

La cinématique dans EMC2

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Introduction	1
1.1	Les jointures (articulations) par rapport aux axes	1
2	Cinématiques triviales	2
3	Cinématiques non triviales	3
3.1	Transformation avant	3
3.2	Transformation inverse	4
4	Détails d'implémentation	5
5	Index	6

Chapter 1

Introduction

Habituellement quand nous parlons de machines CNC, nous pensons à des machines programmées pour effectuer certains mouvements et effectuer diverses tâches. Pour avoir une représentation unifiée dans l'espace de ces machines, nous la faisons correspondre à la vision humaine de l'espace en 3D, la plupart des machines (sinon toutes) utilisent un système de coordonnées courant, le système Cartésien.

Le système de coordonnées Cartésiennes est composé de 3 axes (X, Y, Z) chacun perpendiculaire aux 2 autres.¹

Quand nous parlons d'un programme G-code (RS274NGC) nous parlons d'un certain nombre de commandes (G0, G1, etc.) qui ont comme paramètres (X- Y- Z-). Ces positions se réfèrent exactement à des positions Cartésiennes. Une partie du contrôleur de mouvements d'EMC2 est responsable de la translation entre ces positions et les positions correspondantes de la cinématique de la machine².

1.1 Les jointures (articulations) par rapport aux axes

Une jointure, pour une machine CNC est un des degrés physiques de liberté de la machine. Elle peut être linéaire (vis à billes) ou rotative (table tournante, jointures d'un bras robotisé). Il peut y avoir n'importe quel nombre de jointures sur une machine. Par exemple, un robot classique dispose de 6 jointures et une fraiseuse classique n'en a que 3.

Sur certaines machines, les jointures sont placées de manière à correspondre aux axes cinématiques (jointure 0 le long de l'axe X, jointure 1 le long de l'axe Y et jointure 2 le long de l'axe Z), ces machines sont appelées machines Cartésiennes (ou encore machines à cinématiques triviales). Ce sont les machines les plus courantes parmi les machines-outils mais elles ne sont pas courantes dans d'autres domaines comme les machines de soudage (ex: robots de soudage de type puma).

¹Le mot "axes" est aussi communément (et incorrectement) utilisé à propos des machines CNC, il fait référence aux directions des mouvements de la machine.

²Cinématique: une fonction à deux voies pour transformer un espace Cartésien en espace à jointures

Chapter 2

Cinématiques triviales

Comme nous l'avons vu, il y a un groupe de machines sur lesquelles chacun des axes est placé le long d'un des axes Cartésiens. Sur ces machines le passage, du plan de l'espace Cartésien (le programme G-code) au plan de l'espace jointure (l'actuateur actuel de la machine), est trivial. C'est un simple plan 1:1:

```
pos->tran.x = joints[0];
```

Dans l'extrait de code ci-dessus, nous pouvons voir comment le plan est fait: la position X est identique avec la jointure 0, Y avec la jointure 1 etc. Nous nous référons dans ce cas à une cinématique directe (une transformation avant), tandis que dans l'extrait de code suivant il est fait référence à une cinématique inverse (ou une transformation inverse):

```
joints[0] = pos->tran.x;
```

Comme on peut le voir, c'est assez simple de faire la transformation d'une machine à cinématique banale (ou Cartésienne). Cela devient un peu plus compliqué si il manque un axe à la machine.¹²

¹Si la machine (par exemple un tour) est montée avec seulement les axes X, Z et A et que le fichier d'init d'EMC2 contient uniquement la définition de ces 3 jointures, alors l'assertion précédente est fausse. Parce-que nous avons actuellement (joint0=x, joint1=Z, joint2=A) ce qui suppose que joint1=Y. Pour faire en sorte que cela fonctionne dans EMC2 il suffit de définir tous les axes (XYZA), EMC2 utilisera alors une simple boucle dans HAL pour l'axe Y inutilisé.

²Une autre façon de le faire fonctionner, est de changer le code correspondant et recompiler le logiciel.

Chapter 3

Cinématiques non triviales

Il peut y avoir un certain nombre de types de configurations de machine (robots: puma, scara; hexapodes etc.) Chacun d'eux est mis en place en utilisant des jointures linéaires et rotatives. Ces jointures ne correspondent pas habituellement avec les coordonnées Cartésiennes, cela nécessite une fonction cinématique qui fasse la conversion (en fait 2 fonctions: fonction en avant et inverse de la cinématique).

Pour illustrer ce qui précède, nous analyserons une simple cinématique appelée bipode (une version simplifiée du tripode, qui est déjà une version simplifiée de l'hexapode).

bipod.png

Figure 3.1: Définir un bipode

Le bipode dont nous parlons est un appareil, composé de deux moteurs placés sur un mur, à cet appareil un mobile est suspendu par des fils. Les jointures dans ce cas sont les distances entre le mobile et les moteurs de l'appareil (nommées AD et BD sur la figure [\[cap:Bipod-setup\]](#)).

La position des moteurs est fixée par convention. Le moteur A est en (0,0), qui signifie que sa coordonnée X est 0 et sa coordonnée Y également 0. Le moteur B est placé en (Bx, 0), ce qui veut dire que sa coordonnée X est Bx.

Notre pointe mobile se trouvera au point D défini par les distances AD et BD, et par les coordonnées Cartésiennes Dx, Dy.

La tâche de la cinématique consistera à transformer les longueurs des jointures en (AD, BD) en coordonnées Cartésiennes (Dx, Dy) et vice-versa.

3.1 Transformation avant

Pour effectuer la transformation de l'espace jointure en espace Cartésien nous allons utiliser quelques règles de trigonométrie (le triangle rectangle déterminé par les points (0,0), (Dx,0), (Dx,Dy) et le triangle rectangle (Dx,0), (Bx,0) et (Dx,Dy).

Nous pouvons voir aisément que $AD^2 = x^2 + y^2$, de même que $BD^2 = (Bx - x)^2 + y^2$.

Si nous soustrayons l'un de l'autre nous aurons:

$$[AD^2 - BD^2 = x^2 + y^2 - x^2 + 2 * x * Bx - Bx^2 - y^2]$$

et par conséquent:

$$[x = \frac{AD^2 - BD^2 + Bx^2}{2 * Bx}]$$

De là nous calculons:

$$[y = \sqrt{AD^2 - x^2}]$$

Noter que le calcul inclut la racine carrée de la différence, mais qu'il n'en résulte pas un nombre réel. Si il n'y a aucune coordonnées Cartésienne pour la position de cette jointure, alors la position est dite singulière. Dans ce cas, la cinématique inverse retourne -1.

Traduction en code:

```
double AD2 = joints[0] * joints[0];  
return 0;
```

3.2 Transformation inverse

La cinématique inverse est beaucoup plus simple dans notre exemple, de sorte que nous pouvons l'écrire directement:

$$[AD = \sqrt{x^2 + y^2}]$$

$$[BD = \sqrt{(Bx - x)^2 + y^2}]$$

ou traduite en code:

```
double x2 = pos->tran.x * pos->tran.x;
```

Chapter 4

Détails d'implémentation

Un module cinématique est implémenté comme un composant de HAL, et il est permis d'exporter ses pins et ses paramètres. Il consiste en quelques fonctions "C" (par opposition aux fonctions de HAL):

- `` int kinematicsForward(const double *joint, EmcPose *world, const KINEMATICS_FORWARD_FLAGS *fflags, KINEMATICS_INVERSE_FLAGS *iflags)`` Implémente la fonction cinématique avant, comme décrite dans la section [\[sub:Forward-transformation\]](#).
- `` int kinematicsInverse(const EmcPose * world, double *joints, const KINEMATICS_INVERSE_FLAGS *iflags, KINEMATICS_FORWARD_FLAGS *fflags)`` Implémente la fonction cinématique inverse, comme décrite dans la section [\[sub:Inverse-transformation\]](#).
- `KINEMATICS_TYPE kinematicsType (void)` Retourne l'identificateur de type de la cinématique, typiquement `KINEMATICS`.
- `` int kinematicsHome(EmcPose *world, double *joint, KINEMATICS_FORWARD_FLAGS *fflags, KINEMATICS_INVERSE_FLAGS *iflags)`` La fonction prise d'origine de la cinématique ajuste tous ses arguments à leur propre valeur à une position d'origine connue. Quand elle est appelée, cette position doit être ajustée, quand elle est connue, comme valeurs initiales, par exemple depuis un fichier INI. Si la prise d'origine de la cinématique peut accepter des points de départ arbitraires, ces valeurs initiales doivent être utilisées.
- `int rtapi_app_main(void)`
- `void rtapi_app_exit(void)` Il s'agit des fonctions standards d'installation et de la désinstallation des modules RTAPI.

Quand ils sont contenus dans un seul fichier source, les modules de la cinématique peuvent être compilés et installés par `«comp`»`. Voir la manpage `«comp(1)»` pour d'autres informations.

Chapter 5

Index

C

cinématique, [1](#)
Cinématique triviale), [1](#)

M

Machines Cartésiennes, [1](#)
Machines CNC, [1](#)
