Etalonnage de robots par vision

Nicolas Andreff

Novembre 2012

Fondation unit
Université Numérique Ingénierie et Technologie

GDR ROBOTIQUE

Institut Français de Mécanique Avancée

U.V. Étalonnage et Identification des Systèmes

TP Étalonnage Géométrique

Nicolas Andreff

Année 2004–2005

1 Position du problème

Dans ce TP, nous allons chercher à étalonner le robot Motoman.

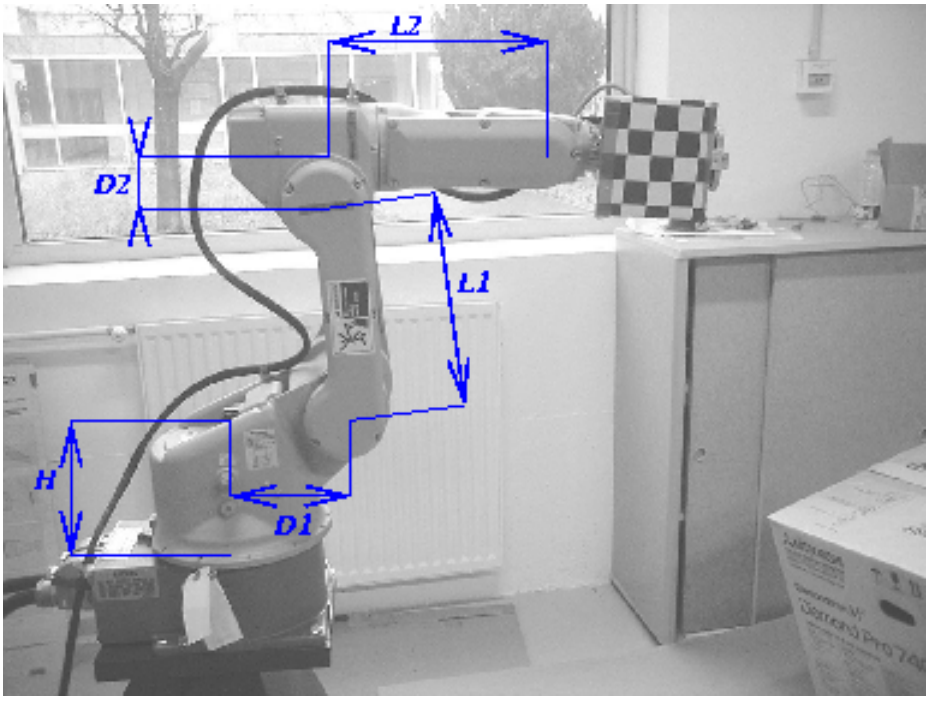

FIGURE 1 – Le robot Motoman et ses données constructeur.

Les données constructeur de ce robot sont (voir Figure 1) :

$$\begin{aligned} H &= 200mm \\ D_1 &= 150mm \\ L_1 &= 260mm \\ D_2 &= 70mm \\ L_2 &= 260mm \end{aligned}$$

Le logiciel SoftEIS permet de commander le robot Motoman en position moteur $\mathbf{q} = (q_1, q_2, q_3, q_4, q_5, q_6)$. A chaque position moteur $\mathbf{q}$ correspond, du fait de la géométrie du robot, la pose (position/orientation) ${}^0T_6$ du repère terminal du robot (R_6) par rapport à son repère de base (R_0). Le repère terminal du robot est situé au centre du poignet.

Cette pose du robot n'est pas mesurable directement. Pourtant, les tâches robotiques sont le plus souvent exprimées dans l'espace Cartésien, c'est-à-dire en terme de poses du robot. Aussi, pour pouvoir faire une commande Cartésienne, est-il nécessaire de pouvoir estimer la pose. Pour cela, il faut écrire un modèle géométrique direct, représentant aussi fidèlement que possible, la pose du robot en fonction de la configuration moteur $\mathbf{q}$ et de la géométrie du robot.

1.1 Première modélisation

Donner l'expression du modèle géométrique direct de ce robot (formalisme de Khalil-Kleinfinger) en fonction des des opérateurs rotation et translation (**Rot(axe,valeur)** et **Trans(axe,valeur)**).

Ecrire la fonction Matlab

$$[T] = MyMGD(\theta, H, D_1, L_1, D_2, L_2)$$

qui fournit la matrice homogène contenant la position et l'orientation du repère terminal en fonction des positions articulaires θ.

Vérifier la validité de votre fonction par rapport au robot en visualisant la position et l'orientation du repère terminal via la fonction **afficheRepere(Matrice,taille)**.

Remarque : Aidez-vous des opérateurs **Rot(axe,valeur)** et **Trans(axe,valeur)** déjà programmés.

1.2 Prise en compte des caractéristiques géométriques des moteurs

Êtes-vous d'accord avec la relation suivante ?

$$\theta(i) = K_i q(i) + q_0(i) \tag{1}$$

avec K_i, rapport de réduction et q_0, décalage angulaire ?

Pouvez-vous donner la valeur de ces paramètres ?

1.3 Importance des données géométriques

Mettre en évidence le besoin d'estimer finement les longueurs H, D_i et L_i, $\forall i = 1, 2$.

2 Modélisation

2.1 Écriture du modèle géométrique sans dispositif de mesure

Écrire l'expression mathématique du modèle géométrique direct fourni par la CAO et corrigé des décalages articulaires et inverses des rapports de réduction.

Rappel :

$$^{i-1}T_i = R(x,\alpha_i)T(x,d_i)R(z,\theta_i)T(z,r_i) \quad (2)$$

et par convention $\alpha_1 = d_1 = 0$

2.2 Dispositif de mesure

Ce modèle n'est pas utilisable tel quel. En effet, il faudrait pouvoir mesurer $^0T_6$ ce qui est impossible directement. Par conséquent, nous mettons en place un système de mesure (ici, caméra et mire, cf étalonnage caméra)

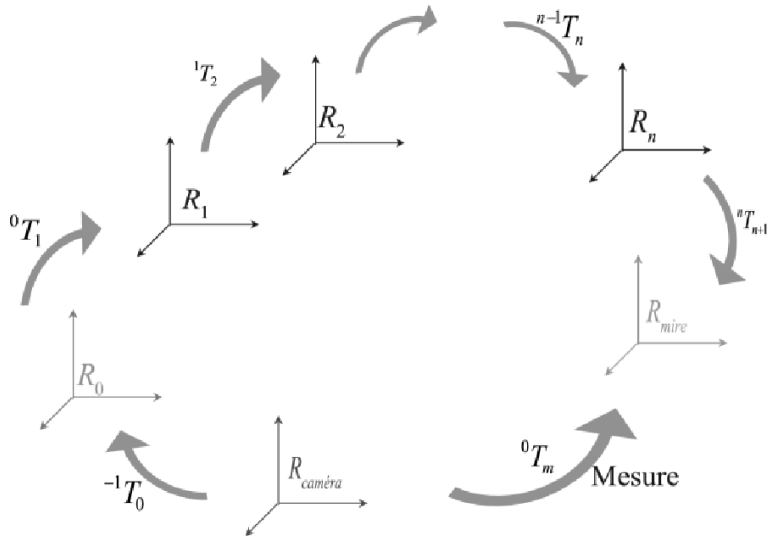

FIGURE 2 – Intégration du dispositif de mesure dans la chaîne cinématique d'un robot à n articulations.

Pour un robot à n articulations, on peut modéliser, de manière générique, $^{-1}T_0$ (respectivement $^nT_{n+1}$) par 6 paramètres indépendants :

$$\left\{ \begin{array}{rcl} ^{-1}T_0 & = & R(z,\gamma_c)T(z,b_c)R(x,\alpha_c)T(x,d_c)R(z,\theta_c)T(z,r_c) \\ ^nT_{n+1} & = & R(z,\gamma_m)T(z,b_m)R(x,\alpha_m)T(x,d_m)R(z,\theta_m)T(z,r_m) \end{array} \right. \quad (3)$$

Montrer qu'on peut alors se ramener à la forme canonique :

$$\begin{array}{rcl} ^{-1}T_1 & = & R(x,\alpha_0)T(x,d_0)R(z,\theta_0)T(x,r_0)R(x,\alpha'_1)T(x,d'_1)R(z,\theta'_1)T(z,r_c) \\ ^{n-1}T_{n+1} & = & R(x,\alpha'_n)T(x,d'_n)R(z,\theta'_n)T(x,r'_n)R(x,\alpha_{n+1})T(x,d_{n+1})R(z,\theta_{n+1})T(z,r_{n+1}) \end{array} \quad (4)$$

avec

$$\begin{array}{cccccccccccc} \alpha_0 & = & 0 & \alpha'_1 & = & \alpha_c & \alpha'_n & = & \alpha_n & \alpha_{n+1} & = & \alpha_m \\ d_0 & = & 0 & d'_1 & = & d_c & d'_n & = & d_n & d_{n+1} & = & d_m \\ \theta_0 & = & \gamma_c & \theta'_1 & = & \theta_1 + \theta_c & \theta'_n & = & \theta_n + \gamma_m & \theta_{n+1} & = & \theta_m \\ r_0 & = & b_c & r'_1 & = & r_1 + r_c & r'_n & = & r_n + b_m & r_{n+1} & = & r_m \end{array} \quad (5)$$

Par conséquent, nous ne pourrons identifier séparément les couples : (θ_1, θ_c), (r_1, r_c), (θ_n, γ_n) et (r_n, b_m).

2.3 Paramètres du modèle

Dresser la liste des paramètres à identifier (ξ) et en donner une première estimation.

N.B. : Pour obtenir une estimation des paramètres de ${}^{-1}T_0$ (paramètres caméra) et ${}^{6}T_7$ (paramètres mire), il faut aller faire les mesures sur le robot !

3 Algorithme d'identification sans les paramètres moteurs

Dans un premier temps, nous allons fixer les paramètres moteurs (rapport de réduction à 1 et décalage angulaire à 0), afin de diminuer le nombre de paramètres à identifier. Dans cette partie, ces paramètres sont donc fixés et n'apparaissent pas dans le vecteur ξ.

3.1 Matrice jacobienne

Notons $X_r(j) = (R_r^j, t_r^j)$ la j^{ieme} mesure de ${}^cT_m$ ($={}^{-1}T_7$) et $X_m(q_j, \xi) = (R_m^j, t_m^j)$ l'estimation de ${}^cT_m$ obtenue à partir du modèle construit ci-avant, des paramètres géométriques ξ et des positions moteurs (q_j) associées à la j^{ieme} mesure.

Notons ΔX_j l'erreur entre $X_r(j)$ et $X_m(q_j)$, calculée numériquement par la fonction **erreurTR(T_1,T_2)**. Comme vu en cours, nous pouvons alors écrire :

$$\Delta X_j = \psi_j \Delta\xi \text{ où } \psi_j = \psi(q_j, \xi) \text{ est la matrice jacobienne du MGD } (\tfrac{\partial X}{\partial \xi})$$

Sachant que :

$$\psi_{\alpha_i} = \begin{bmatrix} x_{i-1} \wedge L_{i-1} \\ x_{i-1} \end{bmatrix},\ \psi_{d_i} = \begin{bmatrix} x_{i-1} \\ 0_{3x1} \end{bmatrix},\ \psi_{\theta_i} = \begin{bmatrix} z_i \wedge L_i \\ z_i \end{bmatrix},\ \psi_{r_i} = \begin{bmatrix} z_i \\ 0_{3x1} \end{bmatrix},\ \psi_{K_i} = q(i)\psi_{\theta_i},\ \psi_{q_{0i}} = \psi_{\theta_i}$$

écrire l'expression par bloc de ψ_j.

Les formules concernant ψ_{K_i} et $\psi_{q_{0i}}$ ne seront utilisées que dans le chapitre suivant.

On rappelle que x_{i-1} est obtenu par la matrice homogène ${}^cT_{i-1}$, z_i par ${}^cT_i$ et que L_i est le vecteur d'origine le centre du repère (R_i) et d'extrémité le centre du repère (R_{n+1}).

3.2 Fonction Repères

Écrire la fonction $[Reperes] = MGD(q, \xi)$ où $[Reperes] = \begin{bmatrix} {}^cT_0 \\ {}^cT_{1'} \\ . \\ . \\ {}^cT_m \end{bmatrix}$

3.3 Fonction Regresseur

Utiliser la fonction précédente afin de construire la fonction regresseur ($\psi = Regresseur(q, \xi)$), permettant d'obtenir la matrice jacobienne $[\psi]$.

3.4 Écriture du système d'identification

Écrire la fonction [A,b]=systeme(LesQ,LesX,ξ) qui permet de construire le système suivant :

$$Ax = b \tag{6}$$

avec $A = \begin{bmatrix} \psi_1 \\ . \\ . \\ \psi_e \end{bmatrix}$, $b = \begin{bmatrix} \Delta X_1 \\ . \\ . \\ \Delta X_e \end{bmatrix}$, $x = \Delta\xi$, e : nombre de mesures, $LesQ = \begin{bmatrix} q_1 \\ . \\ . \\ q_e \end{bmatrix}$, et $LesX = \begin{bmatrix} X_r(1) \\ . \\ . \\ X_r(e) \end{bmatrix}$

3.5 Fonction identification

Écrire la fonction $[\xi, nbIter, erreur] = identifier(LesQ, LesX, \xi_0)$ qui réalise l'identification à partir des fonctions précédentes.

N.B. : Pour résoudre le système (6), on pourra utiliser :

$$x = A\backslash b \text{ ou } x = pseudoinv(A) * b$$

3.6 Acquisition de données

Le logiciel SoftEIS permet d'enregistrer simultanément la configuration codeur courante (fichiers **.art**) et l'image courante (fichiers **.gif**).

La boîte à outils TOOLBOX_CALIB permet d'estimer la pose ${}^cT_m$ de la mire par rapport à la caméra, à partir d'une image et des paramètres intrinsèques de la caméra.

N.B. : Pour créer $LesQ$ et $LesX$:

. Initialisation :

$$LesQ = [] \quad (7)$$
$$LesX = [] \quad (8)$$

. A chaque nouvelle mesure :

$$LesQ = [LesQ; q] \quad (9)$$
$$LesX = [LesX; X] \quad (10)$$

où q aura été lu dans le fichier **.art** approprié (fonction **dlmread** sous matlab) et X aura été calculé par vision.

3.7 Identification

Procéder à l'identification des paramètres ξ. Donner et interpréter les résultats obtenus. Afin d'apprécier la précision de la Jacobienne, demander le conditionnement du regresseur (matrice A) calculée à chaque itération (fonction Matlab **cond(A)**).

Comparer la position et l'orientation du repère terminal avec le nouveau vecteur ξ à la position réelle via les fonctions **repere** et **afficheRepere**, ainsi que par observation directe sur le robot.

3.8 Amélioration de la méthode numérique

La valeur du conditionnement de la matrice jacobienne reste loin de 1. Ceci est lié aux différences d'ordre de grandeur entre les paramètres. En effet les longueurs (en millimètres) sont de l'ordre de 100 et les angles (en radians) de l'ordre de l'unité. Une telle différence perturbe l'algorithme.

Que peut-on envisager pour améliorer l'algorithme ? Dans la suite du TP nous exprimerons les valeurs des longueurs en mètre et les angles en radians, afin de travailler avec le même ordre de grandeur pour tous les paramètres. Que doit-on changer dans les programmes MatLab ?

Demander les nouveaux fichiers à un encadrant attentionné !

Recommencer la procédure d'identification. Qu'en concluez-vous ?

4 Algorithme d'identification complet

Reprenez la méthode précédente en considérant les rapports de réduction et les décalages angulaires comme paramètres du vecteur ξ. Combien de paramètres sont maintenant à identifier ?

1 Écriture de la jacobienne
2 Écriture de la fonction **Repere2**.
3 Écriture de la fonction **Regresseur2**.
4 Écriture du système d'identification.
5 Écriture de la fonction d'identification.
6 Identification (avec les mêmes valeurs puis augmenter le nombre de mesures)
7 Validation