

La cinématique embarquée au cœur des contrôleurs

Autor(en): **John, Othmar / Gaille, François**

Objekttyp: **Article**

Zeitschrift: **Bulletin.ch : Fachzeitschrift und Verbandsinformationen von Electrosuisse, VSE = revue spécialisée et informations des associations Electrosuisse, AES**

Band (Jahr): **96 (2005)**

Heft 15

PDF erstellt am: **24.09.2024**

Persistenter Link: <https://doi.org/10.5169/seals-857826>

Nutzungsbedingungen

Die ETH-Bibliothek ist Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Inhalten der Zeitschriften. Die Rechte liegen in der Regel bei den Herausgebern.

Die auf der Plattform e-periodica veröffentlichten Dokumente stehen für nicht-kommerzielle Zwecke in Lehre und Forschung sowie für die private Nutzung frei zur Verfügung. Einzelne Dateien oder Ausdrucke aus diesem Angebot können zusammen mit diesen Nutzungsbedingungen und den korrekten Herkunftsbezeichnungen weitergegeben werden.

Das Veröffentlichen von Bildern in Print- und Online-Publikationen ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Die systematische Speicherung von Teilen des elektronischen Angebots auf anderen Servern bedarf ebenfalls des schriftlichen Einverständnisses der Rechteinhaber.

Haftungsausschluss

Alle Angaben erfolgen ohne Gewähr für Vollständigkeit oder Richtigkeit. Es wird keine Haftung übernommen für Schäden durch die Verwendung von Informationen aus diesem Online-Angebot oder durch das Fehlen von Informationen. Dies gilt auch für Inhalte Dritter, die über dieses Angebot zugänglich sind.

La cinématique embarquée au cœur des contrôleurs

Algorithmes cinématiques directs et inverses dans les contrôleurs de robots industriels

Asservir un axe ou une articulation avec une haute précision et une grande puissance dans une machine ou un robot peut s'avérer être une opération difficile, particulièrement si le mouvement désiré est la combinaison de plusieurs moteurs couplés. La cinématique directe et inverse permet de faire le pont entre la position de l'extrémité du système (par exemple l'outil) et l'action nécessaire sur les axes ou les moteurs afin d'atteindre ce point. Ceci nécessite de la part du contrôleur une puissance de calcul embarquée plus importante et bien évidemment des algorithmes adéquats.

Dans les domaines tels que la robotique [1], l'animation virtuelle sur ordinateur ou encore tous les domaines nécessitant un asservissement d'un système dans l'espace, il est nécessaire de positionner précisément l'extrémité du mécanisme puis, de retrouver par la suite les coordonnées de l'extrémité dudit système.

Othmar John et François Gaille

En robotique, afin de permettre à l'opérateur de programmer la trajectoire du système contrôlé, il est nécessaire que le robot sache comment il doit se mouvoir pour se rendre au point imposé. Le contrôleur est chargé d'interpréter les coordonnées de l'utilisateur et de traduire celles-ci en coordonnées articulaires du robot. C'est le principe même de la cinématique.

Les équations de cinématique modélisant le comportement des robots sont soit des équations liées aux contraintes des articulations (robot série), soit des équations de fermeture de boucle (robot parallèle). Dans les deux cas, ces équations comportent une description mathématique découlant des relations trigonométriques. De part leur nature, des relations fortement non linéaires sont obtenues, pouvant poser des problèmes complexes quant à leur résolution.

La cinématique des robots a été étudiée sous des approches différentes. La

plupart s'appuient sur des résultats antérieurs utilisés pour l'étude des mécanismes ou sur des outils mathématiques plus généraux.

Cet article a pour objectif de rappeler dans un premier temps les notions fondamentales de cinématique directe et inverse utilisées dans les robots industriels. Puis la méthodologie employée pour résoudre la problématique de la cinématique d'une chaîne articulée (robot série) sera décrite. Finalement, un exemple d'implémentation des résultats obtenus dans les contrôleurs multi-axes programmables PMAC¹⁾ sera présenté.

La cinématique

La cinématique joue un rôle capital dans la description des mouvements des

robots industriels dans l'espace. Par définition, il s'agit de l'étude des systèmes mécaniques à chaque instant t du point de vue de leur position et de leur mouvement dans l'espace, ceci indépendamment des causes engendrant ces mouvements. La cinématique est un ensemble d'équations définissant la position et le mouvement d'un système physique tel que, par exemple, le bras d'un robot anthropomorphique.

Quel que soit le domaine étudié, le problème de la cinématique est divisé en deux: la cinématique directe (Forward Kinematics) et la cinématique inverse (Inverse Kinematics). La cinématique directe et inverse permet donc de faire le pont entre la position de l'extrémité du système (par exemple l'outil) et l'action nécessaire sur les axes ou les moteurs afin d'atteindre ce point.

La cinématique directe correspond à l'ensemble des relations mathématiques liant l'état actuel du robot ou de la machine (état défini par la position de ses axes ou de ses articulations) au point visé par l'extrémité de son outil ou de son préhenseur (figure 1a). La cinématique directe est utilisée lors de l'initialisation du système de contrôle afin de définir la position et l'orientation de l'extrémité du robot (ou de l'outil) asservi.

La cinématique inverse est, quant à elle, l'ensemble des relations mathématiques liant un point de l'espace donné par la position et l'orientation de l'extrémité du robot, aux valeurs articulaires que devrait avoir le système pour at-

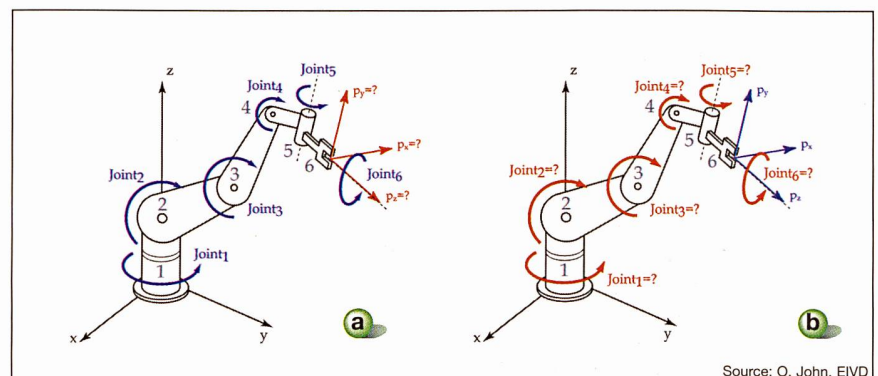
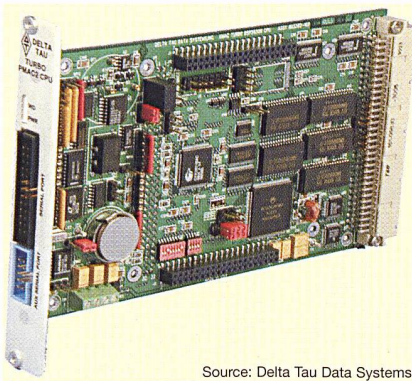


Figure 1 Définition de la cinématique directe a) et inverse b)



Source: Delta Tau Data Systems

Figure 2 Vue d'un contrôleur Turbo PMAC 3U

teindre ce point (figure 1b). Le problème de la cinématique inverse est plus délicat que celui de la cinématique directe. En effet, la cinématique inverse est soumise à la problématique de configurations multiples. A savoir, pour un point donné par la position et l'orientation de l'extrémité du système, plusieurs configurations sont généralement possibles. De plus, un robot peut perdre un ou plusieurs degrés de liberté suivant sa configuration.

Afin d'illustrer l'importance de la cinématique, prenons l'exemple d'une table traçante dans le plan Oxy. Un moteur serait attribué à la composante x et un autre à la composante y. Dans une telle configuration, lors de l'initialisation du système, le contrôleur exécutera l'algorithme de cinématique directe afin de déterminer la position actuelle de l'outil. Par la suite, le contrôleur aura recours exclusivement à l'algorithme de cinématique inverse afin de spécifier la position de l'intersection des deux axes (position de l'outil).

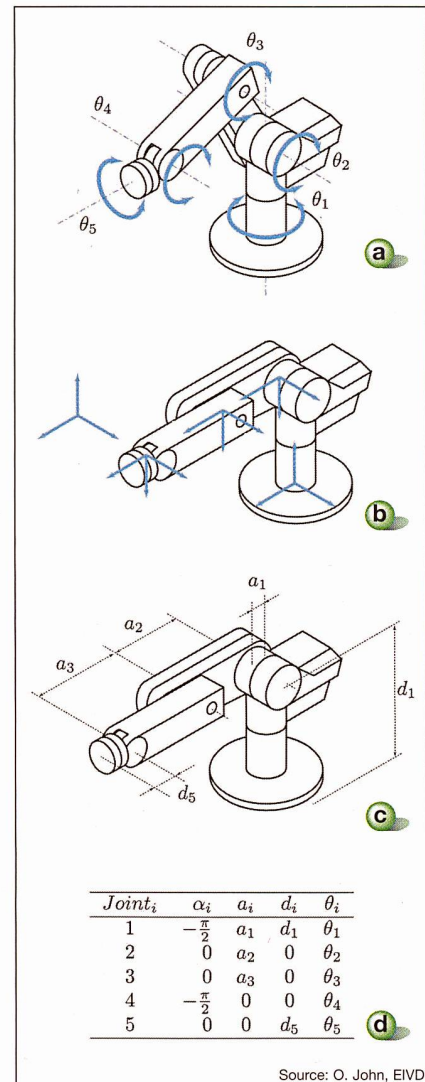
Le contrôleur d'axes

Un contrôleur d'axes (Motion Controller) est un système d'asservissement de moteurs ou plus généralement d'axes. Les quatre lettres formant l'abréviation PMAC correspondent à Programmable

Multi Axis Controller; un contrôleur Turbo PMAC est illustré en figure 2. Un contrôleur peut travailler en boucle fermée ou en boucle ouverte dans le cas d'un moteur pas-à-pas par exemple. Dans le cas d'un asservissement en boucle fermée, le contrôleur reçoit une consigne et à l'aide d'un retour d'informations (fermeture de boucle) il asservit l'axe. Le retour se fait généralement par le biais d'un capteur d'état du système (capteur de position).

L'information de fermeture de boucle est traitée par un régulateur, le plus généralement un régulateur proportionnel intégral dérivé (PID). A l'entrée d'un PID, la différence instantanée entre la consigne et la mesure donne l'erreur (figure 3). Ensuite le régulateur PID génère un signal de commande à partir du signal d'erreur précédemment évalué. Le signal de commande est traité par les trois opérations précitées. Dans les contrôleurs PMAC, chacun des 32 axes gérés simultanément a un régulateur PID dédié. Sans détailler chacune des composantes, le principe de régulation PID va permettre de stabiliser un moteur afin de générer les équations de mouvements. Les équations de mouvements correspondent aux graphes représentant le comportement du moteur en fonction du temps (accélération, vitesse et position). Les régulateurs PID assignés à chaque axe peuvent être paramétrés automatiquement à l'aide de logiciels dédiés aux contrôleurs, ceci afin d'éviter un comportement indésirable: vibrations des axes, réaction disproportionnée à une contrainte mécanique, arrêt brusque des moteurs, etc.

Une particularité importante des contrôleurs Turbo PMAC consiste en l'intégration au cœur de leur architecture d'une partie entièrement programmable. Cet espace est destiné à accueillir les algorithmes de cinématique pour les configurations spécifiques à l'application de l'utilisateur. Un atout majeur de cette configuration consiste en la possibilité d'avoir l'entier des fonctions à disposi-



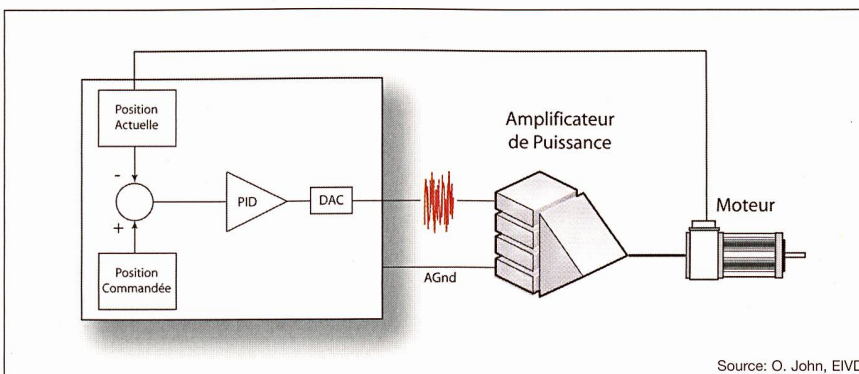
Source: O. John, EIVD

Figure 4 Les quatre étapes de la mise en équations utilisées

tion déjà implémentées dans le contrôleur tels que l'algorithme d'interpolation linéaire et circulaire, régulateur PID, lookahead, etc. La rapidité d'exécution de la cinématique est un autre avantage. Cet espace mémoire est divisé en deux et est disponible pour chaque système de coordonnées. La moitié de cet espace est attribué à la cinématique directe (Forward Sub-Routine) et l'autre à la cinématique inverse (Inverse Sub-Routine). L'essence même du travail en vue de développer un système dédié à une application spécifique consiste à implémenter les deux sous-routines.

Méthodologie suivie

Afin de pouvoir implémenter les algorithmes de cinématique au cœur d'un contrôleur, il convient en tout premier lieu de dériver les équations du système à asservir. Ceci n'est possible que si le système observé est décrit dans un forma-



Source: O. John, EIVD

Figure 3 Illustration de la régulation pour un moteur asservi

lisme mathématique. De ce fait, il convient de suivre les étapes suivantes:

- 1) mettre en équations le système contrôlé
- 2) calculer la matrice du bras (dans le contexte d'une chaîne articulée)
- 3) dériver les relations de la cinématique directe et inverse
- 4) implémenter ces équations dans le contrôleur Turbo PMAC.

Étape 1: mise en équation du système

La mise en équations de la configuration correspond à la représentation sous forme mathématique (matricielle) dudit système (un robot par exemple). Pour réaliser ceci, le formalisme de Denavit et Hartenberg (D-H) est appliqué. Ce dernier permet de simplifier le problème de la mise en équation avec toutefois certaines limitations dont il ne serait question dans ce contexte. Quatre étapes sont nécessaires à la traduction sous forme mathématique d'un système mécanique. La première opération admet la numérotation de chaque articulation en commençant par sa base, puis en remontant jusqu'à son extrémité (figure 4a). Lors de celle-ci, il est utile de relever la capacité de rotation de chaque articulation. La seconde opération consiste à assigner un système de coordonnées locales à chaque articulation en suivant le formalisme de D-H (figure 4b). Afin de conclure l'identification du système, la troisième étape

consiste à relever les paramètres relatifs à la configuration, c'est-à-dire les décalages dans les axes x et z (figure 4c). La dernière étape consiste à regrouper dans un tableau les paramètres identifiés lors de l'opération précédente (figure 4d).

A ce stade, il est possible de représenter la transformation d'une articulation à celle qui la suit ou inversement. En vue de résoudre la cinématique d'une chaîne articulée, il convient non pas d'exprimer une articulation en fonction de celle qui la suit, mais plutôt de la définir à partir de la base du système à son extrémité. Pour réaliser ceci, une post-multiplication des matrices homogènes de transformation est suffisante. Il est toutefois fortement conseillé d'avoir recours à un logiciel de mathématique afin d'éviter toute erreur de multiplication matricielle.

Formellement, les rotations sont décrites mathématiquement de façon différente des translations [2]. En effet les rotations donnent lieu à des multiplications matricielles, alors que les translations conduisent à des additions vectorielles. Lorsque plusieurs mouvements sont chaînés, cette approche conduit à des expressions extrêmement lourdes. De plus, il n'est pas commode de représenter un seul mouvement par deux opérations distinctes. C'est afin d'éviter ce type de problèmes que les coordonnées homogènes sont utilisées. Ces dernières permettent de représenter par une seule matrice un

mouvement composé à la fois de rotations et de translations. Si plusieurs mouvements sont chaînés, il suffit alors de multiplier les matrices correspondantes.

Par la suite, le terme «transformation» est employé pour désigner un mouvement quelconque (rotation, translation ou combinaison des deux). Les matrices homogènes de transformation D-H permettent formellement de décrire une transformation d'une articulation à une autre. Par exemple, la matrice de l'équation (1) décrit la transformation permettant de se rendre de l'articulation n à l'articulation n+1.

Étape 2: la matrice du bras

La matrice décrivant la position du préhenseur d'un robot par rapport à sa base est communément appelé la matrice du bras. Dans l'expression ci-dessus, l'indice n indique que l'on traite une chaîne articulée à n degrés de liberté; l'indice 0 indique que l'on exprime la position du n^{ème} repère dans le référentiel de base du système. L'équation (2) représente la post-multiplication des matrices articulaires (matrice homogène de transformation D-H) définissant la matrice du bras.

$${}^0A_n = {}^0A_1 {}^1A_2 \dots {}^{n-1}A_n \tag{2}$$

Étape 3: dériver les équations de cinématique

A l'aide de la matrice du bras, les équations de position peuvent être directement dérivées en isolant les éléments de la dernière colonne de ladite matrice. En ce qui concerne les équations d'orientation, une simple application de formule permet de les dériver.

$${}^{n+1}A_n = \begin{bmatrix} \cos(\theta_{n+1}) & -\cos(\alpha_{n+1}) \cdot \sin(\theta_{n+1}) & \sin(\alpha_{n+1}) \cdot \sin(\theta_{n+1}) & a_{n+1} \cdot \cos(\theta_{n+1}) \\ \sin(\theta_{n+1}) & \cos(\alpha_{n+1}) \cdot \cos(\theta_{n+1}) & -\sin(\alpha_{n+1}) \cdot \cos(\theta_{n+1}) & a_{n+1} \cdot \sin(\theta_{n+1}) \\ 0 & \sin(\alpha_{n+1}) & \cos(\alpha_{n+1}) & d_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Forward Kinematics Program (1)

```
; Definitions to variables with fixed functions
#define Mtr1KinPos    P1      ; #1 pos in cts for kinematics
#define XKinPos      Q7      ; X-axis pos in deg for kin

; Definitions to variables with open functions
#define Mtr1Scale    Q120    ; Counts per degree of crank
#define Rad          Q121    ; Radius of crank
#define Len          Q122    ; Length of crank arm
#define L2MinR2      Q123    ; Len^2-Rad^2
#define Theta        Q124    ; Intermediate angle term
#define CosTheta     Q125    ; Intermediate trig term
#define Dtheta       Q126    ; Velocity term

; Forward kinematics program buffer for repeated execution
IF (Mtr1Homed=1) ; Valid position reference?
  Theta=(Mtr1KinPos/Mtr1Scale)%-180
  CosTheta=COS(Theta)
  XKinPos=Rad*CosTheta+SQRT(Rad*Rad*CosTheta*CosTheta-L2MinR2)
  Dtheta=0 ; Starts at zero velocity
ENDIF
```


Inverse Kinematics Program (2)

Répété chaque 10 ms

```
; Additional definitions for inverse kinematics program
#define ThetaP      Q131      ; Possible positive soln
#define ThetaM      Q132      ; Possible negative soln
#define DThetaP     Q133      ; Velocity if positive soln
#define DThetaM     Q134      ; Velocity if negative soln
#define D2ThetaP    Q135      ; Accel if positive soln
#define D2ThetaM    Q136      ; Accel if negative soln

; Inverse kinematics program buffer for repeated execution
CosTheta=(XKinPos*XKinPos-L2MinR2)/(2*Rad*XKinPos)
IF(CosTheta!>1.0 AND CosTheta!<-1.0); Valid position?
  ThetaP=ACOS(CosTheta)      ; Tentative positive soln
  ThetaM=-ThetaP             ; Tentative negative soln
  DThetaP=(ThetaP-Theta)%-180 ; Velocity for positive soln
  DThetaM=(ThetaM-Theta)%-180 ; Velocity for negative soln
  D2ThetaP=DThetaP-DTheta    ; Accel for positive soln
  D2ThetaM=DThetaM-DTheta    ; Accel for negative soln
  IF (ABS(D2ThetaP)<ABS(D2ThetaM)) ; Use positive soln
    Theta=ThetaP             ; Select and save for next cycle
    DTheta=DThetaP          ; Select and save for next cycle
  ELSE
    Theta=ThetaM             ; Select and save for next cycle
    DTheta=DThetaM          ; Select and save for next cycle
  ENDIF
  MtrlKinPos=MtrlKinPos+DTheta*MtrlScale
ELSE ; Command out of range
  CS1RunTimeErr=1 ; Set to stop
ENDIF
```

Une approche analytique (parfois géométrique) doit cependant être employée lors de la résolution de la cinématique inverse. Même lors d'une résolution géométrique, la matrice du bras est aussi utilisée afin de mettre en évidence des identités trigonométriques. Elle se réduit en utilisant tour à tour les matrices inverses de chaque articulation en partant successivement de l'extrémité pour terminer à la base du système. De cette façon, il est possible d'exprimer les équations pour chaque angle articulaire.

Etape 4: implémentation de la cinématique dans le contrôleur

Le processus de cinématique inverse nécessite la conversion des coordonnées de base du système en coordonnées des articulations ou de moteurs. La cinématique inverse permet à l'utilisateur de la machine de programmer les mouvements de l'outil dans les coordonnées de l'atelier – usuellement des coordonnées cartésiennes et angulaires – laissant le contrôleur commander les positions requises des articulations.

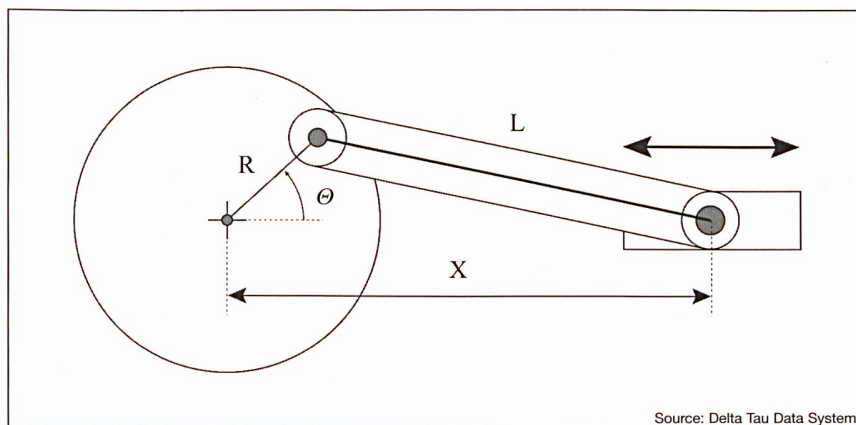
Spécifier le mouvement dans un système à l'aide de la cinématique inverse

est un processus décomposé en étapes multiples:

- Calcul des équations de mouvement dans les coordonnées de base (vitesse, accélération, décélération, etc.)
- Interpolation grossière du mouvement dans les coordonnées de base
- Conversion des points intermédiaires dans les coordonnées des articulations
- Interpolation fine du mouvement dans les coordonnées des articulations

e) Feedback / Feedforward basés sur les états commandés et actuels des articulations.

L'étape c) est l'étape décisive où se calcule la cinématique inverse et constitue le cœur de la problématique abordée. Delta Tau Data Systems propose quatre méthodes différentes d'application de la cinématique inverse faisant usage du contrôleur Turbo PMAC. Le travail décrit



Source: Delta Tau Data Systems

Figure 5 Illustration schématique de l'arbre à cames

est basé sur la méthode faisant usage des routines de cinématique.

Dans cette méthode, chaque système de coordonnées peut avoir une sous-routine de cinématique inverse dédiée et écrite par l'utilisateur dans un langage de haut niveau. Elle s'exécute automatiquement à chaque segment périodique (à la fréquence du servo-moteur de 9 kHz) convertissant les coordonnées d'outil dans les coordonnées des articulations.

Les routines du firmware calculent automatiquement le mouvement pour les coordonnées du système puis effectue l'interpolation grossière (segmentation) dans les coordonnées de l'outil. Lorsque l'option de cinématique inverse pour le système de coordonnées est activée, cette sous-routine utilisateur est appelée automatiquement. Elle calcule les coordonnées des articulations intermédiaires qui sont alors transmises à la routine d'interpolation fine du PMAC.

De plus, les points intermédiaires dans les coordonnées des articulations peuvent être filtrés à travers l'algorithme d'anticipation d'erreur du Turbo PMAC (Turbo PMAC's Lookahead algorithm). Ceci permet de détecter toute violation potentielle des limites de vitesse ou d'accélération des articulations, ainsi que l'anticipation de changements brusques de trajectoires.

En vue d'implémenter cette stratégie dans un Turbo PMAC, deux programmes spéciaux sont requis pour le système de coordonnées. Un pour s'affranchir de la cinématique directe (nécessaire quand le système exécute un programme afin d'assurer que le premier mouvement soit exécuté correctement), l'autre pour l'exécution de la cinématique inverse. Une fois ces programmes définis, ils sont automatiquement exécutés à la bonne vitesse par le Turbo PMAC.

Exemple d'application: Un arbre à cames

Le principe de l'arbre à cames [3] est utilisé dans beaucoup d'applications afin de transformer un mouvement de rotation en un mouvement linéaire réciproque. Les presses faisant usage d'arbres à cames commencent à avoir recours à des servo-moteurs afin d'offrir une plus grande flexibilité dans le cycle de pression qu'elles exercent. Les utilisateurs cherchent généralement à programmer ces mécanismes directement en terme de grandeurs linéaires et non pas en déplacements angulaires de la partie rotative de l'arbre à cames. Les algorithmes de cinématique des contrôleurs Turbo PMAC permettent d'implémenter cette méthode

de programmation en calculant automatiquement la transformation entre la position linéaire du préhenseur et la position angulaire de l'arbre à cames.

La géométrie dans ce mécanisme est relativement évidente et ne nécessite pas de représentation matricielle. De ce fait, la méthode de dérivation des équations de cinématique vue jusqu'ici ne sera pas utilisée ici. Toutefois, il sera essentiellement question d'illustrer une implémentation concrète d'un système dans le langage PMAC. La difficulté de l'algorithme de cinématique dans cet exemple réside dans le choix entre les deux angles possibles de l'arbre à cames pour une position donnée du préhenseur. Il existe plusieurs méthodes possibles pour prendre cette décision; dans cet exemple, le choix consiste à minimiser l'accélération exercée sur le volant de l'arbre à cames.

Description du mécanisme

Le mécanisme choisi (figure 5) comprend un bras de longueur L lié par une articulation pivotante sur le préhenseur contraint de se déplacer dans un seul axe horizontal. A l'autre extrémité du bras se trouve l'articulation reliant ce dernier au volant de l'arbre à cames à une distance de R du centre du volant. Le volant de l'arbre à cames est mû par un servo-moteur.

La cinématique directe

En cinématique directe, la position linéaire X est recherchée en fonction de l'angle Θ formé entre la base et l'intersection du volant et du bras L. A l'aide de la loi du cosinus et en observant le triangle formé par les longueurs R, L et X, l'équation (3) est aisément posée. L'équation (3) sous une forme quadratique en X. Une fois cette dernière résolue, il en résulte l'expression de la cinématique directe donnée en (5). Du fait des contraintes physiques, le bras ne peut se déplacer que vers la droite selon la figure 5; de ce fait, seule la solution positive est valide. La définition des variables d'environnement ainsi que l'implémentation de la cinématique directe sont représentées dans l'encadré (1). Ce programme sera exécuté automatiquement à chaque fois que le système commencera un nouveau mouvement.

$$L^2 = X^2 + R^2 - 2RX \cos \theta \quad (3)$$

$$X^2 - 2RX \cos \theta + R^2 - L^2 = 0 \quad (4)$$

$$X = R \cos \theta \pm \sqrt{R^2 \cos^2 \theta - R^2 + L^2} \quad (5)$$

La cinématique inverse

En cinématique inverse, l'angle Θ est recherché en fonction de la position linéaire X (problème inverse). L'équation du cosinus appliquée pour la cinématique directe est réarrangée et il en résulte l'équation (6). De cette équation, l'angle Θ est isolé afin de donner l'équation de cinématique inverse (7). Ici, les deux solutions de l'arc-cosinus sont potentiellement valides et il convient de définir une méthode permettant de résoudre le problème comme illustré dans l'encadré (2).

$$2RX \cos \theta = X^2 + R^2 - L^2 \quad (6)$$

$$\theta = \pm \cos^{-1} \left[\frac{X^2 + R^2 - L^2}{2RX} \right] \quad (7)$$

La méthode développée lors de ce travail présente d'une part, l'avantage d'offrir une approche systématique à l'étude de la cinématique directe et inverse des chaînes articulées. D'autre part, les équations mathématiques dérivées par celle-ci sont directement exploitables. En effet, le contrôleur Turbo PMAC peut accueillir les équations de cinématique telles que dérivées par la méthode, ce qui simplifie grandement leur implémentation et réalise ainsi un gain considérable de temps et d'efforts.

Références

- [1] Othmar John: Forward and Inverse Kinematics Programs For Robots, In Delta Tau's PMAC Motion Language, Travail de diplôme EIVD, Yverdon-les-Bains, 2004
- [2] Jean-Daniel Dessimoz: Commande des robots et des systèmes automatisés, Robotique et Automatisation, EIVD, Yverdon-les-Bains, 2003
- [3] Curt Wilson: Turbo PMAC Program Examples, Crank Arm Kinematics Example, Delta Tau Data Systems, Inc., Chatsworth (CA) USA, 2004

Informations sur les auteurs

Othmar John est Ing. Dipl. HES de l'Ecole d'ingénieurs du Canton de Vaud (EIVD), filière Informatique Technique.

Haute Ecole Spécialisée de la Suisse Occidentale (HES-SO), EIVD, 1401 Yverdon-les-Bains, ojohn@crog.org

François Gaille, Dr. ès sciences en physique, est professeur à l'EIVD.

Haute Ecole Spécialisée de la Suisse Occidentale (HES-SO), EIVD, 1401 Yverdon-les-Bains, francois.gaille@eivd.ch

¹ PMAC est un produit de l'entreprise Delta Tau Data Systems, Inc. Cette entreprise est spécialisée dans les contrôleurs de mouvements multi-axes. La maison mère est basée aux Etats-Unis, à Chatsworth dans la banlieue de Los Angeles (CA). Elle est représentée en Europe au travers de sa filiale suisse basée à Neuhausen. www.deldata.com

Cet article est basé sur un projet de diplôme présenté à la Journée Technique JT'05 de l'EIVD.