

CONCLUSION

Dans ce travail, l' algorithme de régulation MPC a été inséré dans le modèle odométrique d'un robot mobile non-holonome et à roues différentielles afin de guider et garder en même temps ce dernier sur une trajectoire que nous imposons via l'insertion des coordonnées d'une succession de points. Pour cela il nous a fallu effectuer une étude de deux régulations en cascade, l'une consistant à réguler la vitesse du robot (régulation de la vitesse des moteurs) l'autre à réguler sa position (traduire la trajectoire en un profil de vitesse applicable aux deux roues pour pouvoir effectuer son suivi).

En posant comme hypothèse simplificatrice le fait que le mouvement des roues est un roulement sans glissement et que la trajectoire est décrit sur un terrain plat, il a pu être constaté que l'algorithmes prédictif permet de réaliser le suivi de trajectoires quelconques. Qui plus est, les principes développés permettent de conserver un comportement similaire au cas réel, malgré de multiples variations des paramètres pouvant intervenir (vitesse linéaire du mobile, angle d'inclinaison du mobile par rapport à la trajectoire, horizon de prédiction, distance du mobile par rapport à la trajectoire ...).

Toutefois, grâce à l'algorithme étudié dans ce travail, il a été remarqué que la commande MPC use bien de son nom d'algorithme de prédiction car lors de l'emprunt d'un virage, la commande MPC grâce à N_p envisage à l'avance le virage. En effet, à la sortie de l'algorithme, on obtient un profil de vitesse permettant au robot un enchaînement fluide des segments de droite formant la trajectoire.

Pour terminer, déjà que cette première approche dans l'étude du mouvement d'un robot non holonome soit un pas pour s'initier dans le domaine de l'automatique mobile qui est très vaste. Ainsi plusieurs études sur la modélisation du robot non holonome reste à voir dont la considération des obstacles présents au cours du mouvement (ce qui implique une étude de l'odométrie visuelle), la considération d'un roulement avec glissement, la prise en compte des caractéristiques du terrain de travail (terrain en pente, sol plat ou rugueux,etc.). Cependant les intérêts qu'elle nous apporte sont d'ores-et-déjà nombreux dont toutes applications sur le guidage d'engins (assistance de conduite, control d'une main robot,etc.) et rejoint naturellement tout l'intérêt des systèmes embarqués.

ANNEXE 1

Historique de la commande MPC

- Aperçu chronologique et application industrielle de la commande prédictive

La première notion de commande par prédiction a connu le jour vers l'année 1959 par Smith. Ce principe, appelé « Prédicteur de Smith » avait pour but de compenser l'effet de retard pur dans la boucle de commande notons que retard en approche systémique correspond au temps nécessaire pour le transport de matières ou d'énergie. En d'autre terme, elle consiste à prédire la sortie du système avec une avance égale au retard pur estimé L qui pourra après être comparée à la consigne. (voir fig A.1.1.).Cependant, cette méthode présentait aussi ses inconvénients d'être à la fois très sensible aux erreurs de modélisation et est incapable de stabiliser des procédés instables en boucle ouverte.

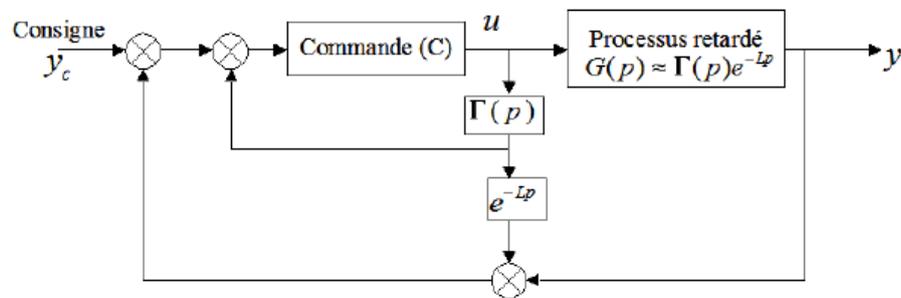


Figure A.1.1. Principe du prédicteur de Smith

Au début des années 1960, Propoi lui, s'est plutôt intéressé à la stabilité de l'approche par commande prédictive en se basant sur un modèle en utilisant la méthode de la programmation linéaire dont le principe est d'insérer dans l'algorithme de commande, un élément de prédiction de l'évolution de la sortie du procédé donnée par un modèle. Le calculateur détermine alors, à l'instant d'échantillonnage présent, la séquence de commandes à appliquer sur un horizon de prédiction. A la prochaine période d'échantillonnage, seule la première composante de cette séquence solution est effectivement appliquée au procédé et la résolution recommence de la même façon en prenant en compte les nouvelles mesures du procédé et ainsi de suite. Néanmoins, ce ne fut qu'au milieu des années 70 que la commande prédictive a été instaurée au sein de l'industrie grâce à l'utilisation du logiciel dénommé IDCOM, développé par Richalet et Al. Par ce fait, les premières applications de la commande prédictive ont été réalisées dans le domaine pétrolier et pétrochimique. En effet, jusqu'à nos jours, de nombreuses catégories et dénominations de la commande prédictive ont été proposées ; nous pouvons en citer quelques uns suivant l'évolution du principe de cette méthode prédictive :

- Premièrement, le MPHC (Model Predictive Heuristic Control), encore dénommé MAC (Model Algorithmic Control) qui est une approche appliquée aux systèmes industriels multivariables et est basée sur des prédictions sur un horizon temporel

étendu, impose des trajectoires de référence pour les sorties et minimise la variance de l'erreur. Le modèle du procédé utilisé pour générer les valeurs prédites, sur tout l'horizon fixé est basé sur la réponse impulsionnelle.

- Deuxièmement, le DMC (Dynamic Matrix Control) proposé par des ingénieurs de Shell, Cutler et Ramaker, reprend de nombreuses idées de IDCOM, mais les systèmes y sont représentés par leur réponse à un échelon. Cet algorithme est appliqué à des systèmes multivariables linéaires sans contraintes ainsi, l'erreur de poursuite est minimisée en spécifiant le comportement futur des sorties et les commandes optimales sont calculées par la méthode des moindres carrés.
- Troisièmement, l'EHAC (Extended Horizon Adaptive Control) qui est une stratégie de commande prédictive pour les systèmes monovariables, utilise des modèles E/S pour maintenir la sortie future (calculée via la résolution d'une équation diophantienne) le plus près possible de la consigne pendant une période donnée au-delà du retard pur du système.
- Quatrièmement, l'EPSAC (Extended Prediction Self-Adapted Control) introduit une commande constante pour un système non-linéaire et utilise un prédicteur sous-optimal à la place de la résolution de l'équation diophantienne ; l'article offre également une démonstration de stabilité.
- Cinquièmement, le GPC (Generalized Predictive Control), basée sur un modèle de type CARIMA, introduit un horizon de prédiction sur la commande et agit conformément au principe de l'horizon fuyant. Elle peut être appliquée aux systèmes à non minimum de phase, aux systèmes instables en boucle ouverte ainsi qu'aux systèmes avec retards purs variables.
- Sixièmement, le PFC (Predictive Functional Control) qui est un algorithme prédictif simple, utilisé surtout pour des systèmes SISO industriels rapides et/ou non linéaires. Il permet le réglage direct des paramètres (par exemple la constante de temps) associées au temps de montée donc pratique pour les ingénieurs. Cependant, son algorithme est peu performant pour cause de simplicité d'utilisation.
- Septièmement, le CRHPC (Constrained Receding Horizon Predictive Control) qui propose de prendre en compte des contraintes terminales sous forme « égalité » sur la sortie sur un horizon fini au-delà de l'horizon de prédiction ;
- Enfin, la MPC (Model Predictive Control) formulée dans l'espace d'état, utilise le formalisme de la représentation d'état pour faciliter l'analyse de la stabilité et de la robustesse.

Ces différentes catégories de méthode prédictive citées sont considérées comme l'essence de la méthode prédictive proprement dite ; néanmoins, concernant son évolution dans le

domaine industrielle, on peut recenser quatre (4) générations de MPC si on se base sur l'évolution de son algorithme. Cette chronologie d'évolution est résumée par la figure suivante.

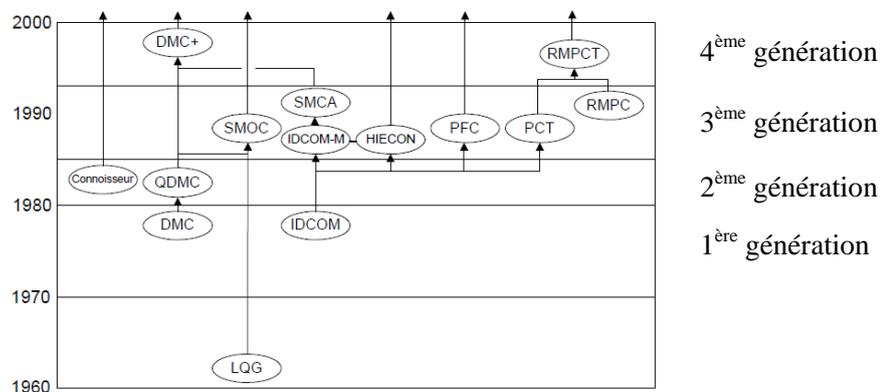


Figure A.1.2. Généalogie approximative d'algorithme de MPC linéaire

De toutes les commandes dites avancées, la commande prédictive doit sa popularité dans le domaine industriel du fait de sa performance grâce au concept technique auquel elle a été conçue, ainsi qu'à sa facilité de mise en œuvre. Son application s'étend dans plusieurs secteurs industriels surtout là où le régulateur PID présente ses limites.

Depuis les années 1970, la commande prédictive touche plusieurs champs d'activités et a été utilisée avec succès dans différentes applications industrielles. En effet, en 2003 plus de 4000 applications au total, dans le monde, fut recensées selon qu'elle soit exploitée pour des approches linéaires ou non linéaires cela, sans rajouter les solutions locales développées en interne des entreprises. (Voir les tableaux A.1.1 et A.1.2)

Area	Aspen Technology	Honeywell Hi-Spec	Adersa	Invensys	Total	%
Raffinage	1200	480	280	25	1985	48,5
Pétrochimie	450	80	-	20	550	3,4
Chimie	100	20	3	21	144	3,5
Papier	18	50	-	-	68	1,7
Air et Gaz	-	10	-	-	10	0,2
Métallurgie	8	6	7	16	37	0,9
Alimentaire	-	-	41	10	51	1,3
Polymère	17	-	-	-	17	0,4
Cuisson	-	-	42	3	45	1,1
Aérospatial	-	-	13	-	13	0,3
Automobile	-	-	7	-	7	0,2
Autre	40	50	1045	30	1165	28,5
Total	1833	696	1438	125	4092	100

Tableau A.1.1. Bilan des applications industrielles de la commande prédictive linéaire en 2003

Area	Adersa	Aspen Technology	Continental Controls	DOT Products	Pavillon Technologies	Total	%
Chimie	2		15		5	22	23,7
Polymère		1		5	15	21	22,6
Air et gaz			18			18	19,3
Raffinage					13	13	14
Alimentaire					9	9	9,7
Autres	1	5	3			9	9,7
Papier					1	1	1
Total	3	6	36	5	43	93	100

Tableau A.1.2. Bilan des applications industrielles de la commande prédictive non linéaire en 2003

Ces deux tableaux montrent que la commande prédictive est présente dans la majorité des domaines et que, selon les chiffres, la commande prédictive a tout au moins démontré ses performances dans diverses applications. En outre, même si la commande prédictive offre de nombreuses caractéristiques intéressantes, historiquement, seules deux d'entre elles ont joué un rôle déterminant ; ce sont : le domaine du raffinage et de la pétrochimie. Cependant, l'application de cette technique n'a cessé d'évoluer et de multiples avancées ont été réalisées dans des domaines plus variés (médecine : robot de chirurgie, verrerie...).

ANNEXE 2

Algorithme de Levenberg Marquardt

L'algorithme de Levenberg-Marquardt, ou algorithme LM, permet d'obtenir une solution numérique au problème de minimisation d'une fonction, souvent non linéaire et dépendant de plusieurs variables. L'algorithme interpole l'algorithme de Gauss-Newton et l'algorithme du gradient. Plus stable que celui de Gauss-Newton, il trouve une solution même s'il est démarré très loin d'un minimum. Cependant, pour certaines fonctions très régulières, il peut converger légèrement moins vite. L'algorithme fut découvert par Kenneth Levenberg, puis publié par Donald Marquardt.

C'est un problème qui se présente souvent en régression linéaire et non linéaire.

A.2.1. Application à la méthode des moindres carrés

Énoncé

Son application principale est la régression au travers de la méthode des moindres carrés : étant donné un certain nombre de paires de données (t_i, y_i) , on cherche le paramètre \mathbf{a} de la fonction $f(t|\mathbf{a})$ de sorte que la somme des carrés des déviations :

$$S(\mathbf{a}) = \sum_{i=1}^m [y_i - f(t_i|\mathbf{a})]^2$$

soit minimale.

Solution

La procédure de l'algorithme est itérative. On part d'un paramètre initial, que l'on suppose « assez proche » d'un minimum, et qui constituera le vecteur \mathbf{p} de départ. Dans beaucoup de cas, un paramètre de départ « standard », tel que $\mathbf{p}^T=(1,1,\dots,1)$ fonctionnera sans problème. Dans certains cas, il n'y a convergence que si le vecteur de départ n'est pas trop éloigné de la solution.

À chaque itération, on remplace \mathbf{p} par une nouvelle estimation $\mathbf{p} + \mathbf{q}$. Afin de déterminer \mathbf{q} , les fonctions $f_i(\mathbf{p} + \mathbf{q})$ sont approchées en étant linéarisées :

$$f(\mathbf{p} + \mathbf{q}) \approx f(\mathbf{p}) + \mathbf{J}\mathbf{q}$$

où on a noté \mathbf{J} la jacobienne de \mathbf{f} en \mathbf{p} .

À un minimum de la somme des carrés S , on a : $\nabla_{\mathbf{q}} S = 0$. En dérivant le carré de l'expression [1] de droite, qui s'annule donc, on obtient :

$$(J^T J)q = J^T[y - f(p)]$$

d'où l'on tire aisément q en inversant $J^T J$.

Dans l'inversion matricielle, tout va dépendre du rapport entre la valeur propre la plus "grande" en norme, et la valeur propre la plus petite ; ce rapport, appelé "conditionnement de matrice", va concrètement refléter la robustesse de l'algorithme face au bruit.

Le point essentiel de l'algorithme de Levenberg-Marquardt est d'approcher cette équation, en l'« amortissant » un peu.

$$(J^T J + \lambda \cdot \text{diag}(J^T J))q = J^T[y - f(p)].$$

Le facteur d'amortissement positif λ est ajusté à chaque nouvelle itération. Si la diminution de S est rapide, on peut utiliser une valeur plus faible ce qui rapproche l'algorithme de celui de Gauss-Newton. Si en revanche une itération est peu efficace, on peut augmenter λ ce qui rapproche cette fois l'algorithme de celui du gradient. Un tel facteur d'amortissement est utilisé par exemple dans la régularisation de Tikhonov, utilisée pour résoudre certains problèmes linéaires.

Si on a effectué plus d'un certain nombre d'itérations, ou bien que l'on s'est approché suffisamment d'un minimum, la procédure se termine et renvoie le paramètre p comme estimation de la solution.

A.2.2. Choix du paramètre d'amortissement

De nombreux arguments, plus ou moins heuristiques ont été proposés afin de déterminer le meilleur facteur d'amortissement λ . Des démonstrations théoriques montrent que certains choix garantissent une convergence locale - mais peuvent afficher une convergence faible près de l'optimum.

La valeur absolue de tout choix dépend de l'échelle du problème. Marquardt recommandait de commencer à partir de λ_0 et avec un facteur $\nu > 1$. On pose alors au départ $\lambda = \lambda_0$ et calculons la somme des carrés des déviations $S(p)$ après une itération, en utilisant le facteur d'amortissement $\lambda = \lambda_0$, puis en utilisant λ/ν . Si les deux derniers renvoient un point moins bon que le point de départ, on augmente λ en le multipliant par ν , jusqu'à atteindre un point meilleur avec un nouveau facteur $\lambda \nu^k$ pour un certain k .

Si l'utilisation du facteur λ/ν donne une somme plus faible, alors il est pris comme nouvelle valeur de λ et l'algorithme continu. Si l'utilisation de λ/ν donne une somme plus importante, mais que l'utilisation de λ donne une somme plus faible, alors λ est conservé.

ANNEXE 3

Matrice HESSIENNE

En mathématiques, la matrice hessienne (ou simplement la hessienne) d'une fonction numérique f est la matrice carrée, notée $H(f)$, de ses dérivées partielles secondes.

Plus précisément, étant donnée une fonction f à valeurs réelles $f: \mathbb{R}^n \rightarrow \mathbb{R} : f(x_1, x_2, \dots, x_n)$, et en supposant que toutes les dérivées partielles secondes de f existent, le coefficient d'indice i, j de la matrice hessienne de f vaut

$$H_{ij}(f) = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

ou, en d'autres termes,

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

On appelle hessien (ou discriminant hessien) le déterminant de cette matrice.

Le terme « hessien » a été introduit par James Joseph Sylvester, en hommage au mathématicien allemand Ludwig Otto Hesse.

Soit notamment f une fonction de classe \mathcal{C}^2 définie sur un ouvert U de l'espace E , à valeurs réelles. Sa matrice hessienne est bien définie et en vertu du théorème de Schwarz, elle est symétrique.

On appelle forme hessienne la forme quadratique associée à la matrice hessienne lorsque celle-ci est symétrique.

Application à l'étude des points critiques

On suppose f une fonction de classe \mathcal{C}^2 sur un ouvert U . La matrice hessienne permet, dans de nombreux cas, de déterminer la nature des points critiques de la fonction f , c'est-à-dire des points d'annulation du gradient.

- Condition nécessaire d'extremum local

- si a est un point de minimum local de f , alors c'est un point critique et la hessienne en a est positive (c'est-à-dire que la forme hessienne est positive)
- si a est un point de maximum local de f , alors c'est un point critique et la hessienne en a est négative (c'est-à-dire que la forme hessienne est négative)

En particulier, si la hessienne en un point critique admet au moins une valeur propre strictement positive et une valeur propre strictement négative, le point critique est un point col.

- **Condition suffisante d'extremum local**

Précisément, un point critique de f est dit **dégénéré** lorsque le discriminant hessien s'annule, autrement dit lorsque 0 est valeur propre de la hessienne. En un point critique non dégénéré, le signe des valeurs propres (toutes non nulles) détermine la nature de ce point (point d'extremum local ou point col):

- si la hessienne est définie positive, la fonction atteint un minimum local au point critique
- si la hessienne est définie négative, la fonction atteint un maximum local au point critique
- s'il y a des valeurs propres de chaque signe, le point critique est un point col.

Dans ce dernier cas, on définit l'indice du point critique comme le nombre de valeurs propres négatives.

En dimension deux notamment, le discriminant hessien étant le produit des valeurs propres, son signe suffit à déterminer la nature d'un point critique non dégénéré.

Enfin pour un point critique dégénéré, aucune de ces implications n'est vraie.

- **Courbe hessienne**

Si C est la courbe algébrique d'équation projective (homogène) $f(x, y, z) = 0$, on appelle courbe hessienne (ou simplement hessienne) de C la courbe dont l'équation projective est $|H(f)|(x, y, z) = 0$, où $|H(f)|$ est le hessien (le déterminant de la matrice hessienne) de f . La hessienne de C a pour intersection avec C les points critiques et les points d'inflexion de C . Si C est de degré d , sa hessienne est de degré $3(d-2)$; d'après le théorème de Bézout, le nombre des points d'inflexion d'une courbe régulière de degré d est donc $3d(d-2)$, ce qui est un cas particulier d'une des formules de Plücker.

ANNEXE 4

Fonction atan2 de Matlab

En trigonométrie, la fonction atan2 à deux arguments est une variation de la fonction arctangente. Pour tout arguments réels x et y non nuls, $\text{atan2}(y,x)$ est l'angle en radians entre la partie positive de l'axe des x d'un plan, et le point de ce plan aux coordonnées (x,y) . Cet angle est positif pour les angles dans le sens anti-horaire dit sens trigonométrique (moitié haute du plan, $y > 0$) et négatif dans l'autre (moitié basse du plan, $y < 0$).

La fonction atan2 fut en premier lieu introduite dans les langages de programmation informatique, mais elle est désormais aussi couramment utilisée dans les autres sciences et ingénieries. Elle est au moins aussi vieille que le langage de programmation Fortran et on la trouve maintenant dans la plupart des autres langages.

En termes mathématiques, atan2 retourne la valeur principale de la fonction argument appliquée sur le nombre complexe $x+iy$. Soit :

$$\text{atan2}(y, x) = \text{Pr arg}(x+iy) = \text{Arg}(x+iy)$$

Le résultat pourrait varier de 2π sans aucun impact sur l'angle, mais pour garantir son unicité, on utilise la valeur principale dans l'intervalle $] -\pi ; \pi]$. Soit $-\pi < \text{atan2}(y, x) \leq \pi$.

La fonction atan2 est utilisée dans beaucoup d'applications impliquant des vecteurs de l'espace Euclidien, comme pour trouver la direction d'un point à un autre. Une des utilisations principales est la conversion des matrices de rotation en angles d'Euler, pour faire pivoter des représentations graphiques informatiques.

Dans certains langages informatiques, l'ordre des paramètres est inversé, ou la fonction est dénommée différemment. Sur les calculatrices scientifiques, le résultat de la fonction est souvent issu de la conversion des coordonnées rectangulaires (x,y) en coordonnées polaires.

Motivation

La fonction arc tangente à un seul argument ne fait pas la différence entre des directions diamétralement opposées. Par exemple, l'angle antihoraire à partir de l'axe des x jusqu'au vecteur $\langle 1, 1 \rangle$, calculé de la façon habituelle comme $\arctan(1/1)$, donne $\pi/4$ (radians), ou 45° . De la même façon, l'angle entre l'axe des x jusqu'au vecteur $\langle -1, -1 \rangle$, avec la même méthode $\arctan(-1/-1)$, donne encore $\pi/4$, même si la réponse clairement attendue serait plutôt $-3\pi/4$, ou -135° .

La fonction "atan2" prend en compte le signe des deux composants du vecteur, et place l'angle dans le bon quadrant. Ainsi, $\text{atan2}(1, 1) = \pi/4$ et $\text{atan2}(-1, -1) = -3\pi/4$.

De plus, la méthode habituelle ne fonctionne pas dans le cas d'un angle de $\pm\pi/2$ (or $\pm 90^\circ$). Par exemple, une tentative pour trouver l'angle entre l'axe des x et le vecteur $\langle 0, 1 \rangle$, nécessite

l'évaluation de arctan (1/0), qui échoue à cause de la division par zéro. Alors qu'atan2 (1,0) donne la bonne réponse $\pi/2$.

Quand les calculs sont effectués à la main, les corrections de quadrant nécessaires et la gestion des exceptions peuvent être faites par observation, mais dans un programme informatique, il est extrêmement utile de disposer d'une fonction unique qui délivre toujours un résultat correct non ambigu.

- **Définition**

Pour $y \neq 0$:

$$\text{atan2}(y, x) = \begin{cases} \varphi \cdot \text{sgn}(y) & x > 0 \\ \frac{\pi}{2} \cdot \text{sgn}(y) & x = 0 \\ (\pi - \varphi) \cdot \text{sgn}(y) & x < 0 \end{cases}$$

Où φ est l'angle compris dans $[0, \pi/2[$ de façon à ce que $\tan(\varphi) = \left| \frac{y}{x} \right|$ et **sgn** est la fonction signe.

Et:

$$\text{atan2}(0, x) = \begin{cases} 0 & x > 0 \\ \text{non défini} & x = 0 \\ \pi & x < 0 \end{cases}$$

Note :

- Cela donne des résultats compris dans $]-\pi, \pi]$, qui peut être translaté dans $[0, 2\pi[$ en ajoutant 2π aux valeurs négatives.
- Traditionnellement, atan2(0,0) est non défini.
 - La fonction C, et la plupart des autres implémentations informatiques, sont conçues pour les efforts de transformations des coordonnées cartésiennes en polaires, et donc définissent toujours atan2(0,0). Dans les implémentations sans zéro signé, ou lorsque sont donnés des arguments zéro positif, le résultat est normalement défini à 0. La valeur rendue sera toujours comprise dans $]-\pi, \pi]$ plutôt que sortir en erreur ou retourner NaN ("Not a Number", i.e "Pas un Nombre").
 - Les systèmes gérant les mathématiques symboliques devraient normalement retourner une valeur non définie pour atan2(0,0) ou bien signaler qu'une anomalie s'est produite.

Pour les systèmes comme la virgule flottante IEEE 754 qui savent gérer le zéro signé, les infinis ou NaN, il est habituel d'implémenter des extensions capables d'étendre les résultats possibles pour inclure $-\pi$ et -0 . Ces dernières pourraient aussi retourner NaN ou lever une exception lorsqu'un des arguments reçus est un NaN.