



**HAL**  
open science

# Positionnement d'un robot pour le dévissage automatisé grâce à des marqueurs ArUco : évaluation de la robustesse d'un système faible coût

C Devos, H Dauvilaire, S Benshir Caloustian, Pierre-Thomas Dautre

## ► To cite this version:

C Devos, H Dauvilaire, S Benshir Caloustian, Pierre-Thomas Dautre. Positionnement d'un robot pour le dévissage automatisé grâce à des marqueurs ArUco : évaluation de la robustesse d'un système faible coût. *Rechercher et enseignement Agile pour une industrie soutenable*, S.mart, May 2025, Vogüé, France. ⟨hal-05114089v1⟩

**HAL Id: hal-05114089**

**<https://hal.science/hal-05114089v1>**

Submitted on 16 Jun 2025 (v1), last revised 27 Feb 2026 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Positionnement d'un robot pour le dévissage automatisé grâce à des marqueurs ArUco : évaluation de la robustesse d'un système faible coût

C.DEVOS<sup>a</sup>, H.DAUVILAIRE<sup>b</sup>, S. BENSRRHIR, P-T. DOUTRE

<sup>a</sup> S.smart G.A.

[camille.devos@grenoble-inp.fr](mailto:camille.devos@grenoble-inp.fr)

## Introduction

L'automatisation des tâches industrielles par des robots s'est développée ces dernières années, permettant de remplacer l'humain dans des opérations pénibles, d'améliorer la sécurité des opérateurs et d'augmenter l'efficacité des chaînes de production. Parmi les tâches industrielles les plus fréquemment automatisées figurent l'assemblage et le désassemblage de pièces ou de systèmes mécaniques. Comme mentionné par l'étude [1], le dévissage représente environ **40 %** des opérations de désassemblage dans l'industrie. Ces opérations sont donc des candidats pertinents pour l'automatisation. Pour réaliser du dévissage automatisé, des robots industriels sont généralement utilisés. Toutefois, cette tâche reste complexe et nécessite répétabilité, précision et adaptabilité. L'émergence de nouveaux capteurs a permis aux robots d'acquérir des capacités de perception avancées, améliorant leur interaction avec l'environnement et les objets à manipuler.

Bien que ces nouveaux capteurs facilitent l'estimation de la position et de l'orientation d'objets dans un espace 3D, cette tâche reste un défi, en particulier pour les tâches d'assemblage [2] ou de désassemblage de pièces par des robots. Des travaux récents [3] montrent que l'automatisation du dévissage de pièces nécessite des techniques avancées de contrôle robotique afin de réaliser un positionnement précis. Des solutions clés en main existent déjà, principalement basées sur des systèmes de vision industrielle. Toutefois, elles sont souvent propriétaires et coûteuses, limitant leur accessibilité, leur intégration et leur compatibilité avec des systèmes robotiques génériques.

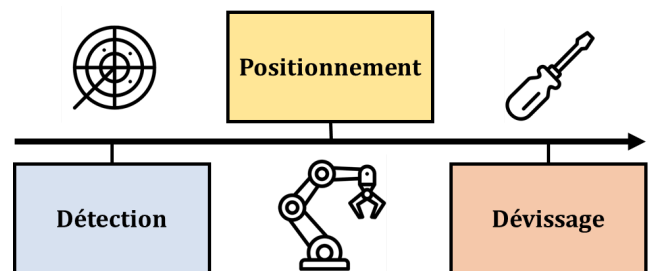
L'objectif de cet article est de proposer une méthode de détection d'objet et de positionnement d'un bras robot dans une optique de dévissage automatisé, en utilisant une solution de vision industrielle à faible coût. Ce travail s'inscrit dans un contexte de récupération d'un boîtier contenant des informations de traçabilité. Ce boîtier est initialement vissé par un opérateur sur la culée d'un tronc destiné à une exploitation en scierie. Le

dévissage du boîtier se fait donc dans un contexte où la position et l'orientation de la vis ne sont pas maîtrisées et inconnues. Cette variabilité rend la détection de la cible particulièrement critique, ce qui entraîne une exigence forte sur le développement d'une solution fiable.

Dans la première partie, nous présenterons une étude de l'état de l'art sur les approches de détection d'objet dans un espace 3D, les capteurs et algorithmes, ainsi que les solutions de dévissage et les contraintes et exigences associées. Nous détaillerons en deuxième partie les choix techniques retenus pour la conception du système proposé. La troisième section sera consacrée à l'évaluation de la robustesse du système. Enfin, une analyse approfondie des résultats permettra de déterminer la pertinence de notre approche avant de conclure sur son adaptabilité pour des applications pratiques de dévissage.

## Etat de l'art

L'automatisation des tâches de dévissage peut être décomposées en trois étapes clés, comme illustré en *figure 1* : Détection de la cible, positionnement et dévissage.



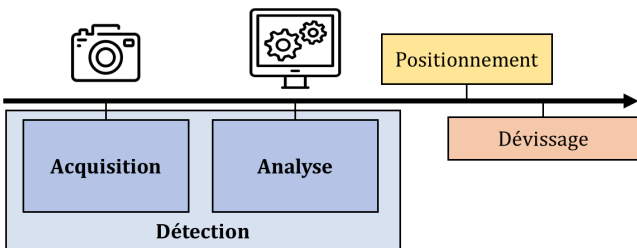
**Fig. 1:** Gamme de dévissage automatisé

L'automatisation des processus de vissage est couramment rencontrée dans l'industrie, en particulier dans des secteurs comme l'automobile, où elle est mise en œuvre depuis longtemps. L'étude [4], portant sur l'automatisation des opérations d'assemblage dans ce domaine, indique que déjà à l'époque, **50 %** des opérations de vissage étaient automatisées. Pour ce faire, des têtes de vissage spécialisées, telles que celles

adaptées aux têtes hexagonales, sont employées afin de faciliter l'engagement de l'outil sur la vis. En revanche, l'intégration du dévissage automatisé dans des applications industrielles reste plus rare. Contrairement au vissage, il n'existe pas de solutions standardisées pour le guidage du dévissage. Les têtes de dévissage doivent s'adapter à la vis selon son orientation, ce qui rend le processus plus complexe. Tandis qu'il est relativement simple de concevoir un avaloir pour une tête de type "mâle", guider un outil dans une tête de type "femelle" présente davantage de difficultés techniques.

Les solutions existantes pour le dévissage automatisé sont moins répandues et souvent spécifiques à des applications de haute précision. Un seul article [3] évoque le dévissage automatisé, mais il se limite à une configuration fixe, ce qui ne permet pas de traiter des scénarios avec des objets à orientation et position variable. D'après notre étude bibliographique, aucune étude n'a été publiée sur l'angle limite de positionnement d'un outil de vissage ou de dévissage par rapport à une vis. Toutefois, les recommandations des fabricants d'outillage stipulent que l'outil peut être orienté jusqu'à un maximum de  $5^\circ$  pour des têtes complexes, telles que celles de type Torx, et  $20^\circ$  pour des têtes plus classiques avant de risquer un phénomène de "cam-out". Une mauvaise inclinaison peut entraîner également une usure prématurée de la vis ou un transfert inefficace du couple de serrage.

Le positionnement de l'outil de dévissage est donc particulièrement critique. Dans le cadre d'un processus automatisé, il est donc nécessaire que l'étape préliminaire de **détection** de la cible soit correctement réalisée et suffisamment fiable. Comme illustrée en *figure 2*, cette phase de détection peut être décrite comme la suite de deux étapes : **l'acquisition** des données à l'aide de moyens physiques (capteurs), et **l'analyse** de ces données pour les **convertir** en commande pour le bras robotisé.



**Fig. 2:** modélisation de la détection

Pour chacune des ces étapes intervenant dans la détection, des solutions techniques ont été développées au cours des dernières années. Celles-ci sont décrites dans la partie suivante, où nous identifions également les avantages et les limites associés à chaque produit ou service présenté.

Certaines solutions commerciales clés en main intègrent à la fois le matériel nécessaire à l'acquisition des données et le logiciel utilisé pour leur traitement. Parmi les technologies disponibles, les solutions propriétaires comme celles de **Sick**, **Keyence** ou **Cognex** proposent des systèmes qui incluent à la fois le matériel et le logiciel. Ces solutions sont très conviviales, mais

elles sont fermées, rigides et peuvent être très coûteuses. De plus, une formation est généralement nécessaire pour utiliser le plein potentiel de ces solutions. Leur principal avantage réside dans leur prise en main rapide, mais leur manque de flexibilité et leur coût élevé les rendent difficiles d'accès pour de nombreuses applications. Une alternative consiste alors à concevoir un système personnalisé en sélectionnant et en intégrant soi-même les technologies d'acquisition et de traitement adaptées aux besoins spécifiques.

L'étape d'acquisition de l'image nécessite la mise en place de moyens matériels. Le choix du capteur utilisé influence directement la robustesse et l'exactitude des mesures. Les **caméras monoculaires** [5], bien que peu coûteuses et largement disponibles, présentent des limites en termes de perception de profondeur. Les **caméras stéréoscopiques** [6] offrent quant à elles une perception de profondeur plus précise en exploitant la disparité entre deux images, mais demeurent cependant sensibles aux variations d'éclairage et aux surfaces peu texturées. De leur côté, les capteurs **LiDAR** [7] fournissent une mesure directe de la distance en émettant des impulsions laser, ils souffrent toutefois d'une résolution limitée et d'un coût élevé, ce qui peut restreindre leur adoption dans des applications à faible budget. Ainsi, le choix entre ces technologies dépend du compromis entre coût, précision et robustesse face aux conditions environnementales. De manière générale, ces appareils ont pour principal avantage leur capacité à être intégrés facilement dans différentes méthodes de détection en fonction de besoins spécifiques, bien qu'elles exigent des compétences techniques avancées pour leur configuration et leur intégration.

Une fois l'acquisition de l'image réalisée, celle-ci doit être analysée pour en extraire les données nécessaires au positionnement (coordonnées et orientation de la cible). La vision industrielle classique, utilisant des outils des opérations de traitement d'images, est une méthode maîtrisée depuis longtemps pour la détection d'objets. Elle présente l'avantage d'être ouverte et flexible, et offre des vitesses de détection record. Cependant, elle nécessite des compétences techniques pointues dans le domaine pour une mise en œuvre efficace, et ses performances peuvent rapidement atteindre leurs limites lorsque l'on se confronte à des environnements complexes ou à des tâches plus exigeantes.

Une autre solution prometteuse est l'utilisation de **l'intelligence artificielle** (IA). Les réseaux neuronaux et les modèles d'apprentissage automatique offrent une grande puissance de détection et de classification comme démontré dans les études [8], [9] mais ils présentent des inconvénients majeurs, notamment un coût computationnel élevé, rendant leur utilisation difficile sur des systèmes à ressources limitées.

Une autre approche consiste à exploiter des **marqueurs de repères** (fiducial markers) imprimés et installés au préalable sur la cible à détecter. Les algorithmes de traitement liés à l'utilisation de tels marqueurs sont souvent déjà intégrés dans des bibliothèques **open source** comme **OpenCV**. L'utilisation de marqueurs Aruco (*figure 3*) est donc une option

populaire dans le monde de la robotique pour la détection d'objets.

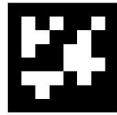


Fig. 3. Exemple de marqueurs ArUco

Le caractère open-source des ces bibliothèques au sein d'une communauté active offre une multitude de cas d'applications et d'exemples de code. Ce type de marqueur est remarquable pour son faible coût, son accessibilité, et sa précision dans des conditions contrôlées. Toutefois, l'étude [10] montre que la détection de ces marqueurs peut être corrompue si l'environnement n'est pas suffisamment éclairé ou en cas d'occlusion partielle de ceux-ci.

Une solution simple et économique consiste à utiliser une **caméra monoculaire** associée à des marqueurs **ArUco**, constitue une solution pertinente dans notre contexte, où un seul objet est à démonter. L'état de l'art révèle également que l'automatisation du dévissage reste un domaine largement sous-exploré. Cette lacune dans la littérature motive l'intérêt de notre étude. Ainsi, la problématique suivante émerge :

- Comment un robot peut-il détecter et se positionner avec précision pour réaliser une opération de dévissage ?
- Comment exploiter des technologies open-source et accessibles pour proposer une alternative faible coût aux solutions industrielles propriétaires ?

Cet article explore cette problématique en présentant une solution combinant vision industrielle à faible coût et un système de positionnement robotique simple à implémenter. Ce système aura pour objectif de réaliser des opérations de dévissage sur des objets dont le positionnement et l'orientation est variable.

## Proposition

### 1- Moyens mis en oeuvre

La méthodologie proposée dans la suite de l'article repose sur l'utilisation des moyens suivants :

- Le système d'automatisation utilisé est un **bras robot UR16e** du constructeur Universal Robots. Ce modèle 6 axes, de charge utile 16kgs, est réputé pour sa légèreté et sa polyvalence. L'utilisation de ce modèle est imposée par les moyens disponibles au laboratoire au moment où la solution a été développée, mais il avait été sélectionné au préalable pour la facilité d'accès à ses données via des protocoles de communication standards et ouverts.
- Nous utilisons un **marqueur ArUco** de type 6x6 avec des dimensions de 3x3 cm, imprimé en noir et blanc et fixé au boîtier cible.
- L'acquisition des images est réalisée à l'aide d'une **caméra Intel RealSense**, sélectionnée pour son bon rapport qualité-prix et sa polyvalence. Cette caméra est montée directement à l'extrémité du bras du robot.

## 2- Méthode de détection et de positionnement.

Nous proposons de structurer notre méthode en deux grandes étapes, la détection et le positionnement (fig 3). Après le déploiement du robot en position d'acquisition, l'image capturée est transférée vers l'ordinateur. Une phase de traitement de l'image et d'identification permet d'obtenir les coordonnées du marqueur. Les coordonnées du marqueur et les coordonnées de l'outil servent à alimenter une phase de conversion, qui génère une commande en langage UR Script. Cette commande est envoyée au robot via une connexion socket, et entraîne le positionnement du bras.

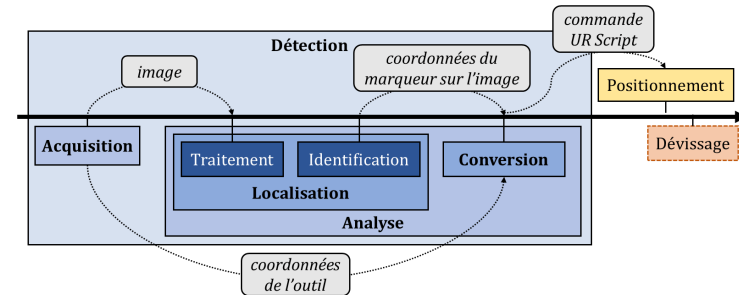


Fig. 3: modélisation de la détection

Lors de la phase d'**acquisition**, le bras robot se déploie en position "photo", qui a été prédéfinie au préalable. La caméra montée sur un support proche de l'outil du robot prend une photo, qui est transférée à l'ordinateur via port USB. Grâce à une fonction du robot, on récupère la position actuelle de l'outil du robot sous la forme d'un tableau (1):

$$PositionOutil = (Tx, Ty, Tz, Rx, Ry, Rz)_{base} \quad (1)$$

$(Tx, Ty, Tz)$  représente le vecteur de translation et  $(Rx, Ry, Rz)$  le vecteur de rotation de l'outil, le tout par rapport à la base du robot (origine du repère).

La phase de **localisation** de la cible sur l'image est prise en charge sous python. Les deux étapes de traitement et d'identification sont réalisées grâce aux fonctions Aruco intégrées à la bibliothèque OpenCV (`cv2.aruco.detectMarkers`, `cv2.aruco.estimatePoseSingleMarkers`). On en extrait les coordonnées du marqueur aruco dans l'image, qui correspondent aux coordonnées du boîtier dans le repère de l'outil. La [figure 5] représente la position des différents repères.

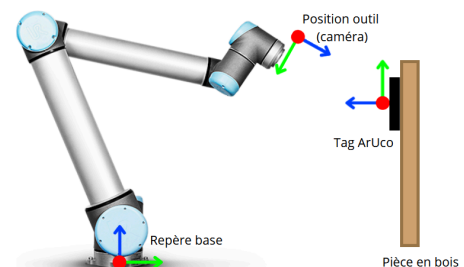


Fig. 5. Position du tag par rapport à l'outil (caméra)

Pour déplacer l'outil du robot au bon endroit, nous avons besoin d'obtenir la position du marqueur par rapport à la base du robot. Pour ce faire, la phase de **conversion** permet de transformer les deux vecteurs de

translation et de rotation en une matrice de transformation pour les deux positions (l'outil/base et marqueur/outil). Pour faire le produit matriciel, la matrice de transformation doit être de la forme (2).

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \quad (2)$$

**R** : matrice 3x3 de rotation obtenue grâce à la formule de Rodrigues et le vecteur de rotation.

**t** : vecteur de translation

Si on développe **R** on obtient la forme (3) :

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

**R<sub>ij</sub>** : Composants de la matrice de rotation 3x3, définissant l'orientation.

**T<sub>x</sub>, T<sub>y</sub>, T<sub>z</sub>** : Coordonnées de translation (position).

Une fois ces transformations réalisées, on effectue le produit des deux matrices 4x4 à l'aide d'une fonction de la bibliothèque *numpy* appelée *dot*. Ainsi on peut exprimer la position du tag par rapport à la base du robot et déplacer l'outil au centre du tag.

Cette position est transférée au bras robot sous forme de code UR Script via une connexion socket. Lors de ce premier **positionnement**, la tête outil du robot est déplacée dans le plan **x y** (par rapport à la caméra) afin **d'aligner le marqueur au centre de l'image**, comme illustré en figure 6.

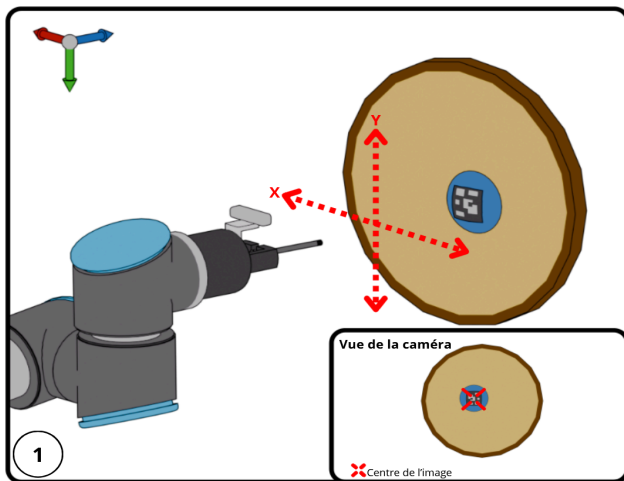


fig 6 : alignement de la caméra avec le tag

S'ensuit, un déplacement relatif en z (outil), vers le tag de 200 mm. Une nouvelle acquisition d'image est alors effectuée afin d'avoir la position absolue du tag par rapport à la base du robot. Un nouveau **positionnement** est alors réalisé pour aligner la caméra avec le marqueur, cette fois en tenant compte de son orientation (Figure 7).

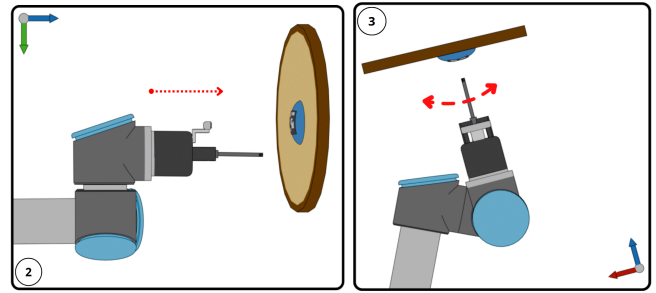


fig 7 : Approche, deuxième capture et orientation

Cependant, l'alignement de la caméra ne constitue pas l'objectif final. Ainsi, dans une dernière étape (Figure 8), un **déplacement relatif** est effectué afin d'aligner l'outil du robot avec le marqueur, en se basant sur la position actuelle du robot.

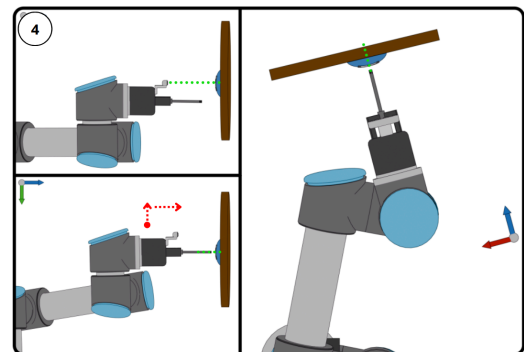


fig 8 : Déplacement relatif

## Protocole d'évaluation

### 1- Moyens mis en oeuvre

Pour chaque mesure, nous utilisons le dispositif illustré en Figure 9, composé de deux caméras montées perpendiculairement sur un support mécanique en forme d'équerre, aligné avec les axes **x** et **y**. Ces caméras sont fixées à une distance de nnn mm du marqueur.

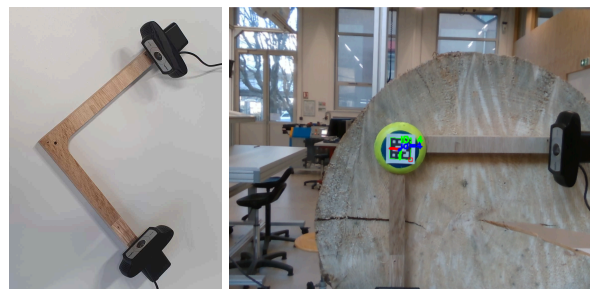


Fig. 9. Système pour mesure du positionnement de l'outil par rapport au tag

### 2- Protocole

Un total de **100 essais** ont été réalisés, au cours desquels les écarts de translation et de rotation entre la position finale de l'outil du robot et la position réelle de la pièce seront mesurés de façon graphique. A chaque essai, la position du tag ainsi que l'orientation du support sur lequel il est fixé variera de  $\pm 10^\circ$  selon **x** et **y**. Deux photos sont prises à chaque essai (une par caméra). On peut voir un exemple d'acquisition dans la figure 10.

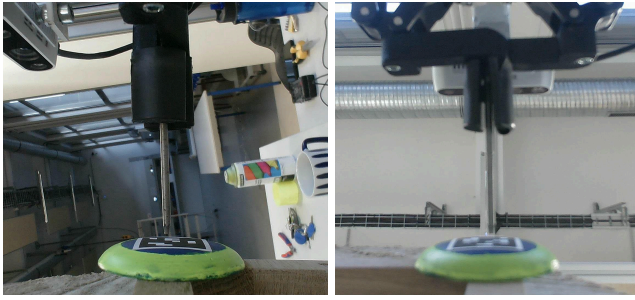


Fig. 10. Exemple d'acquisition selon x et y pour un essai

Pour chaque image, nous effectuons les mesures avec le logiciel GeoGebra comme sur la figure 11. Une droite a été tracée pour représenter l'axe du tag et une autre pour l'orientation de l'outil. L'angle entre ces deux droites a été mesuré afin de quantifier l'orientation de l'outil par rapport à la normale du tag. L'écart en translation a été calculé en projetant la position du bout de l'outil sur l'axe du tag.



Fig. 11. Mesure pour une acquisition

Tous les résultats seront enregistrés dans un logiciel de type tableur. Grâce à ces données, nous pourrions déterminer les écart-type et les valeurs moyennes pour les mesures de translation et de rotation.

## Résultats

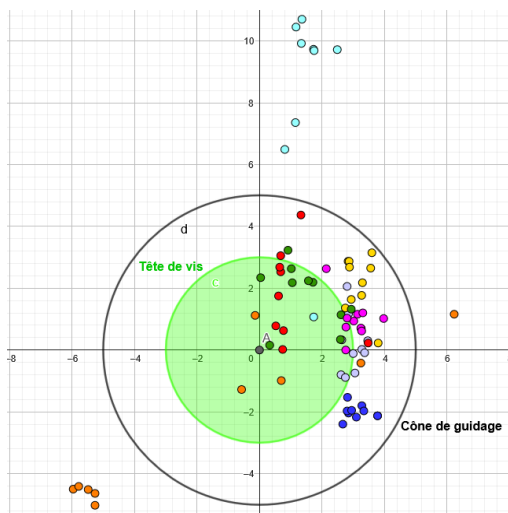


Fig. 12. Écart mesuré en positionnement (mm)

La figure 12 présente un nuage de points illustrant la position de l'outil par rapport au marqueur (tag) après chaque essai. Chaque point correspond à une mesure individuelle de cette position. Deux cercles concentriques sont superposés au centre. Le cercle vert

délimite la surface occupée par la vis. Le second cercle, plus large, matérialise le cône de guidage intégré à l'objet à dévisser.

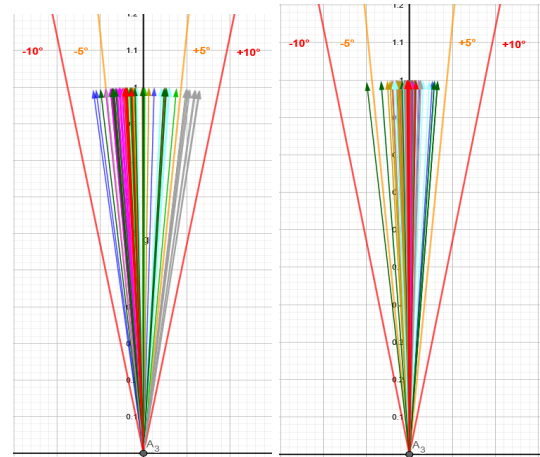


Fig. 13. Variation en orientation de l'outil par rapport au marqueur selon x à gauche et y à droite

Le champ de vecteur dans la figure 13 illustre l'orientation de l'outil à chaque essai. Chaque vecteur, représente l'angle d'inclinaison de l'outil par rapport au marqueur.+

## Analyse

Dans l'analyse des résultats, la précision effective du robot en positionnement sera négligée. En effet, d'après le constructeur [11], le robot a une précision de  $\pm 0.1$  mm à vide (16Kg). Les résultats obtenus, illustrés dans les figures 12 et 13, illustrent les variations de positionnement et d'orientation lors des essais. Une analyse statistique de ces erreurs permet d'évaluer la précision et la répétabilité du système testé.



## Analyse des erreurs de centrage :

- **31 %** des essais présentent un écart de centrage inférieur à  $\pm 3$  mm en x et y, ce qui correspond à un positionnement de l'embout dans la tête de vis (figure 14).
- **82 %** des essais présentent un écart de centrage inférieur à  $\pm 5$  mm en x et y, ce qui correspond à un positionnement de l'embout dans le cône de guidage (en vert sur la figure 14).
- **18 %** des essais présentent un décalage supérieur à 5 mm en x ou y, ce qui indique des cas d'échec significatif. Ces écarts semblent groupés, ce qui suggère une erreur systématique probablement liée à un défaut de déplacement relatif.

On en déduit donc que la majorité des positionnements peuvent aboutir à un dévissage. La moyenne et l'écart type sont mesurés pour les valeurs d'écarts en x et en y, et donnés dans le tableau suivant.

Paramètre	Moyenne ( $\mu$ ) [mm]	Ecart-type ( $\sigma$ ) [mm]
$\Delta x$	1.82	2.26
$\Delta y$	1.26	3.40

**Tab 1 :** Erreurs de centrage

La moyenne du décalage est de 1.82mm en x et de 1.26mm en y. La solution idéale présenterait un écart nulle, il est cependant utile de remarquer que cette erreur moyenne peut provenir d'une mauvaise calibration lors de la dernière étape du positionnement (figure 8). La valeur de l'écart type traduit une dispersion relativement faible des erreurs de positionnement (2.26mm en x et 3.40 mm en y), ce qui traduit une répétabilité plutôt satisfaisante de notre solution.

### Analyse des erreurs angulaires

- **98 %** des essais ont une erreur angulaire de rotation en y  $<$  à  $5^\circ$ .
- **87 %** des essais ont une erreur angulaire de rotation en x  $<$  à  $5^\circ$ .

Paramètre	Moyenne ( $\mu$ ) [°]	Ecart-type ( $\sigma$ ) [°]
Rotation en x	0.23	3.25
Rotation en y	1.29	2.22

**Tab 2 :** Erreurs de positionnement

Examinons nos résultats à la lumière des recommandations des fabricants d'outillage présentées dans l'état de l'art. Ces recommandations stipulent qu'un outil peut être incliné jusqu'à  $5^\circ$  pour des têtes complexes (type Torx) et  $20^\circ$  pour des têtes plus classiques avant de compromettre la transmission du couple de dévissage ou d'endommager la tête de vis.

Ces résultats suggèrent donc que le dévissage de la pièce étudiée est réalisable dans les conditions expérimentales définies, sans risque de mauvais déroulement du dévissage ou de détérioration de la vis.

### Conclusion et perspectives

Cette étude s'inscrit dans le cadre de l'automatisation des opérations de dévissage, une tâche complexe en raison des variations de position et d'orientation des objets à dévisser. L'alignement précis de l'outil sur la vis est nécessaire pour assurer une transmission optimale du couple. Une mauvaise orientation de l'outil peut entraîner un phénomène de "cam-out", provoquant son dérapage et risquant d'endommager la vis, compromettant ainsi l'efficacité de l'opération. Pour répondre à ce défi, une solution combinant l'utilisation de marqueurs ArUco, d'une caméra embarquée sur un bras robotisé et de la bibliothèque OpenCV sous Python a été développée puis évaluée. Cette approche permet d'atteindre un positionnement suffisamment précis pour répondre aux exigences du dévissage automatisé, faisant de l'exploitation de technologies open-source et de capteurs à faible coût une alternative crédible aux solutions industrielles propriétaires.

Plusieurs axes d'amélioration peuvent être envisagés pour renforcer la robustesse du système. L'utilisation de marqueurs ArUco, combinant plusieurs motifs ArUco au

sein d'une même structure, permettrait d'améliorer la précision de la détection, notamment en cas de variations d'éclairage. De plus, l'impact de la luminosité ambiante pourrait être réduit en isolant optiquement la cible des interférences extérieures ou en mettant en place un éclairage homogène et maîtrisé afin de stabiliser la détection. Il est également évident que l'ajout de marqueurs sur un grand nombre de composants à désassembler est peu envisageable, car ces marqueurs devraient être durables face au temps et aux contraintes environnementales. Enfin, l'intégration d'un système couplant vision et capteur de force dans une optique d'approche hybride similaire à l'étude[12] permettrait de corriger les écarts de positionnement liés aux imprécisions de détection.

### Références

- [1] R. Li *et al.*, « Unfastening of Hexagonal Headed Screws by a Collaborative Robot », *IEEE Trans. Autom. Sci. Eng.*, vol. 17, n° 3, p. 1455-1468, juill. 2020, doi: 10.1109/TASE.2019.2958712.
- [2] K. Samarawickrama et R. Pieters, « 6DoF assembly pose estimation dataset for robotic manipulation », *Data Brief*, vol. 56, p. 110834, oct. 2024, doi: 10.1016/j.dib.2024.110834.
- [3] D. Mironov, M. Altamirano, H. Zabihifar, A. Liviniuk, V. Liviniuk, et D. Tsetserukou, « Haptics of Screwing and Unscrewing for Its Application in Smart Factories for Disassembly », in *Haptics: Science, Technology, and Applications*, D. Prattichizzo, H. Shinoda, H. Z. Tan, E. Ruffaldi, et A. Frisoli, Éd., Cham: Springer International Publishing, 2018, p. 428-439. doi: 10.1007/978-3-319-93399-3\_37.
- [4] M. Freyssenet, « L'AUTOMATISATION DU MONTAGE AUTOMOBILE », 1994.
- [5] T.-J. Lee, D.-H. Yi, et D.-I. "Dan" Cho, « A Monocular Vision Sensor-Based Obstacle Detection Algorithm for Autonomous Robots », *Sensors*, vol. 16, n° 3, Art. n° 3, mars 2016, doi: 10.3390/s16030311.
- [6] « Stereo vision for view synthesis | IEEE Conference Publication | IEEE Xplore ». Consulté le: 11 février 2025. [En ligne]. Disponible sur: <https://ieeexplore.ieee.org/document/517171>
- [7] Y. Li *et al.*, « DeepFusion: Lidar-Camera Deep Fusion for Multi-Modal 3D Object Detection », présenté à Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, p. 17182-17191. Consulté le: 7 février 2025. [En ligne]. Disponible sur: [https://openaccess.thecvf.com/content/CVPR2022/html/Li\\_DeepFusion\\_Lidar-Camera\\_Deep\\_Fusion\\_for\\_Multi-Modal\\_3D\\_Object\\_Detection\\_CVPR\\_2022\\_paper.html](https://openaccess.thecvf.com/content/CVPR2022/html/Li_DeepFusion_Lidar-Camera_Deep_Fusion_for_Multi-Modal_3D_Object_Detection_CVPR_2022_paper.html)
- [8] Y. Xiang, T. Schmidt, V. Narayanan, et D. Fox, « PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes », arXiv.org. Consulté le: 7 février 2025. [En ligne]. Disponible sur: <https://arxiv.org/abs/1711.00199v3>
- [9] T. Zheng, C. Zhang, S. Zhang, et Y. Wang, « Deep Learning-Based 6-DoF Object Pose Estimation Considering Synthetic Dataset », *Sensors*, vol. 23, n° 24, Art. n° 24, janv. 2023, doi: 10.3390/s23249854.
- [10] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, et N. Vitzilaios, « Fiducial Markers for Pose Estimation », *J. Intell. Robot. Syst.*, vol. 101, n° 4, p. 71, mars 2021, doi: 10.1007/s10846-020-01307-9.
- [11] « [https://www.universal-robots.com/media/1820985/07\\_2021\\_ur16e\\_fact\\_sheet\\_fr\\_web.pdf](https://www.universal-robots.com/media/1820985/07_2021_ur16e_fact_sheet_fr_web.pdf) ». Consulté le: 20 mars 2025. [En ligne]. Disponible sur: [https://www.universal-robots.com/media/1820985/07\\_2021\\_ur16e\\_fact\\_sheet\\_fr\\_web.pdf](https://www.universal-robots.com/media/1820985/07_2021_ur16e_fact_sheet_fr_web.pdf)
- [12] K.-H. Ahn, M. Na, et J.-B. Song, « Robotic assembly strategy via reinforcement learning based on force and visual information », *Robot. Auton. Syst.*, vol. 164, p. 104399, juin 2023, doi: 10.1016/j.robot.2023.104399.