure 2.1 au paragraphe 2.2. Ici, la partie réseau correspond à un retard pur variable $T_{r_{(A)}}(t)$ pour l'aller et $T_{r_{(R)}}(t)$ pour le retour. Le système distant est modélisé par un filtre linéaire $S(p)$. Au niveau de la *base*, un rebouclage est effectué en utilisant un correcteur proportionnel de gain K.

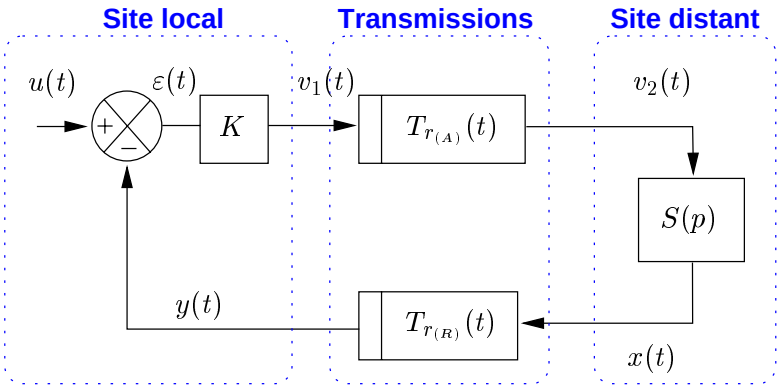


Figure 3.1 – *Système étudié*

Comme dans toute étude de stabilité, nous imposons une consigne nulle $u(t) = 0$. Le bloc de gain K et le bloc de retard $T_{r_{(R)}}(t)$ sont permutables ; en effet, ils agissent indépendamment sur deux caractéristiques différentes du signal : amplitude pour le gain et temps pour le retard. Il est alors possible d'agréger les deux blocs de retard $T_{r_{(A)}}(t)$ et $T_{r_{(R)}}(t)$ en un seul bloc $T_r(t) = T_{r_{(A+R)}}(t) = T_{r_{(A)}}(t) + T_{r_{(R)}}(t)$. Globalement, nous obtenons le système équivalent illustré en figure 3.2.

3.2.2 Système du premier ordre

Dans un premier temps, nous nous plaçons dans le cas où $S(p)$ est un filtre passe-bas du premier ordre de gain G et de constante de temps τ.

$$S(p) = \frac{G}{1 + \tau.p} \tag{3.1}$$

En considérant le système en boucle fermée représenté en figure 3.2, nous obtenons l'équation :

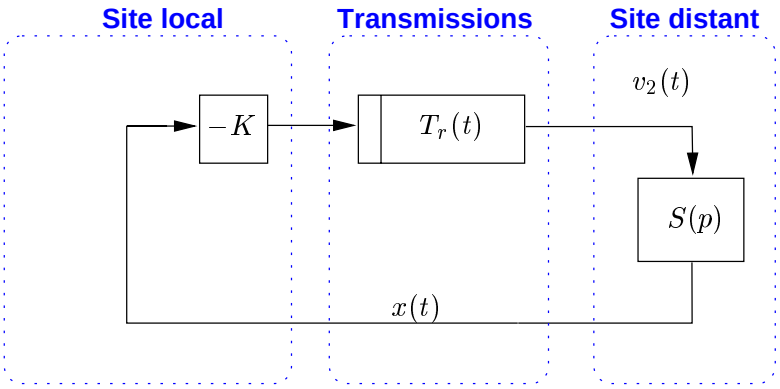


Figure 3.2 – *Système simplifié équivalent*

$$x(t) + \tau.\dot{x}(t) = G.v_2(t) = -G.K.x\big(t - T_r(t)\big) \quad \Leftrightarrow \quad \dot{x}(t) + \frac{x(t)}{\tau} = -\frac{K.G}{\tau}.x\big(t - T_r(t)\big)$$

D'où :

$$\dot{x}(t) + \frac{x(t)}{\tau} + \frac{K.G}{\tau}.x\big(t - T_r(t)\big) = 0$$

Posons $\alpha = \tau^{-1}$ et $\beta = \frac{K.G}{\tau}$, ce qui mène à l'équation différentielle non linéaire du premier ordre (3.2).

$$\dot{x}(t) + \alpha.x(t) + \beta.x\big(t - T_r(t)\big) = 0 \tag{3.2}$$

Nous allons traiter différents cas particuliers de retards :

- retard constant avec différentes conditions de stabilité éventuellement dépendantes de la valeur de ce retard,
- cas particulier de retard variable discontinu : évolution pseudo-linéaire,
- cas général de retard variable continu et borné.

3.2.3 Etude de la stabilité pour un retard constant

Si le retard est constant, $T_r(t) = T_{r_c}$ avec $T_{r_c} \in \mathbb{R}^+$, l'équation (3.2) devient (3.3) :

$$\dot{x}(t) + \alpha.x(t) + \beta.x(t - T_{r_c}) = 0 \tag{3.3}$$

Il est alors possible d'utiliser le formalisme de Laplace pour étudier la stabilité du système. La fonction de transfert en boucle ouverte (équation 3.4) équivalente à (3.3) se déduit aisément de la figure 3.2.

$$H_{BO}(p) = K_c \times e^{-T_{r_c}.p} \times \frac{G}{1+\tau.p} \tag{3.4}$$

Stabilité indépendante du retard

Le critère de TSYPKIN [NIC 97] précise qu'un système dont la fonction de transfert en boucle ouverte telle que :

$$H_{BO}(p) = \frac{P(p).e^{-T_{r_{ci}}.p}}{Q(p)} \qquad \text{avec} \quad deg(P)+1 = deg(Q) = n$$

est stable indépendamment de $T_{r_{ci}}$ si et seulement si :

$$|Q(j.\omega)| > |P(j.\omega)| \qquad \forall\, \omega \in \mathbb{R}$$

Dans notre cas, $P(p) = K_{ci}.G$, $Q(p) = 1+\tau.p$ et $deg(P)+1 = deg(Q) = 1$.

$$|Q(j.\omega)| > |P(j.\omega)| \quad \forall\, \omega \in \mathbb{R} \quad \Leftrightarrow \quad \left|\frac{K_{ci}.G}{1+j.\tau.\omega}\right| < 1 \quad \forall\, \omega \in \mathbb{R} \quad \Rightarrow \quad K_{ci}.G < 1 \tag{3.5}$$

Ici, une vérification de ce résultat est aisée. Considérons l'équation caractéristique du système $T(p) = 1 + H_{BO}(p) = 0$:

$$T(p) = 1 + \tau.p + K_{ci}.G.e^{-T_{r_{ci}}.p} = 0 \tag{3.6}$$

Pour que le système soit stable, il faut que cette équation ne possède que des zéros à partie réelle strictement négative. Recherchons d'abord les racines à partie réelle nulle : $p = j.\omega$.

$$1 + j.\tau.\omega + K_{ci}.G.e^{-j.T_{r_{ci}}.\omega} = 0 \quad \Leftrightarrow \quad \begin{cases} 1 + K_{ci}.G.\cos(T_{r_{ci}}.\omega) = 0 \quad (\Re) \quad \text{et} \\ \tau.\omega - K_{ci}.G.\sin(T_{r_{ci}}.\omega) = 0 \quad (\Im) \end{cases} \tag{3.7}$$

Pour que le système soit stable quel que soit $T_{r_{ci}}$, une condition simple et suffisante est que $1 + K_{ci}.G.\cos(T_{r_{ci}}.\omega)$ ne s'annule jamais pour tout $w \in \mathbb{R}$, soit $K_{ci}.G < 1$.

Revenons alors au cas général où $p = a + j.\omega$. L'équation caractéristique devient alors :

$$(3.6) \quad \Rightarrow \quad \begin{cases} 1 + a.\tau + K_{ci}.G.e^{-a.T_{r_{ci}}}.\cos(T_{r_{ci}}.\omega) = 0 & (\Re) \quad \text{et} \\ \tau.\omega - K_{ci}.G.e^{-a.T_{r_{ci}}}.\sin(T_{r_{ci}}.\omega) = 0 & (\Im) \end{cases} \tag{3.8}$$

Observons la partie réelle de l'équation (3.8). En supposant $K_{ci}.G < 1$, nous constatons qu'il n'existe pas de solution où $a > 0$ car :

$$|K_{ci}.G.e^{-a.T_{r_{ci}}}.\cos(T_{r_{ci}}.\omega)| < 1 < |1 + a.\tau| \quad \forall\, w \in \mathbb{R} \quad (a > 0 \quad \text{et} \quad K_{ci}.G < 1)$$

Ce qui confirme la stabilité du système quel que soit $T_{r_{ci}}$ constant. Cette vérification ne démontre cependant pas que la condition $K_{ci}.G < 1$ est nécessaire. C'est au paragraphe suivant que nous prouverons son caractère nécessaire.

Stabilité dépendant d'un retard maximum

Il peut être intéressant d'agrandir la zone de stabilité du système en considérant que le retard présent dans le système étudié est constant et borné à $T_{r_{cd}}$.

Pour déterminer l'ensemble des valeurs K_{cd} assurant la stabilité du système dans ce cas, nous étudions le système à la limite de la stabilité : $T_r(t) = T_{r_{cd}}$ ($T_{r_{cd}} \in \mathbb{R}^+$) et nous cherchons les racines de l'équation caractéristique présentes sur l'axe imaginaire.

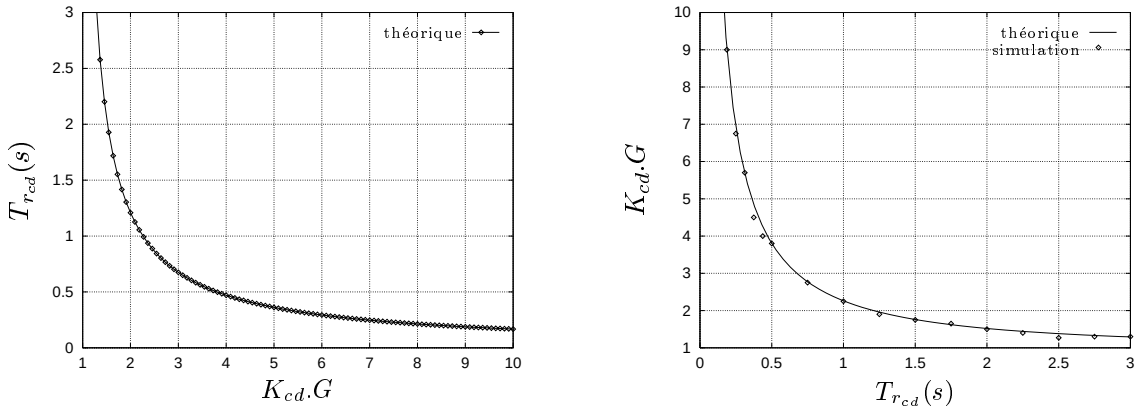
En reprenant le système d'équation (3.7) et en remplaçant $T_{r_{ci}}$ par $T_{r_{cd}}$, on obtient :

$$\begin{cases} \omega.T_{r_{cd}} = \arccos\left(\dfrac{-1}{K_{cd}.G}\right) & (\Re) \quad \text{et} \\ \tau.\dfrac{\omega.T_{r_{cd}}}{T_{r_{cd}}} = K_{cd}.G.\sin(\omega.T_{r_{cd}}) = K_{cd}.G.\sqrt{1 - \cos^2(\omega.T_{r_{cd}})} & (\Im) \end{cases}$$

D'où

$$\begin{aligned} \frac{\tau.\arccos\left(\frac{-1}{K_{cd}.G}\right)}{T_{r_{cd}}} &= K_{cd}.G.\sqrt{1 - \left(\frac{1}{K_{cd}.G}\right)^2} \\ &\Leftrightarrow \\ T_{r_{cd}} &= \frac{\tau.\arccos\left(\frac{-1}{K_{cd}.G}\right)}{\sqrt{(K_{cd}.G)^2 - 1}} \end{aligned} \tag{3.9}$$

Il semble assez difficile d'inverser littéralement cette fonction, cependant une inversion numérique a été effectuée sous Matlab en fixant τ=1 s. La sous-figure 3.3(a) correspond à l'équation (3.9) et la sous-figure 3.3(b) à son inverse. Dans 3.3(b), toute valeur de $K_{cd}.G$ appartenant au demi-plan inférieur à la courbe résulte en un système stable. Nous pouvons vérifier que $T_{r_{cd}} \to +\infty$ quand $K_{cd}.G \to 1^+$, ce qui prouve le caractère nécessaire de la condition envisagée dans la section précédente.



(a) $T_{r_{cd}} = f(K_{cd}.G)$ (b) $K_{cd}.G = f^{-1}(T_{r_{cd}})$

Figure 3.3 – *Représentation graphique de la relation entre $T_{r_{cd}}$ et $K_{cd}.G$*

Validation par simulations

Nous avons vérifié cette inégalité sous Matlab/Simulink avec un système monodimensionnel en prenant plusieurs valeurs de $T_{r_{cd}}$ variant de 0,1 s à 3 s et en fixant arbitrairement $\tau = 1$ s et $G = 1$. Le modèle est présenté sur la figure 3.4.

Pour chaque valeur $T_{r_{cd},j}$, nous avons fixé arbitrairement le gain $K_{cd,j}$ de manière à placer le système en limite de stabilité. Nous avons obtenu une dizaine de points $(T_{r_{cd},j}, K_{cd,j})$ que nous avons superposés à la courbe théorique (sous-figure 3.3(b)). Les résultats ainsi obtenus, même avec une méthode assez approximative, sont conformes aux résultats théoriques précédents.

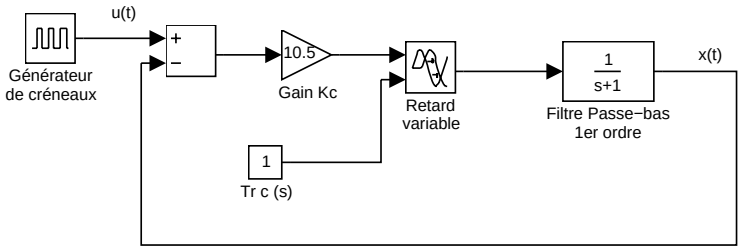


Figure 3.4 – *Modèle de simulation utilisé (retard constant)*

3.2.4 Etude de stabilité pour un retard pseudo-linéaire

Pour cette étude, nous utiliserons le $T_r(t)$ suivant $(T_{r_{pl}} \in \mathbb{R}^{*+})$:

$$T_r(t) = t - n.T_{r_{pl}} \quad \text{pour} \quad n.T_{r_{pl}} < t \leq (n+1).T_{r_{pl}} \quad \text{avec} \quad n \in \mathbb{N} \tag{3.10}$$

C'est une évolution pseudo-linéaire illustrée en figure 3.5. L'intérêt d'un signal pseudo-linéaire est de pouvoir intégrer l'équation (3.2) dans un cas particulier de retard $T_r(t)$ variable et inférieur ou égal à $T_{r_{pl}}$.

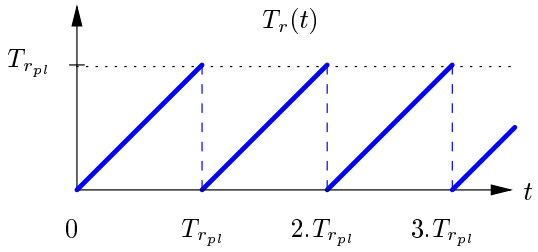


Figure 3.5 – *Evolution de* $T_r(t)$

En tenant compte de l'expression de $T_r(t)$ définie dans (3.10), l'équation (3.2) devient l'équation (3.11) :

$$\dot{x}(t) + \alpha.x(t) + \beta.x\left(n.T_{r_{pl}}\right) = 0 \quad \text{pour} \quad n.T_{r_{pl}} < t \leq (n+1).T_{r_{pl}} \tag{3.11}$$

Stabilité

Pour déterminer la stabilité, il suffit alors d'étudier l'évolution des échantillons $x\left(n.T_{r_{pl}}\right) \quad \forall\, n \in \mathbb{N}$.

Pour déterminer le prochain échantillon de $x(t)$, $x\left((n+1).T_{r_{pl}}\right)$, il suffit d'intégrer l'équation différentielle linéaire du premier ordre (3.11) :

Solution générale de l'équation différentielle ($\alpha = \tau^{-1} \neq 0$) : $x_g(t) = f(t).e^{-\alpha.t}$

Nous réinjectons $x_g(t)$ dans (3.11) pour déterminer $f(t)$ à la constante c près :

$$f(t) = -\frac{\beta}{\alpha}.x\left(n.T_{r_{pl}}\right).e^{-\alpha.t} + c$$

et donc :

$$x(t) = -\frac{\beta}{\alpha}.x\left(n.T_{r_{pl}}\right) + c.e^{-\alpha.t}$$

En identifiant $x(t)$ à $x\left(n.T_{r_{pl}}\right)$ à l'instant $t = n.T_{r_{pl}}$, nous déterminons c et ainsi l'équation récurrente (3.12).

$$x\left((n+1).T_{r_{pl}}\right) = \left\{\left(1 + \frac{\beta}{\alpha}\right).e^{-\alpha.T_{r_{pl}}} - \frac{\beta}{\alpha}\right\}.x\left(n.T_{r_{pl}}\right) \tag{3.12}$$

Contrairement à [HIR 80], nous ne considérons pas la solution pour $\alpha = 0$ puisque cela correspondrait à $\tau \to \infty$, ce qui n'a pas de sens physique dans cette étude.

La stabilité dans ce cas est facilement déterminée : étant donné que nous obtenons une suite géométrique, celle-ci convergera si le module de la raison est strictement inférieur à 1 :

$$|r| = \left| \left(1 + \frac{\beta}{\alpha}\right) .e^{-\alpha.T_{r_{pl}}} - \frac{\beta}{\alpha} \right| \qquad \alpha = \tau^{-1} \quad \text{et} \quad \beta = \frac{K_{pl}.G}{\tau}$$

$$\text{système stable} \quad \Leftrightarrow \quad |r| < 1 \quad \Leftrightarrow \quad \left| (1 + K_{pl}.G).e^{-T_{r_{pl}}/\tau} - K_{pl}.G \right| < 1$$

Soit

$$\begin{cases} K_{pl}.G.(e^{-T_{r_{pl}}/\tau} - 1) + e^{-T_{r_{pl}}/\tau} > 0 \quad \text{et} \quad K_{pl}.G.(e^{-T_{r_{pl}}/\tau} - 1) + e^{-T_{r_{pl}}/\tau} < 1 \\ K_{pl}.G.(e^{-T_{r_{pl}}/\tau} - 1) + e^{-T_{r_{pl}}/\tau} < 0 \quad \text{et} \quad K_{pl}.G.(1 - e^{-T_{r_{pl}}/\tau}) - e^{-T_{r_{pl}}/\tau} < 1 \end{cases}$$

Ce qui revient à :

$$\begin{cases} K_{pl}.G < \dfrac{e^{-T_{r_{pl}}/\tau}}{1 - e^{-T_{r_{pl}}/\tau}} \quad \text{et} \quad K_{pl}.G > \dfrac{e^{-T_{r_{pl}}/\tau} - 1}{1 - e^{-T_{r_{pl}}/\tau}} \\ K_{pl}.G > \dfrac{e^{-T_{r_{pl}}/\tau}}{1 - e^{-T_{r_{pl}}/\tau}} \quad \text{et} \quad K_{pl}.G < \dfrac{1 + e^{-T_{r_{pl}}/\tau}}{1 - e^{-T_{r_{pl}}/\tau}} \end{cases}$$

Or $0 < e^{-T_{r_{pl}}/\tau} < 1$ et $K_{pl}.G > 0$ d'où :

$$\begin{cases} 0 < K_{pl}.G < \dfrac{e^{-T_{r_{pl}}/\tau}}{1 - e^{-T_{r_{pl}}/\tau}} \\ \dfrac{e^{-T_{r_{pl}}/\tau}}{1 - e^{-T_{r_{pl}}/\tau}} < K_{pl}.G < \dfrac{1 + e^{-T_{r_{pl}}/\tau}}{1 - e^{-T_{r_{pl}}/\tau}} \end{cases} \quad \Leftrightarrow \quad 0 < K_{pl}.G < \frac{1 + e^{-T_{r_{pl}}/\tau}}{1 - e^{-T_{r_{pl}}/\tau}} \qquad (3.13)$$

$$(3.13) \quad \Leftrightarrow \quad 0 < K_{pl}.G < \left[\tanh\left(\frac{T_{r_{pl}}}{2.\tau}\right) \right]^{-1} \qquad (3.14)$$

La figure 3.7 illustre l'évolution de $K_{pl}.G$ en fonction de la valeur maximale $T_{r_{pl}}$ du retard $T_r(t)$. Lorsque le retard est nul, le système en boucle ouverte étant du premier ordre, il sera toujours stable après rebouclage, ceci quel que soit le gain global $K_{pl}.G$. Cette propriété se vérifie sur cette courbe par $K_{pl}.G \to +\infty$ quand $T \to 0^+$.

Remarque 4 *Dans ce cas de retard variable, on retrouve également la condition :*

$$\left[\textit{système stable indépendamment de } T_{r_{pl}}\right] \Leftrightarrow K_{pl}.G < 1$$

En effet, $\left[\tanh\left(\frac{T_{r_{pl}}}{2.\tau}\right)\right]^{-1} > 1$ *pour tout* $T_{r_{pl}} \in \mathbb{R}^{*+}$. *Ainsi le système est stable quel que soit* $T_{r_{pl}}$ *dans la bande* $0 < K_{pl}.G < 1$.

Extension à un système linéaire de dimension quelconque

Supposons qu'il soit possible de modéliser le système par la représentation d'état présentée en (3.15) :

$$\dot{X}(t) + A.X(t) + B.X\big(t - T_r(t)\big) = 0 \tag{3.15}$$

Il est alors possible d'étendre la conclusion précédente pour le même $T_r(t)$. En effet, (3.15) avec (3.10) donne (3.16) :

$$\dot{X}(t) + A.X(t) + B.X(n.T_{r_{pl}}) = 0 \tag{3.16}$$

qui, une fois intégrée, donne (3.17), A étant supposée inversible (autrement dit, le système ne possède que des pôles simples) :

$$\begin{aligned} X(n.T_{r_{pl}} + \delta t) &= e^{A.\delta t}.X(n.T_{r_{pl}}) + \int_0^{\delta t} e^{A.(n.T_{r_{pl}} - \sigma)}.B.X(n.T_{r_{pl}}).d\sigma \\ &= \left(e^{A.\delta t} + A^{-1}.\left[e^{A.\delta t} - I\right].B\right).X(n.T_{r_{pl}}) \end{aligned} \tag{3.17}$$

Ainsi, on obtient l'équation récurrente (3.18)

$$X\big((n+1).T_{r_{pl}}\big) = \left(e^{A.T_{r_{pl}}} + A^{-1}.\left[e^{A.T_{r_{pl}}} - I\right].B\right).X(n.T_{r_{pl}}) = \Phi.X(n.T_{r_{pl}}) \tag{3.18}$$

Et la condition de stabilité consiste à ce que les valeurs propres de Φ soient toutes de module strictement inférieur à 1.

Simulations

Nous avons vérifié cette inégalité sous Matlab/Simulink avec un système monodimensionnel en prenant plusieurs valeurs de $T_{r_{pl}}$: 100 ms, 200 ms, ... 1 s et en fixant arbitrairement $\tau = 1$ s et $G = 1$. Le modèle est présenté sur la figure 3.6. Nous avons adopté la même méthode que pour le cas d'un retard constant pour déterminer les différentes valeurs de K_{pl} en fonction de $T_{r_{pl}}$. Nous avons ainsi obtenu une dizaine de points $(T_{r_{pl},j},\ K_{pl,j})$ que nous avons superposés à la courbe théorique (figure 3.7) représentant l'équation (3.14). Les résultats obtenus, même avec une méthode assez approximative, sont conformes aux résultats théoriques précédents.

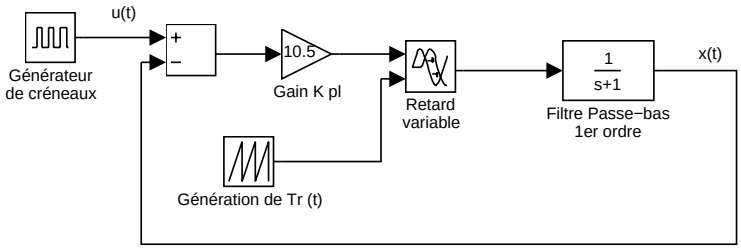


Figure 3.6 – *Schéma de simulation utilisé (retard pseudo-linéaire)*

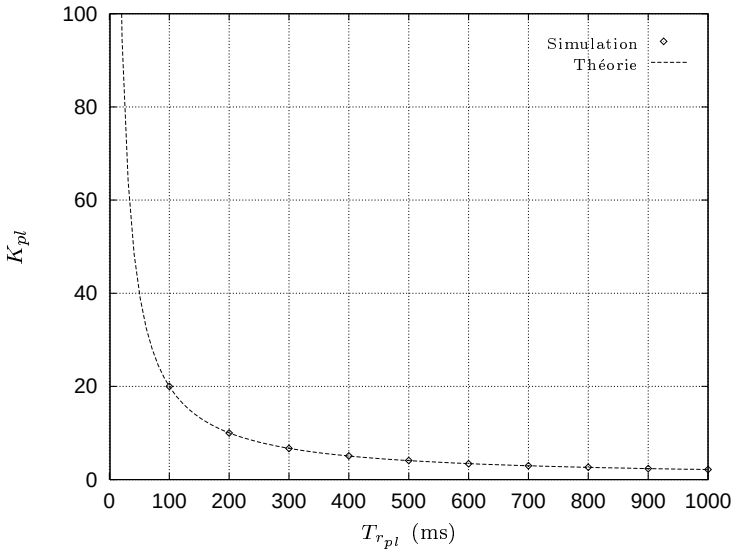
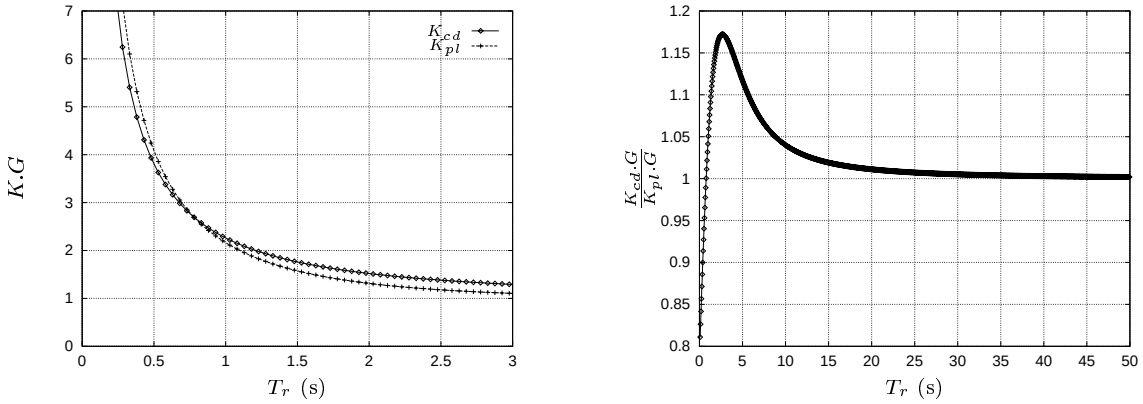


Figure 3.7 – *Evolution du gain limite de stabilité (retard pseudo-linéaire)*

3.2.5 Comparaison des résultats

Afin de se rendre compte de l'impact de la variation du retard $T_r(t)$ sur la condition de stabilité, la sous-figure 3.8(a) permet de visualiser la superposition des deux courbes $K.G = f(T_r)$ dans les cas *retard constant* et *retard pseudo-linéaire*. La sous-figure 3.8(b) représente le rapport $\frac{K_{cd}.G}{K_{pl}.G} = f(T_r)$.



(a) Superposition

(b) Rapport

Figure 3.8 – *Comparaison des gains limites de stabilité dans les deux cas étudiés*

De manière empirique, nous pouvions imaginer que le cas d'un retard variable, par exemple pseudo-linéaire, serait globalement moins stable qu'un retard constant de même valeur maximale. La figure 3.8 permet de se rendre compte que pour de faibles amplitudes (un peu moins d'une seconde), un système à retards variables pseudo-linéaires est plus stable qu'un système à retards constants. Cette constatation est importante car nous pouvions penser a priori qu'un retard constant était préférable du point de vue de la stabilité, dans tous les cas.

En fait, nous pensons qu'il faudrait étudier ce signal en le décomposant en séries de Fourrier et analyser la condition de stabilité en fonction des harmoniques. Il s'agit d'une piste à étudier ultérieurement...

D'autre part, il semble que plus le retard augmente, moins l'allure du retard (ici pseudo-linéaire) influe sur la condition de stabilité. En effet, le rapport $\frac{K_{cd}.G}{K_{pl}.g}$ tend vers 1^+ quand T_r tend vers $+\infty$.

3.2.6 Autres méthodes

La méthode précédente d'étude de stabilité utilisée dans le cas d'un retard constant utilise une approche fréquentielle. Il existe une autre approche fréquentielle fondée sur les faisceaux matriciels, commentée par [NIC 97], qui donne les mêmes résultats.

Il existe également des méthodes applicables aux retards variables fondées sur une approche temporelle utilisant une *technique par fonction de* Lyapunov–Razumikhin.

Cette approche aboutit au théorème 1 :

THÉORÈME 1
Soit le système scalaire défini par l'équation (3.3) avec :

$$x(t) \in \mathcal{C}^1 \quad \forall t \in [-T_{r_{max}}, \infty], \qquad 0 \leqslant T_r(t) < T_{r_{max}} \qquad et \quad T_r(t) \in \mathcal{C}^0.$$

1. *Si $\alpha \geqslant |\beta|$ et $\alpha + \beta > 0$, alors le système est stable indépendamment de la borne supérieure $T_{r_{max}}$ du retard $T_r(t)$.*

2. *Si $\beta > |\alpha|$, alors la stabilité est garantie tant que $T_r(t) < T_{r_{max}}$ avec :*

$$T_{r_{max}} = \frac{\alpha + \beta}{\beta.|\alpha| + \beta^2}$$

Dans le cas du système monodimensionnel du premier ordre, la condition (1) est équivalente à $K_{max}.G \leqslant 1$. La valeur limite $T_{r_{max}}$ de la condition (2) est :

$$T_{r_{max}} = \frac{\frac{1}{\tau} + \frac{K_{max}.G}{\tau}}{\frac{K_{max}..G}{\tau^2} + \left(\frac{K_{max}..G}{\tau}\right)^2} = \tau \times \frac{1 + K_{max}.G}{K_{max}.G \times (1 + K_{max}.G)} = \frac{\tau}{K_{max}.G}$$

Simulations

Nous avons testé l'application de ce dernier théorème au système monodimensionnel en employant la même méthode que pour les précédentes simulations. Le retard $T_r(t)$ est une fonction sinusoïdale oscillant entre 0 et $T_{r_{max}}$. Le modèle de simulation utilisé est représenté figure 3.9.

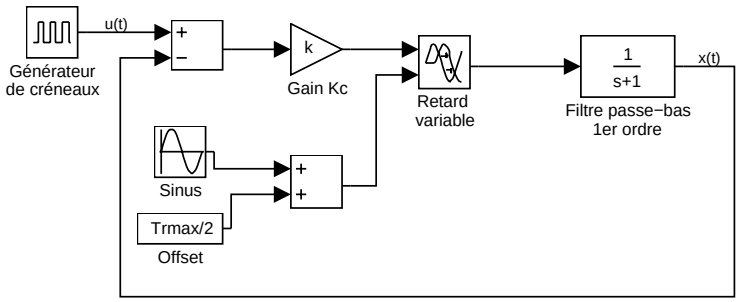


Figure 3.9 – *Modèle de simulation utilisé (cas sinusoïdal)*

Globalement, nous sommes arrivés aux mêmes conclusions que dans le cas particulier pseudo-linéaire ; nous retrouvons bien la même fonction $K_{max} = f(T_{r_{max}})$ (cf figure 3.10)

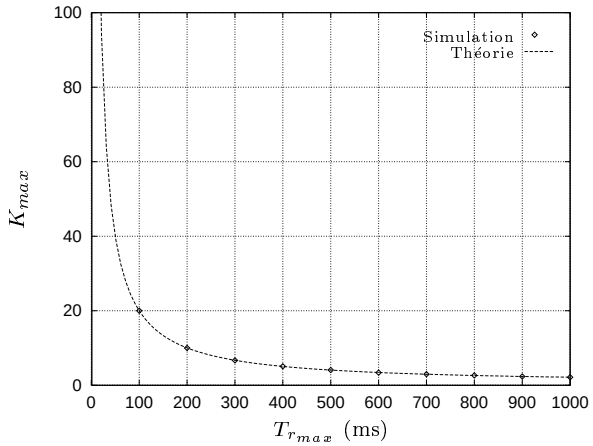
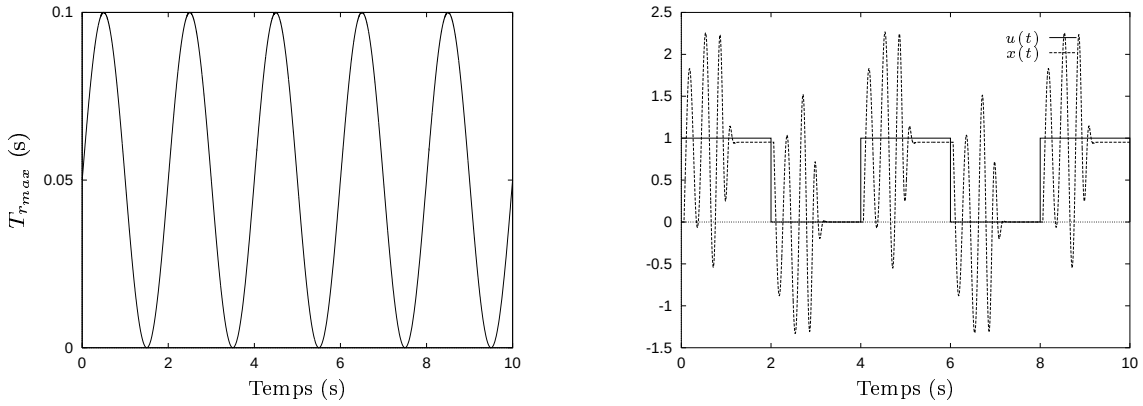


Figure 3.10 – *Evolution du gain limite de stabilité (retard sinusoïdal)*

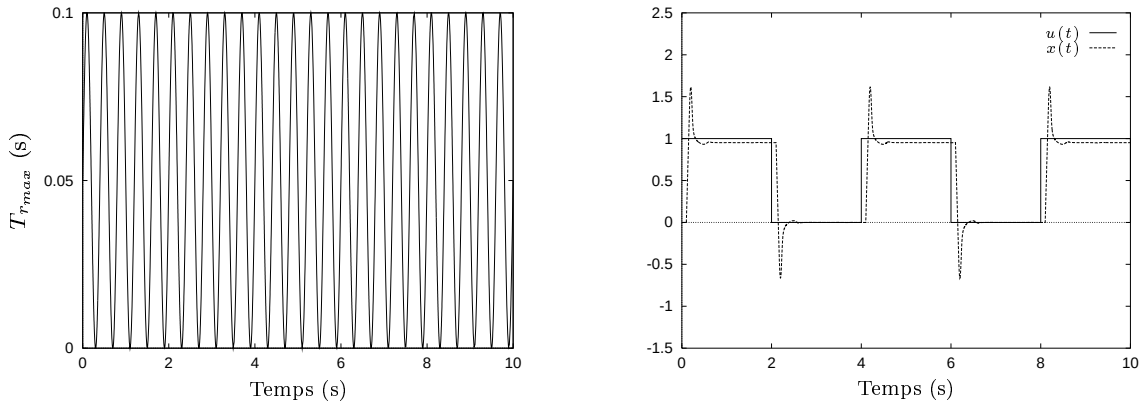
Nous avons reproduit trois cas dans les mêmes conditions $(K_{max}, T_{r_{max}})$ mais avec la fréquence du signal variable : 1, 5 et 10 Hz, résultats représentés figures 3.11, 3.12 et 3.13.



(a) Retard (b) Signaux

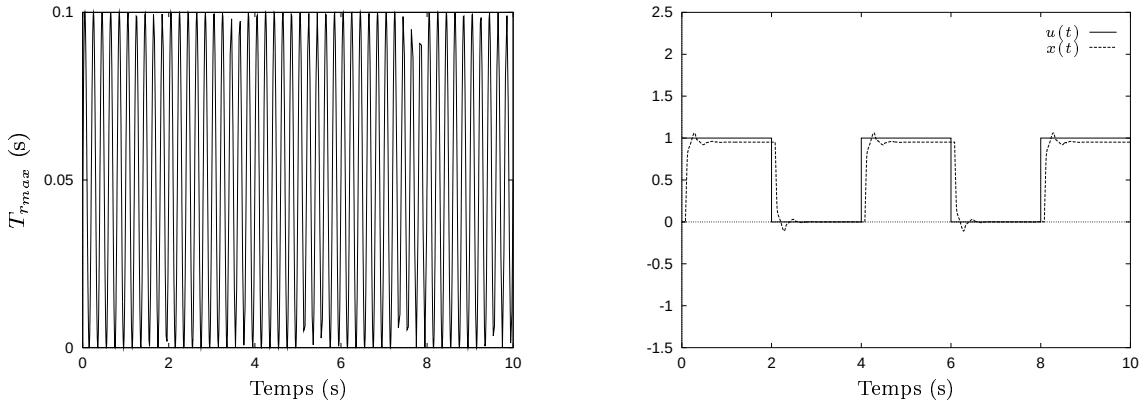
Figure 3.11 – *Evolution des signaux pour un retard sinusoïdal de fréquence 1 Hz*

Nous pouvons ainsi constater que lorsque le retard est périodique, sa période n'influe pas sur la stabilité mais agit énormément sur l'allure des signaux (notamment sur les oscillations en mode transitoire).



(a) Retard (b) Signaux

Figure 3.12 – *Evolution des signaux pour un retard sinusoïdal de fréquence 5 Hz*



(a) Retard (b) Signaux

Figure 3.13 – *Evolution des signaux pour un retard sinusoïdal de fréquence 10 Hz*

3.2.7 Système du second ordre

Reprenons le schéma de la figure 3.2 avec, cette fois-ci, un système du second ordre avec retards constants $T_{r_{(A)}}$ et $T_{r_{(R)}}$ ($T_{r_c} = T_{r_{(A)}} + T_{r_{(R)}}$) ; le système simplifié est représenté figure 3.14. $(K_d, K_p, K, G) \in (\mathbb{R}^{*+} \times \mathbb{R}^{*+} \times \mathbb{R}^{*+} \times \mathbb{R}^{*+})$.

$$S(p) = \frac{G.e^{-T_{r_c}.p}}{p^2 + K_d.p + K_p} \tag{3.19}$$

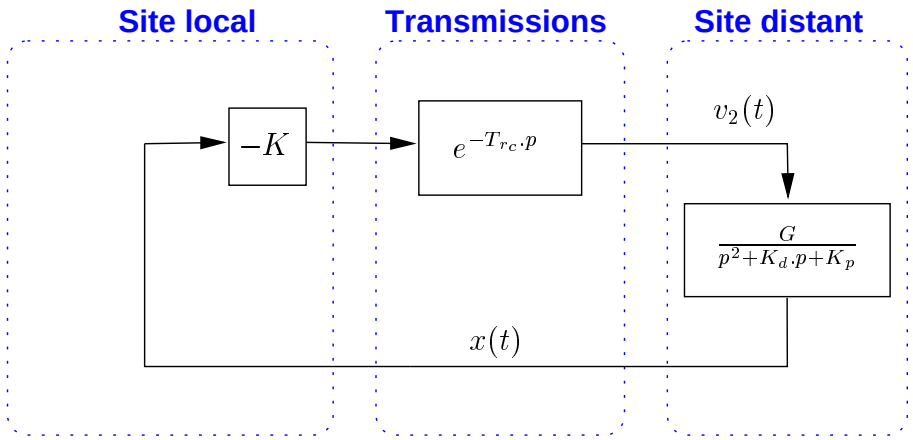


Figure 3.14 – *Système du second ordre étudié*

La fonction de transfert en boucle ouverte $H_{BO}(p)$ et l'équation caractéristique $T(p) = 0$ sont données respectivement dans les équations (3.20) et (3.21).

$$H_{BO}(p) = K.S(p) = \frac{K.G.e^{-T_{r_c}.p}}{p^2 + K_d.p + K_p} \tag{3.20}$$

$$T(p) = 1 + H_{BO}(p) = p^2 + K_d.p + K_p + K.G.e^{-T_{r_c}.p} = 0 \tag{3.21}$$

Afin de déterminer quel gain K permet d'obtenir un système stable en fonction du retard total T_{r_c}, nous allons employer la méthode fréquentielle à l'instar du cas du premier ordre.

Etudions le cas limite des pôles situés sur l'axe imaginaire. En posant $p = j.\omega$ ($\omega \in \mathbb{R}$), l'équation caractéristique devient (3.22).

$$(3.21) \Leftrightarrow \begin{cases} -\omega^2 + K_p + K.G.\cos(\omega.T_{r_c}) = 0 \quad (\Re) \quad \text{et} \\ K_d.\omega - K.G.\sin(\omega.T_{r_c}) = 0 \quad (\Im) \end{cases} \tag{3.22}$$

Condition suffisante

Reprenons la partie imaginaire de l'équation (3.22). Soient $f_1(\omega)$ et $f_2(\omega)$ les fonctions définies par l'équation (3.23).

$$\begin{cases} f_1(\omega) = K_d.\omega \\ f_2(\omega) = K.G.\sin(\omega.T_{r_c}) \end{cases} \tag{3.23}$$

La figure 3.15 représente ces deux fonctions avec $K_d = 2$, $T_{r_c} = 1$ s et $K.G = 1$.

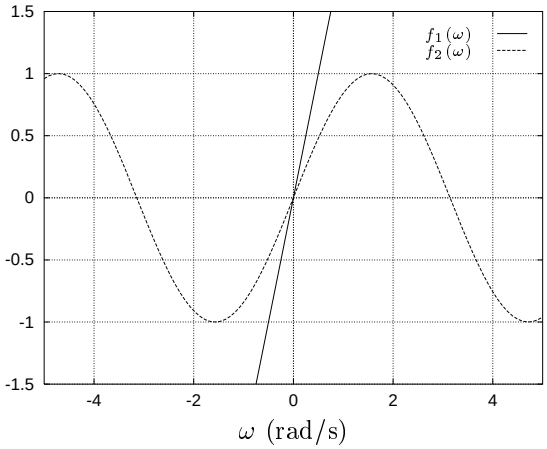


Figure 3.15 – *Fonctions $f_1(\omega)$ et $f_2(\omega)$*

En observant ces deux fonctions, nous pouvons affirmer que si la valeur absolue de la pente à l'origine de $f_1(\omega)$ est supérieure à celle de $f_2(\omega)$, alors, il n'existera qu'une seule solution $\omega_0 = 0$ annulant la partie imaginaire de l'équation caractéristique (3.22). Cette condition est exprimée par l'équation (3.24).

$$\left|\frac{\mathrm{d}f_1(\omega)}{\mathrm{d}\omega}_{(\omega=0)}\right| \geqslant \left|\frac{\mathrm{d}f_2(\omega)}{\mathrm{d}\omega}_{(\omega=0)}\right| \Leftrightarrow K_d \geqslant K.G.T_{r_c} \Leftrightarrow K.G \leqslant \frac{K_d}{T_{r_c}} \tag{3.24}$$

Nous devons vérifier que la solution de la partie imaginaire $\omega_0 = 0$ est aussi solution de la partie réelle :

$$\Re(T(j.\omega_0)) = 0 + K_p + K.G > 0 \tag{3.25}$$

Celle-ci est strictement positive donc $\omega_0 = 0$ n'est pas solution de l'équation caractéristique. Ainsi, lorsque l'inégalité (3.24) est respectée, il n'existe pas de pôle sur l'axe imaginaire. Nous allons maintenant vérifier qu'il n'existe pas non plus de pôle instable, autrement dit de pôle appartenant au demi-plan droit.

Avec $p = a + j.b$, $a \in \mathbb{R}^{*+}$ et $b \in \mathbb{R}^*$ (le point (0,0) a déjà été étudié), l'équation caractéristique (3.21) devient (3.26).

$$(3.21) \Leftrightarrow \begin{cases} a^2 - b^2 + K_d.a + K_p + K.G.e^{-a}\cos(b.T_{r_c}) = 0 \quad (\Re) \quad \text{et} \\ 2.a.b + K_d.b - K.G.e^{-a}.\sin(b.T_{r_c}) = 0 \quad (\Im) \end{cases} \tag{3.26}$$

Observons une fois de plus la partie imaginaire. Nous pouvons constater d'emblée que, a étant strictement positif et b non nul :

$$|(2.a + K_d).b| > |K_d.b| = |f_1(b)| > |f_2(b)|$$

$$K.G.\sin(b.T_{r_c}) = f_2(b) > e^{-a}.K.G.\sin(b.T_{r_c})$$

Donc :

$$K.G.e^{-a}.\sin(b.T_{r_c}) < f_2(b) < f_1(b) < |(2.a + K_d).b| \qquad \Rightarrow \qquad \Im(3.26) > 0$$

Il s'ensuit que la condition (3.24) est forcément respectée au niveau de la partie imaginaire, ce qui ne laisse qu'une solution possible pour la partie réelle : $p_0 = a_0 + i.b_0$ avec $b_0 = 0$ et ceci quel que soit $a_0 \in \mathbb{R}^+$. Or :

$$\Re(T(p_0)) = a_0^2 + K_d.a_0 + K_p + K.G.e^{-a} > 0 \tag{3.27}$$

Ainsi, lorsque la condition suffisante (3.24) (représentée en figure 3.16) est respectée, il n'existe pas de pôle à partie réelle positive ou nulle. Le système est donc stable. Il faut toutefois noter que cette condition n'est pas nécessaire ; il existe donc des cas où cette inégalité n'est pas respectée et où le système est toutefois stable. Nous pouvons d'autre part remarquer que cette condition est indépendante de K_p.

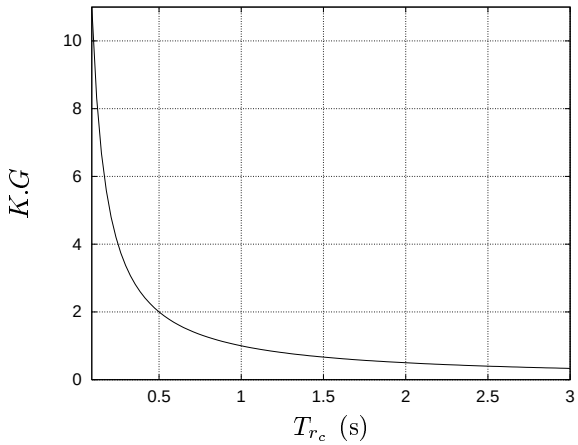


Figure 3.16 – *Evolution de $K.G$ en fonction du retard T_{r_c}*

Simulations

A l'instar des études précédentes au premier ordre, nous avons vérifié le bien fondé de l'inégalité (3.24) sous Matlab/Simulink. Nous avons utilisé un système monodimensionnel (représenté en figure 3.17) dans lequel nous avons fait varier le retard T_{r_c}. Nous avons noté, pour chaque valeur $T_{r_{ci}}$ du retard, le gain $K_i.G$ mettant le système à la limite de la stabilité en maintenant K_d et K_p constants. Nous avons également fait varier le retard en maintenant le gain $K.G$ constant et en faisant varier K_d puis K_p.

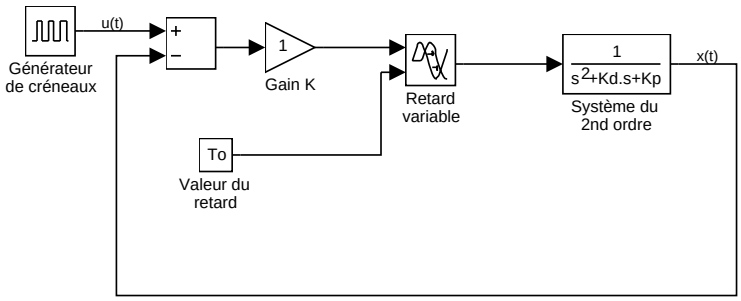


Figure 3.17 – *Schéma de simulation du second ordre (retard constant)*

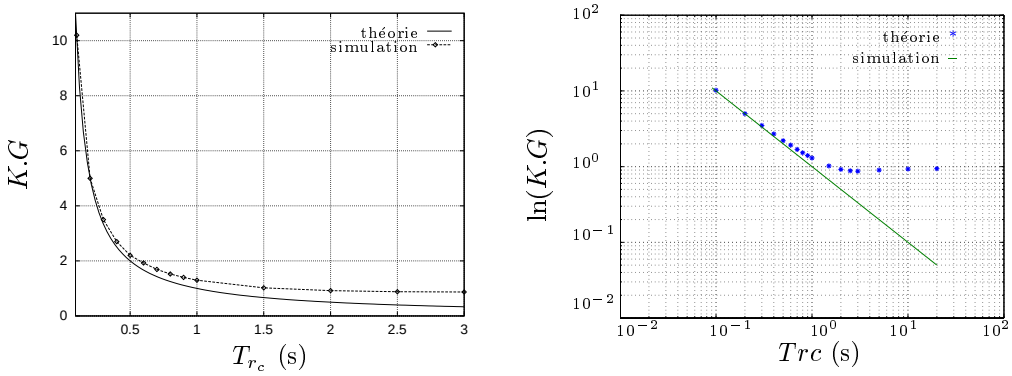


Figure 3.18 – *Variation du gain limite $K.G$ en fonction du retard T_{r_c}*

La figure 3.18 présente les conditions limites de stabilité théorique et expérimentale par rapport à $K.G$ lorsque $K_d = K_p = 1$. Nous avons également représenté ces deux courbes dans des échelles logarithmiques afin de mieux nous rendre compte de la différence entre les résultats théoriques et de simulations. A retard T_{r_c} donné, le système est stable pour toute valeur de $K.G$ inférieure à celle de la courbe expérimentale. Nous pouvons en conclure que la condition que nous avons trouvée est assez proche de la condition réelle de stabilité, surtout pour de faibles retards. Par contre, lorsque le retard T_{r_c} évolue, notre condition devient assez défavorable.

La figure 3.19 représente les conditions limites de stabilité par rapport à K_d lorsque $K.G = 1$ et $K_p = 1$. Le système est stable pour toute valeur de K_d supérieure aux valeurs limites représentées. Nous pouvons également constater que notre condition limite est dans la zone de stabilité. Comme pour la figure (3.18), la condition est assez

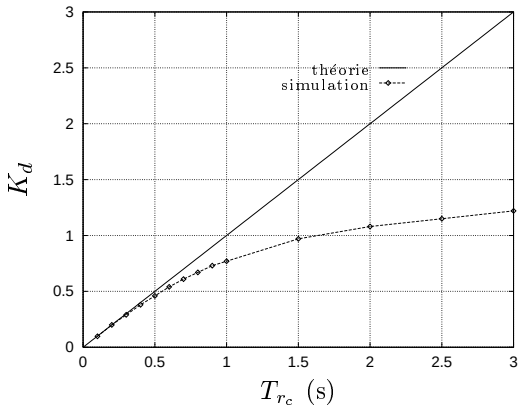


Figure 3.19 – $K_d = f(T_{r_c})$, $K_p = 1$, $K.G = 1$

proche de la réalité pour des retards faibles mais elle est de plus en plus pessimiste au fur et à mesure que le retard T_{r_c} augmente.

Après de nombreux essais, nous avons pu constater que les variations de K_p n'interviennent pas au niveau de la stabilité du système tant que la condition (3.24) est respectée.

Conclusions sur cette condition suffisante

Parce que l'équation caractéristique s'y prête bien, nous avons réussi à obtenir une condition de stabilité valable pour des retards inférieurs à une demi-seconde.

$$K.G \leqslant \frac{K_d}{T_{r_c}}$$

Au-delà d'une seconde, cette condition devient pessimiste par rapport à la réalité vis-à-vis de K_d.

Condition nécessaire et suffisante

Nous avons tenté d'obtenir une condition de stabilité sous forme littérale plus proche de la réalité.

Nous sommes donc repartis du système d'équation (3.22) avec $p = j.\omega$ afin de trouver quels sont les pôles p_i présents sur l'axe imaginaire.

$$\Re(3.26) \quad \Leftrightarrow \quad \frac{\omega^2 - K_p}{K.G} = \cos(\omega.T_{r_c}) \tag{3.28}$$

$$\Im(3.26) \quad \Leftrightarrow \quad \frac{K_d.\omega}{K.G} = \sin(\omega.T_{r_c}) \tag{3.29}$$

$$(3.28) \quad \Leftrightarrow \quad \omega^2 = K.G.\cos(\omega.T_{r_c}) + K_p \tag{3.30}$$

$$(3.29) \quad \Leftrightarrow \quad \frac{K_d^2.\omega^2}{K^2.G^2} = \sin^2(\omega.T_{r_c}) = 1 - \cos^2(\omega.T_{r_c}) \tag{3.31}$$

En remplaçant ω^2 dans l'équation (3.31) à l'aide de l'égalité (3.30), nous obtenons l'équation (3.32) qui est un polynôme du second degré en $\cos(\omega.T_{r_c})$ (cf. équation (3.33))

$$\frac{K_d^2}{K^2.G^2}\left[K.G.\cos(\omega.T_{r_c}) + K_p\right] = 1 - \cos^2(\omega.T_{r_c}) \tag{3.32}$$

$$(3.32) \Leftrightarrow \cos^2(\omega.T_{r_c}) + \left(\frac{K_d^2}{K.G}\right)\cos(\omega.T_{r_c}) + \left(\frac{K_p.K_d^2}{K^2.G^2} - 1\right) = 0 \tag{3.33}$$

Posons $a = 1$, $b = \frac{K_d^2}{K.G}$, $c = \left(\frac{K_p.K_d^2}{K^2.G^2} - 1\right)$ et $x = \cos(\omega.T_{r_c})$:

$$(3.33) \Leftrightarrow a.x^2 + b.x + c = 0 \tag{3.34}$$

Le déterminant Δ (développé à l'équation (3.35)) permet de déterminer l'existence ou non de solutions à l'équation (3.33). Si $\Delta < 0$, les racines de l'équation sont complexes. Or $\cos(\omega.T_{r_c})$ ne peut être complexe car $(\omega, T_{r_c}) \in (\mathbb{R} \times \mathbb{R}^+)$, ce qui revient à dire qu'il n'y a pas de pôle sur l'axe imaginaire.

Nous obtenons ainsi une inégalité portant sur K et dépendant uniquement des paramètres K_d, K_p et G mais pas du retard T_{r_c}. Cette inégalité n'est cependant pas suffisante pour donner l'information de stabilité. Si nous la respectons, nous savons simplement que nous ne serons pas en limite de stabilité. La figure 3.20 représente l'évolution de la fonction $\Delta(K.G)$ en fonction de $K.G$ pour $K_d = K_p = 1$.

Etudions le cas où $\Delta \geqslant 0$.

$$\Delta = b^2 - 4.a.c = \frac{K_d^4}{K^2.G^2} - 4\left(\frac{K_p.K_d^2}{K^2.G^2} - 1\right) \tag{3.35}$$

Les solutions de l'équation (3.34) sont alors x_i avec $i = \{1; 2\}$ tels que ($\lambda_1 = -1$ et $\lambda_2 = +1$) :

$$x_i = \frac{-b - \lambda_i.\sqrt{\Delta}}{2a} \tag{3.36}$$

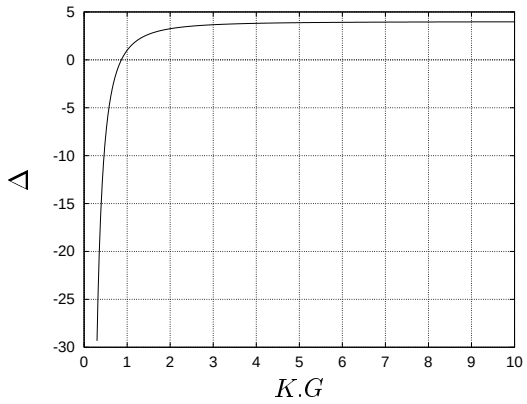


Figure 3.20 – *Evolution de* $\Delta(K.G)$

Les solutions x_1 et x_2 risquent d'être (l'une et/ou l'autre) de norme strictement supérieure à 1. Or, comme $x = \cos(\omega.T_{r_c})$, les solutions ont forcément une norme inférieure ou égale à 1 et :

$$\theta_i = (\omega.T_{r_c})_i = \arccos\left(\frac{-b - \lambda_i\sqrt{\Delta}}{2}\right) \tag{3.37}$$

En repartant de l'équation (3.29), il est alors possible d'éliminer ω et de trouver une relation entre $K.G$ et T_{r_c}.

$$\begin{aligned} (3.29) \quad &\Leftrightarrow \quad \frac{K_d.\omega.T_{r_c}}{K.G.T_{r_c}} = \sin(\omega.T_{r_c}) \\ &\Leftrightarrow \quad K.G.T_{r_c} = \frac{K_d.(\omega.T_{r_c})}{\sin(\omega.T_{r_c})} = \frac{K_d.(\omega.T_{r_c})}{\sqrt{1-\cos^2(\omega.T_{r_c})}} \end{aligned} \tag{3.38}$$

Ainsi, nous obtenons une relation reliant T_{r_c} à $K.G$ pour les deux solutions θ_i avec $i = \{1; 2\}$:

$$
\begin{aligned}
T_{r_c i}(K_i) &= \frac{K_d.\theta_i}{K_i.G.\sqrt{1-cos^2(\theta_i)}} \\
&= \frac{K_d.\arccos\left(\frac{-b-\lambda_i\sqrt{\Delta}}{2}\right)}{K_i.G.\sqrt{1-\left(\frac{-b-\sqrt{\Delta}}{2}\right)^2}} \\
&= \frac{K_d.\arccos\left(\frac{-\frac{K_d^2}{K_i.G}-\lambda_i\sqrt{\frac{K_d^4}{K_i^2.G^2}-4\left(\frac{K_p.K_d^2}{K_i^2.G^2}-1\right)}}{2}\right)}{K_i.G.\sqrt{1-\left(\frac{-\frac{K_d^2}{K_i.G}-\lambda_i.\sqrt{\frac{K_d^4}{K_i^2.G^2}-4\left(\frac{K_p.K_d^2}{K_i^2.G^2}-1\right)}}{2}\right)^2}} \qquad (3.39)
\end{aligned}
$$

(3.40)

Cette relation correspond au cas limite où au moins un pôle se trouve sur l'axe imaginaire, donc à la limite de stabilité. Nous n'avons pas d'inégalité.

Nous avons inversé numériquement cette relation ; pour chaque valeur $G.K_i$, une seule des deux solutions $T_{r_c i}$ est exploitable : celle qui est refusée est soit complexe, soit négative. Le résultat est présenté en figure 3.21. La courbe « théorique » correspond à l'union des solutions T_{r_c} acceptables. Comme $K_d = K_p = 1$, la condition $\Delta \geqslant 0$ est équivalente à $G \geqslant \sqrt{3/4}$. Sur cette figure, nous avons superposé les résultats expérimentaux et les résultats théoriques.

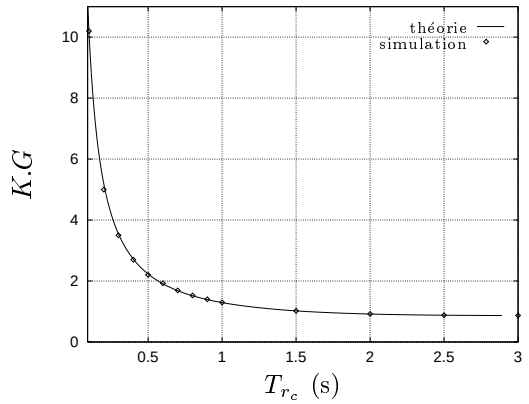


Figure 3.21 – *Evolution de* $K.G = f(T_{r_c})$

Nous avons ainsi obtenu les conditions limites de stabilité mais nous n'avons pas démontré théoriquement s'il faut que $K.G$ soit supérieur ou inférieur à la limite déterminée pour que le système soit stable. Heureusement, la condition suffisante précédente ainsi que les résultats de simulation nous ont permis de lever l'ambiguïté.

3.2.8 Conclusions sur la stabilité

L'ensemble de cette étude permet de déterminer, pour un système du premier ordre (dont l'équation caractéristique est équivalente à (3.6)) et du second ordre (dont l'équation caractéristique est équivalente à (3.21)) et pour un retard global constant, les gains limites du correcteur proportionnel permettant d'assurer la stabilité d'un système.

Le cas du système du premier ordre a été étudié en présence d'un retard variable, continu et borné. Lors de cette étude, nous avons pu distinguer deux cas intéressants :

- $K.G < 1$: le système sera stable quelle que soit sa borne supérieure,
- $K.G < K_{max}.G$: le système sera stable tant que le gain ou le retard ne dépasse pas respectivement K_{max} et $T_{r_{max}}$.

Nous avons également pu constater sur le système du premier ordre (dont la fonction de transfert en boucle ouverte est équivalente à (3.1)) associé à un correcteur de type gain que, s'il est relativement facile de garder un tel système stable, il est nettement plus difficile de régler les problèmes de dépassements et d'oscillations en mode transitoire. En effet, celles-ci dépendent de la vitesse de variation du retard. Plus ce dernier varie lentement (à amplitude constante), plus le système s'éloigne des consignes.

Ce type de correction proportionnelle est en général insuffisant pour obtenir une réponse rapide, sans erreur statique ni dépassement exagéré, d'autant plus que le gain est limité en valeur supérieure pour cause de stabilité. Un seul paramètre ne permet pas de doser tous ces comportements. C'est pourquoi l'automaticien fait généralement appel au minimum à un correcteur plus performant, un proportionnel intégral ou dérivé.

L'étude du système du deuxième ordre est également applicable pour un processus associé à un correcteur tous deux du premier ordre, tant que nous retrouvons la même équation caractéristique (3.21) et que le retard global est constant (cf. figure 3.22). Ce qui permet de corriger un système du premier ordre de manière un peu plus fine, en s'assurant de la stabilité du système ainsi obtenu.

La fonction de transfert du second ordre donnée dans l'équation (3.19) peut correspondre au comportement d'un moteur électrique classique. Ainsi, cette étude va nous permettre d'appliquer la condition de stabilité déterminée précédemment dans de nombreux cas de robots téléopérés.

La généralisation de l'étude de stabilité en présence de retards pseudo-linéaires pour un système de dimension quelconque peut, en fait, s'appliquer à un système d'ordre n. En effet, il suffit de faire appel à une représentation sous forme d'état. Si ce type de retard est purement théorique, il nous a mené sur une piste à explorer ultérieurement : il serait en effet intéressant d'étudier la condition de stabilité d'un système de dimension n (équivalent à un système d'ordre $m < n$ et de dimension $n - m$) en décomposant le retard variable en séries de FOURRIER.

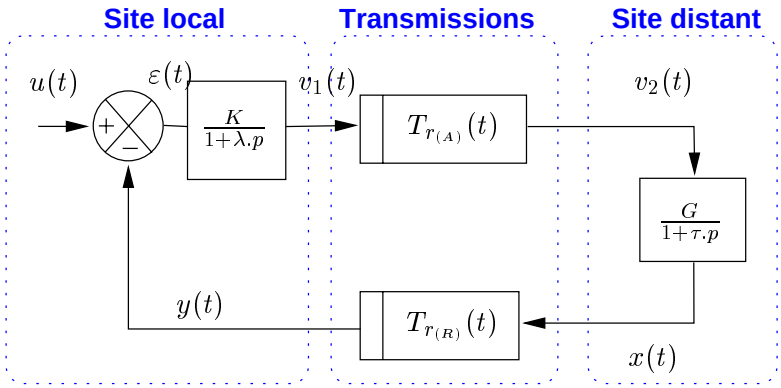


Figure 3.22 – *Système du premier ordre corrigé avec un correcteur du même ordre*

3.3 Régulation des retards

3.3.1 Finalité

Les variations des retards de transmission au cours des dialogues entre la *base* et le *système distant* provoquent deux principaux problèmes :

- une augmentation de l'instabilité du système du fait de la variation aléatoire des retards en cas de système global bouclé avec un $K.G \geqslant 1$,
- une perte de périodicité des signaux de commande $c_r(t)$ et de retour d'information $i_r(t)$ pouvant nuire, d'une part à la stabilité globale du système et d'autre part, à la qualité de prédiction des modules de prédiction distante développés ultérieurement en §3.4 (déformation temporelle des signaux).

Pour ces raisons, il est préférable d'obtenir un retard constant. Ainsi la palette de gains disponibles est un peu plus grande et la périodicité des signaux est conservée.

3.3.2 Mesure des temps de transmission aller–retour

Chaque trame émise par la *base* et le *système distant* est datée. Chaque trame reçue par les deux applications est acquittée ; l'accusé de réception contient la date d'émission de la trame acquittée dès réception. Ainsi à la réception des accusés, il est possible de calculer le temps de vol total réseau $T_{rés.(A+R)}(t)$: aller $T_{rés.(A)}(t)$ + retour $T_{rés.(R)}(t)$ en comparant la date d'arrivée de l'accusé et la date d'émission de la trame acquittée. Il représente typiquement le temps total écoulé sur le réseau.

Cependant, faute de synchronisation temporelle des horloges des deux applications exécutées sur deux postes distincts indépendants l'un de l'autre, il ne nous est pas possible de mesurer un temps de transmission aller $T_{rés.(A)}(t)$ ou retour $T_{rés.(R)}(t)$ individuellement. Nous avons donc dû émettre l'hypothèse que le délai aller est égal au délai retour :

$$T_{rés._{(A)}}(t) = T_{rés._{(R)}}(t) = \frac{T_{rés._{(A+R)}}(t)}{2} \tag{3.41}$$

3.3.3 Principe

Nature du problème

Les trames de données sont émises périodiquement par la *base* et le *système distant*. Le passage par le médium de transmission leur inflige des retards variables rendant apériodique la réception de ces trames. Lorsque ces données doivent être resynchronisées les unes par rapport aux autres, nous devons alors faire appel à une sorte de « régulateur temporel ».

Nature des retards dus au réseau

Les retards infligés par le réseau $T_{rés._{(A|R)}}(t)$ peuvent être décomposés en deux parties :

- une partie constante du retard due aux divers temps de propagation et temps de traversées minimaux des éléments de commutation du réseau. Cette partie est notée $T_{rés.\ fixe_{(A|R)}}$,
- une partie variable du retard due au partage des ressources du réseau, notée $T_{rés.\ var._{(A|R)}}(t)$,

$$T_{rés._{(A|R)}}(t) = T_{rés.\ fixe_{(A|R)}} + T_{rés.\ var._{(A|R)}}(t) \tag{3.42}$$

Principe de régulation adopté

Les trames réceptionnées par la *base* et le *système distant* arrivant de manière apériodique, le principe de la régulation consiste à les intercepter, à les conserver le temps nécessaire pour pouvoir les libérer périodiquement (cf. figure 3.23). Ainsi les modules en aval des régulateurs reçoivent des trames périodiques (de même période qu'à leur émission initiale).

Fonctionnement détaillé

Le régulateur est constitué d'une file d'attente *FIFO (First-In, First-Out)* : les données ressortent de la file dans l'ordre dans lequel elles sont entrées (figure 3.24).

Cette file d'attente est préalablement remplie avec un certain nombre de trames (régime transitoire) $n_{désiré}$. Lorsque le nombre désiré d'éléments dans la file $n_{désiré}$ est

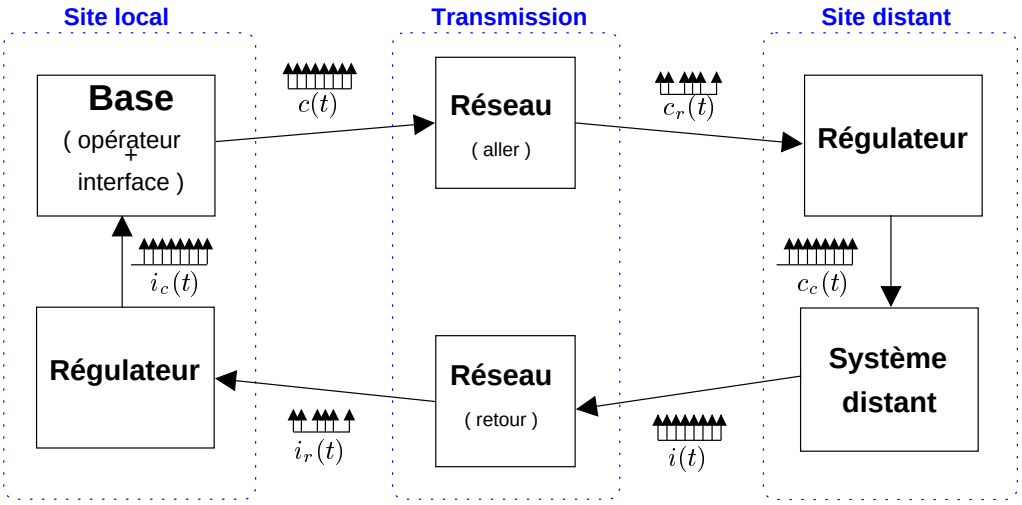


Figure 3.23 – *Insertion des régulateurs*

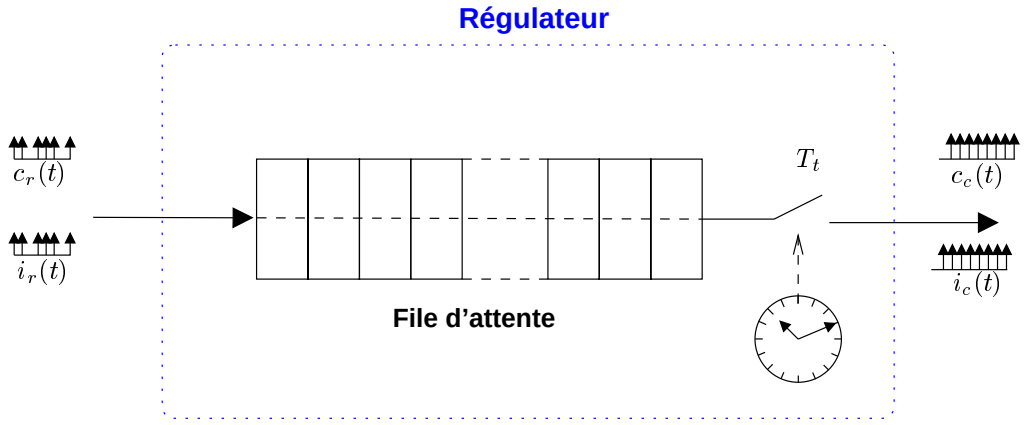


Figure 3.24 – *Structure des régulateurs*

atteint, le vidage périodique commence (régime permanent). Le nombre d'éléments dans la file permet de compenser les variations de retard ; si une trame arrive plus tôt que prévu, la file d'attente grandit. Si, par contre, une trame tarde à arriver, la file se vide à chaque période de transmission jusqu'à la réception de la trame en retard. Si la file se vide entièrement, la périodicité des trames en sortie du régulateur est compromise et un signal d'alarme est émis. Cela signifie que le remplissage initial $n_{désiré}$ était insuffisant. Soit il a été sous-estimé, soit la charge du réseau a augmenté et l'écart-type des délais de transmission des trames a augmenté. S'il s'agit simplement d'un retard passager, la file oscille autour de sa taille désirée $n_{désiré}$.

Stratégies en cas de vidage complet

Les diverses stratégies évoquées ci-dessous dépendent du type de commande locale implémentée au niveau du *système distant* ainsi que de la tâche à effectuer.

1. **Répéter la toute dernière consigne reçue**. Si les consignes sont uniquement les positions désirées d'un effecteur, cette solution peut être adoptée ; il en résultera un arrêt de l'effecteur le temps que la situation revienne à la normale. Si, parmi les consignes, il y a des vitesses désirées, les trajectoires générées risquent d'être déformées. S'il y a un asservissement global sur la trajectoire, un vidage complet de la file d'un des deux régulateurs peut être considéré comme une perturbation. Par conséquent cette stratégie peut être également adoptée. Il faut cependant remarquer que si l'effecteur est lancé à vive allure et qu'il ne reçoit pas de consigne d'arrêt à temps, les conséquences peuvent en être fâcheuses.

2. **Extrapoler les dernières consignes**. Si les consignes correspondent à des positions désirées, cette stratégie ne peut être employée que s'il est acceptable d'obtenir de légères modifications de la trajectoire qui rejoint nécessairement le trajet désiré dès le retour à la normale.

3. **Mettre le système en pause** Cette solution peut s'adapter à beaucoup de cas, cependant elle nécessite de préprogrammer une séquence de mise en pause au niveau du *système distant*. S'il s'agit d'un véhicule sous-marin, nous pouvons imaginer un mode autonome dans lequel le submersible arrête sa course et maintient une position immobile.

Il est également possible d'envisager la première ou la deuxième stratégie pendant quelques échantillons puis de passer en pause (stratégie numéro 3) ensuite.

Dimensionnement

Pour calculer la taille désirée de la file $n_{désiré}$, une phase préliminaire d'audit du réseau est effectuée ; la *base* et le *système distant* envoient chacun une série de trames de test permettant de calculer différents temps de trajet aller + retour $T_{rés._{(A+R)_i}}$.

En première approche, nous avons adopté le calcul suivant afin de déterminer $n_{désiré}$. Nous utilisons l'amplitude maximale $\Delta T_{rés._{(A+R)}}$ des $T_{rés._{(A+R)_i}}$ observés pendant la période d'audit du réseau, la période de transmission T_t ainsi qu'un coefficient de sécurité $1,5 \leqslant \eta \leqslant 2$. Le résultat est donné dans l'équation (3.43) en tenant compte de l'équation (3.41).

$$n_{désiré} = \mathrm{E}\left(\eta \times \frac{\Delta T_{rés._{(A|R)}}}{T_t}\right) \tag{3.43}$$

En supposant $\eta = 1$, ce calcul dimensionne la file de façon qu'elle se vide complètement le temps d'attendre la trame suivante lorsque deux trames sont séparées par un laps de temps égal à l'amplitude maximale des variations des délais de transmission mesurée lors de la phase d'audit. Ainsi, un coefficient de sécurité $\eta > 1$ permet d'éviter de vider complètement la file. En présence d'un laps de temps aussi élevé, elle devrait au pire, ne contenir que n_{min} éléments.

$$n_{min} = \mathrm{E}\left((1-\eta) \times n_{désiré}\right) \tag{3.44}$$

Une fois le mode permanent atteint, la téléopération peut débuter. Cependant si les retards dus au réseau viennent à vider entièrement la file, la périodicité des données sera perdue et il faudra donc réinitialiser l'ensemble communication (avec une file mieux dimensionnée) + *système distant* (remis dans une configuration connue). Le temps maximum passé dans la file d'attente est noté $T_{rég.\ max_{(A|R)}}$ et est calculé selon l'équation (3.45)

$$T_{rég.\ max_{(A|R)}} = n_{désiré_{(A|R)}} \times T_t \tag{3.45}$$

Différents temps de vol

En supposant que le régulateur fonctionne parfaitement en mode permanent, le temps total de vol aller ou retour d'un message $T_{total_{(A|R)}}(t)$, temps écoulé entre son émission et sa sortie du régulateur après réception, devient alors constant puisque chaque trame est émise et régulée à la même période de transmission T_t. Le temps de vol total aller ou retour $T_{total_{(A|R)}}(t)$ peut alors se décomposer ainsi :

- partie constante du retard $T_{rés.\ fixe_{(A|R)}} = min(T_{rés._{(A|R)}}(t))$,
- partie variable du retard $T_{rés.\ var._{(A|R)}}(t) = T_{rés._{(A|R)}}(t) - T_{rés.\ fixe_{(A|R)}}$,
- temps de passage dans la file $T_{rég._{(A|R)}}(t)$, variable et compensant les fluctuations de $T_{rés.\ var._{(A|R)}}(t)$.

Soit :

$$T_{total_{(A|R)}}(t) = T_{rés.\ fixe_{(A|R)}} + T_{rés.\ var._{(A|R)}}(t) + T_{rég._{(A|R)}}(t) = \text{Constante} \tag{3.46}$$

Et, en tenant compte de (3.45) ($n_{(A|R)}(t)$ est le nombre d'échantillons dans une des files à tout instant) :

$$T_{rés.\ var._{(A|R)}}(t) + T_{rég._{(A|R)}}(t) = T_{rég.\ max_{(A|R)}} = max(n_{(A|R)}(t)) \times T_t \tag{3.47}$$

La figure 3.25 illustre les différentes facettes des temps de vol aller ou retour $T_{total_{(A|R)}}(t)$ évoquées ici.

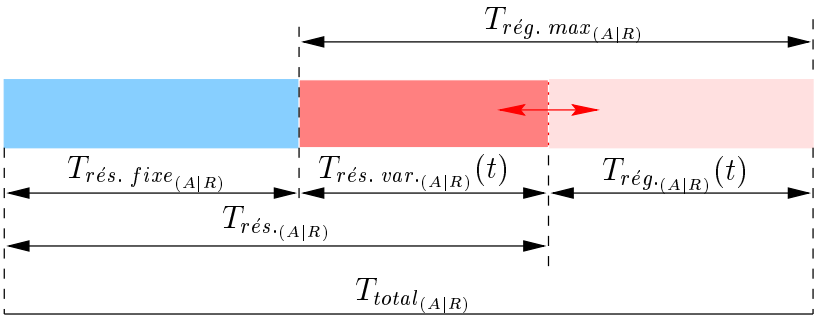


Figure 3.25 – *Décomposition du temps de vol total aller ou retour* $T_{total_{(A|R)}}(t)$

La *base* et le *système distant* déterminent et se transmettent mutuellement leurs $n_{désiré}$ respectifs. Ils peuvent donc calculer chacun $T_{rég.\ max_{(A)}}$ et $T_{rég.\ max_{(R)}}$. D'autre part, une mesure régulière de $T_{total_{(A+R)}}(t)$ est effectuée par le biais de messages parcourant la boucle entière de la figure 3.23. Ils peuvent donc déterminer $T_{rés.\ fixe_{(A+R)}}$ car :

$$\begin{aligned} T_{total_{(A+R)}}(t) &= T_{total_{(A)}}(t) + T_{total_{(R)}}(t) = \text{Cste} && (3.48)\\ &= (T_{rés.\ fixe_{(A)}} + T_{rég.\ max_{(A)}}) + (T_{rés.\ fixe_{(R)}} + T_{rég.\ max_{(R)}}) \\ &= T_{rés.\ fixe_{(A+R)}} + (T_{rég.\ max_{(A)}} + T_{rég.\ max_{(R)}}) && (3.49) \end{aligned}$$

Donc :

$$T_{rés.\ fixe_{(A+R)}} = T_{total_{(A+R)}}(t) - (T_{rég.\ max_{(A)}} + T_{rég.\ max_{(R)}}) \tag{3.50}$$

Cependant, comme précédemment, il n'est pas possible d'en déduire $T_{rés.\ fixe_{(A)}}$ ou $T_{rés.\ fixe_{(R)}}$ sans émettre d'hypothèse supplémentaire.

Nous pouvons considérer sans faire d'hypothèse abusive que :

$$T_{rés.\ fixe_{(A)}} = T_{rés.\ fixe_{(R)}} = \frac{T_{rés.\ fixe_{(A+R)}}}{2} \tag{3.51}$$

$$= \frac{T_{total_{(A+R)}}(t) - (T_{rég.\ max_{(A)}} + T_{rég.\ max_{(R)}})}{2} \tag{3.52}$$

En effet, étant donnée la définition donnée à $T_{rés.\ fixe_{(A|R)}}$, les messages parcourent le même nombre d'organes de commutation et possèdent le même temps de propagation.

D'autre part, si les retards dus au réseau sont très variables, il y a de fortes chances que le régulateur possède un $n_{désiré}$ élevé et que $T_{rés.\ fixe_{(A|R)}} \ll T_{rég.\ max_{(A|R)}}$. Dans ce cas :

$$T_{rés.\ fixe_{(A|R)}} \ll T_{rég.\ max_{(A|R)}} \qquad \Rightarrow \qquad T_{total_{(A|R)}} \approx T_{rég.\ max_{(A|R)}} \tag{3.53}$$

L'intérêt de ce calcul est qu'il permet de synchroniser la *base* et le *système distant* entre eux. Connaissant le temps de vol de chaque message, il est alors aisé de déterminer la date d'émission de chacun des messages réceptionnés. C'est un avantage indéniable pour les modules de prédiction décrits dans §3.4, page 131.

Perspectives d'améliorations

Il est possible d'envisager une adaptation en ligne de la taille de la file mais cela impose de légères variations de période aux instants de commutation. Il faut alors arrêter toute manœuvre sensible le temps de régénérer la file ; les données en sortie du régulateur ne sont pas valides du point de vue temporel pendant ce régime transitoire.

Il est également possible de moduler la période de transmission en fonction du comportement du réseau. Il serait ainsi possible d'obtenir une régulation plus fine pour des petits retards comme dans le cas de Béziers. Cela suppose, d'une part les mêmes conditions évoquées précédemment pour l'adaptation de la taille de la file et, d'autre part, que les différentes composantes entrant dans l'amélioration de la téléopération (notamment l'estimation/prédiction) soient capables de gérer des signaux échantillonnés à période variant par paliers.

3.3.4 Validation par simulations

Nous avons modélisé le fonctionnement de ce régulateur et nous l'avons adapté au modèle de simulation développé au §2.5, page 60.

Nous avons réalisé de nombreux tests en utilisant deux cas de figure précédemment étudiés : Montpellier–Béziers et Montpellier–Rutgers (le matin). Ces simulations nous ont permis de :

– valider le concept en simulation,

– observer les faiblesses du système,
– déterminer un coefficient de sécurité η.

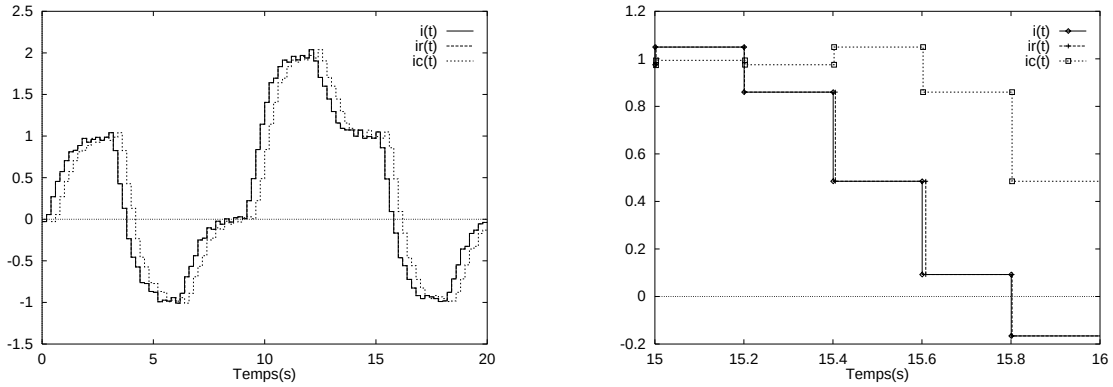
Le tableau 3.1, page 113, met en relation les seuils déterminés empiriquement avec les mesures sur les retards imposés par le bloc réseau. Ces mesures sont l'écart-type et l'amplitude de ces retards. Ces résultats sont normalisés en tenant compte de la période de transmission T_t.

La méthode que nous avons employée pour déterminer si le choix du seuil $n_{désiré}$ est adapté ou non est d'observer la taille de la file d'attente des régulateurs et de surveiller ses valeurs inférieures n_{min}. Ainsi, si le seuil est de 13 et que la file descend parfois jusqu'à 3 échantillons, il semble dangereux de diminuer ce seuil sans craindre de vider totalement la file. Cela suppose évidemment un comportement relativement constant du réseau.

Toutes les données présentées dans cette section correspondent au signal de retour avant passage par le lien de transmission $i(t)$ et après passage $i_r(t)$.

Courte distance : Montpellier–Béziers

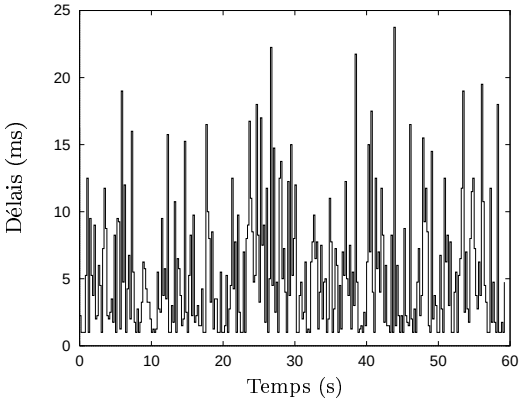
La figure 3.26 représente l'évolution des données au moment de leur émission par le *système distant*, après passage par le réseau et une fois le régulateur de la *base* traversé. L'échelle temporelle de cette figure est telle que la courbe de $i(t)$ et celle de $i_r(t)$ sont quasiment confondues. A première vue, la courbe $i_c(t)$ ressemble à $i(t)$, translatée d'environ 400 ms. Vue la quasi-constance du retard global $T_{total_{(R)}}(t)$ présenté figure 3.28 autour de 402,25 ms, nous pouvons confirmer le bon fonctionnement de ce régulateur.



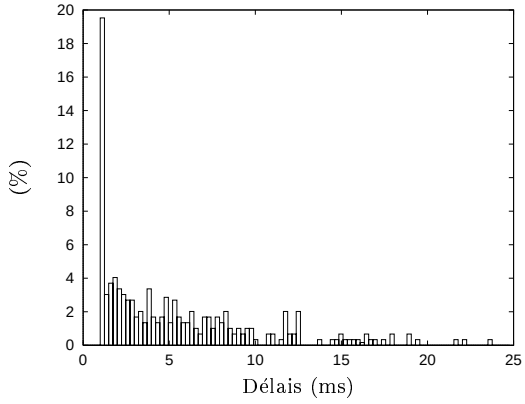
(a) Vue générale (b) Détails

Figure 3.26 – *Simulation : action d'un régulateur sur les signaux (cas Béziers)*

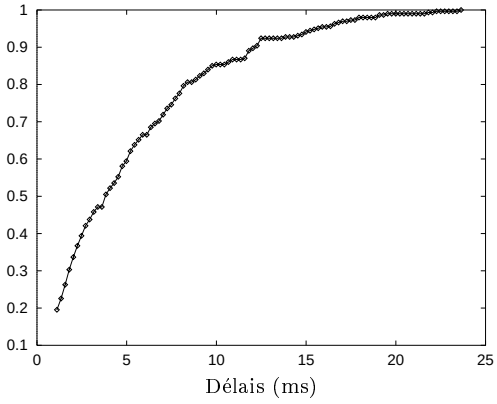
Le cas de Béziers est intéressant dans le sens où la période de transmission T_t est largement supérieure à l'amplitude des retards dus au réseau $T_{rés._{(R)}}$ (rapport de 1 pour



(a) Evolution



(b) Distribution



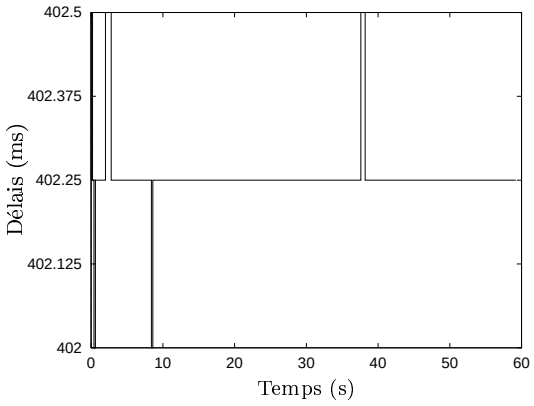
(c) Répartition

Figure 3.27 – *Simulation : délais dus au réseau (cas Béziers)*

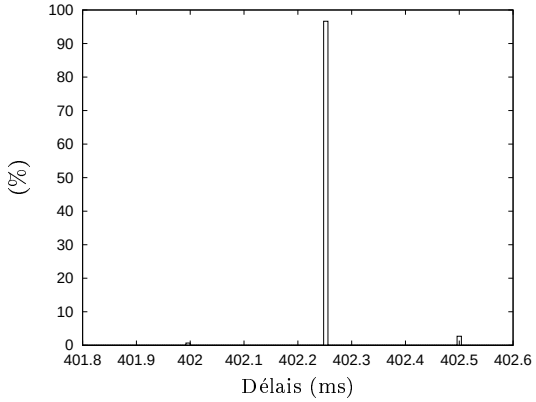
8). Ainsi la régulation ne pose aucun problème (l'occupation de la file d'attente varie peu de 1 à 3 échantillons ($n_{désiré} = 2$), comme le montre la figure 3.30). Cependant le retard global obtenu $T_{total_{(R)}}(t)$ est démesuré : en moyenne, 400 ms (cf figure 3.28) pour un retard maximum sans régulation $T_{rés._{(R)}max}$ de 24 ms (cf figure 3.27). Ceci représente un rapport d'environ 17 et ce, en ayant fourni un seuil $n_{désiré}$ aussi petit que possible. Cet écart est très net sur la figure 3.29.

La qualité des périodes (représentée figure 3.31) en sortie est ici excellente ; seuls quelques écarts de 250 μs (environ 0,1%), de l'ordre du pas de simulation ont été relevés sans que nous puissions les expliquer.

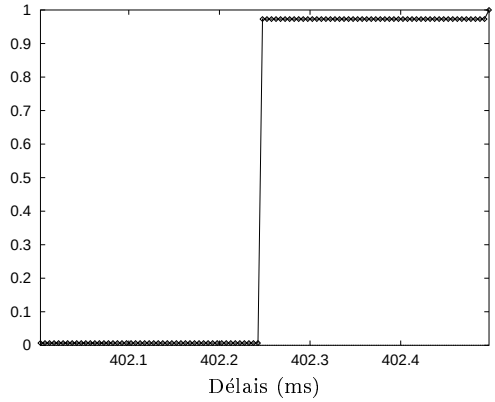
Cette étude permet de qualifier le comportement du régulateur dans des situations auxquelles il n'est pas destiné à faire face a priori. Les résultats présentés ici permettent de préciser qu'un tel régulateur ne convient pas à des faibles variations de retards du fait du choix initial de la période de transmission. En effet, la périodicité des signaux $c_c(t)$ et $i_c(t)$ en sortie des régulateurs est excellente mais à un prix prohibitif : le temps de traversée des régulateurs $T_{rég._{(A|R)}}(t)$ est beaucoup trop élevé en comparaison des retards



(a) Evolution



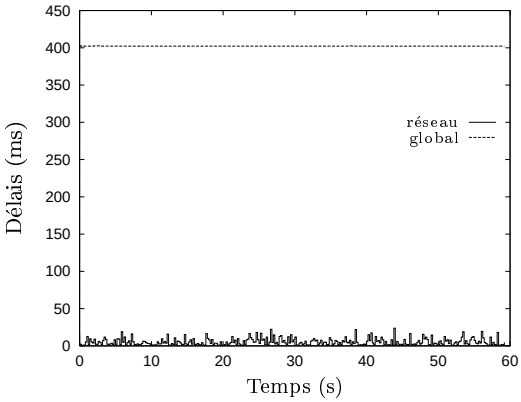
(b) Distribution



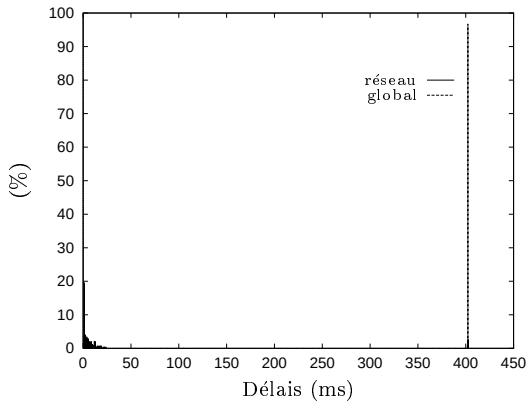
(c) Répartition

Figure 3.28 – *Simulation : retards après régulation (cas Béziers)*

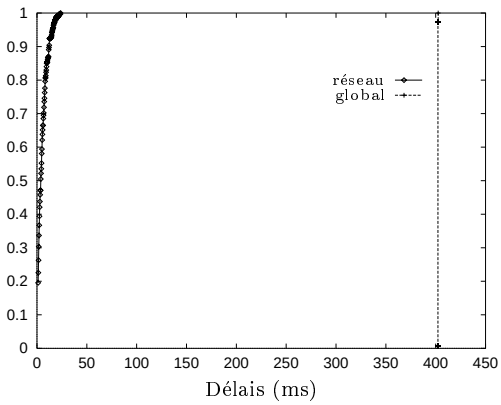
dus au réseau $T_{rés_{\cdot(A|R)}}(t)$. A l'avenir, il faudra envisager la possibilité de sélectionner une période de transmission compatible avec le type de retard observé pendant la période d'audit.



(a) Evolution

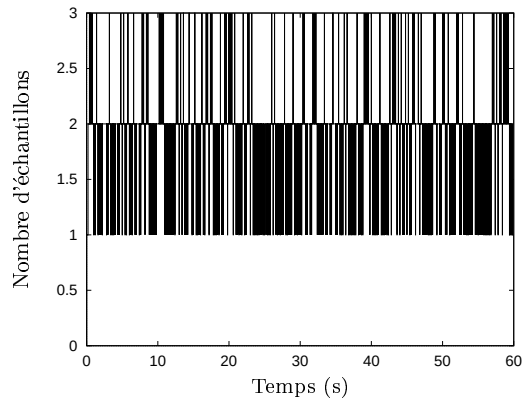


(b) Distribution

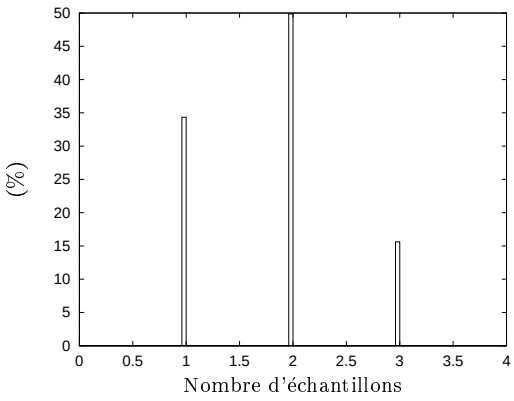


(c) Répartition

Figure 3.29 – *Simulation : comparaison des retards réseau et globaux (cas Béziers)*

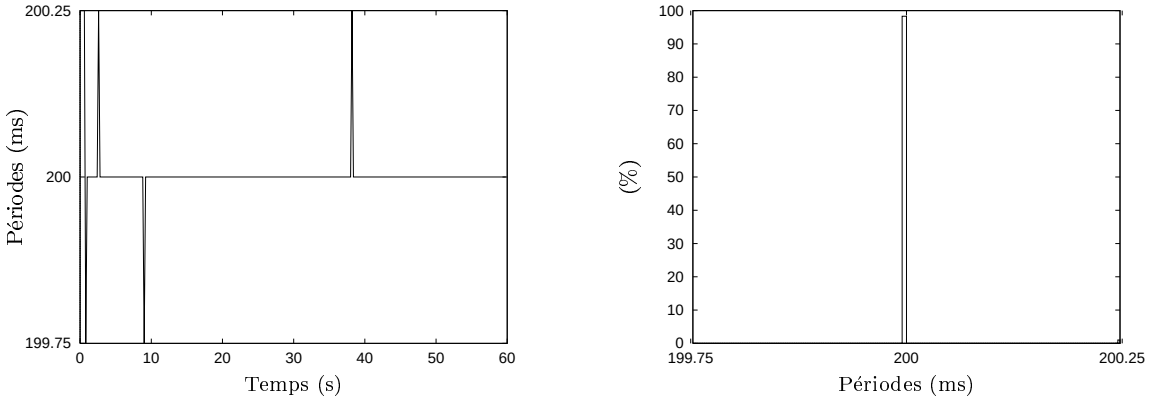


(a) Evolution



(b) Distribution

Figure 3.30 – *Simulation : évolution de la taille d'un régulateur (cas Béziers)*



(a) Evolution (b) Distribution

Figure 3.31 – *Simulation : évolution des périodes en sortie d'un régulateur (cas Béziers)*

Longue distance : Montpellier–Rutgers (le matin)

Ce cas d'étude est beaucoup plus équilibré que le précédent en matière de résultats de régulation. Il représente d'ailleurs un cas de téléopération à longue distance, ce pour quoi le régulateur a été étudié.

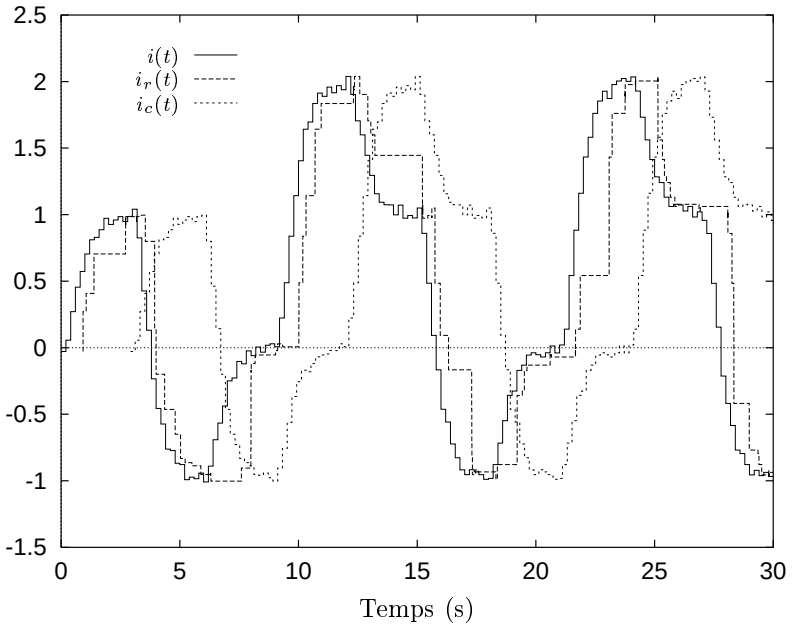
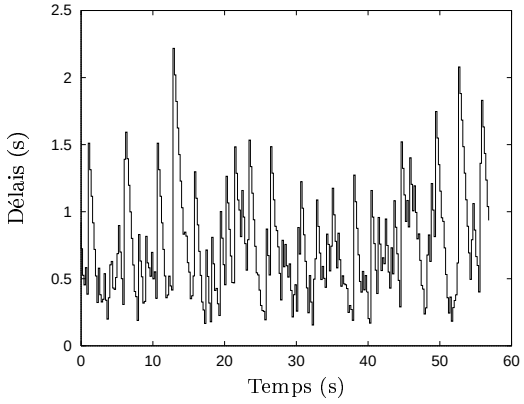
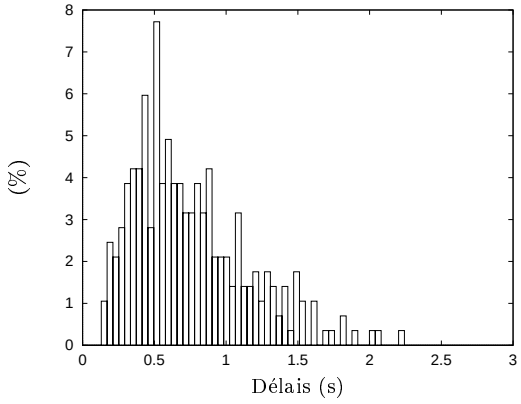


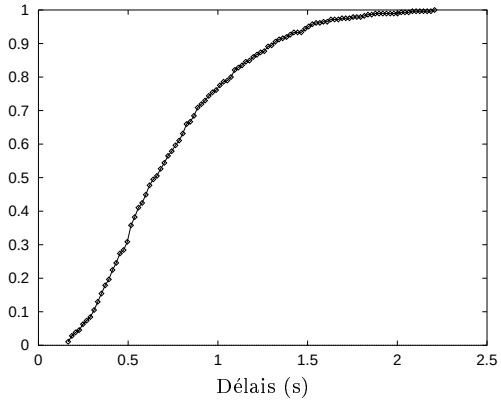
Figure 3.32 – *Simulation : action d'un régulateur sur les signaux (cas Rutgers)*



(a) Evolution



(b) Distribution



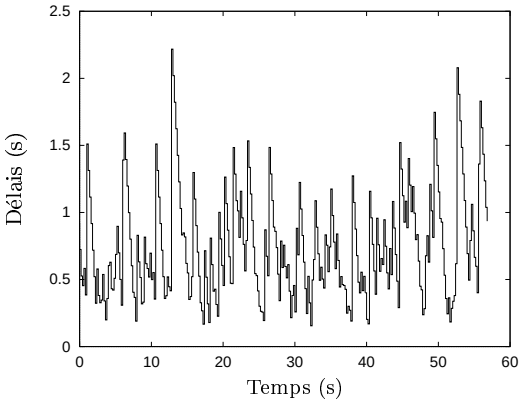
(c) Répartition

Figure 3.33 – *Simulation : retards imposés par le réseau (cas Rutgers)*

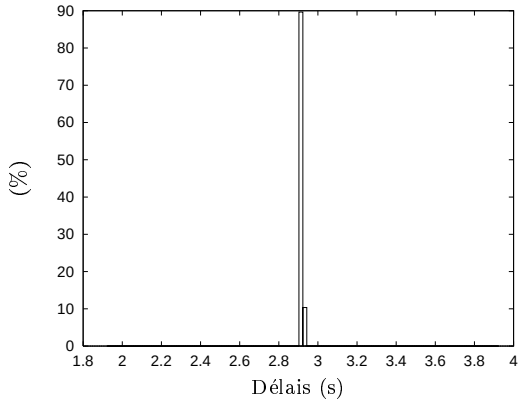
L'utilisation de la file n'est peut-être pas optimale car il y a au minimum 3 échantillons dans la file en régime permanent, mais des essais à $n_{désiré} = 10$ ont provoqué un vidage complet de la file tandis que tous les essais à $n_{désiré} = 11$ que nous avons effectués se sont passés sans incident.

La file est remplie avec en moyenne 10,8 échantillons. Son niveau oscille autour de $n_{désiré}$, ce qui signifie que le comportement du réseau est relativement stable. Le temps moyen de parcours de la file $\overline{T_{rég._{(R)}}}$ est de 2,17 s. Le retard global moyen $\overline{T_{total_{(R)}}}$ de 2,9 s est à comparer avec un retard moyen brut $\overline{T_{rés._{(R)}}}$ de 750 ms, soit un accroissement relatif moyen de près de 300%.

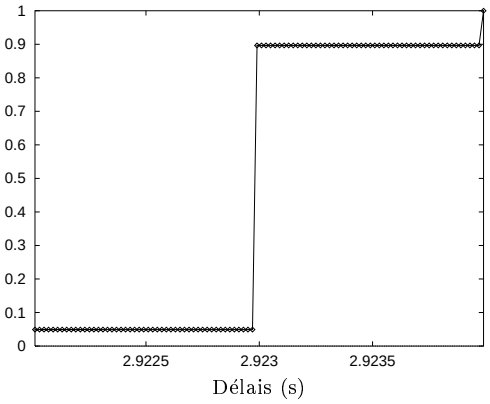
Le rapport entre le retard global moyen $\overline{T_{total_{(R)}}}$ (cf figure 3.34) et le retard brut maximum $T_{rés._{(R)}max}$ (cf figure 3.33) est de 1,03 dans ce cas avec un taux d'occupation de la file très variable : 3 à 14 éléments pour un $n_{désiré} = 11$ (cf. figure 3.36). Nous pouvons constater sur la figure 3.35 que les retards avant et après régulation sont du même ordre de grandeur.



(a) Evolution

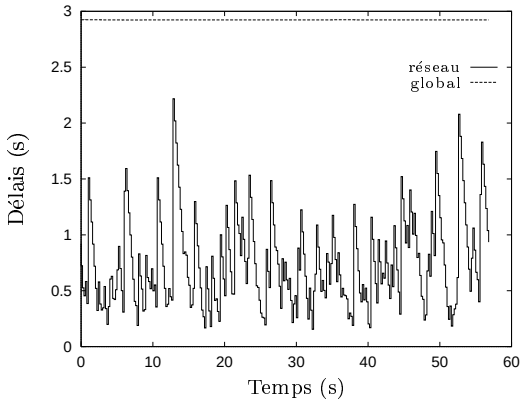


(b) Distribution

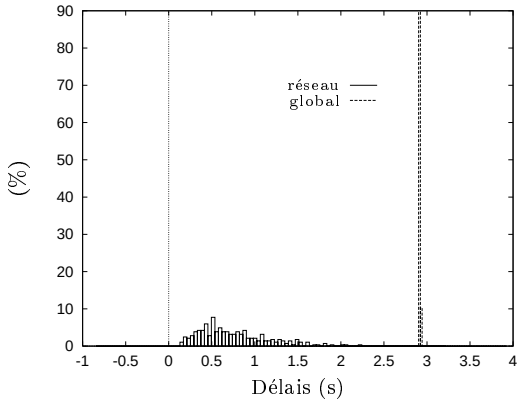


(c) Répartition

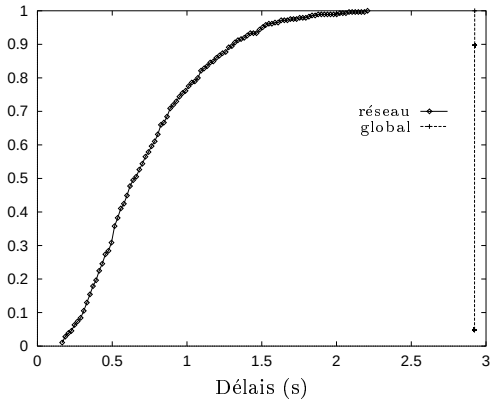
Figure 3.34 – *Simulation : retards après régulation (cas Rutgers)*



(a) Evolution

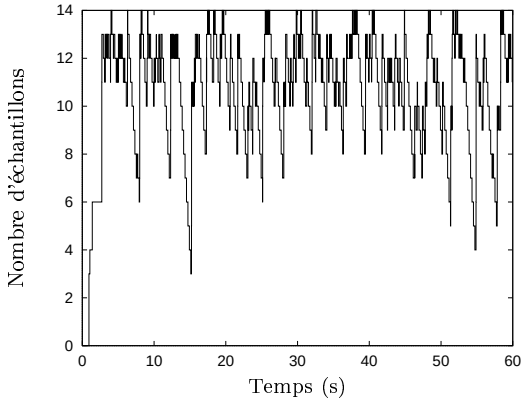


(b) Distribution

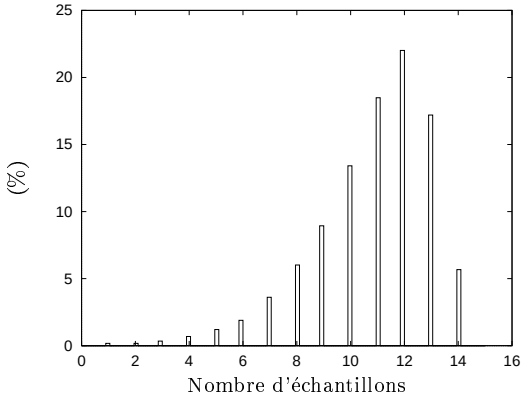


(c) Répartition

Figure 3.35 – *Simulation : comparaison des retards avant et après régulation (cas Rutgers)*



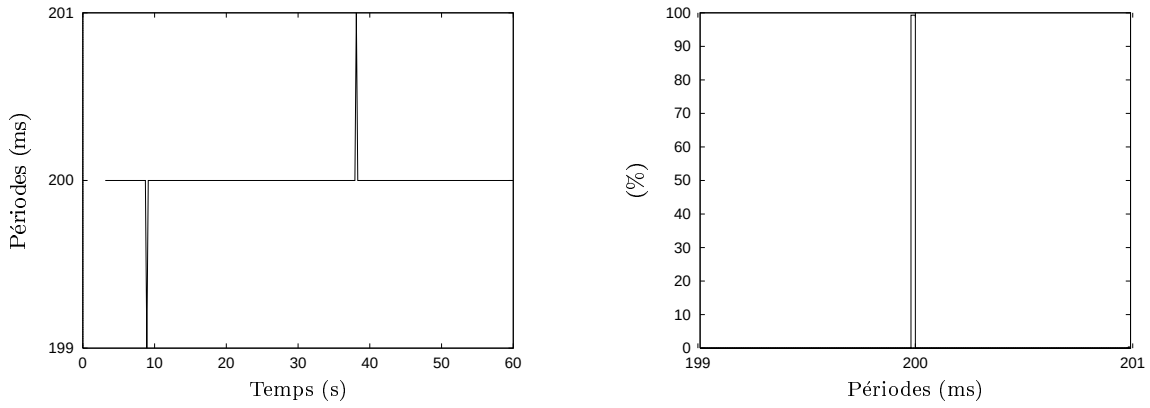
(a) Evolution



(b) Distribution

Figure 3.36 – *Simulation : évolution de la taille d'un régulateur (cas Rutgers)*

Du point de vue de la périodicité du signal en sortie du régulateur, seuls deux petits écarts de 1 ms (soit un écart relatif de 0,5 %) sont à déplorer, ce qui est négligeable. La figure 3.37 donne l'évolution et la distribution de cette périodicité.



(a) Evolution (b) Distribution

Figure 3.37 – *Simulation : évolution des périodes en sortie d'un régulateur (cas Rutgers)*

La figure 3.38 propose un agrandissement du début du fonctionnement du régulateur. Y sont mentionnés le régime transitoire pendant lequel la file se remplit au fur et à mesure de l'arrivée des données ainsi que le régime permanent débutant dès que $n_{désiré}$ est atteint.

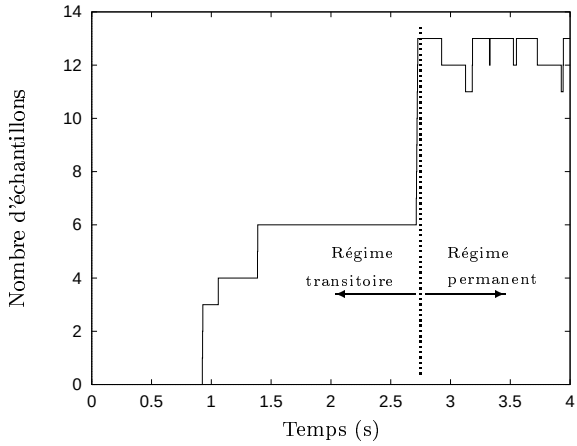


Figure 3.38 – *Simulation : détails du remplissage de la file (cas Rutgers)*

Conclusions à propos de ces résultats

Globalement, la périodicité du signal corrigé n'est pas tout à fait parfaite. Quelques écarts de l'ordre du pas de simulation apparaissent. Nous n'avons pas réussi à déterminer les raisons de ces écarts ponctuels mais leur faible occurrence et leur faible amplitude font qu'ils peuvent être négligés. Sans compter que dans la réalité, nous n'obtiendrons pas une périodicité aussi parfaite. En outre il suffit de baisser le pas de simulation pour les réduire d'autant mais à un coût de temps de simulation supérieur.

Cible	$n_{désiré}$	$\frac{\sigma_{T_{rés.(A+R)}}}{T_t}$	$\frac{\Delta T_{rés.(A+R)}}{T_t}$	n_{min}	$\eta_{déduit}$
Béziers	2	0,02	0,1	1	20
Rutgers (matin)	13	2	10	3	1,3

Tableau 3.1 – *Simulation : résultats des différents essais de détermination de la taille désirée*

Dans le tableau 3.1, la colonne $n_{désiré}$ indique le $n_{désiré}$ attribué empiriquement dans chacun des essais précédents. Les colonnes $\left(\frac{\sigma_{T_{rés.(A+R)}}}{T_t}\right)$ et $\left(\frac{\Delta T_{rés.(A+R)}}{T_t}\right)$ reprennent les résultats obtenus précédemment. n_{min} indique le nombre minimum de données dans la file observé lors des simulations. $\eta_{déduit}$ a été déterminé selon l'équation suivante :

$$\eta_{déduit} = \frac{n_{désiré}}{\mathrm{E}\left(\frac{\Delta T_{rés.(A+R)}}{T_t}\right)}$$

Sachant qu'il n'est pas possible d'attribuer un $n_{désiré}$ inférieur à 2 sous peine de risquer de vider entièrement la file, nous n'avons pas pris en compte $\eta_{déduit}$ dans le cas à courte distance. Nous avons finalement adopté le calcul suivant pour la détermination de $n_{désiré}$:

$$n_{désiré} = 2 + \mathrm{E}\left(\eta \times \frac{\Delta T_{rés.(A+R)}}{T_t}\right) \tag{3.54}$$

Avec $\eta = (n_{désiré} - 2) \cdot \left(\frac{T_t}{\Delta T_{rés.(A+R)}}\right)$ dans le cas de Rutgers.

3.3.5 Validation expérimentale

Nous présentons tout d'abord la manière dont nous avons intégré les régulateurs dans la chaîne de téléopération expérimentale.

Nous commentons ensuite les expérimentations effectuées pour de faibles retards (liaison directe entre BASE et MANIMOB).

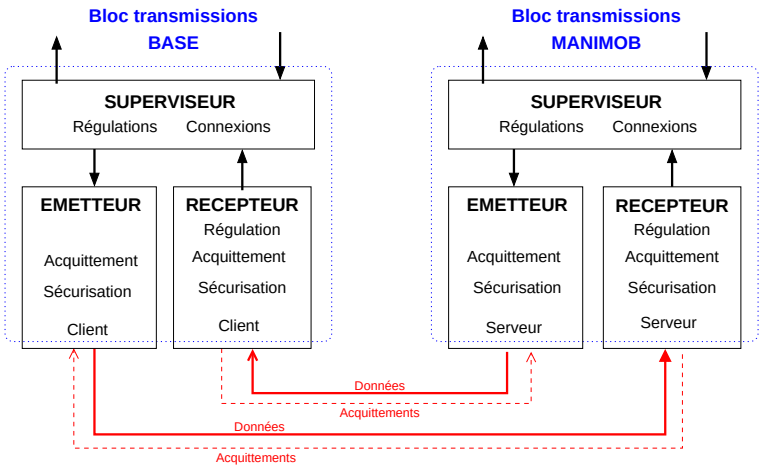


Figure 3.39 – *Séparation des fonctions d'émission et de réception*

Enfin, nous commentons les résultats des tests à longue distance pour le cas Rutgers (matin).

Certaines de ces expérimentations ont été publiées dans [LEL2 99] ; il ne s'agit pas exactement des cas d'étude Montpellier–Béziers et Montpellier–Rutgers, mais nous arrivons aux mêmes conclusions.

Insertion des régulateurs sur la plate-forme expérimentale

Deux régulateurs viennent s'insérer chacun dans les blocs de réception de BASE et de MANIMOB, cf. figure 3.40.

Les deux applications attendent mutuellement la fin de la période d'audit du réseau pour passer en régime permanent. Elles échangent leur $n_{désiré_{(A|R)}}$ respectif afin de pouvoir se synchroniser l'une par rapport à l'autre.

Résultats à courte distance

Dans ce cas de figure, les applications BASE et MANIMOB sont exécutées sur deux ordinateurs différents reliés entre eux via le réseau *Ethernet* local. Etant donné la topologie de ce réseau, seul un commutateur sépare les deux ordinateurs. Ceci correspond au cas de figure illustré par la figure 2.3, page 40.

Les deux applications ont calculé $n_{désiré} = 2$ selon la méthode définie par l'équation (3.54). En moyenne, les retards $T_{r_{A+R}}(t)$ sont plus faibles dans cette étude que dans l'étude en simulation. Les conditions initiales ont fait en sorte que dans le cadre de cet essai, la taille de la file (cf. figure 3.45) a oscillé entre 2 et 3. D'autres tests dans les mêmes conditions ont donné une évolution entre 1 et 2 échantillons.

Les courbes présentées ici n'incluent pas la période d'audit. Nous pouvons remarquer une loi statistique « étrange » pour les retards dus au réseau $T_{r_{A+R}}(t)$. Au vu de la

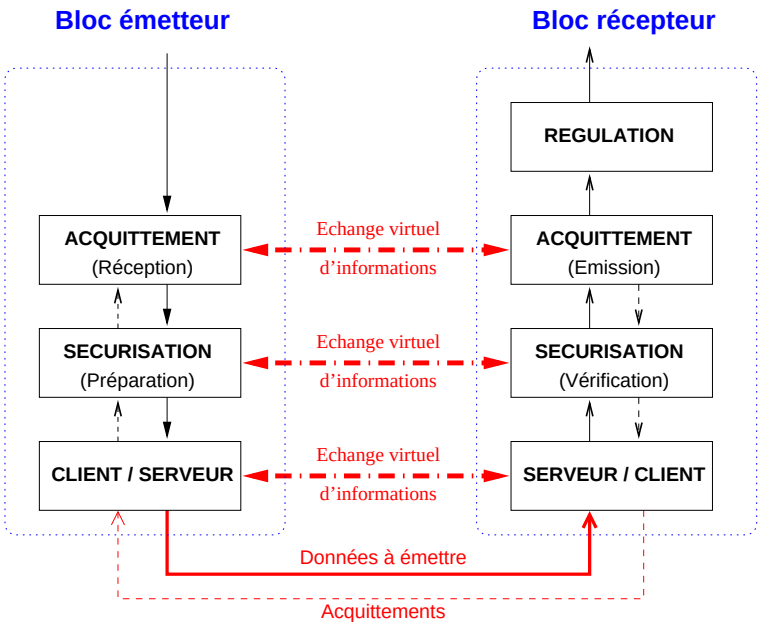


Figure 3.40 – *Ajout des régulateurs aux blocs d'émission et de réception*

répartition et de la distribution de cette fonction (cf. figure 3.42), il semblerait qu'il y ait superposition de deux phénomènes dont les valeurs nominales seraient environ 15 ms et 30 ms. Etant donné que ces retards sont mesurés au niveau de notre couche d'acquittement, il faut noter que toutes les couches inférieures influent sur ces retards. Cependant, nous n'avons pas su donner d'explication plus précise sur l'allure de cette distribution.

En régime permanent, les variations des retards aller–retour globaux $T_{total_{(A+R)}}(t)$ varient selon le même ordre de grandeur que les retards aller–retour réseau $T_{rés._{(A+R)}}(t)$ (écart-type d'environ 13 ms). Ceci est dû au système d'exploitation qui n'est pas en temps réel et qui induit des dispersions, comme nous avons pu l'étudier dans §2.3.3, page 47. Etant donné les circonstances, nous estimons que cet écart type est acceptable vis-à-vis de la période de transmission T_t de 200 ms.

La moyenne des retards aller–retour réseau de 23 ms est passée à 1,2 s après régulation.

Les périodes des signaux en entrée (figure 3.46) et en sortie (figure 3.47) du régulateur de BASE sont toujours bien centrées autour de T_t, soit 200 ms. Le régulateur a légèrement amélioré l'écart type qui passe de 14,4 ms à 8,9 ms mais l'amplitude n'est pas très bonne : 175 ms à comparer à la période de transmission T_t.

Etant donnés les résultats obtenus en termes de périodes et de retard aller–retour global, nous pouvons conclure que, dans ce type de situations, le régulateur est plus nuisible qu'utile. En effet, l'inconvénient consistant à multiplier par 50 le temps initial de trajet aller–retour n'est pas compensé par une nette amélioration des périodes et de la régularité du retard total.

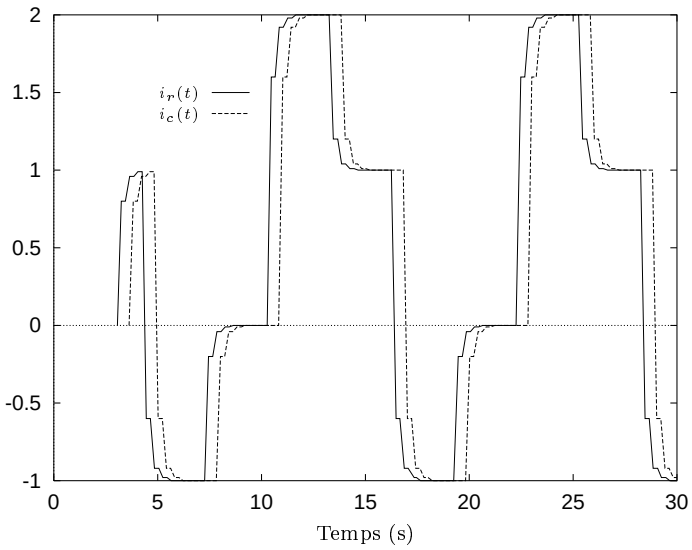
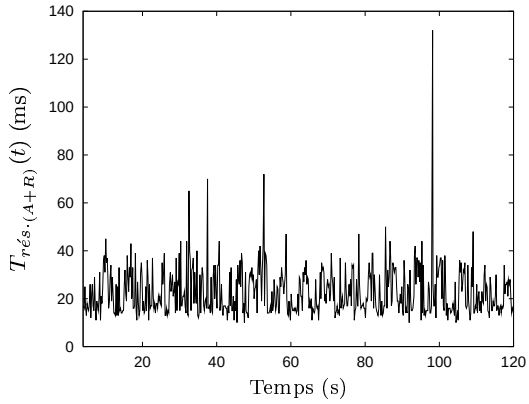
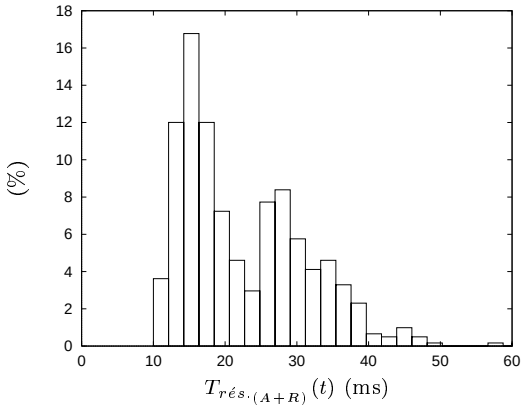


Figure 3.41 – *Expérimentation : évolution temporelle des signaux $i_r(t)$ et $i_c(t)$ (cas Béziers)*

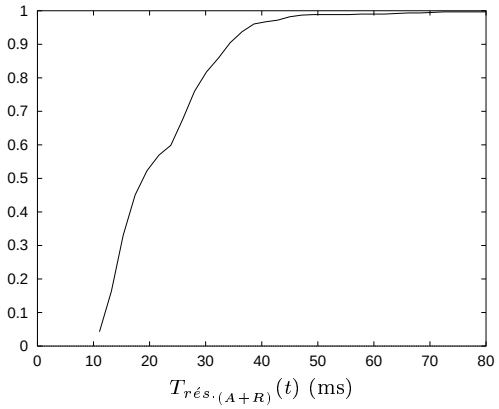
Il faut cependant nuancer cette conclusion car les mesures ont été effectuées au niveau du régulateur de BASE qui est l'hôte le moins précis du point de vue temps réel comme nous avons pu le constater dans §2.3.3, page 47.



(a) Evolution

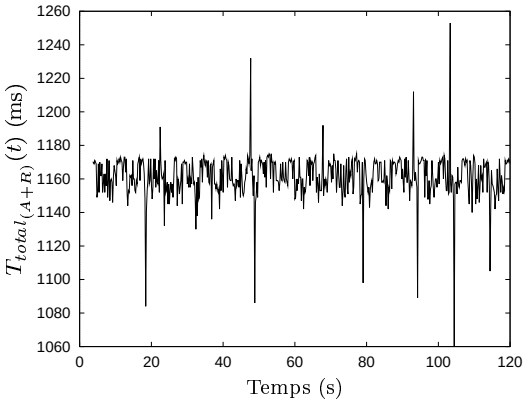


(b) Distribution

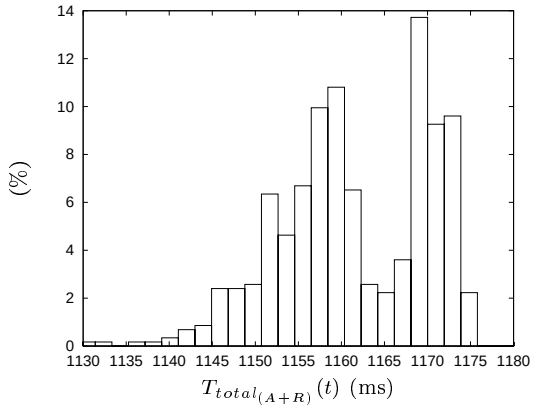


(c) Répartition

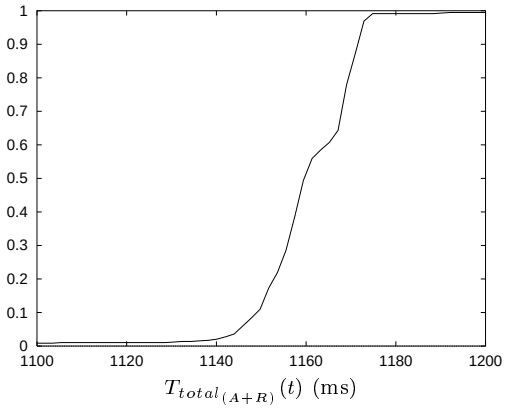
Figure 3.42 – *Expérimentation: retards dus au réseau (cas Béziers)*



(a) Evolution

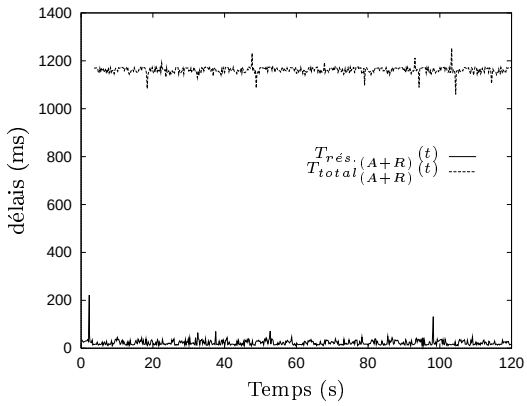


(b) Distribution

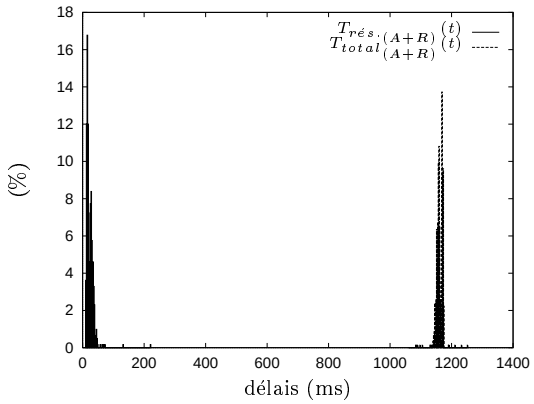


(c) Répartition

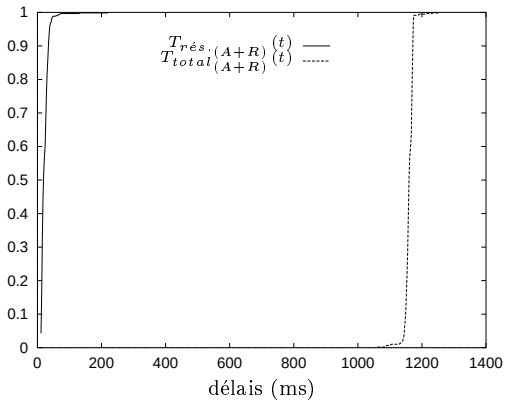
Figure 3.43 – *Expérimentation : retards globaux (cas Béziers)*



(a) Evolution

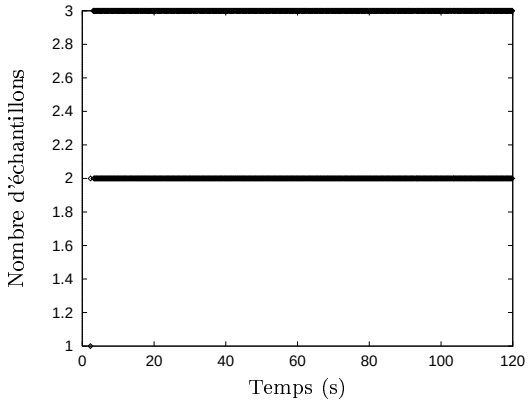


(b) Distribution

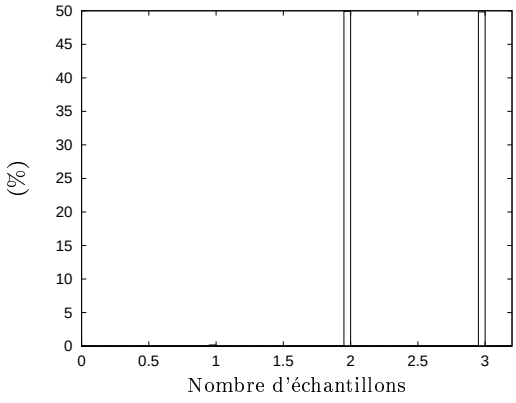


(c) Répartition

Figure 3.44 – *Expérimentation : comparaison des retards (cas Béziers)*

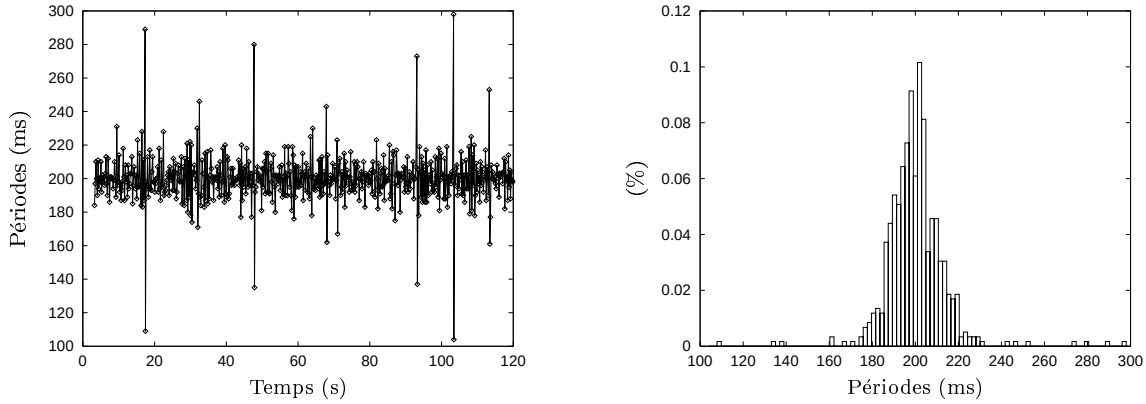


(a) Evolution



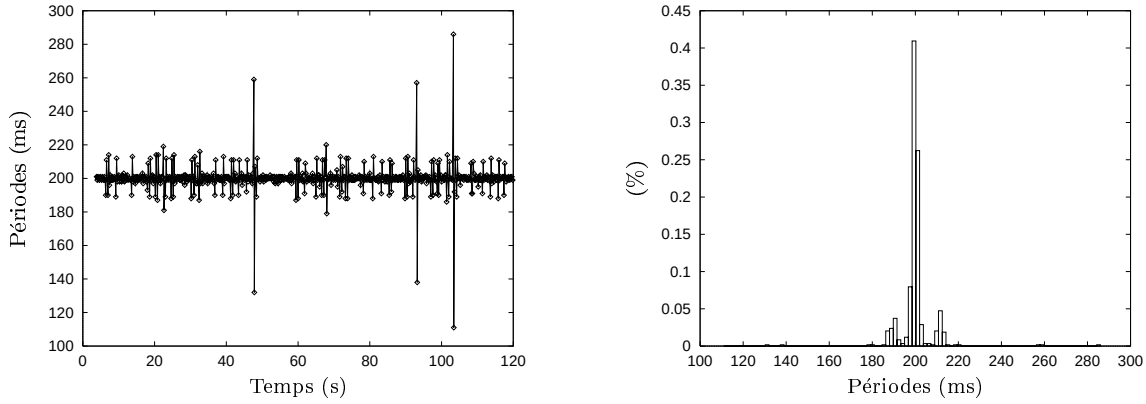
(b) Distribution

Figure 3.45 – *Expérimentation : évolution de la file d'un des régulateurs (cas Béziers)*



(a) Evolution (b) Distribution

Figure 3.46 – *Expérimentation : périodes avant régulation (cas Béziers)*



(a) Evolution (b) Distribution

Figure 3.47 – *Expérimentation : périodes en sortie d'un des régulateurs (cas Béziers)*

Insertion du relais

En pratique, il est difficile de préparer des expérimentations de téléopération à longue distance car cela nécessite la collaboration de personnes situées dans des lieux éloignés selon les cas à étudier. C'est pourquoi nous avons développé le programme RELAIS : il s'intercale entre les programmes BASE et MANIMOB afin de simuler une connexion lente paramétrable à souhait en fonction des cas à étudier (cf figure 3.48). L'application RELAIS, exécutée sur une tierce machine, est invisible du point de vue du client de BASE et du serveur de MANIMOB : ces deux derniers se comportent exactement comme s'ils étaient directement connectés l'un à l'autre. Du fait des retards infligés aux données transitant par RELAIS, BASE et MANIMOB ont tous deux l'impression d'être naturellement séparés par un réseau plus ou moins lent, à retards variables, conséquence d'un éloignement fictif. Il est alors possible de faire des expériences avec de grands délais de transmission en se cantonnant à des postes internes au laboratoire. La figure 3.49 représente ainsi l'architecture de téléopération employée pour de longues distances fictives.

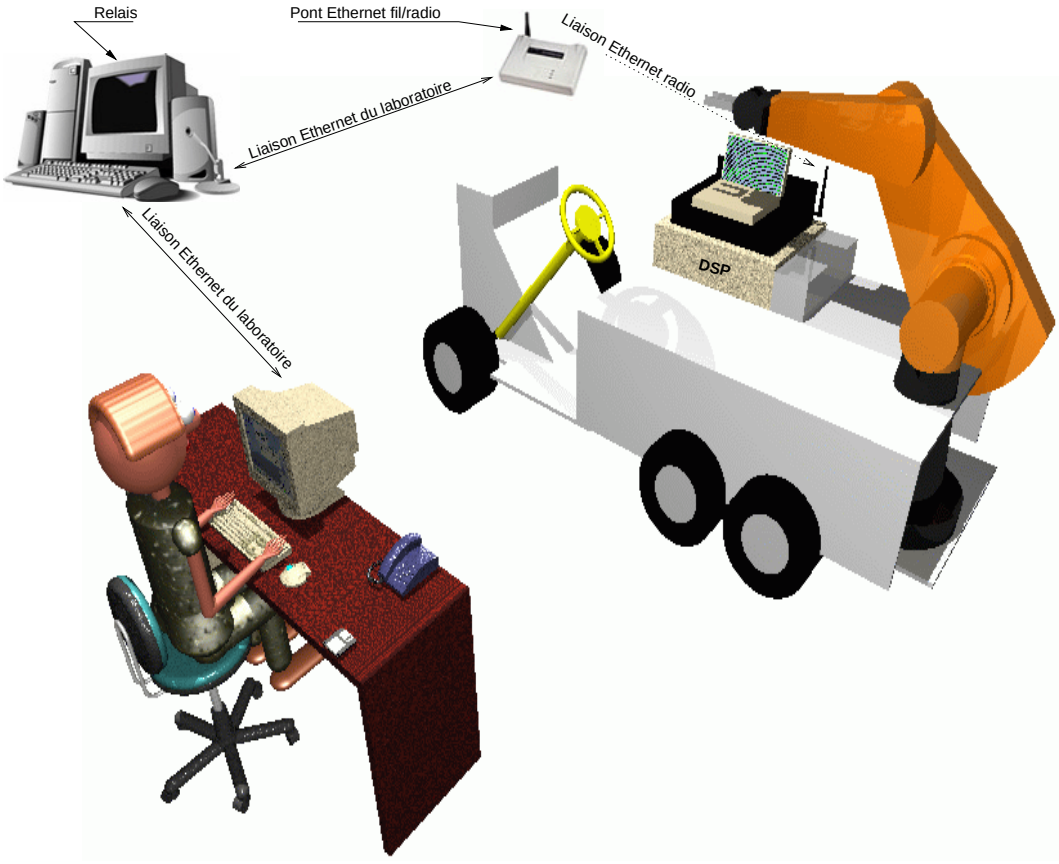


Figure 3.48 – *Plate-forme de téléopération avec relais*

Le relais inclut deux serveurs qui dialoguent avec les clients *TCP* de BASE ainsi que deux clients qui se connectent aux serveurs de MANIMOB. Il se contente d'intercepter

les données qui transitent ainsi entre les deux applications et de les retarder selon une loi statistique de retard programmée (exponentielle inverse ou normale). Il est possible de modifier les paramètres de ces lois afin de tester différents cas de figure.

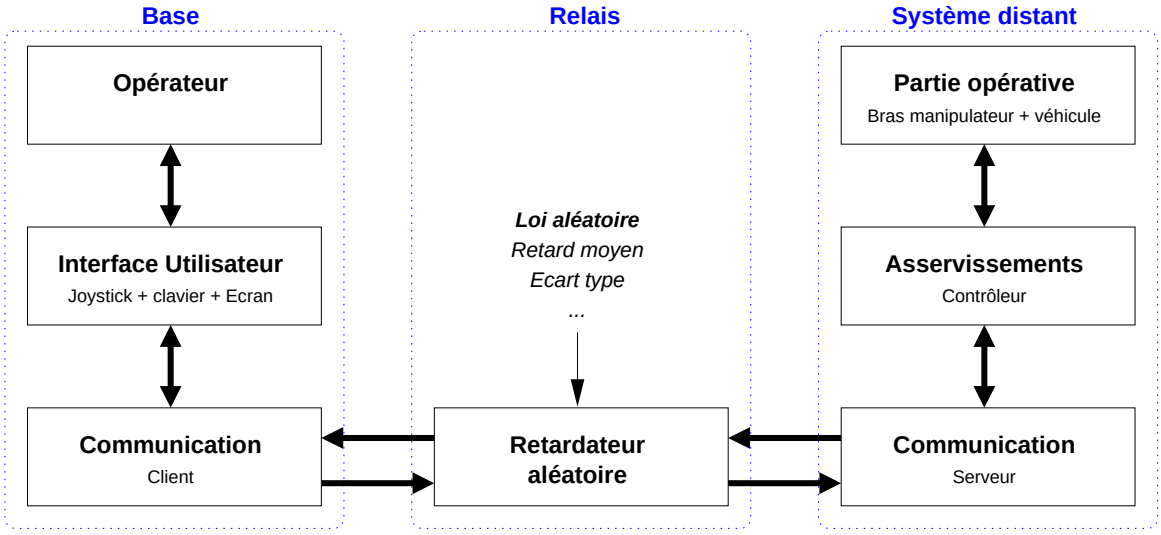


Figure 3.49 – *Architecture globale de téléopération à longue distance fictive*

RELAIS est simplement constituée d'une file d'attente dont les éléments sont empilés au fur et à mesure de leur réception. Au moment de l'empilage, une date de sortie leur est associée, calculée en fonction des paramètres fournis au relais. Comme RELAIS agit au niveau de la couche application, il ne doit surtout pas modifier l'ordre des données qu'il intercepte. En utilisant le principe de la file, nous sommes certains que les données ressortiront de la file dans le même ordre qu'elles y sont entrées. Cela a cependant pour inconvénient de modifier les dates réelles de sortie. En effet, si par hasard les dates de deux échantillons successifs étaient inversées, le premier arrivé sortirait à la date prévue suivi immédiatement du second — qui devrait déjà être sorti. Ainsi, si la distribution désirée des retards a un écart type assez élevé pour que deux dates successives soient rétrogrades, la distribution réelle des retards en sortie de la file sera déformée.

Nous n'avons pas étudié comment cette distribution pouvait évoluer. Cependant, nous avons tout de même cherché expérimentalement à représenter les déformations qui pourraient intervenir dans le cas d'une liaison fictive Montpellier–Rutgers. Nous avons alors paramétré le relais avec la loi de Poisson identifiée en §2.4.2, page 55.

La figure 3.51 donne une idée de la façon dont sont déformés les signaux après passage par le relais.

La sous-figure 3.52(a) permet de comparer les retards désirés et ceux réellement obtenus. Globalement, nous obtenons des retards légèrement plus élevés (la moyenne passant de 305 ms à 485 ms, soit un accroissement relatif de 60 %) et variant un peu plus (l'écart-type passe de 320 ms à 372 ms : +16 %).

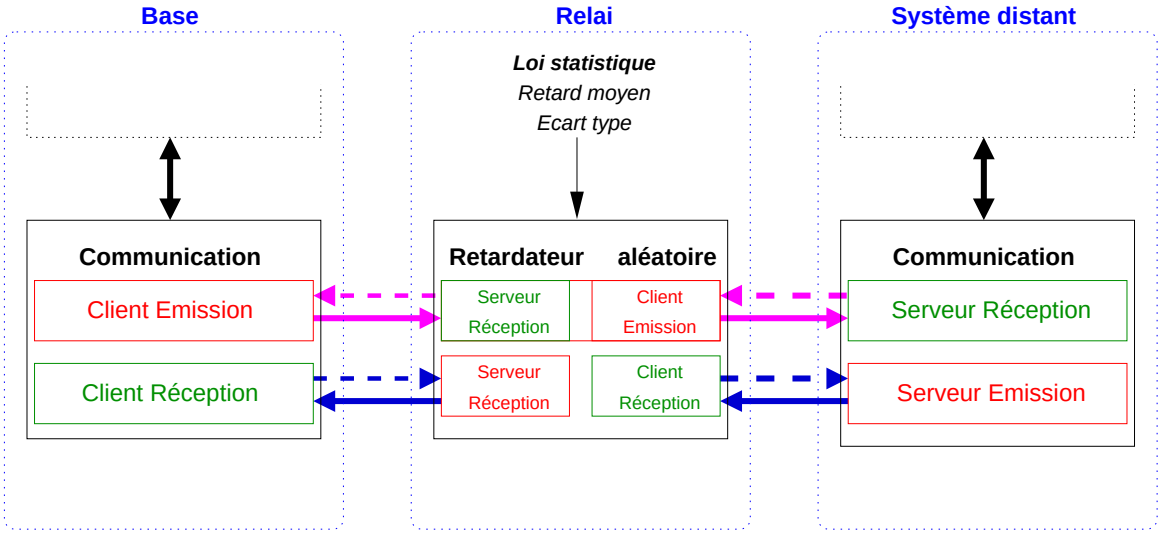


Figure 3.50 – *RELAIS : insertion dans la chaîne de transmission*

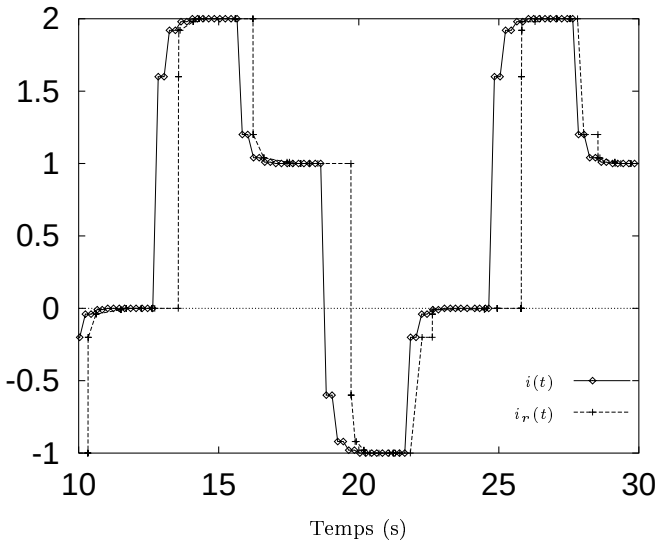
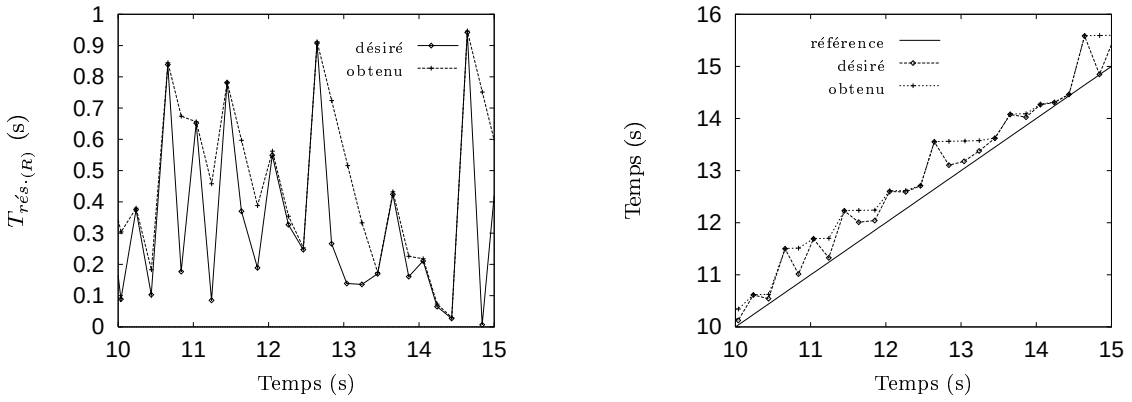


Figure 3.51 – *RELAIS : action sur les données (cas Rutgers AM)*

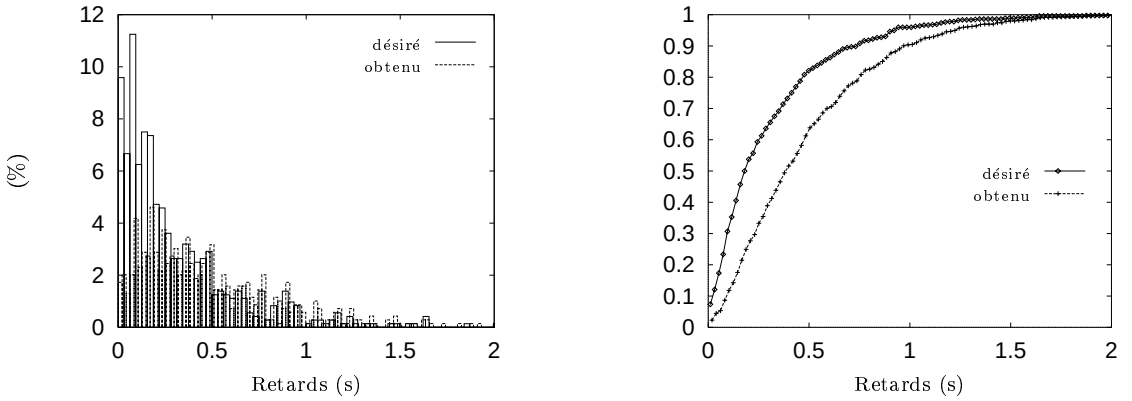


(a) Retards (b) Dates de sortie

Figure 3.52 – *RELAIS : profil des retards désirés et obtenus (cas Rutgers AM)*

La sous-figure 3.52(b) illustre les dates de sortie désirées et obtenues. Nous vérifions facilement le bon fonctionnement du relais en notant que, lorsque la courbe des dates désirées baisse localement, la courbe des retards obtenus reste constante. Cela signifie que toutes les données ressortent successivement, quasiment immédiatement.

La figure 3.53 permet de comparer les distributions et répartitions désirées et obtenues. Nous pouvons constater que la distribution des retards obtenus est beaucoup plus étalée que celle des retards désirés, conformément à nos attentes. Il semblerait, vu la forme de la distribution et le léger coude au départ de la courbe de répartition des retards obtenus, que nous obtenions ici une gaussienne au lieu d'une loi de Poisson.



(a) Distribution (b) Répartition

Figure 3.53 – *RELAIS : comparaison des retards désirés et obtenus (cas Rutgers AM)*

RELAIS nous a rendu un grand service, faute d'interlocuteurs outre atlantique. Etant paramétrable, il peut simuler n'importe quel type de lien de transmission utilisable lors d'une téléopération. Cependant, du fait de sa conception, il agit sur les données au

niveau de la couche application. Mais son utilisation présente des inconvénients :

- il doit maintenir l'ordre des données, ce qui peut influer sur la distribution des retards au-delà d'un certain écart-type dépendant de la période d'émission initiale T_t des données et du retard réellement dû au réseau local,
- nous ne tenons pas compte des traitements opérés dans les couches réseau intermédiaires, or ces traitements s'avèrent non négligeables du point de vue de la distribution réelle des retards.

D'autre part, si nous étudions les retards calculés entre le moment où les données sont émises depuis la couche matérielle (cf. figure C.1, page 176) et le moment où elles sont réceptionnées au niveau de cette même couche par l'hôte destinataire, nous constatons que ces retards cumuleraient tous les traitement successifs de toutes les couches de protocoles traversées au niveau du relais (celles du récepteur et celles de l'émetteur) et des retards obtenus au niveau du relais. Il y a donc beaucoup de traitement qui n'est pas pris en compte dans le paramétrage du relais et dont il faudra tenir compte dans les résultats suivants.

Pour être plus précis, il faudrait agir au plus bas niveau possible. Ainsi, en intercalant un routeur *IP* modifié pour retarder les trames qu'il traite, nous pourrions déjà éviter les traitements des couches de transport *TCP* et de nos propres couches d'acquittement et de sécurisation. Un tel routeur peut se concevoir sans trop de difficultés à partir d'un *PC* équipé de deux cartes réseau et dont le système d'exploitation serait LINUX. En effet, ce système est particulièrement adapté pour ce genre d'applications. Enfin, l'idéal serait d'agir directement au niveau de la couche physique, ce qui sous-entend de développer un relais électronique.

Résultats à longue distance : Montpellier–Rutgers (matin)

La figure 3.54 donne un aperçu des signaux entrant et sortant du régulateur de BASE. Nous retrouvons bien l'allure du signal $i(t)$ représenté en figure 3.51 avant son passage par RELAIS.

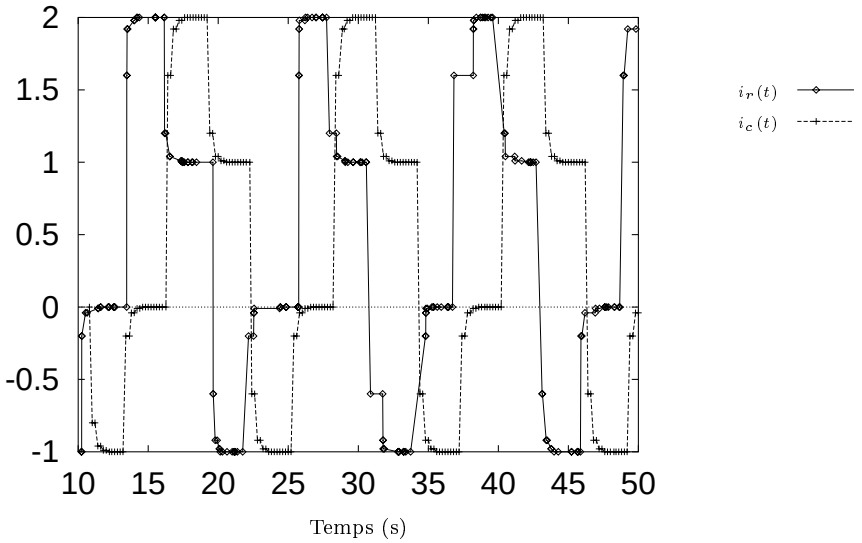


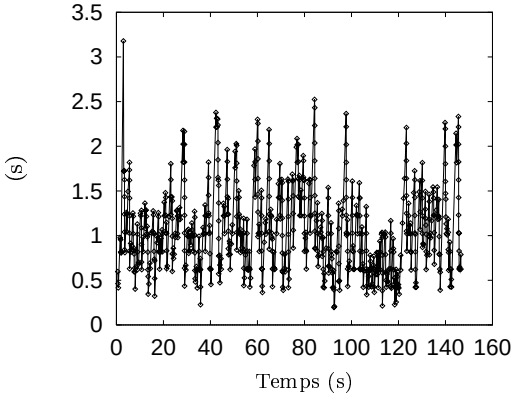
Figure 3.54 – *Expérimentation : action du régulateur de BASE sur les données transmises (cas Rutgers)*

Encore une fois, les retards aller–retour dus au réseau $T_{rés.(A+R)}(t)$ ont une distribution et une répartition un peu déroutantes, d'autant plus cette fois ci, que RELAIS participe activement à l'élaboration de ces retards. Nous pouvons constater sur la figure 3.3.5 que nous sommes assez loin des retards désirés (nous les avons reportés sur cette figure après avoir doublé leurs valeurs puisque nous considérons ici un retard aller–retour) tant au niveau des distributions que des valeurs caractéristiques : nous observons une moyenne de 610 ms pour les retards aller–retour désirés contre 1069 ms et des écarts types respectifs de 635 ms et 465 ms.

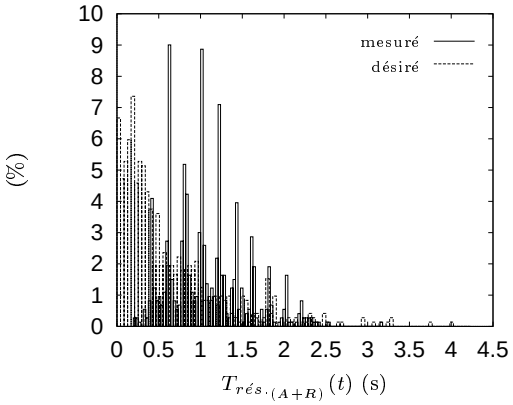
En ce qui concerne les variations du retard global aller–retour (visibles en figure 3.56), leur moyenne se situe autour de 6,85 s avec un écart type de l'ordre de 12 ms et une amplitude de 480 ms, du même ordre que ceux de l'expérimentation Montpellier–Béziers. Ce qui tend à prouver qu'il s'agit d'une faiblesse en précision de l'horloge de l'application. Avant régulation, le retard aller–retour avait une amplitude d'environ 3 s ; ici au moins, il y a un gain.

La période en sortie du régulateur (cf figure 3.57) est toujours aussi bien centrée sur 200 ms et l'écart type est de l'ordre de 15 ms. Cet écart type serait acceptable si l'amplitude maximale autour de la moyenne était du même ordre de grandeur. Malheureusement, ce n'est pas le cas puisqu'elle atteint localement 155 ms, ce qui n'est pas négligeable à côté de la période de transmissions.

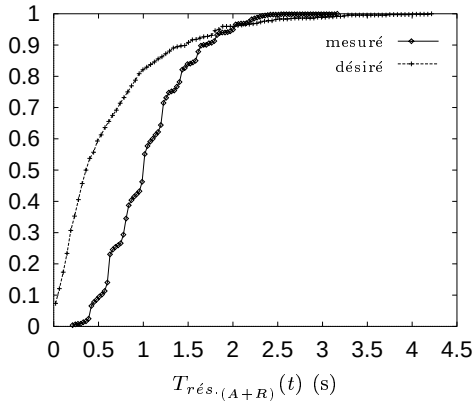
La figure 3.58 illustre l'encombrement de la file du régulateur. A première vue, il semblerait que $n_{désiré}$ soit beaucoup trop grand puisque la file n'est pas optimisée avec



(a) Evolution



(b) Distribution

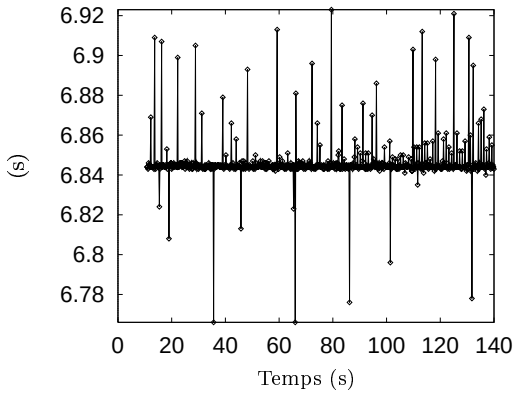


(c) Répartition

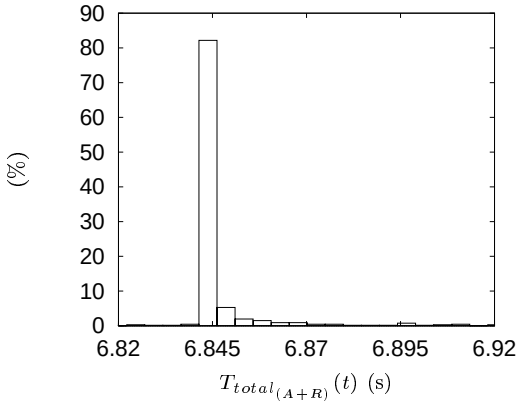
Figure 3.55 – *Expérimentation: retards réseau mesurés (cas Rutgers)*

un $n_{min} = 7$. Cependant, les conditions initiales jouent beaucoup sur la moyenne du nombre d'échantillons présents dans la file. D'autres essais ont laissé apparaître un $n_{min} = 4$, ce qui est satisfaisant, vu les variations du remplissage avec un $n_{désiré} = 13$. Pour information, en régime permanent, la moyenne est située ici à 15,6 échantillons, l'écart type de 2 échantillons et l'amplitude de 12.

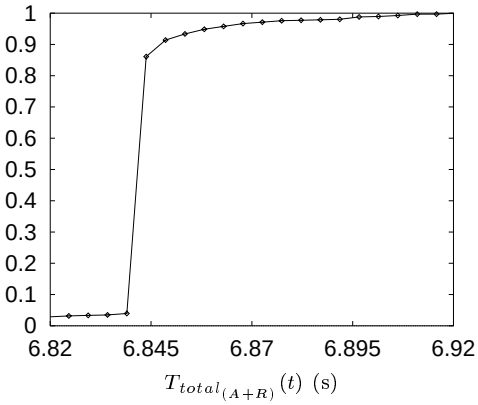
La figure 3.59 permet de comparer les retards avant et après régulation. Nous pouvons remarquer que les retards réseau sont loin d'atteindre les retards globaux (maximum à environ 3 s pour $T_{rés.(A+R)}(t)$ contre 7 s environ pour $T_{total_{(A+R)}}(t)$) mais l'écart est ici nettement moins disproportionné que dans le cas de la liaison Montpellier–Béziers. Il faut penser que des retards très localisés ont pu élever le $n_{désiré}$ et, par conséquent, élever le $T_{total_{(A+R)}}(t)$ à ce point pendant la période d'audit.



(a) Evolution

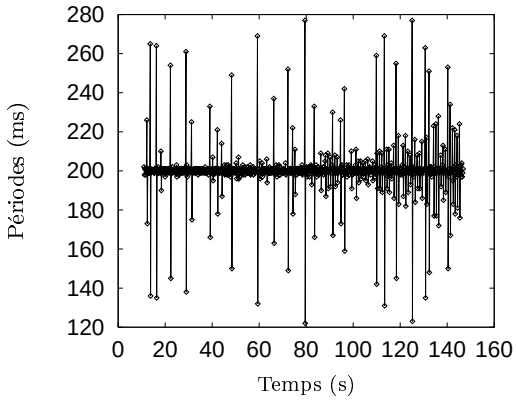


(b) Distribution

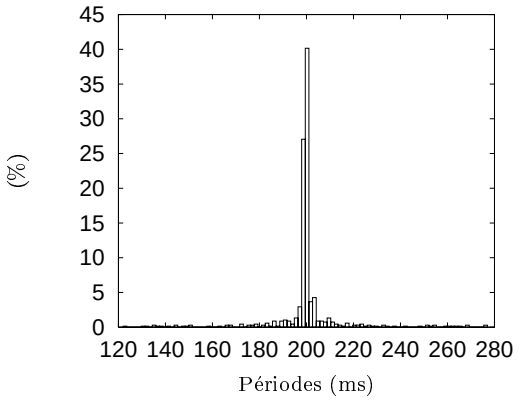


(c) Répartition

Figure 3.56 – *Expérimentation: retards globaux mesurés (cas Rutgers)*

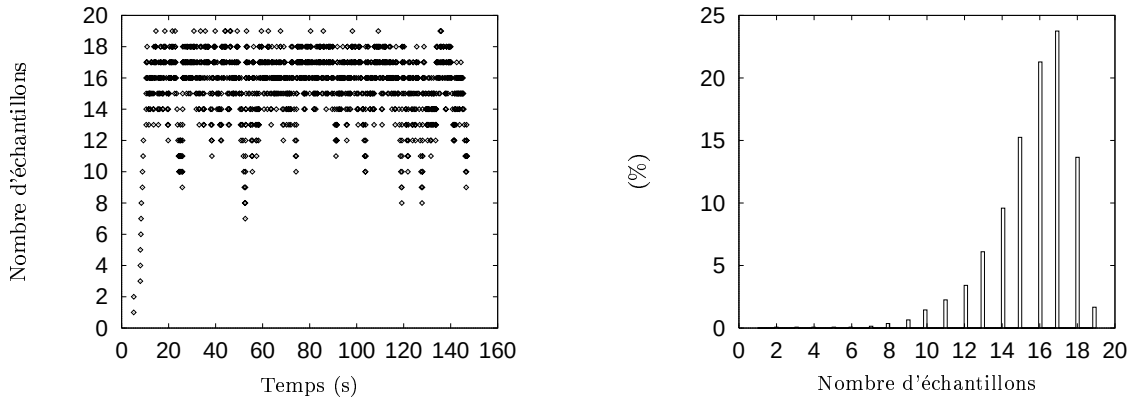


(a) Evolution



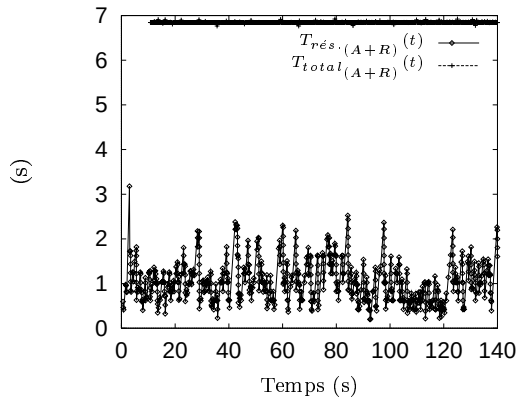
(b) Distribution

Figure 3.57 – *Expérimentation: période en sortie d'un régulateur (cas Rutgers)*

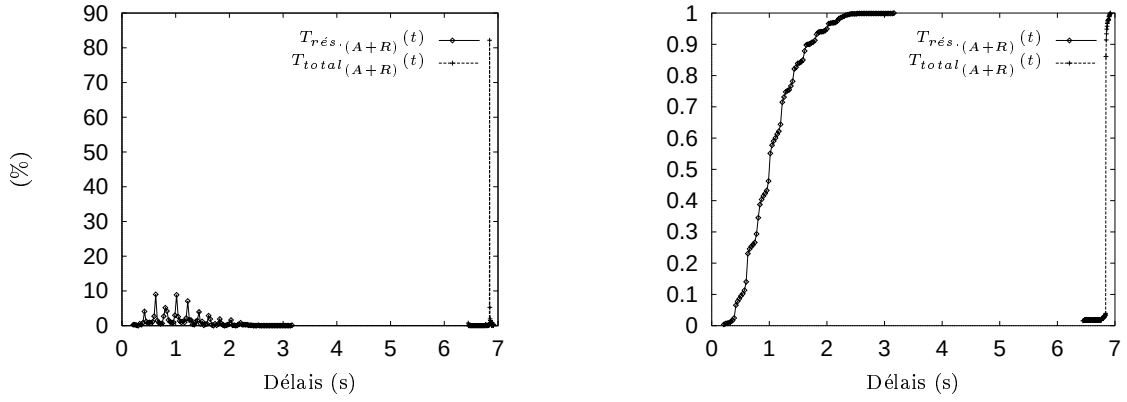


(a) Evolution (b) Distribution

Figure 3.58 – *Expérimentation : taille de la file d'attente (cas Rutgers)*



(a) Evolution



(b) Distribution (c) Répartition

Figure 3.59 – *Expérimentation : comparaison entre retards avant et après régulation (cas Rutgers)*

Conclusions sur ces expérimentations

Dans un premier temps, nous avons pu vérifier expérimentalement l'équation suivante ($n_A(t)$ et $n_R(t)$ sont le nombre d'échantillon dans chacun des régulateurs) :

$$T_{total_{(A+R)}}(t) = min(T_{rés._{(A+R)}}(t)) + max(n_A(t) + n_R(t)) \times T_t$$

La qualité de l'horloge de BASE ne permet pas de conclure objectivement sur l'utilité d'un tel régulateur à courte distance. En effet, les régulateurs n'apportent pas ici d'amélioration notable du point de vue de la constance du retard global et de la période des signaux et détériorent nettement l'amplitude du retard aller–retour.

Cependant nous pouvons tout de même tirer quelques conclusions. Ces résultats montrent qu'une adaptation de la période de transmissions est indispensable pour obtenir des résultats corrects lorsque les retards dus aux transmissions ont une échelle inférieure à cette période. Ils montrent aussi clairement que les systèmes d'exploitation présents sur et autour du manipulateur mobile, c'est-à-dire *Windows* 95 et *Windows NT4* ne permettent pas de réaliser des fonctions aussi basiques que celles testées. Un recours à un *vrai* système d'exploitation temps–réel est donc indispensable pour obtenir des résultats tangibles. Cela impose toutefois de remplacer l'ensemble *DSP* par un matériel compatible.

En ce qui concerne les résultats à longue distance fictive, les mêmes constatations s'appliquent, bien que la période de transmission soit mieux adaptée. Nous avons pu également constater pendant les divers essais que les conditions initiales de remplissage des files des régulateurs influençaient la tenue des régulateurs en régime permanent de manière non négligeable.

De plus, la conception du relais n'est pas assez réaliste pour simuler une liaison à longue distance. Du fait de son insertion au niveau *TCP*, il doit conserver les trames dans l'ordre d'émission. Nous observons alors une déformation de la loi statistique désirée pour les retards infligés aux trames qu'il traite.

3.3.6 Conclusions sur la régulation

Le modèle de simulation que nous avons développé nous a permis, dans une première phase, de tester le bien fondé du principe de régulation adopté. Nous avons obtenu de très bons résultats, notamment du point de vue de la régularité des périodes des signaux en sortie des régulateurs. Ces simulations nous ont permis de déterminer une méthode de calcul en vue de paramétrer la taille appropriée des files des régulateurs $n_{désiré}$ en fonction des caractéristiques du médium de transmission (cf équation (3.54)).

Nous avons pu observer les limitations de ce type de régulation pour des transmissions à courte distance. Il apparaît en effet que, lorsque les variations des retards dus au médium de transmission sont nettement inférieures à la période de transmission, le régulateur est obligé de garder les échantillons pendant une durée proportionnelle à cette période, ce qui a pour effet un temps de trajet global nettement disproportionné par rapport au temps de trajet dû uniquement au réseau.

Les expérimentations que nous avons réalisées nous ont permis de confirmer le choix de la méthode de calcul de $n_{désiré}$. En effet, les différentes séries d'essais effectués avec les valeurs de $n_{désiré}$ calculées ainsi ont fonctionné sans que les files se vident complètement et sans surdimensionnement notable lorsque les retards de transmissions sont du même ordre de grandeur que la période T_t.

La qualité des périodes en sortie des régulateurs est très dépendante du système d'exploitation utilisé pour la *base* et le *système distant*. Notre équipement n'était pas vraiment adapté à ce type de travail puisque nous n'avions pas de système d'exploitation temps réel compatible avec notre matériel. Il en a résulté des résultats d'une qualité médiocre quant à la période en sortie des régulateurs et à la constance des retards globaux.

3.4 Estimation de l'état du système distant

3.4.1 Contexte

Nous nous plaçons dans le cas où les retards sont correctement régulés, c'est-à-dire en régime permanent tel que nous l'avons défini en §3.3.3, page 97. Notre but est, d'une part, d'estimer l'état actuel du système téléopéré et, d'autre part, de prédire son état dès l'émission des consignes. Le bloc de prédiction/estimation trouve sa place dans la *base* et se nourrit des signaux émis et reçus par celle-ci. La figure 3.60 représente l'ensemble de la boucle de téléopération qui est commentée sous forme d'étapes :

1. A l'instant t, l'opérateur envoie une consigne $x_d(t)$.
2. Cette consigne est échantillonnée toutes les $T_t = 200\ ms$ et envoyée au *système distant* via le lien de transmission : $c(t)$.
3. Le lien de transmission la retarde d'un laps de temps noté $T_{r_{(A)}}(t)$, générant ainsi le signal $c_r(t) = c(t - T_{r_{(A)}}(t))$.
4. Elle parvient au *système distant* à l'instant $t + T_{r_{(A)}}(t)$ où elle est stockée par son régulateur pour ne ressortir qu'à l'instant $t + T_{total_{(A)}}$.
5. Elle est transmise à la commande représentée ici par le filtre passe-bas du second ordre échantillonné utilisé jusqu'ici et dont les paramètres sont donnée en §2.4, page 51.
6. Le *système distant* transmet alors la position actuelle $i(t) = x(t - T_{total_{(A)}})$ à la *base*.
7. Cette dernière reçoit cette donnée à l'instant $t + T_{total_{(A)}} + T_{r_{(R)}}$.
8. Cette donnée ressort alors du régulateur de la *base* à l'instant $t + T_{total_{(A)}} + T_{total_{(R)}} = t + T_{total_{(A+R)}} : x(t - T_{total_{(A+R)}})$

9. Elle est alors transmise au prédicteur/estimateur. Celui-ci, à partir des signaux $X_d(t)$ et $X(t - T_{total_{(A+R)}})$ se charge de générer les signaux $\hat{c}[k - K_{(A)}]$, $\hat{c}_c[k] = \hat{x}[k - K_{(A)}]$ et $\hat{i}_c[k] = \hat{x}[k]$.

Ainsi, pour chaque consigne $x_d[k]$ envoyée, la position mesurée au moment de la réception de la consigne par la commande du bras manipulateur est récupérée par le prédicteur de la *base* après un laps de temps égal à $T_{total_{(A+R)}}$.

Comme les signaux $c(t)$ et $i_c(t)$ sont périodiques, leurs versions échantillonnées à la période T_t sont notées respectivement $c[k] = x_d[k]$ et $i_c[k] = x[k - K_{(A+R)}]$ avec $K_{(A+R)} = \frac{T_{total_{(A+R)}}}{T_t}$. Etant donné que l'échantillonneur de l'émetteur et du régulateur utilisent la même horloge, le temps $T_{total_{(A+R)}}$ est forcément un multiple de T_t.

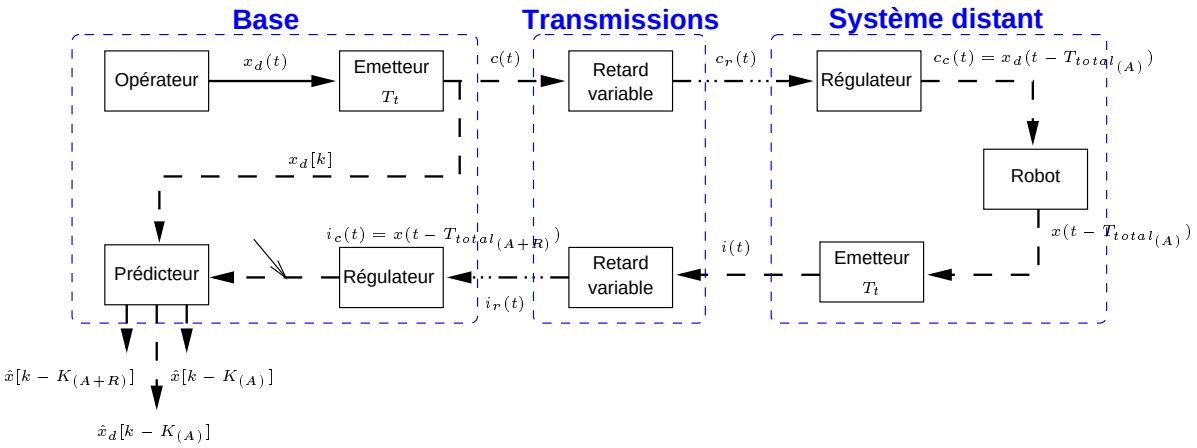


Figure 3.60 – *Architecture globale pour la prédiction/estimation*

3.4.2 Estimateur/prédicteur

Au niveau de l'estimation du comportement actuel du robot, la difficulté réside dans le fait que nous devons comparer notre signal estimé avec le signal réel à chaque pas d'échantillonnage pour obtenir une estimation dynamique. Or, les signaux réels ne sont disponibles qu'après un temps égal au retard de retour. La solution que nous avons adoptée est présentée en figure 3.61.

Les retards aller et les retards retour sont connus, ils ont été identifiés en §3.3.3, page 100 et sont calculés en temps réel. A partir du moment où l'estimateur/prédicteur connaît les délais aller et retour, le fait que ceux-ci soient égaux n'a pas réellement d'importance. Nous obtenons simplement un système symétrique.

Le bloc « identificateur » décrit ci-dessous se charge d'identifier les paramètres du *système distant*. Il génère ainsi les coefficients du filtre échantillonné équivalent au comportement du robot. Ces coefficients ainsi qu'une information sur la stabilité du filtre qui leur est associé sont envoyés aux deux filtres à coefficients variables. Ces

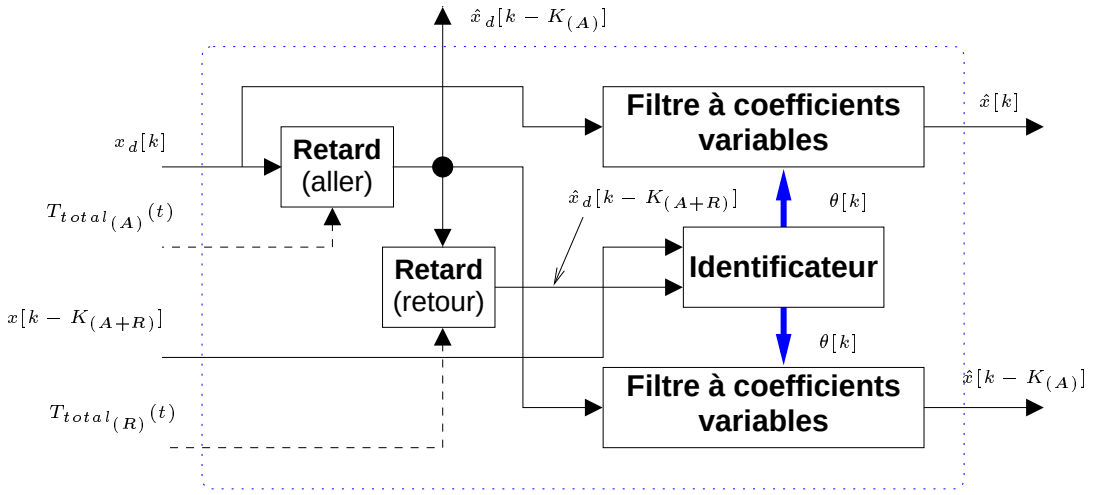


Figure 3.61 – *Détails du bloc de prédiction*

deux filtres génèrent pour l'un une estimation de l'état du *système distant* $\hat{x}[k-K_{(A)}]$ grâce à une version retardée (du retard aller après régulation $T_{total_{(A)}}$) des consignes : $x_d[k-K_{(A)}]$ et, pour l'autre, une prédiction (ou estimation anticipée) de l'état du *système distant* $\hat{x}[k]$ avant même qu'il ne reçoive les consignes. Cette prédiction est réalisée en utilisant directement le signal $x_d[k]$ comme entrée du filtre.

3.4.3 Identificateur

L'avantage de la conception de l'estimateur/prédicteur envisagé précédemment est de pouvoir utiliser n'importe quel type d'« identificateur ».

Dans cette section, nous avons décidé de faire appel à un filtre adaptatif à gradient. Cette architecture a été présentée initialement dans [LEL1 99]. En §4.2.3, page 144, nous avons préféré un filtre adaptatif fondé sur l'algorithme des moindres carrés récursifs.

Filtre adaptatif à gradient

Le bloc identificateur est construit à partir d'un filtre adaptatif normalisé du premier ordre issu de [BEL 89]. Son équation récurrente est la suivante ($y[k]=\hat{x}[k-K_{(A+R)}]$ et $u[k]=x_d[k-K_{(A+R)}]$) :

$$y[k+1]=a_{1e}[k].u[k]+a_{2e}[k].u[k-1]+b_{1e}[k].y[k] \tag{3.55}$$

où $a_{1e}[k]$, $a_{2e}[k]$ et $b_{1e}[k]$ sont les coefficients estimés à l'instant k

Ce bloc calcule les coefficients les plus appropriés (selon la méthode du gradient) en comparant son estimation du signal de retour $\hat{x}[k-K_{(R)}]$ avec le signal de retour réel

$x[k - K_{(R)}]$. Pour ce faire, le bloc utilise un filtre échantillonné à coefficients variables ayant comme entrée les consignes de l'opérateur retardées d'un laps de temps égal au temps de trajet total aller–retour $T_{total_{(A+R)}}$. Le processus est schématisé en figure 3.62.

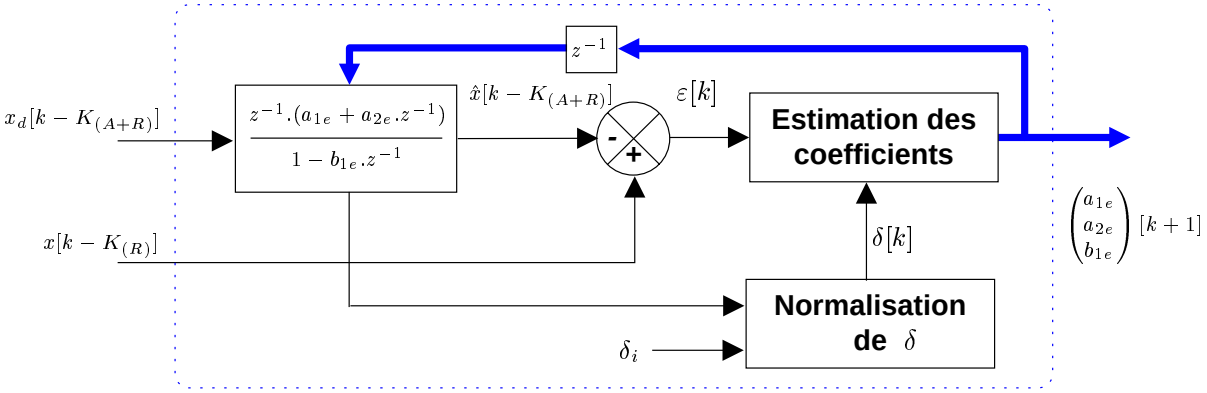


Figure 3.62 – *Détails du bloc filtre adaptatif*

Les coefficients ont pour expression :

$$\varepsilon[k] = x[k - K_{(R)}] - \hat{x}[k - K_{(R)}] \tag{3.56}$$
$$a_{1e}[k] = a_{1e}[k-1] + \delta[k].\varepsilon[k].\varepsilon[k-1] \tag{3.57}$$
$$a_{2e}[k] = a_{2e}[k-1] + \delta[k].\varepsilon[k].\varepsilon[k-2] \tag{3.58}$$
$$b_{1e}[k] = b_{1e}[k-1] + \delta[k].\varepsilon[k].\hat{x}[k - K_{(R)} - 1] \tag{3.59}$$

δ_k est un paramètre dynamique ; le bloc *Normalisation* calcule la moyenne des carrés des N derniers échantillons et divise δ_i par cette moyenne.

$$\delta[k] = \frac{\delta_i}{\frac{1}{N+1}\sum_{i=0}^{i=N} (x[k - K_{(R)} - i])^2} \tag{3.60}$$

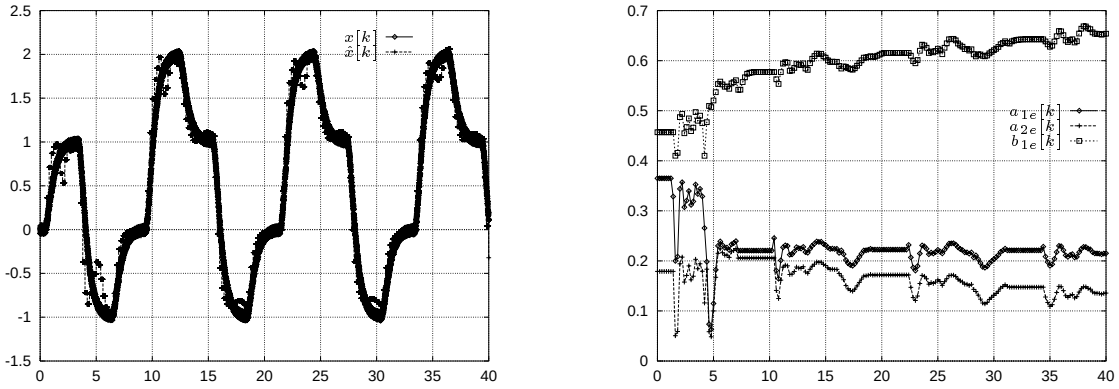
Un rebouclage vérifie la stabilité des nouveaux coefficients. En cas d'instabilité, ces coefficients sont recalculés par une valeur de δ_k inférieure, ceci m fois. Un signal prévient les autres blocs si aucun résultat stable n'a été trouvé à terme et les coefficients émis sont les derniers coefficients stables connus.

3.4.4 Validation par simulation

La figure 3.63(a) montre que l'estimation en régime permanent est ici fiable et converge assez rapidement mais les changements de consignes éloignent localement

l'estimation de la réalité. La figure 3.64(a) montre que l'estimation en régime permanent est meilleure dans ce cas. Mais le régime transitoire est plus long à converger :

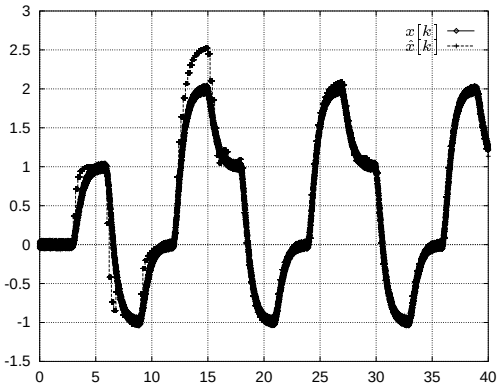
- de 0 à 7 s : le signal d'entrée du bloc estimateur est nul ; le filtre adaptatif ne peut réagir avec une entrée constamment nulle,
- de 7 à 12 s : les coefficients sont stables mais varient très rapidement. Ceci est dû au bloc normalisateur qui n'est pas encore en régime permanent. En effet, à cause des premières valeurs nulles, δ_k est trop élevé, ce qui donne une amplitude exagérée aux variations des coefficients estimés. La figure 3.64(b) illustre cette évolution.



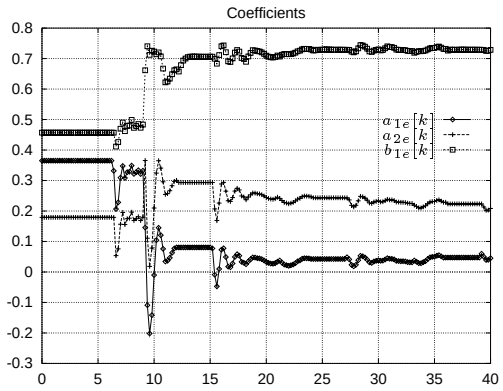
(a) État du système distant et estimation

(b) Évolution des coefficients

Figure 3.63 – *Résultats de simulation pour Montpellier–Béziers*



(a) État du système distant et estimation



(b) Évolution des coefficients

Figure 3.64 – *Résultats de simulation pour Montpellier–Rutgers*

3.4.5 Conclusion sur la prédiction/estimation

Le schéma que nous avons proposé dans la figure 3.60 permet, d'une part, d'estimer l'état du système distant à chaque instant et, d'autre part, de prédire son état dès l'émission de la consigne par l'opérateur. Il a l'avantage d'être indépendant de la méthode employée pour l'identification du modèle du *système distant* et du modèle lui-même. Nous pourrions imaginer d'employer un modèle beaucoup plus précis et pas nécessairement linéaire.

Dans cette étude, nous avons fait appel à un filtre à gradient. Il s'agissait avant tout de valider le concept avec un filtre facile à mettre en œuvre et peu gourmand en temps de calculs.

3.5 Conclusion

Nous nous sommes donc penché en premier lieu sur les problèmes de stabilité liés à des systèmes du premier ordre associés à un correcteur proportionnel dans des situations à retard constant et variable sous différentes allures. Nous avons pu distinguer deux cas intéressants :

- $K.G < 1$: le système sera stable quel que soit le retard (constant ou variable tant qu'il est continu)
- $K.G < K_{max}(T_{rmax}).G$ où $K_{max}(T_{rmax})$ dépend de la forme de $T_r(t)$: le système sera stable pour un gain $K < K_{max}$ dépendant de la borne supérieure T_{rmax}.

Nous avons pu remarquer que certains cas de retards variables peuvent « préserver » un peu plus de stabilité qu'un retard constant, contrairement à ce que nous pouvions présumer.

Nous avons également étudié un système du second ordre en présence de retards constants. Nous avons réussi à déterminer une expression littérale de la limite de stabilité en fonction du retard et des différents paramètres du système. Ce résultat est particulièrement intéressant pour nous car il va nous permettre de commander des moteurs et donc des robots en présence de retards constants. Ce résultat peut également aider à synthétiser un correcteur du premier ordre pour commander un processus du premier ordre, ce qui est déjà plus pratique qu'un simple correcteur proportionnel.

Le besoin d'obtenir un retard constant dans la boucle de téléopération nous a logiquement poussé à développer un compensateur de retards. Si le concept a été validé en simulation, les moyens techniques mis en œuvre ne nous ont pas permis de le valider expérimentalement de manière probante surtout lorsque les retards de transmission sont faibles devant la période d'émission T_t. Nous en avons tout de même tiré quelques enseignements notamment sur le besoin d'adapter la période de transmissions et la taille des files des régulateurs aux variations lentes des caractéristiques des retards.

Notre étude bibliographique nous a conforté dans l'idée qu'une estimation, voire même une prédiction de l'état du *système distant*, est indispensable en terme d'ergo-

nomie de l'interface homme-machine. Nous avons donc élaboré un système apte, d'une part, à estimer l'état du système distant à chaque instant et, d'autre part, à prédire son état dès l'émission de la consigne par l'opérateur. Il s'agit d'un système indépendant de la technique employée pour l'identification du modèle du *système distant* et du modèle lui-même de ce système. Il a été testé en simulation avec un filtre à gradient mais nous n'avons malheureusement pas eu la possibilité de le tester expérimentalement.

Chapitre 4

Application

4.1 Introduction

Afin de tester nos études sur un cas réel de téléopération à longue distance, nous avons imaginé une tâche particulière à réaliser au sein d'un scénario représentatif. Ce travail a été présenté dans [LEL 00].

Nous nous sommes placés dans le cas où nous avions besoin de transporter et de déposer une charge explosive dans un chantier de construction. Ce genre de manipulation est généralement effectué par un artificier, à ses risques et périls.

Nous avons donc proposé une solution semi-robotisée permettant de limiter les risques d'explosion accidentelle en présence d'hommes. Nous avons proposé l'utilisation d'un manipulateur mobile. La tâche à effectuer peut se répartir en quatre phases.

La première phase consiste à transporter la charge explosive jusqu'au lieu de dépose. La partie véhicule du manipulateur mobile est téléopérée par l'opérateur qui la téléguide depuis son poste pendant que le bras manipulateur tient la charge en lui évitant tout choc pouvant déclencher son explosion. Cette commande d'impédance a été développée et publié dans [FRA 95]. Ainsi l'opérateur n'a pas a se soucier des vibrations et chocs — de toute façon, il est très difficile pour lui de les anticiper —; il peut se concentrer uniquement sur le chemin qu'il emprunte. De plus, il peut transporter la charge à une vitesse supérieure à celle d'un humain transportant la charge lui-même. L'avantage de ce choix, comparé à une solution complètement automatisée, est que nous pouvons modifier l'itinéraire du véhicule et éviter les obstacles relativement facilement.

Une fois arrivé à proximité du lieu de dépose, il y a de fortes chances pour que le véhicule ne soit pas positionné correctement pour déposer la charge simplement. La deuxième phase consiste alors à amener le véhicule et le bras dans une configuration permettant de déposer la charge le plus facilement possible. Nous faisons alors appel aux algorithmes de mouvements coordonnés du bras et du véhicule présentés dans

Cas	**Montpellier–Béziers**	**Montpellier–Rutgers**	
		matin	après-midi
Retard moyen (ms)	4.55	770	1270
Amplitude des variations (ms)	7,19	590	610

Tableau 4.1 – *Paramètres des retards pour les phases de téléopération*

[DAU 99].

Une fois le véhicule et le bras positionnés de telle façon que l'emplacement final soit situé dans l'espace opérationnel du bras, l'opérateur n'a plus qu'à téléopérer le bras pour achever délicatement le mouvement de dépose de la charge. C'est la troisième phase.

Enfin, la phase finale consiste à dégager le manipulateur mobile de la zone d'explosion, par une téléopération similaire à celle de la phase 1, sans mouvement du bras manipulateur.

Du point de vue des distances, l'opérateur a besoin d'être hors d'atteinte si une manipulation malheureuse venait à arriver. Plusieurs solutions sont envisageables. La plus générique consiste à implanter une liaison *Ethernet* sans fil entre l'opérateur et le manipulateur mobile de telle sorte que les études précédentes sont applicables. L'opérateur peut également être situé à une distance supérieure, par exemple, dans un bureau. Le recours à une liaison spécialisée (pour la liaison bureau–chantier) couplée une liaison hertzienne (qui couvre l'ensemble du chantier) peut être envisagée.

Nous nous sommes appliqué ici à mettre en œuvre la phase de dépose finale. Cette phase consiste à téléopérer le bras manipulateur quand le véhicule est immobile. Il s'agit d'effectuer des mouvements dans l'espace cartésien. Nous n'avons pas envisagé de commande à retour d'effort car nous n'avions pas de dispositif maître capable de restituer des efforts. Ce type de commande sera à implémenter ultérieurement à l'aide, par exemple, d'un joystick à retour d'effort. Il faudra à ce moment diminuer nettement la période de transmission car ce type de système est plus gourmand en bande passante.

4.2 Description de la méthode de commande

4.2.1 Schéma global

1. A l'instant t, l'opérateur envoie une consigne de position dans l'espace opérationnel :
$$X_d(t) = {}^t[x_d\ y_d\ z_d\ \varphi_d\ \theta_d\ \psi_d](t)$$

2. Cette consigne est échantillonnée toutes les $T_t = 200\ ms$ ($X_d[k]$) et envoyée au *système distant* via le lien de transmission : c'est le signal $c(t)$.

3. Le lien de transmission la retarde d'un laps de temps noté $T_{r_{(A)}}(t)$, générant ainsi le signal $c_r(t) = c(t - T_{r_{(A)}}(t))$.

4. Elle parvient au *système distant* à l'instant $t + T_{r_{(A)}}(t)$ où elle est stockée par son régulateur pour ne ressortir qu'à l'instant $t + T_{total_{(A)}}$: c'est le signal $c_c(t)$.

5. Elle est transmise à la commande en position dans l'espace opérationnel. Vu de l'extérieur, la position de l'effecteur est régie par l'équation :

$$\ddot{\varepsilon}(t) + K_v.\dot{\varepsilon}(t) + K_p.\varepsilon(t) = 0 \qquad \text{où} \qquad \varepsilon(t) = X_d(t - T_{total_{(A)}}) - X(t - T_{total_{(A)}})$$

6. Le *système distant* échantillonne alors la position actuelle $X(t - T_{total_{(A)}})$ puis la transmet à la *base* sous la forme du signal $i(t)$.

7. Cette dernière reçoit ce signal à l'instant $t + T_{total_{(A)}} + T_{r_{(R)}}$.

8. Ce signal ressort alors du régulateur de la *base* à l'instant : $t + T_{total_{(A)}} + T_{total_{(R)}} = t + T_{total_{(A+R)}}$ sous la forme de $X(t - T_{total_{(A+R)}})$.

9. Il est alors transmis au prédicteur. Ce dernier, à partir des signaux $X_d(t)$ et $X(t - T_{total_{(A+R)}})$ se charge de générer les signaux $\hat{X}_d[k - K_{(A)}]$[1], $\hat{X}[k - K_{(A)}]$ et $\hat{X}[k]$.

Ainsi, chaque consigne X_d émise est exécutée après un laps de temps égal à $T_{total_{(A)}}$. Chaque mesure de l'état du *système distant* est réceptionnée après un laps de temps égal à $T_{total_{(R)}}$, soit $T_{total_{(A+R)}}$ après l'émission de la consigne correspondante.

Comme les signaux $c(t)$ et $i_c(t)$ sont périodiques, leurs versions échantillonnées à la période T_t sont notées respectivement $c[k] = X_d[k]$ et $i_c[k] = X[k - K_{(A+R)}]$ avec $K_{(A+R)} = \frac{T_{total_{(A+R)}}}{T_t}$. Etant donné que l'échantillonneur de l'émetteur et du régulateur utilisent la même horloge, le temps $T_{total_{(A+R)}}$ est forcément un multiple de T_t.

4.2.2 Commande en position par découplage non linéaire

Le principe de la commande par découplage non linéaire est présenté en annexe D. La modélisation du robot *PUMA* 560 est présentée en annexe E.

Nous nous plaçons dans le cas où $F_e = 0$, autrement dit, dans le cas où le poids de la charge est négligeable vis-à-vis du poids des diverses articulations du robot. Pour ne pas alourdir inutilement l'écriture, nous omettrons volontairement d'écrire systématiquement que les variables X, X_d, y, F, ... ainsi que leurs dérivées respectives dépendent du temps.

$$(E.15) \qquad \Rightarrow \qquad F = A_X(X).\ddot{X} + H_X(X, \dot{X}) \tag{4.1}$$

1. $K_{(A)} = \mathrm{E}\left(\frac{T_{total_{(A)}}}{T_t}\right)$

La relation entre F et X étant non linéaire et couplée, nous effectuons un bouclage tel que la nouvelle relation entre l'entrée et X soit linéaire et découplée : ainsi les propriétés du système seront celles des systèmes linéaires et le découplage permettra d'agir séparément sur chacune des directions de l'espace cartésien. Nous choisissons alors comme sortie $y = X$.

La nature physique des matrices A_q, N_t et J_m est telle que A est définie positive, donc inversible (A est définie par l'équation (E.8)). Il en résulte que $A_X = ({}^tJ)^{-1}.A.J^{-1}$ est aussi inversible.

En définissant le vecteur d'état $x_X = \begin{bmatrix} X \\ \dot{X} \end{bmatrix}$, nous obtenons la représentation d'état :

$$\begin{cases} \dot{x}_X &= \begin{bmatrix} \dot{X} \\ -A_X^{-1}(X).H_X(X,\dot{X}) \end{bmatrix} + \begin{bmatrix} 0 \\ A_X^{-1}(X) \end{bmatrix}.F \\ y &= X \end{cases} \tag{4.2}$$

La dimension de l'espace d'état est $n = 2 \times N = 2 \times 6 = 12$. Posons :

$$A_X^{-1} = [C_1\ C_2\ C_3\ C_4\ C_5\ C_6]$$

Par identification avec le système général (Σ) de l'annexe D :

$$f(x_X) = \begin{bmatrix} \dot{X} \\ -A_X^{-1}(X).H_X(X,\dot{X}) \end{bmatrix} \tag{4.3}$$

$$g_i(x_X) = \begin{bmatrix} 0 \\ C_i(X) \end{bmatrix} \tag{4.4}$$

$$u_i = F_i \tag{4.5}$$

$$h_i(x_X) = X_i \tag{4.6}$$

En appliquant la méthode proposée dans l'annexe D, nous obtenons la matrice de découplage :

$$\forall i,j \in [1;6], \qquad \Delta_{i,j} = \mathbb{L}_{g_j}(\mathbb{L}_f(h_i)) = C_{ij} \tag{4.7}$$

Comme nous avons fait l'hypothèse de nous placer en dehors des singularités, nous pouvons linéariser et découpler le système partout sauf sur les singularités. Il suffit de choisir $F = \alpha(X,\dot{X}) + \beta(X,\dot{X}).F_X$ avec :

$$\alpha(X,\dot{X}) = A_X(X).a(X,\dot{X}) \tag{4.8}$$

$$\beta(X,\dot{X}) = A_X(X).b(X,\dot{X}) \tag{4.9}$$

Le vecteur $a(X,\dot{X})$ est défini $\forall i \in \{1,2,\ldots,N\}$ par :

$$a_i(X,\dot{X}) = k_{p_i}.h_i(X,\dot{X}) + k_{v_i}.\mathbb{L}_f(h_i(X,\dot{X})) - \mathbb{L}_f^2(h_i(X,\dot{X})) \tag{4.10}$$

et $b(X,\dot{X}) = \mathbb{I}_6$, $\mathbb{I}_6$ étant la matrice identité de dimension (6×6).

Le bouclage solution s'écrit alors :

$$F = A_X(X).(k_p.X + k_v.\dot{X}) + H_X(X,\dot{X}) + A_X(X).F_X \tag{4.11}$$

avec $k_p = \mathrm{diag}(k_{p_i})$ et $k_v = \mathrm{diag}(k_{v_i})$

Par ailleurs, $\sum_{i=1}^{6}(\rho_i+1) = 2\times 6 = 12$: le système est complètement linéarisable. Le changement de coordonnées qui permet d'obtenir cette représentation linéaire serait :

$$\forall i \in [1;6] \qquad z_i = \begin{bmatrix} X_i & \dot{X}_i \end{bmatrix}$$

Il n'y a donc pas de changement d'état à effectuer et le système bouclé est la réunion des 6 sous-systèmes suivants :

$$\begin{cases} \dot{z}_i &= \begin{bmatrix} 0 & 1 \\ k_{p_i} & k_{v_i} \end{bmatrix}.z_i + \begin{bmatrix} 0 \\ 1 \end{bmatrix}.F_{X_i} \\ y_i &= \begin{bmatrix} 0 & 1 \end{bmatrix}.z_i \end{cases} \tag{4.12}$$

Ceci suppose naturellement que l'identification du modèle dynamique est sans erreur. Il est alors possible de réaliser un placement de pole à partir des constantes k_{p_i} et k_{v_i} qui doivent être ici strictement négatives pour que le système soit stable. Afin de donner à k_{p_i} et k_{v_i} leur rôle de gain (positif) en position et en vitesse, nous remplacerons k_{p_i} et k_{v_i} par $-k_{p_i}$ et $-k_{v_i}$ avec cette fois k_{p_i} et k_{v_i} strictement positifs. Nous obtenons finalement la relation globale :

$$\ddot{X} = -k_p.X - k_v.\dot{X} + F_X \tag{4.13}$$

Si nous voulons que le système suive une trajectoire de référence $X_d(t)$, il suffit de prendre comme loi de commande :

$$F_x = \ddot{X}_d + k_v.\dot{X}_d + k_p.X_d \tag{4.14}$$

Alors l'équation vérifiée par l'erreur de position $\varepsilon_X = X_d - X$ est :

$$\ddot{\varepsilon}_X + k_v.\dot{\varepsilon}_X + k_p.\varepsilon_X = 0 \tag{4.15}$$

Comme les coefficients k_{p_i} et k_{v_i} sont choisis positifs, il n'y a pas d'erreur en régime permanent :

$$\lim_{t\to\infty} \varepsilon_X = 0$$

De plus, l'équation d'évolution de l'erreur ne dépend ni du système, ni de la trajectoire à suivre $X_d(t)$, elle est parfaitement réglable à partir des matrices k_p et k_v pour réaliser les performances souhaitées.

Finalement, la loi de bouclage solution de ce problème dans le cas d'un suivi de trajectoire $X_d(t)$ est de la forme :

$$F = A_X(X).\left[\ddot{X}_d + k_v.(\dot{X}_d - \dot{X}) + k_p.(X_d - X)\right] + H_X(X,\dot{X}) \tag{4.16}$$

Le schéma correspondant à l'asservissement est présenté figure 4.1.

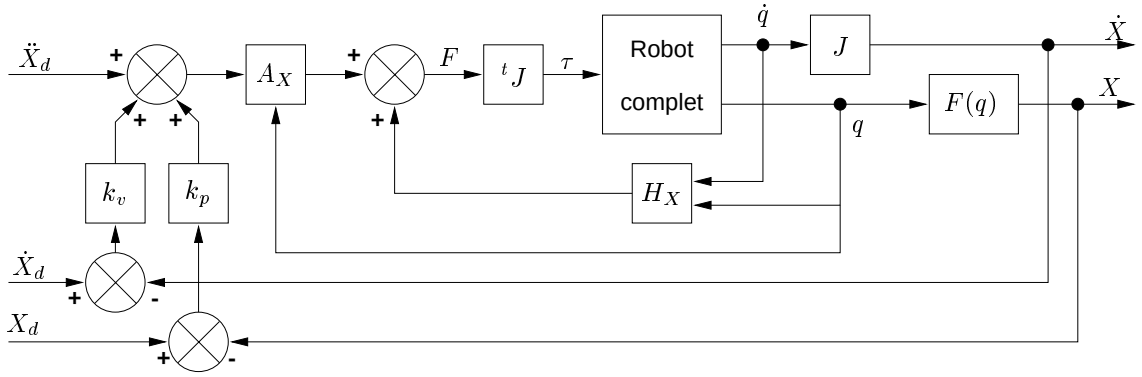


Figure 4.1 – *Commande dynamique en position dans l'espace opérationnel*

4.2.3 Détails des régulateurs

Nous implémentons ici une régulation dynamique des retards. Nous conservons le principe de la phase initiale d'audit présentée en §3.3.3, page 97 mais l'écoute du comportement des retards du réseau se poursuit jusqu'à la fin de la téléopération.

Prédiction des retards

Afin de prédire l'évolution globale des retards, nous avons fait appel à un filtre de Kalman discret (cf. annexe G) associé à un modèle triple–intégrateur et à une

fonction d'autocorrelation $\phi[k]$ donnée à l'équation (4.17). Cette technique est détaillée dans [BOZ 83]. Il s'agit initialement de prédire la trajectoire d'un mobile (un avion de chasse) pour anticiper ses mouvements et proposer un angle de tir conséquent. Le rapport avec l'évolution des paramètres des retards est assez éloigné mais l'application de ce modèle reste très générale.

$$\phi[k] = \sigma_Q^2 . \mathrm{e}^{-a.|k|.T_t} \tag{4.17}$$

Ce modèle discret possède une matrice de transition A :

$$A = \begin{bmatrix} 1 & T_t & \frac{T_t^2}{2} \\ 0 & 1 & T_t \\ 0 & 0 & 1 \end{bmatrix}$$

et une matrice de covariance Q :

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_Q^2 \end{bmatrix}$$

Nous avons arbitrairement fixé σ_Q de telle sorte qu'il corresponde à un bruit d'amplitude 1 ms, c'est-à-dire la résolution de la mesure des retards. Nous obtenons alors les matrices suivantes:

$$B = [0]$$

$$R = \sigma_Q^2$$

$$H = [1\,0\,0]$$

$$P_0 = \mathbb{I}_3 . \sigma_Q^2$$

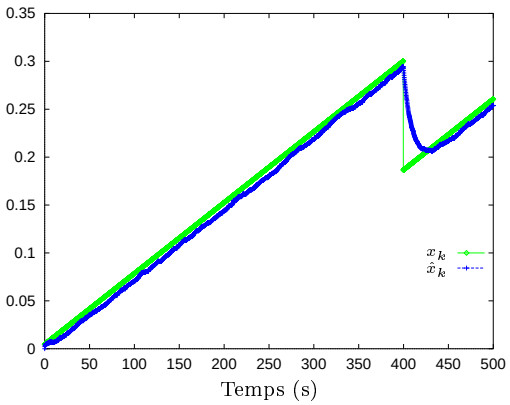
Ce système est dédoublé et utilisé pour l'estimation, d'une part, de la moyenne $\overline{T_{rés.(A+R)}}$ et, d'autre part, de l'écart-type $\sigma_{T_{rés.(A+R)}}$ des retards de transmissions. La mesure de la moyenne peut être obtenue par filtrage passe-bas (avec une constante de temps de l'ordre de 10 s par exemple) ou par le calcul d'une moyenne à horizon fini (avec un horizon de 10 s par exemple). En ce qui concerne l'écart-type, un calcul d'écart-type à horizon fini suffit.

En reprenant le système d'équation décrit en annexe G avec ces matrices, nous obtenons une estimation de ces deux signaux. Pour obtenir une prédiction plus lointaine,

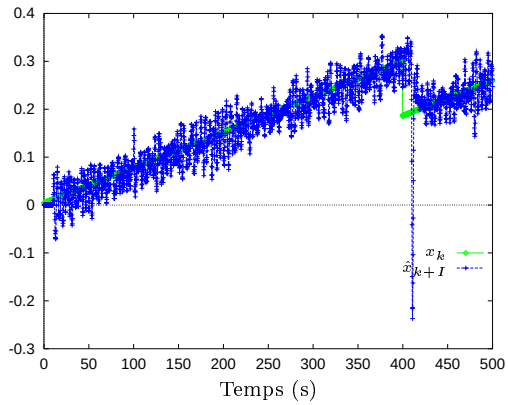
il suffit d'appliquer $\hat{x}_{k+I} = A^I.\hat{x}_k$, $\hat{x}_{k+I}$ pouvant représenter la moyenne ou l'écart-type des retards :

Une solution pour obtenir une prédiction encore plus avancée dans le temps est de sous-échantillonner les retards — par exemple, en ne mesurant qu'une valeur sur deux.

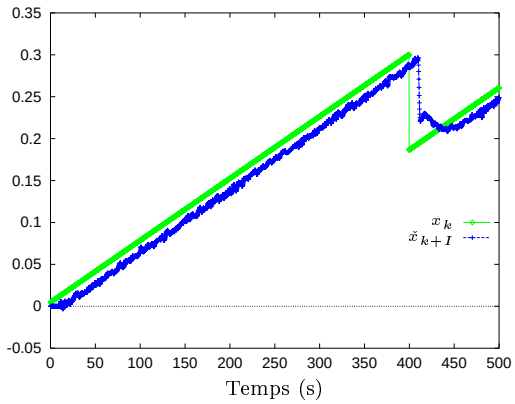
N'ayant pas de mesure de l'évolution de la moyenne de retards de transmission en notre possession, nous avons imaginé une moyenne variant principalement linéairement avec quelques discontinuités. La sous-figure 4.2(a) représente cette moyenne ainsi que l'état du filtre de KALMAN en fonction du temps. La sous-figure 4.2(b) présente, elle, la prédiction de l'évolution de la moyenne avec une anticipation de 10 s. Cette prédiction étant plutôt bruitée et possédant quelques valeurs locales très éloignées de l'allure globale, nous avons appliqué un filtre passe-bas de constante de temps de 10 s à ce signal. Nous obtenons alors le signal $\check{x}_{k+I}$ représenté à la sous-figure 4.2(c).



(a) Moyennes



(b) Prédiction à 10 s



(c) Prédiction à 10 s lissée

Figure 4.2 – *Résultats de simulation pour la prédiction de la moyenne des retards*

Evolution des paramètres de transmissions

La prédiction de la variance et de la moyenne du retard ramenées à la période d'échantillonnage T_t nous permet de décider d'un éventuel changement de taille des files d'attente et de la période d'échantillonnage.

Supposons que les régulateurs soient paramétrés à un $n_{désiré}$ donné. Si la variance augmente de $n\%$, nous pouvons nous attendre à ce que l'amplitude des retards augmente d'autant. Le test consiste donc à prédire un éventuel vidage des files des régulateurs. De même, si la variance des retards diminue d'autant, il peut être plus intéressant de diminuer la taille des files.

Si c'est le cas, un signal est émis par le bloc de mesure/prédiction des retards pour entamer une procédure de modification des paramètres de communication. Dès réception de ce signal par le *système distant*, le manipulateur est placé en mode « pause », c'est à dire qu'il ne tient plus compte des éventuelles consignes qu'il reçoit et reste immobile.

A ce moment, les régulateurs sont vidés d'éventuelles consignes parasites. Un calcul des nouveaux paramètres de communication (période T_t puis $n_{désiré}$ des régulateurs) a alors lieu en tenant compte des j dernières mesures de retard et des j prédictions des retards supposés arriver. Ainsi nous pondérons à part égale la prédiction opérée par le filtre de KALMAN et les retards réels les plus récents.

En ce qui concerne la période de transmissions T_t, nous nous proposons de la déterminer par la relation (4.18), sachant que nous sommes limités technologiquement par la vitesse des ordinateurs sur lesquels sont exécutées les applications de téléopération BASE et MANIMOB (éventuellement également RELAIS), par la rapidité des divers algorithmes à exécuter à chaque pas d'échantillonnage ainsi que par la bande passante du médium de transmissions.

$$T_t = \max(\frac{\Delta T_{rés._{(A+R)}}}{10}, 50\,\text{ms}) \tag{4.18}$$

Le cas $T_t = \frac{\Delta T_{rés._{(A+R)}}}{10}$ est identique à celui de l'expérimentation avec Rutgers commentée en §3.3.5, page 126 où nous avons obtenu une résolution du temps global aller–retour $T_{total_{(A+R)}}$ dix fois inférieure au retard moyen. Ce rapport nous permet ensuite de régler plus finement $n_{désiré}$ en fonction de l'amplitude des retards. Le cas $T_t = 50$ ms est une limite inférieure dépendant des divers critères cités ci-dessus.

Le choix de $n_{désiré}$ est effectué selon la méthode exposée en §3.3.3, page 99 en tenant compte de la nouvelle période de transmissions T_t.

Ensuite, les transmissions redémarrent immédiatement avec ces nouveaux paramètres. Nous nous retrouvons dans un mode transitoire similaire à celui présenté en §3.3.3, page 97, suivi du nouveau mode permanent.

4.2.4 Détails du prédicteur

Architecture globale

Le bloc de prédiction reprend le principe de l'architecture proposée en §3.4, page 131. Cependant, nous avons opté pour une méthode d'identification dynamique du système distant fondée sur l'algorithme des moindres carrés récursifs d'ordre N (cf. annexe F).

Ainsi, le schéma de la figure 3.61, page 133 est repris ici en l'adaptant à nos besoins en figure 4.3.

Les deux entrées $x_d[k]$ et $x_{rr}[k]$ de la figure 3.61 deviennent ici respectivement $X_d[k]$ et $X[k - K_{(A+R)}]$. Il s'agit ici de vecteurs de 6 composantes de position/orientation de l'effecteur du bras manipulateur. En fait, nous avons ici 6 schémas de la figure 3.61 en parallèle et indépendants puisque la commande du bras découple les 6 composantes de $X[k]$.

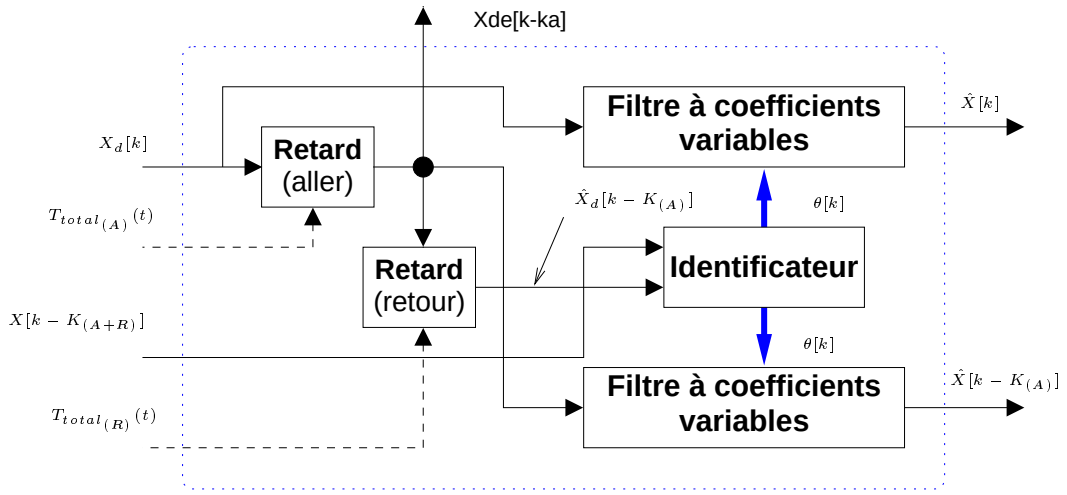


Figure 4.3 – *Détails du bloc de prédiction*

Nous avons fait appel à deux blocs « filtre à coefficients variables » afin d'obtenir d'une part une estimation de l'état du robot en temps réel $\hat{X}[k - K_{(A)}]$ et d'autre part une prédiction de son état dès que l'opérateur envoie ses consignes $\hat{X}[k]$.

D'autre part, ce bloc exporte également le signal de consignes lors de sa prise en compte par le robot $\hat{X}_d[k - K_{(A)}]$.

Bloc identificateur

En §3.4.3, page 133, nous avons fait appel à un filtre adaptatif à gradient. Ici, nous avons préféré utiliser une méthode d'identification fondée sur l'algorithme des moindres carrés récursifs (cf. annexe F).

Le système à identifier est le bras manipulateur commandé dans l'espace opérationnel par la commande linéarisante et découplante proposée précédemment. Vu de l'extérieur, cette commande possède une relation entrées/sorties sous la forme de l'équation (4.15). Le bloc identificateur doit donc identifier les coefficients k_{v_i} et k_{p_i} pour chaque composante i du vecteur de position/orientation. Si nous considérons que la consigne de position varie lentement vis-à-vis du robot, alors nous pouvons poser :

$$\ddot{X}_d(t) \approx 0 \qquad \text{et} \qquad \dot{X}_d(t) \approx 0 \tag{4.19}$$

Ainsi, pour chaque composante i ($i \in [1;6]$), le système a une fonction de transfert de la forme de :

$$H_i(p) = \frac{1}{p^2 + k_{v_i}.p + k_{p_i}} \tag{4.20}$$

Le système se comporte donc globalement comme un filtre proportionnel–dérivé.

Les consignes et les retours d'informations ayant lieu sous forme échantillonnée, appliquons la transformation bilinéaire pour obtenir un modèle échantillonné du système :

$$p = \frac{2(1-z^{-1})}{T_t(1+z^{-1})} \qquad \Rightarrow \qquad H_i(z) = \frac{b_{0_i} + b_{1_i}.z^{-1} + b_{2_i}.z^{-2}}{1 + a_{1_i}z^{-1} + a_{2_i}z^{-2}} \tag{4.21}$$

avec :

$$b_{0_i} = T_t^2./\gamma_i \tag{4.22}$$

$$b_{1_i} = 2.b_{0_i} \tag{4.23}$$

$$b_{2_i} = b_{0_i} \tag{4.24}$$

$$a_{1_i} = \frac{-8 + 2.k_{v_i} + 2.k_{p_i}.T_t^2}{\gamma_i} \tag{4.25}$$

$$a_{2_i} = \frac{4 + k_{p_i}.T_t^2}{\gamma_i} \tag{4.26}$$

où

$$\gamma_i = 4 + 2.k_{v_i} + k_{p_i}.T_t^2 \tag{4.27}$$

Ainsi, les vecteurs $\theta_i[k]$, $Z_i[k]$ et la matrice $H_i[k]$ sont :

$$\begin{cases} \theta_i[k] & = {}^t\begin{bmatrix} a_{1_i}[k] & a_{2_i}[k] & b_{0_i}[k] & b_{1_i}[k] & b_{2_i}[k] \end{bmatrix} \\ Z_i[k] & = {}^t\begin{bmatrix} y[k-N] & y[k-N+1] & \cdots & y[k] \end{bmatrix} \\ H_i[k] & = \begin{bmatrix} y[k-N-1] & y[k-N-2] & u[k-N] & u[k-N-1] & u[k-N-2] \\ y[k-N] & y[k-N-1] & u[k-N+1] & u[k-N] & u[k-N-1] \\ \vdots & & & & \vdots \\ y[k-1] & y[k-2] & u[k] & u[k-1] & u[k-2] \end{bmatrix} \end{cases}$$

$$\text{avec} \begin{cases} \dim(\theta_i[k]) & = (5 \times 1) \\ \dim(Z_i[k]) & = ((N+1) \times 1) \\ \dim(H_i[k]) & = ((N+1) \times 5) \end{cases}$$

Bloc filtre à coefficients variables

Le filtre de ce bloc reprend donc la fonction de transfert échantillonnée (4.21) avec les coefficients $b_{0_i}[k]$, $b_{1_i}[k]$, $b_{2_i}[k]$, $a_{1_i}[k]$ et $a_{2_i}[k]$ calculés en temps réel.

4.3 Conclusion

L'architecture de la commande à implanter étant maintenant entièrement définie dans le cadre de la phase de téléopération du bras manipulateur, il nous reste concrètement à terminer de programmer ce schéma dans un premier temps en simulation puis sur notre plate-forme d'expérimentation.

Enfin, l'ensemble de la manipulation proposée en introduction de ce chapitre sera à expérimenter en extérieur. Il est d'ores et déjà prévu d'améliorer cette expérimentation en implémentant un contrôle de commutation de commandes en cours de développement au *LIRMM*, par Andreu et Carbou au niveau de la commande du *PUMA*. L'intérêt est de pouvoir passer d'une commande à une autre sans discontinuité ; par exemple d'une commande d'impédance à une génération de trajectoire.

Conclusion générale et perspectives

Les développements présentés dans ce document font partie des fondations permettant de réaliser des expériences de téléopération à longue distance.

Nous nous sommes attaché à mettre en œuvre une structure de commande bas-niveau de telle sorte que nous puissions ultérieurement développer des schémas de commande plus évolués s'appuyant sur ce travail.

Nous avons alors cherché à identifier et modéliser les différents composants entrant dans un schéma de téléopération simple. Ainsi, nous avons analysé le comportement du médium de télécommunications dans le cadre d'une application de téléopération. Dans notre cas, il s'agissait d'un réseau informatique comportant les couches de protocoles *TCP/IP*. Nous avons étudié le cas d'une téléopération à faible distance à travers un réseau *Ethernet* local (LAN[2]) ainsi que le cas d'une téléopération à longue distance (transatlantique) à travers un réseau plus étendu (WAN[3]), ici l'*Internet*.

Cette étude suppose que le *système distant* est correctement asservi localement et est modélisable sous forme d'un filtre échantillonné passe-bas d'ordre 2. A partir de cette hypothèse peu réductrice, nous avons proposé, dans un premier temps, un moyen simple de réguler les variations des retards de transmission. Cette régulation permet de rendre l'ensemble de la chaîne de téléopération isochrone et de synchroniser la *base* et le *système distant*. Cependant, nous n'avons pas pu valider expérimentalement le bon fonctionnement de cette régulation, faute d'un environnement logiciel temps–réel au niveau du *PC* superviseur. Au chapitre 4, nous avons toutefois proposé une méthode pour rendre la période de transmission et les paramètres des régulateurs variables par paliers. En anticipant sur une longue période les variations des caractéristiques des retards de transmissions, il est alors possible d'améliorer les performances de cette structure et de limiter les interruptions momentanées de transmission.

La synchronisation entre la *base* et le *système distant* a pour intérêt de permettre de générer une estimation de l'état du *système distant* en temps réel ainsi qu'une prédiction de son comportement dès l'émission des consignes. Nous avons ainsi proposé une structure d'estimation/prédiction indépendante du modèle du *système distant* et de la technique employée pour son observation/identification. Le principe a été validé

2. *Local Area Network*
3. *Wide Area Network*

en simulation avec un filtre adaptatif à gradient. Au chapitre 4, nous avons préconisé l'utilisation d'un filtre fondé sur les moindres carrés récursifs qui demande plus de calculs mais qui est plus efficace.

D'autres briques de fondation sont en cours de développement :

- passage de la structure de commande actuelle temps-réel mono-tâche (*DSP*) en une structure temps-réel multi-tâches,
- automatisation du choix (séquentiel) des algorithmes de commande en fonction de la tâche (travaux en cours de *David Andreu* et *Jean-Damien Carbou* au *LIRMM*),
- intégration des algorithmes de déplacement d'un manipulateur mobile développés notamment dans [DAU 99],
- intégration des algorithmes de manipulation d'objets fragiles [FRA 95]), ...

Les applications de ce travail sont nombreuses. Nous pouvons notamment citer la surveillance de zone contaminée avec prélèvement et analyse de matériaux. Pour cette application, nous pourrions envisager deux modes de travail : un mode local où l'opérateur est à quelques mètres du téléopérateur et un mode de téléopération à distance. Les transmissions locales seraient alors réalisées via des liaisons de type BLUETOOTH[4] et les liaisons à longue distance via une liaison de type *GPRS*[5]. Ces technologies de transmission sans fil sont actuellement en pleine émergence et seront peut-être un jour un nouveau terrain d'applications pour la téléopération.

4. http://www.bluetooth.com/
5. http://www.wirelessdevnet.com/articles/apr2000/gprs.html

Nous abordons maintenant des améliorations auxquelles nous avons pensé ainsi que des idées à développer afin de compléter ce travail. Dans un souci de clarté, nous avons repris la structure de ce rapport.

Modélisations

Caractéristiques des transmissions

Nous nous sommes intéressé à l'étude de deux protocoles *IP*, *ICMP* (qui n'est pas directement dédié à la transmission de données) et *TCP*. Pour compléter cette étude, il serait peut-être intéressant d'étudier également les caractéristiques du protocole *UDP*. Cela permettrait de distinguer le temps de travail (désassemblage/ré-assemblage dans l'ordre des trames, acquittements) dû au protocole *TCP*, qui n'existe pas avec *UDP* et que nous pourrions alléger pour certaines applications moins gourmandes en terme de fiabilité des transmissions.

D'autre part, pour compléter ces mesures, il serait également intéressant de jouer sur le *TOS*[6] pour spécifier le caractère urgent d'une trame par rapport aux autres et déterminer si cela peut jouer sur les temps et la régularité de la transmission.

A terme, nous souhaitons proposer un protocole de niveau application dédié au transmissions en temps réel dans un but de commande de robots (en local comme à longue distance).

Environnement de simulation

L'environnement de simulation développé est encore incomplet. Il serait utile de pouvoir faire varier dans le temps les caractéristiques des retards imposés par les blocs réseau.

Etudes

Stabilité

L'étude menée dans ce rapport est une première approche. Il serait enrichissant de prolonger cette étude en fournissant des outils permettant de déterminer la stabilité de systèmes de dimension et de degré quelconques, que ce soit par une méthode approchée ou non. Pour cela, nous avons proposé une piste consistant à étudier les conditions de stabilité en présence de retards variables en décomposant ceux-ci en séries de Fourrier.

6. *Type of Service*

Régulation

Le calcul de la taille des files des régulateurs a été fondé sur l'observation de la moyenne, de l'écart-type et de l'amplitude des retards de transmissions. Peut-être serait-il intéressant de se pencher sur l'étude des modes [STR 00] de ces retards pour obtenir un dimensionnement plus précis.

Nous avons pu également constater pendant les divers essais que les conditions initiales de remplissage des files des régulateurs influençaient la tenue des régulateurs en régime permanent de manière non négligeable. Nous devrons envisager d'étudier plus précisément cet aspect ultérieurement. Il est également envisageable de rendre les régulateurs dynamiques de manière à adapter leur $n_{désiré}$ aux évolutions lentes des paramètres de transmissions : moyenne et écart-type (par exemple).

Nous avons constaté que le fait de ne pouvoir synchroniser de façon assez précise deux hôtes rendait la régulation des retards un peu plus ardue. Peut-être pourrions-nous envisager l'utilisation d'une horloge externe à l'ordinateur, obtenue à partir d'un récepteur *GPS*. En effet, les 24 satellites en orbite utilisés pour le positionnement possèdent des horloges très précises et régulièrement corrigées qui permettent de régler l'horloge de tout récepteur *GPS* avec une résolution de 500 ns, suffisante pour nos études.

Expérimentations

Compte tenu des résultats obtenus au cours de ces travaux, il nous faudra migrer le système d'exploitation du *PC* superviseur vers un *vrai* O.S. temps–réel tel que QNX ou LINUX–RTCette opération est indispensable pour la suite de cette étude.

Nous pensons établir un réseau local de type *Ethernet* autour d'un commutateur à priorités. Un *PC* serait dédié aux asservissements bas-niveau, au moyen d'un système d'exploitation temps–réel. Une second *PC* serait, lui, dédié à la supervision des transmissions. D'autres unités pourraient venir s'ajouter simplement pour assurer les fonctions de retour vidéo, localisation *GPS*, ...

Application

Ce travail est complété par la mise en place d'un ensemble de visualisation à distance. En effet, une caméra est actuellement placée sur le bras manipulateur. Elle permet de filmer le poignet du manipulateur dans son environnement. A terme, nous ajouterons une seconde caméra à l'avant du véhicule qui sera destinée à la conduite du véhicule. Ces caméras seront reliées à un second *PC* qui émettra ces images sur le réseau local *Ethernet* via une carte *Ethernet* radio. Ainsi, n'importe quelle machine reliée à l'*Internet* est susceptible d'afficher les images filmées par ces caméras. Cette visualisation à distance est indispensable dans le cadre de la téléopération.

Un travail futur, sortant probablement du domaine de cette étude, consistera à synchroniser les images reçues avec les données émises par le logiciel *serveur* du manipulateur mobile au moyen des améliorations développées au paragraphe 3.3.

Bibliographie

[AND 88] R. J. Anderson et M. W. Spong, « *Bilateral Control of Teleoperations with Time Delay* », IEEE Conf. on Decision and Control, 1988, pp. 167–173.

[AND 89] R. J. Anderson et M. W. Spong, « *Bilateral Control of Teleoperations with Time Delay* », IEEE Trans. on Automatic Control, May 1989, Vol. 34, No. 5, pp. 494–501.

[AND2 89] R. J. Anderson et M. W. Spong, « *Asymptotic Stability for Force Reflecting Teleoperators with Time Delay* », IEEE Trans. on Automatic Control, May 1989, Vol. 34, No. 5, pp. 494–501.

[AND 92] R. J. Anderson et M. W. Spong, « *Asymptotic Stability for Force Reflecting Teleoperators with Time Delay* », The Intl. Journal of Robotics Research, April 1992, Vol. 11, Nb. 2, pp. 135–149.

[AND 96] R. J. Anderson, « *Building a Modular Control System using Passivity and Scattering Theory* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'96) , Albuquerque, Nouveau Mexique, USA, April 1997, pp. pp. 698–705.

[AND 99] N. Ando, J- H. Lee et H. Hashimoto, « *A Study on Influence of Time Delay in Teleoperation* », Proc. of the IEEE/ASME Intl. Conf. on Advanced Intelligent Mechatronics, September 1999, pp. 317–322.

[ARM 86] B. Armstrong, O. Khatib et J. Burdick, « *The Explicit Dynamic Model and Inertia Parameters of the* PUMA *560 arm* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'86) , San Francisco, California, USA, April 1986, pp. pp. 510–518.

[BAC 92] P. G. Backes, « *Multi-Sensor based Impedance Control for Task Execution* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'92) , Nice, France, May 12–14, 1992, pp. 1245–1250.

[BAC 00] P. G. Backes, K S Tuo, J. S. Norris, G. K. Tharp, J. T. Slostad, R. G. Bonitz et K. S. Ali, « *Internet-Based Operations for the Mars Polar Lander Mission* », Proc. of the IEEE/ASME Intl. Conf. on Advanced Intelligent Mechatronics, September 1999, pp. 317–322.

[BAL 98] J. Baldwin, A. Basu et H. Zhang, « *Predictive Windows for Delay Compensation in Telepresence Applications* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'98) , Leuven, Belgium, May 1998, pp. 135–149.

[BEJ 90] A. K. Bejczy, W. S. Kim et S. C. Venema, « *The Phantom Robot: Predictive displays for Teleoperation with Time Delays* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'90) , Cincinnati, Ohio, USA, May 13–18, 1990, pp. pp. 546–551.

[BEL 89] M. Bellanger, « *Analyse des signaux et Filtrage numérique adaptatif* », Editions Masson, 1989.

[BOZ 83] C. A. Bozzo, « *Le filtrage optimal et ses applications aux problèmes de poursuite – tome III* », Editions Tec & Doc, 1983.

[CAB 2000] C. Caby, A. Crosnier et P. Dauchez, « *Interaction model for programming teleoperated missions* », 31st International Symposium on Robotics (ISR'2000), Montreal, Canada, May 14-17, pp. 214–219.

[CHA 96] P. H. Chang et J. W. Lee, « *A Model Reference Observer for Time-Delay Control and its applications to Robot Trajectory Control* », IEEE Transactions on Control Systems Technology, vol. 4, no. 1, January 1996.

[CON 90] L. Conway, R. A. Volz et M. W. Walker, « *Teleautonomous systems: projecting and coordinating intelligent action at a distance* », IEEE Transactions on Robotics and Automation, vol. 2, no. 6, pp. 146–158, April 1990.

[DAM 98] M. Dambrine, F. Gouaisbaut, W. Perruquetti et J. P. Richard, « *Robustness of sliding mode control under delays effects : a case study* », Proc. of the 2nd IMACS-IEEE Computational Engineering in Systems Applications Conf. (CESA'98), Tunisie, Avril 1998, pp. 817–821.

[DAU 99] P. Dauchez , P. Fraisse , A. Lelevé et F. Pierrot , « *Experimental Results on Motion Generation and Control of a Non-Holonomic Mobile Manipulator* », Proc. of the Intl. Symp. on Robotics (ISR'99), Tokyo, Japan, october 1999.

[DEL 93] P. De Larminat, « *Automatique — commande de systèmes linéaires* », Editions Hermes, Paris.

[DEP 97] P. Depasquale, J. Lewis et M. Stein, « *A Java Interface for asserting Interactive Telerobotic Control* », Proc. of the SPIE Conf. on Intelligent Systems, Pittsburgh, Pennsylvania, USA, October 14–15, 1997, pp. 159–169.

[FER 65] W. R. Ferrell, « *Remote manipulation with Transmission Delay* », IEEE Trans. on Human Factors in Electronics, September 1965, vol. 6, pp. 24–32.

[FER 67] W. R. Ferrell et T. B. Sheridan, « *Supervisory Control of Remote Manipulation* », IEEE Spectrum, October 1967, vol. 4, No 10, pp. 81–88.

[FOU 91] J. Founda, « *Teleprogramming: Towards Delay-Invariant Remote Manipulation* », PhD Thesis, University of Pennsylvania, Computer Science Department, 1991.

[FRA 95] P. Fraisse , P. Dauchez , F. Pierrot et L. Cellier, « *Mobile Manipulation of a Fragile Object* », 4th International Symposium on Experimental Robotics (ISER'95), Stanford, California, USA, June 30-July 2, 1995.

[GOE 64] R. Goertz, « *Manipulator system development at ANL* », Proc. of the 12th Remote Systems Technology Conf., Argonne National Laboratory, 1964, pp. 117–136.

[GOL 95] K. Goldberg, M. Mascha, S. Gentner N. Rothenberg, C. Sutter et J. Wiegley, « *Desktop Teleoperation via the World Wide Web* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'95) , Nagoya, Aichi, Japan, May 21–27, 1995, pp. 654–659.

[GOL 00] K. Goldberg, B. Chen, R. Solomon, S. Bui, B. Farzin, J. Heitler, D. Poon et G. Smith, « *Collaborative Teleoperation via the* Internet », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'2000) , Nagoya, Aichi, Japan, May 21–27, 1995, pp. 2019–2024.

[GOU 99] A. Gourdon, P. Vieyres, P. Poignet M. Szpieg et P. Arbeille, « *A Telescanning Robotic System using Satellite Communication* », European Medical and Biological Engineering Conf., Novembre 1999, Vienne, Autriche.

[GRA 00] S. Grange, T. Fong et C. Baur, « *Effective Vehicle Teleoperation on the World Wide Web* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'2000) , Nagoya, Aichi, Japan, May 21–27, 1995, pp. 2007–2012.

[HIR 80] K. Hirai et Y. Satoh, « *Stability of a System with Variable Time Delay* », IEEE Transactions on Automatic Control, June 1980, No.3, vol. AC-25, pp. 552–554.

[HIR 94] G. Hirzinger, K. Landzettel et Ch. Fagerer, « *Telerobotics with large time delays — the ROTEX experience* », Proc. of the Intl. Conf. on Intelligent Robots and Systems (IROS'94), Munich, Germany, 1994, pp. 571–578.

[IND 97] Indrawanto, J. Swevers et H. Van Brussel, « *Robust Decentralized Adaptive Control of Robot Manipulators* », Proc. of the 5th IFAC Symp. On Robot Control (SY.RO.CO.), Nantes, France, September 5–7 1997, pp. 221–226.

[JAM 96] K. Jamsa et K. Cope, « *Programmation Internet en C et C++* », Editions International Thomson Publishing, 1996.

[KAL 60] R.E. Kalman, « *A New Approach to Linear Filtering and Prediction Problems* », Transaction of the ASME — Journal of Basic Engineering, March 1960, pp. 33–45.

[KH1 97] A. Khedar , C. Tzafestas, P. Coiffet, T. Kotoku, S. Kawabata, K. Iwamoto, K. Tanie, I. Mazon, C. Laugier et R. Chellali, « *Parallel Multi-Robots Long Distance Teleoperation* », Proc. of the Conf. on Advanced Robotics (ICAR'97), Monterey CA, USA, July 7–9 1997, pp. 1007–1012.

[KH2 97] A. Khedar , C. Tzafestas et P. Coiffet, « *The Hidden Robot Concept — High Level Abstraction Teleoperation* », IEEE/RSJ Intl. Conf. on Intelligent Robotics and Systems, Grenoble, France, Sept. 7–11, 1997, vol. 3 pp. 1818–1824.

[KH3 97] A. Khedar , P. Coiffet, T. Kotoku et K. Tanie, « *Multi-Robots Teleoperation - Analysis and Prognosis* », 6th IEEE Intl. Workshop on Robot and Human Communication (ROMAN'97), Sendai, Japan, Sept. 29 – Oct. 1, 1997, pp. 166–171.

[KIM 92] W. S. Kim, B. Hannaford et A. K. Bejczy, « *Force Reflection and Shared Compliant Control in Operating Telemanipulators with Time Delay* », IEEE Transactions on Robotics and Automation, vol. 2, no. 8, pp. 176–185, April 1992.

[KOS 96] K. Kosuge, H. Murayama et K. Takeo, « *Bilateral Feedback Control of Telemanipulators via Computer Network* », Proc. of the Intl. Conf. on Intelligent Robots and Systems (IROS'96), Osaka, Japon, November 4-8, 1996, pp. 1380–1385.

[KOS 97] K. Kosuge et H. Murayama, « *Bilateral Feedback Control of Telemanipulator via Computer Network in Discrete Time Domain* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'97) , Albuquerque, Nouveau Mexique, USA, April 1997, pp. 2219–2224.

[LEL 98] A. Lelevé , P. Fraisse , A. Crosnier , P. Dauchez et F. Pierrot , « *Towards Virtual Control of Mobile Manipulators* », Proc. of the 3rd World Automation Congress (WAC'98), Anchorage, USA, may 1998.

[LEL1 99] A. Lelevé , P. Fraisse , P. Dauchez et F. Pierrot , « *Modeling and Simulation of Robotic Tasks Teleoperated through the* Internet », Proc. of the International Conf. on Advanced Intelligent Mechatronics (IEEE/ASME'99), Atlanta, USA, September 1999.

[LEL2 99] A. Lelevé , P. Fraisse , P. Dauchez et F. Pierrot , « *Teleoperation through the* Internet*: Experimental Results with a Complex Manipulator* », Proc. of the Intl. Symp. on Robotics (ISR'99), Tokyo, Japan, october 1999.

[LEL 00] A. Lelevé , P. Dauchez , P. Fraisse et F. Pierrot , « *An Enhanced Mobile Manipulator* », Proc. of the 4th World Automation Congress (WAC 2K), Maui, Hawaii, USA, june 2000.

[LEU 92] G. M. H. Leung et B. A. Francis, « *Bilateral Control of teleoperators with time delay through a digital communication channel* », Proc. of the 30th Annual Allerton Conf. on Communication, Control and Computing, September, 1997, pp. 692–701.

[LEU 97] G. M. H. Leung, B. A. Francis et J. Apkarian, « *Bilateral Controller for Teleoperators with Time Delay via μ-Synthesis* », IEEE Transactions on Robotics and Automation, vol. 11, no. 1, pp. 105–116, 1997.

[LUO 97] R. C. Luo et T. M. Chen, « *Remote supervisory control of a Sensor Based Mobile Robot via Internet* », Proc. of the Intl. Conf. on Intelligent Robots and Systems (IROS'97), Grenoble, France, 1997, pp. 1163–1168.

[MAD 96] D. Maddalena, W. Prendin et A. Terribile, « *Supervisory Control Telerobotics reaches the Underwater Work Site* », Proc. of the 6th IARP Workshop on Underwater Robotics, Toulon, France, March 27–29 1996.

[MIT 95] M. Mitsuishi, T. Hori et T. Nagao, « *Predictive, Augmented and Transformed Information Display for Time Delay Compensation in Tele-Handling/Machining* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'95) , Nagoya, Aichi, Japan, May 21–27, 1995, pp. 45–52.

[NIC 97] S. I. Niculescu, « *Systèmes à retard - Aspects qualitatifs sur la stabilité et la stabilisation* », Editions Diderot, Collection Arts & Science, 1997.

[NIE 90] G. Niemeyer et J-j. E. Slotine, « *Stable Adaptive Teleoperation* », Proc. of the 1990 American Control Conf., May 1990, vol. 2, pp. 1186–1191.

[NIE1 97] G. Niemeyer et J-j. E. Slotine, « *Using Wave Variables for System Analysis and Robot Control* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'97) , Albuquerque, Nouveau Mexique, USA, April 1997, pp. 2212–2218.

[NIE2 97] G. Niemeyer et J-j. E. Slotine, « *Designing Force Reflecting Teleoperators with Large Time Delays to Appear as Virtual Tools* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'97) , Albuquerque, Nouveau Mexique, USA, April 1997, pp. 2212–2218.

[OBO 97] R. Oboe et P. Fiorini, « *Issues on Internet-based Teleoperation* », Proc. of the 5th IFAC Symp. On Robot Control (SY.RO.CO.), Nantes, France, September 3–5 1997, pp. 611–617.

[OTS 95] M. Otsuka, N. Matsumoto, T. Idogaki, K. Koguse et T. Itoh, « *Bilateral Telemanipulator System with Communication Time Delay Based on Force-Sum-Driven Virtual Internal Models* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'95) , Nagoya, Aichi, Japan, May 21–27, 1995, pp. 344–350.

[PAU 92] R. P. Paul, T. Lindsay, C. Sayers et M. Stein, « *Time-delay insensitive virtual-force reflecting teleoperation* », Proc. in Artificial Intelligence, Robotics and Automation in Space, Toulouse, France, Septembre 1992, pp. 55–67.

[PAU 93] R. P. Paul, C. Sayers et M. Stein, « *The Theory of Teleprogramming* », Journal of Robotics Society of Japan, nb. 11, pp. 14–19.

[PAU 96] E. Paulos et J. Canny, « *Delivering Real Reality to the World Wide Web via Telerobotics* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'1996) , Minneapolis, Minnesota, USA, April 1996, pp. 1694–1699.

[PEÑ 00] L. F. Peñín, K. Matsumoto et S. Wakabayashi, « *Force Reflection for Time-Delayed Teleoperation of Space Robots* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'2000) , San Francisco, California, USA, April 2000, pp. 3210–3125.

[PER 96] G. Perec, « *Experimental Demonstration of the tomatotopic organization in the soprano* », Proc. of the 6th JFF Symp., Acqueville, France, August 14–16 1996, pp. 105–110.

[POO 95] P. K. Pook et D. H. Ballard, « *Remote Teleassistance* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'95) , Nagoya, Aichi, Japan, May 21–27, 1995, pp. 944–949.

[RAJ 89] G. J. Raju, G. C. Verghese et T. B. Sheridan, « *Design Issues in 2-port Network Models of Bilateral Remote Manipulation* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'89) , Scottsdale, Arizona, USA, May 14–19, 1989, vol. 3, pp. 1316–1321.

[RAS 96] A. Rastogi et P. Migram, « *Augmented Telerobotic Control: a visual interface for unstructured environments* », University of Toronto, Canada, http://gypsy.rose.utoronto.ca/people/anu_dir/papers/atc/atcDND.html.

[RIC 98] J. P. Richard, « *Some trends and tools for the study of Time Delay Systems* », Proc. of the 2nd IMACS-IEEE Computational Engineering in Systems Applications Conf. (CESA'98), Tunisie, Avril 1998, Vol. P, pp. 27–43..

[SAY 96] C. P. Sayers, D. R. Yoerger, R. P. Paul et J. S. Lisiewicz, « *A Manipulator Work Package for Teleoperation from Unmanned Untethered Vehicles — Current Feasibility and Future Applications* », Proc. of the IARP Workshop on Subsea Robotics, Toulon, France, 1996.

[SHE 86] T. B. Sheridan, « *Human Supervisory Control of Robot Systems* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'86) , San Francisco, USA, 1986, pp. 808–812.

[STE94] M. R. Stein, « *Behavior-Based Control for Time-Delayed Teleoperation* », PhD. Thesis, University of Pennsylvania, 1994.

[STE 97] M. R. Stein et K. Sutherland, « *Sharing resources over the Internet for Robotics Education* », Proc. of the SPIE Conf. on Intelligent Systems, Pittsburgh, Pennsylvania, USA, October 14–15, 1997, pp. 132–139.

[STR 00] O. Strauss, F. Comby et M. J. Aldon, « *Rough Histograms for Robust Statistics* », 15th Intl. Conf. on Pattern Recognition (ICPR'2000), Barcelona, Catalonia, Spain, 3-8 September 2000.

[SUN 97] Y. J. Sun, J. G. Hsieh et H. C. Yang, « *On the stability of Uncertain Systems with multiple time-varying Delays* », IEEE Transactions on Automatic Control, vol. 42, no. 1, January 1997.

[TAN 98] M. Tanimoto, F. Arai, T. Fukuda et M. Negoro, « *Augmentation of Safety in Teleoperation System for Intravascular Neurosurgery* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'98) , Leuven, Belgium, May 1998, pp. 2890–2895.

[TAR 96] T.-j. Tarn et K. Brady, « *A Framework for the Control of Time-Delayed Telerobotic Systems* », Proc. of the 5th IFAC Symp. On Robot Control (SY.RO.CO.), Nantes, France, September 3–5 1997.

[TAY 97] K. Taylor et B. Dalton, « *Issues in Internet Telerobotics* », Proc. of the Intl. Conf. on Field and Service Robotics (FSR' 97), Canberra, Australia, 1997.

[TAY 95] K. Taylor et J. Trevelyan, « *A Telerobot on the World Wide Web* », Proc. of the Nat. Conf. of the Australian Robot Association, Melbourne, Australia, July 5–7 1995.

[TSU 97] Y. Tsumaki et M. Uchiyama, « *A Model-Based Space Teleoperation System with Robustness against Modeling Errors* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'97) , Albuquerque, Nouveau Mexique, USA, April 1997, pp. 1594–1599.

[TUR 97] H. Turchi, A. Crosnier et P. Fraisse , « *Realtime Environment for Mission Programming of Telerobotics Systems* », Proc. of the SPIE Conf. on Intelligent Systems, Pittsburgh, Pennsylvania, USA, October 14–15, 1997, pp. 22–24.

[VER 81] J. Vertut, A. Micaelli, P. Marchall et J. Guittet, « *Short Transmission Delay in a Force Reflective Bilateral Manipulator* », Proc. of 4th Rom-An-Sy, Warsaw, 1981, pp. 269–285.

[VOG 99] J. Vogel, B. Brunner, K. Landzettel et G. Hirzinger, « *Internet Virtual Reality Technologien zur Fernvisualisierung für Teleservice- und Telerobotikanwendungen* », Industrielle Automation und Internet/Intranet-Technologie, pp 199–210. Editions Verlag, 1999

[YOK 00] Y. Yokokohji, T. Imaida etT. Yoshikawa, « *Bilateral Control with Energy Balance Monitoring under Time-Varying Communication Delay* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'2000) , Nagoya, Aichi, Japan, May 21–27, 1995, pp. 2684–2689.

[WAT 81] K. Watanabe et M. Ito, « *An observer for linear feedback control laws of multivariable systems with multiple delays in controls and outputs* », Systems and Control Letters, Vol. 1, No. 1, July 1981, pp. 54–59.

[YOU 91] K. Youcef-Toumi et C. C. Shortlidge, « *Control of robot manipulators using time delay* », Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA'91) , Sacramento CA, USA, April 1991, pp. 2391–2395.

Glossaire

Quelques définitions liées à la téléopération

Téléopération : désigne les principes et les techniques qui permettent à l'opérateur humain d'accomplir une tâche à distance, à l'aide d'un système robotique d'intervention, commandé à partir d'une station de contrôle, par l'intermédiaire d'un canal de télécommunication.

Téléopération assistée par ordinateur (*TAO*) : Il s'agit d'une amélioration de la téléopération classique où l'informatique vient s'interposer entre l'opérateur et l'esclave pour palier leur séparation physique. Plusieurs implémentations sont possibles ; il peut s'agir d'une interface utilisateur plus ou moins intelligente (modules de pré-simulation, d'anticipation de mouvement et de collision, ...) afin d'aider l'opérateur dans sa tâche.

Réalité virtuelle : désigne tout système qui procure à l'opérateur humain la sensation d'immersion dans et la capacité d'interaction avec un environnement virtuel, c'est à dire basé sur un modèle de synthèse entièrement généré par ordinateur.

Réalité augmentée : caractérise tout système qui améliore la perception de l'opérateur vis à vis de l'environnement réel, généralement par superposition d'images de synthèse sur des images réelles ou vidéo.

Téléprésence : concept qui représente une application idéale des technologies de réalité augmentée et de réalité virtuelle. La sensation d'immersion de l'opérateur dans la scène de téléopération ainsi que la transparence du système sont en effet des qualités recherchées en téléopération.

Télérobotique : c'est une forme avancée de téléopération où le système téléopéré peut fonctionner sans la supervision de son opérateur (de façon autonome) durant de courtes périodes, ce qui fait de lui un robot.

Contrôle superviseur : dans de nombreux cas, les tâches à effectuer par le manipulateur sont bien modélisées et répétitives. Il est alors intéressant de préprogrammer

ces tâches de telle sorte que l'opérateur ne soit plus responsable que du placement initial du manipulateur, de sa supervision et de la gestion des événements non prévus.

Présence virtuelle : en téléopération bilatérale, l'opérateur se fie à un retour d'informations concernant la scène de téléopération souvent non naturel et incomplet. Ce retour, généralement composé de plusieurs flux vidéo, oblige l'opérateur à reconstituer mentalement la scène. La technologie de « présence virtuelle » tente d'améliorer cette interface homme-machine en utilisant la réalité virtuelle pour représenter les retours d'information en 3 dimensions et générer des ordres en fonction des gestes de l'opérateur.

Annexe B - Variables d'ondes

Définition

La méthode consiste à redéfinir le flux de puissance d'un système ${}^t\dot{x}(t).f(t)$ à l'aide d'un flux « aller » $\frac{1}{2}{}^t u(t).u(t)$ et d'un flux « retour » $\frac{1}{2}{}^t v(t).v(t)$.

$$P(t) = {}^t\dot{x}(t).f(t) = \frac{1}{2}{}^t u(t).u(t) - \frac{1}{2}{}^t v(t).v(t) \tag{B.1}$$

Ainsi $u(t)$ est associé à une onde aller (ou entrante) et $v(t)$ à une onde retour (ou sortante). Les variables d'ondes $u(t)$ et $v(t)$ sont des fonctions bijectives simples de $\dot{x}(t)$ et de $f(t)$:

$$\begin{aligned} u(t) &= \frac{b.\dot{x}(t) + f(t)}{\sqrt{2b}} \\ v(t) &= \frac{b.\dot{x}(t) - f(t)}{\sqrt{2b}} \end{aligned} \tag{B.2}$$

où b est l'impédance caractéristique de l'onde qui peut être une constante strictement positive ou une matrice définie positive. Cette impédance est choisie en fonction des caractéristiques souhaitées du système.

Passivité

Application aux variables d'ondes

Cette notion a été décrite dans l'annexe A. Dans ce cas d'étude :

$$(A.8) \Leftrightarrow \int_0^t \frac{1}{2}{}^t v(\tau).v(\tau)\mathrm{d}\tau \leqslant \int_0^t \frac{1}{2}{}^t u(\tau).u(\tau)\mathrm{d}\tau + \gamma^2 \qquad \forall t \geqslant 0 \tag{B.3}$$

Autrement dit, un système est passif si l'énergie de l'onde sortante $v(t)$ est limitée à l'énergie entrante $u(t)$ ou stockée initialement dans le système.

Un système passif en notation ($\dot{x}(t)$,$f(t)$) est également passif en variables d'ondes. Une composition en série de systèmes passifs est également passive. Cependant les compositions en parallèle ou les rétroactions qui sont passives en notation ($\dot{x}(t)$,$f(t)$) ne le sont plus en variables d'ondes.

Passivité d'un système à retard

Dans le cas d'un réseau de HILBERT, la puissance entrante ou sortante d'un système dépend du produit des variables énergétiques $f(t)$ et $\dot{x}(t)$. S'il un retard est introduit dans le système, la passivité n'est plus garantie. Dans le cas des variables d'ondes, les puissances entrante et sortante dépendent uniquement respectivement de $u(t)$ et de $v(t)$. Ainsi, si l'onde sortante est retardée, sa puissance est stockée temporairement sans altération de la passivité du système.

Supposons que $v(t) = u(t - T_r)$. L'équation B.3 est satisfaite et l'énergie stockée vaut :

$$E_{\text{stockée}} = \int_{t-T_r}^{t} \frac{1}{2}{}^{t}u(\tau).u(\tau)\mathrm{d}\tau$$

Application à des transmissions à retards

Considérons un bloc de transmissions aller–retour tel que celui présenté en figure B.1. Les relations entre les différentes variables sont données par les équations du système (B.4).

$$\begin{array}{lll} u_m(t) & = \frac{b.\dot{x}_m(t)+f_m(t)}{\sqrt{2.b}} & \qquad u_s(t) = u_m(t-T) \\ v_m(t) & = v_s(t-T) & \qquad v_s(t) = \frac{b.\dot{x}_s(t)-f_s(t)}{\sqrt{2.b}} \end{array} \tag{B.4}$$

Pour l'instant, le système est volontairement défini sans préciser si les forces $f_m(t)$ et $f_s(t)$ ainsi que les vitesses $\dot{x}_m(t)$ et $\dot{x}_s(t)$ sont des entrées ou des sorties.

Ainsi, le maître peut commander soit la vitesse :

$$\dot{x}_m(t) = \sqrt{\frac{2}{b}}v_m(t) + \frac{1}{b}f_m(t) \tag{B.5}$$

soit la force :

$$f_m(t) = b.\dot{x}_m(t) - \sqrt{2.b}.v_m(t) \tag{B.6}$$

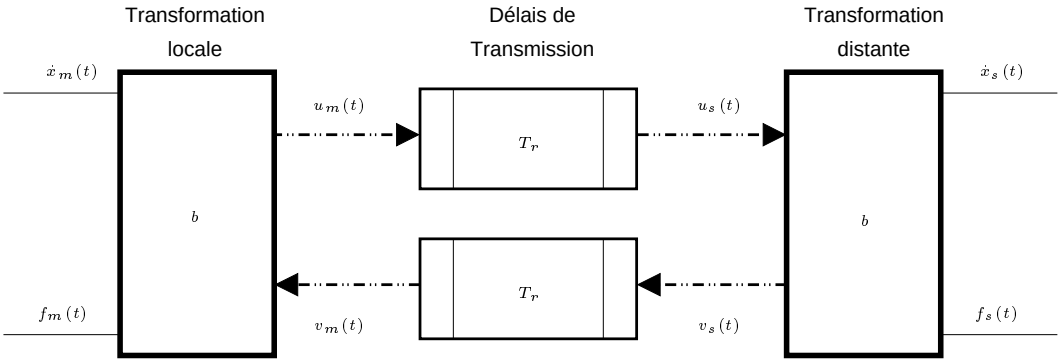


Figure B.1 – *Modèle du bloc de transmissions avec des variables d'ondes*

De même pour l'esclave :

$$\dot{x}_s(t) = \sqrt{\frac{2}{b}} u_s(t) - \frac{1}{b} f_s(t) \tag{B.7}$$

ou

$$f_s(t) = -b.\dot{x}_s(t) + \sqrt{2.b}.u_s(t) \tag{B.8}$$

Annexe C - Protocoles pour l'*Internet*

Organisation générale

Problématique

Nul besoin de présenter ce qu'est l'*Internet*, étant donné sa très forte et récente médiatisation en France.

Grand outil de communication, cet inter-réseau se prête assez bien aux problèmes impliquant un contrôle à distance. De manière non exhaustive, on y compte les télé-services, le télé-travail, le télé-achat, la télé-collaboration, le télé-enseignement, la télé-formation, le télé-diagnostic, la télémaintenance, la télé-médecine, la télé-information culturelle et enfin la téléopération.

La plupart du temps ces applications nécessitent de véhiculer du son et de l'image (animée) et sont donc très gourmandes en bande passante. La plupart des réseaux locaux a une technologie adaptée aux problèmes spécifiques à résoudre. Il est impossible de trouver une technologie satisfaisant tous les types de besoins. Ainsi les réseaux *Ethernet*ou Token-Ring se spécialisent dans l'acheminement d'informations de tous types en favorisant le partage des ressources entre tous les utilisateurs. Les réseaux de terrain, amenés à véhiculer des informations cruciales pour le fonctionnement d'une chaîne de fabrication, par exemple, favorisent la sécurité et l'aspect temps-réel. Cette variété de technologies rend tous ces réseaux locaux autonomes et sans intercommunication possible telle quelle.

L'insertion des protocoles *TCP/IP* entre les architectures propriétaires et les utilisateurs, associée à l'interconnexion de différents types de réseaux à l'aide de « *passerelles* », permet donc d'interconnecter tout type de réseau informatique. Ces protocoles s'adaptent, de façon transparente pour l'utilisateur, à chaque architecture réseau sur laquelle ils viennent se greffer et donnent ainsi l'illusion d'un réseau homogène. Cependant, le prix à payer réside dans la perte des spécificités et services particuliers offerts par chacune de ces architectures ; les services réseau que fournissent les protocoles

TCP/IP reposent sur l'intersection de la plupart des services disponibles pour chaque catégorie. Toutefois, si une architecture possède une bande passante et un temps d'accès particulièrement supérieurs à la moyenne, ceux-ci ne seront heureusement pas perdus, mais seulement très légèrement diminués par le traitement supplémentaire imposé par *TCP/IP* pour assurer le bon transport des données.

Protocoles qui font l'*Internet*

La hiérarchie des protocoles créés pour l'*Internet* est présenté dans la figure C.1).

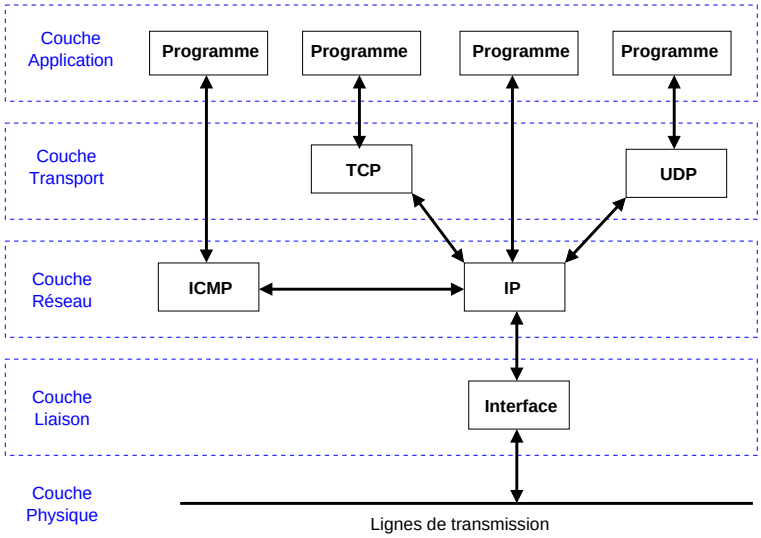


Figure C.1 – *Architecture* TCP/IP

L'adoption quasi-universelle de la suite de protocoles *TCP/IP* en fait son principal intérêt. La naissance de l'*Internet* date du début des années 80 aux Etats-Unis, au moment où le DARPA (*Defense Advanced Research Projects Agency*) créa la série de protocoles *TCP/IP* pour ses réseaux de recherche *Arpanet* et militaire *Milnet*.

Le protocole *IP* (*Internet Protocol*) est utilisé au moins au niveau de chaque passerelle (routeur) entre les différents réseaux à interconnecter afin d'orienter (router) les paquets d'informations vers le réseau local dans lequel est situé le destinataire (repéré par son *adresse IP*), en passant éventuellement par plusieurs autres réseaux. On appelle ce type de réseau, un **réseau routé** : chaque paquet suit sa propre route qui est optimisée à chaque instant.

Il y a plusieurs inconvénients à ce type de routage :

- le service est dit non-fiable : la remise des paquets n'est pas garantie ; un paquet peut être perdu, dupliqué, ... sans que l'émetteur ni le récepteur ne l'apprennent ;
- il n'y a pas de reprise d'erreur ;

– lorsqu'un long message est fragmenté en plusieurs paquets — afin de pouvoir traverser tous les réseaux locaux dont la technologie impose une longueur de trame donnée — et que ceux-ci prennent des routes différentes en fonction de la charge des noeuds rencontrés, il est fort probable que ces paquets n'arrivent pas dans le bon ordre. Cette faiblesse rend difficile l'acheminement de données en temps réel sur le réseau tel que le transport de la voix ou d'un flux video. Il revient alors aux protocoles de niveau supérieur à *IP* (*TCP* notamment) de se charger de cette résolution, si besoin est.

IP est le protocole sur lequel se fondent les protocoles de transport *TCP* (*Transmission Control Protocol*) et *UDP* (*User Datagram Protocol*), lesquels sont utilisés à leur tour pour certaines applications telles que `Telnet` (connexion à une station à distance), *FTP* (transfert de fichiers), *SMTP* (transfert de courriers électronique), *NFS* (accès à un système de fichiers à distance), ...

Nous décrirons un peu mieux les protocoles *TCP* et *ICMP* qui ont été utilisés dans notre travail.

Routage

Lorsqu'un paquet est adressé à un hôte situé dans un autre réseau local que celui de l'émetteur, celui-ci est acheminé de passerelle en passerelle jusqu'à son destinataire. Il existe de nombreux protocoles chargés de déterminer le meilleur chemin que doit prendre le paquet en fonction de la taille du réseau à gérer. Citons, par exemple, *RIP* (*Routing Information Protocol*), plutôt orienté sur le routage de petits réseaux. Il utilise un algorithme permettant de trouver le chemin le plus court, c'est à dire le nombre de passerelles $n < 16$ à traverser le plus faible. La valeur 16 indique une impossibilité. Cet algorithme se fonde sur l'utilisation d'une table de routage actualisée toutes les 30 s et échangée entre les différents routeurs.

Protocole IP

La figure C.2 représente l'en-tête *IP* que toute trame voyageant sur l'*Internet* possède. Le format et le fonctionnement détaillé de ce protocole sont fournis dans la `RFC 791`.

Le numéro de version actuel est 4, bien que la version 6 soit déjà parue. Le champ `IHL: Internet Header Length` correspond à la longueur de l'en-tête en mots de 32 bits. Il pointe ainsi sur le début des données et est > 5.

Le champ `Type of Service` donne une indication sur la qualité du service désiré. Cette qualité se définit en terme de priorité plus ou moins élevée par rapport au reste du traffic.

Le champ `Total Length` contient la longueur totale du datagramme *IP* en octets : en-tête + données. Etant donné qu'il est codé sur 16 bits, s'il y a besoin d'émettre

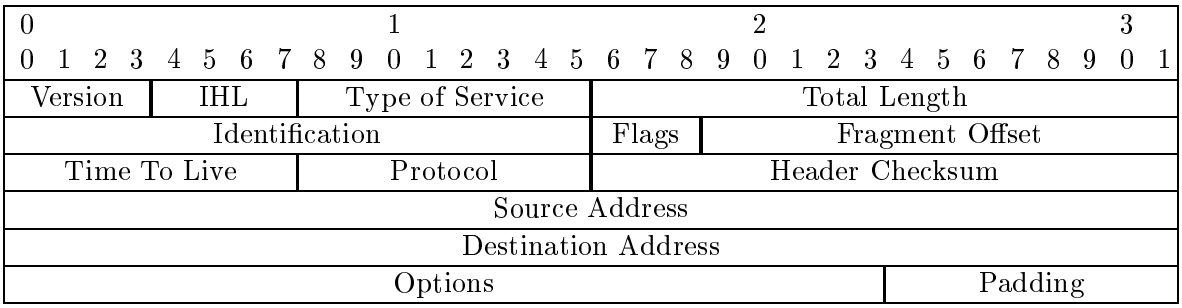


Figure C.2 – *Format d'un en-tête* IP

des données dont la taille excède $2^{16} - 20 \approx 64$ Ko (l'en-tête *IP* fait, au minimum, 20 octets), il faudra fractionner les données en plusieurs trames.

Les champ `Identification`, `Flags` et `Fragment Offset` permettent de fragmenter, si besoin est, un datagramme trop long pour passer sur une architecture de réseau donnée.

Le champ `Time to Live` indique la durée de vie d'un datagramme. Si ce champ contient une valeur nulle, la trame est détruite. Ce champ est modifié chaque fois qu'un routeur analyse l'en-tête *IP*. Le temps est mesuré en secondes ; cependant, chaque routeur rencontré sur le chemin décrémente ce champ d'au moins une unité même si le datagramme est resté moins d'une seconde dans le routeur. Il faut donc considérer ce champ comme une limite supérieure de durée de vie. Le but est de détruire des datagrammes ne trouvant jamais de destinataire ou bloqués dans une route fermée.

Le champ `Protocol` indique le protocole immédiatement supérieur à *IP*.

Les champs `Source Address` et `Destination Address` contiennent l'adresse *IP* de l'expéditeur et du destinataire.

Enfin, diverses options peuvent être ajoutées sur la façon dont les routeurs devront gérer cette trame, sur le niveau de sécurité (*Unclassified* à *Top Secret*), ...

Protocole ICMP

Le protocole *ICMP* (*Internet Control Message Protocol*) gère principalement des messages de contrôle et d'erreur que ne fournit pas *IP*. Il est défini initialement dans la `RFC 777` et corrigé dans la `RFC 792`. Des versions plus récentes existent mais sont associées à la version 6 d'*IP* (la version actuelle la plus courante est 4).

Plusieurs type de messages sont prévus :

- demande d'écho et sa réponse avec ou sans datage horaire,
- erreurs : hôte non accessible, défaut dans un paquet, ...
- contrôle de flux,

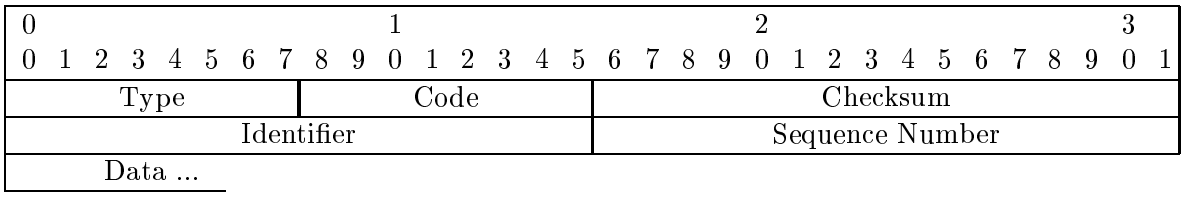


Figure C.3 – *Format d'une trame* ICMP *Echo*

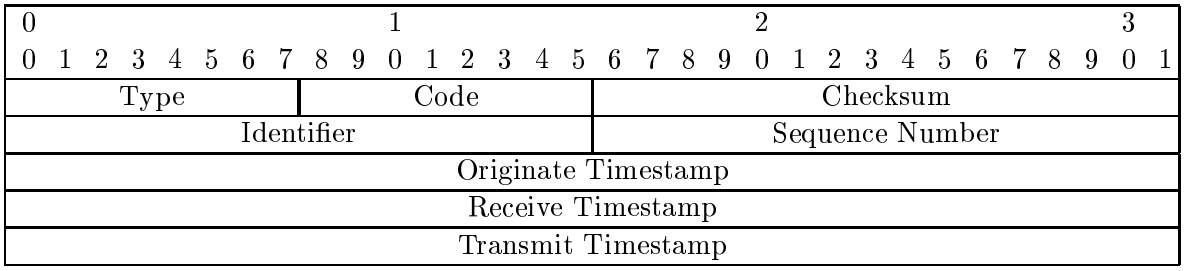


Figure C.4 – *Format d'une trame* ICMP *Timestamp*

– ...

Echo

Les messages `Echo` et `Echo Reply` suivent le format de la trame illustrée en figure C.3. Cet trame est placée dans la zone de données d'une trame *IP* (encapsulation).

Dans l'en-tête *IP* associé, l'adresse de la source dans la trame de demande d'écho devient celle de destination dans la trame de réponse.

Le champ `type` prend la valeur 8 pour une demande d'écho et 0 pour le message de réponse. La somme de contrôle est codée sur 16 bits ; son calcul est explicité dans la `RFC 792`.

Les champs `Code`, `Identifier` et `Sequence Number` permettent de différencier les retours d'échos quand plusieurs demandes n'ont pas encore abouti.

Les données émises dans la demande d'écho sont recopiées dans la réponse.

Horodatage

Il existe également un service d'horodatage que nous n'avons pas exploité, faute de temps, mais qu'il serait intéressant d'utiliser ultérieurement, en remplacement de la méthode écho + mesure temps d'aller–retour.

Les messages `Timestamp` et `Timestamp Reply` utilisent le format de trame illustré en figure C.4. Cette trame est concaténée à un en-tête *IP* avant d'être émise sur le réseau.

Le fonctionnement est semblable à celui des messages `echo` ; `type` = 13 pour les

demandes d'horodatage et 14 pour les réponses.

Les champs `timestamp` contiennent le nombre de millisecondes écoulées depuis minuit en temps universel. `Originate Timestamp` correspond à l'heure à laquelle l'émetteur a envoyé la trame de demande d'horodatage. `Receive Timestamp` est la date de réception de la demande par l'hôte cible. Enfin, `Transmit Timestamp` correspond à l'heure d'envoi de la trame de réponse.

Protocole *TCP*

Le but du protocole *TCP* (*Transmission Control Protocol*) est d'assurer un service de transport de l'information fiable :

- établissement d'une connexion duplex et contrôle de cette connexion,
- les octets d'un message sont reçus dans l'ordre d'émission,
- un système d'acquittement garantit l'arrivée de toutes les données,
- transfert bufferisé : le récepteur choisit sa vitesse de lecture des informations reçues et l'émetteur attend d'avoir assez d'octets avant d'envoyer une trame.

Le modèle de connexion entre deux machines s'appelle « *client–serveur* ». Le serveur est un programme exécuté sur une des machines, qui attend les connexions de machines clientes. Dès qu'une ou plusieurs machines tentent de se connecter au serveur, ce dernier leur ouvre une session de communication qui sera fermée à l'initiative du client ou du serveur selon le cas. Un serveur qui n'accepte qu'une seule connexion simultanée est appelé «*serveur itératif* ».

Afin qu'une machine puisse rendre divers services sous forme de serveurs (résolution de noms, accès distant, ...), chaque serveur écoute les connexions sur un « *port* » donné (un nombre codé sur 16 bits). Les services habituels ont des ports fixés : 7 pour echo, 21 pour *FTP*, 23 pour *Telnet*, 25 pour *SMTP*, ... Il est possible de faire une analogie avec une adresse postale : l'adresse *IP* correspond au nom de la rue et le numéro de port au numéro dans la rue.

Programmation *Internet*

Dans les années 80, le DARPA finança l'université de Californie de Berkeley afin de développer une interface de programmation (en anglais *API: Application Program Interface*) pour créer des applications de communication entre hôtes sur les réseaux *TCP/IP*. Cette *API* s'appelle l'interface « *socket* ».

Une *socket* représente l'extrémité d'une conversation réseau entre deux hôtes. Une *socket* se charge de guetter les demandes de connexions de clients, d'effectuer une

demande de connexion pour une application cliente, d'envoyer des données à l'interlocuteur et de réceptionner les données émises par celui-ci.

Le programmeur paramètre ses *sockets* pour l'utilisation d'un protocole particulier au choix (*IP*, *TCP*, *UDP* ou direct). Il possède de nombreuses routines permettant d'effectuer des tâches de serveur, de client, de résolution d'adresses, ...

L'*API socket* Berkeley a été développée dans l'univers *UNIX* mais elle est conçue pour être portable sur n'importe quel type de système d'exploitation. Microsoft® a créé sa propre *API Winsock* adaptée au monde *Windows* et dérivant des *sockets*.

Une bonne référence pour la programmation réseau avec les *sockets* ou les *Winsocks* est [JAM 96]. En ce qui concerne les protocoles liés à l'*Internet*, la source originelle réside dans les `RFCs` (*Requests For Comments*) qui sont notamment accessibles sur le site de l'`UREC`[8] (Unité Réseaux du CNRS).

8. http://www.urec.fr/standard

Annexe D - Commande linéarisante

Principe

Considérons un système possédant une représentation d'état linéaire vis-à-vis de la commande :

$$(\Sigma) \qquad \begin{cases} \dot{x} = f(x) + \sum_{i=1}^{m} u_i.g_i(x) \\ y = h(x) \end{cases} \tag{D.1}$$

où :

- x est le vecteur d'état de dimensions n,
- $u = {}^t[u_1, \ldots, u_m]$ est le vecteur de commande de dimension m,
- y est un vecteur composé de m sorties indépendantes,
- les fonctions f,$g : \mathbb{R}^n \to \mathbb{R}^n$ et $h : \mathbb{R}^n \to \mathbb{R}^m$ sont analytiques.

La méthode consiste à réaliser un bouclage de la forme $u(x) = \alpha(x) + \beta(x).v$ tel que la nouvelle relation $v \to y$ soit celle d'un système linéaire et découplé.

Pour ce faire, il est nécessaire d'introduire l'opérateur de Lie $\mathbb{L}$ qui est défini de la manière suivante:

étant donnés un champ de vecteurs $w : \mathbb{R}^n \to \mathbb{R}^n$ et une fonction scalaire $s : \mathbb{R}^n \to \mathbb{R}$, la dérivée de Lie de s par rapport à w s'écrit:

$$\mathbb{L}_w(s) = \sum_{j=1}^{n} \frac{\partial s}{\partial x_j} w_j \tag{D.2}$$

Le résultat $\mathbb{L}_w(s)$ étant une fonction scalaire, il est à nouveau possible d'appliquer l'opérateur de Lie avec n'importe quel champ de vecteurs. Notamment, $\mathbb{L}_w(\mathbb{L}_w(s))$ sera noté $\mathbb{L}_w^2(s)$.

A chaque sortie y_i, nous associons un nombre caractéristique ρ_i défini par :

$$\rho_i = \inf \left\{ k \ / \ \exists j \in \{1, \ldots, m\} \quad \text{tel que} \quad \mathbb{L}_{g_j}(\mathbb{L}_f^k(h_i)) \neq 0 \right\} \tag{D.3}$$

Il est possible de montrer que $\rho_i + 1$ représente l'ordre de la première dérivée de la sortie y_i sur laquelle agit directement la commande. De plus, si ρ_i est infini, donc non défini, cela signifie que la sortie y_i n'est pas influencée par les entrées du système.

Si tous les ρ_i sont définis, nous construisons une matrice appelée matrice de découplage $\Delta(x)$ de dimension $(m \times m)$:

$$\forall i, j \in \{1, \ldots, m\}, \qquad \Delta_{i,j}(x) = \mathbb{L}_{g_j}(\mathbb{L}_f^{\rho_i}(h_i(x))) \tag{D.4}$$

Nous pouvons maintenant énoncer le théorème :

Théorème 3
Une condition suffisante pour qu'il existe une loi de bouclage de la forme

$$u(x) = \alpha(x) + \beta(x).v$$

solution du problème de linéarisation et découplage entrée/sortie est que la matrice de découplage $\Delta(x)$ soit inversible ; il suffit alors de choisir

$$\alpha(x) = \Delta^{-1}(x).a(x) \qquad et \qquad \beta(x) = \Delta^{-1}(x).b(x) \tag{D.5}$$

avec $a_i(x) = a_{0,i}.h_i(x) + a_{1,i}.\mathbb{L}_f(h_i(x)) + \ldots + a_{\rho_i,i}.\mathbb{L}_f^{\rho_i}(h_i(x)) - \mathbb{L}_f^{\rho_i+1}(h_i(x))$

et $\begin{cases} b_{i,j} = 0 & si\ i \neq j \\ b_{i,j} = b_i \end{cases}$

où les $a_{j,i}$ et les b_i sont des constantes arbitraires.

Nous définissons alors de nouvelles variables d'état :

$$\forall i \in \{1, \ldots, m\}, \qquad z_i = \begin{bmatrix} h_i & \mathbb{L}_f(h_i) & \ldots & \mathbb{L}_f^{\rho_i}(h_i) \end{bmatrix}$$

En fait, la $k^{\text{ième}}$ composante de z_i n'est autre que la $(k-1)^{\text{ième}}$ dérivée par rapport au temps de y_i.

Après bouclage, le vecteur z_i vérifie l'équation :

$$\dot{z}_i = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & \ldots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & \ldots & 0 & 1 \\ a_{0,i} & a_{1,i} & a_{2,i} & \ldots & \ldots & a_{\rho_i-1,i} & a_{\rho_i,i} \end{bmatrix} .z_i + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} .v_i \tag{D.6}$$

et $y_i = [1, 0, \ldots, 0].z_i$.

Ceci définit un sous-système (Λ_i) *de dimension* $\rho_i + 1$. *Soit* (Λ) *l'ensemble des sous-systèmes* (Λ_i), *alors le système* (Σ) *après bouclage a le même comportement entrée/sortie que* (Λ): *on a bien réalisé une linéarisation entrée/sortie et chaque sortie* y_i, *ne dépend que de l'entrée* v_i.

REMARQUE 5 *La dimension du nouvel espace d'état est:*

$$\sum_{i=1}^{m}(\rho_i + 1)$$

.

Si $\sum_{i=1}^{m}(\rho_i + 1) < n$, *nous avons créé des modes inobservables qui peuvent être instables. En revanche, si* $\sum_{i=1}^{m}(\rho_i + 1) = n$, *alors les deux représentations d'état* x *et* $z = [{}^t z_1 \ {}^t z_2 \ \ldots \ {}^t z_m]$ *sont équivalentes; nous avons donc complètement linéarisé le système.*

Annexe E - Modélisation du bras manipulateur *PUMA* 560

Modèle dynamique articulaire

Nous nous plaçons tout d'abord dans le cas général d'un robot manipulateur à chaîne cinématique ouverte simple. Pour que la modélisation soit complète, il faut prendre en compte non seulement la chaîne cinématique, mais aussi les actionneurs et les variateurs.

Pour la partie cinématique, en s'appuyant sur le formalisme de LAGRANGE ou celui de NEWTON-EULER, nous déterminons la relation entre les forces/couples transmis aux articulations (Γ_q) et les positions, vitesses et accélérations articulaires ($q, \dot{q}, \ddot{q}$). Nous obtenons sous forme matricielle l'équation :

$$\Gamma_q = A_q(q).\ddot{q} + H_q(q, \dot{q}) + {}^tJ.F_e \tag{E.1}$$

avec $H_q(q, \dot{q}) = B_q(q, \dot{q}) + C_q(q, \dot{q}) + Q_q(q) + F_{sv}(q, \dot{q})$ et :

$A_q(q)$	matrice d'inertie,
$B_q(q, \dot{q})$	vecteur des forces de CORIOLIS,
$C_q(q, \dot{q})$	vecteur des forces centrifuges,
$Q_q(q)$	vecteur des forces de gravité,
$F_{sv}(q, \dot{q})$	vecteur des forces de frottements (secs et visqueux),
J	jacobienne du robot
F_e	effort (force/couple) exercé par l'organe terminal sur son environnement.

Par la suite, nous appellerons N le nombre d'articulations, qui sont toutes motorisées puisqu'il s'agit d'un robot à chaîne ouverte simple : $\dim(\Gamma_q)=N$. Ainsi tous les vecteurs sont de dimension N et les matrices de dimension $(N \times N)$.

La seule équation (E.1) ne peut pas suffire pour modéliser le système car nous n'avons pas directement accès au vecteur Γ_q. Il dépend du couple transmis par les

actionneurs, qui lui-même est une fonction assez complexe des tensions d'alimentation.

En ce qui concerne la partie actionneurs et variateurs, la modélisation dépend du type de matériel utilisé. Nous considérerons donc le cas du robot *PUMA* 560 où chaque actionneur est un moteur à courant continu entraînant une articulation par l'intérmédiaire d'un réducteur et alimenté par un variateur de courant.

En négligeant les effets gyroscopiques, la mise en équation de la partie mécanique conduit à la relation suivante :

$$\Gamma_m = J_m.\ddot{m} + f_m.\dot{m} + C_m \tag{E.2}$$

avec :

Γ_m	vecteur des couples électromagnétiques des moteurs,
m	vecteur des positions angulaires des moteurs,
$J_m = \text{diag}[J_{mi}]$	où J_{mi} représente l'inertie propre du moteur i augmentée de l'inertie des organes de transmission et ramenée à l'arbre moteur,
$f_m = \text{diag}[f_{mi}]$	où f_{mi} est le coefficient de frottement visqueux du moteur i,
C_m	vecteur des couples résistants au niveau des moteurs.

Pour se ramener aux articulations, nous définissons la matrice des rapports de transmission N_t. Nous tenons compte des éventuels couplages entre les différentes articulations, ce qui se traduit par la présence de termes extra-diagonaux non nuls dans N_t. En supposant la transmission parfaite, c'est-à-dire sans jeu ni élasticité ni phénomène d'hystérésis, nous obtenons la relation :

$$\dot{m} = N_t.\dot{q} \tag{E.3}$$

qui se traduit au niveau des couples par :

$$C_m = {}^tN_t{}^{-1}.\Gamma_q \tag{E.4}$$

Les équations (E.2), (E.3) et (E.4) donnent alors (E.5) :

$$\Gamma_m = J_m.N_t.\ddot{q} + f_m.N_t.\dot{q} + {}^tN_t{}^{-1}.\Gamma_q \tag{E.5}$$

D'où, en tenant compte de (E.1), l'équation vérifiée par Γ_m :

$$\Gamma_m = {}^tN_t{}^{-1}.\left\{\left[{}^tN_t.J_m.N_t + A_q(q)\right].\ddot{q} + H_q(q,\dot{q}) + {}^tN_t.f_m.N_t.\dot{q} + {}^tJ.F_e\right\} \tag{E.6}$$

Nous pouvons nous ramener à une équation du type de (E.1) en définissant τ par :

$$\Gamma_m = {}^tN_t{}^{-1}.\tau \tag{E.7}$$

τ représente ainsi le couple qui aurait été transmis aux articulations si nous avions pu négliger les effets dynamiques des moteurs. τ vérifie l'équation de la dynamique :

$$\tau = A(q).\ddot{q} + H(q,\dot{q}) + {}^tJ.F_e \tag{E.8}$$

$$\text{avec } \begin{cases} A(q) & = A_q(q) + {}^tN_t.J_m.N_t \\ H(q,\dot{q}) & = H_q(q,\dot{q}) + {}^tN_t.f_m.N_t.\dot{q} \end{cases}$$

Il reste maintenant à relier τ au vecteur des tensions appliquées U, qui constitue le véritable vecteur de commande. Nous savons que le couple moteur Γ_m est proportionnel au courant de l'induit :

$$\Gamma_m = K_c.I \tag{E.9}$$

avec I le vecteur des courants d'induit et $K_c = \text{diag}[K_{c_i}]$ où K_{c_i}est la constante de couple du moteur i. D'autre part, chaque moteur est alimenté en courant par un variateur qui joue le rôle de convertisseur tension/courant. En supposant les variateurs parfaits, nous avons :

$$I = K_a.U \tag{E.10}$$

avec $K_a = \text{diag}[K_{a_i}]$ où K_{a_i} est le gain courant/tension du variateur i.

En fait, les variateurs réalisent un asservissement du courant I. Et comme il s'agit de processus purement électriques, il est facile de rendre les constantes de temps intervenant dans ces asservissements négligeables à côté de celles de la partie mécanique, ce qui justifie l'équation (E.10).

Les équations (E.7), (E.9) et (E.10) donnent alors (E.11):

$$\tau = {}^tN_t.K_c.K_a.U \tag{E.11}$$

La matrice ${}^tN_t.K_c.K_a.U$ étant constante et inversible, il y a équivalence entre τ et U. Nous pouvons considérer finalement que le vecteur de commande est τ. Ainsi l'équation (E.8) modélise complètement l'ensemble du processus. Nous avons donc pu obtenir un modèle de la même forme que (E.1) tout en considérant la dynamique complète du système et moyennant peu d'approximations.

Modèle dynamique cartésien

Il s'agit maintenant de déterminer, à partir de l'équation (E.8), le modèle dynamique dans l'espace cartésien.

Soit X le vecteur position/orientation de l'organe terminal, défini dans un repère de référence. Nous choisissons une représentation cartésienne telle que le nombre de composantes de X soit égal au nombre de degrés de libertés cartésiens (ce qui exclut notamment la représentation par les cosinus directeurs), de telle sorte que les composantes de X soient indépendantes entre elles. Par ailleurs, nous nous limiterons dorénavant aux robots non redondants, ce qui est le cas des *PUMA* 560. Ainsi X est constitué de N composantes indépendantes. Il en résulte, d'une part, que comme τ est de dimension N, nous pourrons appliquer le principe du bouclage découplant et linéarisant (présenté en annexe D) à la sortie $y = X$ et, d'autre part, que le modèle géométrique direct du robot traduit une relation bijective entre X et q :

$$X = \mathcal{F}(q) \tag{E.12}$$

Pour déterminer le modèle dynamique du robot dans l'espace opérationnel, nous avons également besoin du modèle différentiel direct :

$$\dot{X} = J(q).\dot{q} \tag{E.13}$$

Nous nous plaçons en dehors des singularités, d'où: $\dot{q} = J^{-1}(q).\dot{X}$.

Ainsi les vecteurs $\begin{bmatrix} q \\ \dot{q} \end{bmatrix}$ et $\begin{bmatrix} X \\ \dot{X} \end{bmatrix}$ sont équivalents. De plus nous pouvons poser :

$$F = \left({}^tJ\right)^{-1}.\tau \tag{E.14}$$

F représente le vecteur des forces/moments qui serait appliqué à l'environnement si τ était intégralement transmis à l'effecteur.

Sachant que $\begin{cases} q &= \mathcal{F}^{-1}(X) \\ \dot{q} &= J^{-1}(q).\dot{X} \\ \ddot{q} &= J^{-1}(q).[\ddot{X} - \dot{J}(q,\dot{q}).\dot{q}] \end{cases}$

nous montrons facilement que l'équation de la dynamique (E.8) s'écrit, dans l'espace opérationnel :

$$F = A_X(X).\ddot{X} + H_X(X,\dot{X}) + F_e \tag{E.15}$$

$$\text{avec} \quad \begin{cases} A_X(X) & = {}^tJ^{-1}.A(q).J^{-1}(q) \\ H_X(X,\dot{X}) & = -{}^tJ^{-1}.A(q).J^{-1}(q).\dot{J}(q,\dot{q}).\dot{q} + {}^tJ^{-1}(q).H(q,\dot{q}) \end{cases}$$

$$\text{où} \quad \begin{cases} q & = \mathcal{F}^{-1}(X) \\ \dot{q} & = J^{-1}(\mathcal{F}^{-1}(X)).\dot{X} \end{cases}$$

Il est alors possible d'appliquer le principe de bouclage découplant et linéarisant au système décrit par l'équation (E.15).

Annexe F - Identification dynamique de systèmes échantillonnés

---

La méthode suivante a pour but de déterminer les coefficients d'un filtre linéaire échantillonné (à la période T_e) à partir de ses dernières entrées et sorties. A cette fin, elle utilise le principe des moindres carrés récursifs d'ordre N.

Soit un système linéaire échantillonné dont la fonction de transfert $F(z)$ est de la forme de (F.1) (avec $m \leqslant n$):

$$F(z) = \frac{Y(z)}{U(z)} = \frac{b_0 + b_1.z^{-1} + \ldots + b_m.z^{-m}}{1 + a_1.z^{-1} + \ldots + a_n.z^{-n}} \tag{F.1}$$

En mesurant l'évolution de $y[k]$ et de $u[k]$, nous cherchons à obtenir les a_i ($i \in [1;n]$) et les b_j ($j \in [0;m]$).

Cela nous fait donc $m + n + 1$ paramètres à identifier au cours de N mesures avec $N \gg m + n + 1$.

A l'instant $t = k.T_e$, la relation entre les N dernières mesures et les coefficients a_i ($i \in [1;n]$) et b_j ($j \in [0;m]$) peut s'écrire sous la forme de l'équation (F.2) :

$$Z[k] = H[k].\theta + \varepsilon \tag{F.2}$$

Dans laquelle :

$$\begin{cases} \theta & = {}^t\begin{bmatrix} a_1 & a_2 & \cdots & a_n & b_0 & b_1 & \cdots & b_m \end{bmatrix} \\ Z[k] & = {}^t\begin{bmatrix} y[k-N] & y[k-N+1] & \cdots & y[k] \end{bmatrix} \\ H[k] & = \begin{bmatrix} y[k-N-1] & \cdots & y[k-n-N] & u[k-N] & \cdots & u[k-m-N] \\ y[k-N] & \cdots & y[k-n-N+1] & u[k-N+1] & \cdots & u[k-m-N+1] \\ \vdots & & & & & \vdots \\ y[k-1] & \cdots & y[k-n] & u[k] & \cdots & u[k-m] \end{bmatrix} \\ \varepsilon & = \mathcal{N}(0,\sigma^2) \end{cases}$$

$$\text{avec} \begin{cases} \dim(\theta) & = ((m+n+1)\times 1) \\ \dim(Z[k]) & = ((N+1)\times 1) \\ \dim(H[k]) & = ((N+1)\times(m+n+1)) \end{cases}$$

La dernière ligne de la matrice $H[k]$ est notée ${}^t h[k]$

L'estimée de la sortie à l'instant k est notée $\hat{y}[k]$. Ainsi, à l'instant $t=(k+1)T_e$, la sortie estimée sera donc :

$$\hat{y}[k+1] = {}^t h[k].\hat{\theta}[k] \tag{F.3}$$

A ce moment, il est possible de prédire le vecteur des coefficients estimés $\hat{\theta}$ à l'instant $k+1$:

$$\hat{\theta}[k+1] = \hat{\theta}_k + \overbrace{K[k]}^{\text{Gain d'estimation}} \times \overbrace{\left\{ y[k+1] - \hat{y}[k+1] \right\}}^{\text{Erreur de prédiction}} \tag{F.4}$$

Soit $P[k]$ la matrice de covariance de l'erreur d'estimation à l'instant k.

$$P[k] = (\theta[k] - \hat{\theta}[k]).{}^t(\theta[k] - \hat{\theta}[k]) = \sigma^2.({}^t H[k].H[k])^{-1} \tag{F.5}$$

Alors :

$$K[k] = \frac{P[k].h[k+1]}{\sigma^2 + {}^t h[k+1].P[k].h[k+1]} \tag{F.6}$$

et

$$P[k+1] = P_k - \frac{P[k].h[k+1].{}^t h[k+1].P[k]}{\sigma^2 + {}^t h[k+1].P[k].h[k+1]} \qquad \text{(F.7)}$$

Il y a donc besoin de définir préalablement $P[0]$ arbitrairement.

Annexe G - Les filtres de Kalman

Introduction

Position du problème

L'estimation de l'état $x(t)$ d'un système dynamique à partir de mesures bruitées peut être divisé en trois classes distinctes selon l'intervalle d'observation $[t_0\, t_1]$:

- la prédiction si $t > t_1$,
- le filtrage si $t = t_1$,
- le lissage si $t_0 < t < t_1$

Modèle mathématique du système à estimer

Le filtre de KALMAN tente d'estimer l'état $x \in \Re^n$ d'un système discret gouverné par les équations linéaires stochastiques suivantes :

$$x_{k+1} = A_k.x_k + B_k.u_k + w_k \tag{G.8}$$

l'état x_k est mesuré par $z_k \in \Re^m$ selon :

$$z_k = H_k.x_k + v_k \tag{G.9}$$

Les variables aléatoires w_k et v_k représentent respectivement le bruit du processus et celui de la mesure. Nous les supposerons indépendants entre eux et de type gaussien, centrés sur 0 et de variance respectivement Q et R.

$$\begin{aligned} p(w) &= \mathcal{N}(0, Q) \\ p(v) &= \mathcal{N}(0, R) \end{aligned}$$

Dans l'équation (G.8), la matrice A $(n \times n)$ calcule l'état x à l'instant $k+1$ en fonction de l'état x précédent (à l'instant k) en l'absence de bruit et de consigne d'entrée.
La matrice B $(n \times l)$ relie les consignes d'entrée $u_k \in \Re^l$ à l'état x_k.
La matrice H calcule la mesure z_k en fonction de l'état x_k (équation (G.9)).

L'algorithme

Le filtre de KALMAN estime l'état d'un processus en utilisant une sorte d'asservissement : le filtre estime cet état à un instant donné et obtient ensuite le retour sous forme de mesure bruitée. Ainsi le système d'équations se divise en deux groupes : les équations de rafraîchissement *(time update)* et les équations de mesure *(measurement update)*. Les premières prédisent a priori l'état suivant du système ainsi que les covariances d'erreur d'estimation, les secondes corrigent la prédiction précédente (qui devient a posteriori) en fonction des mesures.

Les équations de rafraîchissement peuvent être considérées également comme équation de *prédiction* et les équations de mesure comme équations de *correction* (cf figure G.1).

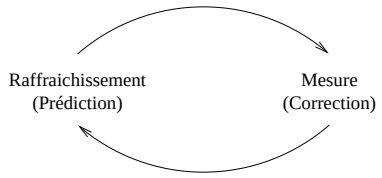


Figure G.1 – *La boucle de prédiction-correction du filtre de* KALMAN

Le tableau (G.1) présente les équations spécifiques pour la prédiction (équations G.10 et G.11) et pour la correction (équations G.12, G.13 et G.14):

La première tâche de la phase de correction consiste à calculer le gain de KALMAN K_k (équation G.12). Ensuite, nous génèrons l'estimation a posteriori (équation G.13) en tenant compte de la mesure. Enfin nous calculons l'estimation a posteriori de la covariance de l'erreur (équation G.14).

A chaque tour, nous reprenons l'estimation a posteriori afin de prédire l'état suivant.

Paramétrages et réglages

Il est nécessaire de connaître la matrice de covariance d'erreur de mesure R_k et la matrice de bruit du processus Q_k avant de démarrer l'algorithme. En général, il s'agit

$$\begin{aligned} \hat{x}_{k+1}^{-} &= A_k.\hat{x}_k + B.u_k && \text{(G.10)} \\ P_{k+1}^{-} &= A_k.P_k.A_k^t + Q_k && \text{(G.11)} \end{aligned}$$

$$\begin{aligned} K_k &= P_k^{-}.H_k^t.(H_k.P_k^{-}.H_k^t + R_k)^{-1} && \text{(G.12)} \\ \hat{x}_k &= \hat{x}_k^{-} + K.(z_k - H_k.\hat{x}_k^{-}) && \text{(G.13)} \\ P_k &= (I - K_k.H_k).P_k^{-} && \text{(G.14)} \end{aligned}$$

Tableau G.1 – *Equations du filtre de* KALMAN *classique*

d'effectuer quelques mesures préliminaires hors ligne.

En ce qui concerne Q_k, le choix est moins déterminant. En effet, cette source de bruit est souvent utilisée pour représenter les incertitudes dans la modélisation.

Origines du filtre de Kalman

Soient $\hat{x}_k^{-} \in \Re^n$ l'état estimé a priori de x au pas k (en connaissant son évolution avant cet instant) et $\hat{x}_k \in \Re^n$ l'estimation a posteriori de x au pas k (étant donnée la mesure z_k). On définit alors les erreurs a priori et a posteriori :

$$\begin{aligned} e_k^{-} &\equiv x_k - \hat{x}_k^{-} \\ e_k &\equiv x_k - \hat{x}_k \end{aligned} \qquad \text{(G.15)}$$

Les covariances des erreurs d'estimation a priori et a posteriori valent donc :

$$\begin{aligned} P_k^{-} &= E[e_k^{-}.e_k^{-t}] \\ P_k &= E[e_k.e_k^t] \end{aligned} \qquad \text{(G.16)}$$

Le principe du filtre de KALMAN est de corriger l'estimation a priori en tenant compte des mesures reçues entre temps. La différence $(z_k - H_k.\hat{x}_k^{-})$ de l'équation G.17 est appelée *résidu* ou encore *innovation de mesure*. Elle donne un aperçu de la différence entre la mesure réelle et la mesure estimée a priori.

$$\hat{x}_k = \hat{x}_k^{-} + K.(z_k - H_k.\hat{x}_k^{-}) \qquad \text{(G.17)}$$

La matrice $K(n \times m)$ est un gain calculé de façon à minimiser la covariance de l'erreur d'estimation a posteriori (équation G.18). Elle se démontre en cherchant le

minimum de la fonction $P_k(K)$ obtenue en manipulant les équations G.15, G.16 et G.17.

$$K_k = P_k^-.H_k^t.(H_k.P_k^-.H_k^t + R_k)^{-1} \tag{G.18}$$

En analysant l'équation G.18, nous pouvons remarquer que plus la covariance de l'erreur de mesure R_k est faible, plus nous avons confiance dans les mesures pour calculer la valeur de l'estimé a posteriori $\hat{x}_k$:

$$\lim_{R_k \to 0} K_k = H_k^{-1}$$

A l'opposé, lorsque la covariance de l'erreur d'estimation a priori tend vers 0, l'estimée a posteriori dépend nettement plus de l'estimée a priori que des mesures z_k :

$$\lim_{P_k^- \to 0} K_k = 0$$