

OBJECT TRACKING CONTROL USING A GIMBAL MECHANISM

D. J. Regner¹, J. D. Salazar^{1,2}, P. V. Buschinelli¹, M. Machado¹, D. Oliveira¹, J. M. Santos³, C. A. Marinho³, T. C. Pinto^{1*}

¹ Mechanical Engineering Department, Labmetro/UFSC - Florianópolis, SC, Brazil

² Automation and Systems Department/UFSC - Florianópolis, SC, Brazil

³ CENPES/Petrobras, Rio de Janeiro, RJ, Brazil

KEY WORDS: ROS, Gazebo, Simulation, Drone, Gimbal, Inspection, Deep learning, Photogrammetry.

ABSTRACT:

This work describes a control solution for real time object tracking in images acquired for a RPAS on an object inspection environment. This, controlling a 3-axis gimbal mechanism to control a camera orientation embedded to a RPAS, using its image processed for feedback. The objective of control is to maintain the target of interest at the center of the image plane. The proposed solution uses a YOLOv3 object detection model in order to detect the target object and determine, thru rotation matrices, the new desired angles to converge the object's position to the center of the image. To compare results of the proposed control, a linear control was tuned using a linear PI algorithm. Simulation and practice experiments successfully tracked the desired object in real time using YOLOv3 in both control approaches presented.

1. INTRODUCTION

The use of RPAS (Remotely Piloted Aircraft System), also named UAV (Unnamed Aerial Vehicle) equipped with multiple sensors has been increasing over the last few years in a wide range of industrial applications, included the oil and gas (O&G) industry, to perform tasks such as visual inspection of structures and components in areas where human access is limited (i.e., offshore platforms) and the activities are expensive, dangerous and high time-consuming. In that context, one of the main components in the offshore platforms are the flexible risers, which are pipelines in charge of transporting oil, gas, water and cables between subsea structures and the platform on water surface (Salazar et al., 2020) (Wang et al., 2016). The inspection of this type of component, as shown in Figure 1, is done by industrial climbers who perform manual measurements and photographic record of point of interest. In this case, the RPAS can be used to perform inspection of risers and the others components in the offshore platform.



Figure 1. Industrial climbers inspecting the emerged riser section.

In that context, the introduction of RPAS in the visual inspection processes helps to perform inspections of different types of components quickly, safely and economically. Despite these

advantages, quantitative or geometric studies of the structures and components have not been conducted to date. In this way, techniques such as photogrammetric 3D reconstruction can be used to perform geometric measurements of risers from images captured by RPAS. In order to generate a good measurement result using photogrammetry, a set of requirements must be fulfilled, such as sequential and overlapping image acquisitions, spatial resolution, object texture and camera positioning network (Luhmann et al., 2014) (Buschinelli et al., 2020) (Salazar et al., 2020). For overcome that, the RPAS must execute specialized trajectories, varying its position and orientation keeping the object within the field of View (FoV) of the camera.

One of the most challenging problems to execute the specialized trajectories by RPAS at offshore oil and gas platforms is maintaining the interest object for inspection centered within the FoV of the camera. This is due to uncontrollable environment variables, like wind, that reduce the flight time of the RPAS and produce unexpected movements making it difficult to image acquisition. Another variable is the proximity of the robot to the large metal structure of the offshore platform, this condition makes the robot more susceptible to electromagnetic interference and intermittent GNSS signal loss. Therefore, are necessary highly skilled pilots to prevent possible accidents with the aircraft. Additionally, this type of inspection requires a second pilot to control the gimbal (that leads with camera movements) maintaining the interest object of inspection within the FoV of the camera. Thus, reducing the impact of the variables mentioned above.

In this paper, an object tracking solution using existent hardware is proposed in order to maintain an interest object (e.g. the riser) centered within the FoV of the camera, compensating the RPAS movements while performing the specialized trajectories for photogrammetric inspection processes. For this, a state-of-art convolutional neural network (CNN) based model YOLO (version 3) (Redmon and Farhadi, 2018) is used for detecting interest object in the image. Once it detected, a controller based in inverse kinematic from rotation matrix determines the new angles positions to be applied by the gimbal mechanism, maintaining the object centered within the Field of view. As

* Corresponding author. tiago.pinto@ufsc.br

a result, this control helps the photogrammetric image acquisition processes. Thus, taking another step towards the automatic inspection processes using RPAS.

2. RELATED WORK

With the increased use of RPAS for inspections, monitoring and obstacle avoidance specialized tasks. A system for tracking objects through image processing has been increasingly developed in order to automate the aircraft's flight path and keep the objects within the FoV of the camera (Altan and Hacıoğlu, 2020) (Cunha et al., 2019).

Similar researches has been conducted to process the images acquired by the RPAS and tracking the target objects. (Yuan et al., 2015) uses RPAS to process images and detect forest fires to posterior track it. While (Greatwood et al., 2017) presents a track control of RPAS to follow an desired object moving on ground to determine its path, differently than (Kendall et al., 2014) that describes a RPAS path planning of a stationary objects. Thus, in this paper is described an object (riser) tracking strategy to perform an inspection on an offshore platform using the detected object to determine the camera orientation to maintain it on gimbal's camera FoV to acquire images in a specialized trajectory for photogrammetry processes.

2.1 You Only Look Once (YOLO) and YOLOv3

Nowadays, RPAS are increasingly being used with different deep learning techniques, and more specifically, those related to convolutional neural networks (CNNs). In that context, YOLO (You Only Look Once) is one of the state-of-the-art CNN-based object detection models for computer vision task in real time. It is has been used for a range wide of applications, such as traffic monitoring (Benjdira et al., 2018), fire detection (Jiao et al., 2020), industrial inspection (Salazar et al., 2020), and rescue applications (McGee et al., 2020)

The YOLO algorithm adopts a single CNN backbone to directly predict bounding boxes and class probabilities from the entire images in one evaluation. Compared with the Faster R-CNN network (Ren et al., 2017), the YOLO network transforms the detection problem into a regression problem. As shown in Figure 2, the input image is divided into an $S \times S$ grid, and each grid is responsible for predicting only one object. If the center of the object falls into a grid cell (for example, the yellow dot represents the center of the riser in the input image), that cell is responsible for the detection of that object. Each grid cell predicts B bounding boxes and confidence scores for those boxes. These confidence scores reflect how likely the box contains an object and how accurate is the boundary box. Simultaneously, C conditional class probabilities are predicted in each cell, regardless of the number of the bounding box number (B). The conditional class probability is the probability that the detected object belongs to a particular class (Redmon et al., 2016) (Salazar et al., 2020).

The YOLOv3 (Redmon and Farhadi, 2018) network is an improvement from its predecessors YOLOv1 (Redmon et al., 2016) and YOLOv2 (Redmon and Farhadi, 2017). It uses multi-scale prediction to detect the final target using the Feature Pyramid Networks idea (Lin et al., 2017). Its backbone named Darknet-53 is a 53 layered CNN that uses skip connections network ideas inspired by Resnet network (Szegedy et al., 2017) to extract features from the images to improve the trade-off between the speed and accuracy.

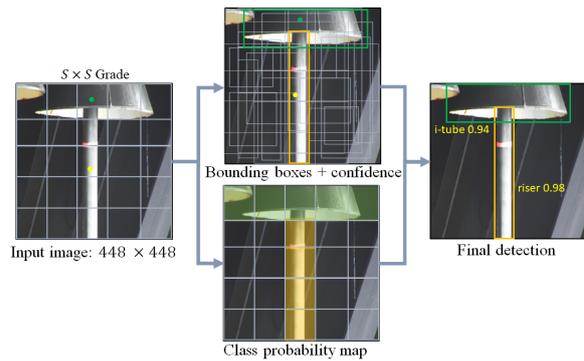


Figure 2. YOLO detection.

2.2 Robot Operating System (ROS)

The Robot Operating System (ROS) is an open source framework with a diversity of libraries and tools collections to standardize a software communication between robot components on a distributed computing resources (Open Source Robotics Foundation, 2010a) (Yoonseok Pyo, 2017) (Joseph, 2018). The concept aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms (Quigley et al., 2015).

2.3 Simulation Environment

Gazebo is a 3D dynamic simulator performing physics simulation at a much higher degree of fidelity, similar to game engines. Also has the ability to simulate populations of robots in complex indoor or outdoor environments. Contains a suite of sensors, and interfaces for both users and programs (Open Source Robotics Foundation, 2014).

2.4 Gimbal mechanism

The gimbal is a mechanism used to control the position of the camera. It is a mechanical device which is designed using rings mounted on axes at right angles to each other. The gimbal's end effector (typically a camera), is in an unstable environment arranged in a stable position using this mechanical device to rejects disturbances such as RPAS motor friction, unbalanced aerodynamics, spring torque forces and structure vibrations (Jakobsen and Johnson, 2005). A traditional use for a Gimbal mechanism, Figure 3, is to stabilize the camera attached on a RPAS for improve images acquisitions.



Figure 3. Example of Three Degrees of Freedom (DoF) Gimbal mechanism used on RPA.

If the position of the camera is not compensated or stabilized during the image acquisition process or autonomous target tracking, problems such as blurred images and focal loss can

generated. (Rajesh and Kavitha, 2016). In this work a 3-axis gimbal is mounted above or under the body of the aircraft and it has an individual controller to each motor to move the gimbal's angles *yaw-roll-pitch* from Z-X-Y axis, respectively. The schematic diagram of gimbal kinematics with 3 revolute joints is shown in Figure 4 (Rajesh and Kavitha, 2016) (Kulkarni and Mohanty, 2013).

To simplify, the intersection of gimbal's coordinate system axis is on the center of camera's optical center, compensating for all the angular movements of the hull It is attached to, and the general principle of the axes arrangement assures it is able to avoid the gimbal lock state during its operation. The configuration, *yaw over pitch over roll*, allows up to 90 degrees of roll or pitch movement by the aircraft before a gimbal lock occurs, which of course is unlikely to ever happen (Tiimus and Tamre, 2010).

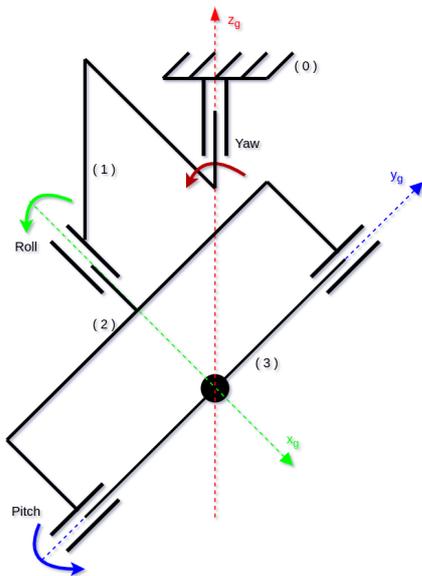


Figure 4. A three revolute joints Gimbal's Kinematic.

To control the position of the gimbal, it must be determine the Gimbal forward kinematics based on rotation matrix of each angle's axis. Initially body (0) is attached to hull, and the transformation from body (0) to body (3) can be described as a multiply of rotation matrices Z-X-Y at gimbal coordinate system.

The rotation matrix of *yaw* angle from the frame of body (0) to the frame of body (1) is:

$$R_1^0 = \begin{bmatrix} \cos(yaw) & -\sin(yaw) & 0 \\ \sin(yaw) & \cos(yaw) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The rotation matrix of *roll* angle from the frame of body (1) and the frame of body (2) is

$$R_2^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(roll) & -\sin(roll) \\ 0 & \sin(roll) & \cos(roll) \end{bmatrix} \quad (2)$$

The rotation matrix of *pitch* angle from the frame of body (2) and the frame of body (3) is

$$R_3^2 = \begin{bmatrix} \cos(pitch) & 0 & \sin(pitch) \\ 0 & 1 & 0 \\ -\sin(pitch) & 0 & \cos(pitch) \end{bmatrix} \quad (3)$$

The forward kinematics is given multiplying (10) (2) (3), resulting in matrix R_3^0 .

$$R_3^0 = \begin{bmatrix} C_y C_p - S_y S_r S_p & -C_r S_y & C_y S_p + C_p S_y S_r \\ C_p S_y + C_y S_r S_p & C_y C_r & S_y S_p - C_y C_p S_r \\ -C_r S_p & S_r & C_r C_p \end{bmatrix} \quad (4)$$

where the variation of each angles results in a new X-Y-Z point to the gimbal end effector, the camera's FoV.

The Gimbal mechanisms are controlled using a fixed coordinate system, (Siciliano et al., 2009) describes a different multiplication order to fixed coordinate system, resulting in \bar{R}_3^0 matrix.

$$\bar{R}_3^0 = \begin{bmatrix} C_y C_r + S_y S_r S_r & C_r S_y S_r - C_y S_r & C_r S_r \\ C_r S_y & C_y C_r & -S_r \\ C_y S_r S_r - C_r S_r & C_r C_y S_r + S_y S_r & C_r C_r \end{bmatrix} \quad (5)$$

3. PROPOSED TRACKING CONTROL

The system of the aircraft communicate between peripherals using ROS. When the RPAS is started a ROS node initializes topics that provide data from sensors embedded in the aircraft, such as position of the gimbal and image acquired from camera, allowing to perform YOLOv3 object detection inference and use the proposal track model to determine the gimbal's angles based on object position in the image.

To compare the performance of the proposed control method were tuned two different approaches of control techniques, a non-linear (inverse kinematic), and conventional linear (proportional-integral).

The optical center of the camera will be positioned at the intersection of gimbal's axis to simplify the problem (Figure 5). Here, the FoV of the camera is projected along X_g axis, thus reducing the system to a robotic problem of two DoF. When varying *pitch* angle result in a movement of the FoV along Y_c axis from image frame, respectively occurs when varying *yaw* angle of gimbal mechanism, the movement between object and camera's FoV is given along X_c .

The proposal control approach is based on gimbal's inverse kinematic formulation using (5). The initial position of the mechanism is centered at $[0\ 0\ 0]^T$, we can determine the distance of the object detected from camera as a nominal distance called x_{fov} , where, usually, is a fixed distance between RPAS and risers, this to guarantee parameter robustness in the inspection process. Considering the FoV of the camera, the initial condition is $[x_{fov}\ 0\ 0]^T$, resulting in $[X_g\ Y_g\ Z_g]^T$ matrix solution:

$$\begin{bmatrix} X_g \\ Y_g \\ Z_g \end{bmatrix} = \begin{bmatrix} C_y C_p & -C_y S_p & S_p \\ S_y & C_y & 0 \\ -C_y S_p & S_y S_p & C_p \end{bmatrix} \begin{bmatrix} x_{fov} \\ 0 \\ 0 \end{bmatrix} \quad (6)$$

where *roll* angle does not change the position of the FoV, just its orientation (portrait or landscape), it has been held at 0 degrees.

From angles matrix of Forward Kinematic, it is determined the Inverse Kinematic formulas of angles based on desired spatial coordinate *X-Y-Z*. *Yaw* and *pitch* can be defined as second and third row of (6)

$$\sin(yaw) = \frac{Y_g}{x_{fov}} \quad (7)$$

$$\sin(pitch) = \frac{-Z_g}{x_{fov} \cos(yaw)} \quad (8)$$

representing the new angle's position:

$$yaw = \arcsin\left(\frac{Y_g}{x_{fov}}\right) \quad (9)$$

$$pitch = \arcsin\left(\frac{-Z_g}{x_{fov} \cos(yaw)}\right) \quad (10)$$

The inverse kinematic formulation angles position are described as (9) and (10), two non-linear dependently equations to predict the angles values based on spatial *X-Y-Z* of gimbal's frame. Although the position data is provided at image coordinate system in pixel, we must transform it to meters. The new angles are determined to move the center of image above to object position, resulting in a reject of disturbance controller when is using an incremental value of angles.

The image projection of the camera and its coordinate system are illustrated in Figure 6. Although the image of the camera is projected along X_g axis at a nominal distance x_{fov} , the frame of gimbal G will always be centered at frame image C . Axis Y_g and Z_g from gimbal mechanism are, respectively, parallel constrained to X_c and Y_c axis of image plane.

To determine the position error of the object related to gimbal's frame, two error function based on desired reference (center of image) were described. Gimbal's axis Z_g and Y_g are expressed as the references less the object position O :

$$Z_g = \left(\frac{pixel\ width}{2} - X_o\right) SR \quad (11)$$

$$Y_g = \left(\frac{pixel\ height}{2} - Y_o\right) SR \quad (12)$$

where X_o and Y_o are coordinates from the bounding box detected by YOLOv3, $\frac{pixel\ width}{2}$ $\frac{pixel\ height}{2}$ are the reference position of gimbal's frame and SR a sensitive parameter of error.

The parameter SR , spatial resolution, used to convert the error signal from pixel to meters and determine the sensibility of error signal. Spatial resolution is a measure of the smallest object that can be resolved by the sensor, or the ground area imaged for the camera's FoV, or the linear dimension on the ground represented by each pixel (Liang et al., 2012). Although a known distance between optical sensor and object is necessary to determine precisely the parameter, it has been used a distance of one meter for the error sensitive parameter.

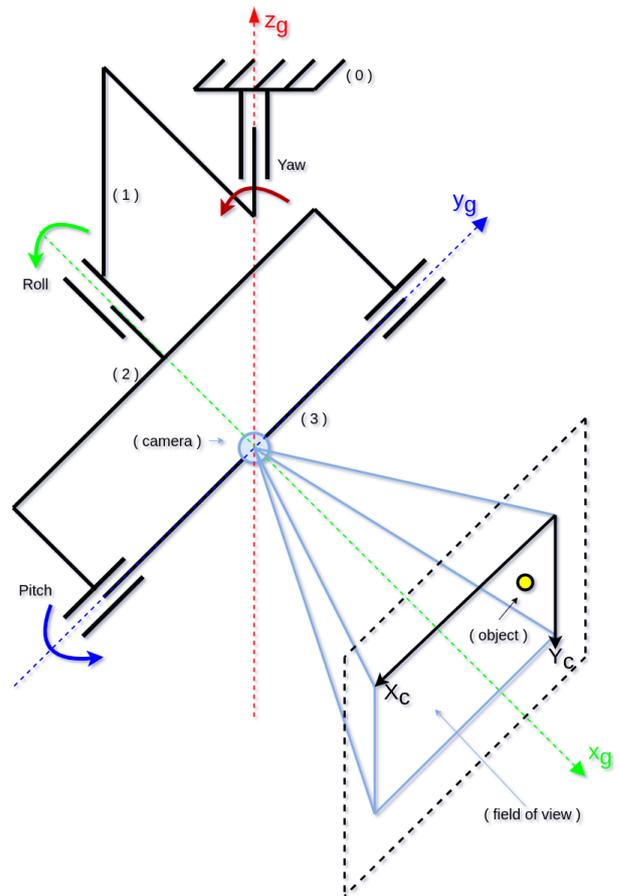


Figure 5. Gimbal mechanism and camera's FoV projected at a x_{fov} distance and the image coordinate.

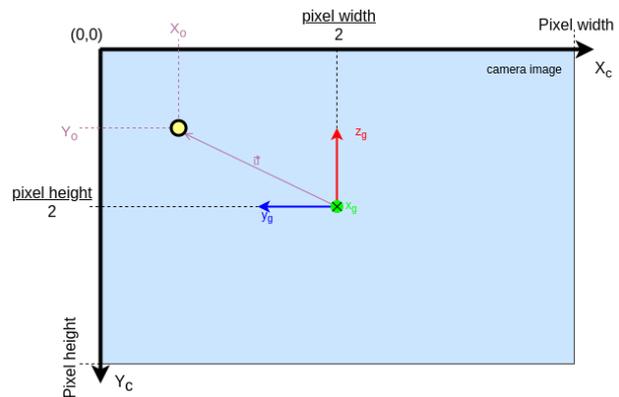


Figure 6. Image coordinate system (C), the gimbal frame (G) projected and detected object (O).

The proposed control was compared to a conventional linear Proportional-Integral (PI) solution. The PI controllers were tuned to work independently to each axis of image's frame. A PI controller, (13), acts on error signal from difference between desire set-point and measured data (Åström and Hägglund, 1995). The errors signals of (13) are (11) and (12).

$$PI(t) = K_c e(t) + \frac{K_c}{T_i} \int_0^t e(t) dt \quad (13)$$

From (13), a discrete variant equation can be described using Z-transformation (Lathi, 2009). The discrete control equation has two parameters to tune performing a reference track and rejecting disturbances with a setting time of one second and 5% overshoot signal. To adjust the PI parameters to achieve the specifications were used a dynamic model from pixel position sensor.

4. MATERIALS

The proposed control were tested in simulation environment using Gazebo Simulator to tune the algorithm and later made practical tests with an aircraft and camera from DJI company, allowing the usage of Software Development Kit (SDK) from DJI (DJI, 2019b). The SDK uses ROS to communicate between sensors and peripherals with a ROS node (Open Source Robotics Foundation, 2010b) when aircraft is initialize. The usage or ROS nodes also guarantee YOLOv3 opens its own ROS nodes to process input images.

The Gazebo environment consists in a CAD model representation of the aircraft Matrice 200 RTK v2 using a *rotors_simulation* package (Furrer et al., 2016) as flight controller. RotorS is a Micro Air Vehicle (MAV) gazebo simulator, it provides some multi-rotor models such as the AscTec (Gurdan et al., 2007) Pelican or Firefly, four and six rotors respectively. The simulator is not limited for the use with these multi-copters, it allows some parameters modification to use on Matrice 200 RTK v2 aircraft. Figure 7 shows the simulation of RPA Matrice 200 RTK v2 and the gimbal mechanism focused on an interest object (e.g. riser).

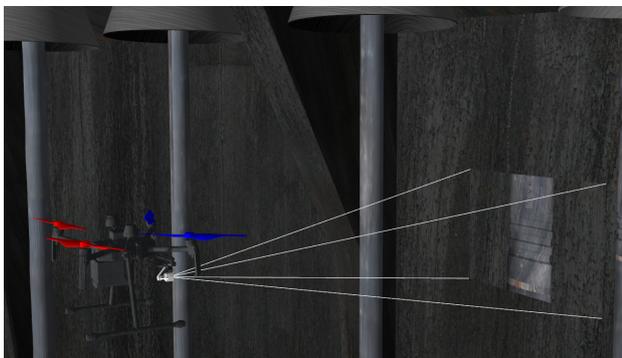


Figure 7. Matrice 200 v2 at Gazebo simulation.

A real experiment was conducted using the DJI Matrice 200 series RTK v2 (DJI, 2019a) with a camera Zenmuse Z30 (DJI, 2019c) attached to a gimbal mechanism, and a Jetson TX2 (NVIDIA, 2019) used to process all data as an onboard computer due to its graphic process unit able to real time process images.

5. EXPERIMENTS AND RESULTS

The Gazebo software environment was used to tests communication between YOLOv3, DJI SDK and control nodes to perform image processing and control gimbal's angles.

Based on experimental results of (Salazar et al., 2020). The YOLOv3 performance evaluation was realized employing a total of 3000 RGB images with 1920x1080 and 4096 x 3000 resolutions. These images were obtained from virtual and real

environments scenarios for riser inspection . They were divided into training and test sets according to the ratio 90:10 (90% training and 10% testing). To avoid overfitting, a simple data augmentation is performed randomly in the training dataset. In that context, the images were pre-processed in terms of brightness, contrast and zoom. Considering the fact that CNNs require a lot of training data before achieve a good performance (training with small dataset affects the generalization of the CNNs) and noticing that the dataset used contains 3000 images, was used the strategy of deep learning, transfer learning. The principal idea of transfer learning is to apply the knowledge learned from certain domain with a large amount of training data to a target domain with insufficient training data. In this case, the backbone network Darknet-53 is pre-trained on the COCO dataset (principal domain) and then the object detection task is transferred into the target domain (the riser detection).

To evaluate the performance of the model we have used the mAP metric. It is calculated by computing the AP (average precision) for different classes and averaging them. AP is a measure that combines recall and precision parameters. They are defined below:

$$Precision(P) = \frac{TP}{TP + FP} \quad (14)$$

$$Recall(R) = \frac{TP}{TP + FN} \quad (15)$$

In these formulas, True positive (TP) indicates the number of correctly detected risers, true negatives (FP) indicate the number the wrong detections, and (FN) indicates the number of missed detections, respectively. Thus, P represents the percentage of right risers detections among all those identified as risers. R refers to the correct rate of detections among all the GT in the dataset. Finally, the AP is approximate to the area under the PR curve. Considering the evaluation metrics, the detector was able to detect risers in the test dataset (300 images) with 99,4% mAP. This high value indicates that when the YOLOv3 algorithm classifies an object as riser, is very highly probable that this object is a riser.

The photogrammetry trajectory performed by the RPAS is a combination of vertical and horizontal displacements (serpentine trajectory) as shown in Figure 8, maintaining constant distance to the riser, keeping spatial resolution constant and camera centered on the object of interest (the riser). (Buschinelli et al., 2020) detail the photogrammetry trajectory used.

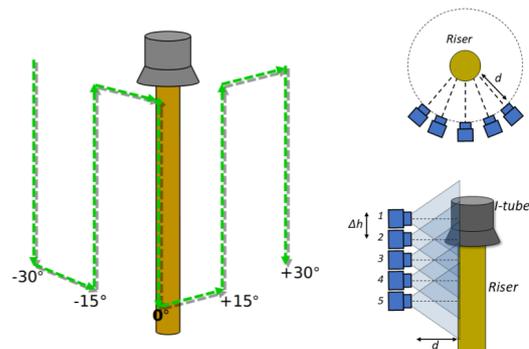


Figure 8. Vertical serpentine with a fixed distance from object.

The Figure 9 illustrates the object detection results using YOLOv3 model on a virtual environment. The riser is detected satisfactory, thus allowing to perform experiments to tracking of risers while RPAS performing the serpentine trajectory. The results of tracking algorithm were able to maintain the object mostly of the time centered at image's resolution. As the riser fills the entire vertical length of the image, the controller practically keeps fixed the value of *pitch* for each vertical step of the path. This value is changed when YOLOv3 detects an anomaly of the object and loses reliability.

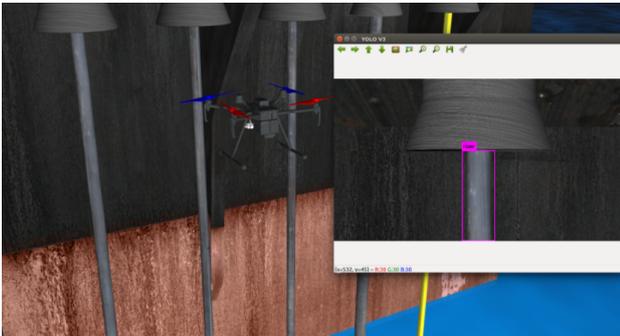


Figure 9. Riser detection at Gazebo simulation using YOLOv3.

The experiments performing on the virtual environment for both control theory strategies had similar responses to maintain the object centered at image resolution, the PI controller rejected slightly faster the disturbances from trajectory acquisition, Figure 10 shows the object pixel position over time to each axis on image resolution.

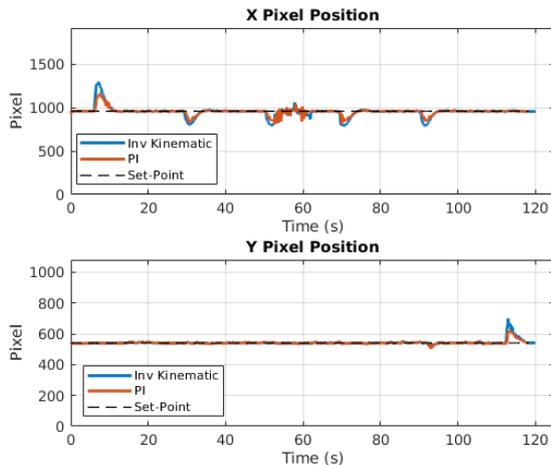


Figure 10. Pixel comparison of different track controllers at Gazebo environment.

Comparing the angle's position of controllers (Figure 11), every moment a vertical serpentine alters the *yaw* angle change to maintain the object centered, performing a fixed distance to riser inspected.

Due the complexity of performing experiments with real risers on an offshore platform (transporting the RPAS is too time consuming and expensive), in this study the different experiments were performing moving the RPAS manually on a reduced vertical serpentine trajectory using a chair representing the target object (the riser). For detecting the chair a YOLOv3 implementation trained on COCO dataset was used. To achieve a

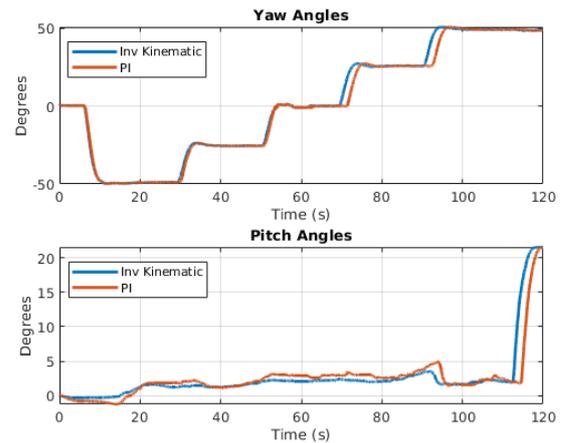


Figure 11. Angle comparison of controllers vertical serpentine trajectory with fixed distance.

good frame rate on Jetson TX2 the input images size was reduced to 192 x 192 pixels. The Figure 12 illustrates how the experiment was carried out.



Figure 12. Practical experiment using YOLOv3 dataset.

To circumvent the small object detection problem, were used a chair to represent the desired object to check if the algorithm will be tracking the object while the RPA is performing its trajectory represented in Figure 13.

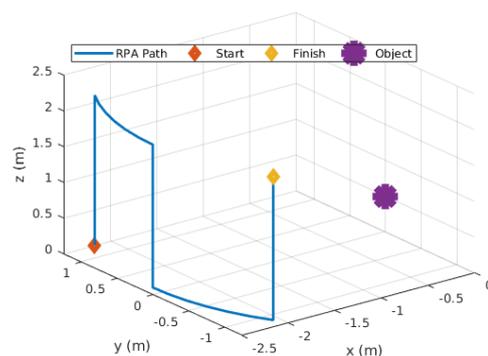


Figure 13. Practical experiment trajectory.

The results of both controllers (Figure 14) were functional to validate algorithm embedded on an onboard computer (Jetson TX2 platform), although it was necessary to reduce the

YOLOv3 input image size to perform the tracking were possible to maintain a real time object tracking using both control approaches. It has some slightly differences between approaches, the non-linear formulas of inverse kinematic demands a strong computational effort, taking more time to reject disturbance of aircraft movement as the PI controller.

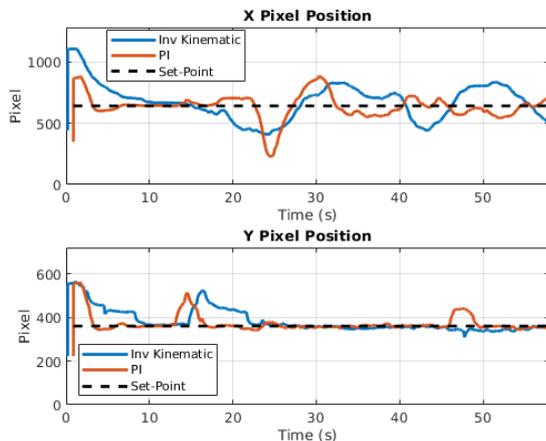


Figure 14. Comparison of pixel position using Inverse Kinematic and PI controllers.

6. CONCLUSIONS AND PROSPECT WORKS

In this paper, we presented the design and implementation of a tracking control algorithm at a gimbal mechanism attached to the DJI M210 RTK v2 RPAS. The state-of-the-art YOLOv3 detection model was used for detecting an object of interest. The CNN-based model detected risers in a scenario that presents different environment conditions with 99.4% mAP in a dataset containing real and virtual images. The bounding box result obtained from the detection process is used to determine the angles of the gimbal to maintain the object centered at the field of view of the camera. The controller proposed used Kinematic matrices to formulate the angle's Inverse Kinematic expression.

Experiments were performed in Gazebo environment to tune and validate the proposed controller from rotation matrices with similar response as the linear controlled using PI. The algorithm was able to detect risers in simulation environment and keep on tracking object while the aircraft was performing its specialized trajectory. Results of different controllers approaches were similar with a faster disturbance rejects of linear controller due to its computational effort.

Practical laboratory experiments were also conducted to validate the algorithm outside the simulation environment. For that, a reduced vertical serpentine trajectory was performed manually with the RPAS and the target object was a chair representing the riser. Also using the onboard computer (Jetson TX2 platform) to process data from sensors of aircraft position and gimbal's mechanism.

The algorithm achieves the goal of maintaining the target object at the field of view of the camera while the RPAS perform the serpentine trajectory, rejecting disturbances from aircraft's movement remotely piloted, it is an initial step to automate specialized inspection processes based in photogrammetric techniques.

Future works includes the optimization of computational effort to process YOLOv3 on an onboard computer, utilize the track theory when RPAS is flying, and posterior test on an offshore riser inspection.

REFERENCES

- Altan, A., Hacıoğlu, R., 2020. Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances. *Mechanical Systems and Signal Processing*, 138.
- Åström, K. J., Hägglund, T., 1995. PID controllers: theory, design, and tuning.
- Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., Ouni, K., 2018. Car detection using unmanned aerial vehicles: Comparison between faster R-CNN and YOLOv3. *arXiv*, 1–6.
- Buschinelli, P., Salazar, J. D., Regner, D., Oliveira, D., Machado, M., Marcellino, G., Sales, D. C., Santos, J. M., Marinho, C. A., Stemmer, M. R., Pinto, T. C., 2020. Targetless Photogrammetry Network Simulation for Inspection Planning in oil and gas industry. *ISPRS 2020*, Nice, France, 1–7.
- Cunha, R., Malaca, M., Sampaio, V., Guerreiro, B., Nousi, P., Mademlis, I., Tefas, A., Pitas, I., 2019. Gimbal control for vision-based target tracking. *Workshop on Signal Processing Computer vision and Deep Learning for Autonomous Systems, EUSIPCO2019*, IEEE Signal Processing Society (SPS). EUSIPCO 2019 Satellite Workshop on Signal Processing, Computer Vision and Deep Learning for Autonomous Systems ; Conference date: 06-09-2019 Through 06-09-2019.
- DJI, 2019a. Matrice 200 series v2. <https://www.dji.com/matrice-200-series-v2/info>. Accessed: 16/08/2020.
- DJI, 2019b. OnBoard SDK Documentation. <https://developer.dji.com/onboard-sdk/documentation>. Accessed: 16/04/2020.
- DJI, 2019c. Zenmuse Z30. <https://www.dji.com/zenmuse-z30/info>. Accessed: 16/08/2020.
- Furrer, F., Burri, M., Achtelik, M., Siegwart, R., 2016. *Robot Operating System (ROS): The Complete Reference (Volume 1)*. Springer International Publishing, Cham, chapter RotorS—A Modular Gazebo MAV Simulator Framework, 595–625.
- Greatwood, C., Bose, L., Richardson, T., Mayol-Cuevas, W., Chen, J., Carey, S. J., Dudek, P., 2017. Tracking control of a UAV with a parallel visual processor. *IEEE International Conference on Intelligent Robots and Systems*, 2017-September, 4248–4254.
- Gurdan, D., Stumpf, J., Doth, K.-M., Achtelik, M., 2007. Ascending Technologies. www.asctec.de/. Accessed: 2020-05-05.
- Jakobsen, O. C., Johnson, E. N., 2005. Control architecture for a UAV-mounted pan/tilt/roll camera gimbal. *Collect. Tech. Pap. - InfoTech Aerosp. Adv. Contemp. Aerosp. Technol. Their Integr.*, 4(September 2005), 2170–2179.
- Jiao, Z., Zhang, Y., Mu, L., Xin, J., Jiao, S., Liu, H., Liu, D., 2020. A YOLOv3-based Learning Strategy for Real-time UAV-based Forest Fire Detection. *Proceedings of the 32nd Chinese Control and Decision Conference, CCDC 2020*, 4963–4967.

- Joseph, L., 2018. *Robot Operating System (ROS) for Absolute Beginners*. 1 edn, Apress, Berkeley, CA.
- Kendall, A. G., Salvapantula, N. N., Stol, K. A., 2014. On-board object tracking control of a quadcopter with monocular vision. *2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014 - Conference Proceedings*, 404–411.
- Kulkarni, S., Mohanty, A., 2013. Analysis of the Gimbaled Platform for the Three Degrees of Freedom Using Differential Equations. *International Conference on Advanced Research In Mechanical Engineering ARME*, 1(May), 31–34.
- Lathi, B. P., 2009. *Linear Systems and Signals*. 2nd edn, Oxford University Press, Inc., USA.
- Liang, S., Li, X., Wang, J., 2012. *Advanced Remote Sensing*. 1 edn, Academic Press.
- Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*.
- Luhmann, T., Robson, S., Kyle, S., Boehm, J., 2014. *Close-Range Photogrammetry and 3D Imaging*. 2 edn, De Gruyter, Germany.
- McGee, J., Mathew, S. J., Gonzalez, F., 2020. Unmanned Aerial Vehicle and Artificial Intelligence for Thermal Target Detection in Search and Rescue Applications. *2020 International Conference on Unmanned Aircraft Systems, ICUAS 2020*, 883–891.
- NVIDIA, 2019. Jetson TX2 Developer kit. <https://developer.nvidia.com/embedded/jetson-tx2-developer-kit>. Accessed: 16/10/2020.
- Open Source Robotics Foundation, 2010a. Wiki ROS. <http://wiki.ros.org>.
- Open Source Robotics Foundation, 2010b. Wiki ROS. <http://wiki.ros.org/Nodes>.
- Open Source Robotics Foundation, 2014. Gazebo Simulator. <http://gazebo.org/>. Accessed: 2020-05-01.
- Quigley, M., Gerkey, B., Smart, W. D., 2015. *Programming Robots with ROS A Practical Introduction to the Robot Operating System*. 53, O'Reilly Media Inc.
- Rajesh, R. J., Kavitha, P., 2016. Camera gimbal stabilization using conventional PID controller and evolutionary algorithms. *IEEE Int. Conf. Comput. Commun. Control. IC4 2015*.
- Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2016-Decem, 779–788.
- Redmon, J., Farhadi, A., 2017. YOLO9000: Better, faster, stronger. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*.
- Redmon, J., Farhadi, A., 2018. YOLOv3: An Incremental Improvement. *arXiv*.
- Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6), 1137–1149.
- Salazar, J. D., Buschinelli, P., Marcellino, G., Machado, M., Rodrigues, H., Regner, D., Oliveira, D., Santos, J. M., Marinho, C. A., Stemmer, M. R., Pinto, T. C., 2020. 3D photogrammetric inspection of risers using RPAs and deep learning in oil and gas offshore platforms. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2, 1265–1272. www.isprs2020-nice.com <https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLIII-B2-2020/1265/2020/>.
- Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G., 2009. *Robotics: Modeling, Planning, and Control*. 16, Springer.
- Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. A., 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. *AAAI*, 4, 12.
- Tiimus, K., Tamre, M., 2010. Camera gimbal control system for unmanned platforms. *Proc. Int. Conf. DAAAM Balt.*, 202–207.
- Wang, C., Shankar, K., Morozov, E. V., 2016. Tailored design of top-tensioned composite risers for deep-water applications using three different approaches. *Advances in Mechanical Engineering*, 9(1), 1–18.
- Yoonseok Pyo, Hancheol Cho, L. J., 2017. *ROS Robot Programming*. 1 edn, ROBOTIS Co.,Ltd.
- Yuan, C., Liu, Z., Zhang, Y., 2015. UAV-based forest fire detection and tracking using image processing techniques. *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, 639–643.