2021 IEEE International Conference on Autonomous Systems (ICAS) | 978-1-7281-7289-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICAS49788.2021.9551173

A VISUAL CONTROL SCHEME FOR AUV UNDERWATER PIPELINE TRACKING

Waseem Akram and Alessandro Casavola

Department of Computer Engineering, Modeling, Electronics, and Systems (DIMES),
University of Calabria, Rende, Italy

ABSTRACT

Inspection of submarine cables and pipelines is nowadays more and more carried out by Autonomous Underwater Vehicles (AUVs) because of their low operative costs, much less than those pertaining to the traditional SHIP/ROV-based (Remotely Operated Vehicles) industrial practice, and for the improvements in their effectiveness due to technological and methodological progress in the field. In this paper, we discuss the design of a visual control scheme aimed at solving a pipeline tracking control problem. The presented scheme consists of autonomously generating a reference path of an underwater pipeline deployed on the seabed from the images taken by a camera mounted on the AUV in order to allow the vehicle to move parallel to the longitudinal axis of the pipeline so as to inspect its status. The robustness of the scheme is also shown by adding external disturbances to the closed-loop control systems. We present a comparative simulation study under Robot Operating System (ROS) to find out suitable solutions for the underwater pipeline tracking problem.

Index Terms— Visual control, underwater pipeline tracking, AUV, ROS.

1. INTRODUCTION

Underwater pipelines are used as a means of transportation for oil, gas or other fluid in an underwater environment. These pipelines are prone to extreme conditions such as temperature, pressure, humidity, sea current, dust, and many more. Thus regular inspection and monitoring of these pipelines are of great importance to ensure safe transportation [1].

From past few decades, the application of autonomous underwater vehicles (AUVs) has been found in both industry and research activities as sophisticated solutions for underwater pipeline inspection and tracking. These vehicles are small in size and equipped with intelligent control, sensors, camera, and automatic navigation and localization systems. By employing camera sensors, the visual control strategies are easily implemented on these vehicles for pipeline tracking without requiring huge computations or human efforts [2].

The nature of AUV is dynamic and nonlinear, in that its behavior is coupled with highly translational and rotational dynamics. Further, these vehicles usually operate over long missions and in a dangerous and unknown environment. Strong perturbations due to sea current or actuator failure are also a source of complexity. Thus, more robust and fault-tolerant control strategies are desired. The control strategies are responsible to generate control efforts and drive the vehicle on the desired mission/trajectory. In this regard, proportional integral derivative (PID) [3], sliding mode control (SMC) [4], model predictive control (MPC) [5], linear quadratic regulator (LQR) [6] have proven to be useful control strategies for trajectory following.

The use of camera sensors and laser scanned LIDAR has allowed researchers to develop vision-based tracking systems. In particular, with the help of image processing and computer vision techniques, the desired target can be detected, located and the target trajectory obtained. This advance vision-based tracking system has substituted the traditional tracking methods such as sonar and acoustic methods and eliminated the tracking error.

In scientific literature, different studies have been carried out on underwater pipeline tracking using a vision-based control scheme. In the following, we present a brief review of some previous examples. In [7], the author proposed an image-based control scheme for AUV for solving pipeline tracking problems using fully-actuated vehicles. A plucker coordinates method is applied for an image-based feedback controller. In the experiments, both model uncertainties and external disturbances are considered. In [8], a propulsion technique is adopted for underwater pipeline tracking. For pipeline detection, morphological operations and Sobel edge detection algorithms are applied to the obtained images. Another work presented in [9] studied a vision-based approach for pipeline tracking tasks using a remotely operated vehicle (ROV) vehicle model. The work addressed image quality issues such as low-brightness and suspended materials in the underwater environment. In [10], underwater cable detection is studied using the edge classification method. In this method, edges are extracted and classified using neural networks and support vector machine algorithms. However, controller design aspects are not discussed in that work.

In this paper, a comprehensive approach is discussed to address a vision-based underwater pipeline tracking system for AUVs. The overall general architecture is shown in Figure

978-1-7281-7289-7/20/$31.00 ©2020 IEEE

1. The proposed idea is taken from [3] and further elaborated in this work. In the latter, a vision-based tracking scheme is presented where the trajectory path is generated by using offline a sequence of images and the achieved path tested on a quadrotor. We extend the solution of that previous work by allowing the online elaboration of the images and the generated path is used to solve a pipeline tracking problem for an underwater vehicle model. The proposed solution consists of three modules. The first module is responsible to accomplish some image processing tasks. In the image processing module, we used color-based pipeline detection. The first module provides inputs to the second coordinate recovery module where a reference path is created. In the control module, a PID velocity controller is adopted. In this part, the controller generates the thruster forces based on the path tracking error. The main contribution of this work is to develop a unified approach for the vision-based underwater pipeline tracking problem using AUV in the presence of external disturbances.

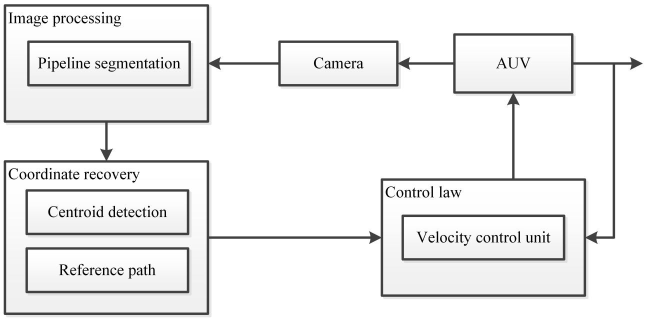

Fig. 1. Vision-based tracking scheme.

2. SIMULATION ENVIRONMENT

In this work, we used the “Unmanned Underwater Vehicle Simulator” (UUV) simulator [11]. It is a set of packages that include plugins and ROS applications that allow one to carry out simulations of underwater vehicles in Gazebo and Rviz tools.

The simulation environment consists of a seabed scene that comes along with the UUV simulator package, and an underwater vehicle used as a tracker. To simulate the underwater pipeline scenario, we have modified the seabed scene by adding a single pipeline using the blender tool. The simulation is performed on ROS melodic distribution installed on Ubuntu 18.04.4 LTS.

2.1. The rexrov model

In this work, we spawned the “rexrov-default” vehicle model available in “uuv-simulator” package of ROS [11]. The vehicle model consists of a mechanical base with a camera and additional sensing devices such as IMU (an inertial measurement unit) and LIDAR. The camera sensor is available with the vehicle model and installed at the bottom of the model that faces downward. The ROS camera easily generates image frames that have 640 pixels in width and 490 pixels in height. The “cv-bridge” plugin converts images from ROS to OpenCV and vice versa.

2.2. Initialization

The simulation initial conditions are described as follows. Let us consider an underwater pipeline placed or suspended on the surface of the ocean floor. A rexrov vehicle model is spawned on the simulation world with default positions and the same for the pipeline, as shown in Figure 2. At the start, the vehicle stays static in the sea and the camera is facing downward.

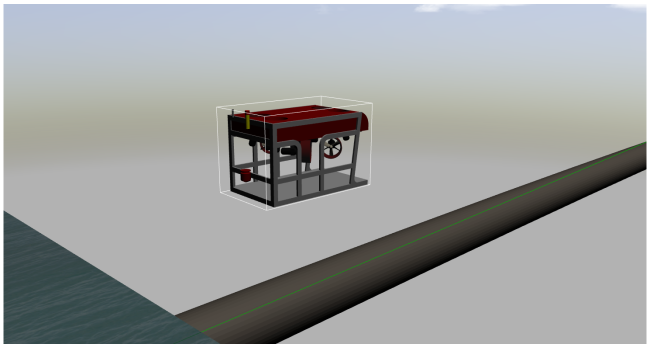

Fig. 2. Testing platform visualized in the Gazebo.

3. PROPOSED SCHEME

The vision-based underwater pipeline tracking system is achieved by using an integrated three-module-based approach. Next, each module is described as follows:

3.1. Image processing

The downward-looking camera installed on the vehicle is used to collect the image sequences to the ROS. The cv-bridge package is used to convert the obtained images from ROS to OpenCV format. Then, various image processing steps are performed to get the pipeline located in the image frame.

The following Algorithm 1 describes the basic steps taken for pipeline detection. The obtained image from the camera is shown in Figure 3. First, the image is resized to a size where the pixels and pipeline are not disturbed in the image. This is required to reduce the size of the image so the algorithm takes less time for computation. Next, the image is converted to the hue saturation value (HSV) format. The first image is used to get the HSV upper and lower values of the pipe. Then, these values are used to segment the pipe from the rest of the image. Once the pipe is successfully segmented, its pixels details are

Algorithm 1 Steps for pipe detection

```
Initialization:
  set: lower and upper HSV of pipe in the image

Online-Phase
  for $t > 0$ do
      Import image from ROS to OpenCV
      Convert: image to HSV
      segment: the pipe pixels from the rest in the image
      save: segmented pipe information in the list
```

stored in a list that will be used in the coordinate recovery module.

(a) Croped image (b) HSV image (c) masked image

Fig. 3. Image processing.

3.2. Coordinate recovery

In order to get the x and y centroid of the pipe, the “moment“ method in OpenCV is used. The moment method calculates the weighted average of the image pixels of the object. In our case, here the pipe object, that was previously segmented and masked in the image, is used as input for the “moment” method. The method returns the centroid cx and cy coordinates of the pipe in the image. This is done by using the following formulas:

$$cx = \frac{1}{n}\sum_{1}^{n} x_i, cy = \frac{1}{n}\sum_{1}^{n} y_i \tag{1}$$

Here, x_i and y_i show x and y coordinates values for each pixel of the pipe and n is the total number of pixel values of the pipe. Next, the target relative x and y positions with respect to vehicle are calculated by using following formulas:

$$\begin{aligned} x &= (cx - width/2) * sensitivity, \\ y &= (cy - height/2) * sensitivity \end{aligned} \tag{2}$$

Here, the width and height of the cropped image are considered, and $sensitivity > 0$ is used to convert the pixels values to the real values as used by the simulated world frame. With the x and y values, a reference path is created during the simulation and given as input to the controller module.

3.3. Control law

In order to track the pipeline, a point-to-point reference path tracking control law is created by using the centroid information of the pipe in the image frame. The basic idea here is that the vehicle should move in order to keep the pipe in the center position of the image frame. This is achieved by setting a threshold such that the vehicle is allowed to move in the longitudinal direction along the reference path if the current error is greater than the threshold. A switching logic as shown in equation (3) is defined to accomplish this task.

$$\begin{aligned} vel.linear(x) &= vel(t), & |e(t)| > \quad threshold \\ vel.linear(x) &= constant, & |e(t)| \leq \quad threshold \end{aligned} \tag{3}$$

where the linear velocity of the vehicle along the x-axis $vel(t)$ is a setpoint produced by the PID control law (4) in the case the tracking error $e(t)$ is greater than the $threshold = 0.4$. The control law is responsible to reduce the tracking error $e(t)$ during the simulation. Once the tracking error is reduced below the threshold, a small positive constant number as velocity setpoint is given to the vehicle for the movement. Specifically, the thruster manager package provided by the “uuv-simulator“ takes the velocity setpoint as an input that is bounded as $[-0.75, -1.3, -0.8] \leq v(t) \geq [0.75, 1.3, 0.8]$ and generates the corresponding thruster’s forces. The control law is defined as:

$$vel(t) = k_p e(t) + k_i \int_0^t e(t)dt + k_d \frac{d}{dt} e(t) \tag{4}$$

Here, $k_p > 0, k_i > 0$ and $k_d > 0$ are the proportional, integral, and derivative gains respectively.

In the rest of this section we discuss the robustness of the scheme to the image blur that causes noise in the estimation of the actual position of the pipeline in the image. This is done by adding a random error in the current vehicle position throughout the simulation. The noise signal is represented by: $N \sim mN(0, 1)$ where N indicates Normal distribution with zero mean and m is a positive multiplicative term. Because of the noise, the current position is updated as follows:

$$cp(t) = cp(t) + N(t) \tag{5}$$

where cp shows the current position of the vehicle. Correspondingly, the updated error becomes

$$e(t) = tp(t) - cp(t) = tp(t) - cp(t) - N(t) \tag{6}$$

where $e(t)$ is the current error, tp target position, cp current position and N is the noise. The velocity command is updated as follows:

$$vel(t) = k_p e(t) + k_i \int_0^t e(t)dt + k_d \frac{d}{dt} e(t) + \\ k_p N + k_i \int_0^t N dt + k_d \frac{d}{dt} N \tag{7}$$

4. SIMULATION RESULTS

For simulation purposes, different case scenarios have been implemented as follows:

1. No-disturbance: in this case, no external disturbances are considered and handled in the controller part. This scenario is used as baseline solution.
2. PID1: in this case, $k_p = 0.5$, $k_i = 0.05$, and $k_d = 0.1$ are set along with the external disturbance 0.1
3. PID2: in this case, $k_p = 0.1$, $k_i = 0.05$, and $k_d = 0.01$ are set along with the external disturbance 0.1

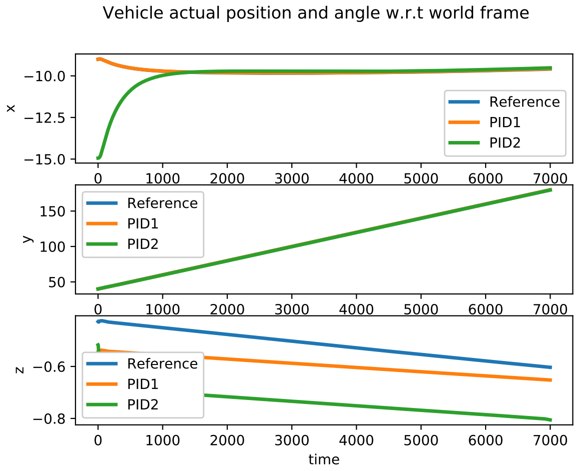

Fig. 4. Vehicle positions w.r.t simulation world frame.

Next, we discuss the simulation results. In Figure 4, the vehicle linear positions x, y and z are shown for time $t_s > 0$. The control was designed to control the x and z position for tracking purposes. In the case of PID1, the vehicle initial x position was set to -12. After a few iterations, the controller reduced the tracking error to zero. In the case of PID2, the initial x position was set to -15. It is observed that the tracking error was eliminated after a few iterations. The parameters configuration used in the PID1 case kept the angular position z close to the reference position. In Figure 5, the vehicle orientations are shown. Here, it is noticed that the vehicle stays stable through simulations. In Figure 6, the control efforts are shown that are required to control the x and z position of the vehicle. As compared with the previous work [3], the presented scheme does not require a predefined tracking path. The use of a built-in camera of AUV and in-taking few image processing steps makes it a simpler scheme for tracking problems online during run-time. Furthermore, the noise rejection capability of the scheme in the tracking position suggests implementing it in the real-world underwater environment.

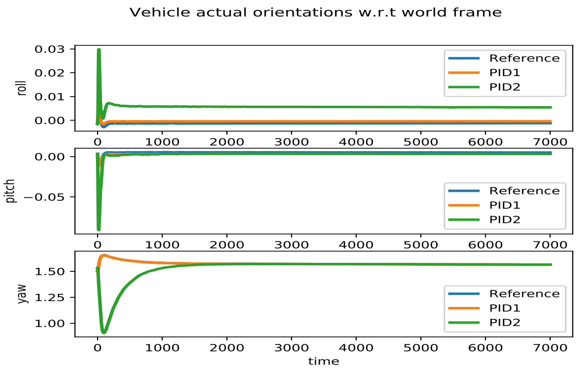

Fig. 5. Vehicle orientations w.r.t simulation world frame.

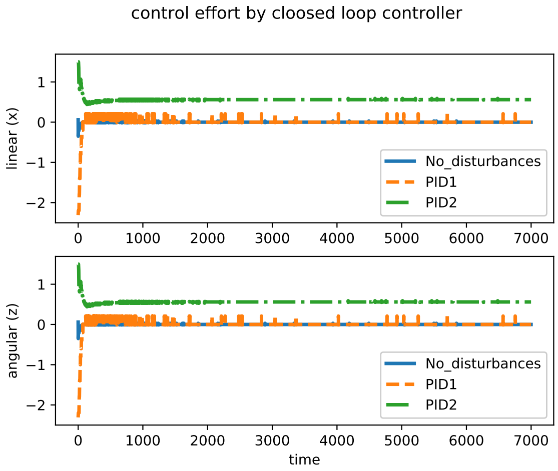

Fig. 6. Control effort of the PID controllers.

5. CONCLUSIONS

The visual-based control law for underwater pipeline tracking through simulation is considered in this work. A testing platform ROS and gazebo simulator is used. A UUV simulator is adopted and modified to perform the tracking of the underwater pipeline. The proposed solution consists of three modules: image processing, coordinate recovery and PID control law. The performance of the tracking system is shown by adding and handling the external disturbance in the controller part. The system is simulated for a straight pipeline laid over the seabed. The simulation results showed a successful tracking scheme regardless of the external disturbance.

6. REFERENCES

[1] Gøril M Breivik, Sigurd A Fjerdingen, and Øystein Skotheim, "Robust pipeline localization for an autonomous underwater vehicle using stereo vision and echo sounder data," in *Intelligent Robots and Computer Vision XXVII: Algorithms and Techniques*. International Society for Optics and Photonics, 2010, vol. 7539, p. 75390B.

[2] Caoyang Yu, Xianbo Xiang, Mingjiu Zuo, and Hui Liu, "Underwater cable tracking control of under-actuated auv," in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*. IEEE, 2016, pp. 324–329.

[3] Yusheng Wei and Zongli Lin, "Vision-based tracking by a quadrotor on ros," *Unmanned Systems*, vol. 7, no. 04, pp. 233–244, 2019.

[4] Jiehao Li, Junzheng Wang, Hui Peng, Yingbai Hu, and Hang Su, "Fuzzy-torque approximation-enhanced sliding mode control for lateral stability of mobile robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[5] Qifan Tan, Penglei Dai, Zhihao Zhang, and Jay Katupitiya, "Mpc and pso based control methodology for path tracking of 4ws4wd vehicles," *Applied Sciences*, vol. 8, no. 6, pp. 1000, 2018.

[6] Alvaro Ortiz, Sergio Garcia-Nieto, and Raul Simarro, "Comparative study of optimal multivariable lqr and mpc controllers for unmanned combat air systems in trajectory tracking," *Electronics*, vol. 10, no. 3, pp. 331, 2021.

[7] Guillaume Allibert, Minh-Duc Hua, Szymon Krupínski, and Tarek Hamel, "Pipeline following by visual servoing for autonomous underwater vehicles," *Control Engineering Practice*, vol. 82, pp. 151–160, 2019.

[8] G Manikandan, S Sridevi, and J Dhanasekar, "Vision based autonomous underwater vehicle for pipeline tracking," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 1, pp. 2347–6710, 2015.

[9] A El-Fakdi, M Carreras, and J Batlle, "Direct policy search reinforcement learning for autonomous underwater cable tracking," *IFAC Proceedings Volumes*, vol. 41, no. 1, pp. 155–160, 2008.

[10] Mehdi Fatan, Mohammad Reza Daliri, and Alireza Mohammad Shahri, "Underwater cable detection in the images using edge classification based on texture information," *Measurement*, vol. 91, pp. 309–317, 2016.

[11] Musa Morena Marcusso Manhães, Sebastian A. Scherer, Martin Voss, Luiz Ricardo Douat, and Thomas Rauschenbach, "UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation," in *OCEANS 2016 MTS/IEEE Monterey*. sep 2016, IEEE.