



UNIVERSITÉ DE
SHERBROOKE

Session S7
Module Robotique - Unité d'APP 2
GEI 744

Commande de robots redondants

Département de génie électrique et de génie informatique
Faculté de génie
Université de Sherbrooke

Hiver 2019

Copyright © 2019 Département de génie électrique et de génie informatique.
Université de Sherbrooke

Note : En vue d’alléger le texte, le masculin est utilisé pour désigner les femmes et les hommes.

Modification du robot David Pivin, ing jr., 2019-02-15

Dernière révision Wael Suleiman, ing., PhD , 2018-03-09

Rédigé par Wael Suleiman, ing., PhD

Copyright © 2012 – 2018 Département de génie électrique et de génie informatique. Université de Sherbrooke

Table des matières

Éléments de compétences visés par l'unité	4
Énoncé de la problématique	5
Connaissances nouvelles à acquérir par la résolution de cette problématique	10
Références essentielles à lire	11
Productions à remettre	12
Formation à la pratique procédurale #1	13
Formation à la pratique en laboratoire #1	15
Formation à la pratique procédurale #2	17
Formation à la pratique en laboratoire #2	20
Formation à la pratique procédurale #3	22
Formation à la pratique en laboratoire #3	24
Formation à la pratique en laboratoire #4	27
Seconde rencontre de tutorat	30
Évaluation de l'unité	31
Grille d'évaluation du rapport final :	32

Éléments de compétences visés par l'unité

GEI 744 : Commande de robots redondants

Compétences :

1. Modéliser la cinématique d'un robot redondant
2. Analyse d'une tâche de manipulation ou de locomotion d'un robot redondant
3. Faire la conception préliminaire d'une architecture de contrôle d'un robot redondant

Contenu :

- robots redondants et leurs applications ;
- structure cinématique redondante ;
- méthodes de cinématique inverse généralisée ;
- méthodes numériques en cinématique inverse ;
- priorité de tâches cinématiques ;
- robots humanoïdes ;
- stabilité dynamique et quasi-statique d'un robot humanoïde ;
- tâches de manipulation et de locomotion ;
- contrôle en force d'un robot redondant

Énoncé de la problématique

Après avoir réalisé et mis en oeuvre le système robotisé d'emballage de chocolat, vous avez eu une offre d'emploi très intéressante dans une grande entreprise multinationale. L'entreprise opère principalement dans le domaine de la robotique industrielle et la robotique de service.

Votre expertise en robotique vous a aidé à vous intégrer facilement dans les équipes de R&D au sein de l'entreprise. Quelques semaines après votre embauche, votre patron vient vous parler d'un projet ambitieux en robotique humanoïde. Il s'agit de participer à la nouvelle compétition internationale qui vient d'être annoncée par DARPA (Defense Advanced Research Projects Agency), où le prix est autour de 2 millions de dollars (Figure 1). Il s'agit également d'une excellente opportunité pour l'entreprise afin de montrer et faire savoir ses expertises en robotique.



FIGURE 1 – “DARPA ROBOTICS CHALLENGE”, *Image : DARPA.*

Votre patron vous présente Nakamura-san, ingénieur et chercheur senior au sein de l'entreprise, et confie le mandat de l'étude de faisabilité à vous deux. Nakamura-san possède déjà un robot humanoïde expérimental, il s'agit du robot Darwin-Mini.

Votre collègue vous explique que la génération de trajectoires locomotrices est un défi majeur en robotique humanoïde. Il vous demande de consulter la littérature sur le sujet, et de son côté, il s'occupera de l'interface informatique sous ROS (Robot Operating System).

D'après la littérature, la première étape serait la réalisation d'un pas quasi-statique. Conscient de l'ampleur de la tâche, vous décidez de diviser la problématique en deux parties : la première partie porte sur la modélisation et la validation de l'architecture de contrôle, et la deuxième porte sur la validation sur le robot Darwin-Mini avec l'aide de votre collègue Nakamura-san. Les sous-étapes de votre mandat sont :

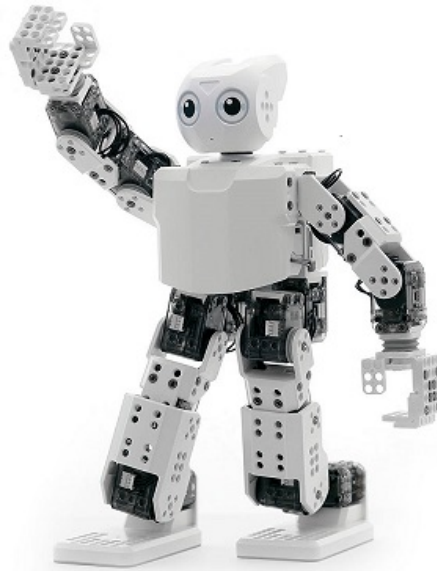


FIGURE 2 – Le robot humanoïde Darwin-Mini.

1. Décrire la cinématique des jambes du robot Darwin-Mini, et poser les paramètres cinématiques des joints et les variables cinématiques des articulations en utilisant la représentation de Denavit-Hartenberg.
2. Définir une trajectoire cubique du pied qui relie deux empreintes \mathbb{A} et \mathbb{B} , ces empreintes sont situés sur le sol qu'il est supposé parfaitement plat, voir Figure 3. Nous supposons que la position vertical du pied, durant la phase de transition, est inférieure à $Z_{\text{pied}} = 0.03 \text{ m}$.
3. Implémenter une cinématique inverse généralisée permettant le suivi de la trajectoire identifiée en 2. Étant donnée la redondance dans la structure cinématique de la jambe, vous décidez d'ajouter une contrainte sur l'orientation du pied afin qu'il soit toujours horizontal. Cette contrainte est considérée comme une tâche cinématique d'une priorité secondaire.
4. Proposer une architecture de contrôle qui permet de réaliser la tâche définie en 3.
5. Développer un algorithme en C++ pour exécuter les trajectoires calculées dans l'étape 3 sur le robot Darwin-Mini avec ROS.

Annexe :

À compléter ...

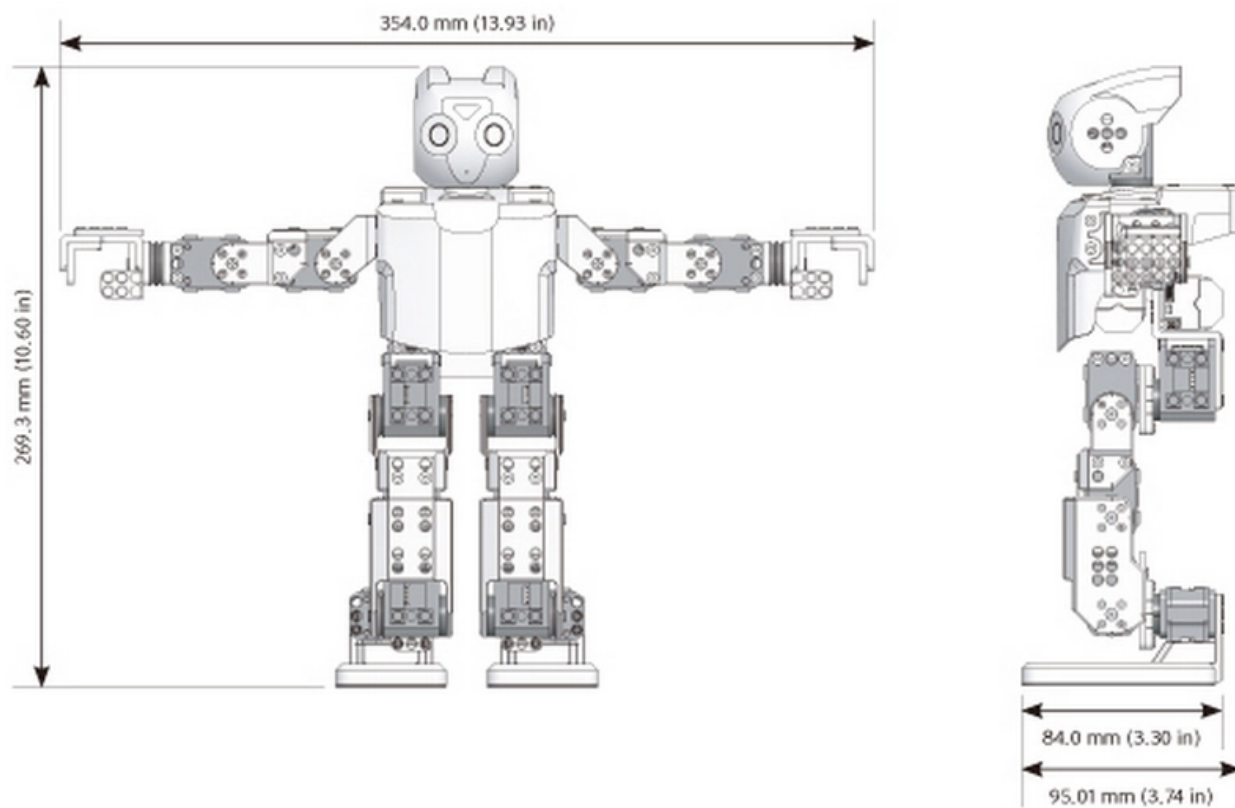


FIGURE 4 – Le robot Darwin-Mini

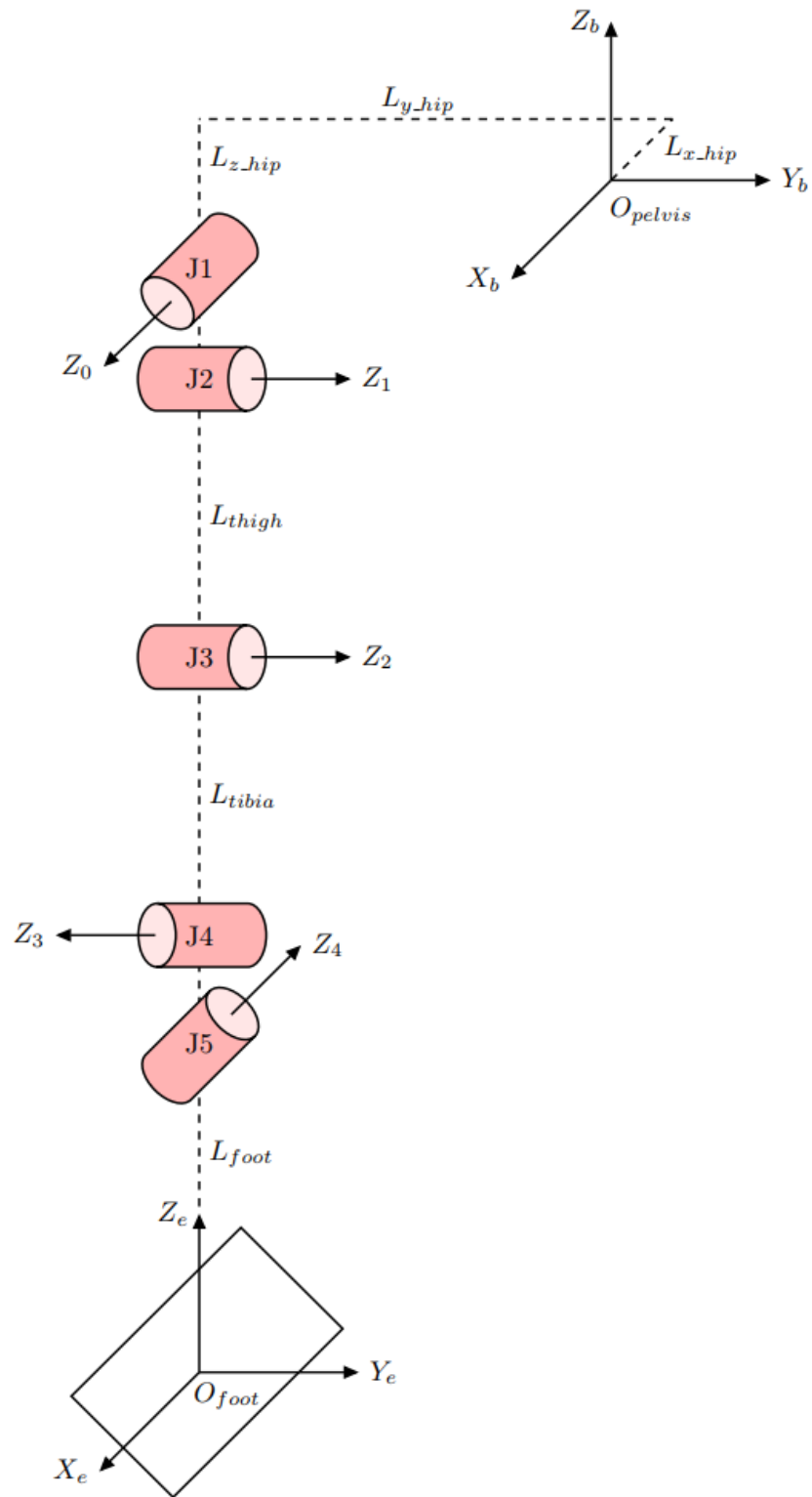


FIGURE 5 - Structure cinématique de la jambe droite du robot Darwin-Mini

N.B L'origine du repère J0 et J1 ainsi que J4 et J5 sont superposé.

Connaissances nouvelles à acquérir par la résolution de cette problématique

Connaissances déclaratives : QUOI

- Les robots redondants et leurs spécifications.
- Le modèle cinématique inverse d'un robot redondant.
- La singularité cinématique d'un robot redondant
- Le modèle cinématique inverse généralisé d'un robot redondant.
- Méthodes de résolution numérique de la cinématique inverse généralisée.
- Priorité de tâches cinématiques
- Contrôle en force d'un robot redondant

Connaissances procédurales : COMMENT

- Choisir la configuration cinématique et les degrés de liberté nécessaires d'un robot redondant afin de réaliser une tâche.
- Calculer analytiquement la matrice jacobienne d'un robot redondant.
- Trouver les configurations singulières d'un robot redondant
- Calculer l'espace nulle d'une matrice jacobienne.
- Calculer une solution numériquement robuste de la cinématique inverse généralisée.
- Calculer l'espace nulle d'une matrice jacobienne.
- Calculer la projection orthogonale sur l'espace nulle d'une tâche cinématique.
- Contraintes naturelles et artificielles.

Connaissances conditionnelles : QUAND

- Choix d'une méthode numériquement robuste pour résoudre la cinématique inverse généralisée.
- Utiliser un contrôleur en force pour un robot redondant.

Références essentielles à lire

Le livre de référence est “**Robotics: Modelling, Planning and Control**, éditeurs : B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, Springer; édition 2009”. Les chapitres à lire sont listés dans l’ordre de lecture :

- Chapitre 2.9 à 2.10
- Chapitre 3.1, 3.2, 3.3 et 3.4
- Chapitre 3.5 à 3.7
- Chapitre 5.1, 5.3 et 5.4
- Chapitre 9.5 et 9.6

Extraits tirés du livre : “**Advanced robotics : redundancy and optimization**, *Yoshihiko Nakamura*, Addison-Wesley Pub. Co., 1991”.

Productions à remettre

Travail de rédaction des productions exigées

1. Consignes pour la préparation du rapport sur papier

- Les rapports doivent être remis par groupe de 2 à 4 personnes.
- Une grille de correction est disponible à la fin du document pour vous guider dans la rédaction du rapport et la réalisation de la simulation.
- Répondre à toutes les questions de la problématique.
- Joindre le programme codé en Matlab prêt à être exécuté.
- Joindre le programme codé en langage C++ associé au contrôleur du robot.
- Remettre un fichier .zip contenant vos code Matlab et C++ prêts à être exécuté ou compilé, une vidéo de la résolution de la problématique sous Gazebo et un fichier .txt contenant les noms et matricules des membres de l'équipe.
- Les modalités finales du dépôt seront communiquées en temps opportun sur le site de l'APP.

2. Modalités d'évaluation du rapport d'APP

- Le rapport d'APP contribue à l'évaluation des éléments de compétence de l'unité.
- La qualité de la communication sera évaluée sur le plan de la qualité de l'écrit et de la présentation. La grille de correction comprend un coefficient qui en tient compte.
- ❖ Inscrire votre nom et votre numéro de matricule sur les documents remis.

Formation à la pratique procédurale #1

Connaissances déclaratives : QUOI

- Les rebots redondants et leurs spécifications.
- Le modèle cinématique inverse d'un robot redondant.
- la singularité cinématique d'un robot redondant

Connaissances procédurales : COMMENT

- Choisir la configuration cinématique et les degrés de liberté nécessaires d'un robot redondant afin de réaliser une tâche.
- Calculer analytiquement la matrice jacobienne d'un robot redondant.
- Trouver les configurations singulières d'un robot redondant

❖ Lectures préalables dans le livre “Robotics: Modelling, Planning and Control” :

1. Chapitre 2.9 et 2.10
2. Chapitre 3.1, 3.2 et 3.3

Exercices à résoudre :

Dans le livre “Robotics: Modelling, Planning and Control” :
exercice 3.2 , exercice 3.3 , exercice 3.4 , exercice 3.5 , exercice 3.6 ,

Exercice supplémentaire :

Une entreprise vous a approché afin de concevoir un robot qui serait capable à déplacer des produits chimiques en éprouvettes entre deux positions (de la position **A** vers la position **B**) en gardant les éprouvettes en position vertical tout au long de la trajectoire, voir Figure 6.

1. En considérant l'espace de travail présenté dans la Figure 6, quel est le nombre minimal de degré de liberté pour réaliser la tâche ?
2. Proposez une structure cinématique d'un robot qui possède le nombre minimal de degré de liberté et capable à réaliser la tâche.

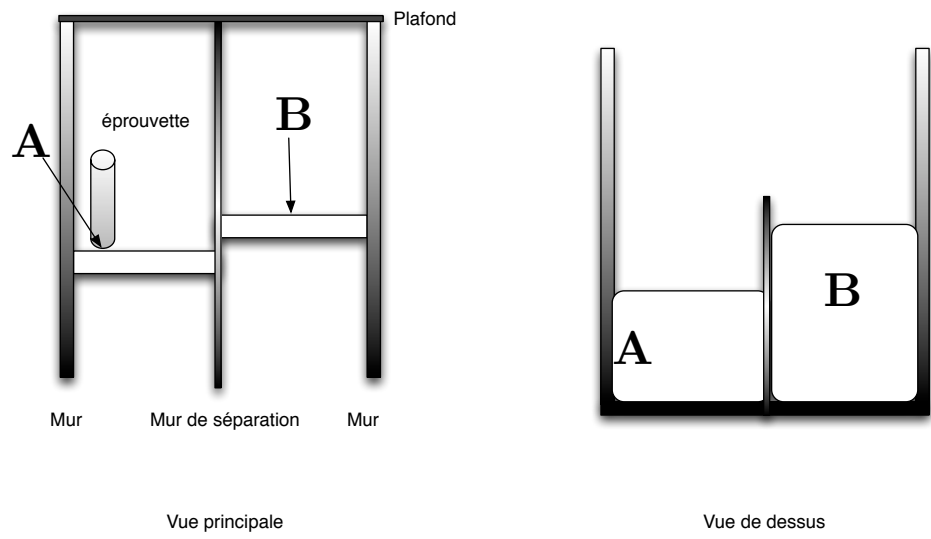


FIGURE 6 – L’espace de travail et les positions initiale et finale de l’éprouvette.

Formation à la pratique en laboratoire #1

Il est prévu quatre activités de laboratoire où l'on collabore par équipe de 2 par ordinateur. Ces activités commenceront à la première semaine et se poursuivront lors des trois autres laboratoires.

Logiciel utilisé dans ce laboratoire : Matlab

Le but du laboratoire #1 :

1. Calculer numériquement la cinématique directe d'une structure cinématique

Dans ce laboratoire, nous intéressons aux trois articulations rotatives de la poignée du robot Comau Smart5 (ci-dessus).

Le tableau des paramètres de Denavit-Hartenberg de la poignée est le suivant :

Link	a_i	α_i	d_i	θ_i
1	0	α	d_4	θ_4
2	0	$-\alpha$	d_5	θ_5
3	1	0	0	$\theta_6 + \frac{\pi}{2}$

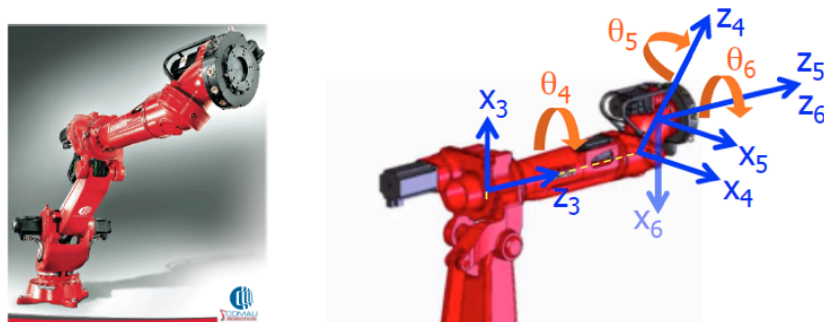


FIGURE 7 – Comau Smart5

Il est demandé de :

1. Implémenter une fonction pour calculer les matrices homogènes A_i^{i-1} à partir des paramètres D-H d'une façon analytique.
2. Calculer la matrice homogène du repère de l'effecteur ("end-effector") dans le repère {3} d'une façon analytique.
3. Calculer la matrice homogène du repère {3} dans le repère de l'effecteur d'une façon analytique.
4. Trouver l'espace du travail de poignée sachant que : $-150^\circ \leq \begin{Bmatrix} \theta_4 \\ \theta_5 \\ \theta_6 \end{Bmatrix} \leq 150^\circ$, et les valeurs numériques : $\alpha = 65^\circ$, $d_4 = 0.75m$, $d_5 = 0.5m$.

5. Calculer la matrice jacobienne exprimée par la relation suivante :

$$\omega_e = J_o \dot{q} \quad (1)$$

où ω_e est la vitesse angulaire de l'effecteur ("end-effector") du robot et q est le vecteur angulaire des articulations $q = [\theta_4, \theta_5, \theta_6]^T$.

6. Trouvez les configurations singulières de l'équation différentielle (1).

Formation à la pratique procédurale #2

Connaissances déclaratives : QUOI

- Le modèle cinématique inverse généralisé d'un robot redondant.
- Méthodes de résolution numérique de la cinématique inverse généralisée.

Connaissances procédurales : COMMENT

- Calculer l'espace nulle d'une matrice jacobienne.
- Calculer une solution numériquement robuste de la cinématique inverse généralisée.

❖ Lectures préalables dans le livre “Robotics: Modelling, Planning and Control” :

1. Chapitre 3.4, 3.5, 3.6, 3.7.1, 3.7.2 et 3.7.3
2. L'article “Feasible pattern generation method for humanoid robots” (voir l'annexe).

Exercices à résoudre :

Dans le livre “Robotics: Modelling, Planning and Control” :
exercice 3.9 , exercice 3.10

Exercice supplémentaire I :

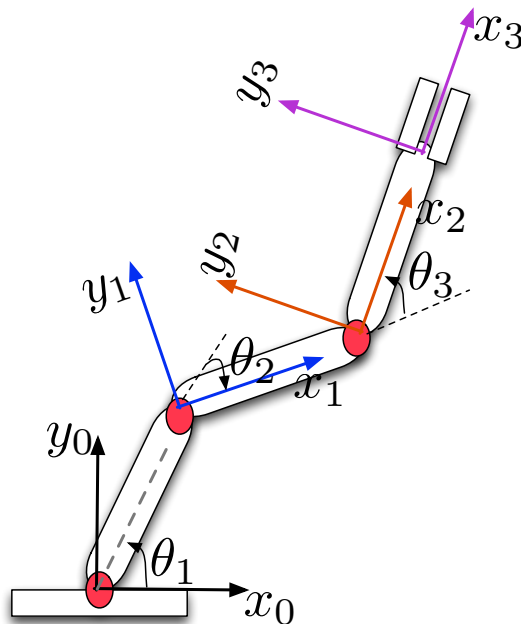


FIGURE 8 – Robot manipulator de 3 segments

Supposons que la position de l'origine du repère 3 est : $\mathbf{P}_e = [x_3 \ y_3]^T$, $\mathbf{q} = [\theta_1 \ \theta_2 \ \theta_3]^T$ et la longueur de chaque lien cinématique est l_i .

En se basant sur les paramètres D-H, nous trouvons que la matrice homogène est donnée par l'équation suivante :

$$T_N^0 = \begin{bmatrix} \cos\left(\sum_{i=1}^3 \theta_i\right) & -\sin\left(\sum_{i=1}^3 \theta_i\right) & 0 & P_x \\ \sin\left(\sum_{i=1}^3 \theta_i\right) & \cos\left(\sum_{i=1}^3 \theta_i\right) & 0 & P_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Où :

$$P_x = \sum_{k=1}^3 l_k \cos\left(\sum_{i=1}^k \theta_i\right)$$

$$P_y = \sum_{k=1}^3 l_k \sin\left(\sum_{i=1}^k \theta_i\right)$$

1. Calculer la matrice jacobienne (J_P) exprimée par la relation suivante :

$$\dot{\mathbf{P}}_e = J_P \dot{\mathbf{q}}$$

2. démontrer que :

$$J_P = \begin{bmatrix} \frac{\partial P_x}{\partial \theta_1} & \frac{\partial P_x}{\partial \theta_2} & \frac{\partial P_x}{\partial \theta_3} \\ \frac{\partial P_y}{\partial \theta_1} & \frac{\partial P_y}{\partial \theta_2} & \frac{\partial P_y}{\partial \theta_3} \end{bmatrix}$$

3. Supposons que la position désirée de l'origine du repère 3 est \mathbf{P}_d et l'erreur de position $\mathbf{e} \triangleq \mathbf{P}_d - \mathbf{P}_e$, proposez une architecture de contrôle qui permet de suivre la trajectoire désirée (\mathbf{P}_d).
4. Calculer la matrice jacobienne J_2 exprimée par la relation suivante :

$$\dot{\mathbf{P}}_2 = J_2 \dot{\mathbf{q}}$$

où \mathbf{P}_2 est l'origine du repère 2.

5. Trouver les configurations singulières de deux premières articulations ($\det(J_2) = 0$).

Exercice supplémentaire II :

Un problème majeur dans la résolution de cinématique inverse généralisée est de s'assurer que les limites angulaire sont respectées ($q_j^+ \leq q_j \leq q_j^-$).

Une solution qui était proposée (voir l'annexe) est de transformer le problème de cinématique inverse en problème d'optimisation exprimé comme suit :

$$\min_{\dot{\mathbf{q}}} \dot{\mathbf{q}}$$

sous les contraintes :

$$\dot{\mathbf{P}}_e = J_P \dot{\mathbf{q}}$$

$$\dot{q}_j^{\min}(q_j) \leq \dot{q}_j \leq \dot{q}_j^{\max}(q_j), \text{ pour } j \in \{1, \dots, n\}$$

$\dot{q}_j^{max}(q_j)$ et $\dot{q}_j^{min}(q_j)$ sont definis comme suit :

$$\dot{q}_j^{max}(q_j) = \begin{cases} \xi \frac{(q_j^+ - q_j) - q_s}{q_i - q_s} & \text{si } q_j^+ - q_j \leq q_i, \\ \dot{q}_j^+ & \text{sinon} \end{cases} \quad (2)$$

$$\dot{q}_j^{min}(q_j) = \begin{cases} -\xi \frac{(q_j - q_j^-) - q_s}{q_i - q_s} & \text{si } q_j - q_j^- \leq q_i, \\ \dot{q}_j^- & \text{sinon} \end{cases} \quad (3)$$

Où q_j^+ , q_j^- , \dot{q}_j^+ et \dot{q}_j^- sont les limites physiques (supérieures et inférieures) des angles et des vitesses pour chaque articulation.

1. Démontrer que la solution du problème d'optimisation respecte la contrainte suivante :

(a) $q_j^+ \leq q_j \leq q_j^-$

2. Quelle est la contrainte sur les valeurs q_i , q_s et ξ afin de s'assurer que $\dot{q}_j^+ \leq \dot{q}_j \leq \dot{q}_j^-$.

Exercice supplémentaire III :

Démontrer l'équation (3.84), page 139 dans le livre "Robotics: Modelling, Planning and Control".

Formation à la pratique en laboratoire #2

Logiciel utilisé dans ce laboratoire : Matlab

Le but du laboratoire #2 :

1. Simuler un robot redondant ainsi que calculer la matrice jacobienne et sa pseudo-inverse dans plusieurs configurations cinématiques.
2. Calculer numériquement la cinématique inverse d'un robot redondant
3. Calculer numériquement la cinématique directe d'un robot redondant

Étape I : Calculer numériquement la cinématique directe d'un robot redondant :

Implémenter un algorithme afin de calculer la cinématique directe en utilisant la matrice homogène (les paramètres de Denavit–Hartenberg).

❖ Note : la matrice homogène est donnée par l'équation suivante :

$$T_N^0 = \begin{bmatrix} \cos\left(\sum_{i=1}^N \theta_i\right) & -\sin\left(\sum_{i=1}^N \theta_i\right) & 0 & P_x \\ \sin\left(\sum_{i=1}^N \theta_i\right) & \cos\left(\sum_{i=1}^N \theta_i\right) & 0 & P_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Où :

$$P_x = \sum_{k=1}^N l_k \cos\left(\sum_{i=1}^k \theta_i\right)$$

$$P_y = \sum_{k=1}^N l_k \sin\left(\sum_{i=1}^k \theta_i\right)$$

où N est le numéro de l'articulation.

Étape II : Calculer la matrice jacobienne et sa pseudo-inverse :

1. Calculer numériquement la matrice jacobienne du robot dans la Figure 9. On suppose que la longueur de chaque lien cinématique est l_i , la valeur angulaire de chaque articulation est θ_i , et $\mathbf{q} = [\theta_1 \ \theta_2 \ \dots \ \theta_7]^T$. La matrice jacobienne est exprimée sous la forme :

$$\dot{\mathbf{P}}_e = J_P \dot{\mathbf{q}}$$

2. Calculer la pseudo-inverse de la matrice jacobienne en utilisant la commande :

`pinv(Jp)`

Étape III : Calculer numériquement la cinématique inverse d'un robot redondant :

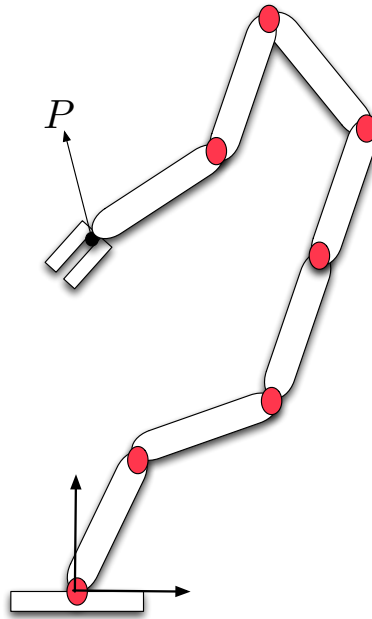


FIGURE 9 – Exemple d'un robot redondant

On suppose que les longueurs l_i sont identiques ($l_i = l = 1$). Résoudre le problème de cinématique inverse généralisé afin que le point P atteigne en 3 seconds la position cartésien $P_d = [4 \quad 3.5]^T$ à partir d'une position initiale P_0 qui correspond à $\mathbf{q} = \mathbf{0}$.

❖ Note : Utilisez une interpolation cubique entre les points P_0 et P_d .

Formation à la pratique procédurale #3

Connaissances déclaratives : QUOI

- Priorité de tâches cinématiques
- Contrôle en force d'un robot redondant

Connaissances procédurales : COMMENT

- Calculer l'espace nulle d'une matrice jacobienne.
- Calculer la projection orthogonale sur l'espace nulle d'une tâche cinématique.
- Contraintes naturelles et artificielles

❖ Lectures préalables dans le livre “Robotics: Modelling, Planning and Control” :

1. Chapitre 5.1, 5.3 et 5.4
2. Chapitre 9.5 et 9.6
3. Extraits tirés du livre : “**Advanced robotics : redundancy and optimization**, Yoshihiko Nakamura, Addison-Wesley Pub. Co., 1991”.

Exercice I :

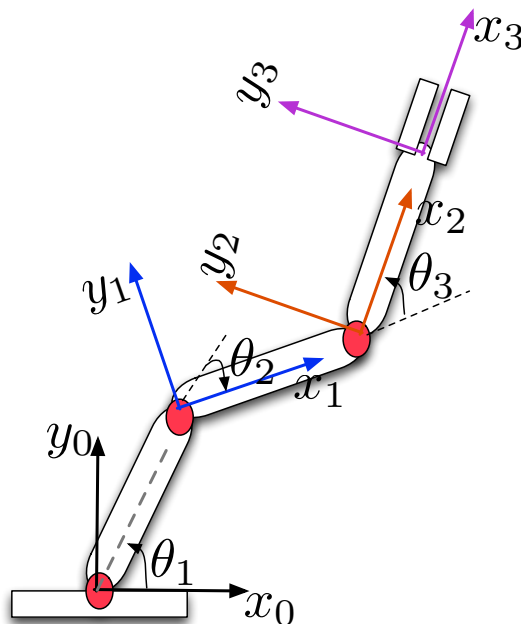


FIGURE 10 – Robot manipulator de 3 segments

1. Donner l'expression de \dot{q} afin que les tâches cinématiques ci-dessous soit réalisés par l'ordre suivant de priorité :
 - (a) Priorité élevée : le point P (l'origine du repère 3) suit une trajectoire cartésienne désirée P_d .

- (b) Priorité modérée : l'orientation du repère 3 suit une séquence des angles désirée Θ_d .
2. Proposer une architecture de contrôle qui permet de résoudre les tâches cinématiques en 1.

Exercice II :

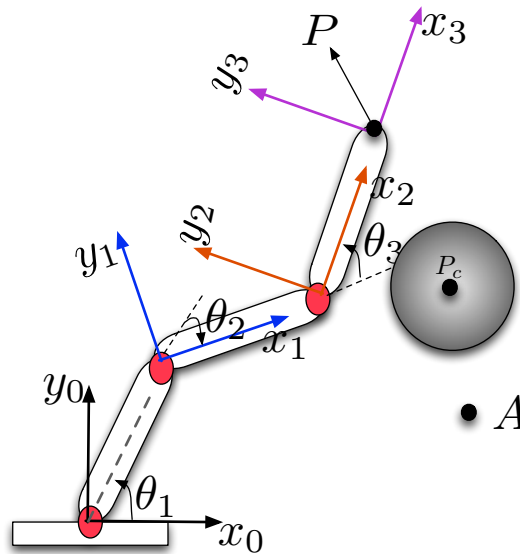


FIGURE 11 – Robot manipulator de 3 segments

1. Calculer la distance d_c entre le point P et le centre du cercle P_c en fonction du vecteur $q = [\theta_1 \ \theta_2 \ \theta_3]^T$.
2. Calculer le dérivé temporel de la distance d_c ($\frac{dd_c}{dt}$) en fonction de \dot{q} et q et la matrice jacobienne J .
3. Donner l'expression de \dot{q} afin que le point P atteigne le point A tout en évitant la collision avec le cercle.
4. Proposer une architecture de contrôle qui permet de résoudre les tâches cinématiques en 3.

Exercices à résoudre :

Dans l'annexe de ce guide : exercice 11.1 , exercice 11.2 , exercice 11.11

Formation à la pratique en laboratoire #3

Logiciel utilisé dans ce laboratoire : Matlab & C++

Le but du laboratoire #3 :

1. Priorité de tâches cinématiques
2. Calculer l'espace nulle d'une matrice jacobienne.
3. Calculer la projection orthogonale sur l'espace nulle d'une tâche cinématique.

Étape I : Calculer la matrice jacobienne liée à la vitesse angulaire :

1. Calculer numériquement la matrice jacobienne du robot dans la Figure 12. On suppose que la longueur de chaque lien cinématique est l_i , la valeur angulaire de chaque articulation est θ_i , et $\mathbf{q} = [\theta_1 \ \theta_2 \ \dots \ \theta_7]^T$. La matrice jacobienne est exprimée sous la forme :

$$\boldsymbol{\omega} = \mathbf{J}_o \dot{\mathbf{q}}$$

où $\boldsymbol{\omega}$ est la vitesse angulaire du repère de la pince (repère 7).

2. Calculer la pseudo-inverse de la matrice jacobienne en utilisant la commande :

```
pinv(Jo)
```

Étape II : Priorité de tâches cinématiques. :

À partir de la position initiale P_0 qui correspond à la vecteur $\mathbf{q}(\text{rad}) = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]^T$, calculer la cinématique inverse afin que les tâches cinématiques classées par ordre de priorité suivantes soient réalisées :

1. Priorité élevée : le point P (l'origine du repère 7) atteigne le point $P_d = [6 \ 3]^T$ en 2 seconds.
2. Priorité modérée : Le repère 7 soit parallèle au repère 0 (l'orientation=0).

Étape III : Calcul de la pseudo-inverse d'une matrice avec C++. :

La librairie C++ Eigen3 permet le calcul matriciel et vectoriel d'une façon efficace et robuste.

Plus de détails sur Eigen3 :

<http://eigen.tuxfamily.org/>

Voici un exemple pour calculer la pseudo-inverse d'une matrice (Damped Least-square (DLS) inverse) :

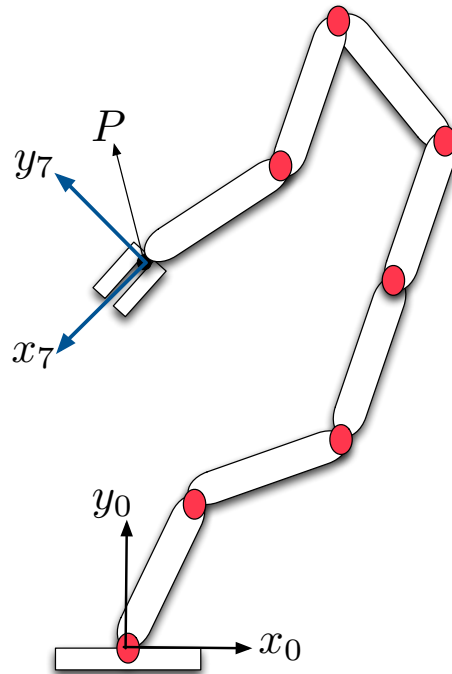


FIGURE 12 – Exemple d'un robot redondant

Listing 1 – main.cpp

```
//APP GEI 744, Copyright 2013, Wael Suleiman
#include <iostream>
#include "Eigen/Dense"
#include "matrixpinv.h"

using namespace Eigen;
using namespace std;

int main()
{
    MatrixXd m(3,6);
    m(0,0) = 3;
    m(1,0) = 2.5;
    m(0,1) = -1;
    m(1,1) = m(1,0) + m(0,1);
    m(2,4)=1;
    m(2,5)=1.7;
    m(1,5)=1.2;

    cout << "The matrix m is:\n" << m << endl;

    // Least-square solution of a linear equation Ax=b

    MatrixXd A=MatrixXd::Random(3, 7);
    cout << "Here is the matrix A:\n" << A << endl;
    VectorXd b=VectorXd::Random(3);
    cout << "Here is the right hand side b:\n" << b << endl;
    cout << "The least-squares solution is:\n"
        << DLS_LE(A,b) << endl;

    // DLS pseudo-inverse of a matrix

    cout << "The DLS pseudo-inverse of A is:\n"
```

```

    << DLS_inverse(A) << endl;

}

```

Listing 2 – matrixpinv.cpp

```

//APP GEI 744, Copyright 2013, Wael Suleiman

#include <stdio.h>
#include <stdlib.h>

#include <Eigen/Dense>
#include <Eigen/Core>
#include <Eigen/SVD>
using namespace Eigen;

MatrixXd DLS_LE(MatrixXd& A, VectorXd& b) {

    return MatrixXd(A.jacobiSvd(ComputeThinU | ComputeThinV).solve(b));
}

MatrixXd DLS_inverse(MatrixXd& A, double epsilon= 1E-3) {

    MatrixXd AT= A.transpose();
    int rows=A.rows();
    MatrixXd I= MatrixXd::Identity(rows,rows);
    MatrixXd DLS_A=A*AT + epsilon*epsilon*I;

    return MatrixXd(AT*DLS_A.inverse());
}

```

Listing 3 – matrixpinv.h

```

// APP GEI 744, Copyright 2013, Wael Suleiman
#include <stdio.h>
#include <stdlib.h>

#include <Eigen/Dense>
#include <Eigen/Core>
#include <Eigen/SVD>
using namespace Eigen;
/**
 * DLS of the linear equation: Ax=b
 * DLS pseudo-inverse of a matrix
 */

MatrixXd DLS_LE(MatrixXd& A, VectorXd& b);

MatrixXd DLS_inverse(MatrixXd& A, double epsilon= 1E-3);

```

Formation à la pratique en laboratoire #4

Logiciels utilisés dans ce laboratoire : ROS, Gazebo et C++

Le but du laboratoire #4 :

1. Simuler un mouvement du robot Darwin-Mini sous Gazebo.

À compléter ...

À compléter ...

À compléter ...

Seconde rencontre de tutorat

Validation des connaissances – Bilan de groupe – Travail personnel de synthèse et d'études correctives.

Évaluations sommative et finale :

Ces évaluations écrites portent sur tous les objectifs d'apprentissage de l'unité.

- Durée 3 heures par examen
- Vous avez le droit d'utiliser Matlab durant l'examen
- Documents permis :
 - ❖ Livre « Robotics : Modelling, Planning and Control »
 - ❖ **ATTENTION** : Vous n'avez **pas** droit à vos notes de cours pendant l'examen

Évaluation de l'unité

La note attribuée aux activités pédagogiques de l'unité est une note individuelle. L'évaluation portera sur les compétences figurant dans la description des activités pédagogiques.

Ces compétences, ainsi que la pondération de chacune d'entre elles dans l'évaluation de cette unité, sont :

	GEI 744-1	GEI 744-2	GEI 744-3	Total
Rapport d'APP (équipe de 2 à 4)	30	100	90	220
Examen sommatif (individuel)	45	140	125	310
Examen final (individuel)	60	165	145	370
Total	135	405	360	900

Grille d'évaluation du rapport final :

Contenu		GEI 744-1	GEI 744-2	GEI 744-3
Partie I :				
1	Dessin de la cinématique de la jambe de Darwin-Mini	6		
2	Repères de chaque articulation de la jambe	12	15	
3	Les paramètres Denavit-Hartenberg	12	10	
4	Les trajectoires cubique (x, y et z) du pied		10	10
5	Les matrices jacobiennes de la position et l'orientation du pied		20	25
6	L'architecture de contrôle		20	30
Partie II :				
7	Interface avec ROS en langage C++		15	15
8	Execution de la trajectoire (vidéo de la simulation sous Gazebo)		10	10
Total		30	100	90

Validation de la résolution de la problématique :

30 points bonus (GEI744-2 :10 points, GEI744-3 : 20 points) seront attribués aux équipes qui réussissent à compiler leurs codes ROS et executer correctement un pas sur le robot Darwin-Mini.

Annexe

- [17] J.K. Salisbury and J. Craig, "Articulated Hands: Force Control and Kinematic Issues," *International Journal of Robotics Research*, Vol. 1, No. 1.
- [18] S. Drake, "Using Compliance in Lieu of Sensory Feedback for Automatic Assembly," Ph.D. Thesis, Mechanical Engineering Department, MIT, September 1977.
- [19] R. Featherstone, S.S. Thiebaut, and O. Khatib, "A General Contact Model for Dynamically-Decoupled Force/Motion Control," *Proceedings of the IEEE Conference on Robotics and Automation*, Detroit, 1999.

EXERCISES

- 11.1** [12] Give the natural constraints present for a peg of square cross section sliding into a hole of square cross section. Show your definition of $\{C\}$ in a sketch.
- 11.2** [10] Give the artificial constraints (i.e., the trajectory) you would suggest in order to cause the peg in Exercise 11.1 to slide further into the hole without jamming.
- 11.3** [20] Show that using the control law (11.14) with a system given by (11.9) results in the error-space equation

$$\ddot{e}_f + k_{vf}\dot{e}_f + (k_{pf} + m^{-1}k_e)e_f = m^{-1}k_e f_{\text{dist}},$$

and, hence, that choosing gains to provide critical damping is possible only if the stiffness of the environment, k_e , is known.

- 11.4** [17] Given

$${}^A_B T = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 10.0 \\ 0.500 & 0.866 & 0.000 & 0.0 \\ 0.000 & 0.000 & 1.000 & 5.0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

if the force–torque vector at the origin of $\{A\}$ is

$${}^A_v = \begin{bmatrix} 0.0 \\ 2.0 \\ -3.0 \\ 0.0 \\ 0.0 \\ 4.0 \end{bmatrix},$$

find the 6×1 force–torque vector with reference point at the origin of $\{B\}$.

- 11.5** [17] Given

$${}^A_B T = \begin{bmatrix} 0.866 & 0.500 & 0.000 & 10.0 \\ -0.500 & 0.866 & 0.000 & 0.0 \\ 0.000 & 0.000 & 1.000 & 5.0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

if the force–torque vector at the origin of $\{A\}$ is

$${}^A_v = \begin{bmatrix} 6.0 \\ 6.0 \\ 0.0 \\ 5.0 \\ 0.0 \\ 0.0 \end{bmatrix},$$

find the 6×1 force–torque vector with reference point at the origin of $\{B\}$.

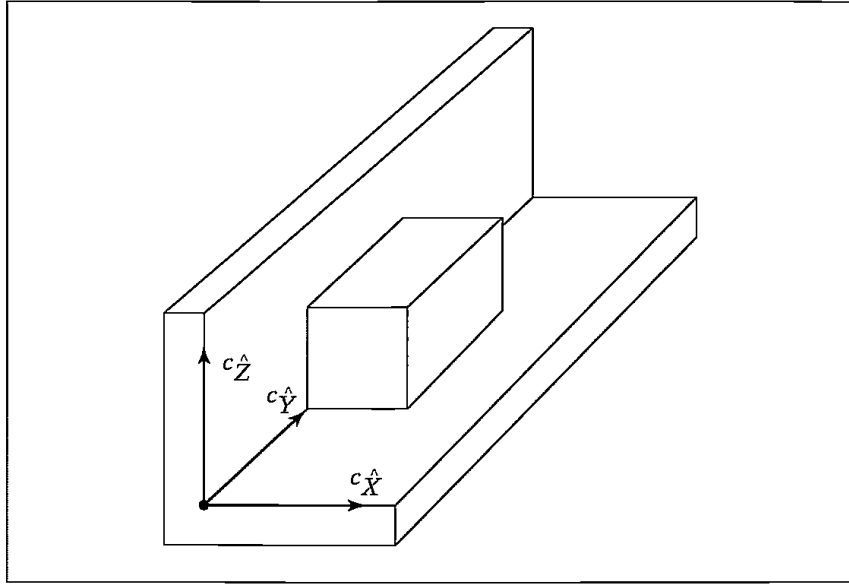


FIGURE 11.14: A block constrained by a floor below and a wall to the side.

- 11.6** [18] Describe in English how you accomplish the insertion of a book into a narrow crack between books on your crowded bookshelf.
- 11.7** [20] Give the natural and artificial constraints for the task of closing a hinged door with a manipulator. Make any reasonable assumptions needed. Show your definition of $\{C\}$ in a sketch.
- 11.8** [20] Give the natural and artificial constraints for the task of uncorking a bottle of champagne with a manipulator. Make any reasonable assumptions needed. Show your definition of $\{C\}$ in a sketch.
- 11.9** [41] For the stiffness servo system of Section 11.7, we have made no claim that the system is stable. Assume that (11.23) is used as the servo portion of a decoupled and linearized manipulator (so that the n joints appear as unit masses). Prove that the controller is stable for any K_v which is negative definite.
- 11.10** [48] For the stiffness servo system of Section 11.7, we have made no claim that the system is or can be critically damped. Assume that (11.23) is used as the servo portion of a decoupled and linearized manipulator (so that the n joints appear as unit masses). Is it possible to design a K_p that is a function of Θ and causes the system to be critically damped over all configurations?
- 11.11** [15] As shown in Fig. 11.14, a block is constrained below by a floor and to the side by a wall. Assuming this contacting situation is maintained over an interval of time, give the natural constraints that are present.

PROGRAMMING EXERCISE (PART 11)

Implement a Cartesian stiffness-control system for the three-link planar manipulator by using the control law (11.23) to control the simulated arm. Use the Jacobian written in frame $\{3\}$.

For the manipulator in position $\Theta = [60.0 \ 90.030.0]$ and for K_{px} of the form

$$K_{px} = \begin{bmatrix} k_{\text{small}} & 0.0 & 0.0 \\ 0.0 & k_{\text{big}} & 0.0 \\ 0.0 & 0.0 & k_{\text{big}} \end{bmatrix},$$