

Optimisation d'un manipulateur parallèle plan

Romain PESSIA, Maxime HOLSTEIN

12 février 2017

Présentation du problème

On s'intéresse ici au sujet numéro 1, c'est-à-dire à l'optimisation d'un robot manipulateur planaire à trois degrés de liberté. En particulier, nous nous sommes tournés vers un manipulateur de type PRR (où la lettre P symbolise une liaison prismatique, et R une liaison rotatoire, comme expliqué dans [1]). Ce robot est composé d'une base triangulaire équilatérale fixe $A_1A_2A_3$ dont chaque côté est muni d'une liaison prismatique actionnant trois barres d'extrémités B_i et de longueurs variables. Trois barres secondaires B_iC_i sont liées aux barres A_iB_i par des liaisons rotoïdes. Enfin, les points C_i forment la plate-forme mobile $C_1C_2C_3$, également de forme triangulaire équilatérale. Son centre de gravité P représente la position de l'effecteur. La Figure 1 offre une représentation du robot et de sa paramétrisation géométrique telle qu'utilisée par S. Caro *et al.* ([2], p. 374).

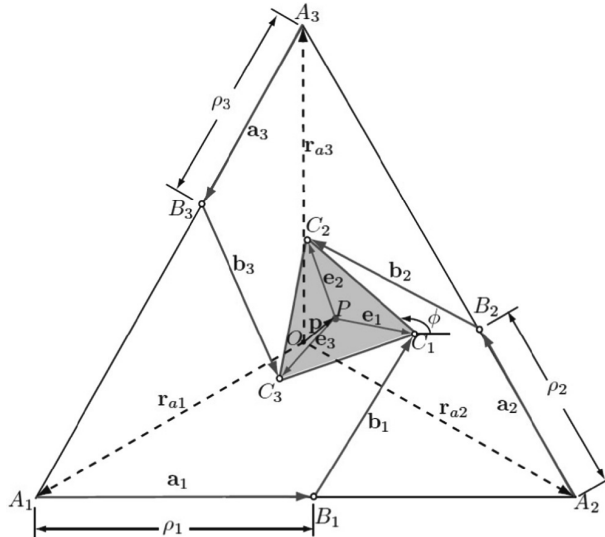


Figure 1: Représentation du robot manipulateur 3-PRR

Le problème d'optimisation demande la minimisation d'un seul paramètre, qui est l'espace total occupé par le robot. On décide alors de résoudre ce

problème en agissant sur les trois paramètres suivants : le rayon r du cercle circonscrit au triangle délimitant la plate-forme mobile ; le rayon R du cercle circonscrit au triangle formant la base du manipulateur ; et la longueur L_b des barres B_iC_i , que l'on choisira identique pour les trois. Le problème d'optimisation consiste alors simplement à minimiser R .

Ce paramétrage impose de prendre en compte deux contraintes géométriques :

- les joints des liaisons prismatiques ne peuvent pas se trouver en-dehors du triangle $A_1A_2A_3$, ce qui implique l'inégalité suivante :

$$0 < \rho_i < \sqrt{3}R \quad (1)$$

- Le robot doit pouvoir être assemblé, d'où la condition sur les données du problème obtenue à partir de la Figure 2. On a ainsi :

$$R \leq 2(L_b + r) \quad (2)$$

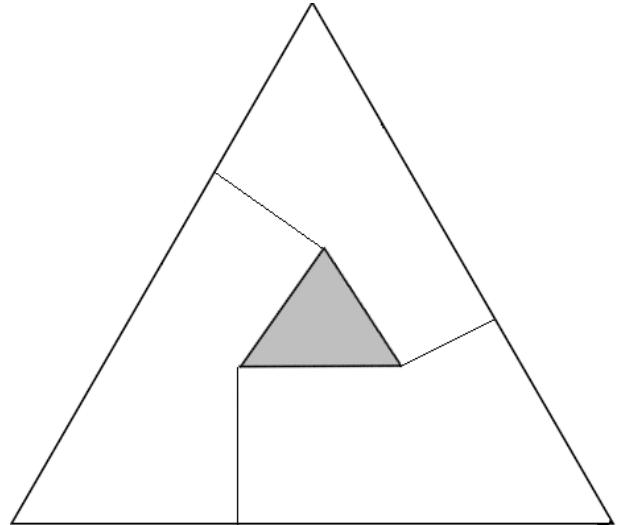


Figure 2: Représentation de la configuration limite de montage du manipulateur

En plus de ces conditions, des contraintes supplémentaires sont imposées par l'énoncé et sont posées comme suit :

- **Espace de travail** : L'espace de travail du robot est déterminé par son rayon R_W et par sa liberté de rotation d'angle $\Delta\phi$. Si on se place dans un espace dont les axes sont dirigés par les coordonnées (x_P, y_P, ϕ) du point P et avec comme origine le point O , l'espace de travail est alors représenté par un cylindre de rayon R_W et de hauteur $\Delta\phi$. On impose dans ce problème $R_W \geq 100$ mm et $\Delta\phi \geq 60^\circ$.
- **Conditionnement de la matrice jacobienne** : Le conditionnement inverse de la matrice jacobienne du robot, ainsi que celui de son inverse, doit être supérieur à 0.1 en tout point de l'espace de travail imposé.
- **Rigidité** : Pour une force de 10 Newton appliquée au point P selon l'axe normal au plan de son déplacement, la plate-forme doit se déplacer d'au plus 0.1 mm.

Modèles géométriques et cinématiques

Pour résoudre le problème d'optimisation, il est nécessaire de connaître les modèles géométriques et dynamiques du manipulateur étudié.

Modèle géométrique direct

Le calcul du modèle géométrique direct pour des manipulateurs planaires à trois degrés de liberté a été étudié, par exemple dans [3]. Il a été montré que calculer le modèle géométrique direct peut être ramené à la recherche des racines d'un polynôme de degré 6. Cependant, nous verrons que l'on peut se contenter ici de connaître le modèle géométrique inverse du manipulateur pour résoudre le problème d'optimisation.

Modèle géométrique inverse

Si on considère les coordonnées (x_P, y_P, ϕ) du point P , on peut retrouver les paramètres ρ_1, ρ_2, ρ_3 qui sont les longueurs des barres actionnées $A_i B_i$. Pour cela, on introduit la notation β_i pour définir les angles formés entre les droites portées par $[B_i C_i]$ et celles portées par $[A_i A_{i+1}]$.

On se place dans le repère \mathcal{R}^0 centré en A_1 et ayant pour axes la droite portée par $[A_1 A_2]$ et sa perpendiculaire verticale orientée vers le haut. Dans ce repère, on peut écrire les relations suivantes sur

les coordonnées de C_1 , notées (x_{c1}, y_{c1}) :

$$\begin{cases} x_{c1} = \rho_1 + L_b \cos \beta_1 \\ y_{c1} = L_b \sin \beta_1 \end{cases} \quad (3)$$

d'une part, et

$$\begin{cases} x_{c1} = x_P + r \cos(\phi + \frac{7\pi}{6}) \\ y_{c1} = y_P + r \sin(\phi + \frac{7\pi}{6}) \end{cases} \quad (4)$$

d'autre part. On peut ainsi en déduire β_1 et surtout ρ_1 . En procédant de façon analogue, on établit que

$$\begin{cases} x_{c2} = \sqrt{3}R - \frac{1}{2}\rho_2 + L_b \cos(\beta_2 + \frac{2\pi}{3}) \\ y_{c2} = \frac{\sqrt{3}}{2}\rho_2 + L_b \sin(\beta_2 + \frac{2\pi}{3}) \\ x_{c3} = \frac{\sqrt{3}}{2}R - \frac{1}{2}\rho_3 + L_b \cos(\beta_3 + \frac{4\pi}{3}) \\ y_{c3} = \frac{3}{2}R - \frac{\sqrt{3}}{2}\rho_3 + L_b \sin(\beta_3 + \frac{4\pi}{3}) \end{cases} \quad (5)$$

$$\begin{cases} x_{c2} = x_P + r \cos(\phi - \frac{\pi}{6}) \\ y_{c2} = y_P + r \sin(\phi - \frac{\pi}{6}) \\ x_{c3} = x_P + r \cos(\phi + \frac{\pi}{2}) \\ y_{c3} = y_P + r \sin(\phi + \frac{\pi}{2}) \end{cases} \quad (6)$$

Ces équations nous donnent alors accès au modèle géométrique indirect du manipulateur.

Modèle cinématique direct

Le calcul du modèle cinématique direct est détaillé dans [2], et il en résulte :

$$\mathbf{J} = \mathbf{B}^{-1} \mathbf{A} \quad (7)$$

avec

$$\mathbf{A} = \begin{bmatrix} b_1^T & r b_1^T E e_1 \\ b_2^T & r b_2^T E e_2 \\ b_3^T & r b_3^T E e_3 \end{bmatrix}$$

et

$$\mathbf{B} = \begin{bmatrix} b_1^T a_1 & 0 & 0 \\ 0 & b_2^T a_2 & 0 \\ 0 & 0 & b_3^T a_3 \end{bmatrix}$$

Hypothèses sur la rigidité

On cherche ici à satisfaire la contrainte suivante de l'énoncé :

“For any force of magnitude 10 N applied on the moving-platform in the plane of motion, the point-displacement of the moving-platform should be lower than 0.1 mm throughout the regular workspace.”

On décide, pour étudier la rigidité du manipulateur planaire, d'émettre une hypothèse simplificatrice sur les bras et les articulations. Il est possible de modéliser les raideurs des articulations, des bras et de la plate-forme par des ressorts virtuels à 1 et 6 degrés de liberté respectivement, ce qui permet d'analyser la rigidité du mécanisme quand celui-ci est soumis à des torseurs en trois dimensions ([2]). Cependant, dans le problème considéré, la masse du manipulateur n'est pas la mesure que l'on cherche *a priori* à minimiser. Néanmoins, il est intéressant de considérer au moins la raideur des bras $B_i C_i$ pour faire apparaître les sections de ceux-ci comme donnée géométrique du problème.

On suppose ainsi que les barres actionnées par les liaisons prismatiques possèdent une raideur infinie quand elles sont commandées, de même que la plate-forme mobile. On simplifie donc le problème pour s'intéresser uniquement aux raideurs des trois barres intermédiaires.

On se place dans le cas où une force de norme F est appliquée au point P dans une direction portée par un vecteur appartenant au plan de mouvement du robot. On peut associer la déformation des trois barres intermédiaires à celle de trois poutres de type "encastrée-libre", et la matrice de raideur du mécanisme peut être écrite sous la forme :

$$\mathbf{K} = \sum_{i=1}^3 \mathbf{K}_i^0 \quad (8)$$

Avec chaque \mathbf{K}_i^0 la matrice de raideur d'une poutre encastrée-libre exprimée dans le repère utilisé précédemment. On ne tient compte ici que des efforts en déplacement dans le plan de déplacement du robot, de telle sorte que chaque matrice k_i peut s'écrire, dans un repère où l'axe des abscisses est orienté selon la longueur de la barre :

$$\mathbf{K}_i = \begin{bmatrix} k_c & 0 \\ 0 & k_f \end{bmatrix} \quad (9)$$

k_c et k_f sont les raideurs des trois barres en compression et en flexion respectivement, de telle sorte que

$$k_c = \frac{a^2 E}{L_b}$$

$$k_f = \frac{3EI}{L_b^3} = \frac{a^4 E}{4L_b^3}$$

Avec E le module de Young du matériau choisi pour les barres, a le côté de leur section choisie carrée, et I leur moment d'inertie qu'on a remplacé par son expression en fonction de a .

On cherche maintenant à exprimer les matrices \mathbf{K}_i dans le repère R^0 , ce qu'on peut faire avec la relation suivante :

$$\mathbf{K}_i^0 = \begin{bmatrix} \cos \beta_i & \sin \beta_i \\ -\sin \beta_i & \cos \beta_i \end{bmatrix} \mathbf{K}_i \quad (10)$$

Connaissant ainsi la matrice de raideur \mathbf{K} , on peut exprimer la condition de rigidité en partant de la relation entre une force F exercée au point P et le déplacement associé $\delta t = [\delta x \quad \delta y]^T$:

$$\mathbf{K} \delta t = F$$

L'énoncé impose que le déplacement engendré par une force de norme 10N soit inférieur à 0.1mm. Après application de l'inégalité de Cauchy-Scharwz, cela revient simplement à l'inégalité :

$$\|\mathbf{K}\| > 10^5 \quad (11)$$

En imposant une rigidité minimum pour le robot, le côté de la section des barres intermédiaires a devient un paramètre supplémentaire du problème. Cependant, puisque a n'est contraint que par une seule inégalité (Formule 11), on peut en déduire sa valeur optimale sans en faire un paramètre de l'algorithme d'optimisation. En particulier, pour que l'inégalité soit vraie pour toutes les valeurs de β_i , on en déduit que a doit vérifier

$$a^8 + 16L_b^4 a^4 > \frac{16L_b^4}{9E^2} \cdot 10^{10}$$

Contrainte de conditionnement

Intéressons-nous maintenant à la condition imposée sur la dextérité du mécanisme:

"The inverse condition number of the normalized kinematic Jacobian matrix should be larger than 0.1 throughout the regular workspace."

Faisons remarquer que le robot est isotrope quand il existe un réel λ tel que $\mathbf{J} \mathbf{J}^T = \lambda I_2$. On peut se servir de cette information pour notre optimisation en choisissant pour les paramètres des valeurs qui tendent à faire approcher ce rapport.

Le problème est ici que la matrice jacobienne n'est pas normalisée puisqu'elle mélange des grandeurs de différentes unités ($\text{m} \cdot \text{s}^{-1}$ et $\text{rad} \cdot \text{s}^{-1}$). Pour parler de conditionnement, il faudrait donc utiliser des coefficients pour normaliser \mathbf{J} , reste alors à choisir les bonnes valeurs pour ces coefficients

D'autre part, le λ de la formule précédente correspond au conditionnement de \mathbf{J} , autrement dit au

quotient de $|\sigma_{max}|$ sur $|\sigma_{min}|$, avec σ_{max} et σ_{min} respectivement la plus grande et la plus petite valeur propre de **J** en valeur absolue. L'énoncé demandant un conditionnement inverse supérieur à 0.1, on a donc la condition suivante à respecter :

$$\frac{|\sigma_{max}|}{|\sigma_{min}|} < 10 \quad (12)$$

Récapitulatif du problème d'optimisation

Maintenant que la plupart des contraintes de l'énoncé ont été exprimées sous forme mathématique, on peut récapituler le problème d'optimisation.

- Les variables du problème sont r , R , L_b et a .
- la fonction à minimiser est :

$$f : (r, R, L_b, a) \mapsto R \quad (13)$$

- Les inéquations à respecter sont :

$$\begin{aligned} R &\leq 2(L_b + r) \\ R_W &\geq 100 \text{ mm} \\ \Delta\phi &\geq 60^\circ \\ a^6 + 16L_b^4 a^4 &> \frac{16L_b^6}{9E^2} \cdot 10^{10} \\ \frac{|\sigma_{max}|}{|\sigma_{min}|} &< 10 \end{aligned} \quad (14)$$

Dans ce qui suit, nous allons discuter de la méthode de résolution de ce problème de façon numérique à l'aide de Matlab et de sa *toolbox* d'optimisation non-linéaire.

Recherche d'une solution sur Matlab

Nous choisissons pour ce problème d'optimisation qui consiste en la minimisation d'une fonction en respectant des contraintes en inégalités, l'utilisation de l'outil **fmincon** sur Matlab. Le problème faisant intervenir des contraintes non-linéaires, la difficulté de la résolution du programme réside dans la fonction **nlincond**, qui retourne pour une valeur donnée des variables du problème, le caractère respecté ou non des inégalités.

Une des premières opérations à effectuer est de calculer, pour une configuration donnée, la valeur

minimale de a correspondante. Pour cela, il suffit d'écrire au début de la fonction **nlincond.m** :

```
syms x;
a = vpasolve(x^8 + 16*Lb^4*x^4 ==
16*Lb^6*10^10/(9*E^2), x, [0 1]);
```

Listing 1: Détermination de la valeur minimum de a dans la fonction **nlincond**

Remarque. On aura pris soin de donner une valeur pour E au préalable. En choisissant l'acier comme matériau pour le robot, on obtient pour des barres intermédiaires de longueur 50 cm une épaisseur minimale théorique de 0.3 mm environ, ce qui implique que la raideur du manipulateur n'est pas contraignante dans la conception du mécanisme avec les simplifications faites. Si la masse devient un facteur important du problème, il convient d'opter pour un modèle moins simplificateur dans le calcul de la raideur du robot.

Une autre opération à calculer en début de fonction est la vérification de la première inégalité de (14), ce qui permet de mettre un terme à l'exécution de la fonction si possible.

Ensuite, il convient de vérifier la validation des conditions sur l'espace de travail. Pour cela, on sélectionne n points sur le périmètre de l'espace de travail désiré. Les coordonnées dans le repère R^0 de ces points sont de la forme

$$\left(\frac{R\sqrt{3}}{2} + R_W \cos \frac{k\pi}{n}, \frac{R}{2} + R_W \sin \frac{k\pi}{n} \right), 0 \leq k < n$$

Puisqu'on désire des valeurs de ϕ dans un intervalle de 60 degrés, on teste pour chacune de ces positions des valeurs de ϕ égales à $-\frac{\pi}{6}$, 0 et $\frac{\pi}{6}$. Cela fait donc $3n$ configurations à tester de par le modèle inverse: si celui-ci ne trouve pas de solution admissible (c'est-à-dire vérifiant, pour i compris entre 1 et 3, $0 \leq \rho_i \leq \sqrt{3}R$) pour une de ces configurations, les variables du problème sont mal choisies et on peut leur assigner de nouvelles valeurs.

Si les conditions sur l'espace de travail sont réunies, il reste à vérifier le conditionnement de la matrice jacobienne, ce qui peut se faire simplement par l'appel de la fonction **eig** qui renvoie l'ensemble des valeurs propres de la matrice passée en argument.

Remarque. Comme dit précédemment, la contrainte sur le conditionnement n'est utile que si la matrice jacobienne est correctement normalisée. Nous considérerons ici qu'un conditionnement inférieur à 200 pour la matrice jacobienne construite à partir de

$$\mathbf{A} = \begin{bmatrix} b_1^T & b_1^T E e_1 \\ b_2^T & b_2^T E e_2 \\ b_3^T & b_3^T E e_3 \end{bmatrix}$$

où l'expression de \mathbf{A} remplace celle utilisée en (7)

L'exécution de l'algorithme ne trouve pas de solution pour $\Delta\phi = 60^\circ$. Il en est de même pour 45 et 35 degrés. Pour $\Delta\phi = 30^\circ$, nous trouvons la solution suivante :

$$Lb = 0.3300$$

$$R = 0.5955$$

$$r = 0.2183$$

Pistes d'amélioration de la recherche de solution

Le problème étudié ne tient compte en aucune façon de la masse du robot, qui est souvent un facteur déterminant dans le choix d'un design particulier de robot puisqu'elle contraint le dimensionnement des moteurs, l'épaisseur des barres ou le choix des matériaux, entre autres. On peut donc songer à faire entrer en jeu la masse m du robot dans le problème, ce qui peut se faire de plusieurs façons :

- Limiter la masse de façon brute en rajoutant à (14) une inégalité définissant une masse limite souhaitée, sous la forme $m \leq m_{max}$. Cela a pour avantage de très peu changer le programme déjà écrit précédemment, au prix d'un résultat probablement perfectible. En effet, on peut imaginer l'existence d'une solution avec une valeur de m encore plus faible satisfaisant toutes les conditions, mais n'étant pas trouvée par le programme. A l'inverse, il est possible qu'aucune solution ne soit trouvée car la minimisation de f n'est pas équivalente à une minimisation de la masse.
- Changer la fonction à minimiser en y faisant intervenir la masse. Par exemple, on peut imaginer une fonction f de la forme

$$f : (r, R, L_b, a) \mapsto N_R \cdot \frac{R}{W_R} + N_m \cdot \frac{m}{W_m} \quad (15)$$

avec N_R et N_m des facteurs de normalisation, et W_R et W_m des facteurs de pondération. Cela permet d'une part de toujours proposer des solutions minimisant à la fois la taille de l'espace occupé par le robot et la masse de celui-ci, et d'autre part de choisir exactement à quel point la minimisation de l'un est privilégiée sur la minimisation de l'autre, grâce aux facteurs de

normalisation et de pondération. Cependant, cela ne permet pas de réaliser l'optimisation des deux paramètres simultanément, seulement celle d'une de leurs combinaisons.

- Changer l'optimisation pour optimiser deux fonctions au lieu d'une (optimisation à objectifs multiples). Cela requiert l'utilisation d'un algorithme différent, comme modeFrontier [4], qui est utilisé dans [2]. Ainsi, on peut obtenir une vision globale de l'optimisation simultanée des deux paramètres.

Une autre piste d'amélioration de l'algorithme est dans le calcul de la matrice jacobienne et de son conditionnement. En effet, ces calculs reviennent (en termes de complexité calculatoire) à la multiplication de matrices entre elles. Ces opérations peuvent être optimisées par des algorithmes comme celui présenté par M. Bläzer dans [5]. Il y prouve qu'on peut obtenir une borne inférieure de 19 opérations pour le produit de deux matrices 3×3 . Pour des problèmes impliquant des matrices jacobiennes de grande taille, il existe d'autres algorithmes optimisant la complexité de la multiplication, comme celui proposé par A. Davie et A. Stothers dans [6].

References

- [1] Merlet, J. (2006). *Parallel robots*. Dordrecht, Netherlands: Springer.
- [2] Caro, S., Chablat, D., Ur-Rehman, R., & Wenger, P. (2011). *Multiobjective Design Optimization of 3-PRR Planar Parallel Manipulators*. Global Product Development, 373-383. doi:10.1007/978-3-642-15973-2_37
- [3] Gosselin, C. M., & Merlet, J. (1994). *The direct kinematics of planar parallel manipulators: Special architectures and number of solutions*. Mechanism and Machine Theory, 29(8), 1083-1097. doi:10.1016/0094-114x(94)90001-9
- [4] ESTECO (2008) Modefrontier, version 4.0.3.
- [5] Bläser, M. (2003). *On the complexity of the multiplication of matrices of small formats*. Journal of Complexity, Volume 19, Issue 1, 43-60. doi:10.1016/S0885-064X(02)00007-9.
- [6] Davie, A., & Stothers, A. (2013). *Improved bound for complexity of matrix multiplication*. Proceedings of the Royal Society of Edinburgh: Section A Mathematics, 143(2), 351-369. doi:10.1017/S0308210511001648