Transformation de coordonnées entre repère objet et repère robot

2026-04-02

Résumé

Ce rapport traite de la problématique de conversion de coordonnées entre un repère local associé à un objet et le repère global utilisé par un système robotique. Pour établir cette correspondance, une méthode d'estimation de la transformation affine est présentée. Elle s'appuie sur un ensemble de points homologues mesurés dans les deux repères et sur une résolution par moindres carrés.

Le document aborde également une procédure permettant de convertir une normale en une orientation exploitable par le robot, afin d'assurer une manipulation cohérente de l'objet dans l'espace.

Table des matières

1. Introduction

Dans de nombreuses applications, les coordonnées d'un objet sont exprimées dans un repère propre à cet objet. Cependant, les mouvements du robot sont définis dans un repère global différent.

Afin de permettre l'interaction entre ces deux systèmes, il est nécessaire de déterminer une transformation permettant de convertir les coordonnées d'un repère vers l'autre.

Cette transformation permet par exemple :

- d'exécuter des trajectoires définies dans le repère de l'objet
- d'utiliser des données provenant de capteurs ou de modèles 3D
- d'adapter automatiquement la position d'un objet dans l'espace

2. Principe général

La transformation entre deux repères tridimensionnels peut être décrite par une transformation affine :

$$T = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$$

où :

- R est la matrice de rotation
- t est le vecteur de translation

Cette matrice permet de convertir un point p_A exprimé dans le repère objet en un point p_B dans le repère robot :

$$p_B = T p_A$$

3. Collecte des points

Pour estimer la transformation, il est nécessaire de disposer d'un ensemble de points correspondants dans les deux repères.

Chaque point est mesuré :

- dans le repère objet
- dans le repère robot

Un minimum de trois points non colinéaires est nécessaire pour déterminer une transformation.

Cependant, l'utilisation d'un nombre plus élevé de points permet d'obtenir une solution plus robuste en utilisant une résolution par moindres carrés.

Les données collectées sont donc de la forme :

$$A_i = (x_i, y_i, z_i)$$

$$B_i = (x_{i'}, y_{i'}, z_{i'})$$

4. Calcul de la transformation

La transformation recherchée doit satisfaire :

$$B = RA + t$$

Afin de résoudre ce problème, on construit une matrice :

$$P = \begin{pmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

Le problème devient alors :

$$PX = B$$

où :

$$X = \begin{pmatrix} R^T \\ t \end{pmatrix}$$

La solution est obtenue par moindres carrés :

$$X = \left(P^T P\right)^{\{-1\}} P^T B$$

La matrice de transformation finale est ensuite reconstruite sous la forme :

$$T = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$$

5. Transformation des normales

Lorsqu'un point possède également une normale de surface, celle-ci doit être transformée correctement.

Contrairement aux positions, les normales ne doivent pas être transformées directement par la matrice R.

La transformation correcte est :

$$n' = \left(R^{\{-1\}}\right)^T n$$

Cette opération correspond à la transformation des vecteurs normaux sous une transformation affine.

6. Conversion des normales en orientation

Dans certaines applications, la normale d'une surface doit être convertie en orientation utilisable par le robot.

6.1. Conversion d'une normale en angles d'Euler

Soit la normale unitaire $\mathrm{n} = (\mathrm{n\_x},\ \mathrm{n\_y},\ \mathrm{n\_z})$ avec $||\mathrm{n}|| = 1$.

On cherche des angles d'Euler $(\alpha,\ \beta,\ \gamma)$ dans la convention Yaw–Pitch–Roll (Z, Y, X) tels que :

$$R = R_{z(\alpha)} R_{y(\beta)} R_{x(\gamma)}$$

avec par exemple :

$$R_{z(\alpha)} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R_{y(\beta)} = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix}$$

$$R_{x(\gamma)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{pmatrix}$$

On aligne l'axe canonique $\mathrm{e\_z} = (0,\ 0,\ 1)$ avec n :

$$R_{z(\alpha)} R_{y(\beta)} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \sin(\beta)\cos(\alpha) \\ \sin(\beta)\sin(\alpha) \\ \cos(\beta) \end{pmatrix}$$

En identifiant ce vecteur à n, on obtient :

$$n_x = \sin(\beta)\cos(\alpha), \quad n_y = \sin(\beta)\sin(\alpha), \quad n_z = \cos(\beta)$$

D'où :

$$\beta = \arctan 2\Big(\sqrt{n_x^2 + n_y^2}, n_z\Big)$$

$$\alpha = \arctan 2\big(n_y, n_x\big)$$

Le troisième angle γ (roll) représente une rotation autour de la normale elle-même. On peut choisir :

$$\gamma = 0$$

La rotation minimale qui aligne e_z avec n est alors :

$$R = R_{z(\alpha)} R_{y(\beta)}$$

6.2. Conversion d'une normale en vecteur de rotation

Une autre représentation des orientations consiste à utiliser un vecteur de rotation.

Une rotation est décrite par :

- un axe de rotation unitaire k
- un angle de rotation θ

Le vecteur de rotation est :

$$r = \theta k$$

Pour aligner l'axe Z avec la normale n = (x, y, z) :

$$\theta = \arccos(z)$$

L'axe de rotation est le produit vectoriel entre Z et n :

$$Z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$k = Z \times n$$

soit :

$$k = \begin{pmatrix} -y \\ x \\ 0 \end{pmatrix}$$

Après normalisation :

$$k = \frac{1}{\sqrt{x^2 + y^2}} \cdot \begin{pmatrix} -y \\ x \\ 0 \end{pmatrix}$$

Le vecteur de rotation final est :

$$r = \theta \cdot \begin{pmatrix} -\frac{y}{\sqrt{x^2+y^2}} \\ \frac{x}{\sqrt{x^2+y^2}} \\ 0 \end{pmatrix}$$

6.3. Comparaison des méthodes

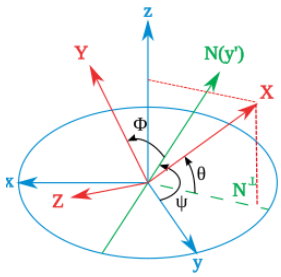

Fig. 1. – Angle d'Euler

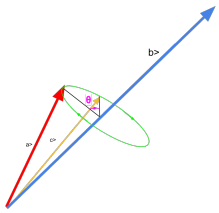

Fig. 2. – Vecteur de rotation

Une normale définit uniquement une direction dans l'espace et ne fournit qu'une contrainte partielle sur l'orientation.

La conversion en angles d'Euler nécessite trois rotations indépendantes. Dans l'approche précédente, la rotation autour de l'axe Z est fixée arbitrairement :

$$r_z = 0$$

La représentation par vecteur de rotation :

- décrit une rotation minimale entre deux directions
- évite des hypothèses arbitraires
- correspond à la représentation utilisée par le robot

7. Implémentation

Algorithme :

1. Mesurer les points de référence dans l'espace de l'objet et dans l'espace du robot
2. Calculer la matrice de transformation pour passer d'un espace à l'autre
3. Transformer les coordonnées de l'espace de l'objet dans celui du robot
4. Convertir la normale en vecteur de rotation

8. Conclusion

La transformation entre repère objet et repère robot constitue une étape fondamentale pour l'intégration de données provenant de différentes sources.

La méthode présentée permet d'estimer efficacement cette transformation à partir d'un ensemble de correspondances de points.

L'utilisation des vecteurs de rotation pour représenter l'orientation offre une solution robuste et compatible avec les systèmes robotiques modernes.

A Annexe

A.1 Code Python

A.1.1 Calcul de la matrice de transformation

```
def create_transformation(A, B):
    P = np.hstack([A, np.ones((N, 1))])
    X, _, _, _ = np.linalg.lstsq(P, B, rcond=None)

    R = X[:3, :].T
    t = X[3, :]

    T = np.eye(4)
    T[:3, :3] = R
    T[:3, 3] = t

    return T
```

A.1.2 Conversion normale en vecteur de rotation

```
def normalize(v):
    v = np.asarray(v, dtype=float)
    n = np.linalg.norm(v)
    return v / n if n > 0 else v
```

```
def normal_to_rxyz(n):
    x, y, z = normalize(n)

    theta = np.arccos(z)
    r = np.sqrt(x * x + y * y)

    rx = -theta * (y / r)
    ry =  theta * (x / r)
    rz = 0

    return np.array([rx, ry, rz])
```