

## CHAPITRE 8

### ÉVALUATION DU PROTOCOLE DE MIGRATION

#### 8.1 Introduction

Le protocole de migration est développé et documenté. Cette section permet d'évaluer qualitativement et quantitativement le protocole de migration afin de connaître son niveau de succès.

#### 8.2 Démarche d'évaluation

L'évaluation du protocole a deux facettes, l'une quantitative et l'autre qualitative.

L'évaluation quantitative du protocole a pour but de déterminer quelles sont les performances résultantes d'un produit de migration utilisé avec un mécanisme plus complexe que le vérin utilisé au CHAPITRE 5, qui avait pour but de faire le choix technologique du produit de migration.

L'évaluation qualitative du protocole a pour but de connaître l'appréciation des utilisateurs face à l'emploi du protocole de migration. Cette évaluation est effectuée en faisant expérimenter le protocole aux utilisateurs, puis en leur soumettant un questionnaire d'évaluation.

#### 8.3 Évaluation quantitative

L'évaluation quantitative est décrite dans cette section. Les critères évalués sont présentés, suivi de la description des éléments à évaluer pour se terminer avec l'analyse des résultats obtenus.

### 8.3.1 Critères évalués

Tous les critères évalués sont mesurés dans deux situations. La première situation est caractérisée par le fait que le contrôleur global et la loi de contrôle sont développés exclusivement avec le CAR Microb. C'est normalement cette situation qui donne les meilleures performances. Les performances obtenues servent de base de comparaison. La seconde situation est caractérisée par l'utilisation du protocole de migration pour intégrer la loi de contrôle au même contrôleur global.

Les couples calculés pour les deux situations par la loi de contrôle sont comparés pour valider l'exactitude des calculs effectués. Trois des quatre critères présentés au CHAPITRE 5 sont évalués, soit le temps de cycle, la mémoire vive utilisée et l'espace disque utilisé. Toutefois, comme le temps de migration ne peut être comparé significativement, il n'est pas mesuré.

### 8.3.2 Description des éléments évalués

Le contrôleur global Bart, décrit à la section 7.3 et dont les paramètres détaillés sont à l'ANNEXE 9, est employé pour l'évaluation. Toutes les mesures sont prises sur les deux plateformes supportées par le protocole de migration, soit sous Windows XP et QNX. Les performances du contrôleur à couple précalculé muni d'un PID, présenté à la Figure 15, sont mesurées.

Chaque articulation du robot doit suivre un profil de vitesse trapézoïdal calculé par le contrôleur global tel qu'illustré à la Figure 20. Par simplicité, chaque articulation suit la même trajectoire. Les informations sont enregistrées durant 7 secondes.

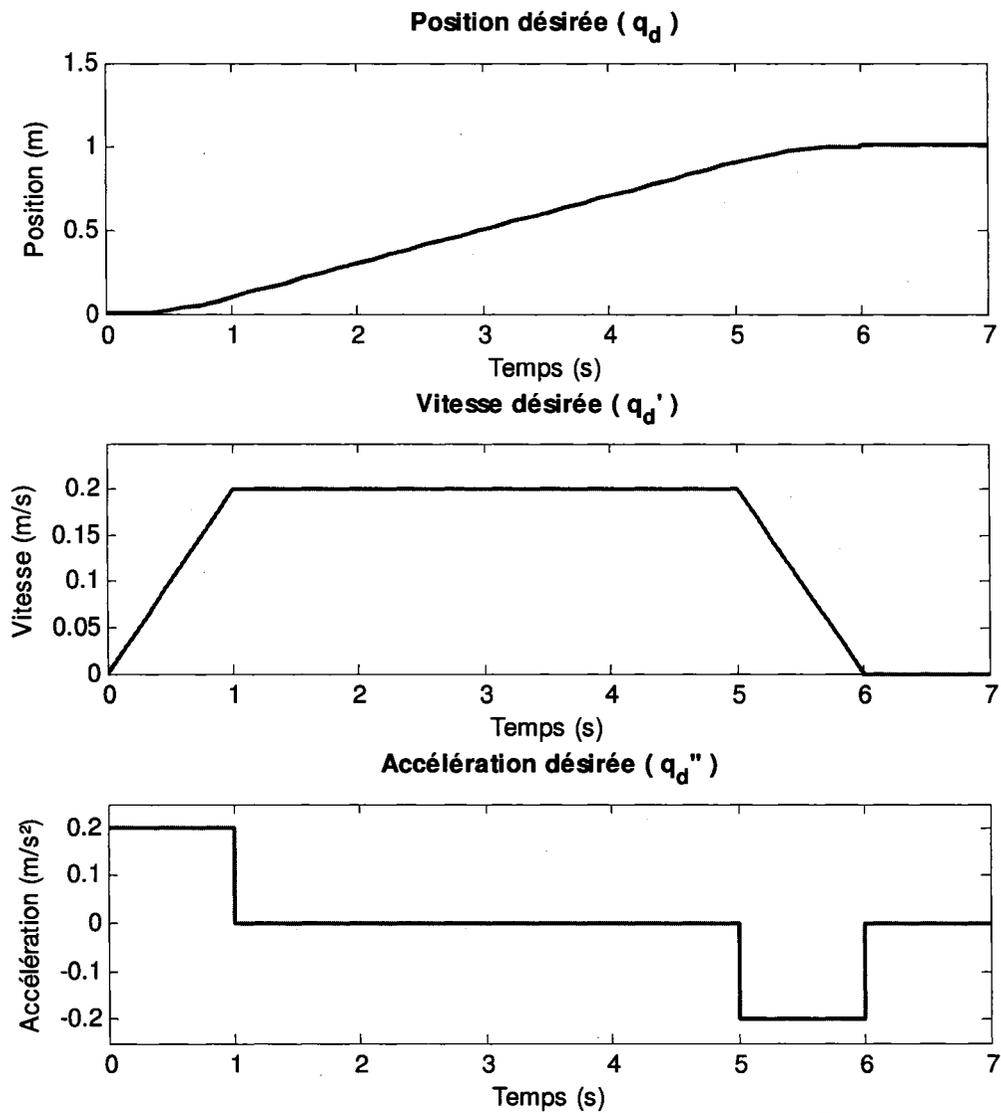


Figure 20 Illustration de la trajectoire désirée

### 8.3.3 Analyse des résultats

Les couples calculés sont dans un premier temps analysés afin de valider les calculs effectués. Par la suite, les critères (temps de calcul, mémoire vive utilisée, espace disque occupé) sont mesurés puis analysés pour connaître les performances réelles du protocole. Le sommaire des résultats est présenté dans le Tableau V. Il est à noter que le

contrôleur codé manuellement a été optimisé de façon à diminuer le temps de calcul présentés dans le Tableau VI.

Tableau V

Sommaire de l'évaluation quantitative

Critères	Contrôleur codé manuellement	Contrôleur Migré	Augmentation
Temps de calcul moyen sous Windows XP (ms)	0,0100	0,0099	-1,3%
Temps de calcul moyen sous QNX (ms)	0,0251	0,0385	53,3%
Mémoire vive utilisée sous Windows XP (Ko)	2788	3012	8,0%
Mémoire vive utilisée sous QNX (Ko)	1748	1876	7,3%
Espace disque occupé sous Windows XP (Ko)	396	432	9,1%
Espace disque occupé sous QNX (Ko)	1736	1886	8,6%

Tableau VI

Comparaison des temps de calculs après optimisation du code manuel

Critères	Contrôleur codé manuellement (optimisé)	Contrôleur Migré	Augmentation
Temps de calcul moyen sous Windows XP (ms)	0,0078	0,0099	26,4%
Temps de calcul moyen sous QNX (ms)	0,0172	0,0385	123,3%

### 8.3.3.1 Couples calculés

Les couples calculés, tant sous Windows XP (voir Figure 21) que sous QNX (Figure 22), diffèrent légèrement lorsqu'ils sont calculés avec Microb ou avec la bibliothèque partagée. Il est suspecté que la méthode d'intégration soit en cause. Bien que les deux calculs utilisent l'algorithme d'intégration de Runge-Kutta d'ordre 4, des

différences dans l'implémentation de cet algorithme peuvent conduire à ces écarts dans les résultats obtenus. Toutefois, l'écart maximal relatif obtenu est en deçà de 0,055 %, ce qui peut être considéré comme étant négligeable.

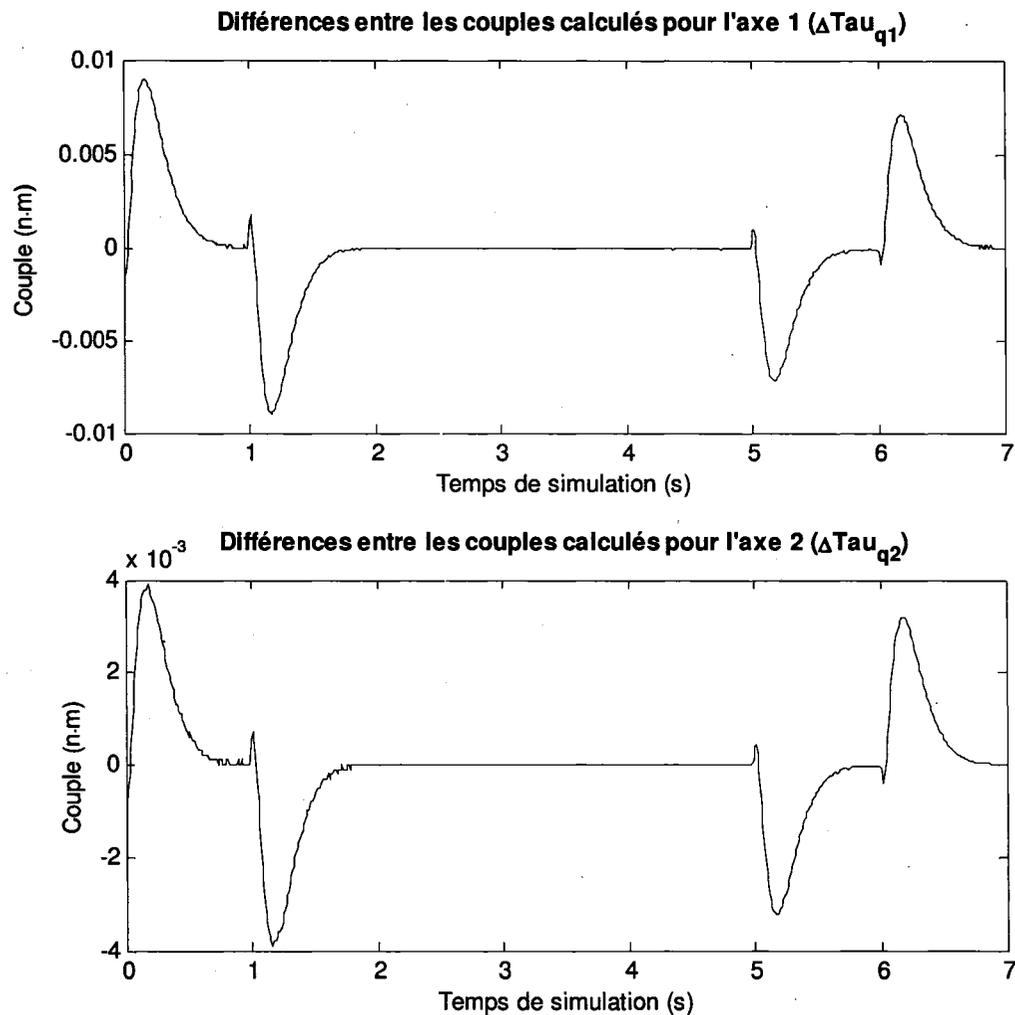


Figure 21 Différence entre les couples calculés par Microb et par la bibliothèque partagée sous Windows XP

Afin de confirmer que c'est l'implémentation de l'intégrateur qui cause cette différence de calcul, la simulation est relancée en posant un gain intégral nul. Cela a pour effet d'annuler complètement l'effet de l'intégrateur. L'erreur obtenue suite à cette

modification est nulle. Ce résultat confirme donc que les calculs sans intégration sont exacts. Cela met également en relief le fait qu'il peut y avoir des différences de calculs d'intégration entre Microb et les bibliothèques partagées.

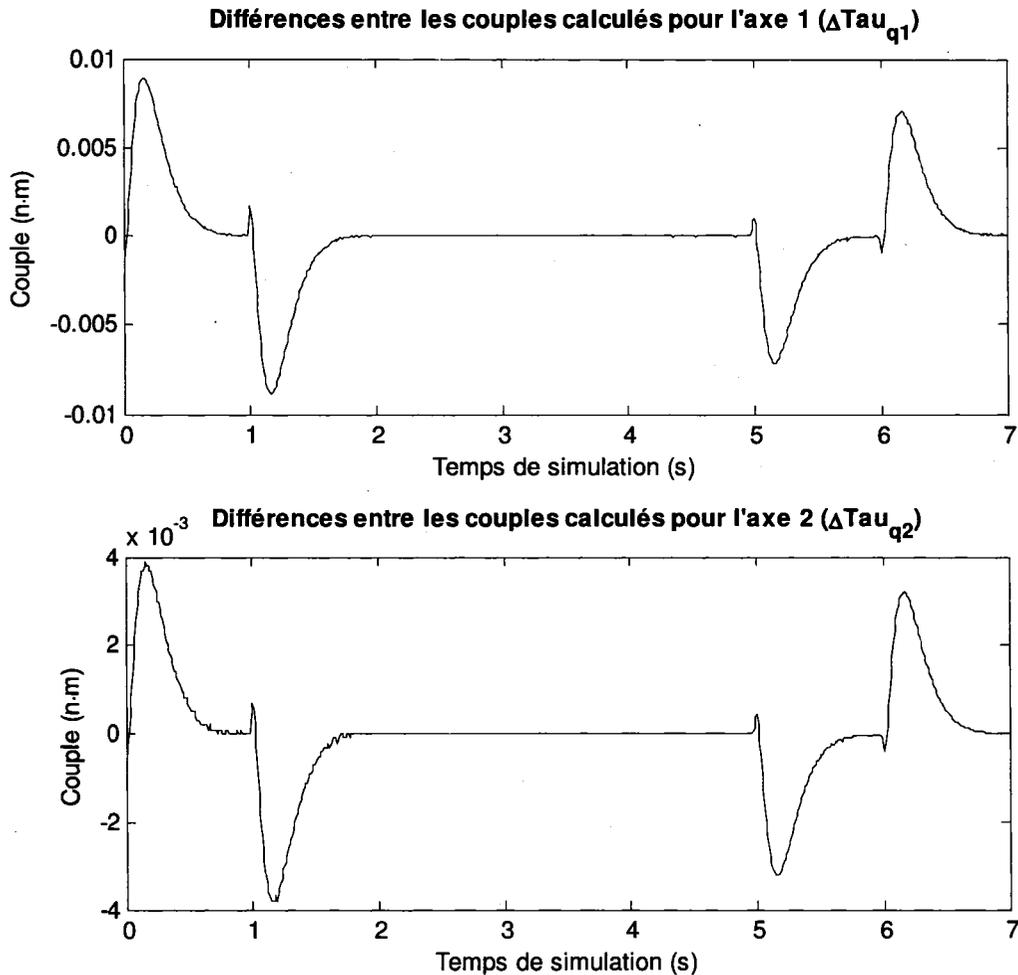


Figure 22 Différence entre les couples calculés par Microb et par la bibliothèque partagée sous QNX

Il est difficile d'affirmer quelle est la méthode d'implémentation de l'algorithme d'intégration qui est supérieure à l'autre. D'un côté, le logiciel Matlab possède une renommée qui le précède. D'un autre côté, une expérience réalisée avec Simulink (voir

Figure 23) montre que le logiciel a recours à l'extrapolation des points pour se rapprocher de la réalité.

Dans cette expérience, deux signaux rampes sont intégrés. Le premier signal vient du bloc rampe natif de Simulink (pente = 1) et le second est généré à partir d'une variable initialisé manuellement ( $x = [0:.1:5; 0:.1:5]'$ ). Il est à noter que les deux signaux utilisés sont numériquement identiques, i.e. que la différence est nulle entre les deux signaux illustrés dans le bloc « Erreur signal ».

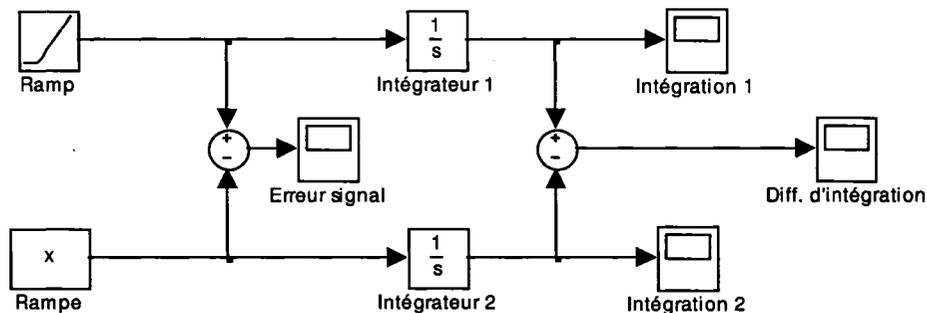


Figure 23 Schéma expérimental sous Simulink

Les différences entre les intégrations suite à une simulation de 5 secondes sont illustrées à la Figure 24. La différence entre les deux méthodes est constante (0,00333 par itération), donc cela signifie qu'il s'agit d'une erreur linéaire.

Dans le cas de l'intégration d'un signal provenant d'un bloc rampe, Simulink n'intègre pas à l'aide d'un bloqueur d'ordre zéro, mais en considérant la pente de la rampe et en extrapolant le résultat. Cependant, lorsque le type de signal est camouflé à Simulink à l'aide d'une variable issue de l'environnement de Matlab, l'intégration est réalisée avec un bloqueur d'ordre zéro et aucune extrapolation n'est réalisée.

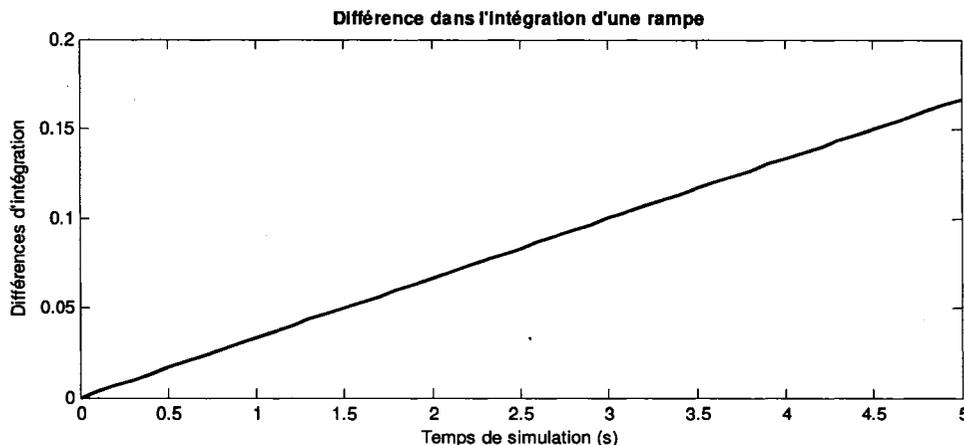


Figure 24 Différence d'intégration d'une rampe

Bien que cette expérience ne prouve pas que Matlab soit dans l'erreur, elle montre que le logiciel ne se comporte pas toujours de la même façon selon le signal reçu en entrée. Il n'est donc pas possible de conclure que l'approche de Matlab ou celle Microb donne des résultats plus exacts. Des tests plus poussés doivent être faits avant de statuer sur la question et cela ne fera pas l'objet de ce mémoire. Toutefois, comme les deux méthodes arrivent à des résultats similaires, elles peuvent être considérées comme équivalentes.

### 8.3.3.2 Temps de calculs

Les temps de calculs sont illustrés dans les graphiques qui suivent (voir Figure 25 à Figure 28). Il est bon de souligner que les données ont été lissées à l'aide de la fonction *smooth* de Matlab pour permettre une meilleure lisibilité des graphiques. Les pointes dans les graphiques représentent un calcul interrompu par le processeur. Sous Windows XP, les calculs de la bibliothèque dynamique sont 1.3 % plus rapide que ceux entièrement effectués uniquement par le contrôleur global. En simplifiant l'équation, il est possible d'optimiser le code manuel pour minimiser le temps de calcul. Cette optimisation permet d'avoir un contrôleur 23% plus rapide que le contrôleur non optimisé sous Microb et 26.4% plus rapide que le contrôleur migré de Simulink.

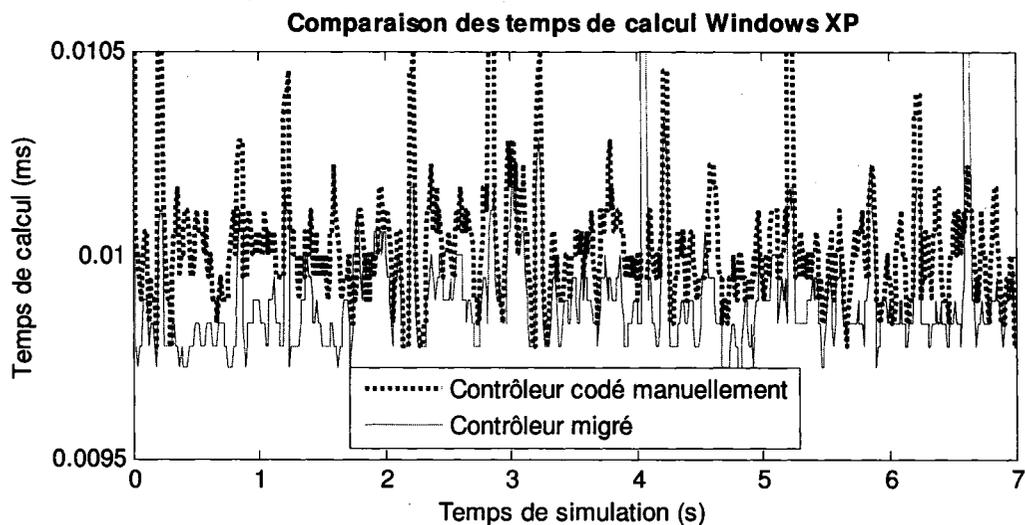


Figure 25 Temps de calcul sous Windows XP

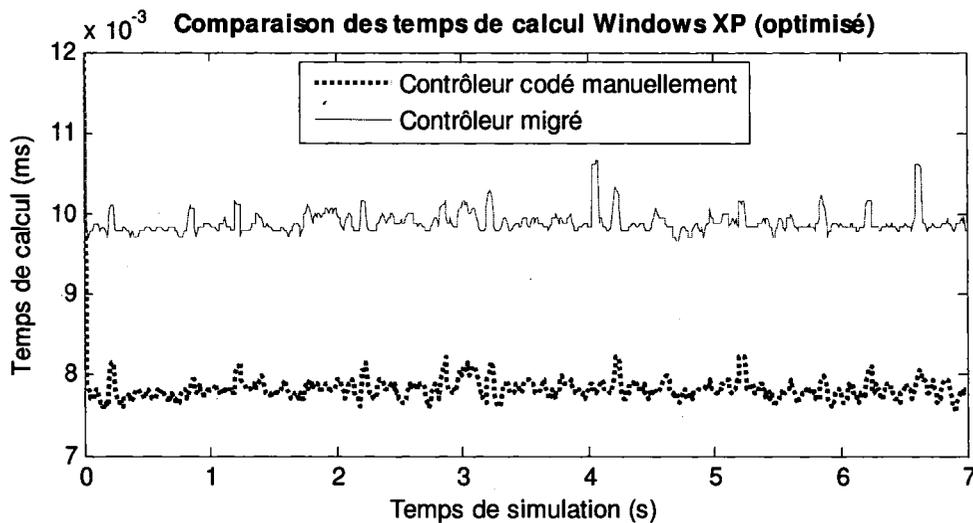


Figure 26 Temps de calcul sous Windows XP après optimisation

Les calculs de la bibliothèque partagée sous QNX sont 53 % plus lents que ceux entièrement effectués uniquement par le contrôleur global. En simplifiant l'équation, il est possible d'optimiser les calculs. Cette optimisation permet d'avoir un contrôleur 31,4 % plus rapide que le contrôleur sous Microb et 123 % plus rapide que le contrôleur

migré. Cette moins bonne performance sous QNX est potentiellement due au fait que les mesures ont été prises sous une machine virtuelle ou encore à la façon dont QNX gère ses bibliothèques partagées.

La comparaison avec les contrôleurs optimisés, tel que présenté dans la Figure 26 et la Figure 28, met en relief le fait que la bibliothèque générée par Matlab optimise les calculs, ce qui peut résulter en un gain de performance net lorsque la loi de contrôle est suffisamment complexe.

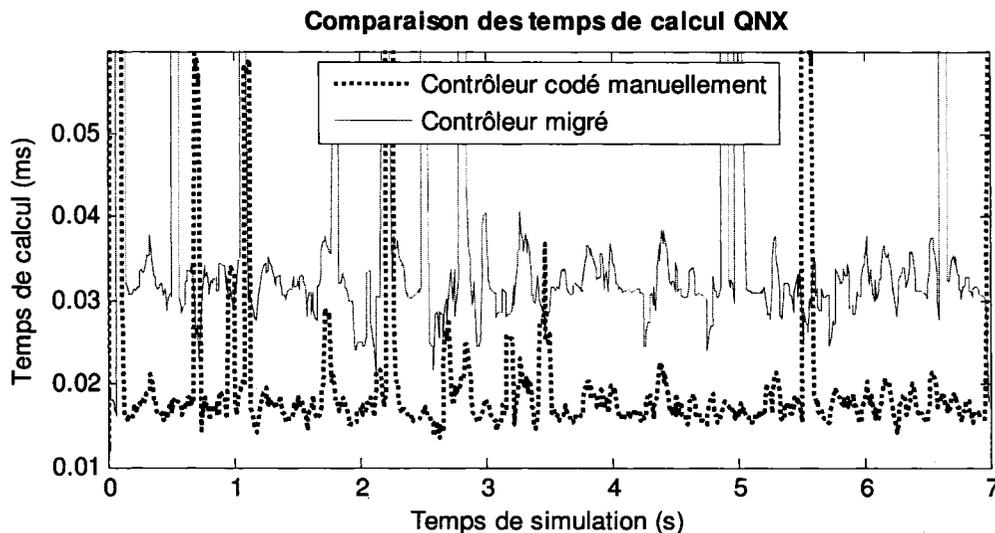


Figure 27 Temps de calcul sous QNX

Le temps supplémentaire requis par les calculs avec la bibliothèque partagée, allant jusqu'à 123 %, peut sembler important mais représente en fait une fraction négligeable du temps de cycle ( $\sim 0,1$  %). Malgré que cette hausse soit supérieure à 10%, elle est considérée comme admissible car elle est peu élevée par rapport à la période d'échantillonnage ( $T = 0.01$  s, 100 Hz). Comme cette hausse est inférieure à 10% de la période d'échantillonnage, elle est considérée comme acceptable [51]. Dans le pire des cas, la perte d'efficacité attendrait 10% du temps de cycle pour un contrôleur fonctionnant à une fréquence 100 fois plus élevée ( $T = 0.0001$  s, 10 000 Hz).

Néanmoins, d'autres tests pourraient être faits avec des lois de commande plus complexes.

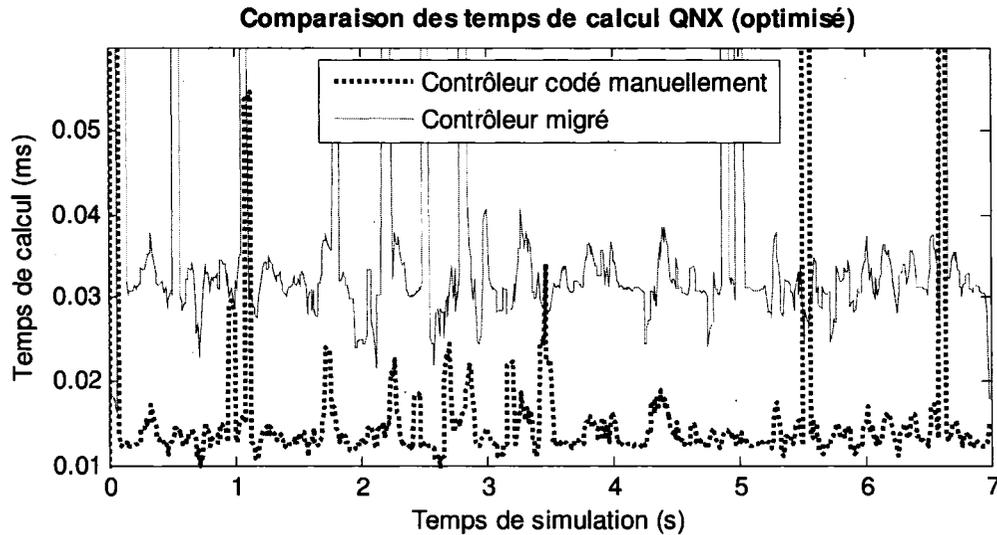


Figure 28 Temps de calcul sous QNX après optimisation

### 8.3.3.3 Mémoire vive utilisée

La mesure de mémoire vive est prise en faisant la différence entre la mémoire vive totale utilisée par le système avant et celle à la fin de l'exécution du contrôleur global. La mémoire vive utilisée sous Windows XP par le contrôleur codé manuellement est de 2788 Ko tandis que la mémoire vive utilisée par le contrôleur migré sous forme de bibliothèque dynamique est de 3012 Ko. Le contrôleur global utilisant une loi de migration migrée augmente la mémoire vive utilisée de 8 % ( $3012 / 2788$ ). Sous QNX, une augmentation de 7.3 % ( $(1748 \text{ Ko} + 128 \text{ Ko}) / 1748 \text{ Ko}$ ) est observée. Comme cette hausse est sous la barre du 10 %, elle est considérée comme acceptable [51].

#### **8.3.3.4 Espace disque utilisé**

L'espace disque utilisée sous Windows XP augmente de 9 % ((396ko + 36ko)/396 Ko) en utilisant le produit de migration. Sous QNX, une hausse similaire de 8.6 % ((1736 Ko + 150 Ko) / 1736 Ko) de l'utilisation de l'espace disque est observée. Comme cette hausse est sous la barre du 10 %, elle est considérée comme admissible [51].

### **8.4 Évaluation qualitative**

L'évaluation consiste à consulter un groupe d'utilisateur. Ce groupe évalue le protocole en le testant puis en répondant à un questionnaire (voir ANNEXE 2). Ce questionnaire est basé sur les besoins des usagers décrits au CHAPITRE 4.

L'expérimentation du protocole pour l'évaluation est similaire à l'exemple d'utilisation présenté à la section 7.3. La principale différence entre l'exemple et l'évaluation tient dans le fait que la migration est effectuée uniquement sous Windows XP au lieu d'être de Windows XP vers QNX comme dans l'exemple.

Les six évaluateurs sont invités à répondre à un questionnaire comportant douze questions ouvertes et une section de commentaires. Les mêmes personnes qui ont effectués la consultation au CHAPITRE 4 répondent à ce sondage, à l'exception de moi-même et d'une autre personne qui n'était malheureusement pas disponible.

Il est à noter que deux erreurs se sont glissées aux questions 6 et 8 au niveau du pourcentage d'augmentation du temps de calcul requis et de la mémoire utilisé: on aurait dû lire respectivement 26,4 % (ou -1.3 %) et 10 % au lieu de 10 % et 40 %. Les réponses à ces deux questions ont donc été ignorées.

Les résultats obtenus suite à cette consultation sont les suivants:

- 83 % des répondants jugent que le protocole de migration remplit son mandat. La personne insatisfaite mentionne que le protocole répond partiellement au mandat car une partie du code doit être programmé par un usager Microb expérimenté;
- 100 % des répondants saisissent bien ce que le protocole de migration fait;
- 100 % des répondants trouvent que l'apprentissage du protocole de migration est aisé;
- 100 % des répondants trouvent la migration facile à effectuer. Un répondant mentionne que le protocole demande quand même un travail d'apprentissage pour le programmeur Microb;
- 67 % des répondants considèrent Real-Time Workshop comme étant un générateur de code suffisamment fiable pour être utilisé dans le développement de contrôleurs de robot. Les autres répondants s'abstiennent;
- 100 % des répondants considèrent le temps de migration requis comme satisfaisant;
- 100 % des répondants considèrent comme acceptable une augmentation d'environ 10 % de l'espace disque (ROM) dû à l'utilisation d'un contrôleur migré;
- 83 % des répondants trouvent l'installation du protocole suffisamment simple et claire. Le répondant insatisfait mentionne que le texte pourrait être retravaillé, ce qui a été fait dans le CHAPITRE 7;
- 100 % des répondants trouvent que le protocole de migration s'intègre harmonieusement à l'environnement Matlab \ Simulink.

Certains ont émis des suggestions pour améliorer le protocole: clarifier la procédure d'installation et d'utilisation du protocole, automatiser la copie des fichiers directement dans Matlab et fournir un exemple complet d'utilisation, incluant ce qu'il faut faire du côté Microb, ce qui a été fait dans le CHAPITRE 7. Le commentaire positif suivant a été émis: « Bon travail, très prometteur pour diverses applications ». En bref, les résultats de

l'évaluation montrent que sur l'ensemble des questions, la vaste majorité des utilisateurs potentiels semblent satisfaits du protocole de migration développé.

### **8.5 Problèmes observés**

Lors des tests sous Windows XP, l'utilisation de la bibliothèque dynamique a pour effet de provoquer une panne dans le contrôleur global lorsqu'il s'approche de 65 000 itérations. Ce problème est probablement dû au mécanisme de chargement de bibliothèque dynamique sous Windows car il n'est pas observé sous QNX. La voie de solution envisageable à emprunter consiste à modifier le chargement de la bibliothèque pour qu'il se fasse au chargement de l'application plutôt que durant son le début de son exécution, comme c'est fait sous QNX.

### **8.6 Conclusion**

L'évaluation du protocole de migration, tant quantitative que qualitative, s'est avérée positive. Voici les points saillants de cette évaluation:

- Les algorithmes d'intégration du produit de migration n'arrivent pas tout à fait aux mêmes résultats que ceux du CAR Microb mais donnent des résultats très semblables (différence relative maximale de 0.05 %);
- Bien que l'augmentation du temps de calcul sous QNX soit plus importante que prévue, les temps de calculs demeurent négligeables par rapport à la période d'échantillonnage (< 0.1 %);
- L'augmentation de la mémoire vive utilisée ainsi que de l'espace disque requis provoquée par l'utilisation du produit de migration est raisonnable car inférieure à 10 % [51];
- La majorité des utilisateurs consultés sont satisfaits du protocole de migration développé;

- Plusieurs utilisateurs apprécieraient que la copie des fichiers soit automatisée.

En conclusion, le protocole atteint les objectifs, tant qualitatifs que quantitatifs, fixés au CHAPITRE 4.

MCours.com