

**Arts et Métiers ParisTech - Sorbonne Université**  
**Bi-cursus : Diplôme d'Ingénieur Arts et Métiers**  
**& Master Sciences et Technologies - Mention Sciences de l'Ingénieur**  
**Parcours "Systèmes Avancés et Robotique"**

## Stage de Fin d'Etudes

travail réalisé par

**Rémy Agneray**

---

Commande prédictive basée modèle - Application  
à l'évitement d'incompatibilité de contrainte

---

*Année universitaire 2017 - 2018*



### Encadrement

M Vincent PADOIS

Tuteur

M Nazih MECHBAL

Professeur responsable



# Remerciements

## Résumé

# Table des matières

<b>Remerciements</b>	<b>iii</b>
<b>Résumé</b>	<b>iv</b>
<b>Table des figures</b>	<b>vii</b>
<b>Nomenclature</b>	<b>viii</b>
<b>1. Introduction</b>	<b>1</b>
1.1. Contexte et applications . . . . .	1
1.2. Etat de l'art : méthode d'évitement d'incompatibilité . . . . .	2
1.2.1. Sébastien Rubrecht : Contributions to the control of constrained robots . . . . .	3
1.2.2. Anis Meguenani : Safe Control of Robotic Manipulators in Dynamic Control . . . . .	3
1.3. Commande prédictive : principe et intérêt . . . . .	4
1.4. Plan du rapport . . . . .	5
<b>2. Incompatibilité des contraintes : étude de cas</b>	<b>6</b>
2.1. Formulation des contraintes . . . . .	6
2.2. Formulation de la commande réactive . . . . .	7
2.3. Exemple de situation de compatibilité . . . . .	8
2.4. Exemple de situation d'incompatibilité . . . . .	8
<b>3. Formulation et implémentation de la commande prédictive</b>	<b>10</b>
3.1. Formulation de la commande . . . . .	10
3.1.1. Hypothèse de constance des termes dynamiques . . . . .	10
3.1.2. Formulation des contraintes . . . . .	11
3.1.3. Formulation réduite du problème de commande . . . . .	11
3.2. Choix de conception pour la simulation . . . . .	12
3.2.1. Environnement et extensions utilisés . . . . .	12
3.2.2. Formulation de la tâche . . . . .	13
3.2.3. Principe de fonctionnement de l'algorithme . . . . .	14
3.3. Exemple d'application de la commande prédictive . . . . .	14
<b>4. Comparaison réactif/prédictif : évitement des incompatibilités</b>	<b>16</b>
4.1. Principe des tests . . . . .	16
4.2. Résultats . . . . .	16
4.2.1. Robot R plan . . . . .	16
4.2.2. Robot RR plan . . . . .	17

4.2.3. Robot PUMA 560 . . . . .	17
4.3. Conclusion sur l'évitement d'incompatibilité . . . . .	18
<b>5. Comparaison réactif/prédictif : compléments</b>	<b>19</b>
5.1. Evitement d'obstacle opérationnel . . . . .	19
5.1.1. Formulation de la contrainte opérationnelle . . . . .	19
5.1.2. Tests d'évitement . . . . .	20
5.2. Aspect énergétique . . . . .	20
5.2.1. Critère d'évaluation considéré . . . . .	20
5.2.2. Résultats et analyse . . . . .	20
<b>6. Pistes d'optimisation du coût de calcul</b>	<b>21</b>
6.1. Intervalle d'amplitude de l'horizon de temps . . . . .	21
6.1.1. Nécessité d'évitement des incompatibilités et borne inférieure . . . . .	21
6.1.2. Borne supérieur et respect de l'hypothèse de constance des termes dynamiques . . . . .	21
6.1.3. Conclusion sur la méthode proposée de définitions des bornes de l'amplitude . . . . .	21
6.2. Choix des solveurs . . . . .	21
6.2.1. Etude des performances des solveurs . . . . .	21
6.2.2. Conclusion sur le choix des solveurs . . . . .	21
<b>7. Conclusion</b>	<b>22</b>
<b>A. Annexe : formulation complète de la commande prédictive</b>	<b>24</b>
<b>B. Annexe : définitions des suites des prédictions de vitesse et de position</b>	<b>25</b>
<b>C. Annexe : définition de la suite des prédictions de la distance au mur</b>	<b>26</b>
<b>D. Annexe : développement du calcul de la contrainte opérationnelle</b>	<b>28</b>
<b>E. Annexe : paramètres par défaut de la simulation</b>	<b>30</b>

# Table des figures

1.1. Illustration schématique de la compatibilité de contraintes . . . . .	2
1.2. Illustration schématique de l'apparition d'incompatibilité . . . . .	2
1.3. Représentation schématique des butées virtuelles . . . . .	3
1.4. Illustration de la commande prédictive . . . . .	4
2.1. Modèle du robot R plan utilisé . . . . .	6
2.2. Evolution temporelle des limites les plus restrictives . . . . .	8
2.3. Représentation des intervalles de contrainte . . . . .	8
2.4. Inversion des limites lors de la situation d'incompatibilité . . . . .	9
2.5. Représentation dans le cas l'incompatibilité observée en simulation . . . . .	9
3.1. Modèles de robot étudiés en simulation . . . . .	13
3.2. Evolution temporelle des limites les plus restrictives - Commande prédictive	14
3.3. Schéma de principe de l'algorithme de simulation . . . . .	15
4.1. Résultats des test d'évitement d'incompatibilité (robot R plan) . . . . .	17
4.2. Résultats des test d'évitement d'incompatibilité (robot RR plan) . . . . .	17
4.3. Résultats des test d'évitement d'incompatibilité (PUMA 560) . . . . .	18

# Nomenclature



# 1. Introduction

La commande constitue l'une des problématique de la robotique, car celle-ci diffère selon l'environnement du robot et selon l'utilisation à laquelle il doit répondre.

## 1.1. Contexte et applications

Dans le cadre de la robotique industrielle de manipulation, les séquences de commande sont précalculées avant le début de fonctionnement du robot. En effet, puisque les tâches constituent un cycle qui ne change pas pendant le fonctionnement, cela permet de déterminer à l'avance les trajectoires à répéter, et par conséquent les commandes. Ces calculs ont pour objectif d'optimiser le temps d'opération et tiennent compte des contraintes intrinsèques au robot (butées, capacités des actionneurs) et de l'environnement (supposé connu et sans évolution pendant le fonctionnement).

Cependant, dans des applications telles que le télé-opération, la robotique collaborative, ou la robotique mobile, les tâches peuvent évoluer pendant le fonctionnement et ne sont pas connues à l'avance. Par conséquent, il est impossible de définir à priori une séquence de commande permettant de répondre aux tâches et s'assurant du respect des contraintes intrinsèques au robot et liées à l'environnement, pendant un horizon de temps infini.

Par exemple, dans le cas d'une plate-forme devant suivre le déplacement d'un humain, le robot doit en permanence estimer la position de sa cible, mais doit aussi être capable de détecter des obstacles dans son environnement en s'assurant la capacité de freinage nécessaire (déterminée par les actionneurs du robot) pour les éviter.

La commande doit alors être réactive, c'est-à-dire être calculée pendant le fonctionnement, pour pouvoir s'adapter à l'évolution de la tâche. Elle doit aussi tenir compte, lors du calcul, des contraintes du robot et de l'environnement. Cependant, l'un des problèmes de la commande réactive est d'assurer, pendant le fonctionnement, la compatibilité des contraintes.

En effet, les contraintes définissent des sous-espaces en associant à l'espace des commandes possibles des conditions sur les valeurs : cela restreint les solutions au problème de commande. Il y a compatibilité de contrainte à un instant donné, lorsque tous les sous-espaces ont une intersection commune à cet instant. Cela est illustré en figure 1.1. La conséquence directe de la compatibilité de contrainte est l'existence d'au moins une solution au problème de commande à l'instant donné, qui respecte les conditions associées aux contraintes.

Par opposition, il y a incompatibilité de contraintes à un instant donné, lorsqu'il n'y a pas d'intersection commune à tous les sous-espaces associés aux contraintes (figure 1.1). La conséquence est alors l'absence de solution au problème de commande respectant les contraintes.

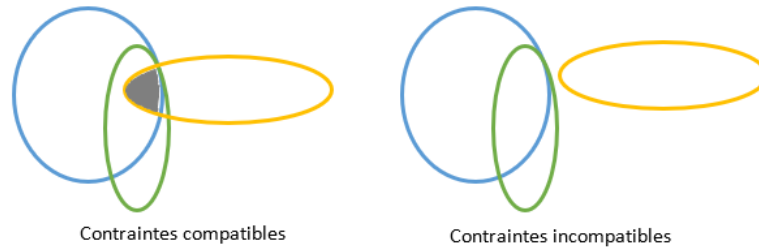


FIGURE 1.1. – Illustration schématique de la compatibilité de contrainte. Chaque ellipse représente une contrainte, et contient les solutions du problème de commande respectant la contrainte. La zone grisée dans le cas des contraintes compatibles représente l'intersection de toutes les ellipses représentant les contraintes.

En commande réactive, il est possible que le robot puisse se retrouver à un instant dans une situation d'incompatibilité, bien que les contraintes aient été compatibles aux instants précédents. En effet, comme illustré en figure 1.2, les sous-espaces associés aux contraintes sont variables avec le temps. Par conséquent, l'intersection commune à tous les sous-espaces est modifiée pendant le fonctionnement du robot, et peut même ne plus exister. Il est alors impossible de calculer une commande à appliquer à partir de cet instant.

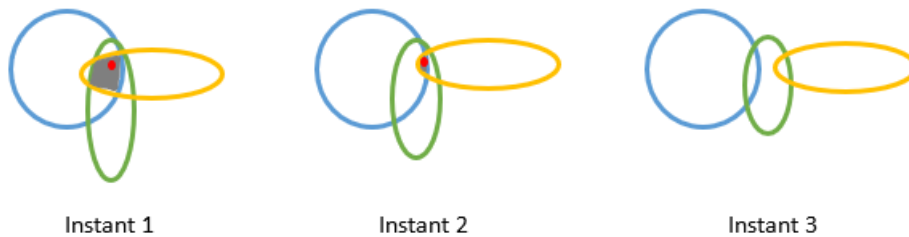


FIGURE 1.2. – Illustration schématique de l'apparition d'incompatibilité lors d'un fonctionnement. Le point rouge représente la solution au problème de commande, choisie à l'intérieur de l'intersection commune. Pendant les différents instants, les sous-espaces varient. Cela peut amener à faire disparaître l'intersection. Il est alors impossible de choisir une solution au problème de commande respectant toutes les contraintes.

## 1.2. Etat de l'art : méthode d'évitement d'incompatibilité

Afin de pouvoir calculer une solution au problème de commande réactive sous contraintes, il est nécessaire d'éviter l'apparition de situation d'incompatibilité. Des recherches ont été menées dans ce sens.

### 1.2.1. Sébastien Rubrecht : Contributions to the control of constrained robots

Dans sa thèse [1], Sébastien Rubrecht considère des contraintes sur l'accélération, la vitesse et la position articulaire. En utilisant le lien temporel entre ces trois grandeurs, les contraintes sont alors reformulées, afin de prendre en compte la distance aux butées et les capacités de décélération.

Ces réécritures peuvent alors être représentées par des "butées virtuelles", recalculées à chaque commande et plus restrictive que les butées intrinsèques au robot (1.3). Elle empêche notamment de s'approcher des butées avec une vitesse trop élevée, où les capacités de freinage seraient insuffisantes pour respecter la contrainte de butée. Le but est alors d'empêcher une incompatibilité entre la contrainte de position (butées intrinsèques) et la contrainte d'accélération (capacités de freinage).

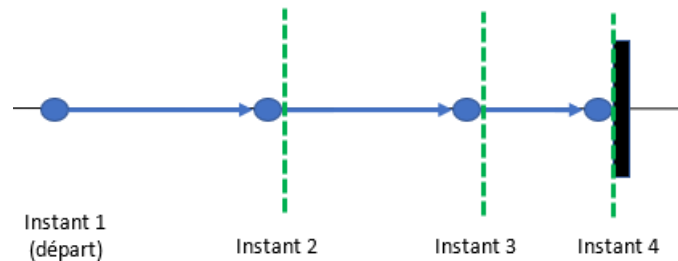


FIGURE 1.3. – Représentation schématique des butées virtuelles dans un cas simple. Les états successifs du robot sont représentés en bleu, la butée intrinsèque au robot est en noir, et les butées virtuelles sont en vert. A chaque instant, la butée virtuelle empêche les états de s'approcher trop rapidement de la butée. Dès que le robot arrive à sa butée, la butée virtuelle se confond alors avec cette dernière.

Cependant, le principal problème de cette approche provient de l'hypothèse sur la constance des capacités de décélération sur un horizon de temps infini. En effet, celles-ci dépendent de la configuration actuelle du robot, et sont donc variables. Il est alors possible que dans une configuration donnée, le robot ne puisse pas freiner suffisamment pour respecter ses butées, alors que le calcul de la butée virtuelle ne détecte pas de problème. La compatibilité entre les contraintes de position et d'accélération n'est donc pas assurée sur un horizon de temps infini.

### 1.2.2. Anis Meguenani : Safe Control of Robotic Manipulators in Dynamic Control

En reprenant la même approche que Sébastien Rubrecht, Anis Meguenani [2] considère en plus une contrainte sur le couple articulaire. Cette variable est alors liée à l'accélération grâce au modèle dynamique du robot.

Les contraintes sur le couple servant dans ce cadre à définir les capacités de freinage. Celles-ci sont alors supposées constantes et connues sur un horizon de temps infini. L'hypothèse ainsi formulée est alors plus forte que celle de Sébastien Rubrecht, car les valeurs maximales et minimales de couple sont cette fois-ci liées aux actionneurs.

Cette reformulation a donc pour but d'éviter l'incompatibilité entre les contraintes de couples et de position. Cependant, le problème posé par cette approche n'est plus l'hypothèse de constance des capacités de freinage sur un horizon de temps infini, mais le coût de calcul de la commande. De plus, il subsiste des problèmes de compatibilité liées à la configuration du robot dans l'espace opérationnel.

### 1.3. Commande prédictive : principe et intérêt

Le travail sur un horizon de temps infini pose problème dans les approches développées en section 1.2. Dans celle de Sébastien Rubrecht, une hypothèse de base peut ne pas être valable en horizon de temps infini, et, dans celle d'Anis Meguenani, le coût de calcul est trop important. L'approche qui pourrait être développée serait alors de travailler sur des horizons de temps fini.

Or, l'utilisation d'un horizon de temps est un des éléments utilisés par la commande prédictive, ou Model Predictive Control (MPC). Ce type de commande se base sur le calcul de prévision de commande respectant des prévisions de contraintes sur un horizon de temps glissant (figure 1.4). Une modélisation explicite est nécessaire pour formuler cette commande.

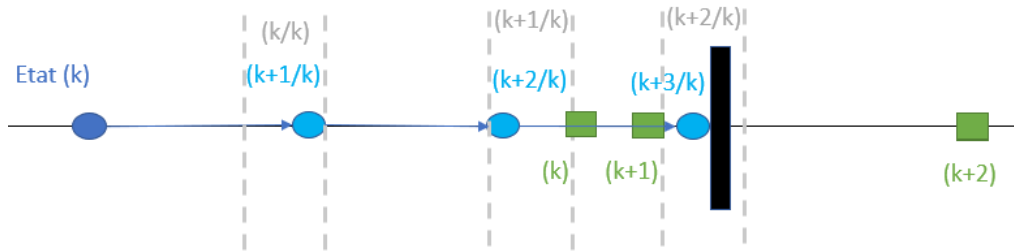


FIGURE 1.4. – Illustration de la commande prédictive calculée à l'instant  $k$ . La figure présente le calcul des prévisions, notées  $k + \dots/k$  sur un horizon de temps. Les états sont représentés en bleu, les objectifs en vert et la butée du robot en noir. En gris sont représentés les intervalles définissant les possibilités sur les prévisions suivantes en considérant les contraintes de décélération.

Il est important de noter que les prévisions de commande ne forment pas obligatoirement une séquence de commande qui sera appliqué à partir de l'instant de calcul. Seule la première prévision sera appliquée en tant que commande lors de l'instant de calcul. Par exemple, le calcul d'un horizon de commande à une itération donne comme résultat la séquence  $(a, b, c, d)$ . La commande appliquée est le premier élément de la séquence, ici  $a$ . Cela résulte d'un changement dans l'état du robot, et, à partir de celui-ci, la commande est recalculée. Cela signifie que les prévisions  $b$ , puis  $c$ , puis  $d$  ne sont pas appliquées en tant que commande.

L'intérêt de la commande prédictive est donc de calculer sur un horizon de temps fini et glissant, ce qui permettra de calculer explicitement les contraintes.

## 1.4. Plan du rapport

Ce rapport comporte sept chapitres.

Le deuxième concerne une étude de cas d'incompatibilité de contraintes en commande réactive, et contient notamment la formulation des contraintes utilisées.

Le troisième chapitre présente la formulation du problème de commande prédictive, ainsi que les choix de conception pour la simulation.

Le chapitre 4 explique la mise en oeuvre des tests de comparaison entre la commande réactive et la commande prédictive, pour l'évitement des incompatibilités de contraintes.

Le cinquième chapitre s'intéresse à d'autres axes de comparaisons entre les deux commandes : l'évitement d'obstacle opérationnel et l'analyse de l'aspect énergétique.

Le chapitre 6 propose des pistes d'optimisation du temps de calcul sur l'amplitude de l'horizon de temps et sur le choix des solveurs.

Le dernier chapitre conclue sur les performances de la commande prédictive mise en place, et discute des améliorations possibles.

## 2. Incompatibilité des contraintes : étude de cas

Afin d'illustrer l'incompatibilité de contrainte, ce chapitre va présenter un exemple d'une telle situation. Pour cela, une simulation d'une commande réactive d'un bras robotique R plan est mise en place. Le robot n'est constitué d'une unique liaison pivot. L'objectif est de suivre une trajectoire certes prédéfinie, mais calculée sans tenir compte de contrainte et envoyée au fur et à mesure pendant la simulation.

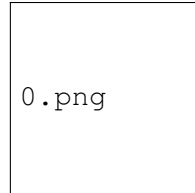


FIGURE 2.1. – Modèle du robot R plan utilisé.

### 2.1. Formulation des contraintes

Les contraintes liées aux capacités des actionneurs et aux butées sont celles considérées. Par conséquent, les grandeurs contraintes sont le couple articulaires  $\tau$  et la position articulaire  $q$ . Ils est alors possible d'écrire, sous forme discrète (à une itération  $k$ ), les formulations naïves pour les contraintes considérées :

$$\forall k \in \{k_0, \dots, k_{end}\},$$

$$\tau_{min} \leq \tau(k) \leq \tau_{max} \quad (2.1)$$

$$q_{min} \leq q(k) \leq q_{max} \quad (2.2)$$

Or, avec un pas de temps  $\Delta t$  suffisamment petit, il est possible d'exprimer  $q(k+1)$  grâce à un développement limité. Soit  $\dot{q}$ , la vitesse articulaire, et  $\ddot{q}$  l'accélération articulaire :

$$\forall k \in \{k_0, \dots, k_{end} - 1\}, q(k+1) = q(k) + \Delta t \dot{q}(k) + \frac{1}{2} \Delta t^2 \ddot{q}(k) \quad (2.3)$$

En s'inspirant de l'approche de S. Rubrecht, cela permet d'écrire l'équation 2.2 en termes d'accélération :

$$\forall k \in \{k_0, \dots, k_{end}\}, \ddot{q}_{min,pos}(k) \leq \ddot{q}(k) \leq \ddot{q}_{max,pos}(k) \quad (2.4)$$

avec les limites d'accélération définies telles que :

$$\ddot{q}_{min,pos}(k) = \frac{2}{\Delta t^2} (q_{min} - q(k) - \Delta t \dot{q}(k)) \quad (2.5)$$

$$\ddot{q}_{max,pos}(k) = \frac{2}{\Delta t^2} \left( q_{max} - \underline{q}(k) - \Delta t \dot{q}(k) \right) \quad (2.6)$$

Cela permet donc d'exprimer des limites d'accélération tenant compte de l'état actuel (position et vitesse) et en fonction de la distance à la butée.

L'équation 2.8 peut de plus, en suivant l'approche d'A. Meguenani, être exprimés en termes de couple, à partir du modèle dynamique de l'équation 2.7.

$$\forall k \in \{k_0, \dots, k_{end}\}, \tau(k) = M(\underline{q}(k)) \ddot{q}(k) + \underline{b}(\underline{q}(k), \dot{q}(k)) + \underline{g}(\underline{q}(k)) \quad (2.7)$$

Avec  $M$ , la matrice d'inertie du robot,  $\underline{b}$ , le vecteur des effets centrifuges et de Coriolis, et  $\underline{g}$  le vecteur des effets dus à la gravité.

Par conséquent :

$$\forall k \in \{k_0, \dots, k_{end}\}, \tau_{min,pos}(k) \leq \tau(k) \leq \tau_{max,pos}(k) \quad (2.8)$$

Avec :

$$\tau_{min,pos}(k) = M(\underline{q}(k)) \ddot{q}_{min,pos}(k) + \underline{b}(\underline{q}(k), \dot{q}(k)) + \underline{g}(\underline{q}(k)) \quad (2.9)$$

$$\tau_{max,pos}(k) = M(\underline{q}(k)) \ddot{q}_{max,pos}(k) + \underline{b}(\underline{q}(k), \dot{q}(k)) + \underline{g}(\underline{q}(k)) \quad (2.10)$$

## 2.2. Formulation de la commande réactive

Les contraintes formulées en équation 2.1 et 2.8 sont utilisées dans le problème de commande réactive. Celui-ci est défini comme une optimisation quadratique, avec le couple articulaire comme variable d'optimisation.

La fonction de coût à optimiser est la distance quadratique entre l'accélération cartésienne  $\ddot{x}$  de l'effecteur (extrémité du robot) et l'accélération désirée  $\ddot{x}_{des}$ . Or, la dérivée du modèle cinématique permet d'avoir un lien entre les accélérations cartésiennes et opérationnelles.

$$\ddot{x}(k) = J(\underline{q}(k)) \ddot{q}(k) + \dot{J} \dot{q}(k) \quad (2.11)$$

$J$  et  $\dot{J}$  sont alors respectivement la matrice Jacobienne du robot et sa dérivée.

Avec le modèle dynamique de l'équation 2.7, le problème de commande peut donc s'écrire :

$$\forall k \in \{k_0, \dots, k_{end}\} :$$

$$\begin{aligned} \tau^*(k) = \underset{\tau}{argmin} \quad & \frac{1}{2} \omega_k \| J(\underline{q}(k)) M^{-1}(\underline{q}(k)) (\tau(k) - (\underline{b}(\underline{q}(k), \dot{q}(k)) + \underline{g}(\underline{q}(k)))) + \dot{J}_k \dot{q}_k - \ddot{x}_{des}(k) \|_2^2 \\ & + \frac{1}{2} \epsilon \| \tau(k) - \underline{g}(\underline{q}(k)) \| \end{aligned} \quad (2.12)$$

contraint par :

$$\begin{aligned} \tau_{min} &\leq \tau(k) \leq \tau_{max} \\ \tau_{min,pos}(k) &\leq \tau(k) \leq \tau_{max,pos}(k) \end{aligned}$$

### 2.3. Exemple de situation de compatibilité

Une première simulation de la commande formulée en section 2.2 est exécutée dans un cas de compatibilité de contrainte, afin d'avoir une expérience témoin. L'objectif est défini tel que le robot doit rejoindre sa butée  $\underline{q}_{max} = \frac{\pi}{2}$ . Le temps de simulation est alors de 12 s.

Dans l'étude de cas, il est possible de définir les limites de couples les plus restrictives :  $\forall k \in \{k_0, \dots, k_{end}\}$  :

$$\tau_{low}(k) = \max(\tau_{min,pos}(k), \tau_{min}) \quad (2.13)$$

$$\tau_{up}(k) = \min(\tau_{max,pos}(k), \tau_{max}) \quad (2.14)$$

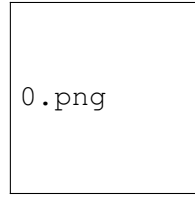


FIGURE 2.2. – Evolution temporelle des limites les plus restrictives.

La figure 2.2 montre l'évolution de  $\tau_{low}$  et de  $\tau_{up}$  pendant la simulation, où aucun cas d'infaisabilité n'est observé. En effet, puisque, pendant toute la simulation :

$$\forall k \in \{k_0, \dots, k_{end}\}, \tau_{low}(k) \leq \tau_{up}(k)$$

il existe donc des solutions possibles pour le problème de commande réactive. La figure 2.3 illustre la superposition des intervalles au cours de la simulation.

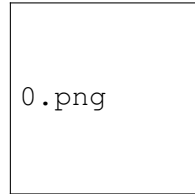


FIGURE 2.3. – Représentation des intervalles de contrainte pendant la simulation sans incompatibilité. Les zones en rouges sont interdites par les contraintes. Les intervalles de solutions possibles se superposent donc pendant la simulation.

### 2.4. Exemple de situation d'incompatibilité

Pour forcer l'apparition d'une incompatibilité, l'objectif est défini en dehors des butées, tel que  $\underline{q}_{end} = \pi \notin [\underline{q}_{max}, \underline{q}_{min}] = [-\frac{\pi}{2}, \frac{\pi}{2}]$ .

Les résultats présentés en figure 2.4 montrent une inversion des limites les plus contraignantes, ayant pour conséquence l'incompatibilité de contrainte. Dans ce cas, puisque  $\tau_{up}(t = 4.78s)$  n'est pas constant et différent de  $\tau_{max}$  :



$$\tau_{max,pos}(t = 4.78s) < \tau_{min}$$

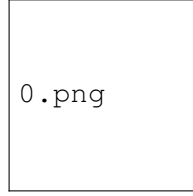


FIGURE 2.4. – Inversion des contraintes les plus restrictives lors de la situation d'incompatibilité.

Cela peut être illustré comme en figure 2.5, et signifie que le deuxième axe se trouve dans un état proche de sa butée inférieure, mais ne possède pas les capacités de décélération nécessaire assurant que celle-ci ne sera pas franchie. Pour le démontrer :

$$\begin{aligned} & \text{Pour } t = 4.78 \text{ s, soit pour } k_1 = 478, \tau_{min} > \tau_{max,pos}(k_1) \\ \Leftrightarrow & \frac{\Delta t^2}{2} (M^{-1}(\underline{q}(k_1)) \tau_{min} - \underline{b}(\underline{q}(k_1), \underline{\dot{q}}(k_1)) - \underline{g}(\underline{q}(k_1))) + \underline{q}(k) + \Delta t \underline{\dot{q}}(k_1) > \underline{q}_{max} \\ & \Leftrightarrow \underline{q}(k_1 + 1) > \underline{q}_{max} \end{aligned}$$

La prochaine position de l'axe franchit donc bien sa butée, même si la capacité de décélération maximale est appliquée en commande.

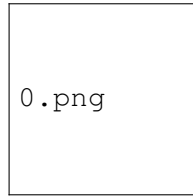


FIGURE 2.5. – Représentation des intervalles de contraintes dans le cas l'incompatibilité observée en simulation

## 3. Formulation et implémentation de la commande prédictive

Afin d'éviter des incompatibilités comme celle mise en évidence en section 2.4, l'objectif est de formuler et de construire une simulation d'une commande prédictive.

### 3.1. Formulation de la commande

La commande prédictive en couple va être mise en place, avec des expressions de contraintes reprenant les équations 2.1 et 2.8. Le but est de calculer un horizon de couple articulaire, afin d'approcher un objectif donné, tout en respectant les contraintes.

Comme pour la figure 1.4, la notation  $(k + i/k)$  désigne la  $i$ -ème prévision d'une grandeur dans l'horizon de temps calculé à partir de l'itération  $k$ .

#### 3.1.1. Hypothèse de constance des termes dynamiques

Afin de pouvoir écrire le problème de commande explicitement, certaines grandeurs seront considérées constantes dans l'horizon de temps. Cela n'est valable que si l'amplitude de l'horizon de temps est faible, et que si le pas de temps entre deux prévisions est suffisamment petit. La variation des termes dynamiques est alors négligeable. Par conséquent :

$$\forall k \in \{k_0, \dots, k_{end}\}, \forall i \in \{0, \dots, N\}, N \in \mathbb{N}^*$$

$$\begin{aligned} M(\underline{q}(k + i/k)) &= M_k, \\ \underline{b}(\underline{q}(k + i/k), \underline{\dot{q}}(k + i/k)) &= \underline{b}_k, \\ \underline{g}(\underline{q}(k + i/k)) &= \underline{g}_k, \\ \underline{J}(\underline{q}(k + i/k)) &= \underline{J}_k, \\ \underline{\dot{J}}(\underline{q}(k + i/k)) &= \frac{\partial \underline{J}}{\partial t}(\underline{q}(k + i/k)) = \underline{\dot{J}}_k \\ \underline{\ddot{J}}(\underline{q}(k + i/k)) \underline{\dot{q}}(k + i/k) &= \underline{\ddot{J}}_k \underline{\dot{q}}_k \end{aligned}$$

Une des conséquences de cette hypothèse est le fait qu'il n'est plus nécessaire de recalculer les termes dépendants de la position et de la vitesse articulaire à chaque prévision. Il est donc possible d'écrire les relations linéaires suivantes (le détail des calculs pour les prévisions de vitesse et de position est en annexe B).

$$\forall k \in \{k_0, \dots, k_{end}\}, N \in \mathbb{N}^* :$$

$$\underline{\tau}(k + i/k) = M_k^{-1} \underline{\ddot{q}}(k + i/k) + \underline{b}_k + \underline{g}_k, \quad i \in \{0, \dots, N\} \quad (3.1)$$

$$\underline{q}(k + i/k) =$$

$$\begin{cases} \underline{q}(k), & i = 0 \\ \underline{q}(k) + i \Delta t \dot{\underline{q}}(k) + \Delta t^2 M_k^{-1} \sum_{j=1}^i \left( \left( \frac{1}{2} + i - j \right) (\underline{\tau}(k + j - 1/k) - \underline{b}_k - \underline{g}_k) \right), & i \in \{1, \dots, N\} \end{cases} \quad (3.2)$$

$$\begin{aligned} & \dot{\underline{q}}(k + i/k) = \\ & \begin{cases} \dot{\underline{q}}(k), & i = 0 \\ \dot{\underline{q}}(k) + \Delta t M_k^{-1} \sum_{j=1}^i (\underline{\tau}(k + j - 1/k)) - i \Delta t M_k^{-1} (\underline{b}_k + \underline{g}_k), & i \in \{1, \dots, N\} \end{cases} \end{aligned} \quad (3.3)$$

### 3.1.2. Formulation des contraintes

Avant de formuler le problèmes de commande, les contraintes sont adaptées pour le calcul des prévisions.

$$\forall k \in \{k_0, \dots, k_{end}\}, N \in \mathbb{N}^*$$

$$\underline{\tau}_{min} \leq \underline{\tau}(k + i/k) \leq \underline{\tau}_{max}, \quad i \in \{0, \dots, N\} \quad (3.4)$$

$$\underline{q}_{min} \leq \underline{q}(k + i/k) \leq \underline{q}_{max}, \quad i \in \{1, \dots, N\} \quad (3.5)$$

Les relations établies en section 3.1.1 vont permettre d'adapter l'écriture de l'équation 3.5.

$$\forall k \in \{k_0, \dots, k_{end}\}, \forall i \in \{1, \dots, N\}, N \in \mathbb{N}^*$$

$$\underline{q}_{tor,min}(k + i/k) \leq \Delta t^2 M_k^{-1} \sum_{j=1}^i \left( \left( \frac{1}{2} + i - j \right) \underline{\tau}(k + j - 1/k) \right) \leq \underline{q}_{tor,max}(k + i/k) \quad (3.6)$$

avec :

$$\underline{q}_{tor,min}(k + i/k) = \underline{q}_{min} - \underline{q}(k) - i \Delta t \dot{\underline{q}}(k) + \Delta t^2 \sum_{j=1}^i \left( \frac{1}{2} + i - j \right) M_k^{-1} (\underline{b}_k + \underline{g}_k) \quad (3.7)$$

$$\underline{q}_{tor,max}(k + i/k) = \underline{q}_{max} - \underline{q}(k) - i \Delta t \dot{\underline{q}}(k) + \Delta t^2 \sum_{j=1}^i \left( \frac{1}{2} + i - j \right) M_k^{-1} (\underline{b}_k + \underline{g}_k) \quad (3.8)$$

### 3.1.3. Formulation réduite du problème de commande

La formulation retenue pour la commande prédictive est une écriture réduite, la réduisant à un problème d'optimisation quadratique cherchant à minimiser l'écart entre une accélération désirée  $\ddot{\underline{x}}_{des}$  et toutes les prévisions d'accélération  $\ddot{\underline{x}}(k + i/k)$ .

$$\forall k \in \{k_0, \dots, k_{end}\}, \forall i \in \{0, \dots, N\}, N \in \mathbb{N}$$

$$\|\ddot{\underline{x}}(k + i/k) - \ddot{\underline{x}}_{des}(k + i/k)\| \quad (3.9)$$

Le problème est contraint par des inégalités, et considère uniquement le couple comme variable d'optimisation, ce qui implique d'exprimer les prévisions d'accélération de la sorte :

$$\begin{aligned}\ddot{\underline{x}}(k + i/k) &= J_k \ddot{\underline{q}}(k + i/k) + \dot{J}_k \dot{\underline{q}}_k \\ &= J_k M_k^{-1}(\underline{\tau}(k + i/k) - (\underline{b}_k + \underline{g}_k)) + \dot{J}_k \dot{\underline{q}}_k\end{aligned}\quad (3.10)$$

En effet, une première formulation utilisant comme variables d'optimisation le couple et l'accélération articulaire avait été considérées. Elle est développée en annexe A, mais, aux vues de la complexité, l'écriture réduite a été privilégiée. Elle remplace l'accélération articulaire par une expression du couple grâce au modèle dynamique.

Le calcul de la suite  $(\underline{\tau}^*(k + i/k))_{i \in \{0, \dots, N\}}$  des prévisions des couples articulaires sur un horizon de temps fini, dont le premier terme définira la commande à appliquer à une itération  $k$  (correspondant à un instant  $t$ ), est donc :

Soit  $N \in \mathbb{N}^*$ ,  $\forall k \in \{k_0, \dots, k_{end}\}$  :

$$\begin{aligned}&(\underline{\tau}^*(k + i/k))_{i \in \{0, \dots, N\}} \\ &= \underset{\underline{\tau}}{\operatorname{argmin}} \frac{1}{2} \omega_k \sum_{i=0}^N \|J_k M_k^{-1}(\underline{\tau}(k + i/k) - (\underline{b}_k + \underline{g}_k)) + \dot{J}_k \dot{\underline{q}}_k - \ddot{\underline{x}}_{des}(k + i/k)\|_2^2 \\ &\quad + \frac{1}{2} \epsilon \sum_{i=0}^N \|r(\underline{\tau}(k + i/k))\|_2^2\end{aligned}\quad (3.11)$$

contrainte par :

$$\begin{aligned}\underline{\tau}_{min} &\leq \underline{\tau}(k + i/k) \leq \underline{\tau}_{max}, \quad i \in \{0, \dots, N\} \\ \underline{q}_{tor,min}(k + i/k) &\leq \Delta t^2 M_k^{-1} \sum_{j=1}^i \left( \left( \frac{1}{2} + i - j \right) \underline{\tau}(k + j - 1/k) \right) \leq \underline{q}_{tor,max}(k + i/k), \\ &i \in \{1, \dots, N\}\end{aligned}$$

## 3.2. Choix de conception pour la simulation

Cette section concerne les choix faits pour l'implémentation de la simulation.

### 3.2.1. Environnement et extensions utilisés

Les scripts mis en place utilisent des fonctionnalités s'ajoutant à Matlab.

Pour la simulation de modèle de robots, la librairie "Robotics Toolbox" de Peter Corke [3]. Trois modèles prédéfinis, dont les paramètres par défaut sont présentés en annexe E, sont alors utilisés :

- un robot R plan (pour les tests avec un modèle simple),
- un robot RR plan (pour les tests sur un modèle simple, mais comportant un couplage),
- le robot PUMA 560.

Concernant la résolution du problème d'optimisation quadratique, deux extensions, CVX [4] [5] et OSQP [6], sont mises en place, en plus du solveur incorporé à Matlab : quadprog [7].

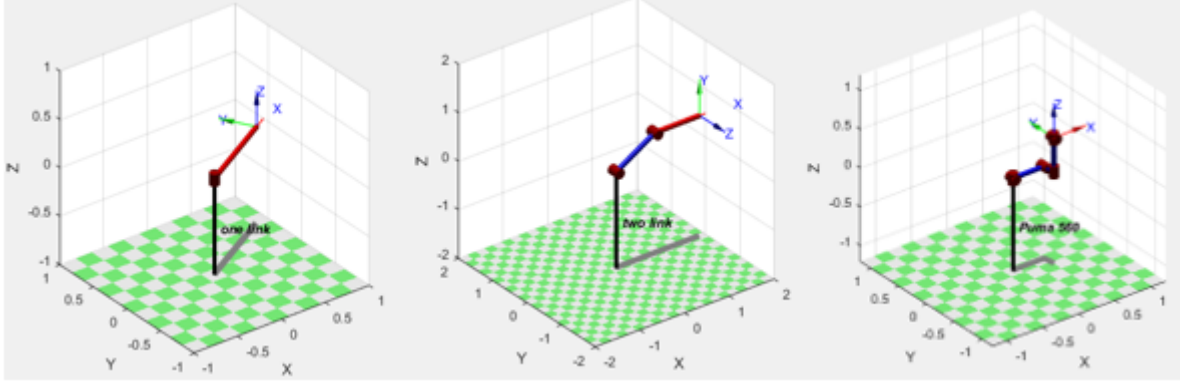


FIGURE 3.1. – Modèles de robot étudiés en simulation. De gauche à droite : robot R plan, robot RR plan, PUMA 560.

### 3.2.2. Formulation de la tâche

Lors de l'implémentation de la commande, un choix est nécessaire concernant la simulation des tâches  $\ddot{x}_{des}(k)$  et de ses prévisions  $\ddot{x}_{des}(k+i/k)$ . Il a alors été décidé de définir les tâches comme la somme du résultat d'un générateur de trajectoire et d'une correction Proportionnelle-Dérivée.

La trajectoire générée cherche à minimiser le maximum de la dérivée de l'accélération (Min Max Jerk [8]) articulaire. Elle est calculée pour toutes les itérations en début de simulation, mais, afin de simuler un comportement réactif, où la tâche est découverte pendant le fonctionnement, une seule itération n'est appelée à la fois pour construire la tâche. Le choix de la trajectoire Min Max Jerk a été motivé par la possibilité de pouvoir facilement la paramétrer et par le fait qu'elle soit applicable dans le cas d'un bras robotique.

Contrairement à la trajectoire ou à la correction, la tâche et ses prévisions sont calculées dans l'espace opérationnel. Par conséquent, la dérivée du modèle cinématique direct est utilisé, afin de passer de l'espace articulaire à l'espace opérationnel, tel que :

$$\forall k \in \{k_0, \dots, k_{end}\}, \forall i \in \{0, \dots, N\},$$

$$\ddot{x}_{des}(k+i/k) = J(q_{task}(k+i/k)) \left( \ddot{q}_{ref}(k+i) + PD(\ddot{q}_{ref}(k+i)) \right) + \dot{J}_k \dot{q}_k \quad (3.12)$$

Avec :

$$q_{task}(k+i/k) = \underline{q}(k) + i \Delta t \dot{\underline{q}}(k) + \sum_{j=1}^i \left( \Delta t^2 \left( \frac{1}{2} + i - j \right) \left( \ddot{\underline{q}}_{ref}(k+j-1) + PD(\ddot{\underline{q}}_{ref}(k+j-1)) \right) \right)$$

Il est à noter que, pour avoir une écriture explicite, l'hypothèse de constance sur l'horizon de temps a été conservée pour le produit  $\dot{J}_k \dot{q}_k$ .

### 3.2.3. Principe de fonctionnement de l'algorithme

A partir de la formulation de la commande et de la génération de tâche, un algorithme de simulation est élaboré. Le principe de fonctionnement de celui-ci est présenté en figure 3.3.

## 3.3. Exemple d'application de la commande prédictive

Afin de valider la commande prédictive implémentée, l'exemple utilisé en section 2.4 est repris. Les paramètres de simulations et la cible  $\underline{q}_{end}$  à atteindre sont les mêmes que pour la simulation en commande réactive. Le but est de savoir si la commande prédictive permet d'éviter l'incompatibilité présente en commande réactive. Celle-ci est alors configurée avec une amplitude d'horizon de temps de 150 ms ( $N$  égal à 15).

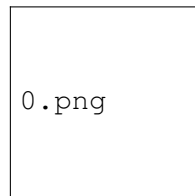


FIGURE 3.2. – Evolution temporelle des limites les plus restrictives - Commande prédictive. Il n'y a pas d'inversion des limites avec cette commande, contrairement à la simulation en commande réactive.

La simulation se passe sans apparition d'incompatibilité. Le robot se maintient en butée une fois que celle-ci est atteinte. Aucune inversion de contrainte n'a lieu (figure 3.2). L'utilisation de la commande prédictive a donc permis d'éviter l'incompatibilité de contrainte.

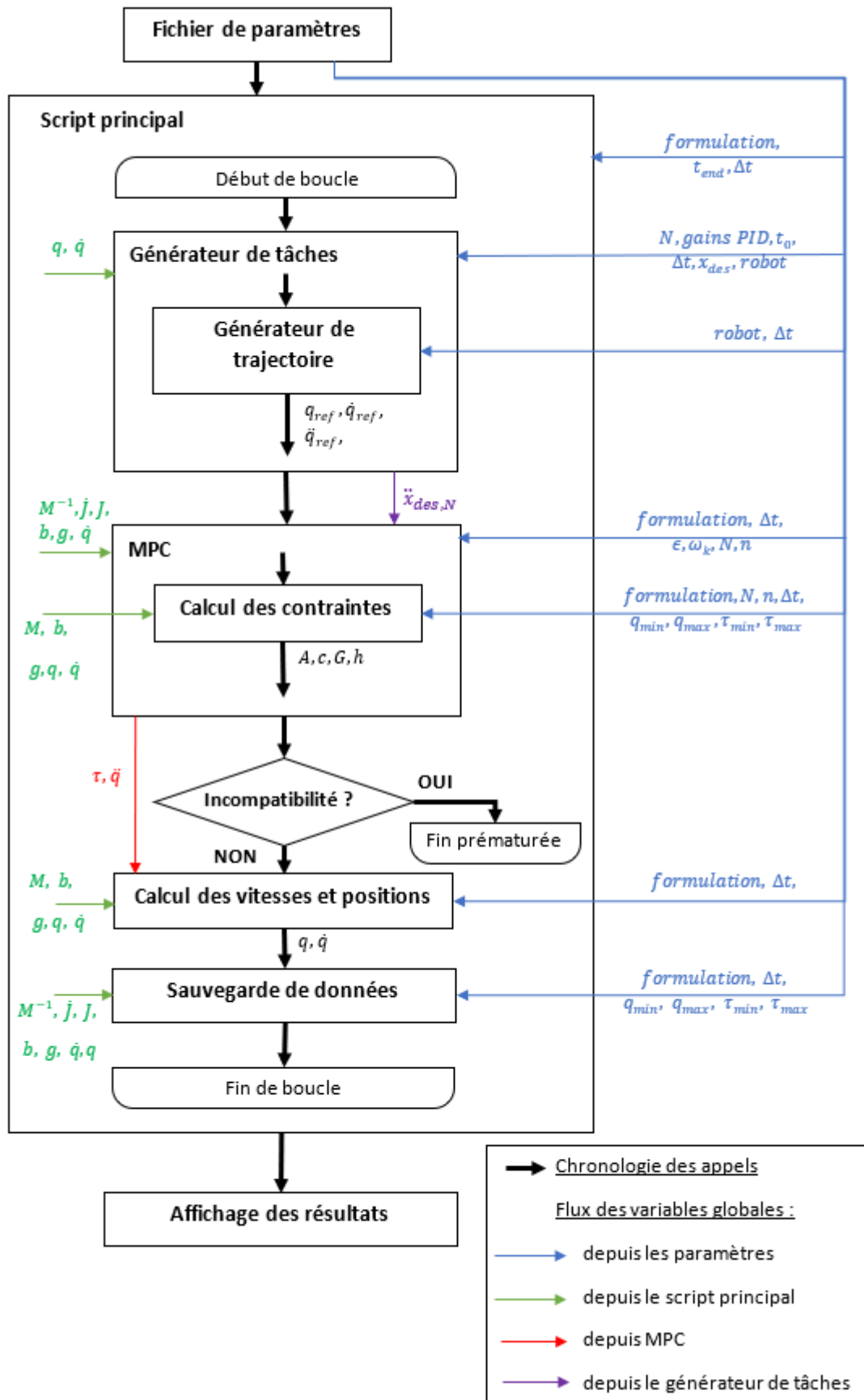


FIGURE 3.3. – Schéma de principe de l'algorithme de simulation.

## 4. Comparaison réactif/prédictif : évitement des incompatibilités

La commande prédictive a permis d'éviter une incompatibilité dans un cas simple. Il faut maintenant savoir si la commande est capable d'éviter ces situations dans des cas différents, et sur des modèles de robots plus complexes.

### 4.1. Principe des tests

Le but des tests est de comparer les capacités de la commande prédictive à éviter les situations d'incompatibilité, avec celles de la commande réactive.

Pour cela, une série d'objectifs à atteindre est définie pour chacun des robots présentés en section 3.2.1. Une simulation est lancée pour chaque objectif de chaque série, une fois en commande réactive et une autre en commande prédictive. Les paramètres de simulation sont ceux par défaut (annexe E), sauf pour la vitesse maximale utilisée par le générateur de trajectoire, qui est de  $0.5 \text{ rad.s}^{-1}$ . Pour chaque objectif ainsi défini, une simulation est lancée, et l'apparition ou non d'une situation d'incompatibilité de contrainte est attendue comme résultat de simulation.

Les séries contiennent des objectifs qui sont à l'intérieur des butées, mais définissent aussi des objectifs qui se trouvent en dehors de celles-ci. Il est alors intéressant de voir comment chaque commande gère l'approche de la butée, et s'il y a apparition de situation d'incompatibilité de contraintes.

### 4.2. Résultats

Cette section présente la définition des objectifs, ainsi que les résultats pour chacun des robots.

#### 4.2.1. Robot R plan

Pour ce robot, les objectifs sont définis dans l'espace articulaire et se répartissent sur un intervalle allant de 0 à 360 degrés, avec un pas de 1 degré. La moitié des objectifs se situe donc en dehors de butées.

La commande prédictive est paramétrée pour avoir une amplitude d'horizon de temps de 110 ms, avec un nombre  $N$  de prévisions de 11.

Les résultats des tests d'évitements sont présentés en figure 4.1. En commande réactive, des situations d'incompatibilité de contrainte apparaissent pendant les simulations pour 176 objectifs différents. En commande prédictive, il n'y a qu'une seule simulation où une situation d'incompatibilité apparaît. Cependant, un test rapide sur la simulation de



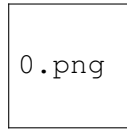


FIGURE 4.1. – Résultats des test d'évitement d'incompatibilité (robot R plan, avec représentation des butées). Chaque point représente un objectif à atteindre, dans l'espace opérationnel.

cet objectif en particulier, permet d'éviter l'incompatibilité, lorsque l'amplitude de la commande prédictive est paramétrée à 100 ms, avec  $N$  égal à 10.

#### 4.2.2. Robot RR plan

La série d'objectifs pour le robot RR plan est définie dans l'espace articulaire telle que :

- le premier axe se déplacera de 0 à 360 degrés, avec un pas de 45 degré,
- pour chaque pas du premier axe, le second axe définira les objectifs entre 0 et 360 degrés, avec un pas de 5 degré.

La majeure partie des objectifs est alors située en dehors des butées. Quant à la commande prédictive, elle est paramétrée en premier lieu avec une amplitude de 120 ms, avec  $N$  égal à 12.

Les résultats pour ce robot sont présentés en figure 4.3. 247 situations d'incompatibilité apparaissent pour les simulations en commande réactive. Pour la commande prédictive, 22 situations sont détectées lors des simulations. Or, pour ces objectifs, et avec une commande prédictive d'amplitude de 150 ms ( $N$  égal à 15), il est possible d'éviter les situations d'incompatibilité.

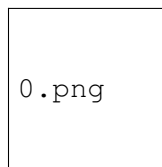


FIGURE 4.2. – Résultats des test d'évitement d'incompatibilité (robot RR plan, avec représentation des butées). Chaque point représente un objectif à atteindre dans l'espace opérationnel.

#### 4.2.3. Robot PUMA 560

Concernant le PUMA 560, les objectifs sont définis tels que :

- le premier axe définit son objectif sur 360 degrés, avec un pas de 45 degré,
- pour chaque pas du premier axe, le deuxième définit son objectif sur 360 degré, avec un pas de 30,
- pour chaque pas du premier et du deuxième axe, le troisième définit son objectif sur 360 degré, avec un pas de 30.

Avec ces objectifs articulaires, il est important de noter que le robot est commandé uniquement pour rejoindre un point dans l'espace opérationnel, et son orientation est laissée libre.

Sur les 1521 simulations, 1109 présentent des situations d'incompatibilité en commande réactive, contre 4 en commande prédictive d'amplitude de 150 ms ( $N$  égal à 15). Ces situations ont lieu pendant des simulations où les objectifs sont les plus éloignés à l'extérieur des butées.

Des nouveaux tests sont alors réalisés avec des amplitudes différentes. La simulation de l'un des quatre objectifs ne détecte pas de situation d'incompatibilité pour une amplitude de 100 ms ( $N$  égal à 10). Les trois autres n'en détectent pas non plus pour une commande avec une amplitude de 160 ms ( $N$  égal à 16).

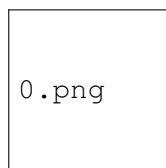


FIGURE 4.3. – Résultats des test d'évitement d'incompatibilité (PUMA 560, avec représentation des butées). Les différentes figures correspondent à des plans différents, dus à la rotation du premier axe et associé à différentes valeurs de  $q_1$ . Les butées ne sont pas représentées pour  $q_1 = 180^\circ$ , car ce plan est déjà en dehors de ces dernières. Chaque point représente un objectif à atteindre dans l'espace opérationnel.

### 4.3. Conclusion sur l'évitement d'incompatibilité

Les tests montrent que la commande prédictive, avec un paramétrage de l'amplitude de l'horizon de temps dépendant de l'objectif, permet d'éviter les situations d'incompatibilité existantes pour une commande réactive, sur les trois robots simulés.

De plus, l'atteinte de l'objectif, ou le maintien en butées pour le cas d'objectifs en dehors de celles-ci, est bien vérifié lors des tests. Les erreurs résiduelles pour les objectifs à l'intérieur des intervalles des butées ne dépassent pas 1 cm pour chacun des robots. Pour le maintien en butée, l'écart à la butée ne dépasse pas 5 degrés.

## 5. Comparaison réactif/prédictif : compléments

En plus de l'évitement des situations d'incompatibilité, les commandes réactives et prédictives sont comparées sur deux autres aspects : l'évitement d'obstacle opérationnel et l'énergie destinée aux actionneurs.

### 5.1. Evitement d'obstacle opérationnel

En plus de ces contraintes internes, le robot doit être capable de respecter celles qui lui sont imposées par les obstacles présents dans son environnement.

Dans le cadre de la simulation, l'obstacle mis en place est un mur, qui est représenté par un plan paramétré avec un vecteur normal  $\underline{n}$  et un point. Le comportement souhaité pour le robot est, lorsqu'il approche du mur, de ne pas faire passer l'effecteur à travers l'obstacle.

#### 5.1.1. Formulation de la contrainte opérationnelle

Une contrainte correspondant à un mur peut être formulée telle que la vitesse opérationnelle à l'itération suivante et dans la direction du mur doit être inférieure à celle permettant d'arriver au mur en un pas de temps.

Il est à noter que la grandeur  $d_o(k)$  correspond à la distance entre l'effecteur et le mur lors de l'état correspondant à l'itération de calcul. Lors d'un fonctionnement réel, celle-ci serait le résultat d'une mesure d'un capteur. En simulation, elle est calculée par un script dédié.

$$\forall k \in \{k_0, \dots, k_{end} - 1\}, \frac{\underline{n}^T}{\|\underline{n}\|} \dot{\underline{x}}(k+1) \leq \frac{d_o(k)}{\Delta t} \quad (5.1)$$

Cette inégalité peut être reformulée pour faire apparaître la vitesse articulaire, et pour devenir des prévisions :

$$\forall k \in \{k_0, \dots, k_{end}\}, \forall i \in \{0, \dots, N-1\}, N \in \mathbb{N}^* \\ \frac{\underline{n}^T}{\|\underline{n}\|} J_k \dot{\underline{q}}(k+i+1/k) \leq \frac{d_o(k+i/k)}{\Delta t} \quad (5.2)$$

A partir du développement de la prévision de distance (annexe C) et de la contrainte (annexe D) :

$$\forall k \in \{k_0, \dots, k_{end}\},$$

$$\begin{aligned}
& \Delta t \frac{\underline{n}^T}{\|\underline{n}\|} J_k M_k^{-1} \left( \underline{\tau}(k + i/k) + \sum_{j=1}^i \left( \left( -i + j - \frac{1}{2} \right) \underline{\tau}(k + j - 1/k) \right) \right) \\
& \leq \frac{d_o(k/k)}{\Delta t} \\
& + (i-1) \frac{\underline{n}^T}{\|\underline{n}\|} J_k \dot{\underline{q}}(k) + \left( 1 - \sum_{j=1}^i j + \frac{1}{2} i \right) \Delta t \frac{\underline{n}^T}{\|\underline{n}\|} J_k M_k^{-1} (\underline{b}_k + \underline{g}_k) + \frac{1}{2} i \Delta t \frac{\underline{n}^T}{\|\underline{n}\|} J_k \dot{\underline{q}}_k \\
& , \forall i \in \{1, \dots, N\}, N \in \mathbb{N}^*
\end{aligned} \tag{5.3}$$

$$\begin{aligned}
\Delta t \frac{\underline{n}^T}{\|\underline{n}\|} J_k M_k^{-1} \underline{\tau}(k/k) & \leq \frac{d_o(k/k)}{\Delta t} + \Delta t \frac{\underline{n}^T}{\|\underline{n}\|} J_k M_k^{-1} (\underline{b}_k + \underline{g}_k) - \frac{\underline{n}^T}{\|\underline{n}\|} J_k \dot{\underline{q}}_k \\
& , \text{ pour } i = 0
\end{aligned} \tag{5.4}$$

### 5.1.2. Tests d'évitement

## 5.2. Aspect énergétique

### 5.2.1. Critère d'évaluation considéré

### 5.2.2. Résultats et analyse

## **6. Pistes d'optimisation du coût de calcul**

### **6.1. Intervalle d'amplitude de l'horizon de temps**

**6.1.1. Nécessité d'évitement des incompatibilités et borne inférieure**

**6.1.2. Borne supérieure et respect de l'hypothèse de constance des termes dynamiques**

**6.1.3. Conclusion sur la méthode proposée de définitions des bornes de l'amplitude**

### **6.2. Choix des solveurs**

**6.2.1. Etude des performances des solveurs**

**6.2.2. Conclusion sur le choix des solveurs**

## **7. Conclusion**

# Bibliographie

- [1] Sébastien Rubrecht, Vincent Padois, Philippe Bidaud, Michel De Broissia, and Max Da Silva Simoes. Motion safety and constraints for multibody robots. *Autonomous Robots, Springer Verlag*, 3(32) :333–349, 2012.
- [2] Anis Meguenani. *Safe Control of Robotic Manipulators in Dynamic Contexts*. PhD thesis, Université Pierre et Marie Curie, 2017.
- [3] P.I. Corke. *Robotics, Vision and Control*. Springer, 2017.
- [4] Inc. CVX Research. CVX : Matlab software for disciplined convex programming, version 2.0. <http://cvxr.com/cvx>, August 2012.
- [5] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blodel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–100. Springer-Verlag Limited, 2008. [http://stanford.edu/~boyd/graph\\_dcp.html](http://stanford.edu/~boyd/graph_dcp.html).
- [6] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP : An operator splitting solver for quadratic programs. *ArXiv e-prints*, November 2017. Provided by the SAO/NASA Astrophysics Data System, <http://adsabs.harvard.edu/abs/2017arXiv171108013S>.
- [7] MathWorks. Quadratic programming algorithms. <http://mathworks.com/help/optim/ug/quadratic-programming-algorithms.html>, June 2018.
- [8] Konstantinos J. Kyriakopoulos and George N. Saridis. Minimum jerk path generation. In *Robotics and Automation, 1988 IEEE International Conference on*, volume 1, pages 364–369, 1988.
- [9] Yang Tan. *Continuous tasks and constraints transitions for the control of robots*. PhD thesis, Université Pierre et Marie Curie, 2016.

## A. Annexe : formulation complète de la commande prédictive

Cette formulation, qui a été la première à être considérée, utilise les couples et les accélérations articulaires comme variables d'optimisation. Pour lier ces variables, le modèle dynamique est utilisé comme une contrainte d'égalité (équation A.2).

Les autres contraintes d'égalité permettent d'exprimer les prévisions de vitesses et positions articulaires en fonction de celles à l'instant  $k$ , à partir de développements limités (respectivement équations A.3 et A.4). Les contraintes d'inégalité traduisent la limitation du couple (équation A.5) et la présence de butées (équation A.6).

Soit  $N \in \mathbb{N}, \forall k \in \{k_0, \dots, k_{end}\}$  :

$$\begin{aligned} (\tau^*(k + i/k))_{i \in \{0, \dots, N\}} = \underset{\tau, \ddot{q}}{\operatorname{argmin}} \quad & \frac{1}{2} \omega_k \sum_{i=0}^N \left\| J_k \ddot{q}(k + i/k) + \dot{J}_k \dot{q}_k - \ddot{x}_{des}(k + i/k) \right\|_2^2 \\ & + \frac{1}{2} \epsilon \sum_{i=0}^N \|r(\tau(k + i/k))\|_2^2 \end{aligned} \quad (\text{A.1})$$

contrainte par :

$$M_k \ddot{q}(k + i/k) + \underline{b}_k + \underline{g}_k = \tau(k + i/k), \quad i \in \{0, \dots, N\} \quad (\text{A.2})$$

$$\dot{q}(k + i + 1/k) = \dot{q}(k + i/k) + \Delta t \ddot{q}(k + i/k), \quad i \in \{0, \dots, N - 1\} \quad (\text{A.3})$$

$$q(k + i + 1/k) = q(k + i/k) + \Delta t \dot{q}(k + i/k) + \frac{1}{2} \Delta t^2 \ddot{q}(k + i/k), \quad i \in \{0, \dots, N - 1\} \quad (\text{A.4})$$

$$\tau_{min} \leq \tau(k + i/k) \leq \tau_{max}, \quad i \in \{0, \dots, N\} \quad (\text{A.5})$$

$$\ddot{q}_{min,pos}(k + i/k) \leq \ddot{q}(k + i/k) \leq \ddot{q}_{max,pos}(k + i/k), \quad i \in \{0, \dots, N\} \quad (\text{A.6})$$

avec :

$$\ddot{q}_{min,pos}(k + i/k) = \frac{2}{\Delta t^2} \left( q_{min} - q(k + i/k) - \Delta t \dot{q}(k + i/k) \right), \quad i \in \{0, \dots, N\}$$

$$\ddot{q}_{max,pos}(k + i/k) = \frac{2}{\Delta t^2} \left( q_{max} - q(k + i/k) - \Delta t \dot{q}(k + i/k) \right), \quad i \in \{0, \dots, N\}$$



## B. Annexe : définitions des suites des prédictions de vitesse et de position

Cette annexe traite de la définition des suites des prédictions de vitesse et de position, inspirée par les travaux de Yang Tan [9].

Soit  $N \in \mathbb{N}^*$ .

$\forall k \in \{k_0, \dots, k_{end}\}$  :

$$\begin{pmatrix} \underline{q}(k+i/k) \\ \underline{\dot{q}}(k+i/k) \end{pmatrix} = A_1 \begin{pmatrix} \underline{q}(k+i-1/k) \\ \underline{\dot{q}}(k+i-1/k) \end{pmatrix} + \underline{b}_1 \underline{\ddot{q}}(k+i-1/k), \quad i \in \{1, \dots, N\}$$

avec :

$$A_1 = \begin{bmatrix} I_n & \Delta t I_n \\ O_n & I_n \end{bmatrix}$$

$$\underline{b}_1 = \begin{pmatrix} \frac{\Delta t^2}{2} \underline{I} \\ \Delta t \underline{I} \end{pmatrix}$$

Par récurrence, il est possible d'écrire le terme général :

$\forall k \in \{k_0, \dots, k_{end}\}$  :

$$\begin{pmatrix} \underline{q}(k+i/k) \\ \underline{\dot{q}}(k+i/k) \end{pmatrix} = A_1^i \begin{pmatrix} \underline{q}(k) \\ \underline{\dot{q}}(k) \end{pmatrix} + \sum_{j=1}^i \left( A_1^{i-j} \underline{b}_1 \underline{\ddot{q}}(k+j-1/k) \right), \quad i \in \{1, \dots, N\}$$

Or :

$$A_1^i = \begin{bmatrix} I_n & i \Delta t I_n \\ O_n & I_n \end{bmatrix}$$

L'écriture littérale donne donc :

$\forall k \in \{k_0, \dots, k_{end}\}$  :

$$\begin{cases} \underline{q}(k+i/k) &= \underline{q}(k) + i \Delta t \underline{\dot{q}}(k) + \sum_{j=1}^i (\Delta t^2 (\frac{1}{2} + i - j) \underline{\ddot{q}}(k+j-1/k)) \\ \underline{\dot{q}}(k+i/k) &= \underline{\dot{q}}(k) + \sum_{j=1}^i (\Delta t \underline{\ddot{q}}(k+j-1/k)) \end{cases}$$

$$i \in \{1, \dots, N\}$$

Il est ensuite possible de remplacer l'accélération grâce au modèle dynamique, ce qui donne les termes généraux suivants :

$\forall k \in \{k_0, \dots, k_{end}\}$  :

$$\begin{cases} \underline{q}(k+i/k) &= \underline{q}(k) + i \Delta t \underline{\dot{q}}(k) + \Delta t^2 M_k^{-1} \sum_{j=1}^i \left( (\frac{1}{2} + i - j) (\underline{\tau}(k+j-1/k) - \underline{b}_k - \underline{g}_k) \right) \\ \underline{\dot{q}}(k+i/k) &= \underline{\dot{q}}(k) + \Delta t M_k^{-1} \sum_{j=1}^i (\underline{\tau}(k+j-1/k)) - i \Delta t M_k^{-1} (\underline{b}_k + \underline{g}_k) \end{cases}$$

$$i \in \{1, \dots, N\}$$

## C. Annexe : définition de la suite des prédictions de la distance au mur

Cette annexe détaille le raisonnement utilisé pour déterminer une expression des prévisions de la distance  $d_o$  à l'obstacle opérationnel (mur), en fonction de la distance à l'instant  $k$ .

Soit B, un point du mur, A le point représentant l'effecteur et  $\underline{n}$ , un vecteur normal au plan définissant le mur.

$$\forall k \in \{k_0, \dots, k_{end}\} :$$

$$d_o(k) = \frac{BA(k) \cdot \underline{n}}{\|\underline{n}\|} = \frac{(\underline{x}(k) - \underline{x}_B) \cdot \underline{n}}{\|\underline{n}\|}$$

$$\dot{d}_o(k) = \frac{\dot{\underline{x}}(k) \cdot \underline{n}}{\|\underline{n}\|} = \frac{\underline{n}^T}{\|\underline{n}\|} (J(q(k)) \dot{\underline{q}}(k))$$

$$\ddot{d}_o(k) = \frac{\ddot{\underline{x}}(k) \cdot \underline{n}}{\|\underline{n}\|} = \frac{\underline{n}^T}{\|\underline{n}\|} (J(q(k)) \ddot{\underline{q}}(k) + \dot{J}(q(k)) \dot{\underline{q}}(k))$$

Il est alors possible d'écrire le développement limité suivant :

$$\forall k \in \{k_0, \dots, k_{end}\}, \forall i \in \{1, \dots, N\}, N \in \mathbb{N} :$$

$$\begin{aligned} d_o(k + i/k) &= d_o(k + i - 1/k) + \Delta t \dot{d}_o(k + i - 1/k) + \frac{1}{2} \Delta t^2 \ddot{d}_o(k + i - 1/k) \\ &= d_o(k + i - 1/k) + \Delta t \frac{\underline{n}^T}{\|\underline{n}\|} (J_k \dot{\underline{q}}(k + i - 1/k)) + \frac{1}{2} \Delta t^2 \frac{\underline{n}^T}{\|\underline{n}\|} (J_k \ddot{\underline{q}}(k + i - 1/k) + \dot{J}_k \dot{\underline{q}}_k) \end{aligned}$$

Par récurrence, et en remplaçant les prévisions de vitesse par l'expression trouvée en annexe B et les prévisions d'accélération grâce au modèle dynamique (équation A.2) :

$$\forall k \in \{k_0, \dots, k_{end}\}, \forall i \in \{1, \dots, N\}, N \in \mathbb{N} :$$

$$\begin{aligned} d_o(k + i/k) &= d_o(k) \\ &+ \Delta t \frac{\underline{n}^T}{\|\underline{n}\|} J_k \left( i \dot{\underline{q}}(k) + \Delta t M_k^{-1} \sum_{l=1}^i \left( \sum_{j=1}^l (\underline{\tau}(k + j - 1/k)) \right) - \sum_{j=1}^i j \Delta t M_k^{-1} (\underline{b}_k + \underline{g}_k) \right) \\ &+ \frac{1}{2} \Delta t^2 \frac{\underline{n}^T}{\|\underline{n}\|} \left( J_k M_k^{-1} \left( \sum_{j=1}^i (\underline{\tau}(k + j - 1/k)) - i (\underline{b}_k + \underline{g}_k) \right) + i \dot{J}_k \dot{\underline{q}}_k \right) \end{aligned}$$

Or :

$$\forall k \in \{k_0, \dots, k_{end}\}, \forall i \in \{1, \dots, N\}, N \in \mathbb{N} :$$

$$\sum_{l=1}^i \left( \sum_{j=1}^l (\tau(k + j - 1/k)) \right) = \sum_{j=1}^i ((i - j + 1) \tau(k + j - 1/k))$$

Par conséquent :

$$\forall k \in \{k_0, \dots, k_{end}\}, \forall i \in \{1, \dots, N\}, N \in \mathbb{N}$$

$$d_o(k + i/k) = d_o(k)$$

$$\begin{aligned} & + \Delta t \frac{n^T}{\|n\|} J_k \left( i \dot{q}(k) + \Delta t M_k^{-1} \sum_{j=1}^i ((i - j + 1) \tau(k + j - 1/k)) - \sum_{j=1}^i j \Delta t M_k^{-1} (\underline{b}_k + \underline{g}_k) \right) \\ & + \frac{1}{2} \Delta t^2 \frac{n^T}{\|n\|} \left( J_k M_k^{-1} \left( \sum_{j=1}^i (\tau(k + j - 1/k)) - i (\underline{b}_k + \underline{g}_k) \right) + i \dot{J}_k \dot{q}_k \right) \end{aligned}$$

et

$$Pour i = 0, d_o(k + i/k) = d_o(k)$$

## D. Annexe : développement du calcul de la contrainte opérationnelle

Cette annexe présente le développement des calculs qui ont permis de déterminer une expression explicite en prévision de couple pour la contrainte correspondant au mur.

$$\forall k \in \{k_0, \dots, k_{end}\}, \forall i \in \{0, \dots, N-1\}, N \in \mathbb{N}^*$$

$$\frac{n^T}{\|n\|} J_k \dot{q}(k+i+1/k) \leq \frac{d_o(k+i/k)}{\Delta t}$$

En développant les prévisions de vitesse et de distance au mur (annexe C) :

$$\begin{aligned} & \frac{n^T}{\|n\|} J_k \left( \dot{q}(k) + \Delta t M_k^{-1} \sum_{j=1}^{i+1} (\tau(k+j-1/k)) - (i+1) \Delta t M_k^{-1} (\underline{b}_k + \underline{g}_k) \right) \\ & \leq \frac{d_o(k)}{\Delta t} \\ & + \frac{n^T}{\|n\|} \left( J_k \left( i \dot{q}(k) + \Delta t M_k^{-1} \sum_{j=1}^i ((i-j+1) \tau(k+j-1/k)) - \sum_{j=1}^i j \Delta t M_k^{-1} (\underline{b}_k + \underline{g}_k) \right) \right) \\ & + \frac{1}{2} \Delta t \frac{n^T}{\|n\|} \left( J_k M_k^{-1} \left( \sum_{j=1}^i (\tau(k+j-1/k)) - i (\underline{b}_k + \underline{g}_k) \right) + i J_k \dot{q}_k \right) \end{aligned}$$

ce qui donne :

$$\begin{aligned} & \forall k \in \{k_0, \dots, k_{end}\}, \forall i \in \{1, \dots, N\}, N \in \mathbb{N}^* \\ & \Delta t \frac{n^T}{\|n\|} J_k M_k^{-1} \left( \sum_{j=1}^{i+1} (\tau(k+j-1/k)) - \sum_{j=1}^i ((i-j+1) \tau(k+j-1/k)) - \frac{1}{2} \sum_{j=1}^i (\tau(k+j-1/k)) \right) \\ & \leq \frac{d_o(k)}{\Delta t} \\ & + (i-1) \frac{n^T}{\|n\|} J_k \dot{q}(k) + \left( (i+1) - \sum_{j=1}^i j - \frac{1}{2} i \right) \Delta t \frac{n^T}{\|n\|} J_k M_k^{-1} (\underline{b}_k + \underline{g}_k) + \frac{1}{2} i \Delta t \frac{n^T}{\|n\|} J_k \dot{q}_k \end{aligned}$$

Le terme contenant les prévisions de couples peut être simplifié :

$$\begin{aligned}
& \left( \sum_{j=1}^{i+1} (\tau(k+j-1/k)) - \sum_{j=1}^i ((i-j+1) \tau(k+j-1/k)) - \frac{1}{2} \sum_{j=1}^i (\tau(k+j-1/k)) \right) \\
&= \tau(k+i/k) + \sum_{j=1}^i \left( \left( 1 - (i-j+1) - \frac{1}{2} \right) \tau(k+j-1/k) \right) \\
&= \tau(k+i/k) + \sum_{j=1}^i \left( \left( -i+j - \frac{1}{2} \right) \tau(k+j-1/k) \right)
\end{aligned}$$

Au final, le contrainte s'écrit :

$$\begin{aligned}
& \forall k \in \{k_0, \dots, k_{end}\}, \\
& \Delta t \frac{n^T}{\|n\|} J_k M_k^{-1} \left( \tau(k+i/k) + \sum_{j=1}^i \left( \left( -i+j - \frac{1}{2} \right) \tau(k+j-1/k) \right) \right) \\
& \leq \frac{d_o(k/k)}{\Delta t} \\
& + (i-1) \frac{n^T}{\|n\|} J_k \dot{q}(k) + \left( 1 - \sum_{j=1}^i j + \frac{1}{2} i \right) \Delta t \frac{n^T}{\|n\|} J_k M_k^{-1} (\underline{b}_k + \underline{g}_k) + \frac{1}{2} i \Delta t \frac{n^T}{\|n\|} J_k \dot{q}_k \\
& , \forall i \in \{1, \dots, N\}, N \in \mathbb{N}^*
\end{aligned}$$

$$\begin{aligned}
& \Delta t \frac{n^T}{\|n\|} J_k M_k^{-1} \tau(k/k) \leq \frac{d_o(k/k)}{\Delta t} + \Delta t \frac{n^T}{\|n\|} J_k M_k^{-1} (\underline{b}_k + \underline{g}_k) - \frac{n^T}{\|n\|} J_k \dot{q}_k \\
& , \text{ pour } i = 0
\end{aligned}$$

## E. Annexe : paramètres par défaut de la simulation

### — Paramètres généraux

Instant de début de simulation (s)	0
Instant de fin de simulation (s)	22
$\Delta t$ (s)	0.01
$\epsilon$	0.1
$\omega_k$	1000
Gain proportionnel $k_p$ ( $s^{-2}$ )	1500
Gain dérivé $k_d$ ( $s^{-1}$ )	$2\sqrt{k_p}$
Vitesse max. utilisée par le générateur de trajectoire (rad/s)	0.2

### — Robot R plan

Vecteur gravité en espace cartésien ( $m.s^{-2}$ )	$(0; -9.81; 0)^T$
$\underline{q}_{min}$ (rad)	$\frac{-\pi}{2}$
$\underline{q}_{max}$ (rad)	$\frac{\pi}{2}$
$\underline{\tau}_{min}$ (N.m)	-10
$\underline{\tau}_{max}$ (N.m)	10
$\underline{q}(k=0)$ (rad)	0.5
$\underline{\dot{q}}(k=0)$ (rad/s)	0

### — Robot RR plan

Vecteur gravité en espace cartésien ( $m.s^{-2}$ )	$(0; 0; -9.81)^T$
$\underline{q}_{min}$ (rad)	$(\frac{-\pi}{4}; \frac{-\pi}{4})^T$
$\underline{q}_{max}$ (rad)	$(\frac{\pi}{4}; \frac{\pi}{4})^T$
$\underline{\tau}_{min}$ (N.m)	$(-20; -10)^T$
$\underline{\tau}_{max}$ (N.m)	$(20; 10)^T$
$\underline{q}(k=0)$ (rad)	$(0.5; 0.5)^T$
$\underline{\dot{q}}(k=0)$ (rad/s)	$(0; 0)^T$

### — PUMA 560

---

Vecteur gravité en espace cartésien ( $\text{m.s}^{-2}$ )	$(0; 0; -9.81)^T$
$\underline{q}_{min}$ (rad)	$(-2.7925; -0.7854; -3.9270; -1.9199; -1.7453; -4.6426)^T$
$\underline{q}_{max}$ (rad)	$(2.7925; 3.9270; 0.7854; 2.9671; 1.7453; 4.6426)^T$
$\underline{\tau}_{min}$ (N.m)	$(-100; -80; -60; -40; -20; -10)^T$
$\underline{\tau}_{max}$ (N.m)	$(100; 80; 60; 40; 20; 10)^T$
$\underline{q}(k=0)$ (rad)	$(0.5; 0.5; 0.5; 0.5; 0.5; 0.5)^T$
$\dot{\underline{q}}(k=0)$ (rad/s)	$(0; 0; 0; 0; 0; 0)^T$