



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

# Control de robots industriales con realimentación visual: Image-Based Visual Servoing

Solanes Galbis, Juan Ernesto ([juasogal@isa.upv.es](mailto:juasogal@isa.upv.es))

Gracia Calandín, Luis Ignacio ([luigraca@isa.upv.es](mailto:luigraca@isa.upv.es))

Departamento de Ingeniería de Sistemas y Automática  
Universitat Politècnica de València

## 1 Resumen de las ideas clave

Dentro del control cinemático de robots industriales tenemos el denominado control por re-alimentación visual en el plano de imagen, más conocido como *Image-Based Visual Servoing* (IBVS). Se trata de que el controlador utilice la información obtenida por un sistema de visión para posicionar la herramienta del robot industrial de forma que realice una manipulación del producto manufacturado. En la actualidad este tipo de control ha cobrado un especial interés por parte de la industria debido, principalmente, a dos factores: la capacidad de potencia de los actuales sistemas de visión artificial, que permiten obtener información en tiempo real del entorno de trabajo del robot; la necesidad de la industria de obtener aplicaciones automáticas flexibles que se adapten a cambios en la línea de producción.

En este artículo vamos a aprender los conceptos básicos del control IBVS que nos permitirán diseñar controladores cinemáticos de robots manipuladores industriales.

## 2 Introducción

El término *visual servoing*, (Chaumette y Hutchinson 2006; Chaumette y Hutchinson 2007; Chaumette y Hutchinson 2008), hace referencia a la tarea de controlar la posición del efector final de un robot relativo a un objeto utilizando características visuales extraídas a partir de técnicas de visión artificial, como podemos ver en el ejemplo de la [figura 1](#). Se trata pues de un campo de investigación muy amplio que asocia diferentes áreas de conocimiento que van desde la robótica y control de sistemas hasta el tratamiento de imagen y el aprendizaje por refuerzo (Burkov 2019).

Si nos centramos únicamente en la parte del control del robot, tenemos que tener en cuenta dos aspectos fundamentales: la **colocación de la cámara** dentro del sistema *visual servoing* y el **espacio de trabajo** dónde vamos a realizar el control.

Por lo que respecta a la colocación de la cámara, existen principalmente dos configuraciones: cuando la cámara esté situada en el efector final del robot (o en la herramienta) y vaya solidaria con el movimiento del mismo, entonces diremos que tenemos una configuración *eye-in-hand*; si por el contrario tenemos que la cámara se encuentra situada en una posición fija ajena al movimiento del robot, entonces diremos que tenemos una configuración *eye-to-hand*, (Corke 2011).

Por otro lado, con referencia al espacio dónde vamos a realizar el control, fundamentalmente también disponemos de dos posibilidades: realizar el control en el espacio Cartesiano  $\mathbb{R}^3$ , o realizar el control en el espacio de imagen  $\mathbb{R}^2$ . Estos dos espacios de trabajo definen los dos controladores fundamentales *visual servoing*: llamaremos **Position-Based Image Servoing** (PBVS) cuando el control lo realicemos en el espacio Cartesiano, mientras que lo llamaremos **Image-Based Image Servoing** (IBVS) cuando el control lo realicemos en el espacio de imagen.

En este artículo vamos ver en detalle el control IBVS con la configuración *eye-in-hand*.

### Sabías que...

Un servo-mecanismo (o más comúnmente llamado *servo*) es un dispositivo mecánico que utiliza información de su estado actual para corregirse en función del objetivo marcado. El término *servo* proviene del término en latín *servus* y fue utilizado por primera vez por J.J.L. Farcot en 1868 en su trabajo *Le Servomoteur* para describir los motores hidráulicos y de vapor empleados para comandar los barcos.

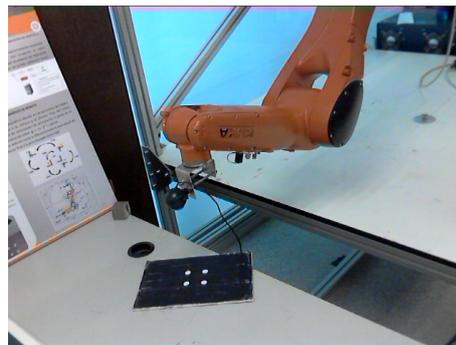
### 3 Objetivos

Al finalizar este documento, el alumno deberá ser capaz de:

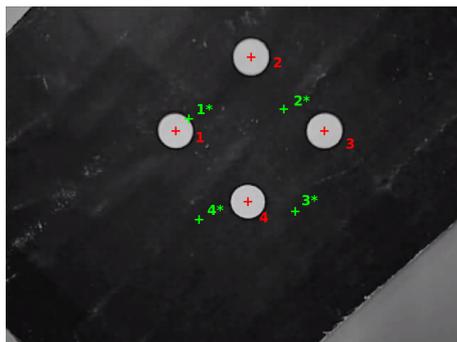
1. Nombrar y definir las principales características del control IBVS.
2. Definir el concepto *matriz Jacobiana de imagen* o *matriz de interacción*.
3. Obtener la expresión de la *matriz de interacción* cuando las características de la imagen son puntos.
4. Obtener la expresión del control IBVS sin restricciones de sistemas mecánicos.
5. Obtener la expresión del control IBVS con restricciones de sistemas mecánicos (aplicación robot manipulador industrial de 6 grados de libertad).
6. Definir las principales virtudes y defectos del control IBVS.



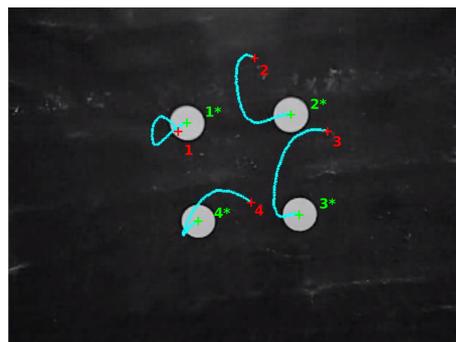
(a) Posición inicial en el espacio cartesiano.



(b) Posición final en el espacio cartesiano.

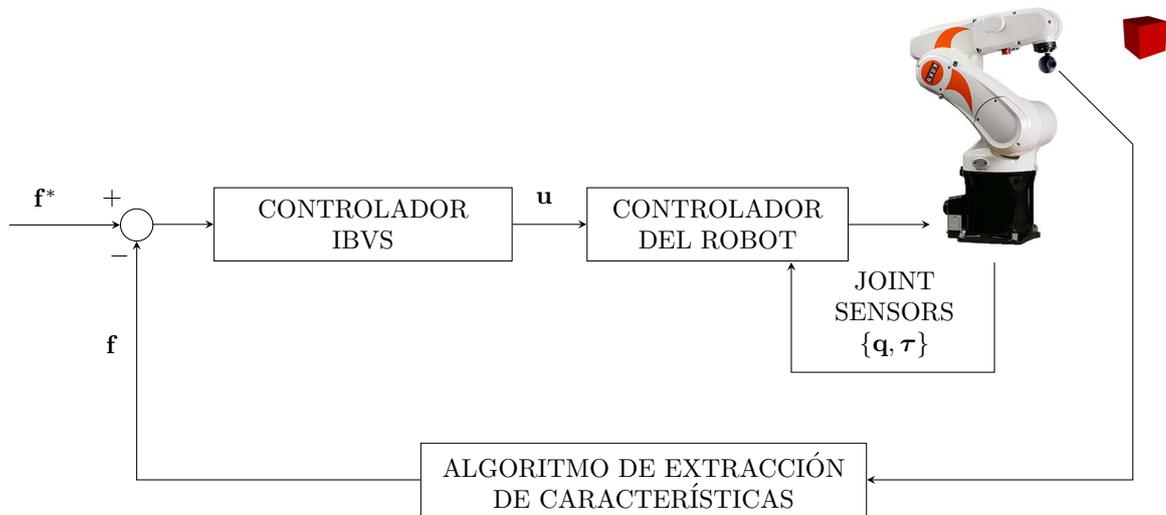


(c) Posición inicial en el plano de imagen.



(d) Posición final en el plano de imagen.

**Figura 1:** Ejemplo de control de robots industriales con realimentación visual IBVS: en las subfiguras c) y d), se muestran en rojo la posición de cada una de las características en el plano de imagen y en verde la posición deseada de cada una de las mismas; en cian se muestra las trayectorias seguidas.



**Figura 2:** Diagrama de bloques del control IBVS, donde  $\mathbf{f}$  y  $\mathbf{f}^*$  representan las coordenadas en el espacio 2D de las características actuales y deseadas, respectivamente,  $\mathbf{u}$  la acción de control,  $\mathbf{q}$  el vector de valores de articulación y  $\boldsymbol{\tau}$  el vector par de motor.

## 4 Desarrollo

### 4.1 ¿Qué entendemos por control Image-Based Visual Servoing (IBVS)?

Tal y como hemos mencionado en la sección de Introducción, el control IBVS se caracteriza por realizar las acciones de control en el espacio de imagen. Se trata de un control cinemático cuyo objetivo es el de reducir el error existente entre la posición de las características actuales del objeto  $\mathbf{f}$  y la posición deseada de esas mismas características  $\mathbf{f}^*$  en el plano de imagen<sup>1</sup>.

La [figura 2](#) muestra el diagrama de bloques asociado al IBVS. Por una parte tenemos que la información obtenida por la cámara es procesada por un algoritmo de visión artificial que nos va a proporcionar el vector de características  $\mathbf{f}$  en cada periodo de control  $T$ . Por otra parte, tenemos que nuestro controlador recibe el error entre  $\mathbf{f}^*$  y  $\mathbf{f}$  de cara a realizar aquellas correcciones oportunas que permitan reducirlo a cero. Estas acciones las envía al controlador del robot que será el encargado de convertirlas a intensidades de cada uno de los motores que lo conforman.

#### ¡MUY IMPORTANTE!

Como bien sabemos, hay que asegurar que el robot reciba una acción de control en cada periodo  $T$ , es decir, que no tengamos pérdidas de periodos que engañen a nuestro controlador. Recordad, tenemos que **MEDIR**, **CALCULAR** y **ACTUAR** en cada periodo. Por lo tanto, es conveniente realizar una planificación de todas aquellas tareas involucradas (*frame rate* de la cámara, algoritmo de visión, retardos por las comunicaciones, cálculos a realizar por el controlador, etc.) y definir el periodo  $T$  para que todas las tareas se puedan realizar dentro del mismo.

<sup>1</sup>Ten en cuenta que  $\mathbf{f}$  es un vector columna que tendrá tantas componentes como características del objeto queramos medir. Esas características las queremos ver de una determinada manera (por ejemplo, para que un robot manipulador pueda coger el objeto), por lo que  $\mathbf{f}^*$  será un vector columna con las mismas componentes que  $\mathbf{f}$  pero con las coordenadas deseadas.

## 4.2 Relación entre movimiento de la cámara y las características proyectadas en el plano de imagen

Vamos a ver ahora cómo al realizar la cámara un movimiento en el plano Cartesiano se produce un movimiento de las características en el plano de imagen. Para ello vamos a suponer que tenemos que nuestro objeto está definido por un punto  $\mathbf{P} = [X_1 \ Y_1 \ Z_1]^T \in \mathbb{R}^3$ , una cámara situada en el espacio cartesiano, cuya ubicación viene dada por la matriz de transformación homogénea  $\mathbf{T}^C$  y el plano de imagen a una determinada distancia focal  $l$ , tal y como vemos en el diagrama de la [figura 3](#)<sup>2</sup>. La proyección del punto  $\mathbf{P}$  sobre el plano de imagen determina el punto  $\mathbf{f} = [u_1 \ v_1]^T \in \mathbb{R}^2$ , siendo  $u_1 = l \cdot \frac{X_1}{Z_1}$  y  $v_1 = l \cdot \frac{Y_1}{Z_1}$  las coordenadas del punto en el plano de imagen con unidades píxel.

A continuación, vamos a derivar cada una de las componentes de  $\mathbf{f}$  y cada una de las componentes de  $\mathbf{P}$  obteniendo el siguiente resultado<sup>3</sup>:

$$\dot{\mathbf{f}} = \begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \end{bmatrix} = \begin{bmatrix} l \cdot \frac{\dot{X}_1}{Z_1} - u_1 \cdot \frac{\dot{Z}_1}{Z_1} \\ l \cdot \frac{\dot{Y}_1}{Z_1} - v_1 \cdot \frac{\dot{Z}_1}{Z_1} \end{bmatrix}. \quad (1)$$

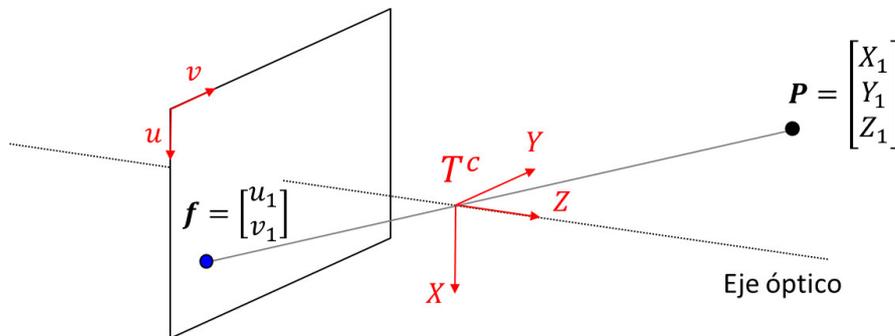
$$\dot{\mathbf{P}} = \begin{bmatrix} \dot{X}_1 \\ \dot{Y}_1 \\ \dot{Z}_1 \end{bmatrix} = \begin{bmatrix} -v_x - \omega_y \cdot Z_1 + \omega_z \cdot Y_1 \\ -v_y - \omega_z \cdot Z_1 + \omega_x \cdot Z_1 \\ -v_z - \omega_x \cdot Z_1 + \omega_y \cdot X_1 \end{bmatrix}. \quad (2)$$

Sustituyendo la [Ecuación 2](#) en la [Ecuación 1](#) tenemos:

$$\dot{\mathbf{f}} = \begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \end{bmatrix} = \begin{bmatrix} -\frac{l}{Z_1} \cdot v_x + \frac{u_1}{Z_1} \cdot v_z + \frac{u_1 \cdot v_1}{l} \cdot \omega_x - (l + \frac{u_1^2}{l}) \cdot \omega_y + v_1 \cdot \omega_z \\ -\frac{l}{Z_1} \cdot v_y + \frac{v_1}{Z_1} \cdot v_z + (l + \frac{v_1^2}{l}) \cdot \omega_x - \frac{u_1 \cdot v_1}{l} \cdot \omega_y - u_1 \cdot \omega_z \end{bmatrix}. \quad (3)$$

La [Ecuación 3](#) podemos reorganizarla y ponerla en formato matricial tal que:

$$\dot{\mathbf{f}} = \begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \end{bmatrix} = \begin{bmatrix} -\frac{l}{Z_1} & 0 & \frac{u_1}{Z_1} & \frac{u_1 \cdot v_1}{l} & -(l + \frac{u_1^2}{l}) & v_1 \\ 0 & -\frac{l}{Z_1} & \frac{v_1}{Z_1} & (l + \frac{v_1^2}{l}) & -\frac{u_1 \cdot v_1}{l} & -u_1 \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \mathbf{L}_{f1} \cdot \mathbf{V}_C, \quad (4)$$



**Figura 3:** Representación del espacio Cartesiano donde una cámara proyecta un punto  $\mathbf{P} \in \mathbb{R}^3$  en su plano de imagen.

<sup>2</sup>Para repasar los conceptos básicos de visión artificial podéis consultar los capítulos 1 y 4 del libro (Corke 2011).

<sup>3</sup>Tengamos presente que la relación entre un movimiento Cartesiano de la cámara con respecto a un punto  $\mathbf{P}$  puede determinarse por  $\dot{\mathbf{P}} = -\mathbf{V} - \boldsymbol{\Omega} \times \mathbf{P}$ , donde  $\mathbf{V} = [v_x \ v_y \ v_z]^T$  y  $\boldsymbol{\Omega} = [\omega_x \ \omega_y \ \omega_z]^T$  son las velocidades cartesianas y angulares de la cámara respecto al punto  $\mathbf{P}$ .

donde  $\mathbf{L}_{f_1}$  es la denominada **matriz de interacción** o jacobiana de la imagen, y  $\mathbf{V}_C$  es la velocidad de la cámara.

### ACTIVIDAD 1

Supongamos que en vez de definir el objeto como un punto, lo definimos como un conjunto de  $n$  puntos, de tal manera que  $\mathbf{f} = [\mathbf{f}_1^T \ \dots \ \mathbf{f}_n^T]^T$ . ¿Cómo quedaría definida la matriz de interacción  $\mathbf{L}_f$ ?

### SOLUCIÓN

El enunciado nos dice que  $\mathbf{f} = [\mathbf{f}_1^T \ \mathbf{f}_2^T \ \dots \ \mathbf{f}_n^T]^T$ . Entonces podemos reescribir la Ecuación 4 como:

$$\dot{\mathbf{f}} = \begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \vdots \\ \dot{u}_n \\ \dot{v}_n \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{l}{Z_1} & 0 & \frac{u_1}{Z_1} & \frac{u_1 \cdot v_1}{l} & -(l + \frac{u_1^2}{l}) & v_1 \\ 0 & -\frac{l}{Z_1} & \frac{v_1}{Z_1} & (l + \frac{v_1^2}{l}) & -\frac{u_1 \cdot v_1}{l} & -u_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\frac{l}{Z_n} & 0 & \frac{u_n}{Z_n} & \frac{u_n \cdot v_n}{l} & -(l + \frac{u_n^2}{l}) & v_n \\ 0 & -\frac{l}{Z_n} & \frac{v_n}{Z_n} & (l + \frac{v_n^2}{l}) & -\frac{u_n \cdot v_n}{l} & -u_n \end{bmatrix}}_{\mathbf{L}_f} \cdot \mathbf{V}_C,$$

Uno de los aspectos que dificultan el trabajar con características puntuales es la necesidad de conocer el valor de  $Z_1$ . Recordemos que los objetos pueden encontrarse en movimiento y que, con una sola cámara, no podemos calcular la profundidad exacta de los objetos. Si bien hay algoritmos de estimación de la profundidad, tales como (Li, Zhao y Ding 2018), lo cierto es que se cometen errores que pueden afectar al rendimiento del controlador.

### ACTIVIDAD 2

Supongamos que en vez de definir el objeto como un punto, lo definimos como una recta definida en términos de  $(\rho, \theta)$ . ¿Cómo quedaría definida la matriz de interacción  $\mathbf{L}_f$ ?

### SOLUCIÓN

Es muy común representar una línea con la parametrización  $(\rho, \theta)$ , donde:

$$v = -u \cdot \tan(\theta) + \frac{\rho}{\cos(\theta)},$$

donde  $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$  es el ángulo a partir del eje horizontal de la imagen hasta la perpendicular de la recta, y  $\rho \in [-\rho_{min}, \rho_{max}]$  es la distancia perpendicular entre el origen de coordenadas del plano de imagen y la línea recta.

Por lo tanto, de forma similar a lo que realizamos para el caso de puntos como características, tenemos que:

$$\dot{\mathbf{f}} = \begin{bmatrix} \dot{\theta} \\ \dot{\rho} \end{bmatrix} = \underbrace{\begin{bmatrix} \lambda_\theta \cdot \sin(\theta) & \lambda_\theta \cdot \cos(\theta) - \rho \cdot \lambda_\theta & -\rho \cdot \sin(\theta) & -\rho \cdot \cos(\theta) & -1 \\ \lambda_\rho \cdot \sin(\theta) & \lambda_\rho \cdot \cos(\theta) - \rho \cdot \lambda_\rho & -\cos(\theta) \cdot (1 + \rho^2) & \sin(\theta) \cdot (1 + \rho^2) & 0 \end{bmatrix}}_{\mathbf{L}_f} \cdot \mathbf{V}_C,$$

donde  $\lambda_\theta = (a \cdot \cos(\theta) - b \sin(\theta))/d$  and  $\lambda_\rho = -(a \cdot \rho \cdot \sin(\theta) + b \rho \cos(\theta) + c)/d$ . Como vemos, al igual que en el caso de los puntos como características requerían información del espacio  $\mathbb{R}^3$ , la profundidad  $Z$ , la línea requiere de la ecuación del plano  $a \cdot X + b \cdot Y + c \cdot Z + d = 0$  que contiene a la línea. Ya que cada línea es la intersección de dos planos, podemos elegir cualquier plano para la que  $d \neq 0$ . Cabe también mencionar que se necesitan al menos tres líneas para posicionar una cámara en el espacio Cartesiano.

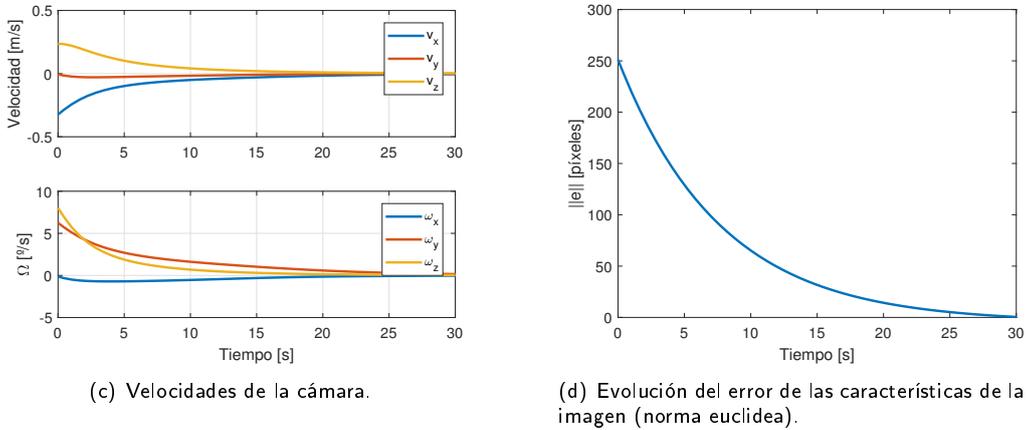
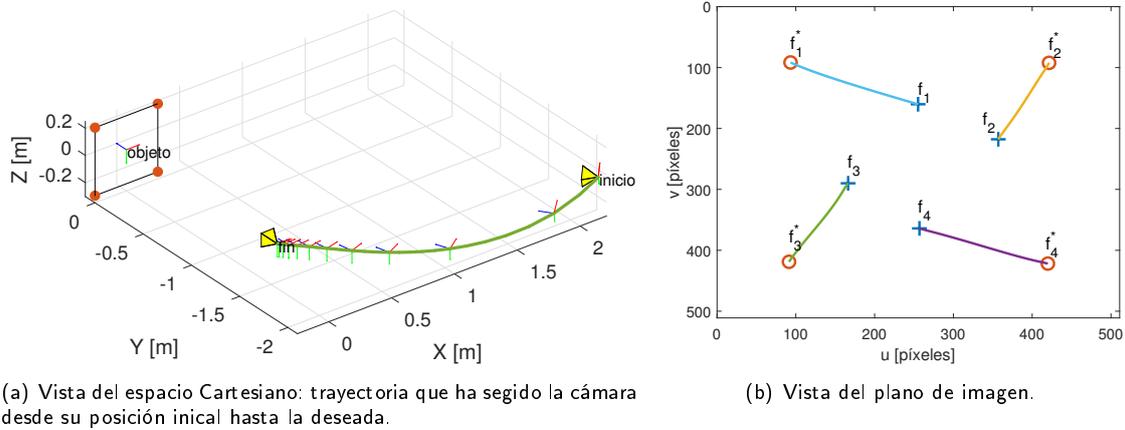


Figura 4: Rendimiento del control IBVS.

### 4.3 Ley de control IBVS

La expresión obtenida en la Ecuación 4 nos determina cómo un movimiento de la cámara en el espacio Cartesiano nos produce un movimiento de las características del objeto en el plano de imagen. Del mismo modo podemos decir:

$$\mathbf{V}_C = \mathbf{L}_f^\dagger \cdot \dot{\mathbf{f}}, \quad (5)$$

donde  $\dagger$  es la pseudo-inversa Moore-Penrose (Rakha 2004). Vemos que un movimiento de las características en el plano de imagen se traduce en un movimiento de la cámara. Si tomamos  $\dot{\mathbf{f}} = -\lambda \cdot (\mathbf{f} - \mathbf{f}^*)$ , por ejemplo, para asegurar un decrecimiento exponencial del error, tenemos que el controlador IBVS tiene la siguiente expresión:

$$\mathbf{V}_C = -\lambda \cdot \mathbf{L}_f^\dagger \cdot (\mathbf{f} - \mathbf{f}^*). \quad (6)$$

Vamos a ver las particularidades que tiene el controlador IBVS dado por la Ecuación 6. Para ello, supongamos la configuración mostrada en la figura 4, cuya implementación se ha realizado utilizando la plataforma Matlab 2017b y la toolbox *Visual Servoing Toolbox - Release 13* (Cervera 2002). Se tiene pues un objeto cuadrado de dimensiones 0.4x0.4 metros situado en el origen de coordenadas tal y como muestra la figura 4(a). Las características detectadas son los cuatro vértices marcados en verde. La cámara, que tiene una resolución de 512x512 píxeles y una distancia focal de 1000, esta situada en las coordenadas (2.15, -2.09, 0) metros y cuya orientación en ángulos de Euler es (68.93, 58.06, -9) grados. Con esta configuración, las características en el plano de imagen se ubican tal y como muestra la figura 4(b) (posición inicial resaltadas con el símbolo + y la posición final resaltada con el símbolo o).

A la vista de los resultados obtenidos, podemos llegar a las siguientes conclusiones:

- La ley de control IBVS realiza **trayectorias rectas en el plano de imagen**, ya que la ley de control dada en la [Ecuación 6](#) trata de reducir el error ( $\mathbf{f} - \mathbf{f}^*$ ).
- La ley de control IBVS realiza **trayectorias curvilíneas en el plano Cartesiano**, ya que esta **no controla directamente la pose de la cámara**.
- Al realizar el control en el plano de imagen, **será más difícil que las características se salgan del plano de imagen** durante el transcurso de la tarea de control.
- Al no realizar el control directamente en el espacio Cartesiano, **no tendremos control sobre la trayectoria que va a seguir la cámara** durante el transcurso de la tarea de control.
- Debido a la ley de decrecimiento exponencial del error, se tendrán **acciones de control fuertes** en los **primeros instantes** de la tarea, mientras que la **convergencia** será **lenta** en los **últimos instantes**.

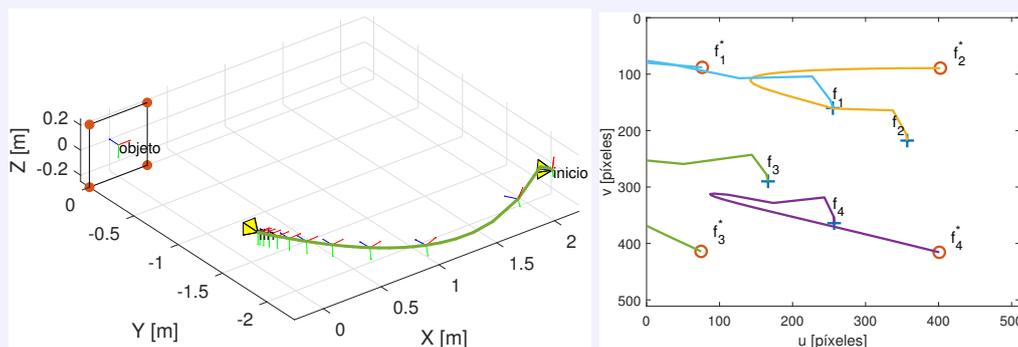
Un último punto importante a saber es que esta ley de control IBVS solamente garantiza una convergencia local, es decir, si las características  $\mathbf{f}$  están situadas cerca de la referencia  $\mathbf{f}^*$ . Además, la forma  $\mathbf{L}_f$  obtenida utilizando puntos como características, presenta diversos problemas: son necesarios, al menos, cuatro puntos para poder obtener correctamente y de forma unívoca la solución deseada; si hay muchos puntos, el cálculo de la pseudo-inversa puede ser computacionalmente costoso; y no se pueden resolver situaciones en las que los puntos están girados  $180^\circ$  respecto de la referencia.

### ACTIVIDAD 3

Supongamos que no podemos tener acceso en cada instante al parámetro de profundidad  $Z_i$  de cada uno de los cuatro vértices del ejemplo anterior, y decidimos mantenerlo constante al valor que tendrá cuando estemos en la referencia, esto es  $Z_1^* = Z_2^* = Z_3^* = Z_4^* = 1,5$  metros. ¿Cómo crees que se verán afectadas las trayectorias de las características obtenidas en el plano de imagen? ¿Seguirán siendo rectas?

### SOLUCIÓN

La [figura 5](#) muestra el resultado del control IBVS si utilizamos  $Z_i^*$  en vez de  $Z_i$  manteniendo los mismos parámetros del controlador. Vemos que las trayectorias en el plano de imagen ya no son líneas rectas e, incluso, las características se han salido del campo de visión de la cámara durante el transcurso de la tarea de control.



**Figura 5:** Rendimiento del control IBVS con  $Z_i^*$ . En la imagen de la izquierda, se representa la trayectoria que ha seguido la cámara desde su posición inicial hasta la deseada en el plano Cartesiano. En la imagen de la derecha, se representa las trayectorias de las características en el plano de imagen.

#### 4.4 Control IBVS con la cámara situada en el efector final de un robot industrial de 6 grados de libertad

Vamos a ver en este apartado cómo modificar la [Ecuación 4](#) de modo que la cámara se mueva solidaria con un brazo robot manipulador industrial. Para ello, tenemos que convertir la velocidad cartesiana de la cámara  $\mathbf{V}_C$  a velocidad de articulación del robot  $\dot{\mathbf{q}}$  de la siguiente forma:

$$\dot{\mathbf{f}} = \mathbf{L}_f \cdot \mathbf{V} \cdot \mathbf{J}_r \cdot \dot{\mathbf{q}}, \quad (7)$$

siendo  $\mathbf{J}_r$  la matriz Jacobiana del robot y  $\mathbf{V}$  la denominada matriz de transformación de giro de velocidad que tiene la siguiente forma<sup>4</sup>:

$$\mathbf{V} = \begin{bmatrix} \mathbf{R} & [\mathbf{t}]_{\times} \mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{bmatrix}, \quad (8)$$

donde  $\mathbf{R}$  y  $\mathbf{t}$  son la matriz de rotación y el vector de translación de la cámara con respecto al efector final del robot. De esta manera, y tomando  $\dot{\mathbf{f}} = -\lambda \cdot (\mathbf{f} - \mathbf{f}^*)$ , tenemos que la ley de control IBVS para robots manipuladores industriales tiene la forma:

$$\dot{\mathbf{q}} = -\lambda \cdot (\mathbf{L}_f \cdot \mathbf{V} \cdot \mathbf{J}_r)^{\dagger} \cdot (\mathbf{f} - \mathbf{f}^*). \quad (9)$$

Vamos a ver las particularidades que tiene el controlador IBVS dado por la [Ecuación 9](#). Para ello, supongamos la configuración mostrada en la [figura 6](#), compuesta por un robot de 6 grados de libertad, una cámara ubicada en el efector final del robot, y un objeto cuadrado cuyas características a detectar son los cuatro vértices<sup>5</sup>. Una vez realizada la simulación, la cual podéis ver en [video](#), vemos que las trayectorias seguida por cada una de las características detectadas, desde su posición final hasta su posición deseada, han seguido líneas rectas (ver [figura 6\(a\)](#)), tal y cómo sucedía en el caso de cámara libre sin robot. Además, vemos como la norma euclídea del error ([figura 6\(c\)](#)) también tiene un comportamiento exponencial decreciente dado por  $\dot{\mathbf{f}} = -\lambda \cdot (\mathbf{f} - \mathbf{f}^*)$ .

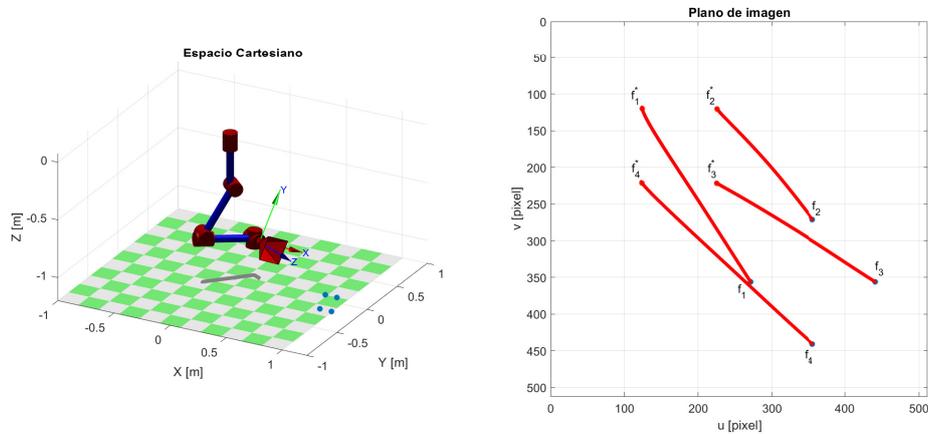
A la vista de los resultados, vemos que la incorporación del robot manipulador no modifica el comportamiento teórico visto en el apartado [subsección 4.3](#). Sin embargo, hay que tener presente que el robot es un mecanismo físico, con limitaciones espaciales y dinámicas que se deberán tener en cuenta a la hora de diseñar e implementar la tarea de *visual servoing*.

#### Sabías que...

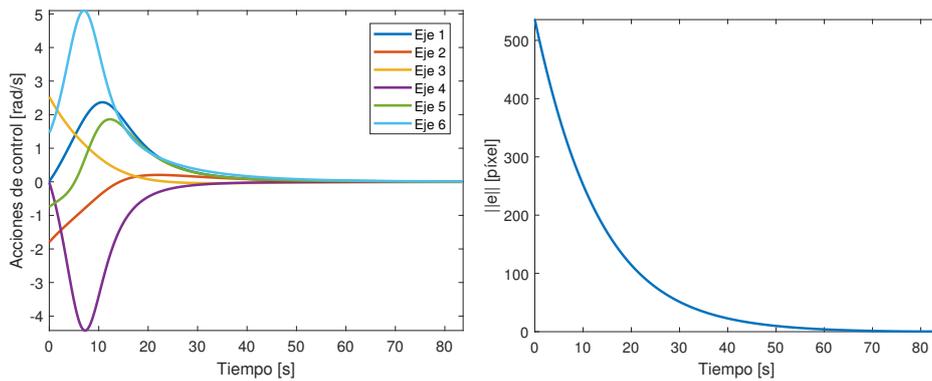
Actualmente, la mayoría de las aplicaciones industriales que utilizan realimentación visual para controlar robots lo hacen también incorporando otros sensores. Es el caso de aplicaciones en la industria de la alimentación, donde la visión artificial se utiliza para determinar la posición de los alimentos en la cinta transportadora en un instante de tiempo, mientras que se utiliza encoders para poder estimar su posición mientras los alimentos están en movimiento (ver ejemplo en [video2](#)). El motivo es que el sensor de visión, aunque proporciona mucha información, también es un sensor "lento" y tiene que ser complementado con otros sensores más "rápidos".

<sup>4</sup>En caso de necesitar repasar estos conceptos básicos de robótica, podéis consultar los capítulos 1 y 3 del libro (Corke 2011).

<sup>5</sup>El simulador se ha implementado en Matlab 2017b utilizando las toolboxes *Robotics Toolbox - Realease 10* (Corke 2019) y *Visual Servoing Toolbox - Release 13* (Cervera 2002).



(a) Vistas del espacio Cartesiano (izquierda) y el plano de imagen (derecha).



(b) Velocidades del robot  $\dot{q}$ .

(c) Evolución del error de las características de la imagen (norma euclídea).

**Figura 6:** Control IBVS con robot manipulador industrial de 6 grados de libertad (ver experimento en [video1](#)).

## 5 Cierre

A lo largo de este objeto de aprendizaje hemos aprendido los fundamentos del control cinemático de robots denominado *Image Based Visual Servoing* (IBVS). Hemos aprendido las dos configuraciones clásicas de todo sistema visual servoing (*Eye-in-hand* y *Eye-to-hand*).

A partir del modelo básico de cámara, hemos aprendido a obtener la relación que existe entre el movimiento realizado por la cámara en el espacio cartesiano con el movimiento que sufre la característica en el plano de imagen. A esta relación la hemos definido como matriz de interacción y hemos obtenido su expresión concreta para el caso de tener un punto como característica detectada del objeto.

Una vez obtenida la relación entre movimiento Cartesiano de cámara y característica en el plano de imagen, hemos aprendido a obtener la ley de control IBVS para el caso sencillo de cámara sin restricciones mecánicas. Por medio de un ejemplo, hemos llegado a las siguientes conclusiones:

1. El control IBVS obtenido realiza trayectorias rectas en el plano de imagen para reducir el error existente entre las características detectadas y deseadas, mientras que realiza una trayectoria curva y no predecible en el espacio Cartesiano.

2. Con esta ley de control tenemos un comportamiento exponencial decreciente del error que hace que la cámara se mueva muy rápido cuando el error es grande pero muy despacio cuando el error es pequeño, es decir, cuando estamos cerca de la posición deseada.
3. Al realizar el control en el plano de imagen, es más difícil que las características se salgan del plano de imagen durante el transcurso de la tarea de control.

Además, hemos aprendido a obtener la ley de control IBVS cuando la cámara tiene el movimiento restringido por un sistema mecánico, concretamente por un robot industrial. A partir de un ejemplo con un robot de 6 grados de libertad hemos visto que el comportamiento es igual que en el caso de la cámara libre, siempre y cuando se respeten las restricciones físicas (tanto cinemáticas como dinámicas) del robot.

## Bibliografía

- Burkov, A. (2019). *The Hundred-page Machine Learning Book*. Andriy Burkov. ISBN: 9781999579517 (vid. pág. 1).
- Cervera, Enric (2002). *Visual Servoing Toolbox - R13*. <http://vstoolbox.sourceforge.net/1> (vid. págs. 6, 8).
- Chaumette, François y Seth Hutchinson (2006). “Visual servo control. I. Basic approaches”. En: *Robotics Automation Magazine, IEEE* 13.4, págs. 82-90. ISSN: 1070-9932. DOI: [10.1109/MRA.2006.250573](https://doi.org/10.1109/MRA.2006.250573) (vid. pág. 1).
- (2007). “Visual servo control. II. Advanced approaches [Tutorial]”. En: *Robotics Automation Magazine, IEEE* 14.1, págs. 109-118. ISSN: 1070-9932. DOI: [10.1109/MRA.2007.339609](https://doi.org/10.1109/MRA.2007.339609) (vid. pág. 1).
- (2008). “Visual Servoing and Visual Tracking”. En: *Springer Handbook of Robotics*, págs. 563-583 (vid. pág. 1).
- Corke, Peter (2011). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Berlin, Germany: Springer-Verlag (vid. págs. 1, 4, 8).
- (2019). *Robotics Toolbox - R10*. <http://petercorke.com/wordpress/toolboxes/robotics-toolbox> (vid. pág. 8).
- Li, Xiangfei, Huan Zhao y Han Ding (2018). “Real-Time Feature Depth Estimation for Image-Based Visual Servoing”. En: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, págs. 7314-7320 (vid. pág. 5).
- Rakha, Medhat A. (2004). “On the Moore-Penrose Generalized Inverse Matrix”. En: *Appl. Math. Comput.* 158.1, págs. 185-200. ISSN: 0096-3003. DOI: [10.1016/j.amc.2003.09.004](https://doi.org/10.1016/j.amc.2003.09.004) (vid. pág. 6).