

CHAPITRE 7

UTILISATION DU PROTOCOLE DE MIGRATION

7.1 Introduction

La section précédente a permis de comprendre le fonctionnement global du protocole de migration. Ce chapitre présente le manuel d'utilisation qui détaille chacune des étapes à franchir pour effectuer la migration d'une loi de commande, du PRC Simulink vers le CAR Microb. L'utilisation pratique du protocole de migration est présentée par la suite d'un exemple de migration.

7.2 Manuel d'utilisation

La migration s'effectue lorsque l'utilisateur possède un modèle Simulink fonctionnel, avec un certain nombre d'entrées et de sorties. Deux phases devront être franchies afin d'utiliser le protocole de migration, soit l'installation et l'initialisation. Dans un premier temps, les outils nécessaires au protocole devront être installés sur la machine de développement ainsi que sur la machine ciblée pour exécuter le contrôleur. Dans un deuxième temps, le programme du contrôleur Microb doit être initialisé afin que ce dernier puisse utiliser le code encapsulé dans la bibliothèque dynamique.

7.2.1 Préalables à l'installation

La bibliothèque Microb (version mai 2006) doit être installée sur les plateformes de simulation et d'exécution. L'ordinateur de simulation doit fonctionner sous Windows XP. Il doit avoir la suite logicielle Matlab (version 7.0.0.19920 (R14)) avec Real-Time Workshop (RTW 6.0) installée de façon fonctionnelle. Il semble que le protocole

fonctionne avec d'autres versions de Matlab mais aucun test approfondi n'a été fait pour valider cette hypothèse et rien ne laisse présager des problèmes de compatibilité.

7.2.2 Installation des fichiers requis par le protocole

L'installation des fichiers requis par le protocole permet de générer le code sur la plateforme de développement, i.e. celle sur lequel le modèle est développé sous Matlab/Simulink. Cette installation permet également de compiler le code sur la plateforme d'exécution, i.e. celle sur lequel la bibliothèque dynamique sera exécutée. Il est possible que ces deux ordinateurs ne fassent qu'un.

L'ajout des cibles permettant de générer des bibliothèques dynamiques sous QNX et sous Windows se fait en copiant deux répertoires dans l'arborescence de Matlab. Les deux répertoires contenant les cibles sont présents sur le disque dans le dossier *Protocole_de_Migration\Installation*. Les deux cibles (*qnx_so* et *win_dll*, voir ANNEXE 3 à ANNEXE 6) doivent être ajoutées dans le répertoire *\$Matlab\$/rtw/c*².

Si la machine d'exécution n'a pas Real-Time Workshop, il faut copier les fichiers source fournis par Real-Time Workshop de manière à permettre la compilation des programmes générés. Le répertoire Matlab pour QNX est présent sur le disque dans le dossier *Protocole_de_Migration\Installation\QNX* et doit être copié à la racine (/) du système de disque.

7.2.3 Initialisation du protocole

Pour ajouter une nouvelle loi de contrôle au contrôleur global, il faut insérer du code permettant au contrôleur global d'utiliser la bibliothèque partagée. Il est à noter que dans

² Notez que *\$Matlab\$* correspond au répertoire racine de Matlab, par exemple « C:\Matlab7\ ».

tous les cas, le contrôleur global doit être recompilé après l'initialisation. Un modèle de code est présenté pour un appel sous Windows à l'ANNEXE 7 et pour QNX à l'ANNEXE 8. Un exemple pour le robot Bart est présenté à l'ANNEXE 11. Notez que la cible QNX génère automatiquement un programme de test dont un exemple est disponible à ANNEXE 13. Le code de ce programme de test généré automatiquement peut être inséré directement dans le contrôleur global.

Sous QNX, il faut modifier le fichier de compilation Microb (*makefile*) en ajoutant le nom de la bibliothèque et également effectuer la compilation initiale du contrôleur global. Par exemple, dans le cas du robot Bart, la bibliothèque doit être ajoutée dans le fichier *makefile* du répertoire racine de Bart à la suite de la liste des autres bibliothèques.

7.2.4 Utilisation du protocole

L'utilisation du protocole s'effectue en six étapes distinctes (voir Figure 11), soit:

1. La modélisation dans l'environnement Simulink;
2. La génération du code;
3. La copie du code de l'environnement de simulation à celui d'exécution;
4. La compilation du code;
5. La copie de la bibliothèque dans l'arborescence du CAR;
6. Le démarrage et la validation du modèle dans le contrôleur global.

Les étapes énumérées ci-dessus sont à répéter jusqu'à l'obtention d'une implantation satisfaisante de la loi de contrôle.

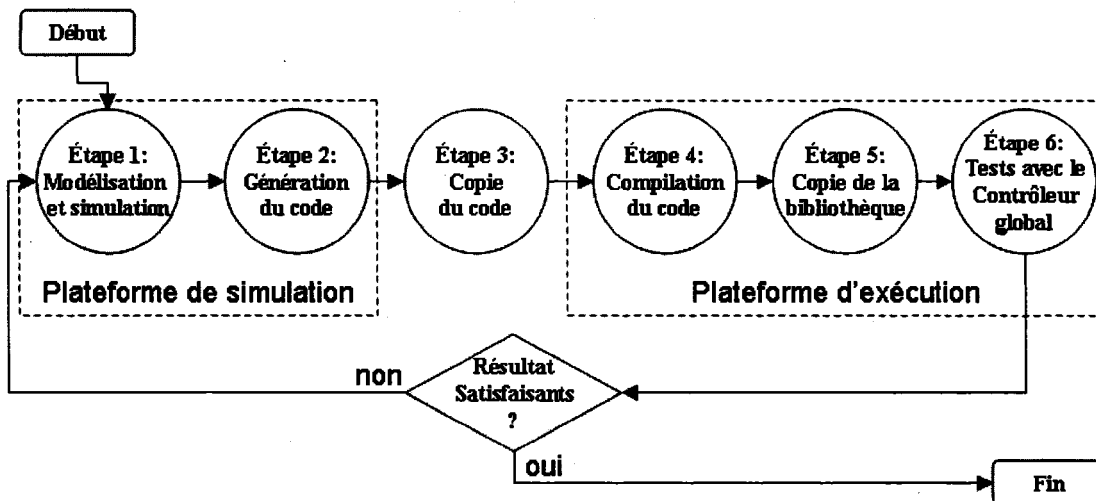


Figure 11 Étapes d'utilisation du protocole de migration

Lors de la modélisation de la loi de contrôle (étape 1), les dimensions de chaque vecteur d'entrées et de sorties doivent être spécifiées à l'intérieur des blocs *inport* et *outport*. De plus, les signaux doivent être identifiés comme étant des données de type double (*Data type = double*) et étant réel (*Signal type = real*) via le menu propriété du bloc (voir Figure 17).

La transmission des signaux de la bibliothèque partagée vers le contrôleur global se fait via un tableau de type *double* et le nombre identifiant le bloc *inport* ou *outport* détermine l'ordre numérique dans ce tableau.

Le modèle doit être configuré pour fonctionner à l'infini avec un pas fixe d'une durée déterminée. Le menu *Simulation / Configuration parameters / Solver* (voir Figure 12) permet de spécifier le temps d'arrêt (*stop time = inf*), le type d'algorithme (*Solver type = Fixed step*) et la période d'échantillonnage (*Fixed step size = 0.01* secondes ou autre). L'algorithme de résolution (*Solver*) utilisé est à la discrétion de l'utilisateur.

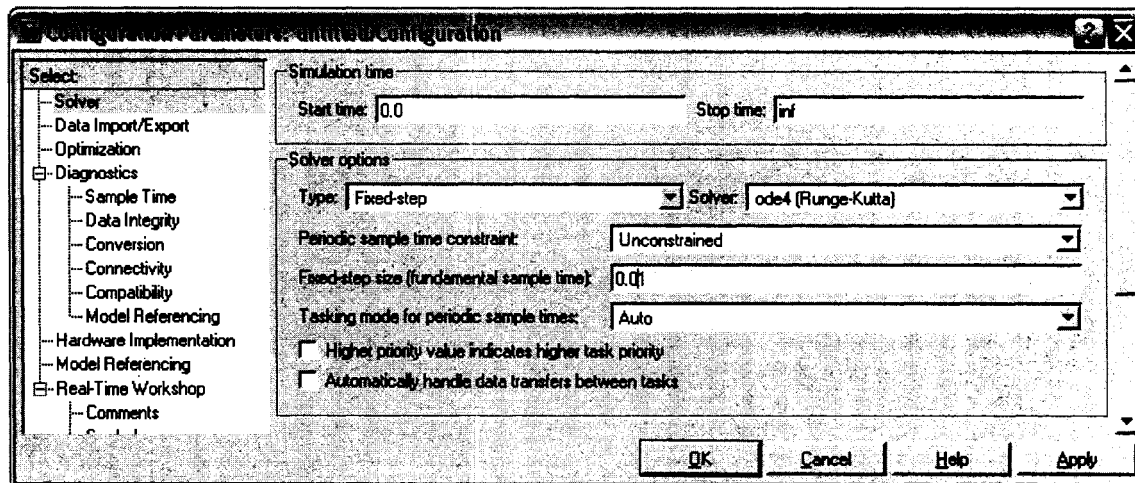


Figure 12 Menu des paramètres de résolution

Migration sous Windows

Pour générer le code et le compiler sous Windows (étapes 2 à 4), il suffit d'aller dans l'interface du modèle Simulink via le menu *Tools / Real-Time Workshop / Options*. Dans l'encadré *Target selection*, il faut appuyer sur le bouton *Browse*, puis choisir dans la liste la cible *win_dll.tlc / Migration Windows (dll)*. Après avoir fermé le menu, la compilation du modèle peut être lancée via le menu *Tools / Real-Time Workshop / Build*. La compilation du modèle devrait se terminer avec succès en affichant dans l'interface Matlab le texte suivant:

```
*** Created dynamic library: $modèle$.dll3
### Successful completion of Real-Time Workshop build procedure for model: <$modèle$>3
```

La bibliothèque dynamique générée sous Windows doit être copiée (étape 5) dans l'arborescence du CAR. Le répertoire par défaut est généralement le suivant:

³ Notez que \$modèle\$ correspond au nom du modèle Simulink, par exemple « CTorque5 ».

`$robot$bin\WindowsNT\PentiumPro`⁴. À ce moment, la loi de contrôle est prête à être testée (étape 6).

Migration de Windows vers QNX

Pour générer le code pour QNX à partir d'une plateforme de simulation Windows (étape 2), il suffit d'aller dans l'interface du modèle Simulink via le menu *Tools / Real-Time Workshop / Options*. Dans l'encadré Target selection, il faut appuyer sur le bouton Browse, puis choisir dans la liste la cible *qnx_so.tlc / QNX Neutrino Real-Time Target (Shared library)*. Après avoir fermé le menu, la génération du code du modèle peut être lancée via le menu *Tools / Real-Time Workshop / Build*. Il est à noter que le code n'est pas compilé et l'interface Matlab affiche le texte suivant:

```
### Make will not be invoked - template makefile is for a different host
### Successful completion of Real-Time Workshop build procedure for model: $modèle$3
```

Le répertoire contenant le code source généré, nommé `$modèle$_qnx_so_rtw`³, doit être copié sur l'ordinateur QNX (étape 3) puis ensuite être compilé (étape 4) en exécutant la commande `make -f $modèle$.mk`³ sous QNX. La compilation du modèle devrait se terminer avec succès en affichant dans le terminal QNX le texte suivant:

```
*** Created executable: $modèle$3
```

La bibliothèque créée nommée `lib$modèle$.so`³ doit être copiée (étape 5) dans l'arborescence du CAR, généralement dans le répertoire `$robot$/lib/QNX6/Pentium`⁴. À ce moment, la loi de contrôle est prête à être testée (étape 6).

⁴ Notez que `$robot$` correspond à la variable d'environnement utilisée par Microb pointant sur le répertoire du contrôleur global courant, par exemple « /Bart » sous QNX ou « C:\Bart » sous Windows.

7.3 Exemple de migration

Un scénario typique de migration est présenté dans cette section pour donner un aperçu des étapes à effectuer pour parvenir à implanter, dans un contrôleur global développé avec le CAR Microb, une loi de contrôle élaborée dans Simulink. Le robot présenté dans cet exemple est également utilisé dans l'évaluation du protocole au CHAPITRE 8.

Dans cet exemple, la plateforme de simulation est différente de la plateforme d'exécution afin d'illustrer le cas le plus complexe. Certains logiciels doivent être présents sur les deux plateformes (voir section 7.2.1). La plateforme de simulation tourne sous Windows XP et la suite complète de Mathworks y est installée, incluant Real-Time Workshop. Sur la plateforme d'exécution tournant sous QNX, le CAR Microb est installé ainsi que le contrôleur global du robot Bart.

Avant d'utiliser le protocole sur un nouvel ordinateur, tant sur la plateforme de simulation que sur la plateforme d'exécution, il est nécessaire de procéder à son installation (voir section 7.2.2). L'installation sur l'ordinateur de simulation requiert que les nouvelles cibles de Real-Time Workshop (*win_dll* et *qnx_so*) soient copiées dans l'arborescence de Matlab. L'installation sur l'ordinateur d'exécution nécessite de copier les répertoires contenant le code source utilisé pour compiler les programmes générés par Real-Time Workshop. Il est également requis d'initialiser des variables d'environnement lorsque le protocole est utilisé sous le système d'exploitation QNX.

Le protocole doit être initialisé (voir section 7.2.3) lorsqu'une nouvelle loi de contrôle est intégrée au contrôleur global, ou encore lorsque les entrées et les sorties de la loi de contrôle sont modifiées. Pour initialiser le protocole, il faut ajouter du code dans le contrôleur global développé à l'aide du CAR Microb (voir ANNEXE 12). De plus, lorsque la plateforme d'exécution fonctionne avec le système d'exploitation QNX, il

faut modifier le fichier de compilation (voir ANNEXE 10) afin qu'il soit en mesure de localiser la bibliothèque lors de la compilation du contrôleur global.

L'utilisation du protocole (voir section 7.2.4), permet d'effectuer les itérations nécessaires pour implémenter la loi de contrôle avec le contrôleur global. Elle rend possible la modification et la validation du modèle de simulation jusqu'à l'obtention de résultats satisfaisants. Cette activité consiste à générer les fichiers en langage C, à les copier, les compiler puis les tester sur la plateforme d'exécution (voir Figure 13). Ces étapes sont répétées jusqu'à l'obtention d'un contrôleur jugé satisfaisant.

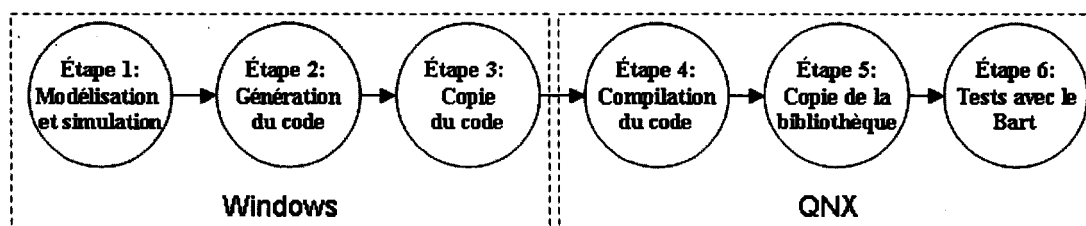


Figure 13 Déroulement de l'implantation d'une loi de contrôle pour Bart

Dans l'exemple présent, le contrôleur à couple précalculé nommé C*Torque4*, dont le schéma est présenté à la Figure 15, est modélisé dans Simulink (voir Figure 16). Les vecteurs d'entrées sont représentés par les blocs Simulink *inport* et sont au nombre de sept ($X, X_p, X_d, X_{pd}, X_{ppd}, k_p, k_d, k_i$). L'unique vecteur de sortie est représenté par le bloc Simulink *outport* nommé Tau. Les paramètres physiques du robot Bart (voir ANNEXE 9) sont encapsulés dans le modèle mais auraient également pu être passée en entrée sous forme de matrice.

Étape 1: Modélisation et simulation

L'objectif poursuivi est de faire migrer une loi de contrôle de Simulink pour l'implanter dans un contrôleur de robot développé sous Microb. Le robot, nommé Bart, est un

manipulateur à deux degrés de liberté dont les deux articulations rotoïdes tel qu'illustré à la Figure 14. Il est couramment utilisé à titre d'exemple dans la littérature [64; 67; 68] pour sa simplicité et sa dynamique bien connue. Les spécifications détaillées de ce robot sont disponibles à l'ANNEXE 9.

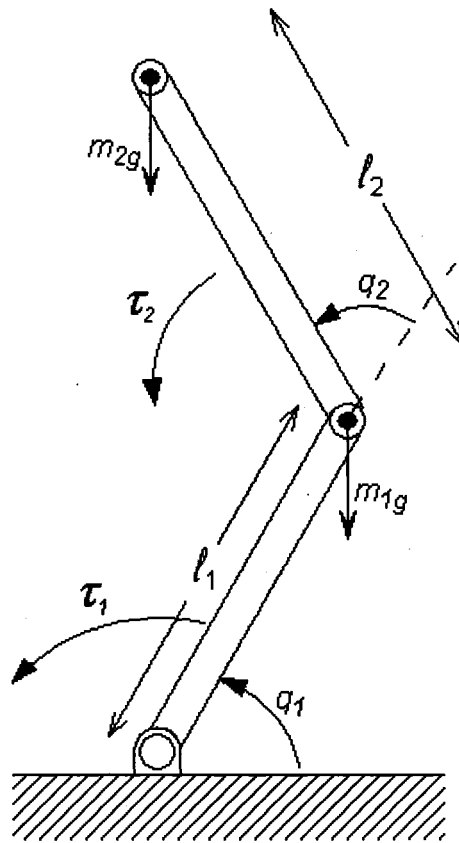


Figure 14 Illustration du robot Bart

La loi de contrôle est un contrôleur à couple précalculé [67]. La loi de contrôle non linéaire a été modifiée afin de lui ajouter un intégrateur. L'usage d'un intégrateur permet de prouver que les états continus et l'intégration temporelle sont bien supportés par la bibliothèque générée.

La loi de contrôle est la suivante:

$$\tau = M(\mathbf{q}) \cdot \mathbf{u} + V(\mathbf{q}, \dot{\mathbf{q}}) + G(\mathbf{q}) + F(\mathbf{q}, \dot{\mathbf{q}})$$

Où:

$M(\mathbf{q})$, $V(\mathbf{q}, \dot{\mathbf{q}})$, $G(\mathbf{q})$ et $F(\mathbf{q}, \dot{\mathbf{q}})$ sont respectivement la matrice de masse ainsi que les vecteurs des forces centrifuges et de Coriolis, de gravité et de frottement. La partie linéaire de la loi de commande (proportionnelle, intégrale et dérivée, avec un terme d'anticipation) est donnée par

$$\mathbf{u} = \ddot{\mathbf{q}}_d + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} + \mathbf{K}_i \int \mathbf{e}$$

\mathbf{q}_d est le vecteur de position désirée

$\dot{\mathbf{q}}_d$ est le vecteur de vitesse désirée

$\ddot{\mathbf{q}}_d$ est le vecteur d'accélération désirée

\mathbf{e} est le vecteur d'erreur sur la position articulaire ($\mathbf{q}_d - \mathbf{q}$)

$\dot{\mathbf{e}}$ est la dérivée du vecteur d'erreur par rapport au temps

$\int \mathbf{e}$ est l'intégrale du vecteur d'erreur par rapport au temps

\mathbf{K}_v , \mathbf{K}_p , \mathbf{K}_i sont les matrices de gains proportionnels, dérivés et intégraux. La valeur de ces gains est respectivement (fixée d'après [68]):

$$\mathbf{K}_v = [37.62 \quad 37.62]$$

$$\mathbf{K}_p = [471.75 \quad 471.75]$$

$$\mathbf{K}_i = [1971.9 \quad 1971.9]$$

Le schéma de la Figure 15, à l'exception du bloc robot, représente le contrôleur que l'on veut encapsuler dans la bibliothèque dynamique afin qu'il retourne le couple τ à partir des signaux d'entrée $\ddot{\mathbf{q}}_d$, $\dot{\mathbf{q}}_d$, \mathbf{q}_d , \mathbf{q} , $\dot{\mathbf{q}}$ ainsi que les gains \mathbf{K}_v , \mathbf{K}_p , \mathbf{K}_i . Le contrôleur global développé avec le CAR Microb a donc la responsabilité de fournir les matrices de gains

(K_v , K_p , K_i), les signaux de consigne (\ddot{q}_d , \dot{q}_d , q_d) ainsi que la lecture des encodeurs (q , \dot{q}).

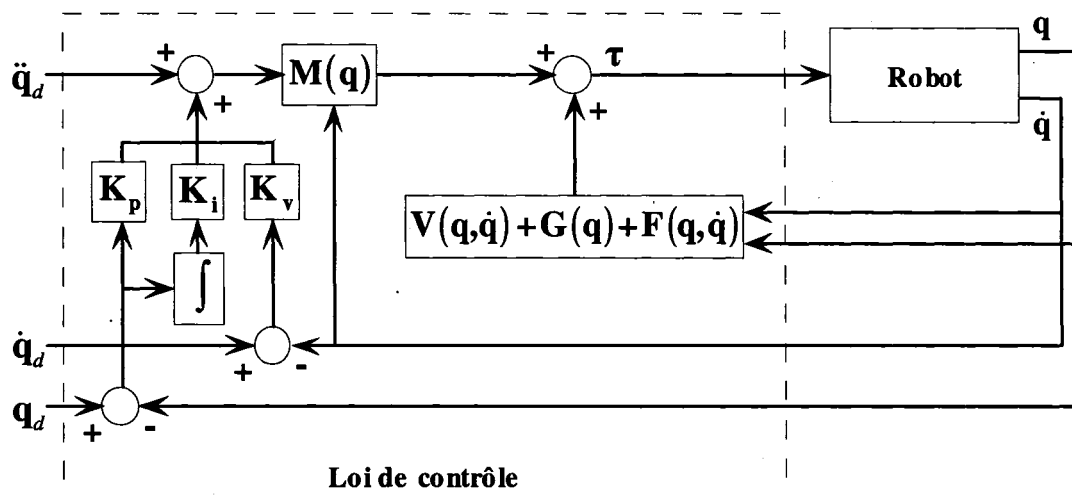


Figure 15 Schéma de la loi de contrôle à couple précalculé (avec PID)

La dimension de chaque vecteur a été spécifiée à l'intérieur des blocs *inport* et *outport* (voir Figure 17). Il est à noter que les signaux ont également été identifiés comme étant des données de type double (*Data type = double*) et qu'il s'agit d'un signal réel (*Signal type = real*).

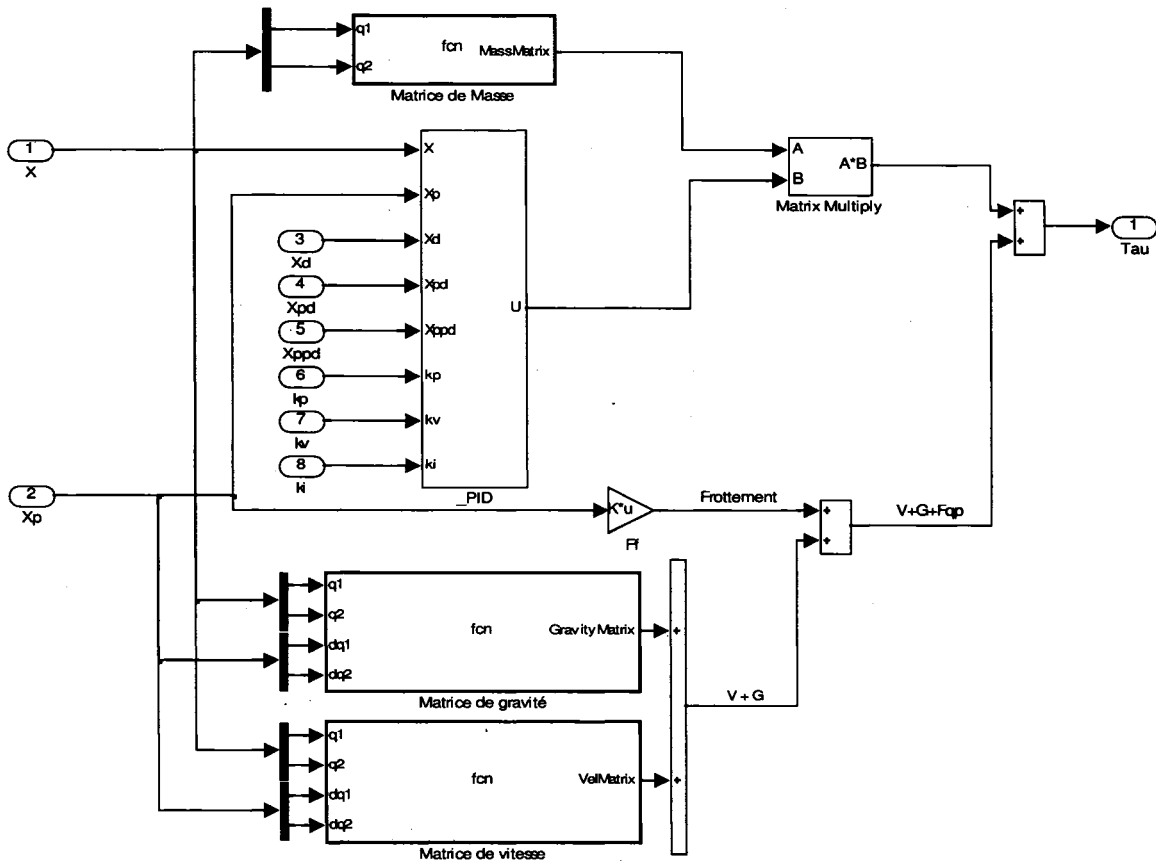


Figure 16 Implémentation sous Simulink de la loi de contrôle

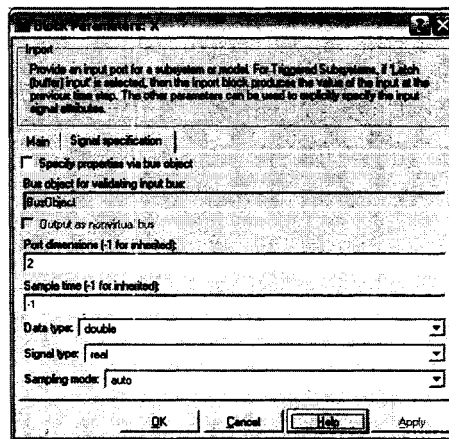


Figure 17 Spécification de la dimension des vecteurs d'entrée et de sortie

Étape 2: Génération du code

Lorsque toutes les dimensions et les types des signaux sont spécifiés, le choix de la cible QNX est fait dans l'interface Simulink via le menu *Tools / Real-Time Workshop / Options* (voir Figure 18). Ensuite le code est généré via la commande Build (voir Figure 19). Il est à noter qu'un programme de test constituant un exemple d'utilisation de code est généré automatiquement pour la plateforme QNX à ce moment (voir ANNEXE 13). Ce code peut également être utilisé pour initialiser le protocole (voir section 7.2.3).

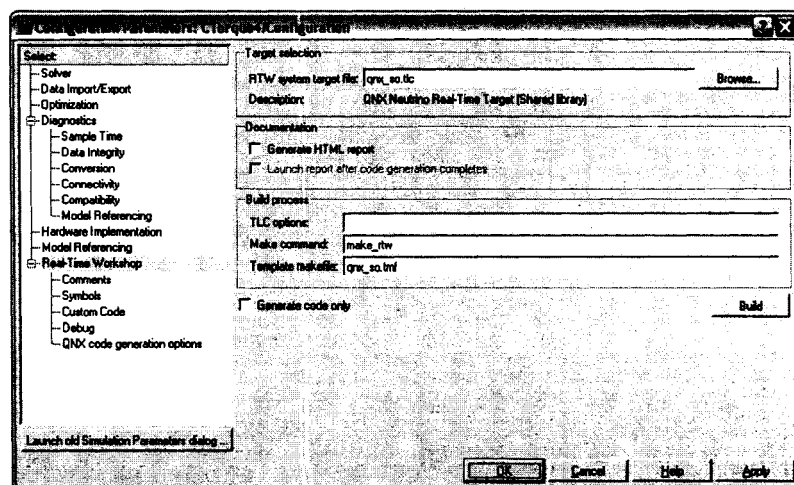


Figure 18 Choix de la cible QNX

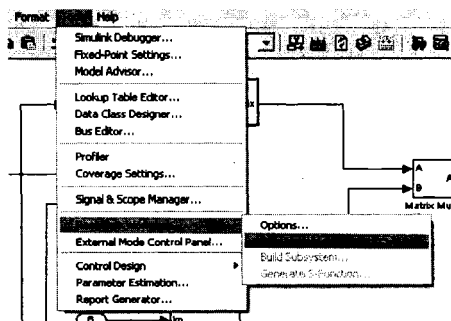


Figure 19 Démarrage de la génération de code

Étape 3: copie du code

Le code ainsi généré contenu dans le répertoire *CTorque4_gnx_so_rtw* doit être copié sur la plateforme QNX.

Étape 4: Compilation du code

La compilation du modèle est lancée dans un terminal dans le répertoire copié à l'aide de la commande *make -f CTorque4.mk*. Cette commande retourne le message « **** Created executable: CTorque4* » lorsqu'elle s'exécute avec succès.

Étape 5: Copie de la bibliothèque

Le fichier généré nommé *libCTorque4.so* doit être copié dans le répertoire du contrôleur Bart (*/Bart/lib/QNX6/Pentium/*).

Étape 6: Tests avec Bart

Le contrôleur global peut être démarré pour tester la nouvelle loi de contrôle ainsi migrée. Si le contrôleur se comporte de façon satisfaisante, le processus est terminé. Dans le cas contraire, l'utilisateur retourne à l'étape 1 pour effectuer la modélisation et la simulation de la loi de contrôle.

En résumé, lorsque les logiciels préalables sont présents sur l'ordinateur effectuant la migration (voir section 7.2.1), il est indispensable d'installer les fichiers requis par le protocole (voir section 7.2.2). Puis lorsqu'une nouvelle loi de contrôle doit être implantée, il est nécessaire de réinitialiser le contrôleur global (voir section 7.2.3). Lorsque ces deux étapes sont franchies, il est possible d'utiliser le protocole pour effectuer l'activité de conception proprement dite (voir section 7.2.4).

7.4 Conclusion

La présentation d'un scénario typique de migration a permis de mettre en contexte les étapes d'installation, d'initialisation et d'utilisation du protocole présentées dans ce chapitre. Le présent chapitre constitue donc le manuel de l'utilisateur du protocole. L'évaluation du protocole est effectuée au chapitre suivant.

MCours.com