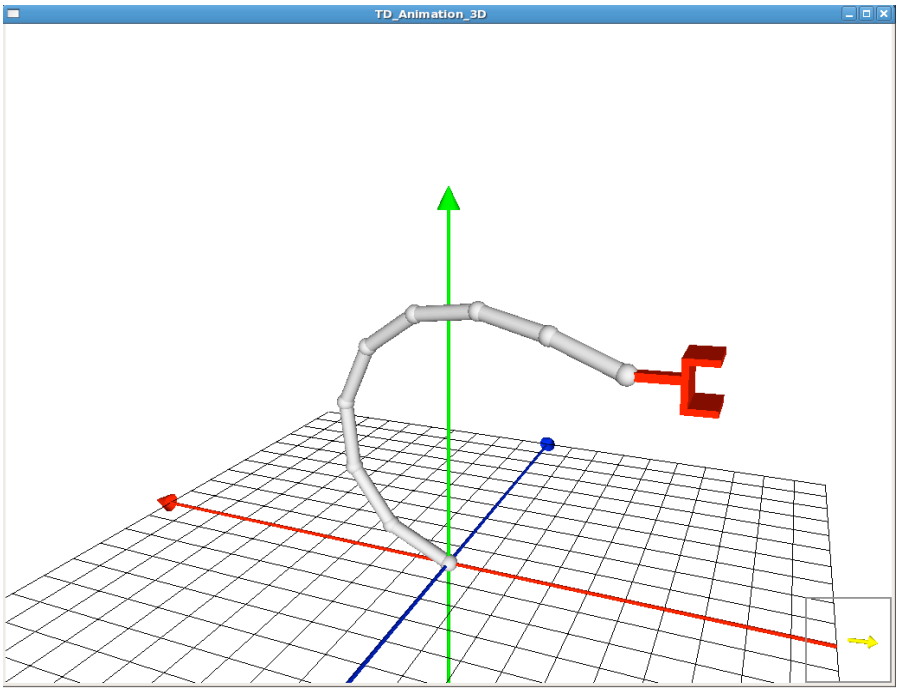
TD Animation 3D

Master I Informatique M. AMMI

Université Paris-11 2016-2017

L'objectif de ce TD est de comprendre et de réaliser une chaîne cinématique inverse représentant le bras d'un robot. Le robot développé doit comprendre 9 Segments, chaque segment comprend 3 rotations (Ox, Oy, Oz). L'effecteur terminal est contraint en position (Pnew : x, y, z) et en rotation (Onew : Ox, Oy, Oz).



Exercice 1 : Cinématique directe

Compléter la fonction « void DirectKinematics(double deltaT) » pour le calcul de la cinématique directe du robot. Cette fonction doit s'appliquer aux segments sélectionnés (int selected) et permet de manipuler les segments avec les touches du clavier de la façon suivante.

- KeyUp/keyDown -> Gamma
- keyLeft/keyRight -> Beta
- keyPgUp/keyPgDown -> Alpha

Exercice 2 : Construction de la Jacobienne

La fonction MakeJacobian(~) prend en entrée le squelette et calcul la matrice Jacobienne.

Complétez cette fonction pour calculer l'ensemble des éléments de la Jacobienne.

La Jacobienne « J » s'exprime sous forme d'une matrice de dimension [_JAC_M_][_JAC_N_] où _JAC_M_ est le nombre de DDL et JAC_N est le nombre de contraintes appliquées à l'effecteur terminal.

- Quelle est la taille de la Jacobienne dans le cas du robot manipulé ?

Les lignes de la Jacobienne expriment la dérivée des fonctions (positions et orientations de l'effecteur terminal) suivant l'ensemble des DDL du système.

- Exprimez les éléments de la Jacobienne

La Jacobienne s'exprime par groupes de 3 colonnes. Chaque groupe correspond à une articulation (articulation 1 -> groupe 1, articulation 2 -> groupe 2, etc.). Chaque colonne (du groupe) correspond à la dérivée des différentes fonctions suivant un DDL (Ox ou Oy ou Oz).

- Proposez un algorithme pour calculer Jacobienne suivant les spécifications indiquées précédemment. L'algorithme doit calculer 3 colonnes de la Jacobienne à la fois.
- Implémenter l'algorithme proposé.

Exercice 3 : Calcul de la cinématique inverse par palier

L'objectif de cette exercice est de compléter la fonction « void InnerIterate() » pour les calculs de cinématique inverse. Nous allons utiliser pour ce calcul la méthode de la Jacobienne transposée.

- Proposez l'algorithme permettant d'obtenir les variations d'angles nécessaires pour atteindre les objectifs. Ce calcul doit comprendre les éléments suivants :
  - Mise en place du vecteur d'entrée « dX »
  - Le calcul de cinématique inverse doit permettre à l'effecteur terminal de converger vers la configuration désirée par palier. À chaque itération, le palier est égal à la demi distance qui sépare la configuration actuelle de la configuration désirée.
  - Intégration d'une matrice de raideur dans le calcul de la Jacobienne « K ». Cette matrice permettra de réduire la flexibilité des articulations en fonction de leur éloignement de l'effecteur terminal.
    - Expliquer la matrice de raideur « K ». voir la fonction « void IKInit() »
  - Mise à jour du squelette à partir du vecteur « dTheta » contenant les variations d'angles calculées.

- Estimation de l'erreur sur la base de la distance algébrique (position et orientation) entre la configuration actuelle et la configuration intermédiaire désirée (½ interval).
- Arrêt des calculs lorsque l'erreur est inférieur au seuil _INNER_EPSILON_ = 15.

- Impléter l'algorithme proposé.

Exercice 4 : Fonction global

La fonction « void InnerIterate() » permet d'atteindre l'objectif par palier. À chaque clique appel à cette fonction, on atteint la moitié de l'objectif final.

- Compléter la fonction « void FindConfiguration () » pour atteindre l'objectif global. Cette fonction fait appel à la fonction void InnerIterate().
  - Estimation de l'erreur sur la base de la distance algébrique (position et orientation) entre la configuration actuelle et la configuration finale désirée.
  - Faire appel à la fonction tan que l'erreur en la configuration actuelle et la configuration désirée est supérieure à OUTER_EPSILON_ = 2.5.