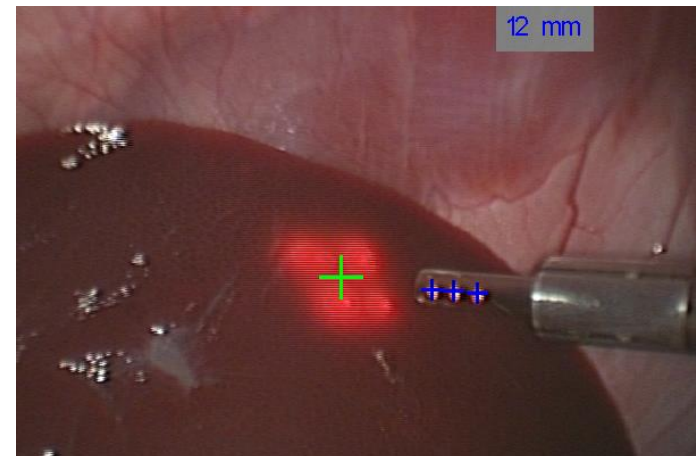




Vision et Commande (3/3)

(Partie 1 : Vision robotique)

Télécom Physique 3A ISAV et Master IRIV
(édition 2019-2020)



Christophe DOIGNON

Professeur à Télécom Physique Strasbourg
ICube, Université de Strasbourg

Bureau C418 – courriel : c.doignon@unistra.fr
Université de Strasbourg, FRANCE



Plan du cours

Chapitre 1 : Introduction

- Introduction,
- Espaces et coordonnées homogènes
- Représentation des rotations
- Applications

Chapitre 2 : De la scène 3D à l'image 2D (*et vice-versa*)

- Acquisition d'images
- Modélisation géométrique
- Recalage 2D-3D rigide (*incluant la Pose d'une caméra*)

CAMERAS



Chapitre 3 : de l'analyse du mouvement apparent aux asservissements visuels

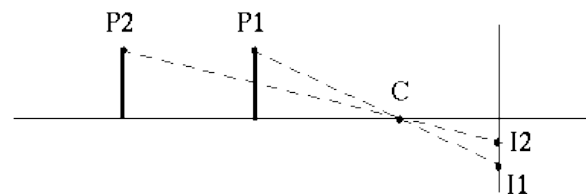
- Analyse du champ de mouvement
- Asservissement visuel basé image (IBVS)
- Asservissement visuel basé position (PBVS)





Chapitre 3

De l'analyse du mouvement apparent aux asservissements visuels





3.1 Analyse du champ de mouvement

- Une **analyse cinématique** n'est envisageable que lorsque la cadence des acquisitions d'images est rapide vis-à-vis des mouvements des objets d'intérêt de la scène perçue.



Asservissement visuel rapide, binoculaire et déporté à 1000 images/s (Tokyo University)



3.1 Analyse du champ de mouvement

➤ Analyse cinématique

- Dans le cas où des variables à asservir sont définies dans l'espace 3-D, ce sont les variations infinitésimales de l'orientation et de la position du capteur visuel qui constituent l'information cinématique pertinente. A partir d'une configuration 3-D de référence (**orientation 3-D et position 3-D**) on peut construire un asservissement visuel 3-D (AV3D) entre le repère du capteur et celui de l'objet d'intérêt

➔ **position-based visual servoing**

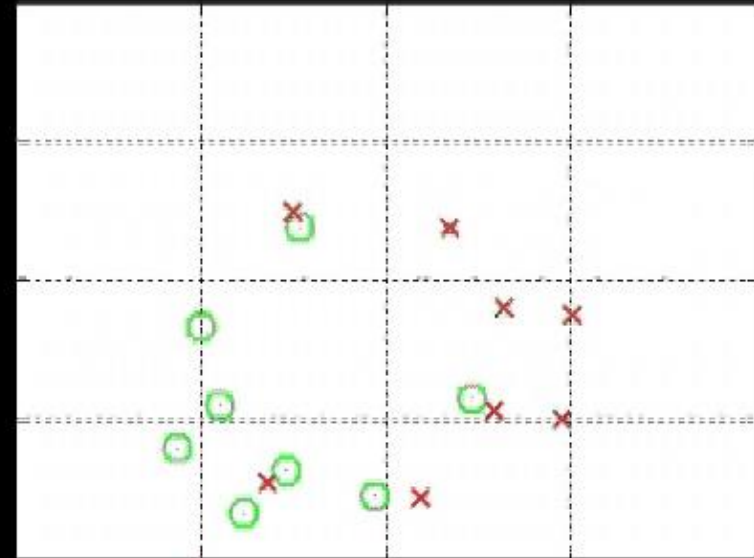


3.1 Analyse du champ de mouvement

➤ Analyse cinématique

- Dans le cas où des variables à asservir sont dans l'image, ce sont les variations infinitésimales des images qui constituent l'information cinématique (apparente). Alors à partir d'une image de référence, (**indices visuels 2-D**) on peut construire un asservissement visuel 2-D (AV2D, appelé aussi basé image)

➔ **image-based visual servoing**



Asservissement visuel à partir de marqueurs (points)



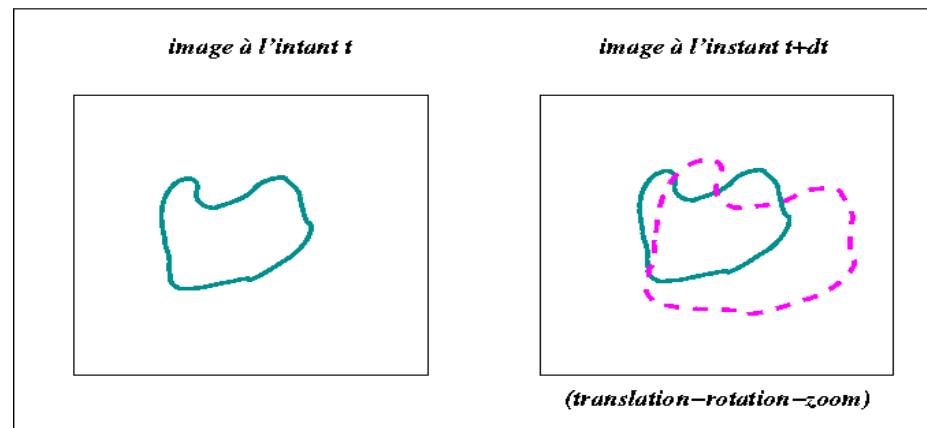
3.1 Analyse du champ de mouvement

➤ Analyse cinématique

- Il existe aussi des techniques utilisant simultanément des variables 2-D et 3-D à asservir : ce sont les techniques dites **hybrides**. Celles-ci sont adaptées au suivi visuel de scènes inconnues, dont l'objet cible peut être reconstruit s'il n'est pas en mouvement, c'est-à-dire réaliser une cartographie. C'est ce que l'on nomme le VSLAM ou SLAM Visuel (*Visual Simultaneous Localisation And Mapping*)

\\VBOXSVR\win-share\cours\ViCo\videos\2006-Silveira_ICRA.avi

- Lorsque des primitives géométriques (ou indices visuels ou amers) simples (points, droites, coniques, moments géométriques,...) sont extraites de l'image par segmentation, une *matrice d'interaction* leur est associée, matrice provenant directement du calcul d'un jacobien lors d'une modélisation cinématique. Si les indices visuels sont iconiques (niveau pixels), on recherchera une vitesse apparente dans l'image, permettant également d'élaborer un suivi visuel (*visual tracking*). Le lien mathématique existant entre la vitesse apparente d'un indice visuel et les vitesses (rotation et translation) dans la scène 3-D de l'objet dont il est la projection s'appelle le champ de mouvement (*motion field*).

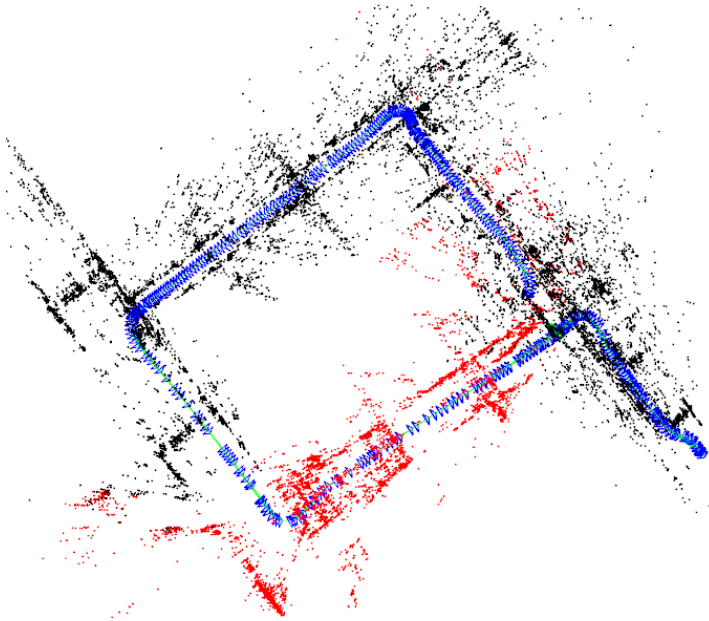




3.1 Analyse du champ de mouvement

Application au suivi visuel et à la cartographie (VSLAM)
(J.M.Montiel, *Université de Saragosse*)

<https://www.youtube.com/watch?v= 9VcvGybsDA>



A droite, image d'un corridor avec les points d'intérêt sélectionnés par l'algorithme ORB SLAM2 et la construction de la carte correspondante à la trajectoire (à gauche, en bleue).

En noir : les points détectés et mémorisés, en rouge les points présents dans l'image courante.



3.1 Analyse du champ de mouvement

➤ Descripteurs locaux





3.1 Analyse du champ de mouvement

➤ Descripteurs locaux

- Des indices visuels plus élaborés peuvent être construits à partir d'une information géométrique locale à chaque point d'intérêt, permettant de pouvoir l'associer plus fidèlement « à son correspondant » dans plusieurs images d'une même séquence (ce n'est pas du tout évident quand on dispose de plusieurs centaines de points d'intérêt par image !).
- Puisque qu'il ne s'agit pas que de la localisation du point d'intérêt mais aussi d'intégrer la distribution de l'intensité et de ses dérivées spatiales (gradient) entre autres, on préfère utiliser le terme de **descripteur local**.
- Ainsi par exemple les descripteurs SIFT (*Scale Invariant Feature Transform; David Lowe*) sont « robustes » aux variations d'échelles, qui émanent le plus souvent d'un rapprochement ou d'un éloignement du système de vision par rapport à l'objet d'intérêt durant la séquence.



3.1 Analyse du champ de mouvement

➤ Descripteurs locaux

- Il en existe d'autres (SURF, ORB, ...) avec la plupart du temps des fonctions et des bibliothèques associées dans OpenCV, pour plusieurs langages de programmation. Tous ces descripteurs ont été conçus pour qu'ils puissent être calculés en temps réel (ou presque...). Certains sont également peu sensibles (plutôt qu'invariants) aux variations d'orientation (dans le plan image), voire à des transformations affines.

Combining Harris Interest Points and the SIFT Descriptor for Fast Scale-Invariant Object Recognition

Pedram Azad, Tamim Asfour, Rüdiger Dillmann
University of Karlsruhe





3.1 Analyse du champ de mouvement

➤ Fonction de tâche

- L'objectif de toute procédure de suivi (visuel) ou d'asservissement (visuel) est de minimiser une erreur vectorielle $e(t)$ qui peut être définie génériquement par la relation :

$$e(t) = s(m(t), a) - s^*(t)$$

- Le vecteur $m(t)$ représente un ensemble de mesures issues de l'image acquise. Ces mesures sont alors employées pour calculer k indices visuels (comme des coordonnées image, par exemple) formant le vecteur $s(t) = s(m(t), a)$, vecteur d'état actuel ou courant, et dans lequel le vecteur a contient des paramètres relatifs à certaines informations intrinsèques sur le système de vision ou en rapport avec des grandeurs géométriques caractéristiques de l'objet d'intérêt, s'il y en a.
- $s^*(t)$ est la référence, c'est-à-dire le vecteur d'état désiré regroupant les *indices visuels souhaités*. S'il est constant, alors on le note simplement s^* . Dans le cas d'un suivi de trajectoires par exemple, on construit souvent une séquence de vecteurs s^* , pour laquelle peut être réalisée une interpolation entre les valeurs successives, ce qui aboutit à des consignes (de fonctions continues et) variables.

➔ *L'analyse cinématique permet l'obtention d'une formulation mathématique entre les variations temporelles de $s(t)$ et la vitesse de la caméra par rapport à l'objet.*

Remarque : la fin de la phrase aurait pu être également «et la vitesse de l'objet par rapport à la caméra », selon qu'on dispose d'une caméra embarquée ou déportée. Même si l'objet et la caméra sont tout deux en mouvement par rapport à un référentiel galiléen, c'est la vitesse relative qui importe ici (et le repère dans lequel elle est exprimée).



3.1 Analyse du champ de mouvement

➤ Commandes cinématiques

- La relation entre $\dot{s}(t) = \frac{ds}{dt}$ et le torseur cinématique $\tau_c(t)$ est supposée être linéaire, et τ_c correspond au vecteur de commande dans le cadre d'un asservissement :

$$\dot{s}(t) = L_{s(t)} \tau_c(t)$$

où $L_{s(t)} = \frac{\partial s}{\partial \tau_c}$ est la **matrice d'interaction** autour de l'état $s(t)$. C'est une matrice jacobienne réelle de dimensions $(k \times m)$ si le torseur cinématique τ_c est composé de m composantes commandables. La plupart du temps $m \leq 6$. Pour une cible (ou objet d'intérêt) fixe, la référence n'évolue pas ($\frac{ds^*}{dt} = 0$) et on a : $\dot{e}(t) = L_{s(t)} \tau_c(t)$ où $e(t)$ est l'erreur vectorielle, appelée **fonction de tâche**.

- Synthèse de la loi de commande : en souhaitant par exemple à obtenir un comportement exponentiel décroissant de la fonction de tâche, on aboutit à une correction proportionnelle : $\tau_c(t) = K_p e(t)$

$$\dot{e}(t) + \lambda e(t) = 0$$

avec λ un gain scalaire positif ou une matrice définie positive. On obtient :

$$\tau_c(t) = -\lambda L_{s(t)}^+ e(t) \quad \Rightarrow \quad K_p = -\lambda L_{s(t)}^+$$

où $L_{s(t)}^+$ est la matrice pseudo-inverse de gauche de la matrice $L_{s(t)}$, c'est-à-dire $L_s^+ = (L_s^T L_s)^{-1} L_s^T$ quand $L_{s(t)}$ est de rang m .



3.1 Analyse du champ de mouvement

➤ Commandes cinématiques

• Remarques :

- ✓ Comme il n'est pas souvent possible en pratique de connaître parfaitement $L_{s(t)}$ ou de connaître parfaitement sa pseudo-inverse $L_{s(t)}^+$, une estimation des valeurs numériques de l'une ou l'autre de ces matrices est effectuée (notation \hat{L}) et les deux lois de commande cinématiques sont alors exprimées par :

$$\tau_c(t) = -\lambda \hat{L}_{s(t)}^+ e(t) \quad \text{ou} \quad \tau_c(t) = -\lambda \widehat{L_{s(t)}^+} e(t)$$

(noter bien les emplacements de l'accentuation « chapeau »).

- ✓ La matrice $L_{s(t)}$ doit être mise à jour à chaque acquisition d'une nouvelle image dans les deux commandes précédentes. Mais il arrive qu'elle soit partiellement mise à jour, voire maintenue constante durant tout l'asservissement. Dans ce dernier cas, la matrice d'interaction est estimée pour l'état désiré L_{s^*} , et la commande cinématique prend alors la forme suivante :

$$\tau_c(t) = -\lambda \hat{L}_{s=s^*}^+ e(t)$$

- ✓ Alors que la linéarisation entre les variations du vecteur d'état s et le vecteur de commande τ_c a été effectuée avec une approximation au premier ordre, $L_{s(t)} = \frac{\partial s}{\partial \tau_c}^T$, on peut obtenir une approximation plus fine, au second ordre, en considérant une pondération entre les matrices d'interaction $L_{s(t)}$ et L_{s^*} appelée ESM (*Efficient Second Order Minimization*):

$$\tau_c(t) = -\lambda \hat{L}_s^+ e(t) \quad \text{où} \quad L_s = (1 - \mu) L_{s^*} + \mu L_{s(t)} \quad \text{et} \quad \mu \in [0,1]$$



3.1 Analyse du champ de mouvement

➤ Commandes cinématiques

- Dans le cas où la commande pourra tendre vers un vecteur constant (on le souhaite nul le plus souvent), on réalisera un *asservissement visuel* en cherchant à minimiser l'erreur par le déplacement d'un dispositif mécanique. Sinon, si tout ou partie du torseur cinématique n'est pas associée à une commande, on pourra utiliser ce formalisme pour réaliser un *suivi visuel*.

Pour la suite du chapitre, nous allons nous intéresser aux asservissements visuels basés image et basés position, pour commander les mouvements d'un robot, indépendamment de sa chaîne cinématique.

❑ **EXERCICE** : Dans le cas où la fonction de tâche n'a pas le même nombre de composantes que le vecteur d'état, on formalise la relation vectorielle de l'erreur ainsi : $e(t) = C (s(t) - s^*(t))$ où C est une matrice réelle (supposée constante ici). Dans le cas d'une consigne constante et d'un gain λ scalaire, montrez d'après ce qui précède que pour obtenir un comportement exponentiel décroissant de l'erreur, la commande cinématique doit s'exprimer par la relation $\tau_c(t) = -\lambda (\widehat{CL_{s(t)}})^+ e(t)$ ou par la relation suivante $\tau_c(t) = -\lambda \widehat{CL_{s(t)}}^+ e(t)$.



3.1 Analyse du champ de mouvement

- Remarque : le signal de sortie du contrôleur visuel est donc un torseur cinématique ici, noté τ_c et dont les composantes sont exprimées dans le repère caméra. Si le repère caméra (R_c) coïncide avec le repère de l'organe terminal (R_e) d'un robot (e pour *end-effector*), nous pouvons utiliser son Jacobien inverse afin de déterminer les vitesses articulaires à programmer pour parvenir à la configuration désirée. Dans la plupart des applications, le repère caméra ne coïncide pas avec celui de l'organe terminal, mais pour les systèmes embarqués (**eye-in-hand**) il est rigidement relié à celui-ci. Une transformation euclidienne entre les deux repères doit alors être estimée afin de traduire la vitesse de la caméra dans le repère de l'organe terminal. A cette fin, le torseur cinématique est noté ici τ_e quand il est exprimé dans (R_e).
- ✓ **Changement de repères d'un torseur cinématique** : en notant par cT_e la transformation euclidienne existant entre le repère (R_c) et le repère (R_e) et exprimée dans le repère (R_c), le changement de repères pour exprimer le torseur cinématique τ correspond à

$$\tau_c = {}^cG_e \tau_e$$

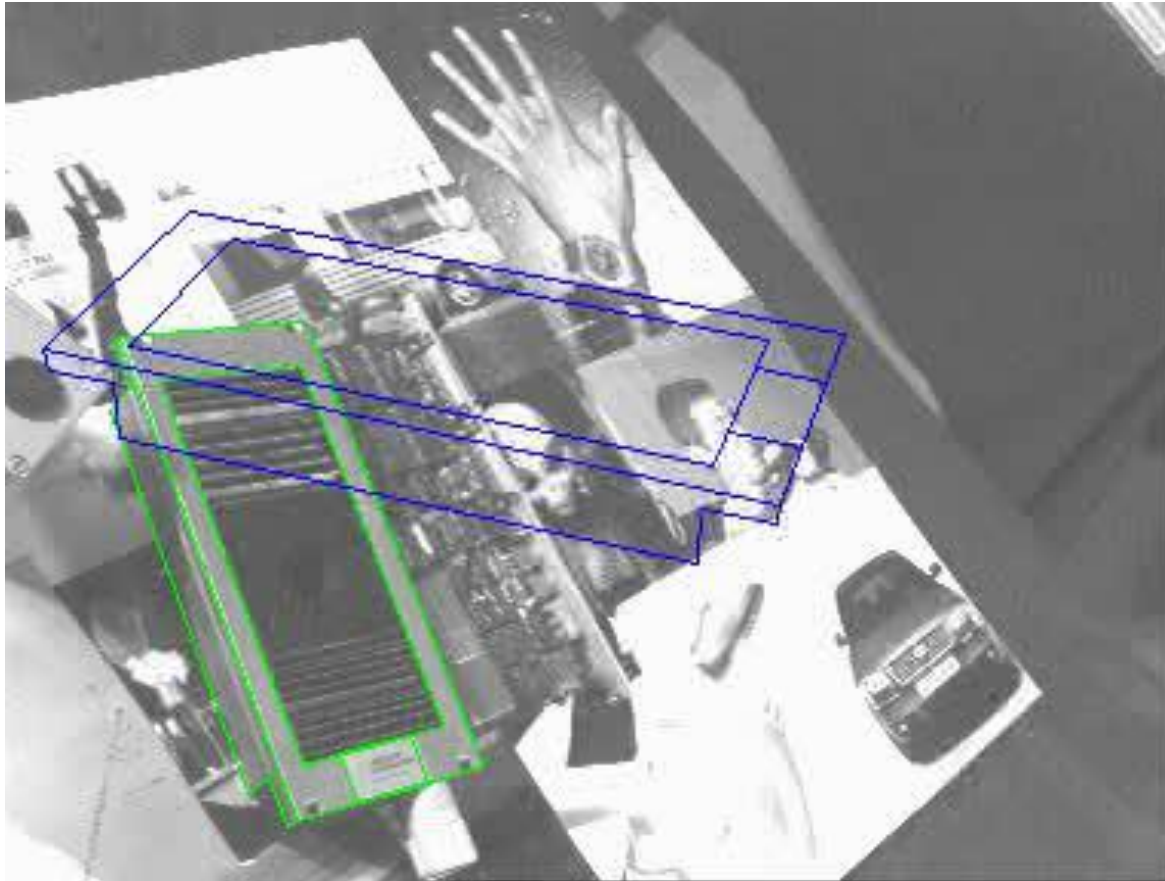
avec la matrice (6×6) cG_e :

$${}^cG_e = \begin{bmatrix} {}^cR_e & [{}^ct_e]_{\times} {}^cR_e \\ 0_{3 \times 3} & {}^cR_e \end{bmatrix} \quad \text{et} \quad {}^cT_e = \begin{bmatrix} {}^cR_e & {}^ct_e \\ 0_{1 \times 3} & 1 \end{bmatrix}$$



3.1 Analyse du champ de mouvement

➤ Descripteurs locaux



Asservissement visuel (régulation robuste) à partir de descripteurs locaux composés de droites. Les droites vertes composent la projection actuelle de l'objet cible (vecteur d'état à asservir), alors que les droites bleues modélisent la projection souhaitée (vecteur d'état désiré : la consigne) (INRIA Rennes)



3.1 Analyse du champ de mouvement

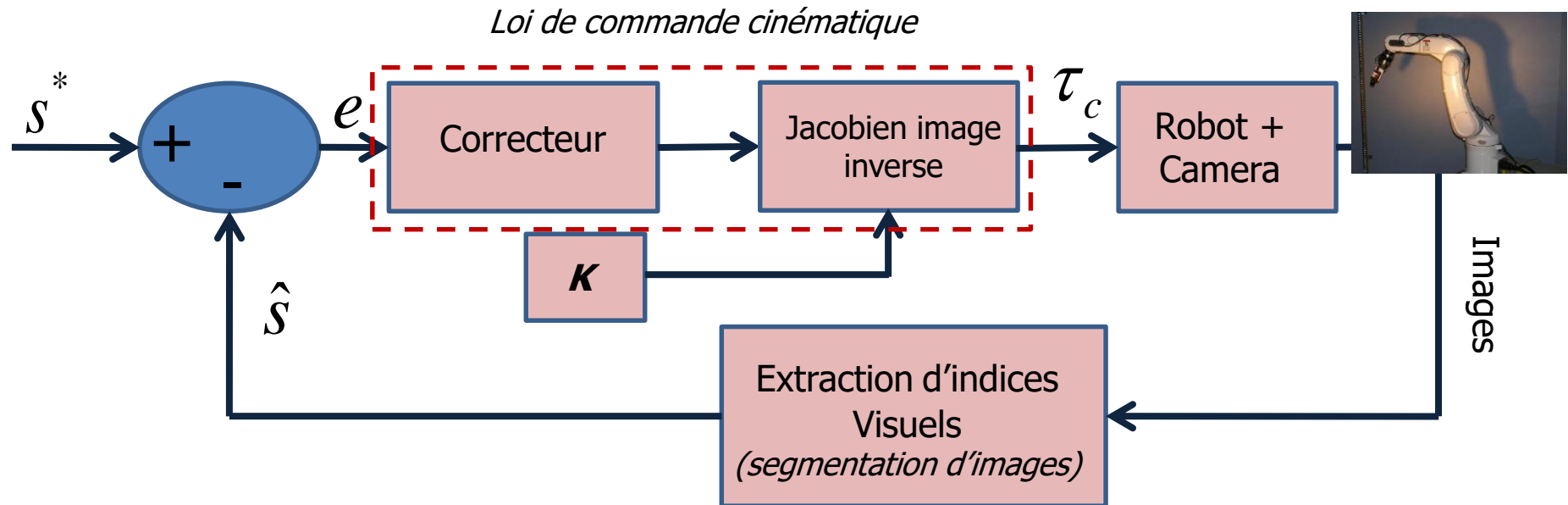
Kinematic Control Algorithms for On-Line Obstacle Avoidance for Redundant Manipulators

Leon Žlajpah
Bojan Nemec

Asservissement visuel avec caméra déportée et redondance structurelle. Quelle est la tâche principale à réaliser?



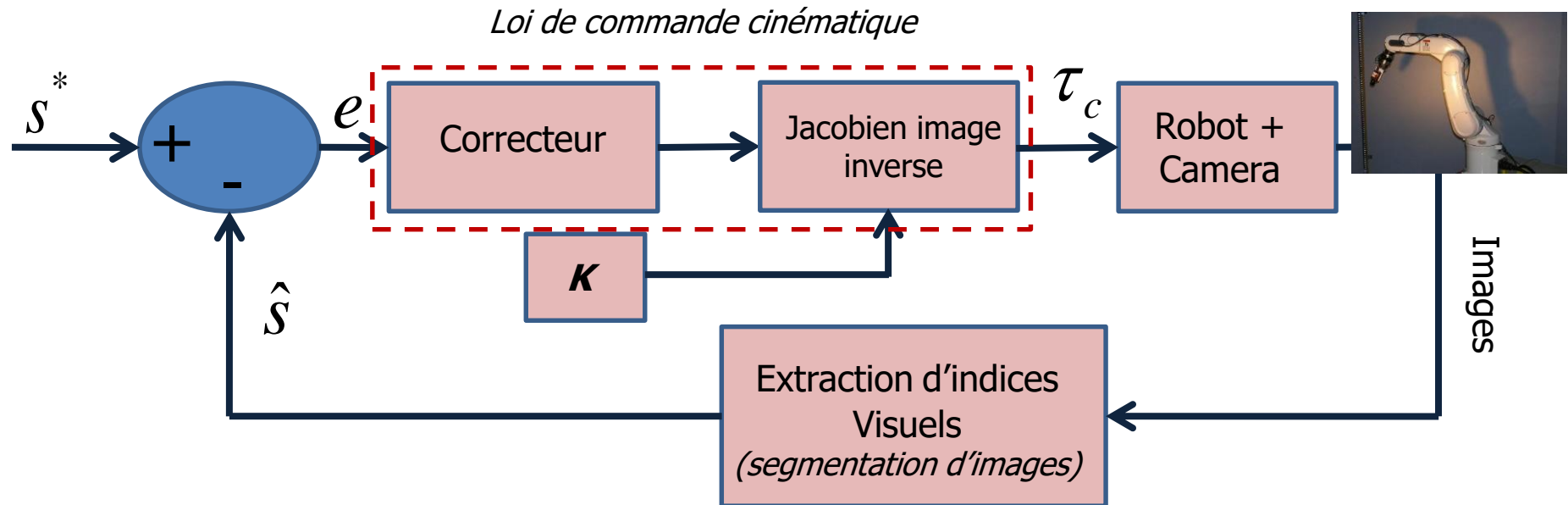
3.2 Asservissement visuel basé image



- **Matrice d'interaction en IBVS** : souvent issue d'une matrice jacobienne, la matrice d'interaction pour les AV basé image représente la relation géométrique entre des variations dans l'image de primitives géométriques (ou indices visuels) et le torseur cinématique lié au solide en mouvement (caméra, objet de la scène, organe terminal d'un robot,...).



3.2 Asservissement visuel basé image



- **Motion field** : Avec les points géométriques comme indices visuels, à une variation infinitésimale des coordonnées (inhomogènes) image $Q = \Pi(P)$, $dQ = (du, dv)^T$, entre deux instants t et $t' = t + dt$, correspond une variation des coordonnées du point 3-D P , $d^c P = (d^c x, d^c y, d^c z)^T$, de l'objet d'intérêt de la scène, exprimée dans le repère (R_c) attaché à la caméra.



3.2 Asservissement visuel basé image

➤ From Motion field to IBVS

- Par la suite, nous nous limiterons à établir la relation mathématique entre les variations des coordonnées de points de l'image et les composantes du torseur cinématique représentant la vitesse de la caméra par rapport à l'objet d'intérêt (ce qui inclut le cas où c'est uniquement l'objet d'intérêt qui est en déplacement), en utilisant le modèle de la projection perspective que l'on dérive :

$${}^c_Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^c_x \\ {}^c_y \\ {}^c_z \\ 1 \end{bmatrix} \Rightarrow {}^c_Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + {}^c_Z \begin{bmatrix} \dot{u} \\ \dot{v} \\ 0 \end{bmatrix} = K \begin{bmatrix} {}^c_{\dot{x}} \\ {}^c_{\dot{y}} \\ {}^c_{\dot{z}} \end{bmatrix} \text{ et } K = \begin{bmatrix} k_{11} & 0 & k_{13} \\ 0 & k_{22} & k_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow {}^c_Z \begin{bmatrix} \dot{u} \\ \dot{v} \\ 0 \end{bmatrix} = K \begin{bmatrix} {}^c_{\dot{x}} \\ {}^c_{\dot{y}} \\ {}^c_{\dot{z}} \end{bmatrix} - {}^c_Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- Remarque : comme la dernière ligne de la matrice K vaut $[0 \ 0 \ 1]$ puisqu'elle représente une transformation affine 2-D, la dernière ligne de l'équation ci-dessus devient inutile.
- La dérivée dP/dt est la vitesse du point 3-D P par rapport à l'origine du repère caméra (R_c), exprimée dans le repère (R_c). On la note $(V_{R_c}^P)_{(/R_c)}$, et on obtient, pour un couple de points appariés (${}^cP, Q$) :

$${}^c_Z \dot{Q} = \begin{bmatrix} 1 & 0 & -u \\ 0 & 1 & -v \end{bmatrix} K (V_{R_c}^P)_{(/R_c)}$$



3.2 Asservissement visuel basé image

➤ From Motion field to IBVS

- Pour aller plus loin, il nous faut maintenant considérer que l'un ou l'autre de l'objet d'intérêt ou de la caméra est fixe, afin que celui-ci puisse être attaché à un repère de type galiléen (en fait, une vitesse constante en norme et en direction est une condition suffisante, ce qu'on appelle un mouvement de translation rectiligne uniforme). En notant ce repère galiléen (R_f) , supposons que le repère de la caméra (R_c) , d'origine C , est animé par rapport à lui de la vitesse linéaire v_c et de la vitesse de rotation ω_c rassemblées dans un torseur $\tau_c = (v_c, \omega_c)$, alors on note :

$$v_c = (V_{R_f}^C)_{(/R_c)}$$

[vitesse de l'origine C du repère (R_c) par rapport à (R_f) , exprimée dans (R_c)]

$$\omega_c = ({}^f\Omega_c)_{(/R_c)}$$

[vitesse du repère (R_c) par rapport au repère (R_f) , exprimée dans (R_c)]

- Si l'objet d'intérêt est fixe, alors tout point 3-D P de cet objet rigide est fixe également $[(V_{R_f}^P)_{(/R_f)} = 0$ – environnement statique, c'est alors le repère objet (R_o) qui lui est attaché qui est de type galiléen, et on peut alors considérer que $(R_o) \equiv (R_f)$], alors en employant la **loi de composition des vitesses** (entre deux repères dont un est galiléen), il vient :

$$(V_{R_f}^P)_{(/R_c)} = {}^cR_f(V_{R_f}^P)_{(/R_f)} = 0 = (V_{R_c}^P)_{(/R_c)} + (V_{R_f}^C)_{(/R_c)} + ({}^f\Omega_c)_{(/R_c)} \wedge CP_{(/R_c)}$$

ou simplement :

$$(V_{R_c}^P)_{(/R_c)} = -v_c - \omega_c \wedge CP_{(/R_c)}$$



3.2 Asservissement visuel basé image

➤ From Motion field to IBVS

- Il vient finalement : $(V_{R_c}^P)_{(/R_c)} = -v_c + CP_{(/R_c)} \wedge \omega_c$

C'est-à-dire : $\dot{Q} = L_Q \tau_c$ avec la matrice d'interaction L_Q :

$$\underbrace{L_Q}_{(2 \times 6)} = \frac{1}{c_z} \begin{bmatrix} 1 & 0 & -u \\ 0 & 1 & -v \end{bmatrix} K \underbrace{[-I \ ; \ [CP_{(/R_c)}] \times]}_{(3 \times 6)}$$

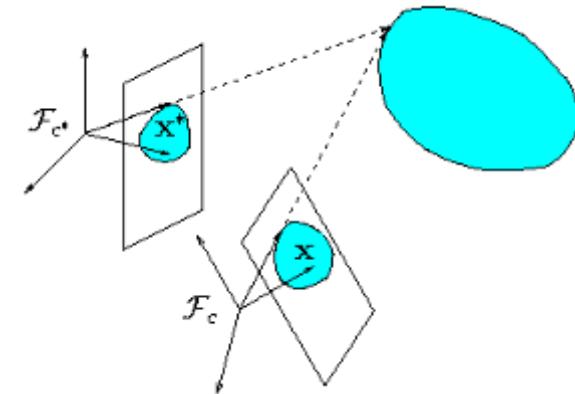
- Remarques : (**vitesse de translation**)

- ✓ Dans la matrice d'interaction (pour un couple (P, Q)), L_Q , de dimensions (2×6) apparaît la « profondeur » $^c z$ du point 3-D par rapport au repère caméra, celle-ci est liée à la vitesse de translation v_c mais pas à la vitesse de rotation ω_c . Ainsi si on note $L_Q = [L_v \ L_\omega]$, on a alors la décomposition suivante :

$$\dot{Q} = L_v v_c + L_\omega \omega_c$$

et la matrice $\underbrace{L_v}_{(2 \times 3)} = -\frac{1}{c_z} \begin{bmatrix} 1 & 0 & -u \\ 0 & 1 & -v \end{bmatrix} K = -\frac{1}{c_z} \begin{bmatrix} \frac{f}{l_x} & 0 & u_c - u \\ 0 & \frac{f}{l_y} & v_c - v \end{bmatrix}$ dépend de $^c z$.

- ✓ L_v dépend linéairement de l'état $Q \rightarrow L_v(Q, ^c z)$





3.2 Asservissement visuel basé image

➤ From Motion field to IBVS

- Il vient finalement : $(V_{R_c}^P)_{(/R_c)} = -v_c + CP_{(/R_c)} \wedge \omega_c$

C'est-à-dire : $\dot{Q} = L_Q \tau_c$ avec la matrice d'interaction L_Q :

$$\underbrace{L_Q}_{(2 \times 6)} = \frac{1}{c_z} \begin{bmatrix} 1 & 0 & -u \\ 0 & 1 & -v \end{bmatrix} \mathbf{K} \underbrace{[-I \ ; \ [CP_{(/R_c)}]_{\times}]}_{(3 \times 6)}$$

- Remarques : (vitesse de rotation)

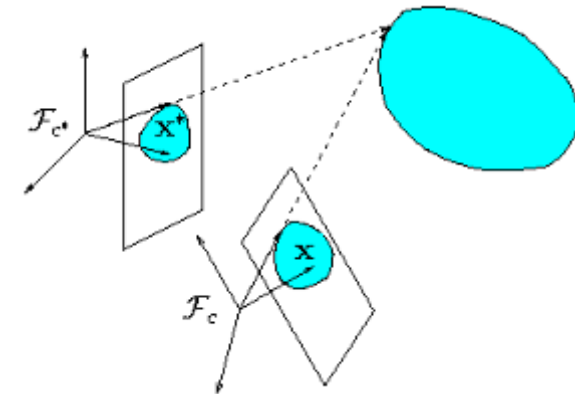
Avec la décomposition $\dot{Q} = L_v v_c + L_\omega \omega_c$,

la matrice $\frac{1}{c_z} [CP_{(/R_c)}]_{\times} = \begin{pmatrix} 0 & -1 & \frac{c_y}{c_z} \\ 1 & 0 & -\frac{c_x}{c_z} \\ -\frac{c_y}{c_z} & \frac{c_x}{c_z} & 0 \end{pmatrix}$; avec $\begin{cases} \frac{c_x}{c_z} = \frac{l_x}{f} (u - u_c) \\ \frac{c_y}{c_z} = \frac{l_y}{f} (v - v_c) \end{cases}$

$$\checkmark \quad \underbrace{L_\omega}_{(2 \times 3)} = \begin{bmatrix} 1 & 0 & -u \\ 0 & 1 & -v \end{bmatrix} \mathbf{K} \frac{1}{c_z} [CP_{(/R_c)}]_{\times} = \begin{pmatrix} f/l_x & 0 & u_c - u \\ 0 & f/l_y & v_c - v \end{pmatrix} \begin{pmatrix} 0 & -1 & -\frac{l_y}{f} (v_c - v) \\ 1 & 0 & \frac{l_x}{f} (u_c - u) \\ \frac{l_y}{f} (v_c - v) & -\frac{l_x}{f} (u_c - u) & 0 \end{pmatrix}$$

Ne dépend pas de c_z .

✓ L_ω dépend de l'état $Q \rightarrow L_\omega(Q)$





3.2 Asservissement visuel basé image

- Remarques : (sur la mise à jour de la matrice d'interaction)

La présence du terme z^c dans les composantes de L_v nécessite que cette grandeur soit estimée ou approchée. Il s'ensuit que l'on ne dispose généralement que d'une estimation plus ou moins précise de L_Q . Trois stratégies sont alors communément employées :

- ✓ La matrice L_Q est mise à jour à chaque acquisition d'une nouvelle image. Ceci nécessite alors l'estimation de $^c z$ (voir chapitre 2). Du fait que cette grandeur est très sensible au bruit et aux erreurs de calibration, un filtrage temporel est conseillé.
- ✓ La matrice L_Q est maintenue constante et égale à la matrice d'interaction pour l'état désiré L_{Q^*} .
- ✓ La matrice L_Q est partiellement mise à jour : le facteur $^c z$ est maintenu constant (égal à une valeur approchée qui peut être la « profondeur » pour l'état désiré Q^* : $L_{v(^c z=z^*)}$ (sous-entendu z^* est la « profondeur » $^c z$ pour $Q = Q^*$). On peut ici remarquer que cela ne modifie pas L_ω par rapport à la mise à jour totale de la matrice L_Q .



3.2 Asservissement visuel basé image

- Remarques : (à propos de l'état désiré)

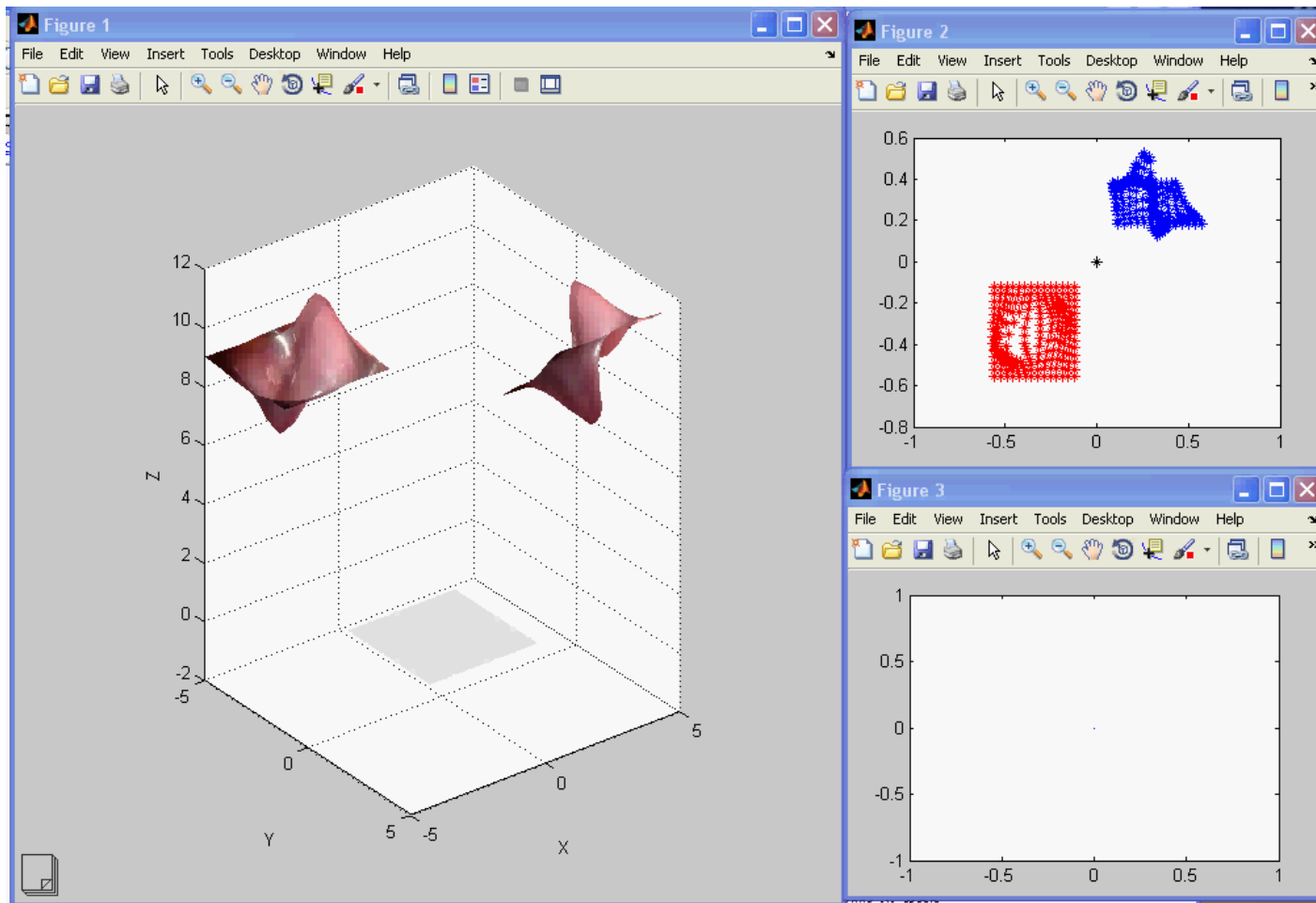
Nous avons vu dans l'analyse cinématique que pour mettre en œuvre un asservissement visuel, il fallait définir une erreur vectorielle $e(t) = s(t) - s^*$ (notation compacte) et par conséquent qu'il fallait définir une référence s^* . Le choix de cette référence n'est pas totalement « libre » et doit être choisie avec précaution; En effet, il faut que celle-ci corresponde à un état atteignable, et ceci correspond à vérifier les deux conditions suivantes :

- ✓ Pour tout asservissement visuel, la référence doit correspondre à des indices visuels visibles dans l'image de référence.
- ✓ Pour un asservissement visuel basé image, $s(t)$ et s^* ont des composantes issues de deux projections perspectives qui peuvent être géométriquement dépendantes (*contrainte épipolaire*).

Ces deux conditions sont remplies en définissant la référence à partir d'une image réelle ou virtuelle du même objet d'intérêt, permettant de vérifier implicitement (sans calcul) la relation $Q_{(t)}^T F Q_* = 0$ (dans le cas où les coordonnées de points image composent les états $s(t)$ et s^* respectivement), avec F la matrice fondamentale, singulière, pour un même point P de l'espace 3-D (au bruit près et erreurs de segmentation près dans le cas d'une image réelle). Les droites image émanant de droites dans l'espace 3-D vérifient implicitement cette condition entre deux vues.



3.2 Asservissement visuel basé image

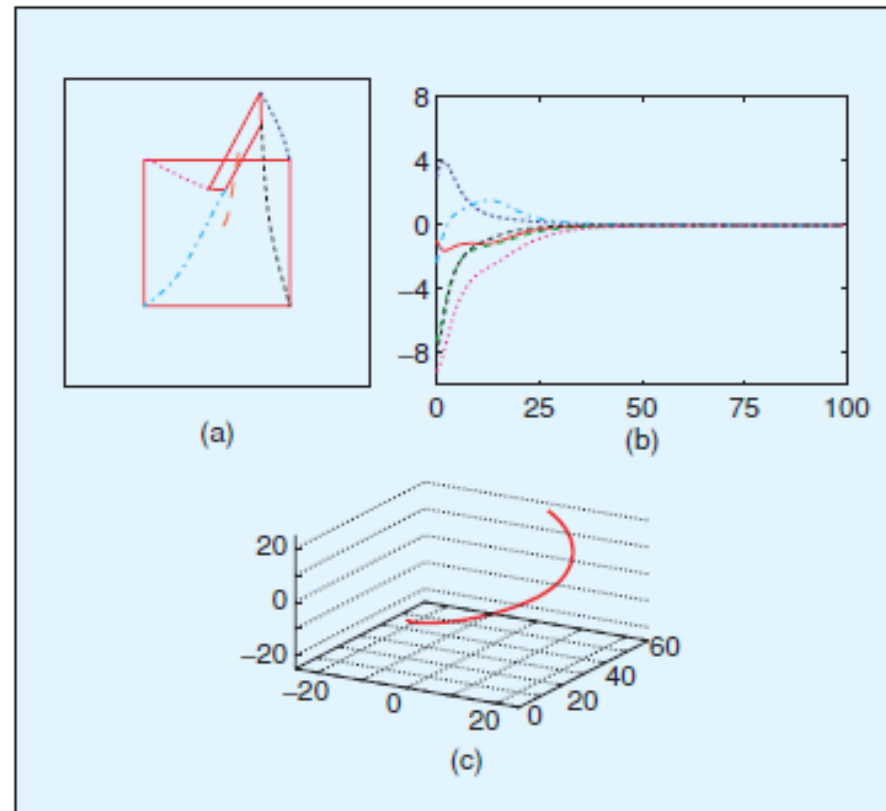


Simulation d' Asservissement Visuel à partir de points échantillonnant une surface gauche (ICube/AVR).

- A droite en haut : évolution de l'erreur dans le plan image (fonction de tâche e).
- A droite en bas : évolution de la commande (torseur cinématique τ)



3.2 Asservissement visuel basé image

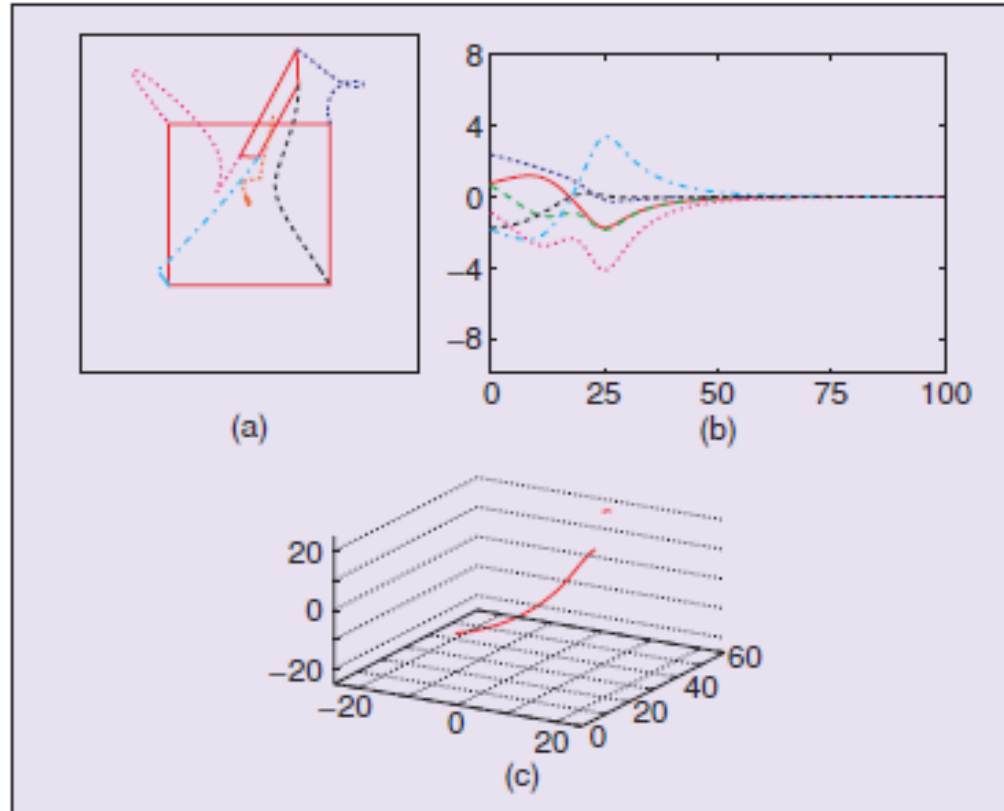


AV basé image avec $L_s = L_Q$ mise à jour à chaque acquisition
(on observe des trajectoires quasi-rectilignes dans l'image).

Visual Servo Control Part I: Basic Approaches, F. Chaumette, S. Hutchinson,
IEEE Robotics and Automation Mag, Décembre 2006



3.2 Asservissement visuel basé image

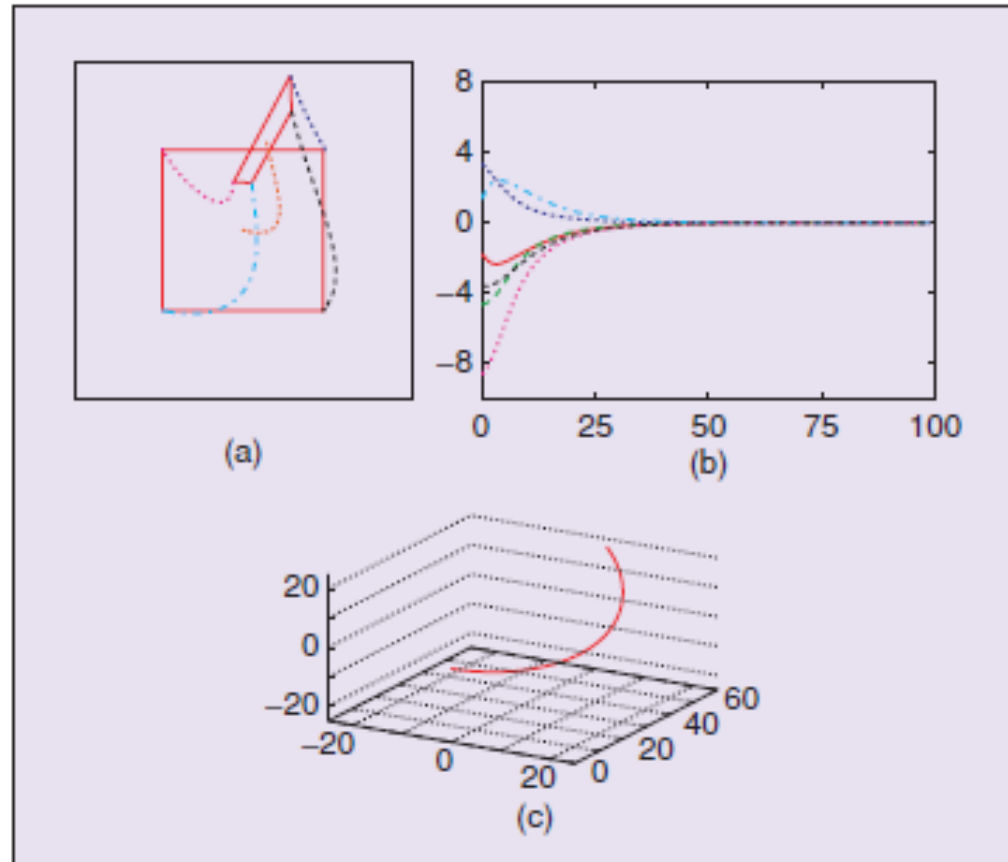


AV basé image avec L_s maintenue constante $L_s = L_Q^*$ (pas sûr que le système converge !).

Visual Servo Control Part I: Basic Approaches, F. Chaumette, S. Hutchinson,
IEEE Robotics and Automation Mag, Décembre 2006



3.2 Asservissement visuel basé image

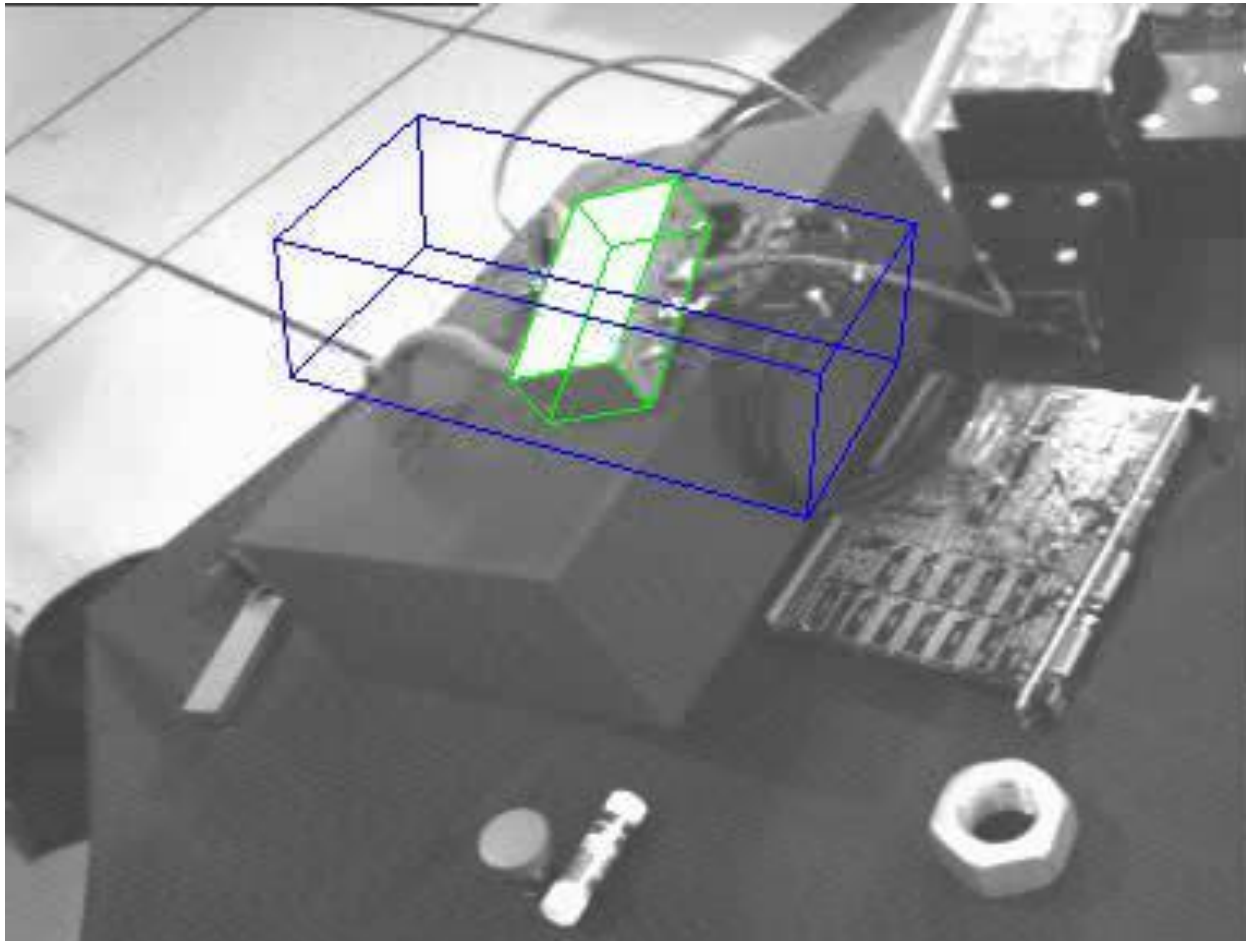


AV basé image avec $L_s = (L_{Q^*} + L_{Q(t)})/2$, (correspond à $\mu = 1/2$).

Visual Servo Control Part I: Basic Approaches, F. Chaumette, S. Hutchinson,
IEEE Robotics and Automation Mag, Décembre 2006



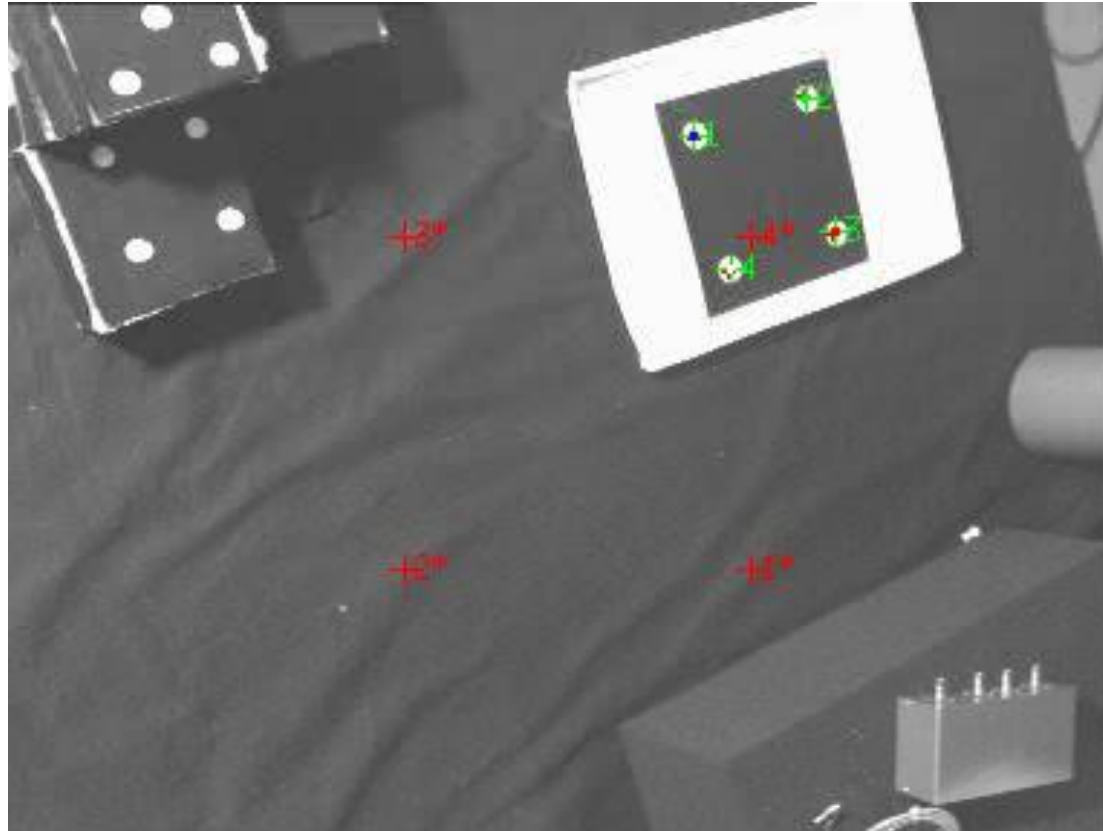
3.2 Asservissement visuel basé image



Asservissement et suivi visuel à partir de droites 3D
en mouvement (et occultations) (INRIA Rennes)



3.2 Asservissement visuel basé image (séquentiel)



Asservissement visuel séquentiel à partir de points
coplanaires (INRIA Rennes)



3.2 Asservissement visuel basé image

➤ AV stéréo



Asservissement visuel basé image monoculaire ou par stéréovision à partir d'une scène texturée



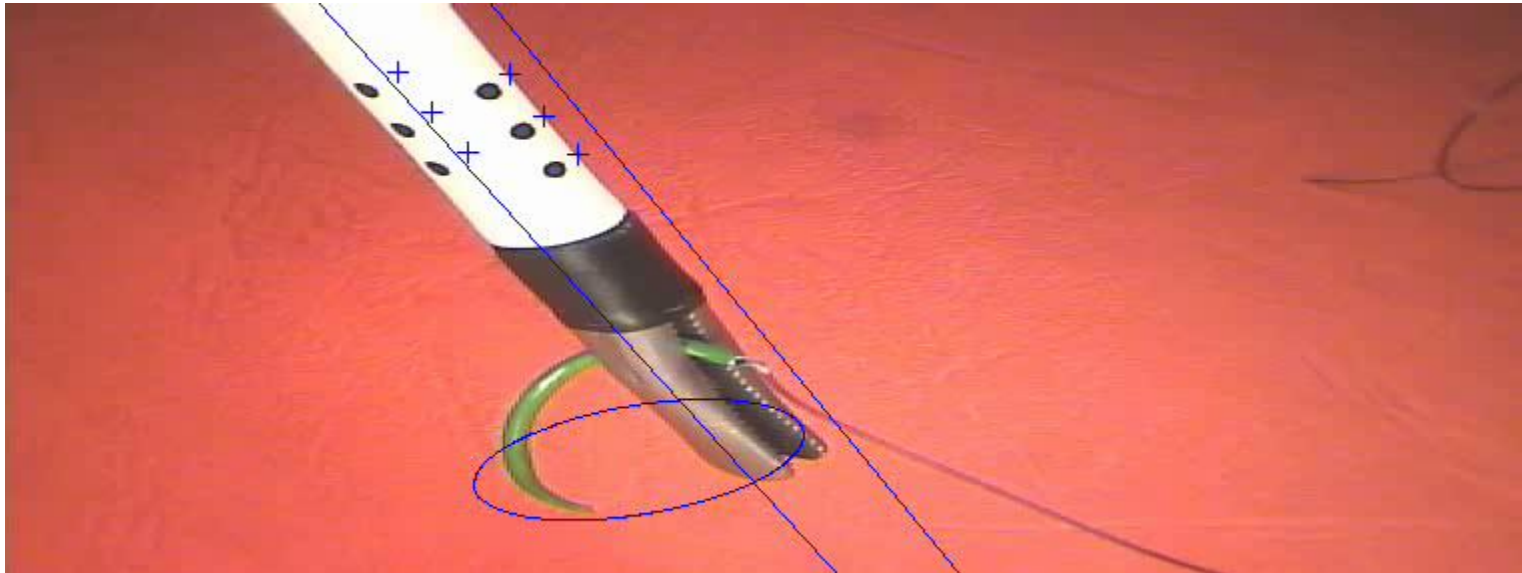
3.2 Asservissement visuel basé image



Asservissement visuel d'un instrument de chirurgie à partir de projection de lumière (et marqueurs lumineux) pour la navigation en chirurgie à invasion minimale (déplacement contraint par la présence d'un point - zéro effort - d'insertion, 4ddl) (ICube/AVR et IRCAD)



3.2 Asservissement visuel basé image (*virtuel* = *recalage*)



Asservissement visuel virtuel d'un instrument de chirurgie (pose de l'instrument à partir de trois types d'indices visuels et d'une camera endoscopique) (ICube/AVR)



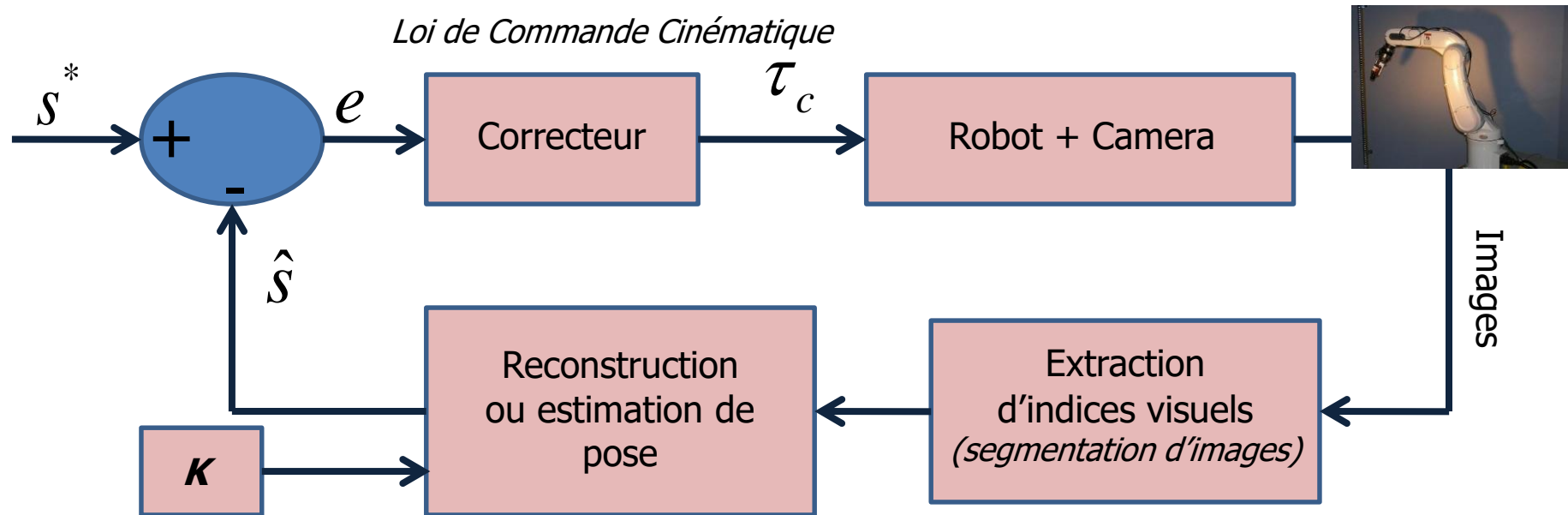
3.2 Asservissement visuel basé image

Asservissement et Reconnaissance en temps réel
avec une caméra RGB-D

<https://www.youtube.com/watch?v=ijhX8xfXKPE>



3.3 Asservissement visuel basé position



➤ Avec les **asservissements visuels basés position**, on impose une orientation désirée et une position désirée (au lieu d'une image désirée dans la section précédente). Il peut s'agir de la totalité des degrés de liberté d'une transformation euclidienne ou seulement d'une partie d'entre eux. Pour cette classe d'asservissement, deux sous-catégories émergent selon que l'on considère une tâche de positionnement (caméra vs objet) ou bien une tâche de navigation. Il s'agira donc successivement :

- de réaliser un recalage 2-D/3-D par rapport à l'objet d'intérêt (**camera pose**) ou
- d'estimer directement un déplacement 3-D relatif (**visual ego-motion**) entre une configuration initiale et une configuration désirée (scène supposée statique dans le cas d'un système de vision monoculaire).

➔ Dans les deux cas, le vecteur d'état s est composé d'indices 3-D.



3.3 Asservissement visuel basé position

➤ AV fondé sur la « pose » de la caméra

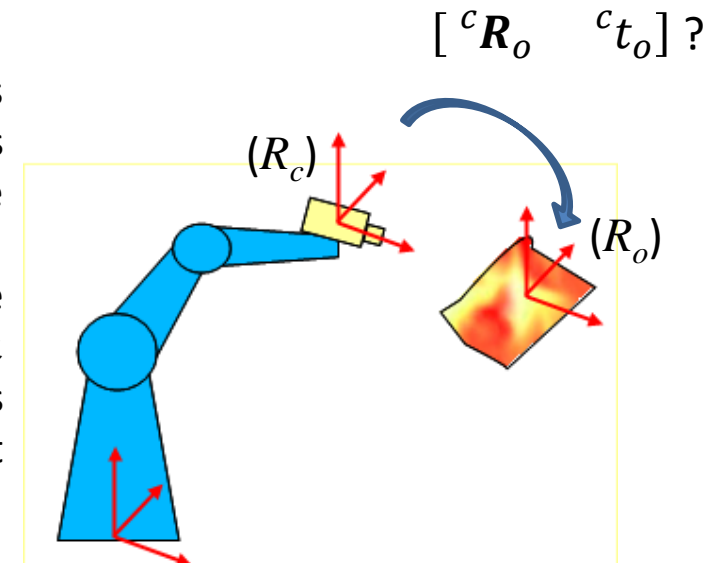
- ✓ Supposons dans un premier temps, et en lien avec le chapitre 2, que nous souhaitons, à partir d'une orientation initiale ${}^cR_o(t = t_i)$ et d'une position initiale ${}^c t_o(t = t_i)$ entre la caméra et l'objet intérêt, réorienter et repositionner la caméra C par rapport à l'objet d'intérêt O afin de l'amener vers une configuration désirée. L'orientation désirée pourra s'exprimer, par exemple, à l'aide d'une matrice de rotation désirée $({}^cR_o)^* = {}^cR_o(t_f)$ et la position désirée (vecteur reliant les origines des repères C et O par $({}^c t_o)^* = {}^c t_o(t = t_i)$.

- Pour cela, un vecteur d'état $s(t)$ est défini avec des informations 3-D estimées à partir des mesures visuelles extraites des images pour réaliser une reconstruction 3-D. Cela passe par une modélisation géométrique de l'objet d'intérêt dans le cas d'un système de vision monoculaire. Il est alors usuel, car pratique, de représenter les rotations ici sous forme de vecteur $r = \theta u$ (telles que ${}^cR_o = \exp([r]_{\times})$, voir section 1.3) afin de les comparer, ce qui aboutit à la définition des vecteurs d'état suivants (voir la figure précédente) :

$$s(t) = ({}^c t_o(t) \quad r(t))^T \quad \text{et} \quad s^* = (({}^c t_o)^* \quad r^*)^T$$

$$\text{avec : } ({}^cR_o)^* = \exp([r^* = r(t_f)]_{\times})$$

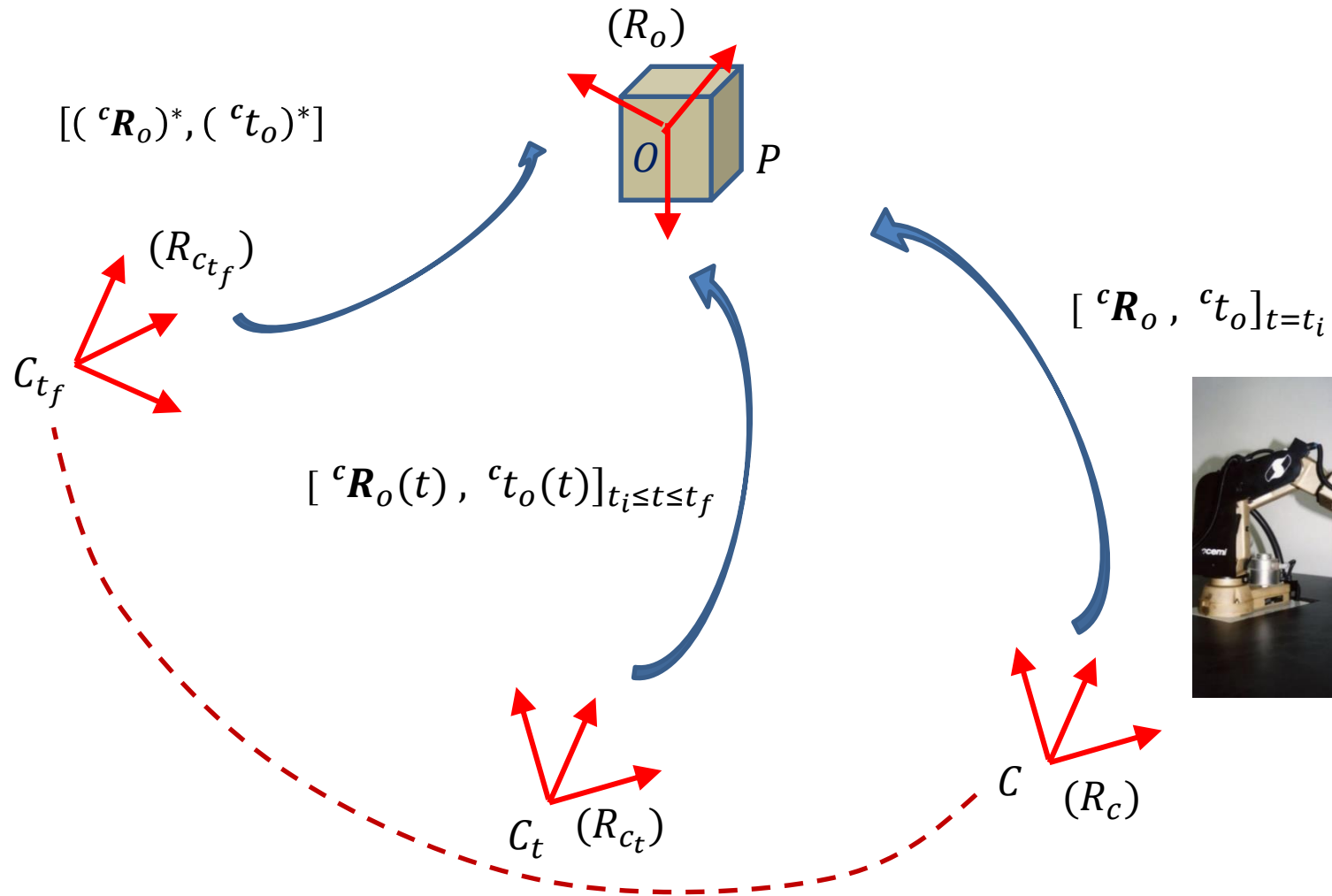
- Remarque : souvent on peut se ramener au problème où $r^* = 0$.





3.3 Asservissement visuel basé position

➤ AV fondé sur la « pose » de la caméra





3.3 Asservissement visuel basé position

➤ AV fondé sur la « pose » de la caméra

- Remarques :

- ✓ il n'y a qu'un seul « temps final » $t = t_f$ qui est défini ici car on considère que l'orientation tend vers l'orientation désirée en même temps que la position tend vers la position désirée, puisque c'est le vecteur d'état tout entier qui est la variable vectorielle à asservir. Cependant il est possible que l'une soit obtenue avant l'autre, ou encore de mettre en oeuvre un AV séquentiel ou partitionné (mais il faut garantir la convergence et la stabilité).
- ✓ Grâce au **recalage** entre le capteur de vision (le plus souvent une caméra) et l'objet d'intérêt dont les images sont capturées, une **tâche de positionnement** d'un système articulé peut être envisagée, même si l'objet d'intérêt est en mouvement. Ce dernier aspect, dynamique, n'est possible que si **l'objet d'intérêt reste visible** par le capteur de vision pendant tout son déplacement.



3.3 Asservissement visuel basé position

➤ AV fondé sur la « pose » de la caméra

- ✓ On définit alors la fonction de tâche $e(t)$ telle que $e(t) = s(t) - s^*$ et un torseur cinématique τ_c tel que $\dot{e}(t) = L_s \tau_c$ où :

$$L_s = \begin{bmatrix} -I_{3 \times 3} & [{}^c t_o]_{\times} \\ 0_{3 \times 3} & L_{\theta u} \end{bmatrix} \quad \text{et} \quad L_{\theta u} = I_3 - \frac{\theta}{2} [u]_{\times} + \left(1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2(\frac{\theta}{2})}\right) [u]_{\times}^2$$

- Remarque : $L_{\theta u}$ est telle que $L_{\theta u}^{-1} \theta u = \theta u$ puisque $L_{\theta u} u = u$

- ✓ Donc, on aura finalement : $\tau_c = -\lambda \widehat{L_s^{-1}} e$, où $L_s^{-1} = \begin{bmatrix} -I_{3 \times 3} & [{}^c t_o]_{\times} L_{\theta u}^{-1} \\ 0_{3 \times 3} & L_{\theta u}^{-1} \end{bmatrix}$

c'est-à-dire plus simplement ($r(t) = \theta u$):

$$\begin{cases} v_c = -\lambda [({}^c t_o)^* - {}^c t_o] + [{}^c t_o]_{\times} (\theta u - L_{\theta u}^{-1} r^*) \\ \omega_c = -\lambda (\theta u - r^*) \end{cases}$$

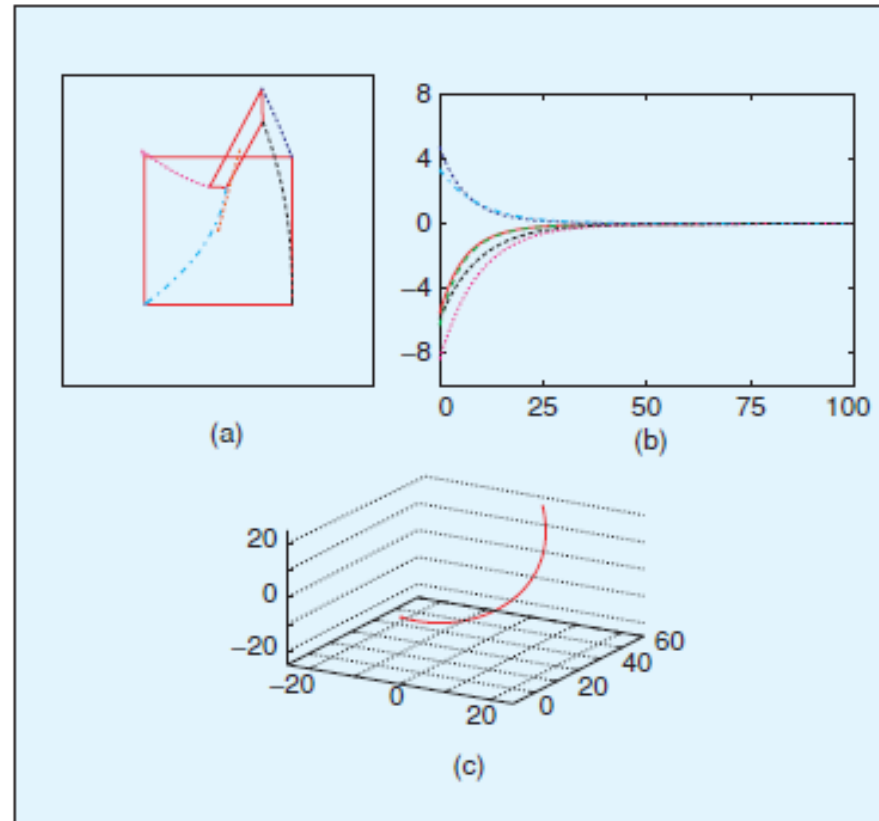
- Remarques :

- ✓ On observe un **découplage partiel** entre les commandes cinématiques de position et d'orientation.
- ✓ La commande s'appuie sur les calculs successifs des positions et des rotations (recalage rigide).
- ✓ L'expression de la commande se simplifie si on choisit $r^* = 0$, c'est-à-dire $({}^c R_o)^* = I_3$.



3.3 Asservissement visuel basé position

➤ AV fondé sur la « pose » de la caméra



Asservissement visuel basé position avec $s(t) = ({}^c t_o \quad \theta \quad u)^T$ et recalage 2-D/3-D rigide (camera pose). On observe que chaque composante du torseur cinématique a un comportement exponentiel décroissant et que l'image de l'origine du repère objet suit une ligne droite.



3.3 Asservissement visuel basé position

➤ AV fondé sur le déplacement relatif de la caméra

Pour cette approche, on définit le vecteur d'état sans faire référence à l'objet, mais directement entre la configuration actuelle et la configuration désirée (ici, on note : ${}^c R_{c_t} = \exp([{}^c r_{c_t}(t)]_x)$) :

$$s(t) = \left({}^c t_{c_t}(t) \quad {}^c r_{c_t}(t) \right)^T \quad \text{et} \quad s^* = (0 \quad 0)^T .$$

et un torseur cinématique τ_c tel que $\dot{s}(t) = L_s \tau_c$ où $L_s = \begin{bmatrix} {}^c R_{c_t} & 0_{3 \times 3} \\ 0_{3 \times 3} & L_{\theta u} \end{bmatrix}$

Donc, en souhaitant un comportement exponentiel décroissant de l'erreur, on a : $\tau_c = -\lambda \widehat{L_s^{-1}} e(t)$, c'est-à-dire plus simplement :

$$\begin{cases} v_c = -\lambda ({}^c R_{c_t})^T {}^c t_{c_t} \\ \omega_c = -\lambda {}^c r_{c_t}(t) \end{cases}$$

➔ **découplage total** entre les commandes cinématiques en position et en orientation.

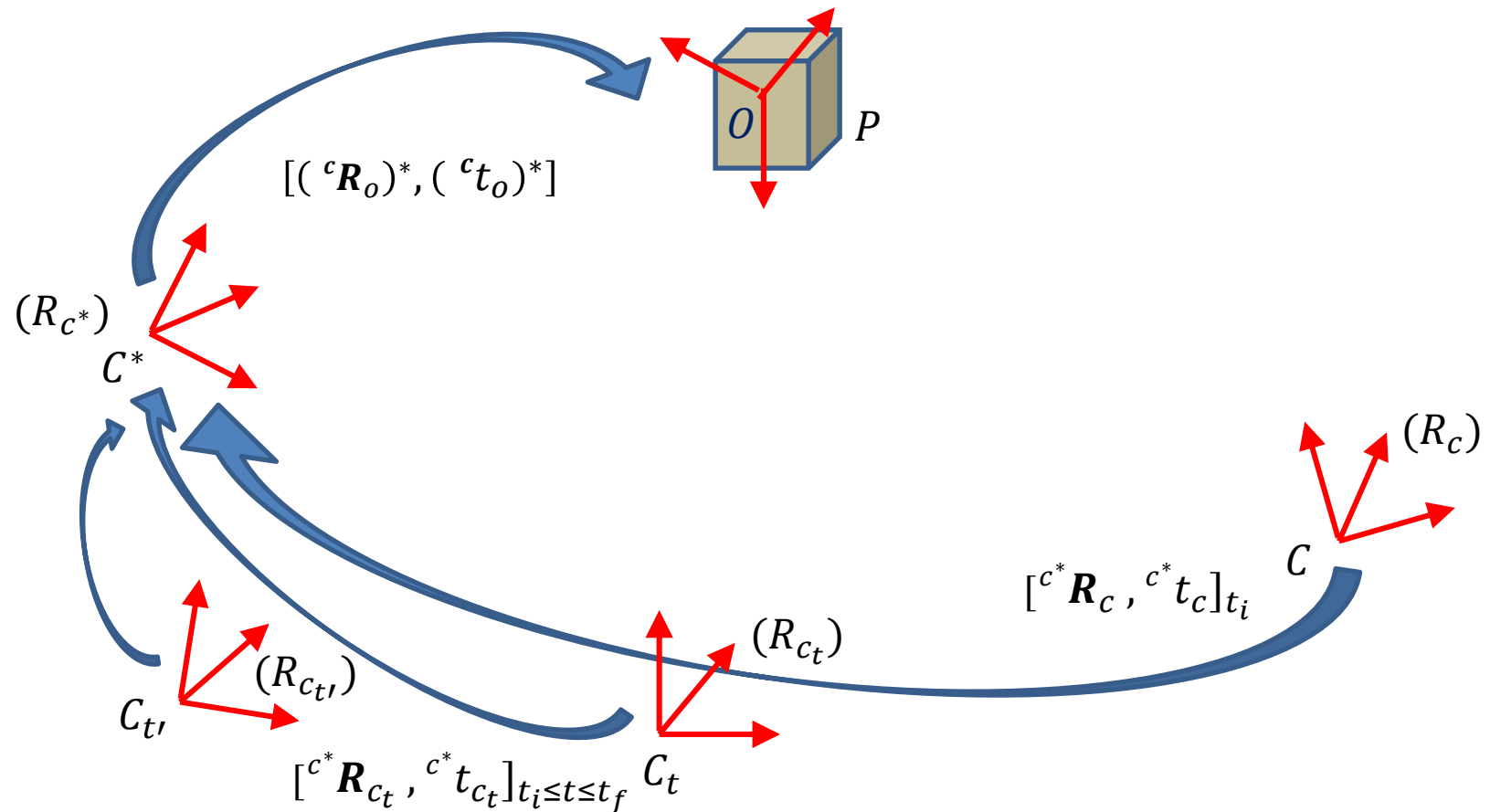
Remarques :

- Le découplage total permet d'envisager des asservissements en position et en orientation avec des gains λ différents, voire des asservissements séquentiels.
- Noter les différences de trajectoires entre les deux fonctions de tâche réalisées.



3.3 Asservissement visuel basé position

➤ AV fondé sur le déplacement relatif de la caméra



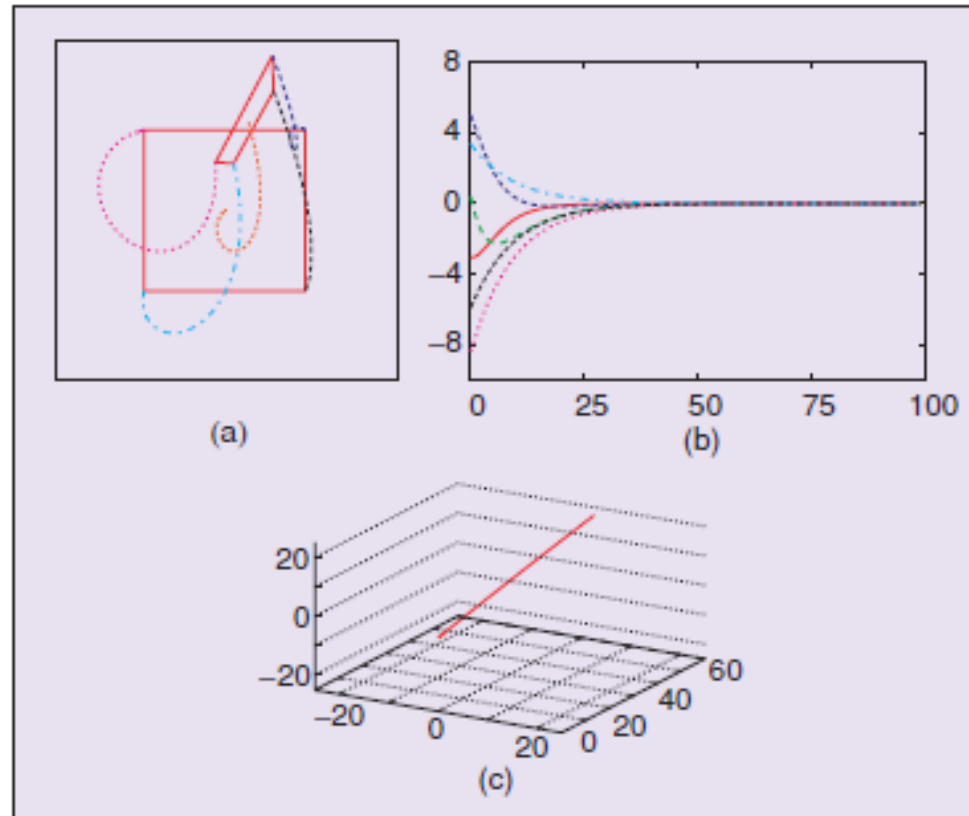
- Les formules suivantes indiquent les relations entre recalages et déplacement relatif de la caméra :

$${}^{c^*}R_c = {}^{c^*}R_o \circ R_c \quad \text{et} \quad {}^{c^*}t_c = C^* C_{(/R_{c^*})} = {}^{c^*}t_o - {}^{c^*}R_c {}^c t_o$$



3.3 Asservissement visuel basé position

➤ AV fondé sur le déplacement relatif de la caméra



Asservissement visuel basé position avec $s(t) = \begin{pmatrix} {}^c t_{ct}(t) & {}^c r_{ct}(t) \end{pmatrix}^T$ et estimation du déplacement relatif de la caméra. On observe que la trajectoire de l'origine du repère objet, O , est rectiligne dans l'espace mais que l'image de O ainsi que les autres projections ont des trajectoires incurvées, et peuvent sortir du champ de vision.

Visual Servo Control Part I: Basic Approaches, F. Chaumette, S. Hutchinson,
IEEE Robotics and Automation Mag, Décembre 2006



3.4 Stabilité des asservissements (*une introduction*)

- Etant donné une fonction $h(t, s)$ de la variable vectorielle s et du temps t , $h : U \times \mathbb{R} \rightarrow \mathbb{R}^k$, continue, et U est un ouvert de \mathbb{R}^k . k est le nombre de composantes de s :
- Si $h(t, s) = f(s(t))$, la fonction f , telle que $f : \mathbb{R} \rightarrow \mathbb{R}^k$, est autonome sur l'ouvert U de \mathbb{R}^k .
 - Si f est continue et différentiable, le système dynamique autonome $\dot{s} = f(s(t))$ admet un point d'équilibre s^* si $f(s^*) = 0$.
 - Si toutes les valeurs propres de la matrice $\frac{\partial f}{\partial s^T}$ en s^* sont à partie réelle (resp. strictement) négative, le point d'équilibre s^* (solution de $\dot{s} = f(s(t))$) est (resp. **asymptotiquement**) **stable**.
 - Dans le cas où on a simplement l'équation $\dot{s} = A s$, où A est une matrice (constante), on a la réciproque : la solution nulle est asymptotiquement stable si et seulement si toutes les valeurs propres de A sont à partie réelle strictement négative. Si on s'intéresse à la **stabilité**, alors on peut autoriser A à avoir des valeurs propres à partie réelle nulle, mais leur multiplicité comme racine du polynôme minimal doit alors être égale à 1.



3.4 Stabilité des asservissements (*une introduction*)

- Dans la section 3.1, nous avons vu que l'analyse cinématique du champ de mouvements amenait à considérer le système dynamique vectoriel $\dot{e}(t) = -\lambda L_s \widehat{L}_s^+ e(t)$ quand un comportement exponentiel décroissant de la fonction de tâche était souhaité ($\dot{e}(t) = -\lambda e(t)$).
- Le **formalisme de Lyapunov** permet d'analyser la stabilité d'un point d'équilibre. Une fonction de Lyapunov est une fonction $\Lambda(s) : \mathbb{R}^k \rightarrow \mathbb{R}$ strictement positive et continue, dont la dérivée temporelle $\dot{\Lambda}(s)$ est négative ou nulle. Ainsi, la norme de la fonction de tâche est souvent prise comme fonction de Lyapunov Λ candidate (mais d'autres choix sont possibles !) :

$$\Lambda(s(t)) = \frac{1}{2} \|s(t) - s^*\|^2 = \frac{1}{2} \|e(t)\|^2 = \frac{1}{2} e(t)^T e(t) ,$$

dont la dérivée s'écrit : $\dot{\Lambda}(s) = \frac{\partial \Lambda}{\partial s^T} \dot{s} = \frac{\partial \Lambda}{\partial s^T} f(s) \underset{\text{ici}}{=} \dot{e}(t)^T e(t) = -\lambda e(t)^T L_s \widehat{L}_s^+ e(t)$, avec $\lambda > 0$,
fonction dérivée qui doit être négative (selon la stabilité du point d'équilibre au sens de Lyapunov).

La condition $L_s \widehat{L}_s^+ > 0$ (matrice définie positive) équivaut à la **stabilité asymptotique** du point d'équilibre, car correspondant à la dérivée d'une fonction de Lyapunov qui est strictement négative $\dot{\Lambda}(s) < 0$. La stabilité asymptotique globale (GAS) est obtenue s'il n'y a qu'un seul point d'équilibre.

- $L_s \widehat{L}_s^+ > 0$ est vraie si toutes les valeurs propres de ce produit matriciel sont strictement positives.



3.4 Stabilité des asservissements (*une introduction*)

- Stabilité des AV3D (PBVS) : étant donné que la matrice $L_{\theta u}$ n'est pas singulière quand $\theta \neq 2k\pi$, on obtient la stabilité asymptotique globale pour ce type d'asservissement, car sous l'hypothèse forte que tous les paramètres de la pose de la caméra aient été parfaitement identifiés, on a $L_s \widehat{L_s}^{-1} = I_6$. Ceci est vrai pour les deux catégories de méthode AV3D que nous avons étudié auparavant, et pour lesquelles la dimension k du vecteur d'état est la même que celle du torseur cinématique ($k = m = 6$ au plus).
- Stabilité des AV2D (IBVS) : le vecteur d'état étant généralement de dimension $k > m$, la stabilité asymptotique est assurée dans un voisinage de $e(t_f) = e^* = 0$, mais comme il peut exister plusieurs minima locaux (configurations où $e^* \in \text{Ker}\{\widehat{L_s^+}\}$), celle-ci n'est pas globale mais locale.



3.5 Asservissements visuels

❑ EXERCICE D'APPLICATION (1/4)

Dans ce problème, nous allons étudier la commande par retour visuel d'un bras robotique composé de deux corps reliés par deux articulations rotoïdes (pivots) d'axes perpendiculaires, plus familièrement appelé tourelle pan-tilt (Figure 1). Une caméra perspective et à interface numérique pour la sortie du signal vidéo, munie d'un objectif de focale fixe sans distorsion, est montée sur l'organe terminal, et on supposera que son centre de projection, C , est placé à l'origine de l'organe terminal équivalent au deuxième corps (C_2), O_2 (voir Figure 2). Cependant le schéma de la Figure 2 montre que les repères caméra (R_c) et du corps (C_2), (R_2), n'ont pas la même orientation. Les paramètres intrinsèques de la caméra sont les composants de la matrice \mathbf{K} de ce système de vision monoculaire dans la relation qui relie les coordonnées homogènes d'un point géométrique de la scène P et sa projection Q dans l'image

$$Q = w \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K} [\mathbf{R} \ t] P \quad (\text{eq 1})$$

\mathbf{R} et t sont respectivement la matrice de rotation et le vecteur de translation entre le repère caméra (R_c) et le repère (R_o) attaché à l'objet de la scène dont P est issu, et sont exprimés dans le repère caméra (R_c). $\mathbf{R} = I$ (identité) et $t = (0,0,0)^T$ si les coordonnées de $P = ({}^c x, {}^c y, {}^c z, 1)^T$ sont exprimées dans (R_c).

Figure 1 : tourelle pan-tilt



Pour les caméras numériques non distordue, la matrice \mathbf{K} est telle que $\mathbf{K} = \begin{bmatrix} f/l_x & 0 & u_c \\ 0 & f/l_y & v_c \\ 0 & 0 & 1 \end{bmatrix}$ où l_x et l_y sont les valeurs des dimensions d'un photo-élément, f est la focale de l'objectif de la caméra, et u_c et v_c sont les coordonnées inhomogènes (en pixels) du point principal. On suppose que le centre de l'image coïncide avec le point principal.



3.5 Asservissements visuels

□ EXERCICE D'APPLICATION (2/4)

Pour la suite du problème, on définit le vecteur d'état $s = (u, v)^T$ à partir des coordonnées inhomogènes de Q , donc dont l'unité est le pixel, et on définit aussi les coordonnées homogènes $\tilde{Q} = K^{-1}Q = w \begin{pmatrix} {}^s x \\ {}^s y \\ 1 \end{pmatrix}^T$ ($w = {}^c z$), \tilde{Q} est un point qui appartient au plan du capteur (s : sensor). Enfin, $(V_{R_c}^P)_{(R_c)}$ est la vitesse du point P par rapport à la caméra, et exprimée dans (R_c) .

Quand $R = I$ (identité) et $t = (0,0,0)^T$ - repères objet et caméra confondus - les coordonnées homogènes de P vérifient :

$$P = C + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tilde{Q}. \rightarrow \text{les } \underline{\text{coordonnées inhomogènes}} \text{ de } P - C \text{ sont égales aux coordonnées homogènes de } \tilde{Q}.$$

Question 1 : D'après le cours (chapitre 3), montrez que si les paramètres intrinsèques sont constants, la dérivée du vecteur d'état, \dot{s} , vaut :

$$\dot{s} = \frac{1}{c_z} D \begin{bmatrix} 1 & 0 & -{}^s x \\ 0 & 1 & -{}^s y \end{bmatrix} (V_{R_c}^P)_{(R_c)} \quad \text{où} \quad D = f \begin{bmatrix} 1/l_x & 0 \\ 0 & 1/l_y \end{bmatrix} \text{ est une matrice } (2 \times 2) \text{ diagonale constante.}$$

Question 2 : La vitesse de déplacement de la caméra embarquée est représentée par le torseur $\tau_c = (v_c, \omega_c)$ regroupant les vitesses de translation $v_c = (V_{R_f}^C)_{(R_c)}$ et de rotation $\omega_c = ({}^f \Omega_c)_{(R_c)}$ du repère (R_c) par rapport à un repère fixe (R_f) (qui peut être attaché à l'objet d'intérêt), et exprimé dans le repère (R_c) . Montrez qu'à partir de l'expression de la vitesse du point P en fonction des composantes du torseur cinématique τ_c de la caméra on a :

$$\dot{s} = D \frac{1}{c_z} \begin{bmatrix} 1 & 0 & -{}^s x \\ 0 & 1 & -{}^s y \end{bmatrix} [-I \quad ; \quad [P - C]_{\times}] \tau_c = L_s \tau_c$$

(où $P - C$ désigne le vecteur construit à partir des coordonnées inhomogènes de P et C dont les coordonnées homogènes valent $C = (0,0,0,1)^T$. Mettre L_s sous la forme d'un produit matriciel $L_s = D L$).



3.5 Asservissements visuels

❑ EXERCICE D'APPLICATION (3/4)

Donner l'expression des composantes de L en fonction uniquement des coordonnées de \tilde{Q} , c'est-à-dire les trois paramètres ${}^s x$, ${}^s y$ et ${}^c z$.

Question 3 : En examinant la modélisation de la Figure 2, déterminer la matrice de rotation 2R_c entre le repère (R_2) et le repère (R_c) . En déduire les valeurs des composantes la matrice de changement de repères pour les torseurs, notée 2G_c , permettant d'exprimer le torseur τ_c dans le repère (R_2) , $(\tau_c)_{(R_2)}$.

Question 4 : De par l'architecture de la chaîne cinématique de la tourelle sur laquelle la caméra est montée (voir Figure 2 ci-après), montrez que le torseur τ_c est fonction des deux variables articulaires θ_1, θ_2 et de leur dérivée temporelle (vitesses articulaires) $\dot{\theta}_1$ et $\dot{\theta}_2$, sous la forme $(\tau_c)_{(R_2)} = J(\theta_1, \theta_2) \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}$, où $J(\theta_1, \theta_2)$ est la matrice jacobienne de la tourelle pan-tilt. Dans quel repère cette matrice jacobienne doit-elle être exprimée ici ? Quelles sont ses dimensions ?

Question 5 : On définit maintenant une fonction de tâche, notée $e(t)$, comme étant la différence entre l'état actuel et l'état désiré : $e(t) = s(t) - s^*$ (on a $s(t) = (u(t), v(t))^T$) quand la caméra observe un seul point P . Quelle est alors l'expression vectorielle de la dérivée de l'erreur en fonction de $L, D, J, \dot{\theta}_1, \dot{\theta}_2$?

Question 6 : En supposant que l'on souhaite un comportement exponentiel décroissant de l'erreur e ($\dot{e} = -\lambda e$; $\lambda > 0$), donner l'expression de la commande cinématique, représentée par les vitesses articulaires de la tourelle, en fonction de λ, L, J et D pour atteindre l'état désiré et constant s^* au centre de l'image.



3.5 Asservissements visuels

❑ EXERCICE D'APPLICATION (4/4)

Question 7 : Quelle est la raison principale qui a pu motiver de choisir le vecteur d'état s comme étant composé des coordonnées inhomogènes de Q , $(u, v)^T$, plutôt qu'à partir des coordonnées inhomogènes de \tilde{Q} , $({}^s x, {}^s y)^T$?

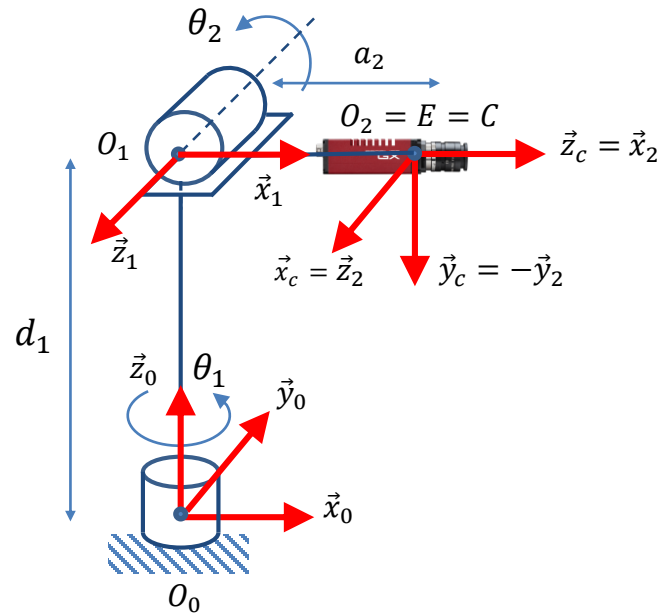


Figure 2 : schématisation de la tourelle pan-tilt