

Control PID basado en odometría visual monocular a un vehículo de superficie marino

Monocular visual odometry based PID control to a marine surface vehicle

MENDOZA-RUIZ, Francisco†, REYES-DE COSS, José Cuauhtémoc*, ALAZKI, Hussain y RULLÁN-LARA, José Luis

Universidad Autónoma del Carmen. Facultad de Ingeniería y Tecnología, México.

ID 1^{er} Autor: *Francisco, Mendoza-Ruiz* / ORC ID: 0000-0003-1865-9930, CVU CONACYT ID: 919556

ID 2^{do} Coautor: *José Cuauhtémoc, Reyes-De Coss* / ORC ID: 0000-0002-3419-905X, Researcher ID Thomson: AAP-3078-2020, CVU CONACYT: 920250.

ID 2^{do} Coautor: *Hussain, Alazki* / ORC ID: 0000-0002-1960-3624, Researcher ID Thomson: M-1706-2017

ID 3^{er} Coautor: *José Luis, Rullán-Lara* / ORC ID: 0000-0002-6007-1025

DOI: 10.35429/JOIE.2020.15.4.29.36

Recibido Julio 20, 2020; Aceptado Diciembre 30, 2020

Resumen

En este trabajo el objetivo es estudiar el comportamiento de un vehículo marino de superficie con un control basado en visión. Utilizamos un algoritmo DMVO (Odometría visual directa monocular) que se basa en la extracción de características de dos imágenes tomadas consecutivamente en dos instantes de tiempo continuos para estimar la posición de un vehículo marino, el cual fue modelado sin tomar en cuenta las perturbaciones externas para aplicar un control de tipo Proporcional Integral Derivativo (PID) basado en visión con el fin de estudiar el comportamiento de este mediante simulaciones, haciendo uso de las herramientas computacionales necesarias para simular un escenario 3D y poder obtener la matriz que contiene la imagen tomada desde el marco de referencia inercial del vehículo. Teniendo en cuenta que un escenario estático que contenga puntos de referencia es necesario para poder así aplicar el algoritmo de control basado en visión propuesto; el escenario virtual fue diseñado en el editor de mundos virtuales de MATLAB que nos permite agregar sensores, como lo son cámaras, y configurar sus parámetros para realizar el estudio, y simulaciones como el seguimiento de trayectorias.

Visión artificial, Control PID, USV

Abstract

The aim of this work is to study the behavior of a marine vehicle applying vision-based control. We use a DMVO algorithm (Direct Monocular Visual Odometry) that is based at the feature extraction of two image frames taken consecutively at two continuous tame frames in order to estimate a marine vehicle's position; vehicle's dynamics are modeled without external perturbations and therefore apply a vision based Proportional Integral Derivative (PID) control for the study of its behavior through simulations using computational tools to simulate a 3D scenario and get the matrix that contains the image from the inertial frame of reference view from the vehicle. First we need a static scenario that contains reference points such that we are able to apply the proposed vision based PID control; the virtual scenario was designed at the MATLAB's virtual world editor that let us add sensors such as cameras, and set their parameters for make the study and simulations such as path following.

Artificial Vision, PID Control, USV

Citación: MENDOZA-RUIZ, Francisco, REYES-DE COSS, José Cuauhtémoc, ALAZKI, Hussain y RULLÁN-LARA, José Luis. Control PID basado en odometría visual monocular a un vehículo de superficie marino. Revista de Ingeniería Innovativa. 2020. 4-15:29-36.

† Investigador contribuyendo como primer autor.

Introducción

Los vehículos autónomos pueden realizar tareas que requieren la precisión para realizar un seguimiento de trayectorias indicadas; es por esto que es de gran importancia su estudio y desarrollo, no sólo para diversas tareas sino para diversos entornos como lo son el terrestre, aéreo y marítimo, por lo que estos últimos están siempre compitiendo en términos de las aplicaciones que pueden realizar, así como su autonomía energética, capacidad de carga y maniobrabilidad (Volker, 2008).

Por otra parte, los vehículos no tripulados más comunes y más estudiados son los de tipo aéreo, denominados UAV (Unmanned Aerial Vehicles por sus siglas en inglés), sobre los cuales se han desarrollado todo tipo de estudios y aplicado todo tipo de algoritmos para el control de su trayectoria y orientación, desde control clásico, inteligente y robusto, comunmente con el soporte de GPSs, sensores inerciales, entre otros; actualmente con las cámaras les permiten estimar su posición y mapear su entorno, dotándolos de la capacidad no sólo de seguir una trayectoria predeterminada, sino de evadir obstáculos lo que les brinda una autonomía aún mayor.

Sin embargo, los estudios e investigaciones realizados enfocados en los vehículos marinos no tripulados antes de la década de los 90 parecían limitarse a la operación remota de estos (Vaneck *et al.*, 1996). Es después de la década de los 90 donde se encuentra un particular interés en ellos para realizar tareas de monitoreo y recolección de datos batimétrico (Manley y Marsh, 2000), por lo que para finales de esa década ya se habían implementado dentro de la estructura de este tipo de vehículos una gran cantidad de sensores en diferentes configuraciones para realizar estas tareas (Bertram, 2008). Con un enfoque similar, se desarrolló el USV denominado SESAMO, el cual fue utilizado en la antártica para una investigación de campo (Cassia y Bono, 2005), así como el ROAZ II, utilizado para realizar tareas de soporte y de manera coordinada con un UUV (Ferreira *et al.*, 2010).

Después de muchos años de desarrollo de los USV, también podemos encontrar investigaciones y experimentos sobre algoritmos para el seguimiento de trayectorias para este tipo de vehículos como se menciona en (Zizzari *et al.*, 2009), donde se tiene el objetivo de dirigir un USV cuyo comportamiento es subactuado, definiendo ciertas leyes de navegación en nivel tanto cinemático como dinámico, utilizando un tipo de control no lineal basado en Lyapunov para el Yaw, así como el control por backstepping (Bubuli *et al.*, 2010) y demostrando así resultados experimentales de la factibilidad de este trabajo aún bajo ciertas perturbaciones como el viento y olas. Un año más tarde, en 2008 con el propósito de converger los errores con respecto al valor de referencia a cero, se mejoran así trabajos anteriores, y se aplican a un USV denominado Charlie (Zezari *et al.*, 2009) desarrollado por la CNR-ISSIA, obteniendo resultados satisfactorios de manera tanto numérica como experimental cumpliendo su propósito inicial.

Las aplicaciones para las que se han utilizado este tipo de vehículos son la caracterización visual y acústica del fondo marino en entornos costeros de aguas poco profundas mediante la adquisición combinada de ecogramas e imágenes de video (Zereik, Bibuli y Bruzzone, 2013), demostrando la efectividad de este tipo de plataformas robóticas. Otra de las aplicaciones recopiladas son las de el monitoreo de las aguas costeras y estuarinas, sin embargo, era necesario el transporte a un laboratorio para su análisis por lo que el desarrollo de la instrumentación in situ sobre el vehículo marino ahora hace posible la recopilación de datos en tiempo real, y con la incorporación de vehículos operados de forma remota se logra una densidad de datos mucho mayor tanto a escala temporal como espacial (Zereik *et al.*, 2015).

Es notable que a medida que se desarrolla la navegación autónoma de los vehículos aéreos, terrestres y acuáticos, la visión y la percepción ha ganado cada vez más atención en los últimos años dado que es posible estimar el movimiento de un robot analizando los cambios de la perspectiva de la cámara a bordo. Los lazos de control con la visión como retroalimentación fueron introducidos para incrementar la flexibilidad y la precisión de los sistemas robóticos (Malis, 2002).

El objetivo de este tipo de sistemas es controlar la dinámica de un robot utilizando información proveniente de un sistema de visión, más generalmente, la visión puede ser usada para controlar diversos sistemas como, por ejemplo, vehículos, drones y submarinos. Los sistemas de visión están generalmente clasificados ya sea dependiendo del número de cámaras o de la posición en la que se encuentran (Sanderson y Weiss, 1980) y los sistemas de una sola cámara son usados generalmente debido a que pueden ser más baratos y su implementación conlleva un menor costo computacional que los sistemas de visión multi-cámara (Hutchinson, 1996).

Por otra parte, usar dos cámaras en una configuración estéreo con un campo de visión común hace más fácil varios problemas severos de computación. Si la o las cámaras se encuentran montadas sobre el robot el sistema es llamado “in-hand”, por el contrario, si la cámara observa el robot desde una perspectiva fija, son sistemas llamados “out-hand”; existen los sistemas híbridos, los cuales cuentan con sistemas de visión In-Hand sobre la estructura, así como un sistema Out-Hand que observa la escena (Malis, 2002).

Modelo dinámico del vehículo marino

Por otra parte, para aplicar control y estudiar el comportamiento de este, es necesario un modelo matemático de la dinámica del vehículo marino con ciertas características, en este caso un vehículo marino tipo Catamarán, cuyo comportamiento ha sido estudiado en (Velueta, 2019) tomando en cuenta no sólo incertidumbres en el modelo sino perturbaciones externas, aplicando SMC (Sliding Mode Control, por sus siglas en inglés) para rechazar las perturbaciones. El modelo está basado en la segunda ley de Newton, como se puede encontrar más detalladamente en (Fossen, 1991) la tarea del modelado, siendo uno de los primeros en estudiar este problema. Definiendo los siguientes vectores de estado mediante las dos ecuaciones siguientes:

$$\eta = [\eta_1 \ \eta_2]^T = [x \ y \ z \ \phi \ \theta \ \psi]^T$$

$$v = [u \ v \ w \ p \ q \ r]^T$$

Donde η es el vector de la posición y orientación del vehículo y v es el vector de velocidades lineal y angular.

Entonces, el modelo completo que incluye la cinemática y la cinética se encuentra definido mediante las siguientes ecuaciones:

$$\dot{\eta} = J(n_2)v$$

$$M\dot{v} + C(v)v + D(v)v = \tau_E + \tau$$

Donde τ es el vector de fuerzas externas y momentos que actúan sobre el cuerpo rígido, τ_E son las fuerzas y momentos de las perturbaciones externas que para nuestro estudio $\tau_E = 0$, y $J(n_2)$ es una matriz diagonal compuesta por dos matrices de rotación $J_1(\eta_2)$ y $J_2(\eta_2)$, definidas como:

$$J_1(n_2) := \begin{bmatrix} c_\psi c_\theta & s_\psi c_\theta + c_\psi s_\theta s_\phi & s_\psi s_\theta + c_\psi c_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\theta + s_\theta s_\phi s_\psi & c_\psi s_\theta + s_\theta s_\psi c_\phi \\ s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}$$

$$J_2(n_2) := \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix}$$

Donde $s_\gamma = \sin(\gamma)$, $c_\gamma = \cos(\gamma)$ y $t_\gamma = \tan(\gamma)$.

Otros parámetros del modelo completo son $M = M_{RB} + M_A$ y $C(v) = C_{RB}(v) + C_A(v)$, donde M_{RB} y C_{RB} son las matrices de inercia y fuerza centrípeta, y M_A y $C_A(v)$, son las matrices de masa añadida debido a la inercia del líquido que rodea el cuerpo rígido. $D(v)$ es la matriz de arrastre hidrodinámico que incluye términos cuadráticos y lineales dentro de ella.

Se puede notar en la matriz $J_1(n_2)$ que existe una singularidad en $\theta = \pm \pi/2$, que para los barcos no es un problema ya que el dominio de trabajo del ángulo Pitch es mucho menor que $\pm \pi/2$. En la figura 1 se pueden apreciar el marco de referencia inercial y el marco referencial, así como el modelo del vehículo marino diseñado para realizar pruebas en tiempo real.

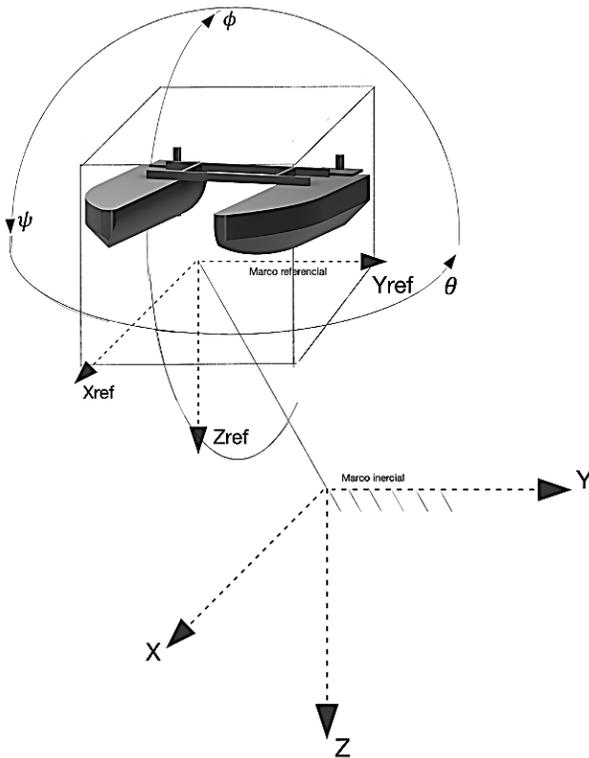


Figura 1 Marcos inerciales y referenciales
Fuente: Elaboración propia

Algoritmo de control PID basado en visión

Para nuestro estudio utilizamos un control PID basado en visión, cuyo sistema está basado en la posición como señal de retroalimentación para la ley de control, cuya ventaja principal es que todo se encuentra en el espacio cartesiano. En la figura 2 podemos encontrar un diagrama representativo del sistema de control que utilizamos en este trabajo.

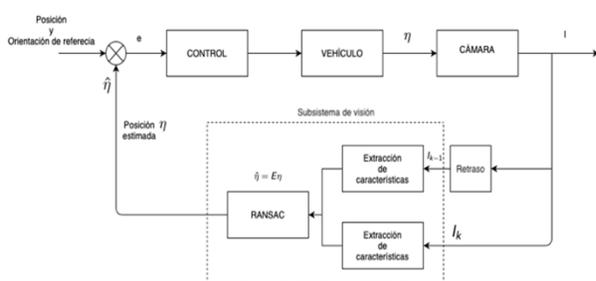


Figura 2 Esquema del sistema de control basado en visión
Fuente: Elaboración propia

Odometría visual monocular 2D-2D

El algoritmo para la estimación sigue el siguiente esquema encontrado en la figura 3, que se encarga de tomar una imagen en el tiempo t y $t - 1$, para así extraer características visuales mediante la técnica SURF (Spedded Up Robus Features), para utilizar el algoritmo RANSAC basado en los puntos extraídos y poder calcular la matriz esencial que nos permite estimar la posición actual del vehículo.

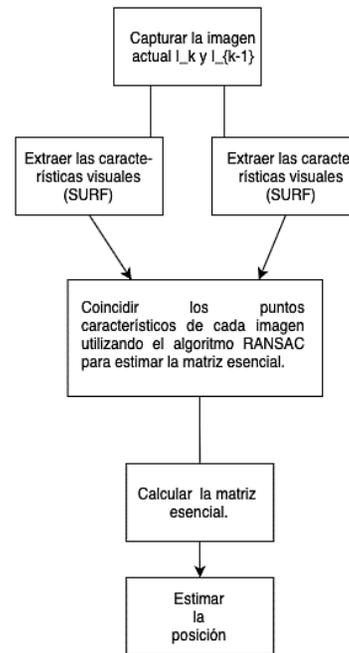


Figura 3 Diagrama de flujo del algoritmo de odometría visual monocular
Fuente: Elaboración propia

La diferencia principal con el esquema estéreo es que en la odometría monocular, es que tanto el movimiento relativo como la estructura tridimensional deben calcularse a partir de datos de trayectoria 2-D. Como se desconoce la escala absoluta, la distancia entre las dos primeras poses de la cámara generalmente se establece en un valor, y cuando se obtiene una nueva imagen, la escala relativa y la posición de la cámara con respecto a los dos primeros cuadros se determinan utilizando el conocimiento de la estructura tridimensional o el tensor trifocal (Hartley y Zisserman, 2003). Entonces, el conjunto de imágenes tomadas se encuentra denotado por $I_{0:n} = \{I_0, \dots, I_1\}$, por lo que la tarea principal del algoritmo es estimar las transformaciones relativas entre las imágenes I_k y I_{k-1} , y aplicar estas a la posición inicial y así obtener la trayectoria de la cámara.

Trabajamos con el modelo para cámara más utilizado, el de un sistema de proyección a través de un agujero, donde $X = (x, y, z)^T$ el punto de la escena proyectada en marco de referencia de la cámara y $p = (u, v)^T$ la proyección sobre en el plano de la imagen medida en pixeles. Este mapeo entre el mundo 3D y el 2D se encuentra dado por la ecuación de proyección perspectiva:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KX = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Entonces, para calcular la correspondencia dada en transformaciones geométricas entre dos imágenes I_k y I_{k-1} de una cámara calibrada, es necesario calcular la llamada Matriz Escencial E, denotada por la siguiente ecuación:

$$E_k \cong \hat{t}_k R_k$$

Donde $t_k = [t_x \ t_x \ t_z]^T$ y:

$$\hat{t}_k = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

Donde el vector t_k representa las traslaciones en las tres dimensiones (t_x, t_y, t_z), y R_k es un matriz ortogonal que contiene la orientación de la cámara en el instante k. Entonces, es posible extraer la traslación y la rotación directamente de la matriz E, utilizando el algoritmo de 5 puntos de Nister, cuya implementación resulta eficiente y se ha vuelto un estándar en la estimación de posición basado en correspondencias 2D-2D [cite].

De lo que se obtiene una solución de la forma $AE = 0$, por lo que resolviendo el sistema los parámetros de E pueden ser calculados, por lo que se SVD (Singular Value Decomposition, por sus siglas en inglés). Este es resuelto utilizando una descomposición eficiente descrita en (Nistér, 2004) y llegar a la forma $E = tR$ donde R es una matriz de rotación de donde es posible obtener directamente la orientación.

Resultados

Para realizar las simulaciones necesarias es necesario utilizar una herramienta de MATLAB llamada Editor de Mundos Virtuales, y así poder obtener una salida de nuestro sistema en términos de una matriz, que es nuestra imagen que utilizamos para estimar la posición mediante el algoritmo de odometría visual directa. En la figura 4 podemos visualizar dicho mundo virtual 3D diseñado, y que nos permite incluir sensores, como lo son cámaras, arreglo de cámara estéreo, sensores de distancia, sensores LIDAR, entre otros.

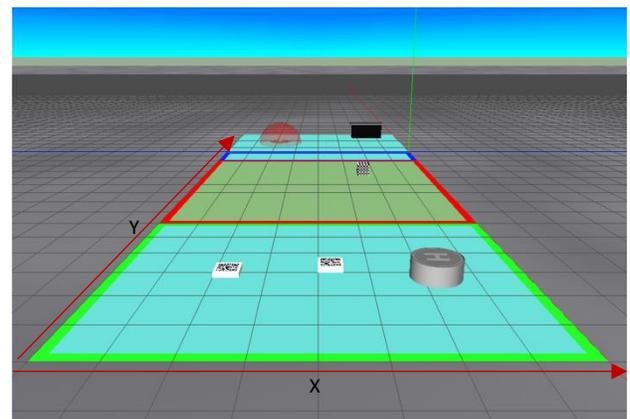


Figura 4 Diseño del mundo virtual creado en MATLAB
Fuente: Elaboración propia

Cabe destacar que, para aplicar este algoritmo de visión, es necesario un escenario estático para poder estimar la posición y obtener los puntos característicos necesarios para que el algoritmo RANSAC pueda encontrar una correspondencia entre los dos cuadros, por lo que podemos observar blancos con patrones definidos que simulan una escena estática. En la figura 5 podemos ver la salida de nuestro sistema, que es una imagen tomada desde la perspectiva del vehículo, y la figura 6 después de extraer las características y agregar los marcadores en estos puntos, para así correlacionar los dos puntos y poder estimar la matriz escencial y aplicar un algoritmo de descomposición, actualizando la posición estimada y usándola como salida de retroalimentación para el control PID.

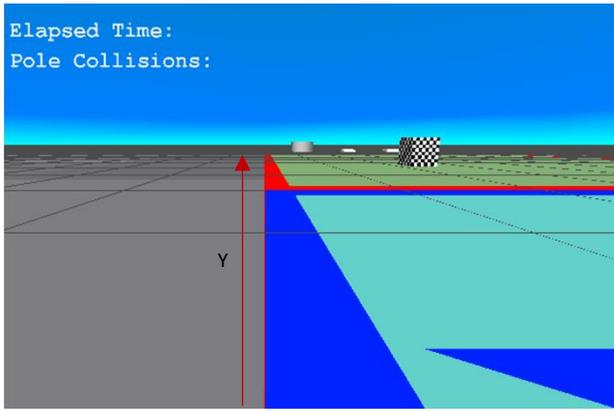


Figura 5 Salida de los sensores virtuales del sistema

Fuente: Elaboración propia

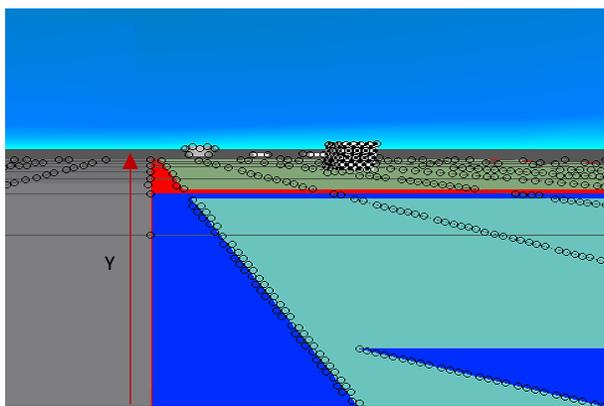


Figura 6 Salida de los sensores virtuales aplicando el extractor de características

Fuente: Elaboración propia

Las constantes proporcional, integral y derivativa fueron obtenidas por ajuste, obteniendo los resultados que se muestran en las figuras 7, 8 y 9. Por otra parte la tabla 1 contiene los parámetros del control PID, que se obtuvieron mediante prueba y error hasta alcanzar la respuesta deseada.

Tipo de constante	Valor
Proporcional	4.243
Integral	0.824
Derivativa	2.439

Tabla 1 Parámetros del controlador PID obtenidos por ajuste

Fuente: Elaboración propia

Otro parámetro importante es la matriz de arrastre hidrodinámico $D(v)$, que depende de la geometría del cuerpo rígido, que contiene una parte lineal y otra no lineal. La parte lineal determinada mediante SolidWorks, y cuyo valor es:

$$D(v) = \begin{bmatrix} -0.5262 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.169 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0408 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0069 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0544 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0913 \end{bmatrix}$$

En la figura 7 podemos observar la salida del sistema y la referencia aplicando el control PID basado en visión, comportándose de una manera adecuada conforme al seguimiento de una trayectoria sinusoidal, definida mediante el siguiente vector de estados:

$$\eta_d = \left[t \quad 10\sin(0.2t) \quad \arctan\left(\frac{y_d}{x_d}\right) \right]^T$$

Con las condiciones iniciales determinadas como $\eta_d = [0 \ 0 \ 0]^T$, y un tiempo de simulación de 50 segundos. La figura 7 muestra la trayectoria deseada y la salida de nuestro sistema, en términos de la variable espacial “x” y “y”.

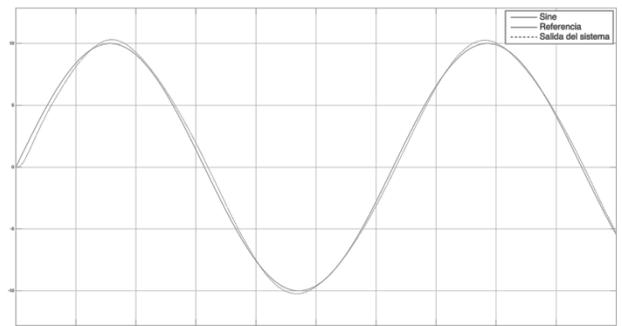


Figura 7 Referencia y salida del sistema. Aplicando control PID basado en visión

Fuente: Elaboración propia

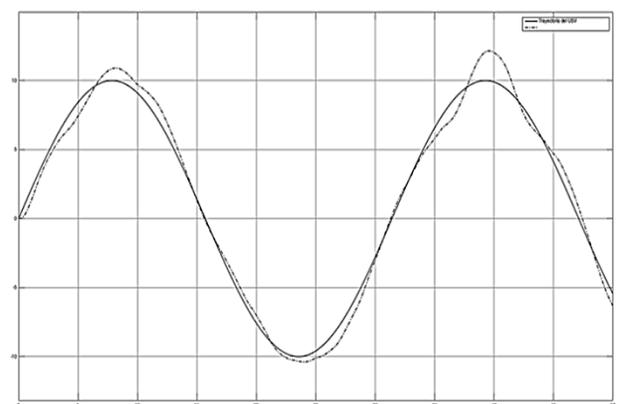


Figura 8 Referencia y salida del sistema. Aplicando control PD convencional

Fuente: Elaboración propia

En la figura 8 se encuentra el error en cada variable de control, como o son la posición y un ángulo de orientación, estos con un valor menor que 1 para la posición, así como el error del ángulo de orientación el cual converge a cero. Por último, en la figura 9 podemos observar cómo se comporta la señal de control para cada variable en el tiempo.

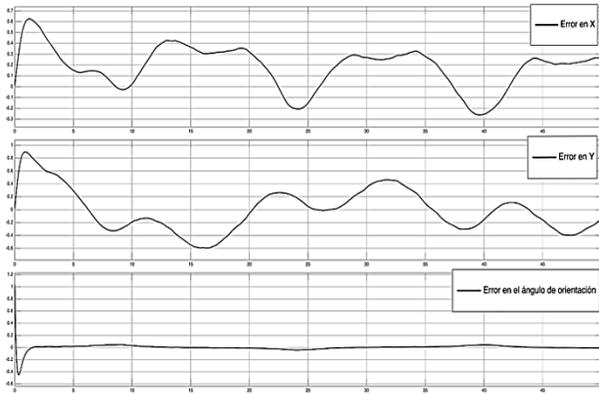


Figura 9 Error con respecto a la referencia dada

Fuente: *Elaboración propia*

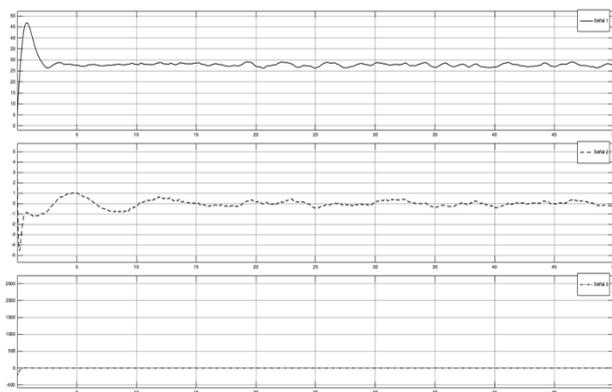


Figura 10 Salidas del control PID

Fuente: *Elaboración propia*

En la figura 8 podemos apreciar el comportamiento de un controlador PD convencional sin visión, en comparación con nuestro método, los resultados son mejores que este.

Agradecimiento

Agradecimientos a la Universidad Autónoma del Carmen por facilitarme los laboratorios de sus instalaciones, material y apoyo humano, así como al Consejo Nacional de Ciencia y Tecnología por el recurso otorgado por pertenecer a la institución dentro del PNPC.

Conclusiones

Se utilizó el modelo de un vehículo marino de superficie sin tomar en cuenta las perturbaciones externas con el fin de estudiar el desempeño al aplicar un control tipo PID basado en visión, llevando el modelo matemático a Simulink de MATLAB con el fin de obtener resultados mediante simulaciones, por lo que también se implementó el algoritmo de visión, y se observó un seguimiento de la trayectoria deseada lo que abre la posibilidad de incluir otro tipo de control con el fin de comparar su desempeño, así como implementar un mejor tipo de estimación basada en visión para mejorar los resultados, ya que el algoritmo MVO utilizado presenta una variación con respecto a la posición real del vehículo. Esto abre la posibilidad de poder implementar este tipo de control en tiempo real sobre un vehículo marino y estudiar su desempeño, sin embargo, es necesario estudiar más tipos de control y visión para escoger el que tenga mejores resultados incluyendo perturbaciones para que su desempeño en tiempo real no se vea afectado.

Referencias

- Bertram, V. (2008). Unmanned surface vehicles-a survey. Skibsteknisk Selskab, Copenhagen, Denmark, 1, 1-14.
- Manley, J. E., Marsh, A., Cornforth, W., & Wiseman, C. (2000, September). Evolution of the autonomous surface craft AutoCat. In OCEANS 2000 MTS/IEEE Conference and Exhibition. Conference Proceedings (Cat. No. 00CH37158) (Vol. 1, pp. 403-408). IEEE.
- Ferreira, H., Martins, R., Marques, E., Pinto, J., Martins, A., Almeida, J., ... & Silva, E. P. (2007, June). Swordfish: an autonomous surface vehicle for network centric operations. In Oceans 2007-Europe (pp. 1-6). Ieee.
- Caccia, M., Bono, R., Bruzzone, G., Spirandelli, E., Veruggio, G., Stortini, A. M., & Capodaglio, G. (2005). Sampling sea surfaces with SESAMO: an autonomous craft for the study of sea-air interactions. IEEE robotics & automation magazine, 12(3), 95-105.

Bibuli, M., Bruzzone, G., Caccia, M., Caiti, A., & Di Lecce, N. (2010). A sliding mode based guidance system for vehicle-following operations. *IFAC Proceedings Volumes*, 43(20), 342-347.

Bibuli, M., Caccia, M., & Lapierre, L. (2007). Path-following algorithms and experiments for an autonomous surface vehicle. *IFAC Proceedings Volumes*, 40(17), 81-86.

Zizzari, A. A., Indiveri, G., Bibuli, M., Bruzzone, G., & Caccia, M. (2009). Path following guidance control with bounded control effort: Application to the charlie unmanned surface vehicle. *IFAC Proceedings Volumes*, 42(18), 109-114.

Zereik, E., Bibuli, M., Bruzzone, G., & Caccia, M. (2013). Jacobian task priority-based approach for path following of unmanned surface vehicles. *IFAC Proceedings Volumes*, 46(33), 114-119.

Zereik, E., Sorbara, A., Bibuli, M., Bruzzone, G., & Caccia, M. (2015). Priority Task Approach for USVs' Path Following Missions with Obstacle Avoidance and Speed Regulation. *IFAC-PapersOnLine*, 48(16), 25-30.

A. C. Sanderson and L. E. Weiss. Image based visual servo control using relational graph error signal. In *Proc. of the Int. Conf. on Cybernetics and Society*, pages 1074–1077, Cambridge, MA, October 1980.

S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.

Velueta, M. J., Rullan, J. L., Ruz-Hernandez, J. A., & Alazki, H. (2019). A Strategy of Robust Control for the Dynamics of an Unmanned Surface Vehicle under Marine Waves and Currents. *Mathematical Problems in Engineering*, 2019.

Fossen, T. I. Nonlinear modelling and control of underwater vehicles. Fakultet for informasjonsteknologi, matematikk og elektroteknikk, 1991.

Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6), 756-770.