

## Chapitre 2. APPLICATION : SUIVI DE LA TRAJECOIRE

### I. LE ROBOT MOBILE

#### Introduction

Quand on parle de robot mobile les connaissances sont tout de suite focalisées sur le moyen de localisation de ce robot ou bien sur la planification de son mouvement (définition d'une trajectoire) ou bien sur la commande du robot proprement dit ou les trois ensembles.

Dans ce qui suit nous nous focaliserons plutôt sur la planification de son mouvement dont seront développés le modèle géométrique qui lui est associé (lié à l'étude cinétique du mobile), son modèle odométrique (permettant d'estimer la position du mobile en mouvement) mais aussi le principe de suivi de chemin ; nous y citeront les méthodes de suivi déjà utilisées et nous insisteront sur la méthode qui sera utilisée pour simuler le mouvement de notre futur robot mobile.

#### 1. Modèle du robot

##### 1.1. Modèle cinématique

Le modèle cinématique est la transformation qui permet de passer des vitesses angulaires des roues à la vitesse instantanée du robot. Cette transformation est divisée en trois étapes. Nous allons calculer la vitesse du milieu des deux roues. Ensuite, nous calculerons la vitesse de n'importe quel point du robot, et enfin, à partir des vitesses nous calculerons la position. [7]

##### a. Modèle cinématique du milieu des axes des roues

Voici le schéma du robot différentiel, respectivement vu de dessus et vu de côté.

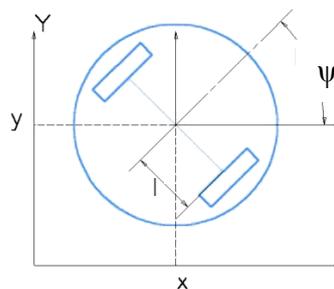


Figure II.1 : Modèle cinématique du robot vu de dessus

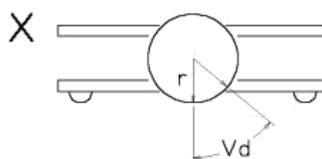


Figure II.2 : Modèle cinématique du robot vu de côté

## Application

Avec :

$2l$  : distance entre les roues

$r$  : rayon de la roue

$V_d, V_g$  : vitesse angulaire respective de la roue droite et gauche

$x, y$  : position du robot

$\psi$  : orientation du robot

Pour commencer, nous allons calculer la vitesse linéaire des roues ( $V_{ld}$  et  $V_{lg}$ ) :

$$V_{ld} = r \cdot V_d \quad \text{et} \quad V_{lg} = r \cdot V_g \quad (2.1)$$

Nous pouvons par ailleurs en déduire la vitesse moyenne du centre des roues comme suit :

$$V_{moy} = \frac{V_{ld} + V_{lg}}{2} \quad (2.2)$$

Pour ramener cette vitesse moyenne dans le repère  $(O, x, y)$  nous devons la décomposer en deux coordonnées suivant l'axe des  $x$  et suivant l'axe des  $y$ .

Ainsi avec l'expression de la vitesse angulaire du robot donnée par :

$$2 \cdot l \cdot \dot{\theta}_\psi = r \cdot V_d - r \cdot V_g \quad (2.3)$$

Donc

$$\dot{\theta}_\psi = \frac{r \cdot V_d - r \cdot V_g}{2 \cdot l} \quad (2.4)$$

NB :  $\psi$  est pris dans le sens trigonométrique

Le modèle cinématique de notre robot sous forme matricielle s'écrit finalement comme la formule 2.5 :

$$\begin{pmatrix} V_x \\ V_y \\ \dot{\theta}_\psi \end{pmatrix} = \frac{r}{2} \cdot \begin{bmatrix} \cos(\psi) & \cos(\psi) \\ \sin(\psi) & \sin(\psi) \\ \frac{1}{l} & \frac{-1}{l} \end{bmatrix} \cdot \begin{pmatrix} V_d \\ V_g \end{pmatrix} \quad (2.5)$$

### b. Modèle cinématique pour n'importe quel point du robot

Parfois, le centre du robot n'est pas le milieu des deux roues. Ou alors, le robot est équipé d'un outil, et c'est la position de cet outil qui nous intéresse. Il est alors possible de calculer le modèle cinématique en prenant en compte cette nouvelle contrainte. Supposons que le point du robot qui nous intéresse est décalé de  $d_1$  et  $d_2$  [13]:

## Application

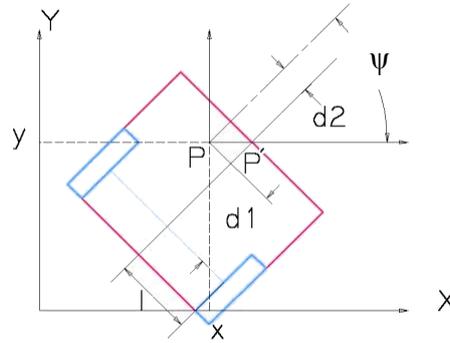


Figure II.3 : Modèle cinématique du robot pour n'importe quel point

Voici la transformation (en position) qui permet de passer du centre de l'axe des roues au point P'.

$$\begin{aligned}x^{P'} &= x + d1 \cdot \cos(\psi) \\y^{P'} &= y + d1 \cdot \sin(\psi) \\ \psi^{P'} &= \psi\end{aligned}\quad (2.6)$$

En dérivant 2.6, on obtient les vitesses au point P' :

$$\begin{aligned}V_x^{P'} &= V_x - dl \cdot \sin(\psi) \cdot \dot{\theta}_\psi = \frac{r}{2} \cdot \left[ \left( \cos \psi - \frac{dl}{l} \cdot \sin \psi \right) V_d + \left( \cos \psi + \frac{dl}{l} \cdot \sin \psi \right) \cdot V_g \right] \\ V_y^{P'} &= V_y - dl \cdot \cos(\psi) \cdot \dot{\theta}_\psi = \frac{r}{2} \cdot \left[ \left( \sin \psi + \frac{dl}{l} \cdot \cos \psi \right) V_d + \left( \sin \psi - \frac{dl}{l} \cdot \cos \psi \right) \cdot V_g \right] \\ \dot{\theta}_{\psi^{P'}} &= \dot{\theta}_\psi\end{aligned}\quad (2.7)$$

Sous forme matricielle, tenons compte de l'expression 2.4 pour la vitesse angulaire, nous obtenons :

$$\begin{pmatrix} V_x \\ V_y \\ \dot{\theta}_{\psi^{P'}} \end{pmatrix} = \frac{r}{2} \cdot \begin{bmatrix} \cos(\psi) - \frac{d1}{l} \sin(\psi) & \cos(\psi) + \frac{d1}{l} \sin(\psi) \\ \sin(\psi) + \frac{d1}{l} \cos(\psi) & \sin(\psi) - \frac{d1}{l} \cos(\psi) \\ \frac{1}{l} & \frac{-1}{l} \end{bmatrix} \cdot \begin{pmatrix} V_d \\ V_g \end{pmatrix}\quad (2.8)$$

Pour obtenir le modèle au point P, il suffit d'effectuer le changement de repère de P' à P, d'où:

$$\begin{pmatrix} V_x \\ V_y \\ \dot{\theta}_{\psi^{P'}} \end{pmatrix} = \frac{r}{2} \cdot \begin{bmatrix} \cos(\psi) - \frac{d1}{l} \sin(\psi) & \cos(\psi) + \frac{d1}{l} \sin(\psi) \\ \sin(\psi) + \frac{d1}{l} \cos(\psi) & \sin(\psi) - \frac{d1}{l} \cos(\psi) \\ \frac{1}{l} & \frac{-1}{l} \end{bmatrix} \cdot \begin{pmatrix} V_d \\ V_g \end{pmatrix}\quad (2.9)$$

## 1.2. Modèle odométrique du robot

En discrétisant le modèle cinématique moyennant des hypothèses de simplification c'est-à-dire que le mouvement des roues est un roulement sans glissement (formule 2.10) [8][13]:

$$\dot{y} \cdot \cos(\psi) - \dot{x} \sin(\psi) = 0 \quad (2.10)$$

Par contre, la vitesse longitudinale s'écrit

$$\dot{y} \cdot \cos(\psi) + \dot{x} \sin(\psi) = V_{moy} \quad (2.11)$$

Le modèle odométrique devient :

$$\left\{ \begin{array}{l} x_{k+1} = x_k + u \cdot \cos\theta_{k+1} dt \\ y_{k+1} = y_k + u \cdot \sin\theta_{k+1} dt \\ \theta_{k+1} = \theta_k + \dot{\theta} dt \end{array} \right. \quad (2.12)$$

Soit

$$\left\{ \begin{array}{l} \theta(k) = \frac{r[\theta_d(k) - \theta_g(k)]}{l} \\ x(k) = x(k-1) + \frac{r(\Delta\theta_d + \Delta\theta_g)}{2} \cdot \cos\left(\frac{\theta(k) - \theta(k-1)}{2}\right) \\ y(k) = y(k-1) + \frac{r(\Delta\theta_d + \Delta\theta_g)}{2} \cdot \sin\left(\frac{\theta(k) - \theta(k-1)}{2}\right) \end{array} \right. \quad (2.13)$$

## 2. Calcul de la position

Une fois qu'on connaît les vitesses instantanées, on peut facilement calculer la position du robot, en intégrant  $V_x$ ,  $V_y$  et  $\dot{\theta}_\psi$ . Pour intégrer ces valeurs, il suffit d'effectuer une sommation discrète des valeurs depuis le début :

$$\begin{aligned} x &= \sum_0^n V_x \\ y &= \sum_0^n V_y \\ \theta_\psi &= \sum_0^n \dot{\theta}_\psi \end{aligned} \quad (2.14)$$

### 3. Méthodes de suivi de chemin

Comme l'idée principale de ce travail est la mise en œuvre d'un suivi de chemin. Plusieurs approches du problème existent, mais la méthode de référence dans le domaine reste celle proposée par le Français Claude Samson [11].

#### 3.1. Loi de commande de C.Samson

C. Samson propose une loi de commande permettant de suivre un chemin de manière géométrique indépendamment de la vitesse du robot. La seule variable de commande utilisée est la vitesse angulaire  $\omega = \dot{\theta}$  du robot [7].

Soit  $C$  le chemin suivi et  $\rho(s)$  sa courbure au point d'abscisse curviligne  $s$ , on note  $R'$  le point de  $C$  le plus proche de  $R$  (point de référence du robot) et  $d$  la distance  $\|RR'\|$  (voir figure II.4). De même on note  $\theta_{des}$  l'angle de la tangente à la courbe en  $R'$  dans le repère principal. Cet angle représente l'orientation désirée pour le robot.

Le but de la loi de commande proposée est de réduire à la fois l'erreur en distance  $d$  et l'erreur en orientation :

$$\theta_e = \theta - \theta_{des} \quad (2.15)$$

Elle est donnée dans sa forme la plus simple par :

$$\omega = v. [\rho(S_{R'}) - k_1 \cdot d - k_2 \cdot \theta_e] \quad (2.16)$$

Où  $k_1$  et  $k_2$  sont des constantes positives

$s_{R'}$  : abscisse curviligne du point  $R'$

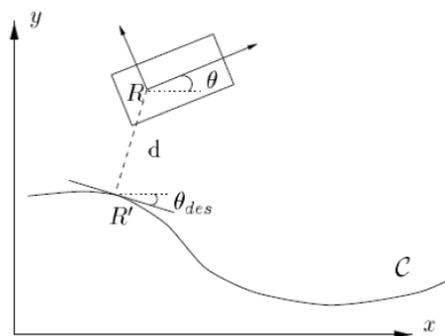


Figure II.4 : Suivi de chemin selon C.Samson

### 3.2. Courbe de Bézier

Pour pouvoir choisir la meilleure trajectoire pour suivre le chemin fourni par un planificateur, il faut établir un critère de choix ; dans tous les cas celui-ci fournira au minimum plusieurs points de passage consécutifs que le robot devra traverser pour atteindre son but final. Entre la position courante du robot et le prochain point de passage à atteindre, nous pouvons déterminer ce que l'on appelle une courbe de rattrapage. Une méthode pour tracer une telle courbe est d'utiliser les équations de Bézier, définie par quatre points de contrôles P1 à P4.

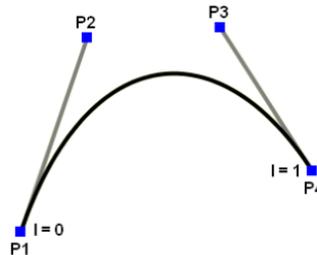


Figure II.5 : Courbe de Bézier

Cette courbe a pour équation paramétrique :

$$B(t) = P_1(1 - t)^3 + 3P_2(1 - t)^2 + 3P_3(1 - t) + P_4t^3 \text{ tel que } t \in [0,1] \quad (2.17)$$

Soit

$$\begin{cases} x(l) = (1 - l)^3x_1 + 3l(1 - l)^2x_2 + 3l^2(1 - l)x_3 + l^3x_4 \\ y(l) = (1 - l)^3y_1 + 3l(1 - l)^2y_2 + 3l^2(1 - l)y_3 + l^3y_4 \end{cases} \quad (2.18)$$

Avec  $0 < l < 1$

A partir de la courbe de rattrapage, nous pouvons déterminer une trajectoire de référence pour le robot, en associant une vitesse de référence à la courbe de rattrapage (Figure II.6). En multipliant cette vitesse de référence par l'horizon de prédiction du système, nous obtenons une distance de référence qui représente la distance que le robot peut raisonnablement parcourir le long de la courbe de rattrapage.

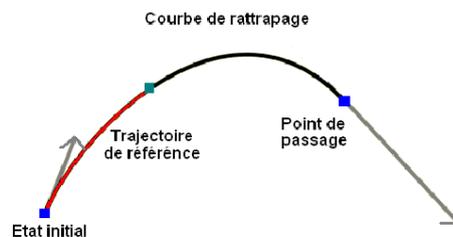


Figure II.6 : Courbe de rattrapage et trajectoire de référence

## II. MPC APPLIQUEES AU SUIVI DE LA TRAJECTOIRE

### Introduction

Le but du projet est la conception d'un robot mobile non holonome capable de suivre un chemin constitué de suites de droites. Pour cela, nous avons mis en cascade deux systèmes de régulation. D'une part la régulation en vitesse des moteurs constituant le robot et d'autre part la régulation sur le chemin.

Comme notre étude consiste à une étude comparative entre deux techniques de commandes, nous allons aborder en premier lieu la régulation en vitesse du moteur et la régulation sur le chemin par commande prédictive, ensuite nous allons voir la même structure de régulation mais avec la commande PID.

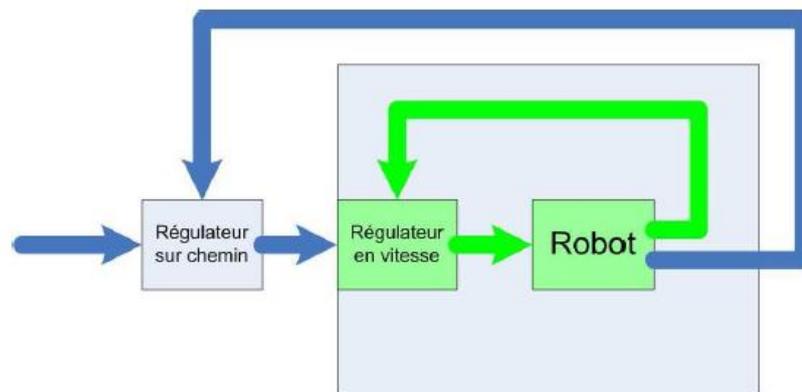


Figure II.7 : Les deux régulateurs en cascade[7]

Dans la littérature, la loi de commande pour le suivi de chemin se base sur l'hypothèse que les vitesses imposées au robot sont directement transmises. Cette hypothèse correspond à une régulation en vitesse parfaite du robot.

Cette séparation entre la régulation en vitesse du robot et le suivi de chemin a été adoptée pour plusieurs raisons :

- bonne séparation des tâches pour une meilleure lisibilité.
- meilleur potentiel de réutilisabilité.
- Détection plus facile des erreurs, plus facile lors de l'implémentation.

Ainsi, nous allons garder cette séparation, pour avoir deux régulations en cascade.

## 1. Régulation en vitesse des moteurs

Pour bien comprendre le fonctionnement de l'asservissement, nous allons nous focaliser sur la commande d'un seul moteur.

Une bonne régulation en vitesse du robot est nécessaire pour espérer un suivi de chemin correct.

La régulation en vitesse doit avoir les caractéristiques suivantes :

- simplicité de conception et de mise en œuvre.
- robustesse, pour ne pas avoir de comportements imprévisibles.
- peu de statisme, pour être proche du modèle idéal en régime permanent.

Pour approcher au mieux ces caractéristiques, nous proposons un régulateur MPC pour la régulation en vitesse du moteur.

Pratiquement, un régulateur MPC nécessite un modèle en temps discret du système LTI. Dans notre cas il s'agit de deux moteurs à courant continu.

### 1.1. Modélisation du moteur

Le système « moteur » est considéré comme un système SISO :

- une entrée : la tension appliquée au moteur.
- une sortie : la position angulaire du moteur, donnée par un encodeur (pratiquement).

Mais pour la mise en évidence de la régulation en vitesse, on se focalisera sur un seul moteur dont le modèle mathématique est le suivant:

$$\left\{ \begin{array}{l} u(t) = e(t) + R \cdot i(t) + L \frac{di}{dt} \\ e(t) = K_e \cdot \omega(t) \\ C_m(t) = K_c \cdot i(t) \\ C_m(t) - C_r(t) = J \frac{d\omega}{dt} \end{array} \right. \quad (2.19)$$

Avec

u : Tension appliquée au moteur

e : Force électromotrice

L, R : inductance et résistance de l'induit

K<sub>c</sub>, K<sub>e</sub> : constante de couple et constante électrique

i : Intensité traversant le moteur

ω : Vitesse de rotation du rotor

## Application

$C_m$  : Couple moteur généré

$C_r$  : Couple résistant

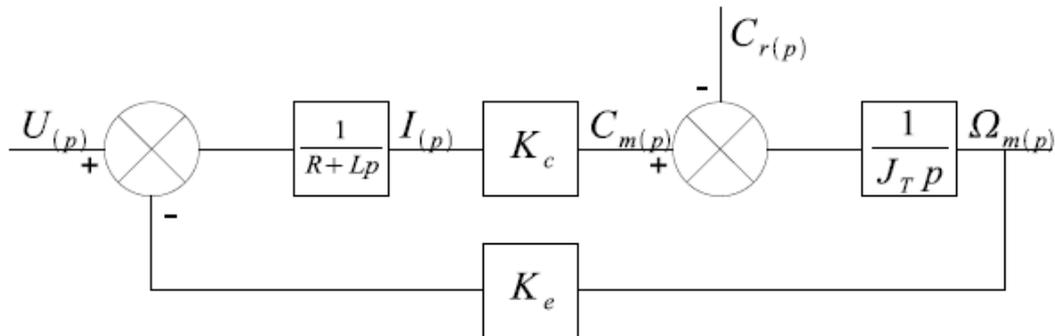


Figure II.8 : schéma fonctionnel du moteur

Comme il s'agit d'un asservissement en vitesse du moteur, la fonction de transfert dans le domaine de Laplace relatif à la tension appliquée au moteur est le suivant :

$$\frac{\Omega(p)}{U(p)} = \frac{Kc}{JLp^2 + JRp + KcKe} \quad (2.20)$$

La description du modèle sous formes de représentation d'état est nécessaire à l'approche MPC de l'asservissement. A partir de la fonction de transfert précédente on peut en déduire la représentation d'état du moteur [12].

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases} \quad (2.21)$$

$k$ : représente la discrétisation de temps,  $k = t \times Te$ .

$Te$  : la période d'échantillonnage du système.

A partir de la fonction de transfert du moteur, on peut ainsi avoir un modèle en représentation d'état pour pouvoir poursuivre l'étude sur l'approche MPC de sa commande en vitesse. De ce fait, il nous faut définir une technique pour extraire les matrices A, B, C, D de la fonction de transfert :

On a la relation directe suivante :

$$F(p) = \frac{\Omega(p)}{U(p)} = \frac{Kc}{JLp^2 + JRp + KcKe} = C(Ip - A)^{-1}B + D \quad (2.22)$$

Avec,

$p$ : représente la variable complexe de Laplace

$I$  : la matrice identité de même dimension que la matrice A.

## Application

La résolution de l'équation (2.22) peut être effectuée à partir d'un logiciel de calcul symbolique comme Mathematica qui nous donne les matrices de l'équation d'état suivant:

$$A = \begin{bmatrix} -\frac{R}{L} & -\frac{K_c K_e}{JR} \\ \frac{R}{L} & 0 \end{bmatrix}; B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; C = \begin{bmatrix} 0 & \frac{K_c}{JR} \end{bmatrix}; D = 0. \quad (2.23)$$

Notons que la représentation d'état d'un système n'est pas unique.

Dans la suite nous allons faire l'analyse du système :

### 1.2. Analyse du système moteur

#### a. Stabilité au sens de Lyapunov

La stabilité du système  $R(A, B, C)$  est, une propriété de la matrice d'évolution d'état  $A$  et ne dépend pas des matrices  $B$  et  $C$ . Pour étudier la stabilité du système, il nous suffit de calculer les valeurs propres de notre matrice d'évolution d'état  $A$  et de vérifier si ceux-ci sont à parties réelles négatives [12].

$$PR(\lambda_i(A)) \in R^- \quad (2.24)$$

#### b. Observabilité au selon les critères de Kalman

La vérification de l'observabilité du système est nécessaire en vue de la synthèse du correcteur. Pour ce faire on doit calculer la matrice d'observabilité  $M_0$  et de vérifier que le rang de celle-ci soit égale à la dimension de la matrice d'évolution d'état  $A$ .

$$M_0 = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (2.25)$$

#### c. Commandabilité selon les critères de Kalman

Le but étant la synthèse du bloc MPC en vue de la commande du système, il est de coutume de vérifier que ce dernier est commandable. Pour cela on calcule la matrice de commandabilité  $M_c$  est de vérifier que le rang de celle-ci soit égal à la dimension de la matrice d'évolution d'état  $A$ .

$$M_c = [B : AB : A^2B : \dots : A^{n-1}B] \quad (2.26)$$

### 1.3. Calcul de la commande

Une fois l'analyse de la stabilité, d'observabilité et de commandabilité faite, on peut passer au calcul de la commande qui est l'objectif principal de l'étude.

La prédiction de la sortie sera de la forme suivante [1]:

Application

$$\hat{\Omega}(k+i) = CA^{i-j-1}x(k) + \sum_{j=1}^{i-1} CA^{i-j-1}Bu(k+j) \quad (2.27)$$

Avec

$$u(k+j) = u(k-1) + \sum_{l=0}^{j-i} \Delta u(k+l) \quad (2.28)$$

On définit alors la fonction de coût quadratique :

$$J = \sum_{i=N1}^{N2} \|\hat{\Omega}(k+i) - \Omega_r(k+i)\|_{\tilde{Q}_j}^2 + \sum_{i=0}^{Nu-1} \|\Delta u(k+i)\|_{\tilde{R}_j}^2 \quad (2.29)$$

Avec,

N1 : Horizon minimal de prédiction

N2 : Horizon maximale de prédiction

Nu : Horizon de commande

$\tilde{Q}_j, \tilde{R}_j$  : Facteurs de pondération

La minimisation de cette fonction coût nous donne la séquence de commande future.

### a. Synthèse d'observateur

La commande MPC nécessite la synthèse d'observateur.

Une solution simple nous permet l'estimation de la vitesse instantanée par différenciation directe de la position[7] :

$$vitesse(k) = \frac{position(k) - position(k-1)}{période\ d'échantillonnage} \quad (2.30)$$

Ceci n'est qu'une approximation de la vitesse, mais elle suffisante pour notre application.

L'équation de l'observateur est le suivant [1]:

$$\tilde{x}_e(k+1) = A_e \tilde{x}_e(k) + B_e \Delta u(k) + K[\Omega(k) - C_e \tilde{x}_e(k)] \quad (2.31)$$

Il nous reste à définir la loi de commande :

### b. Conception du régulateur

Sur la base du modèle d'état discret, on peut synthétiser un régulateur d'état. Pour ce faire, la commande optimale nous semble bien adaptée à notre application. Comme il s'agit d'un asservissement et régulation, on travaille en variable d'écart autour du fonctionnement nominal.

## Application

La fonction coût (2.27) doit être mise sous forme matricielle :

$$J = \frac{1}{2} \sum_{k=0}^N [x^T(k) \tilde{Q}_j x(k) + \Delta u^T(k) \tilde{R}_j \Delta u(k)] \quad (2.32)$$

Les matrices de poids  $\tilde{Q}_j, \tilde{R}_j$  sont choisies diagonales positives et calculées selon les règles de Bryson [12].

Il nous reste à calculer le gain optimal qui va nous donner la tension à appliquer en boucle fermée.

$$\Delta u(k) = Kx(k) \quad (2.33)$$

On ajoute un intégrateur à l'asservissement pour éliminer les erreurs statiques :

$$u(k) = u(k-1) + \Delta u(k) \quad (2.34)$$

- **Action intégrale**

Afin de diminuer le statisme en régime permanent, il est nécessaire d'ajouter des intégrateurs sur les vitesses. L'action intégrale a été choisie itérativement en observant le résultat de la simulation, afin d'obtenir un statisme faible mais en évitant de dépasser la valeur de consigne. La matrice est donc de faible valeur. La matrice  $K_I$  (qui multiplie tous les états).

Le calcul du gain optimal, est issu de l'étude de la commande optimal en passant par l'hamiltonien associé au système on obtient l'expression du gain par [12]:

$$K = -R^{-1}B^T P \quad (2.35)$$

Le but de la régulation MPC est donc de fournir la commande à appliquer au système réel tout en considérant les contraintes liées à la tension d'alimentation disponible et les contraintes de vitesse maximale et couples maximaux du moteur.

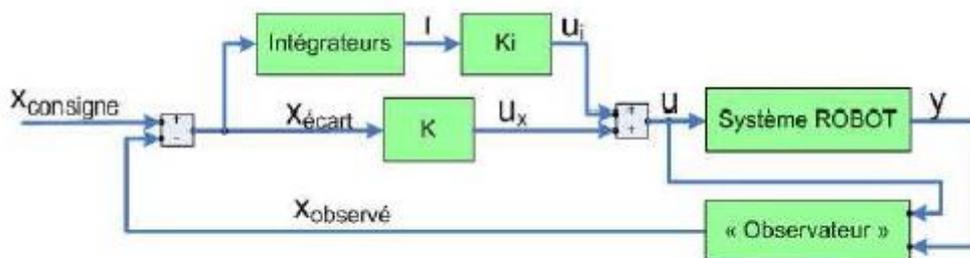


Figure II.9 : Schéma fonctionnel de la régulation en vitesse[7]

### c. Choix des paramètres

Le choix des horizons de prédiction et de commande est étroitement lié à la période d'échantillonnage et à la puissance des calculateurs numériques.

- **Horizon de prédiction  $N_p$**

Un horizon de prédiction de 15 semble bien adapté pour une période d'échantillonnage de 0.1s dans le cas de la régulation de vitesse.

- **Horizon de commande  $N_c$**

Un horizon de commande d'une valeur de 2 est suffisant pour la poursuite de consigne de vitesse. Un horizon de commande supérieur alourdirait inutilement les calculs.

## 2. Régulation sur le chemin

Une fois la régulation en vitesse faite, il nous faut un algorithme de suivi de chemin qui va définir un profil de vitesse à appliquer au robot.

L'objectif est de faire suivre au robot une trajectoire de référence constituée de segments de droites sans que la vitesse linéaire ne s'annule aux points d'intersection. Etant donné que la trajectoire est connue à priori, le but est alors de suivre au mieux cette trajectoire. Autrement dit, l'algorithme doit fournir un profil de vitesse à appliquer aux moteurs à partir de la donnée de la trajectoire de référence. Le rôle du régulateur est de réduire l'écart entre la consigne fournie par l'algorithme et la vitesse réelle des moteurs.



Figure II.10 : Entrée/sortie de l'algorithme

Le modèle est un robot « virtuel » que le robot réel doit suivre. Ce robot « virtuel » est placé à une distance  $N_p$  (Horizon de prédiction) devant le robot réel, et il parcourt la trajectoire de référence qui, pour le robot réel, est le fonctionnement nominal. Mais par le principe de la commande prédictive, seule la première séquence de commande sera réellement appliquée au robot donc le celui-ci ne va effectuer qu'une distance  $N_c$  (horizon de commande) de la distance effectuée par le robot virtuel. On réitère cet enchaînement tant que le robot virtuel n'a pas terminé de suivre la trajectoire de référence.

## Application

Nous allons illustrer le principe de fonctionnement par la suite de figures suivantes, considéré à chaque pas d'échantillonnage.

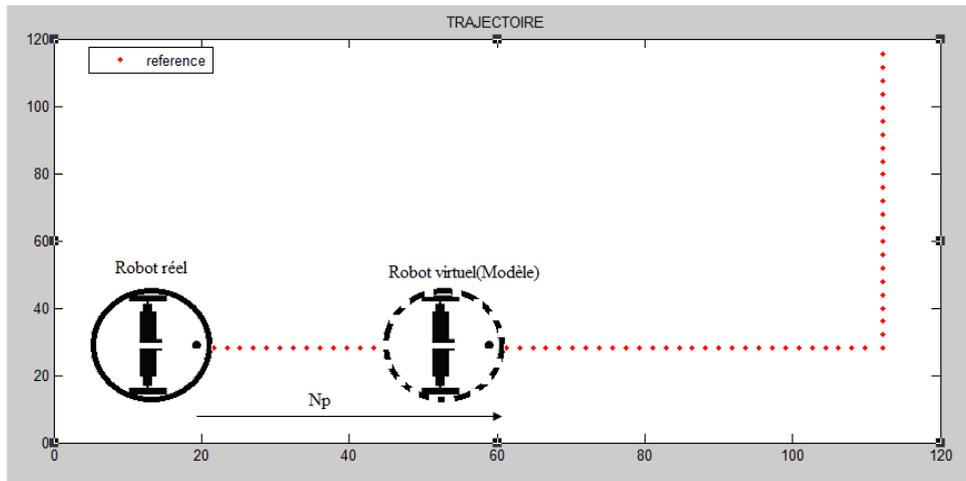


Figure II.11 : Présentation initiale du suivi de chemin à l'instant  $k$

A l'instant initial le robot est positionné au point de départ de la trajectoire de référence et orienté suivant l'angle du premier segment de droite à suivre.

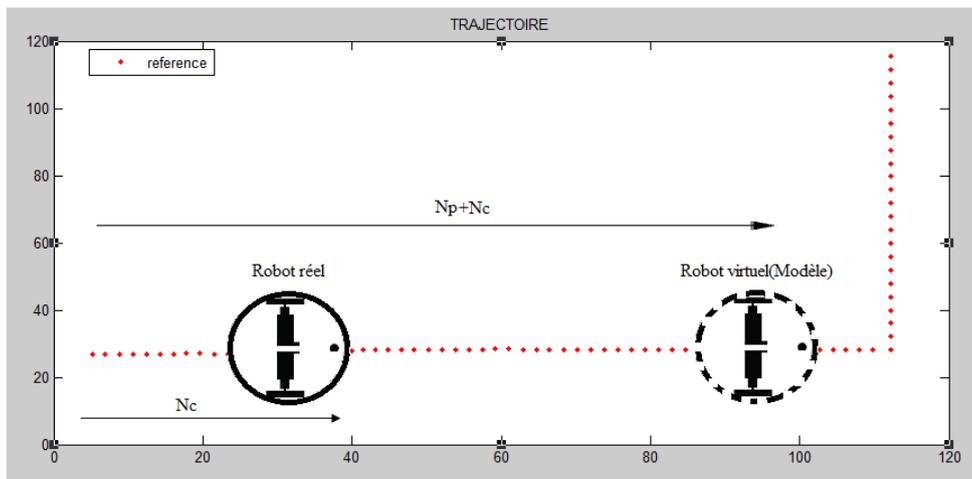


Figure II.12 : configuration à l'instant  $k+1$

A l'instant suivant, le robot virtuel aura parcouru une distance  $Np+Nc$  et le robot réel n'aura fait qu'une distance de  $Nc$  par rapport à l'origine de la trajectoire.

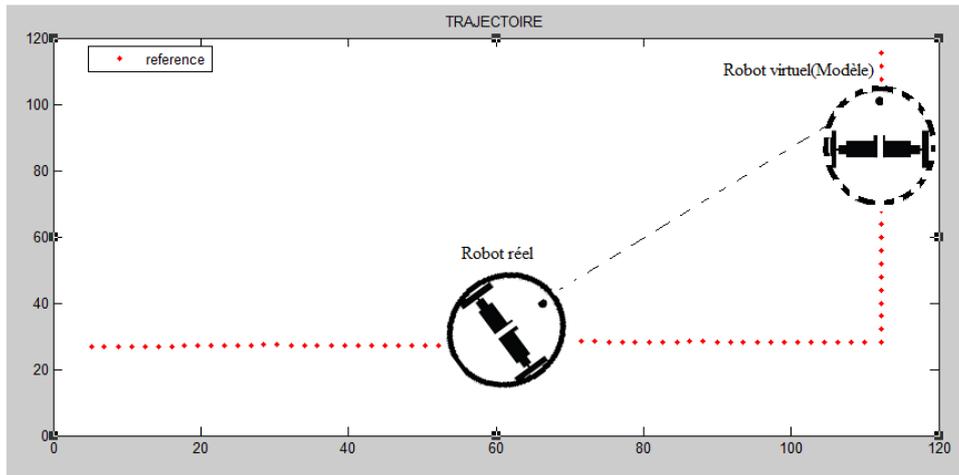


Figure II.13 : configuration pour l'anticipation du virage

Une fois que le robot virtuel a atteint la fin de la première section de droite, le robot réel va parcourir, non pas la trajectoire décrite par le modèle, mais va parcourir la droite formée par le robot virtuel et lui-même ; représentée sur la figure II.13 par le pointillé.

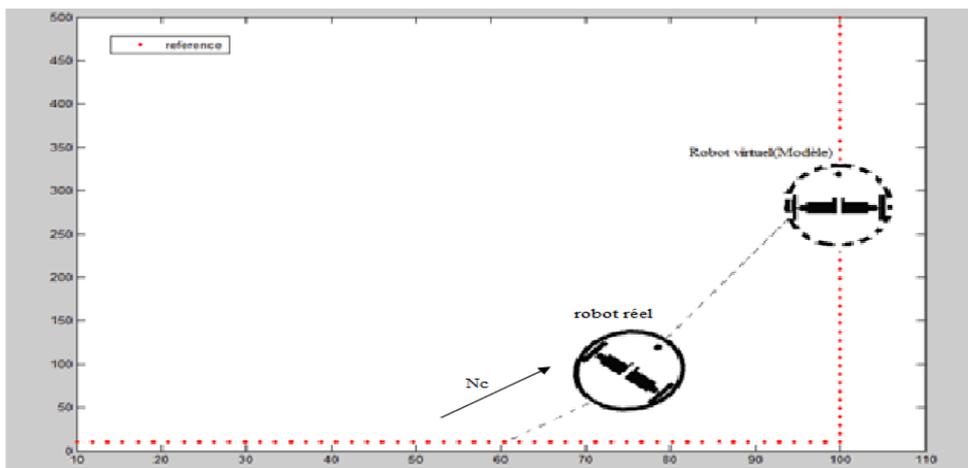


Figure II.14 : Poursuite du virage à l'instant suivant

A l'instant suivant, le robot prend l'orientation du segment de droite formé par le robot réel et son modèle et avance d'une distance  $N_c$  sur cette droite jusqu'à ce qu'il ait atteint à nouveau le second segment.

## 2.1. L'algorithme

Il convient de donner l'algorithme sous forme de diagramme :

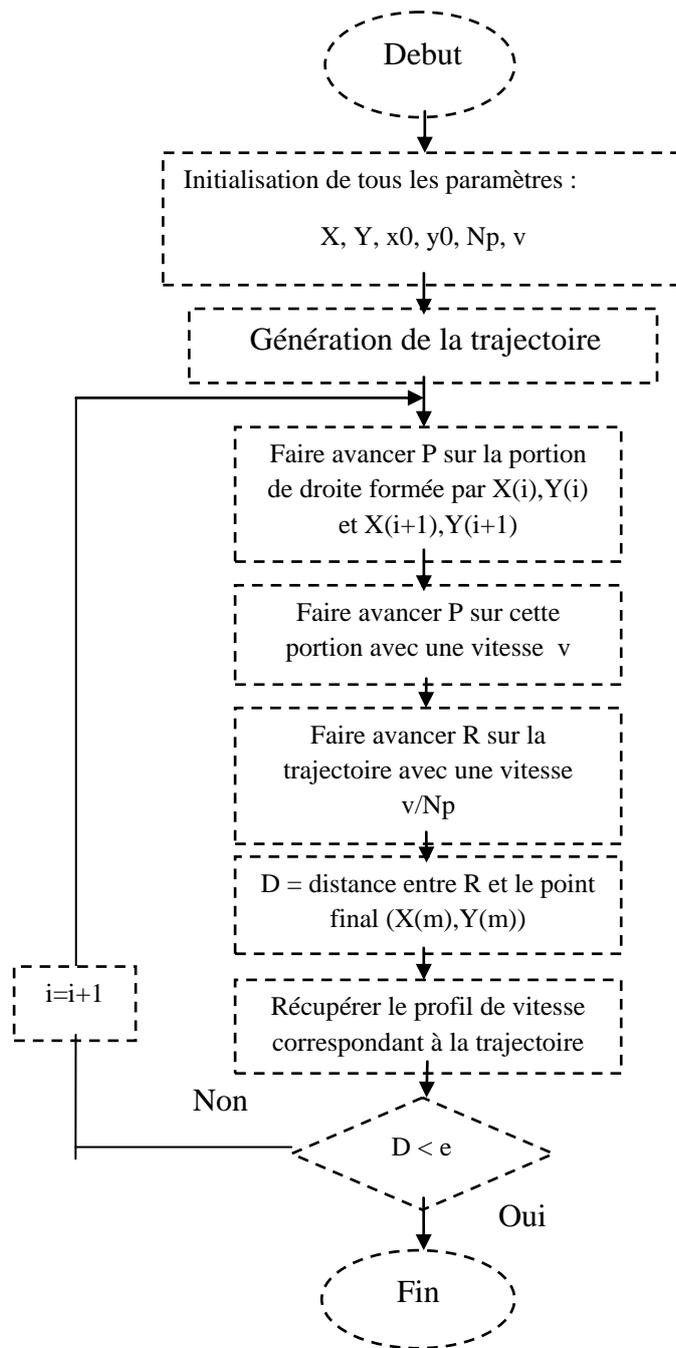


Figure II.15 : Algorithme de suivi de chemin

## 2.2. Interprétation

- Initialisation : On recueille toutes les informations sur la trajectoire, la position initiale du robot, et l'horizon de prédiction à appliquer.
- On génère la trajectoire à partir des points donnés à l'étape 1.
- On fait avancer le modèle, c'est-à-dire le robot virtuel (P) sur la trajectoire de référence avec une vitesse  $v$ .
- On calcule l'angle que fait la droite qui relie le robot réel (R) avec son modèle(P).

## Application

- On fait avancer (R) sur la droite PR avec une vitesse  $v/Np$ .
- On récupère le profil de vitesse.
- On calcule la distance de R par rapport au dernier point de la trajectoire. On réitère les étapes 3-4-5-6 tant que cette distance est supérieure à une distance petite ( $\epsilon$ ).