

Implementación de controladores visuales en dron comercial: estudio comparativo Visual servoing controller for a commercial drone: comparison study

P. A. Ochoa-Salinas ^{a,*}, A. B. Morales-Díaz ^a, H. M. Pérez-Villeda ^b, R. J. Villalobos-Salazar ^a

^aGrupo de robótica y manufactura avanzada, CINVESTAV Saltillo, 25900, Ramos Arizpe, Coahuila, México.

^bComputer Science at University of Innsbruck, Austria

Resumen

Los controladores visuales han sido de gran utilidad para la robótica durante los últimos años. Habilitar a un robot para que perciba visualmente su entorno, le permite interactuar y tomar decisiones basadas en tareas previamente definidas. La fácil adquisición de cámaras, vehículos aéreos o drones comerciales ha fomentado la investigación con ellos. Una herramienta muy útil para crear el enlace entre el robot y los periféricos es el software Robotic Operation System (**ROS**). Empleando lo anterior, se presentan resultados experimentales de una comparativa de controladores visuales clásicos en un dron Tello, mostrando que los controladores presentan características diferentes dependiendo del espacio en el que realizan su tarea.

Palabras Clave: Controlador visual, robótica móvil, dron.

Abstract

Visual controllers have been very useful in robotics in recent years. Allowing a robot to visually perceive its environment enables it to interact and make decisions based on previously defined tasks. The easy acquisition of cameras, aerial vehicles, or commercial drones has allowed to research with them. A very useful tool for creating the link between the robot and its peripherals is the Robotic Operation System (**ROS**) software. Employing those topics previously mentioned, experimental results of a comparison of classical visual controllers are presented using a Tello drone, showing that each controller has its own behavior related to the space where the task is performed.

Keywords: Visual controller, mobile robotics, drone.

1. Introducción

En los últimos años, el avance de las tecnologías en el área de la electrónica ha facilitado el acceso a dispositivos como las cámaras, unidades de control y microprocesadores, entre otros, para su uso en la vida diaria, poniendo a disposición del público en general vehículos móviles como los drones.

Los drones han sido usados en diversas aplicaciones, tales como: tareas de vigilancia, adquisición de imágenes geográficas, toma de video en el mundo del espectáculo e incluso en el apoyo de la agricultura y la ganadería (Parisi *et al.*, 2019).

En el estado del arte se reporta el uso de drones comerciales y controladores visuales para resolver problemas en el área de la robótica. En (Lin y Peng, 2021) se emplea el dron comercial Parrot AR para efectuar la evasión de obstáculos en exteriores; mientras que en el trabajo de (Toro-Arcila *et al.*, 2023) se usa un

Parrot Bebop 2 para realizar tareas simultáneas con un control jerárquico de evasión de obstáculos, mantener el campo de visión de la cámara para seguir características visuales y ejecutar un control servo visual en interiores. En el trabajo de (Caldas *et al.*, 2022) se utiliza el mismo dron, el Parrot Bebop 2 y se emplea su cámara para realizar Localización y Mapeo Simultáneo Visual (vSLAM), fusionando éstos datos con los obtenidos en la Unidad de Movimiento Inercial (IMU) del mismo dron, generando una navegación más robusta.

En (Machkour *et al.*, 2022) se encuentra un estudio exhaustivo del estado del arte en sistemas servo visuales aplicados en la robótica. En el trabajo de (Lin *et al.*, 2021) se muestra un ejemplo de controladores visuales, donde un control basado en imagen adaptivo coordina un robot líder y un seguidor a través de la información de la cámara sin comunicación entre ellos. También con este tipo de controladores se pueden realizar se-

* Autor para correspondencia: pablo.ochoa@cinvestav.mx

Correo electrónico: pablo.ochoa@cinvestav.mx (Pablo Alonso Ochoa-Salinas), america.morales@cinvestav.mx (América Berenice Morales-Díaz), hecvilleda@gmail.com (Héctor Manuel Pérez-Villeda), rodolfo.villalobos@cinvestav.mx (Rodolfo de Jesús Villalobos-Salazar).

Historial del manuscrito: recibido el 16/03/2023, última versión-revisada recibida el 21/06/2023, aceptado el 26/06/2023 publicado el 11/09/2023. **DOI:** <https://doi.org/10.29057/icbi.v11iEspecial2.10696>



guimiento en exteriores y con drones comerciales, tal como se presenta en (Pestana *et al.*, 2013). Tareas peligrosas como remover objetos de cables de alta tensión se pueden resolver con manipuladores aéreos no tripulados y controles visuales basados en posición como lo muestra (Li *et al.*, 2021).

El uso intensivo de este tipo de vehículos motiva el presente trabajo, en el cual se implementaron estrategias de control visual clásico en un dron comercial accesible al público, las cuales se compraron entre sí y dan pauta a su uso en tareas de precisión como la toma de video de cultivos o el conteo de ganado, por mencionar algunas.

El presente trabajo está organizado de la siguiente manera: en la sección 2 se describen los controladores visuales clásicos; en la sección 3 se presentan las leyes de control visual empleadas. En la sección 4 se muestra el diseño experimental, en la sección 5 los resultados obtenidos y finalmente, en la sección 6 se cierra el artículo con algunas conclusiones y posible trabajo a futuro.

2. Controladores visuales clásicos

Para el diseño de un control visual clásico se requiere conocer las restricciones geométricas y la definición del objetivo de control. A continuación, se describen estos tópicos.

2.1. Restricciones geométricas

Las restricciones geométricas son fundamentales en la visión por computadora y se han utilizado durante muchos años para encontrar relaciones entre diferentes imágenes. Entre las más estudiadas y utilizadas en la robótica móvil se encuentran la homografía, la geometría epipolar y el tensor trifocal, las cuales emplean información visual para buscar relaciones entre dos o tres imágenes.

La homografía en particular es una transformación que permite mapear un punto de un plano (\mathbf{x}) a un punto en otro plano (\mathbf{x}') siempre y cuando exista una matriz no singular (\mathbf{H}) que lo cumpla, es decir, $\mathbf{x}' = \mathbf{H}\mathbf{x}$. Esta transformación se utiliza comúnmente para realizar correcciones de perspectiva y distorsiones en imágenes. Un ejemplo visual de la aplicación de la homografía, obtenido de la librería OpenCV (Intel Corporation and Willow Garage Inc., 2021) se presenta en la Figura 1.

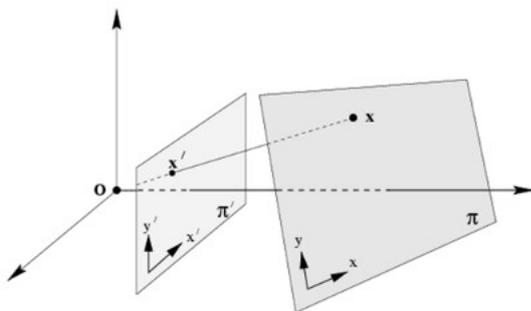


Figura 1: Homografía.

Conociendo los parámetros de cámara, la matriz (\mathbf{H}) se puede descomponer en sus componentes: rotacional (\mathbf{R}) y traslacional (\mathbf{t}), los cuales están expresados en (1):

$$\mathbf{H} = \mathbf{R} - \frac{\mathbf{t}\mathbf{n}^T}{d}, \quad (1)$$

donde \mathbf{n}^T es el vector normal al plano y d es la distancia entre los dos centros ópticos de las imágenes.

2.2. Controladores visuales

Los controladores visuales utilizan la información de la visión por computadora para controlar el movimiento del robot. Según (Chaumette y Hutchinson, 2006), el objetivo de los controladores visuales es reducir el error definido en (2):

$$\mathbf{e}(t) = \mathbf{s}(\mathbf{m}(t), \mathbf{a}) - \mathbf{s}^*, \quad (2)$$

donde $\mathbf{m}(t)$ es un conjunto de medidas visuales encontradas en una imagen y que son variantes en el tiempo, las cuales son utilizadas para definir \mathbf{k} características visuales. Los parámetros adicionales \mathbf{a} hacen referencia al sistema, por ejemplo: parámetros intrínsecos de cámara o los objetos 3D, y \mathbf{s}^* es el conjunto de características visuales deseados. En resumen, (2) se simplifica como $\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s}^*$.

Para mapear las velocidades traslacionales (\mathbf{v}) y rotacionales ($\boldsymbol{\omega}$) de la cámara $\boldsymbol{\xi} = [\mathbf{v}, \boldsymbol{\omega}]^T$ a las variaciones en el tiempo de las características visuales $\dot{\mathbf{s}}$, se emplea la matriz de interacción o el Jacobiano de imagen \mathbf{L}_s . La expresión correspondiente es la siguiente:

$$\dot{\mathbf{s}} = \mathbf{L}_s \boldsymbol{\xi}. \quad (3)$$

Los controladores visuales clásicos y ampliamente estudiados son los basados en imagen y en posición (Chaumette y Hutchinson, 2006). Además, se han investigado distintas formas de combinarlos, creando los controladores híbridos o particionados. Estos últimos, permiten separar los Jacobianos de velocidad traslacional y rotacional de la matriz de interacción, y emplear un método de control distinto. Algunos ejemplos se pueden encontrar en el 2.5D (Malis *et al.*, 1999) y el particionado de (Corke y Hutchinson, 2001). Un controlador visual avanzado es el basado en la homografía (Benhimane y Malis, 2007), que usa su descomposición para encontrar el error de traslación y de rotación, de manera similar al controlador basado en posición.

Los errores y las matrices de interacción que se implementaron para los controladores visuales son descritos a continuación:

- Controlador basado en imagen.

El error es definido en el espacio de las características visuales mencionadas en (2) y la matriz de interacción para un controlador basado en imagen se presenta en (4):

$$\mathbf{L}_{s_1} = \begin{bmatrix} -\frac{\lambda}{z} & 0 & \frac{u}{z} & \frac{uv}{z} & -\frac{\lambda^2+u^2}{\lambda} & v \\ 0 & -\frac{\lambda}{z} & \frac{v}{z} & \frac{\lambda^2+v^2}{\lambda} & -\frac{uv}{\lambda} & -u \end{bmatrix}, \quad (4)$$

donde u y v son las coordenadas en píxeles del plano de imagen de la cámara, λ es la distancia focal y z la profundidad del píxel.

- Controlador basado en posición.

Existen dos alternativas para usar el control basado en posición: la primera es utilizar un modelo 3D del objeto que se pretende alcanzar y la segunda emplea las restricciones geométricas y consiste en llegar a un marco de referencia deseado. En este trabajo se emplea el segundo

método, cuyo error es una traslación y rotación al marco de referencia deseado:

$$\mathbf{e} = \begin{bmatrix} c^* \mathbf{t}_c \\ \theta \mathbf{u} \end{bmatrix}, \quad (5)$$

siendo $\theta \mathbf{u}$ la representación eje-ángulo de la matriz de rotación entre el marco de referencia deseado y el actual, la matriz de interacción para este controlador se define en (6):

$$\mathbf{L}_{s_2} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{\theta_u} \end{bmatrix}, \quad (6)$$

siendo \mathbf{L}_{θ_u}

$$\mathbf{L}_{\theta_u} = \mathbf{I}_3 - \frac{\theta}{2} [\mathbf{u}]_x + \left(1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2\left(\frac{\theta}{2}\right)} \right) [\mathbf{u}]_x^2 \quad (7)$$

y $\text{sinc}(\theta)$ el seno cardinal.

■ Controlador basado en homografía.

El error es similar al del controlador basado en posición, con una parte traslacional y rotacional:

$$\mathbf{e} = \begin{bmatrix} \mathbf{e}_v \\ \mathbf{e}_\omega \end{bmatrix} = \begin{bmatrix} (\mathbf{H} - \mathbf{I})\mathbf{X}^* \\ \mathbf{H} - \mathbf{H}^T \end{bmatrix}, \quad (8)$$

empleando un punto deseado (\mathbf{X}^*), mientras que la matriz de interacción para un controlador basado en homografía se muestra en (9):

$$\mathbf{L}_{s_3} = \begin{bmatrix} \frac{1}{Z^*} & -[\mathbf{e}_v + \mathbf{m}^*]_x \\ [\mathbf{n}^*] & -[\mathbf{n}^*]_x [\mathbf{t}]_x + 2\mathbf{L}_\omega \end{bmatrix}, \quad (9)$$

siendo Z^* la profundidad del punto deseado, \mathbf{m}^* es el punto deseado en coordenadas de imagen y \mathbf{L}_ω

$$\mathbf{L}_\omega = \mathbf{I}_3 - \frac{\text{sin}(\theta)}{2} [\mathbf{u}]_x - \text{sinc}^2\left(\frac{\theta}{2}\right) + (2\mathbf{I} + [\mathbf{u}]_x^2). \quad (10)$$

3. Metodología

3.1. Planteamiento del problema

Empleando un dron comercial Tello de la compañía DJI, es posible realizar una comparación de controladores visuales, aún con sus limitantes (batería, carga, transferencia de datos), usando lenguaje de programación de código abierto.

3.2. Ley de control basado en imagen y posición

La ley de control utilizada en los controladores basado en imagen y posición es la siguiente:

$$\xi = -\lambda \mathbf{L}_s^{-1} \mathbf{e}, \quad (11)$$

en el cual λ es una matriz diagonal de ganancias positivas y \mathbf{L}_s es (4) para el controlador basado en imagen y (6) para el controlador basado en posición. Con la ley propuesta se busca una convergencia exponencial y para comprobar la estabilidad de los controladores visuales, mostrado en (Chaumette y Hutchinson, 2006), se define como función candidata de Lyapunov (12) y su derivada (13)

$$\mathcal{L} = \frac{1}{2} \mathbf{e}^2, \quad (12)$$

$$\dot{\mathcal{L}} = \mathbf{e}^T \dot{\mathbf{e}}. \quad (13)$$

Derivando (2) se tiene que $\dot{\mathbf{e}} = \dot{\mathbf{s}}$ y sustituyendo $\dot{\mathbf{s}}$ en (3) se obtiene:

$$\dot{\mathbf{e}} = \mathbf{L}_s \xi. \quad (14)$$

Finalmente sustituyendo (14) y (11) en (13):

$$\dot{\mathcal{L}} = \mathbf{e}^T \mathbf{L}_s \xi = -\lambda \|\mathbf{e}\|^2 < 0. \quad (15)$$

De (15) se demuestra que el controlador (11) es asintóticamente estable en el sentido Lyapunov. Debido a que la matriz de interacción es estimada para ambos controladores y no la real con respecto al sistema, se requiere que $\hat{\mathbf{L}}_s^{-1}$ sea de rango completo y el producto de $\mathbf{L}_s \hat{\mathbf{L}}_s^{-1} > 0$ sea de definida positiva para asegurar la estabilidad del sistema, con esto también se asegura estabilidad exponencial.

3.3. Ley de control basado en homografía

Una ley de control propuesta de manera local empleada en (Benhimane y Malis, 2007), es la siguiente:

$$\begin{bmatrix} \nu \\ \omega \end{bmatrix} = - \begin{bmatrix} \lambda_\nu \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \lambda_\omega \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{e}_v \\ \mathbf{e}_\omega \end{bmatrix}. \quad (16)$$

Para validar (16), el sistema en lazo cerrado es linealizando cuando el error es igual a cero, por lo que $\theta = \mathbf{t} = 0$, y se obtiene el siguiente sistema lineal:

$$\dot{\mathbf{e}} = - \begin{bmatrix} \lambda_\nu / Z^* & -\lambda_\omega [\mathbf{m}^*]_x \\ \lambda_\nu [\mathbf{n}^*]_x & 2\lambda_\omega \mathbf{I} \end{bmatrix}, \quad (17)$$

cuyos valores propios de la matriz constante son: 2λ , $4Z^*$, $2Z^* + \lambda + \sqrt{(\lambda^2 + 4Z^{*2})}$ y $2Z^* + \lambda - \sqrt{(\lambda^2 + 4Z^{*2})}$, donde $\lambda = \lambda_\nu / \lambda_\omega$, por lo que valores de $\lambda_\nu > 0$ y $\lambda_\omega > 0$, indicando que el sistema (17) es localmente estable ya que se tiene una matriz tipo Hurwitz.

3.4. Ganancias adaptables

Debido a que el error disminuye al aplicar la ley de control (11) y (16), es necesario utilizar ganancias adaptables en función del error. En Visual Servoing Platform (VISP) (Marchand et al., 2005), los autores proponen el cálculo de la ganancia adaptable; modificando dicha ecuación, se propone (18):

$$\lambda = \exp\left(k \frac{\|\mathbf{e}_0\| - \|\mathbf{e}(t)\|}{\|\mathbf{e}_0\|}\right), \quad (18)$$

siendo k una constante positiva, $\|\mathbf{e}_0\|$ es la norma del vector del error inicial y $\|\mathbf{e}(t)\|$ es la norma del vector del error a través del tiempo.

3.5. ROS y dron Tello

ROS (Stanford Artificial Intelligence Laboratory et al., 2018) es un software de código abierto que permite realizar integraciones robóticas de manera rápida y sencilla debido a la estandarización de la estructura de datos que maneja. El arduo trabajo de programadores alrededor del mundo ha hecho que los códigos se actualicen y sean implementados en los distintos proyectos de programación. Actualmente, existen varias librerías que permiten controlar el dron Tello de manera sencilla, por ejemplo, TelloPy ¹, en el cual están basados los controladores de ROS. Esto permite utilizar su estructura de datos para controlar las entradas de velocidad del dron. El dron Tello, cuyas características específicas se pueden encontrar en <https://www.ryzerobotics.com/tello/specs>, tiene un barómetro embebido que le permite mantener una altura predefinida al momento de despegar. Esta ventaja se utilizará para agregarle el control cinemático necesario para desplazarlo de la pose inicial a la pose objetivo.

4. Experimentos

El set de los experimentos se muestra en la Figura 2 y presenta el dron Tello con los marcadores empleados para probar los distintos controladores visuales.



Figura 2: Set de experimentos.

La Figura 3 muestra un esquema general del experimento, donde el cuadro rojo es la pose inicial, el cuadro azul la pose final y la flecha verde el movimiento esperado dada la entrada de control cinemático (11).



Figura 3: Pose inicial y final dron.

Usando la librería Aruco de OpenCv (Intel Corporation and Willow Garage Inc., 2021), se pueden obtener las esquinas de las coordenadas en el plano de imagen de la cámara. Estas se utilizarán para definir las características visuales deseadas s^* . La Figura 4 muestra con puntos rojos, las ocho esquinas obtenidas de la imagen de referencia de dos marcadores Arucos (Garrido-Jurado et al., 2014).

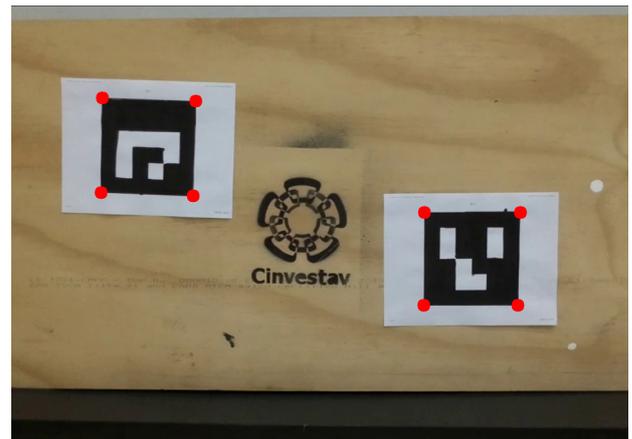


Figura 4: Imagen objetivo.

Se utilizan ganancias adaptables únicamente para las entradas de velocidad traslacional (18), proponiendo una constante $k = 1$, que equivale al 10 % del tiempo característico del sistema (0.1 milisegundos). Para las velocidades rotacionales la constante $\lambda = 0.5$. La pose inicial es similar entre los distintos controladores, con una altura de un metro antes de que se active el control servo visual.

Los parámetros de calibración de la cámara para la imagen de 960 x 720 píxeles transmitida por el dron, que se obtuvieron directamente de las librerías de ROS son los siguientes:

$$\mathbf{K} = \begin{bmatrix} 929.562627 & 0.0 & 487.474037 \\ 0.0 & 928.604856 & 363.165223 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

La arquitectura de comunicación se muestra en la Figura 5,

¹<https://github.com/hanyazou/TelloPy>

en la cual se observa que se utilizó el sistema operativo Ubuntu 18.04 con ROS Melodic; así como también un adaptador inalámbrico Wi-Fi para enviar directamente desde la computadora los comandos de control al dron y recibir la imagen de la cámara integrada de éste.



Figura 5: Arquitectura de comunicación.

La Figura 6 indica el marco referencia del dron Tello.



Figura 6: Marco de referencia dron Tello.

5. Resultados

5.1. Error promedio

En las Figuras 7 y 8 se muestran los errores entre los distintos controladores implementados en el dron Tello. Existe un rango de error en el cual el controlador mantiene estable la pose deseada. Esto se debe a la naturaleza y dinámica del dron, así como a las estimaciones de las matrices de interacción.

El control basado en imagen usa los píxeles de los puntos dados a través de los Arucos y el video del dron, llegando a un error aproximado de 30 píxeles entre los ocho puntos del marcador mostrado en la Figura 4 y la posición deseada.

El control basado en posición busca reducir el error entre los dos marcos de referencia dados. Mientras que en el control basado en homografía, el error es obtenida a través de la homografía calculada y el punto deseado. Ambos errores disminuyen como se muestra en la Figura 8, pero existe un error en la parte rotacional en ambos controladores, debido a que la acción de control no es suficiente.

En los controladores donde se busca la traslación y rotación deseada, el error de traslación disminuye, pero el error de rotación no; esto es debido a que su ganancia es constante y el dron queda en una zona muerta y no permite que gire.

Debido a que el dron Tello mantiene una altitud constante, esto permite que los controladores visuales no requieran agregarle una velocidad de compensación para controlar su peso.

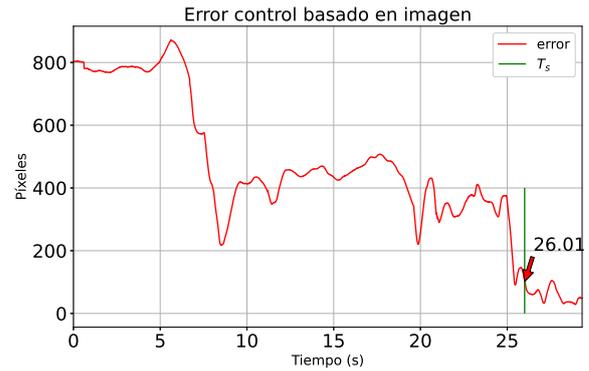


Figura 7: Error controlador basado en imagen.

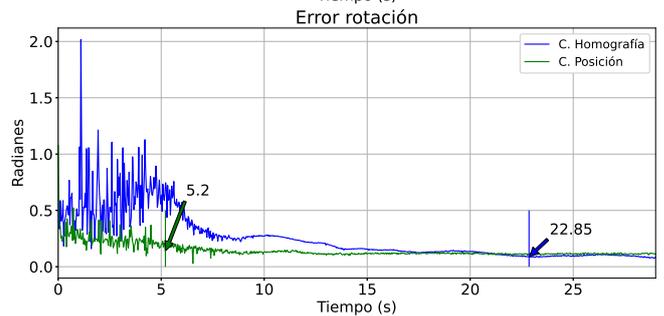
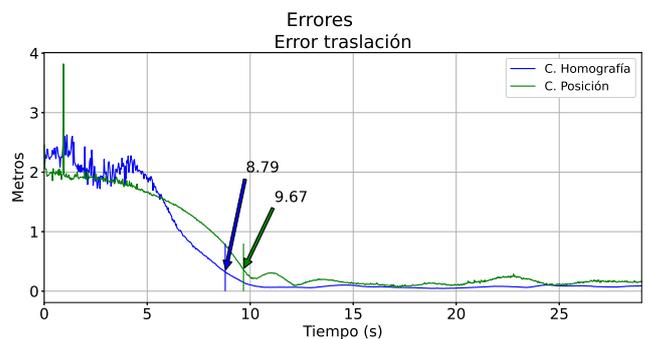


Figura 8: Error controlador basado en posición y homografía.

5.2. Tiempo de asentamiento

El tiempo de asentamiento indica cuanto tarda el controlador en llegar a un estado estable. El tiempo definido para este trabajo es de un 85 % del error inicial. Dicho esto, el controlador que presenta mejor respuesta es el control basado en posición como se muestra en la Figura 8. Por otro lado, el controlador basado en imagen tarda más que los otros dos controladores. El tiempo que tarda cada controlador se indica por las flechas en las Figuras 7 y 8. El movimiento del controlador basado en posición muestra una respuesta más rápida, ya que ésta es lineal, en comparación con el controlador basado de imagen. Por otro lado, el controlador basado en homografía al estar linealizado, presenta más oscilaciones que los otros dos.

5.3. Velocidades

Las velocidades de traslación se muestran en la Figura 9. En el controlador basado en imagen y homografía, las velocidades son más suaves y los cambios menos bruscos que en el controlador basado en posición. Claramente se aprecian movimientos lineales en los controladores que utilizan las poses deseadas, por lo que la velocidad en el eje-Y es mayor al iniciar el control.

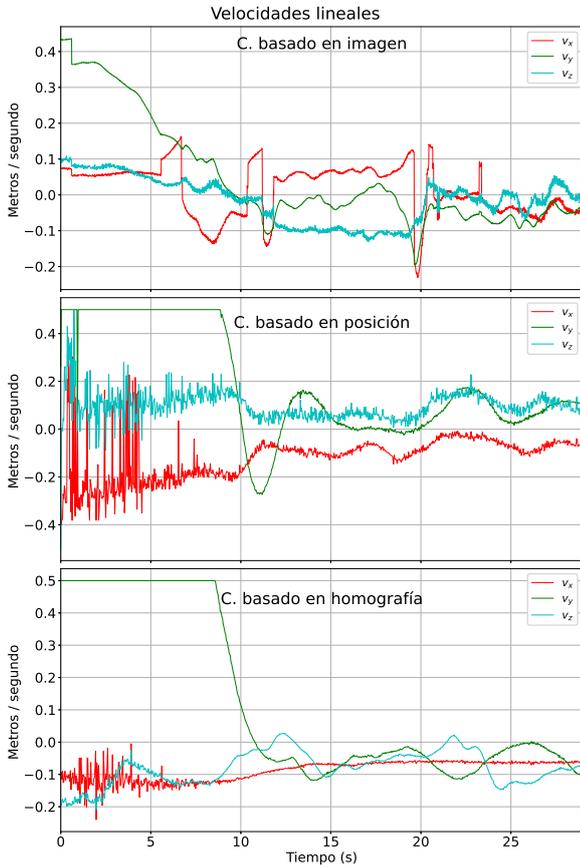


Figura 9: Velocidades lineales.

La velocidad rotacional es únicamente controlada en el eje-Z. En la Figura 10 en $t \approx 25s$, se observa que la velocidad es diferente de cero, debido a que el error es pequeño y la ganancia es constante, lo que genera una zona muerta que no es lo suficientemente grande para generar la rotación en el dron.

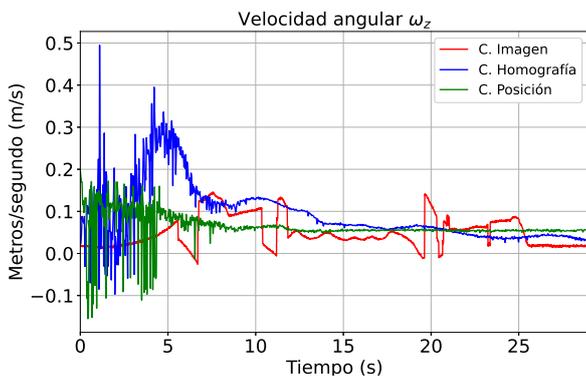


Figura 10: Velocidades angulares.

5.4. Ganancias

Para el caso de las ganancias adaptables (18), se hace una comparación del desempeño de cada controlador, el cual se muestra en la Figura 11. Se aprecia que ocurre una estabilización más rápida en los controladores basados en posición y homografía. Esto se debe a que dichos controladores alcanzan la posición deseada más rápido que el controlador basado en imagen. En los tres casos, la ganancia se estabiliza en un valor cercano a 2.5 1/s.

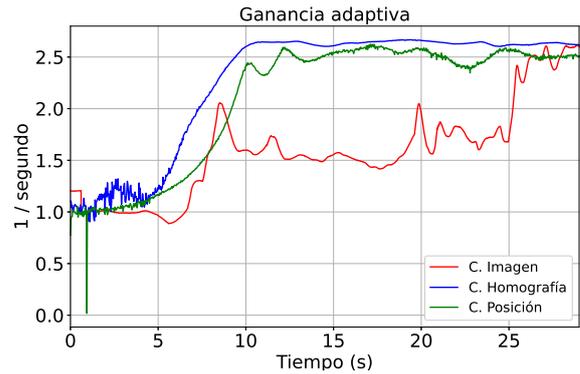


Figura 11: Ganancia.

5.5. Plano de imagen

En las Figuras 12 -14 se muestra el movimiento generado en el plano de imagen de cada controlador. Los puntos en azul son la posición inicial de la cámara, en café los puntos deseados y las estrellas rojas la posición final. Las líneas verdes representan la trayectoria del dron. El controlador basado en imagen muestra una mayor cercanía a la imagen de referencia, lo cual se debe en gran medida al espacio en el que opera el error. Por otro lado, los otros dos controladores calculan una pose estimada a partir de la cual se proyecta la imagen deseada, lo que puede explicar su sesgo con respecto a la referencia.

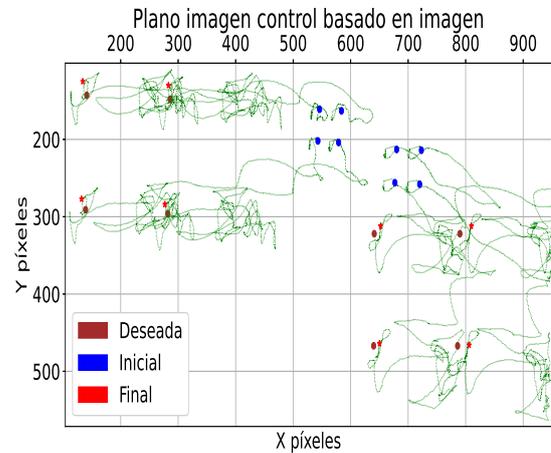


Figura 12: Plano de imagen controlador basado en imagen.

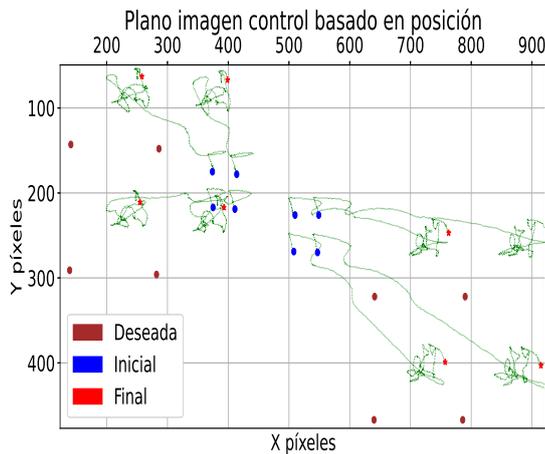


Figura 13: Plano de imagen controlador basado en posición.

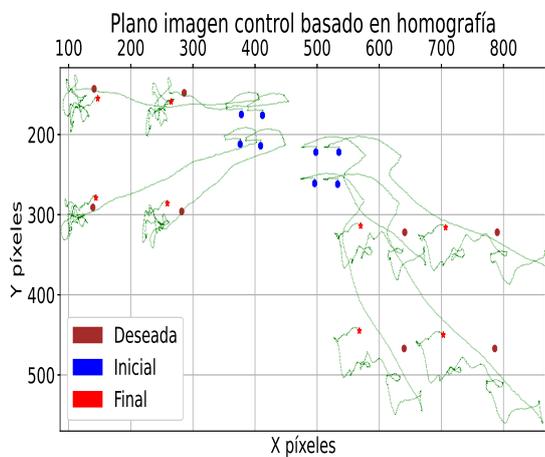


Figura 14: Plano de imagen controlador basado en homografía.

6. Conclusiones

Los distintos controladores empleados en el dron Tello pueden ser útiles para una variedad de actividades, como la navegación en interiores o la creación de mapas del entorno. Por ejemplo, los controladores pueden implementarse junto con herramientas como bag of words, que utiliza una secuencia de imágenes para alcanzar posiciones deseadas.

En este trabajo, se utilizó la misma ganancia para todos los controladores con el fin de hacer una comparación equitativa. Es importante destacar que la sintonización de ganancias es crucial para reducir el error en estado estacionario. Cabe señalar que la ganancia adaptiva mostrada depende del error inicial, para hacer una comparación justa entre los distintos controladores fue necesario estandarizar un proceso de inicialización y evitar lecturas erróneas al momento de adquirir dicho valor.

El enfoque utilizado en el controlador basado en imagen, ayudó a posicionar al dron en la imagen deseada en comparación con los controladores donde el error se basa en las traslaciones y rotaciones. Si se trata de controladores visuales simples que no requieran mucho procesamiento, el dron Tello puede ser capaz de ejecutarlos, ya que el tiempo característico del dron controlado es de cien milisegundos. Sin embargo, si se

trata de controladores más complejos que requieren de mayores recursos, por ejemplo, de una rápida respuesta por parte de la dinámica del dron, es posible que el dron Tello no sea capaz y se requerirá usar otro equipo.

Para agilizar el proceso de inicialización del control visual, como trabajo futuro se pretende incluir la implementación de redes neuronales. Así como también, añadir una ganancia adaptiva para velocidades rotacionales.

Agradecimientos

El primer autor del presente trabajo agradece a CONACYT por el apoyo brindado dentro de su programa nacional de becas, así mismo al CINVESTAV unidad Saltillo por la formación proporcionada al formar parte del grupo de Robótica y Manufactura Avanzada. De igual manera, se agradece al M.C. Neftalí Jonatán González Yances por su apoyo y tutoría en el laboratorio. Finalmente, se agradece a mi esposa, Natalia Alvarado por su amor incondicional, y sobre todo a Dios por todas sus bendiciones y y por todo lo que ha sido realizado, *ad maiorem Dei gloriam*

Referencias

- Benhimane, S. y Malis, E. (2007). Homography-based 2d visual tracking and servoing. *The International Journal of Robotics Research*, 26(7):661–676.
- Caldas, K. A. Q., Benevides, J. R. S., Inoue, R. S., y Terra, M. H. (2022). Autonomous robust navigation system for mav based on monocular cameras. En *2022 INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS)*, International Conference on Unmanned Aircraft Systems (ICUAS), Dubrovnik, CROATIA, JUN 21-24, 2022, pp. 1343–1349.
- Chaumette, F. y Hutchinson, S. (2006). Visual servo control, part i: Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90.
- Chaumette, F. y Hutchinson, S. (2007). Visual servo control, part ii: Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1):109–118.
- Chen, J.-W., Kao, C.-Y., y Lin, Y.-L. (2006). Introduction to h.264 advanced video coding. volumen 2006, pp. 6 pp.–.
- Corke, P. I. y Hutchinson, S. A. (2001). A new partitioned approach to image-based visual servo control. *IEEE Trans. on Robotics and Automation*, 17(4):507–515.
- Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., y Marín-Jiménez, M. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292. ArUco library version 3.2.0.
- Ghaffari, M. y Hall, E. L. (2004). Robotics and nature: from primitive creatures to human intelligence. En Casasent, D. P., Hall, E. L., y Roning, J., editores, *Intelligent Robots and Computer Vision XXII: Algorithms, Techniques, and Active Vision*, volumen 5608, pp. 169 – 176. International Society for Optics and Photonics, SPIE.
- Hussein, A., Al-Kaff, A., de la Escalera, A., y Armingol, J. M. (2015). Autonomous indoor navigation of low-cost quadcopters. En *2015 IEEE International Conference on Service Operations And Logistics, And Informatics (SOLI)*, pp. 133–138.
- Intel Corporation and Willow Garage Inc. (2021). Opencv library. <https://opencv.org/>.
- Lee, M.-F. R. y Yusuf, S. H. (2022). Mobile robot navigation using deep reinforcement learning. *Processes*, 10(12).
- Li, L., Zhang, T., Zhong, H., Li, H., Zhang, H., Fan, S., y Cao, Y. (2021). Autonomous removing foreign objects for power transmission line by using a vision-guided unmanned aerial manipulator. *JOURNAL OF INTELLIGENT & ROBOTIC SYSTEMS*, 103(2).
- Lin, H.-Y. y Peng, X.-Z. (2021). Autonomous quadrotor navigation with vision based obstacle avoidance and path planning. *IEEE ACCESS*, 9:102450–102459.
- Lin, J., Miao, Z., Zhong, H., Peng, W., Wang, Y., y Fierro, R. (2021). Adaptive image-based leader-follower formation control of mobile robots with visibility constraints. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 68(7):6010–6019.

- Machkour, Z., Ortiz-Arroyo, D., y Durdevic, P. (2022). Classical and deep learning based visual servoing systems: a survey on state of the art. *JOURNAL OF INTELLIGENT & ROBOTIC SYSTEMS*, 104(1).
- Malis, E., Chaumette, F., y Boudet, S. (1999). 2½d visual servoing. *Robotics and Automation, IEEE Transactions on*, 15:238 – 250.
- Marchand, E., Spindler, F., y Chaumette, F. (2005). Visp for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotics and Automation Magazine*, 12(4):40–52.
- Nguyen, M. T., Truong, L. H., y Le, T. T. (2021). Video surveillance processing algorithms utilizing artificial intelligent (ai) for unmanned autonomous vehicles (uavs). *MethodsX*, 8:101472.
- Parisi, E. I., Suma, M., Güleç Korumaz, A., Rosina, E., y Tucci, G. (2019). Aerial platforms (uav) surveys in the vis and tir range. applications on archaeology and agriculture. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W11:945–952.
- Pestana, J., Sanchez-Lopez, J. L., Campoy, P., y Saripalli, S. (2013). Vision based gps-denied object tracking and following for unmanned aerial vehicles. En *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1–6.
- Rubio, F., Valero, F., y Llopis-Albert, C. (2019). A review of mobile robots: Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, 16(2):1729881419839596.
- Stanford Artificial Intelligence Laboratory et al. (2018). Robotic operating system.
- Tong, Q., Kim, J., Choi, K., y Lee, Y. S. (2015). High efficient hevcc (h.265) codec design for mobile applications. En *2015 International SoC Design Conference (ISOCC)*, pp. 165–166.
- Toro-Arcila, C. A., Becerra, H. M., y Arechavaleta, G. (2023). Visual path following with obstacle avoidance for quadcopters in indoor environments. *CONTROL ENGINEERING PRACTICE*, 135.