

---

# TPE-ISAV

## Asservissement visuel et commande

Rédigé par:

**Nkimo Lenou Randolph M1-ASI**

Supervisé par:

**Zanne Philippe**

Mai 2023

# 1 Méthodologie

Dans cette partie nous explictons la démarche suivie pour réaliser l'asservissement visuel.

## 1.1 Traitement de l'image

Cette étape a pour but de détecter sur l'image acquise par la caméra (capteur) la position des marqueurs à chaque instant (à chaque acquisition d'une nouvelle image), dans notre asservissement toute les 40ms.

Les marqueurs sont dectectés par seuillage sur la valeur du pixel puis nous déterminons les coordonnées du centre de gravité de chaque marqueur  $c = [c_x \ c_y]^T$  connaissant à priori les coordonnées de l'origine  $[aoi_{x0} \ aoi_{y0}]^T$  et dimensions  $[aoi_{dx} \ aoi_{dy}]^T$  de l'imagette dans laquelle il se trouve.

$$c = \frac{1}{\sum_{i,j} poid_{s,j,i}} \begin{bmatrix} \sum_{i=1}^{aoi_{x0}+aoi_{dx}} \sum_{j=1}^{aoi_{y0}+aoi_{dy}} poid_{s,j,i} * i \\ \sum_{i=1}^{aoi_{x0}+aoi_{dx}} \sum_{j=1}^{aoi_{y0}+aoi_{dy}} poid_{s,j,i} * j \end{bmatrix}$$

avec  $poid_{s,j,i} = 255 - pixel_{j,i}$  ainsi les pixels les plus sombres auront une plus grande pondération.

La mise à jour de la position de l'imagette permet à chaque nouvelle image d'avoir le centre de l'imagette confondu avec le centre de gravité du marqueur.  $\begin{bmatrix} aoi_{x0} \\ aoi_{y0} \end{bmatrix} = \begin{bmatrix} c_x - \frac{1}{2}aoi_{dx} \\ c_y - \frac{1}{2}aoi_{dy} \end{bmatrix}$ .

## 1.2 Reconstruction de la mesure

Dans le cas d'un asservissement visuel 3D, la mesure utile pour l'asservissement est la position cartésien de la cible par rapport à la camera.

A partir du modèle du sténopé de la caméra on determine la position d'un point dans le plan image **repère capteur (u,v)** en fonction de celles d'un point  $M = [x_c \ y_c \ z_c]^T$  se trouvant dans un repère quelconque en particulier dans notre cas **repère caméra**  $R_c$ , du facteur d'agrandissement  $\alpha$  et des coordonnées du centre optique  $[u_0 \ v_0]^T$  de la caméra dans le repère capteur.

$$\begin{cases} u = \alpha_u \frac{x_c}{z_c} + u_0 \\ v = \alpha_v \frac{y_c}{z_c} + v_0 \end{cases}$$

Connaissant à priori la transformation qui nous permet de quitter du repère caméra  $R_c$  au repère objet  $R_o$ , dans ce cas une rotation selon axe z  $R_{co}$  et une translation  $t_{co}$  suivant les axes x,y,z, on exprime les coordonnées images  $p = [u \ v]^T$  de la projection d'un point M en fonction des paramètres intrinsèques de

la caméra, de ces coordonnées  $[x_o \ y_o \ z_o]^T$  dans le repère objet, de  $R_{co} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$  et

$$t_{co} = [t_x \ t_y \ t_z]^T.$$

$$\begin{cases} u = \alpha_u \left( \frac{\cos(\alpha)}{t_z} x_0 + \frac{-\sin(\alpha)}{t_z} y_0 + \frac{t_x}{t_z} \right) + u_0 \\ v = \alpha_v \left( \frac{\sin(\alpha)}{t_z} x_0 + \frac{\cos(\alpha)}{t_z} y_0 + \frac{t_y}{t_z} \right) + v_0 \end{cases}$$

; Avec un point on a un systeme de deux(02) équations à quatre(04) inconnues, pour avoir un système singulier c'est à dire quatre(04) équations et quatre(04) à inconnues on choisira au minimum **deux(02) points**.

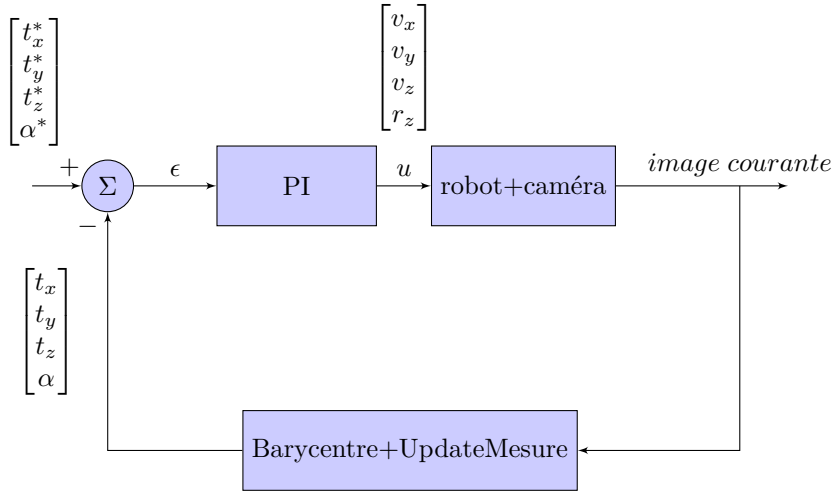
En posant  $a_1 = \frac{\cos(\alpha)}{t_z}$ ,  $a_2 = \frac{\sin(\alpha)}{t_z}$ ,  $a_3 = \frac{t_x}{t_z}$  et  $a_4 = \frac{t_y}{t_z}$ , on modélise ce système sous forme de matrices

comme suit: 
$$\begin{bmatrix} u_{01} - u_0 \\ v_{01} - v_0 \\ u_{02} - u_0 \\ v_{02} - v_0 \end{bmatrix} = \begin{bmatrix} \alpha_u x_0 & -\alpha_u y_0 & \alpha_u & 0 \\ \alpha_v y_0 & \alpha_u x_0 & 0 & \alpha_v \\ \alpha_u x_1 & -\alpha_u y_1 & \alpha_u & 0 \\ \alpha_v y_1 & \alpha_u x_1 & 0 & \alpha_v \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}$$
 qui est sous la forme  $b = Ma$ , donc  $a = M^{-1}b$

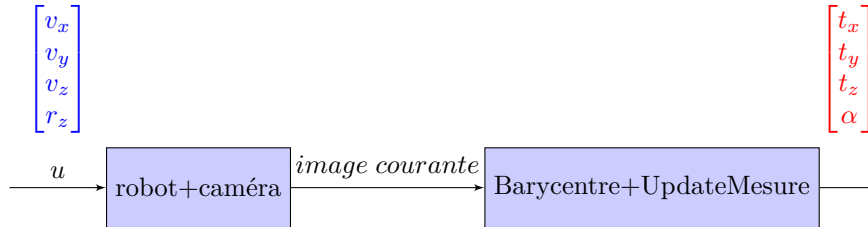
et l'on retrouve la position de la cible en déduisant les paramètres  $t_x$   $t_y$   $t_z$  et  $\alpha$  :  $\alpha = atan(\frac{a_2}{a_1})$ ,  $t_z = \frac{cos(\alpha)}{a_1}$ ,  $t_x = t_z a_3$ ,  $t_y = t_z a_4$ .

### 1.3 Commande

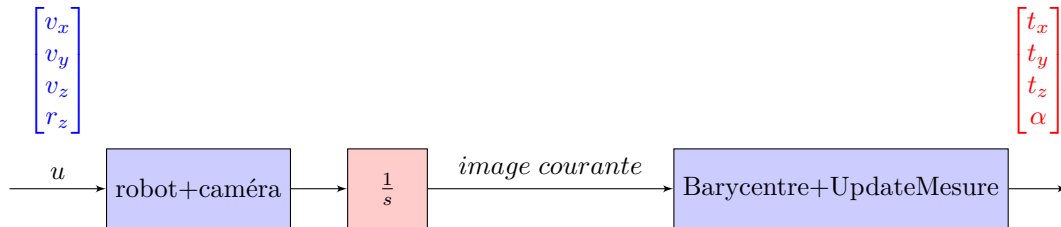
L'asservissement de notre système passe par la définition d'un cahier de charges, nous considérons la plus simple des exigences, avoir un **système précis**(erreur statique nulle). La prise en compte de cette exigence nous enmène à définir comme correcteur un **Proportionnel-Intégrateur(PI)**, ce qui inclut dans notre boucle ouverte la présence d'au moins un intégrateur, donc la certitude d'avoir une erreur statique nulle pour une entrée de type échelon. Le schéma d'asservissement est le suivant:



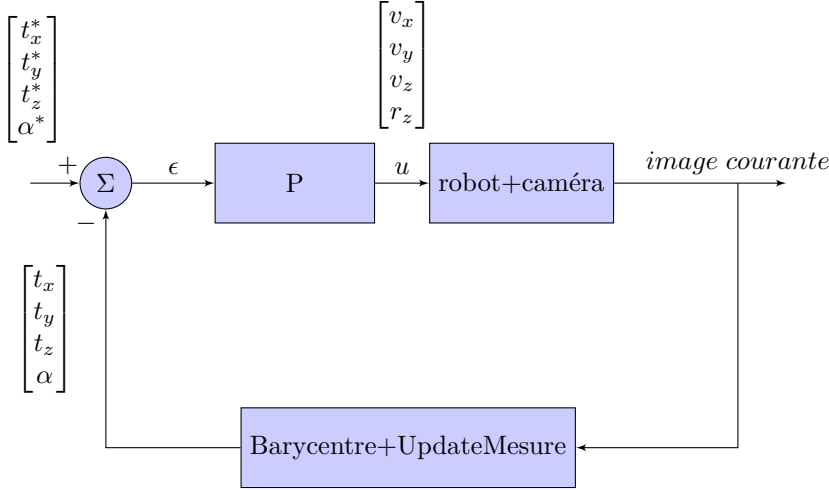
En observant notre boucle d'asservissement, plus exactement ce bloc:



On a en entrée des vitesses et en sortie des positions, ce qui nous permet de conclure sur la présence d'un **intégrateur** dans cette chaîne.



Ainsi on choisira comme correcteur un **Proportionnel**:



La commande se calcule aisément comme suit:  $\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = gain_{trans} \cdot \begin{bmatrix} t_x^* - t_x \\ t_y^* - t_y \\ t_z^* - t_z \end{bmatrix}$  et  $r_z = gain_{rot}[\alpha^* - \alpha]$ .

**⚠ Warning:** Les vitesses envoyées(commande) sont celles du centre de rotation de l'objet dans le repère caméra( $v_{O_o|R_c}$ ), cependant le centre de rotation de l'objet( $O_o$ ) et de l'effecteur( $O_E$ ) sont disjoints de ce fait un mouvement de rotation de l'effecteur entraine la rotation des marqueurs mais aussi une translation suivant les axes x, y.

**✓ Solution:** Pour remédier à ce problème on calcule les nouvelles commandes qui sont les vitesses du centre de rotation de l'effecteur ( $v_{O_E|R_c}$ ) dans le repère camera.

Avec les relations de cinématique du point on a:

$$v_{O_o|R_c} = v_{O_E|R_c} + \overrightarrow{O_oO_E}|_{R_c} \times \vec{\omega} \implies v_{O_E|R_c} = v_{O_o|R_c} - \overrightarrow{O_oO_E}|_{R_c} \times \vec{\omega}, \text{ dans cette expression l'inconnue est}$$

$$\overrightarrow{O_oO_E}|_{R_c} = R_{co} \cdot \overrightarrow{O_oO_E}|_{R_o}; \text{ avec } \overrightarrow{O_oO_E}|_{R_o} = [p_x \ p_y \ 0]^T, \ v_{O_E|R_c} = [v_x \ v_y \ v_z]^T \text{ et } \vec{\omega} = [0 \ 0 \ r_z]^T$$

$$\overrightarrow{O_oO_E}|_{R_c} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_x \\ p_y \\ 0 \end{bmatrix} = \begin{bmatrix} p_x \cos(\alpha) - p_y \sin(\alpha) \\ p_x \sin(\alpha) + p_y \cos(\alpha) \\ 0 \end{bmatrix}$$

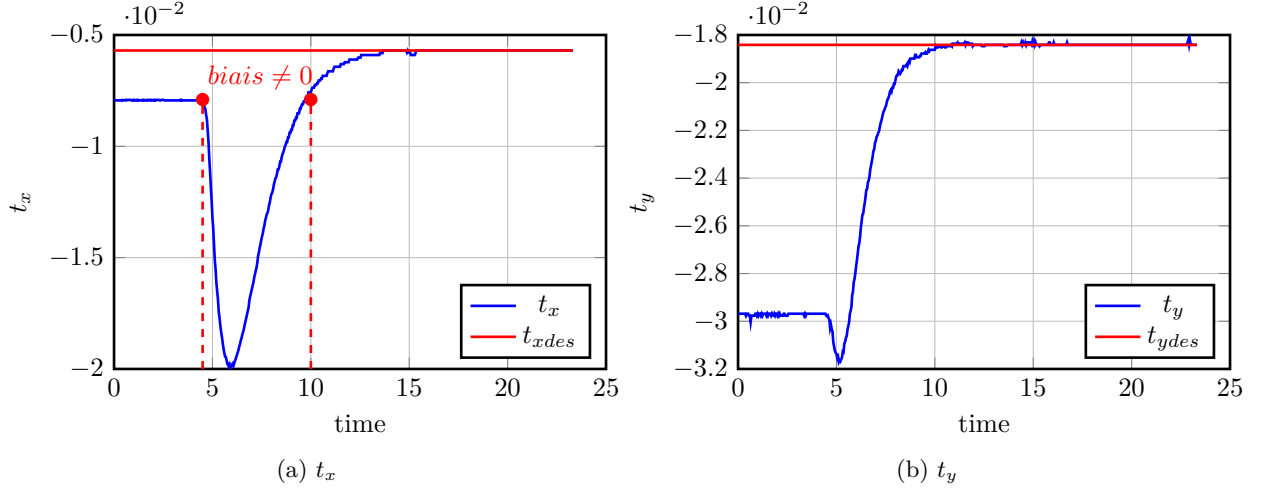
$$\overrightarrow{O_oO_E}|_{R_c} \times \vec{\omega} = \begin{bmatrix} p_x \cos(\alpha) - p_y \sin(\alpha) \\ p_x \sin(\alpha) + p_y \cos(\alpha) \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ r_z \end{bmatrix} = \begin{bmatrix} r_z(p_x \sin(\alpha) + p_y \cos(\alpha)) \\ r_z(p_y \sin(\alpha) - p_x \cos(\alpha)) \\ 0 \end{bmatrix}$$

$$\text{La nouvelle commande est donc } u = \begin{bmatrix} v_x - r_z(p_x \sin(\alpha) + p_y \cos(\alpha)) \\ v_y - r_z(p_y \sin(\alpha) - p_x \cos(\alpha)) \\ v_z \\ r_z \end{bmatrix}$$

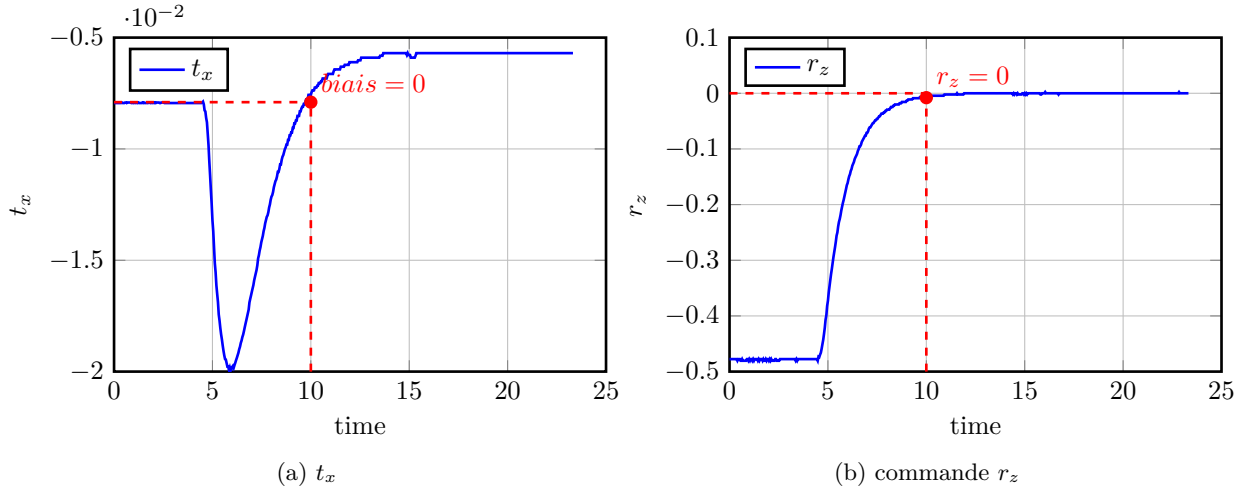
## 2 Analyse et interpretation des résultats

Les courbes tracées ci-dessous sont celles obtenu avec les *données expérimentales*, dans ce cas la convolution entre la rotation de l'effecteur et la translation des marqueur n'a pas été prise en compte, et nous allons observer le comportement de la réponse du système et la commande.

On observe (*Figure 1 – (a)*) une réponse qui converge (vers la valeur de référence  $t_{xdes}$ ) en a peu près **13 secondes**; Mais cependant entre 4,5 *secondes* – 10 *secondes* nous avons un comportement anormal ceci

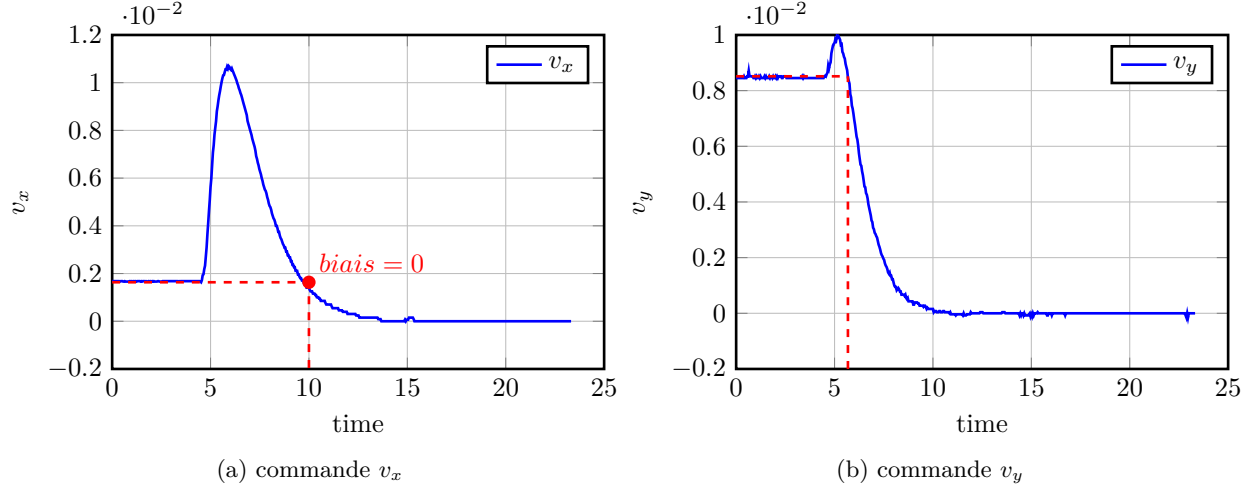
Figure 1: Illustration du comportement des réponses  $t_x$  et  $t_y$ 

est dû au **biais** de translation introduit par la convolution entre rotation de l'effecteur et la translation de l'objet, comme expliqué plus haut on a:  $v_{O_o}|_{R_c} = v_{O_E}|_{R_c} + \overrightarrow{O_o O_E}|_{R_c} \times \begin{bmatrix} 0 \\ 0 \\ r_z \end{bmatrix}$  le **biais** devient **nul** pour  $r_z = 0$  autrement dit quand la commande en rotation converge. Ainsi en analysant le graphe de la commande  $r_z$  (Figure 2) on doit s'attendre à ce qu'elle converge à l'instant  $t \approx 10$  secondes

Figure 2: Illustration du comportement de la réponse  $t_x$  en fonction du biais

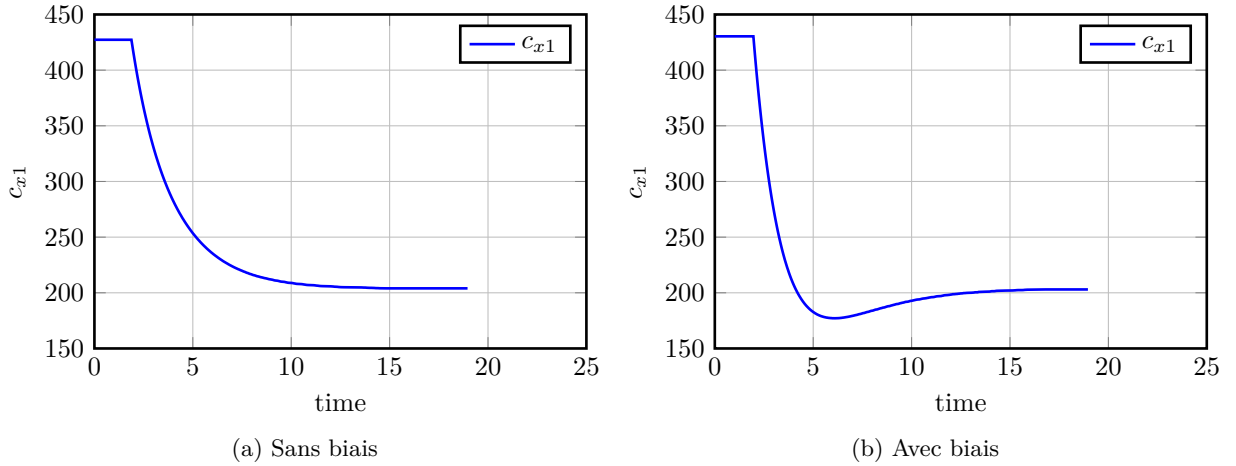
On peut donc prédire aussi ce comportement sur les commandes  $v_x$  et  $v_y$  (Figure 3).

Concernant la commande  $v_y$  (Figure 3) on constate qu'elle reprend son comportement normal avant que la rotation  $r_z$  ne converge [ $r_z = 0$ ] (avant 10 secondes), ceci peut être expliqué par le fait que la rotation entraînée par l'effecteur n'a engendré qu'une très petite translation en  $y$  sur les marqueurs et l'on peut l'observer avec les amplitudes des dépassements sur la commande en  $v_x$  qui est nettement plus importante que celui de la commande en  $v_y$ .

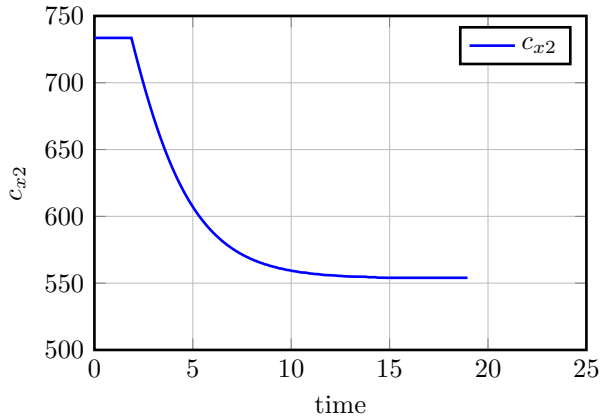
Figure 3: Illustration du comportement des commandes  $v_x$  et  $v_y$  en fonction du biais

Corrigeons le biais introduit par la convolution entre la rotation de l'effecteur et la translation de l'objet et observons le comportement des centres de gravité des marqueurs  $[c_x \ c_y]^T$  dans le cas avec biais et sans biais (Figure 4, 5, 6, 7).

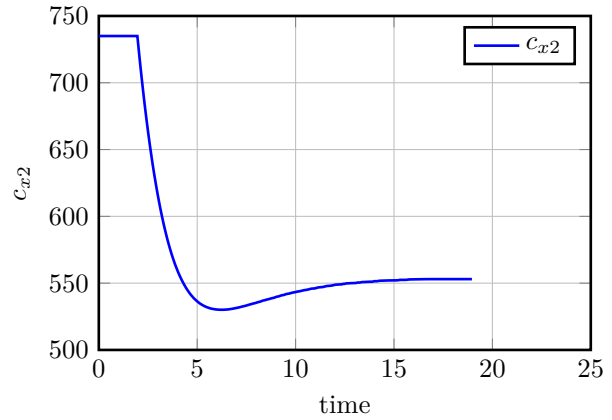
Les données sont celles du simulateur et pour mettre en évidence la correction de la commande, le mouvement du robot n'est qu'une simple rotation.

Figure 4: Centre de gravité  $c_x$  du marqueur 1

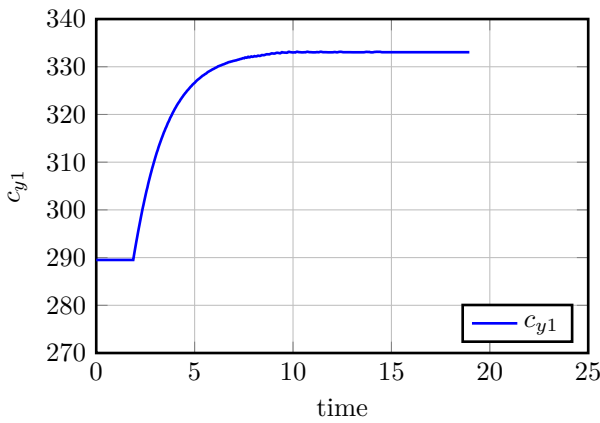
On constate clairement que la correction du biais sur le commande améliore nettement les performances de l'asservissement avec la suppression du dépassement. Le comportement en simulation pour les deux cas (*vidéo*) est disponible [ici](#).



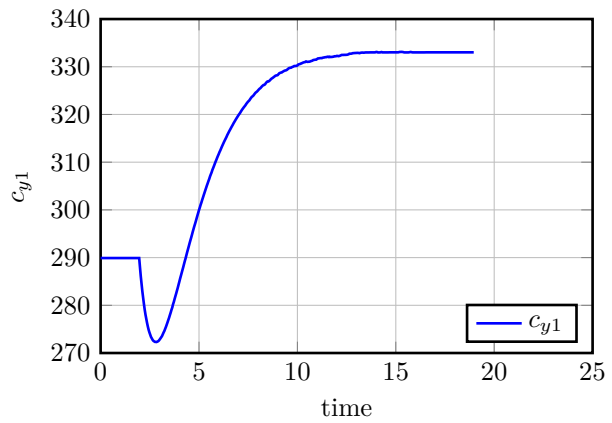
(a) Sans biais



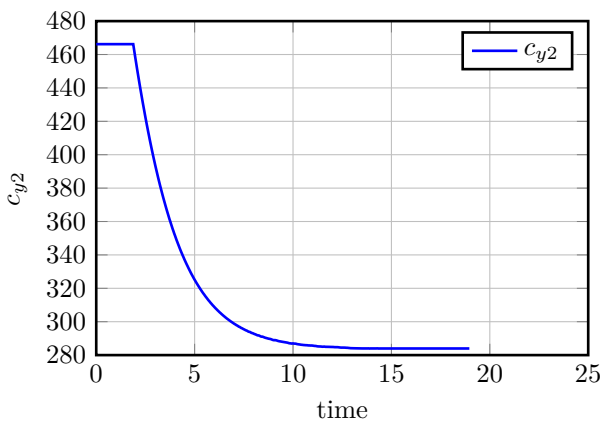
(b) Avec biais

Figure 5: Centre de gravité  $c_x$  du marqueur 2

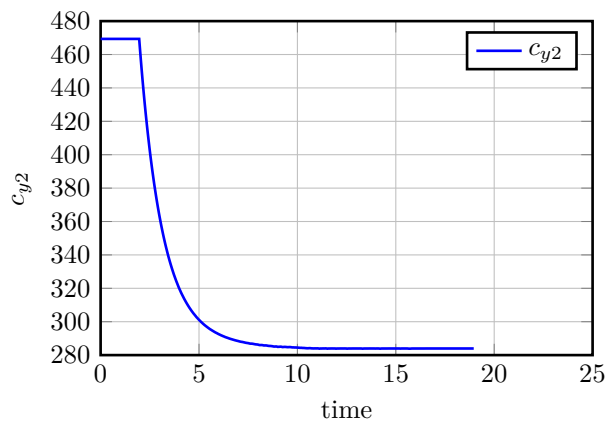
(a) Sans biais



(b) Avec biais

Figure 6: Centre de gravité  $c_y$  du marqueur 1

(a) Sans biais



(b) Avec biais

Figure 7: Centre de gravité  $c_y$  du marqueur 2