Autonomous robotic inspection system for drill holes tilt: feasibility and development by advanced simulation and real testing

Nour M. Morsi*[1], Mario Mata[2], Colin S. Harrison[1], and David Semple[1]
[1]Mechanical Engineering Department, [2]Computing Department,
Glasgow Caledonian University, Glasgow, Scotland
{*Nour.Morsi, Mario.Mata, Colin.Harrison, David.Semple}@gcu.ac.uk

2023 28th International Conference on Automation and Computing (ICAC) | 979-8-3503-3585-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICAC57885.2023.10275276

***Abstract*—Industry 4.0 and forthcoming 5.0 introduce process methods that primarily rely on robotic systems to succeed. Robotic systems provide accuracy in inspection tasks over current methods. However, in the aerospace industry the required accuracy is beyond what a standard robotic arm can provide. This paper aims to introduce a simple finite state-controlled robotic system using image processing to guide the motion and evaluate the tilt of a series of drill holes using one of them as reference hole. The system is developed via simulation in RoboDK and then validated in both a simulated and a real setup, using an ABB GoFa robot. The inspection experiment is performed by aligning the camera optical axis to the first drill hole axis. By taking it as a reference, the rest of the holes are measured and compared with the reference one, and to the ground-truth results. Drill hole detection is refined using a proposed circle optimisation algorithm. Results demonstrate the validity of the approach with an error below 0.5 degrees.**

***Index Terms*—Industrial robotics,drill tilt, finite-state machine, inspection system, machine vision, RoboDK.**

I. Introduction

Industry 4.0 is the present revolution in the manufacturing industry. Industry 4.0 relies on digitizing products and services offered [1]. With the introduction of the internet of things, cyber-physical systems, cloud computing, and digital twin, industry 4.0 aims to introduce smart factories [3], [4]. Most importantly, industry 4.0 has increased the uptake of robotic systems based on the mission of competitiveness [5]. With rapid developments in technology, simulation software used to evaluate a robotic system has reached adequate potential. This enables a company to examine the potential of a robotic system before attempting a major change in the factory line [6], [7]. In a standard inspection process, a robotic system is an accurate, cost and time effective approach than conventional processes [8]. Robotic systems have been increasingly present in various fields for inspection processes in civil [9] , aerospace [10] and pipelines [11]. The types of inspections range from crack detection to hole alignment [12], and weld quality check [13]. Aerospace industry requires stringent accuracy standard of 0.5° in perpendicular accuracy [14] as the deviation impacts the interlayer gap between components, which further degrades the assembly accuracy. The accuracy of a robotic arm is not up to standard with the accuracy required by the aerospace industry [15] [16], unless extra sensors (camera, laser) are combined with the robot to increase the overall robotic system's accuracy. Therefore, this paper focuses on tackling the lack of automation in the aerospace industry by proposing a method to check for drill hole tilts in a series of related drills, very common in aerospace structures. We introduce a simple finite state-controlled autonomous robotic inspection system with computer vision feedback and test it on a ground-truth aluminium block with known drill tilt deviations. It is initially developed and tested in RoboDK simulation package (using Python scripts) before implementation on an ABB GoFa robot [17]. The approach proposed uses visual feedback to achieve closed-loop autonomous motion, avoiding the need for robotic fixed positions to inspect parts, it is enough with leaving the part to inspect within the robot‘s camera field of view. The control code is designed in a modular way to allow easy adaptation to other similar applications in the future. The paper is arranged as follows. Section 2 reviews related works on inspection approaches for drilled holes. Section 3 introduces the simulated and experimental setup. Section 4 describes the proposed approach, and Section 5 will discuss the results. Finally, conclusions are summarized in Section 6.

II. Related Work

A. Autonomous robotic inspection system

In the aerospace industry, drilling fastener holes and mounting holes is a non-trivial process due to the large drill quantity and the type of materials holes must be drilled through [10], [18]. Inaccuracies in the process have introduced non added value operations (shimming, gap checks) [19]. Therefore, several papers have introduced methods to achieve the precision standards needed when drilling the holes, by ensuring the alignment of drill tool with respect to the workpiece. Sitton and Feikert and Von introduced a ONCE robotic drilling system that automates the hole inspection of a clamped wing trailing edge flaps of a Super hornet by using a vision system that measures the theoretical and the actual position of the hole, achieving a 1.5mm positional deviation [20]. Using four inductive sensors, a perpendicularity and deviation measurement was proposed by Furtado achieving the accuracy required [16]. Weidong and Biao introduced a robotic that utilizes four laser displacement sensors along with a camera as visual feedback that ensures a 0.1 mm accuracy of the holes

[10] . Gong and Yuan also proposed three range laser sensors that help calculate the normal vector at the drilling vector. The method proposes an accuracy of 0.02 degree [21]. Carlos, Emilia , Luís, and Guilherme utilised four inductive sensors on a preclamped structure to achieve the tolerance range of ±0.5° [22]. However, in many occasions the drills are made relying on the drilling technology available and the skill of the workers. If the tilt of the holes is not checked before assembling it in place, incorrect drill tilts might prevent the assembly process. Therefore, we implement a single camera-based robotic drill inspection system that relies on taking one out of a set of related holes as a reference, then measuring the relative tilt of subsequent holes. The approach assumes that the drilling process is precise enough to produce correct drills under normal circumstances and that deviations will not be systematic but due to small operator errors due to fatigue or distractions. Then if all holes in the related set are correctly aligned to each other (within 0.5 degrees tilt tolerance), the part will be fit for use. If any of the holes shows an out-of-range tilt, the part is rejected and possibly fixed (or discarded). Of course, even if all holes are perfectly aligned between them, there's still the chance of all of them being non-perpendicular to the part surface with the same deviation, but this would be caused by a systematic error that should not occur in the first place.

B. Finite State Machine based design

For the system proposed, a combination of a general programming language (Python and robot-level language programming is chosen. Python code is run in RoboDK and used at top-level to drive the operation of both the simulated robot and the real robot, connected to the robotic system via Ethernet. We are using a Finite State Machine (FSM) based design to keep the application flow clean and modular, making program testing/debugging easier. The FSM-based application design also eases adaptation and expansion to other future applications [23]. Finite state machines are behavioural models that include distinct states that simulate an envisioned sequential logic for a managed execution mechanism [24]. The machine system remains in one of the collected states until the state changes in response to an input or a control condition. Each state is used to enclose a behavioral step in the application, for instance ”center the object in the image”, ”analyse drill tilt”, etc.

C. Simulation

Simulation programs offer the means to verify and run theoretical simulations prebuilding the robotic assembly [25], offering various features to recreate the natural state of the system, such as work cycle counter and collision detection [26]. RoboDK is used in this work, offering strategies for simulating, and analyzing the robotic configuration [27]. Developing a robotic application this way increases the chances of colliding the robot, therefore testing on the simulator before connecting to the real robot is a must to verify that the proposed system does not produce dangerous motions during development.

D. Image processing

Image processing aids in identifying application targets and extracting relevant metrics for inspection tasks. Visual feedback avoids depending on fixed, pre-programmed positions; it can be enough with just leaving the parts to work on within robot's reach. In our application, drill holes produce characteristic circular shapes that can be exploited for their detection and analysis. There are many circle detection algorithms available [28]– [30]. We have selected to use the Circle Hough Transform (CHT) due to its simplicity, good performance in noisy images, and ability to detect multiple circles –suitable to analyse groups of drill holes related to each other. A Hough Transform uses shape analysis as a constraint equation to relate the points in feature space to parameter values of the examined shape. The Hough transform performs the detection in two steps: an incremental transform mapping and a voting rule are applied, and a search in the accumulator array for the specified parameters with a potential shape [32]. As the number of parameters for detection increases, the required memory space and computational time grow exponentially; for a circle characterization, a 3D accumulator array is utilized to find suitable center (a, b) and radius (r) values. The resolution used in this accumulator needs to be limited to allow for real-time operation. We are using an accumulator step of 1 pixel for radius and centre coordinates, which is good enough to detect the drill holes and move closer to them. However, for drill tilt analysis, 1 pixel resolution limits the overall performance as discussed in section 4.1.6. In this analysis step, we achieve sub-pixel resolution in the detection of the hole circles (front and back) using an optimised circle detection algorithm.

1) Circle detection with sub-pixel accuracy: Due to limited resolution and susceptibility to noise in complex backgrounds, CHT is only used for initial scanning and to guide the robot motion to center the drill holes in the image, but for the drill tilt analysis a refinement of the detected circles is needed. A proposed circle optimisation algorithm searches the whole image for groups of edge pixels that could be part of a circle, then proves / discards this hypothesis by considering all edge pixels in a "ring" around the initial guess. In our case, CHT already provides a good estimation of the circle centre and radius, just with limited resolution and usually slightly off-centered due to noise near the circular edge. To achieve sub-pixel accuracy, we adapt an optimized detection approach as follows:

- Detect edges using Canny edge detection [33].
- Create a sub-image around the candidate circle detected by CHT (a few pixels bigger than it) (Fig. 1(a)).
- A ring-shaped mask is created around the initial radius value obtained from CHT (Fig. 1(b)).
- A bitwise AND operation between the edge sub-image and the mask is applied. Surviving pixels are possible circle edges (Fig. 1(c)). This step replaces the probabilistic hypothesis testing in a conventional RANSAC approach.

- Using the CHT center and radius as initial parameters, an iterative Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [34] is used to calculate the optimal centre and radius fitting a circle to the ring of edge pixels. BFGS is best known for promoting a global best solution for swarm optimization [35]- [37]. The algorithm seeks to find the values for x_o,y_o and r that minimize (1) ,where x_o,y_o represent the sum of all the edge points.

$$f(x) = \sum[(x_o - x_c)^2 + (y_o - y_c)^2 - r^2]^2 \qquad (1)$$

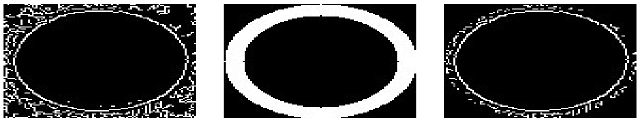
(a) (b) (c)

Fig. 1. Canny edge, ring mask and mask effect.

III. PROPOSED ROBOTIC SETUP

The system comprises an ABB GoFa (CRB 15000) robot and a DFKCU03 ALRAD camera with a 12mm focal length optics. A front lighting and back lighting system are used to ensure a correct illumination and to selectively highlight the frontal and rear circles of the drill hole. This illumination system is not part of the simulator as it will produce neat frontal and rear drill circles in the simulated images; as mentioned earlier, the simulation capabilities are quite limited with respect to the virtual camera. Finally, a rectangular block with multiple drills with known misalignments (0 to 9 degrees) is inspected and deployed both in the simulator and real setup (Fig. 2). The only predefined parameters that the application needs are the following:

- "Home" position to provide a camera field of view covering a large portion of the part to inspect.
- Initial target circles search radius (60-80 pixels as expected by the focal and home position). This parameter can be relaxed if unknown at the cost of a more time-consuming initial search.
- The minimum radius value for drill analysis = 220 pixels. This is selected empirically to ensure a large drill hole appears in the image (which reduces tilt estimation error as discussed in section 4.1) while keeping the hole within the camera focus depth.

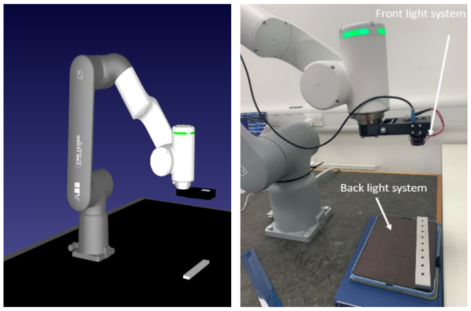

Fig. 2. Simulated setup (left) and real setup (right).

Motion control is implemented using a PID control approach from the visual feedback, therefore the part to inspect can be placed anywhere within the camera field of view from the home position (and in any orientation), avoiding the need of using calibrated fixed positions.

IV. PROPOSED SOFTWARE APPROACH

A. System States

RoboDK is used to run a Python script containing the control program, designed based on a FSM machine approach. This approach allows states to be easily modified, removed, or added to adapt to other similar inspection applications. Execution is initialized in the Home position state, and then proceeds as proposed by the flow diagram shown in Fig. 3 that depicts relevant transitions between the states in the FSM.

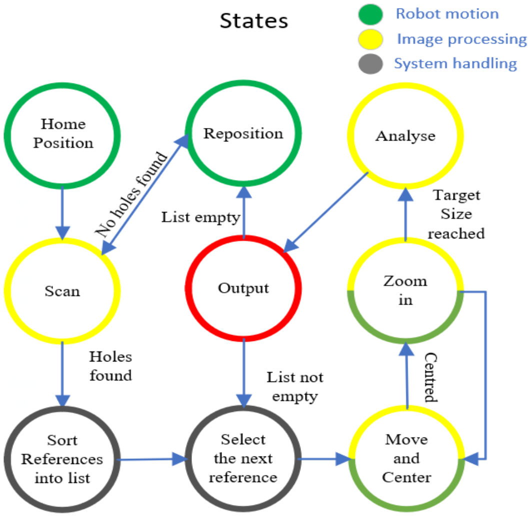

Fig. 3. Proposed flow diagram.

B. Intial search (Scan)

The camera takes an image, and using the CHT, drill holes in the image are found (Fig. 4). If no drills are detected, the robot translates by a fixed amount from its home position and detection is repeated until drills are found, or the whole work area is covered without success.

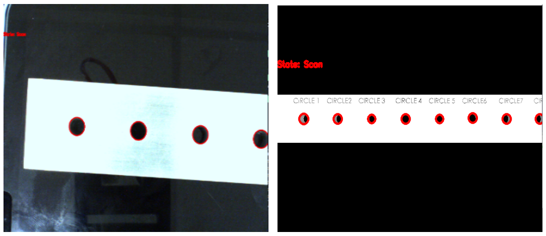

Fig. 4. Scan function in real system (left) and RoboDK simulated setup (right).

C. Move and center

Once a drill hole is selected the system switches to a new state with the purpose of centering the hole in the image; this step is combined with the next step (zooming into the drill hole) until the drill to be inspected appears in the centre of the image and with a large enough apparent radius. Robot motions and image processing are alternated in a closed loop using Algorithm 1. Firstly, an image is taken, and the circle is detected again. The difference between the x and y coordinates of the circle's center and the image center (Cx and Cy) is calculated, and the robot moves proportionally to these differences. A full camera calibration is not required, but the correspondence between image rows/columns and robot base frame is used to convert differences in image pixels to motion on robot axes. The loop is repeated until the camera's optical axis is directly above the circle's center.

Algorithm 1 Move and Center

```
 Input: selected hole x,y,r

if $State : MoveandCenter$ then
   $Takeimage$
   $Detectcenter(x, y)usingCHT$
else
   $xerror = C_x - x$
   $yerror = C_y - y$
   if $(if\ xerror \leq 2\ or\ yerror \leq 2)$ then
      State: Zoom in
   else
      xdelta= xerror *0.2
      ydelta= yerror*0.2
   end if
end if
   Output:robot moves by xdelta,ydelta
```

D. Zoom in

At this point, the robot uses visually directed closed-loop motion to approach the drill. When the circle reaches the desired size, the robot comes to a stop. For the camera's optical axis to remain aligned with the drill's centre, a move-and-center state is additionally carried out after each zoom-in motion.

E. Drill tilt analysis

Once the drill hole is centered in the image and large enough, the sub-pixel circle detection described in section II-D1 is applied. A new image is taken with the backlight on, and the circle in the back of the drill is then detected and refined. The differences between the centres of the front and back circles from a drill hole can then be used to estimate the drill's tilt. An expression to estimate the tilt of the drill is derived using the pinhole camera model Fig. 5. The figure shows a traversal section of the part to inspect at a drill's centre. The relationship between the depth of the part from the camera focal point (S), the drill radius (R), the radius of the front circle in the image plane (r_c), and the focal length is given by (2):

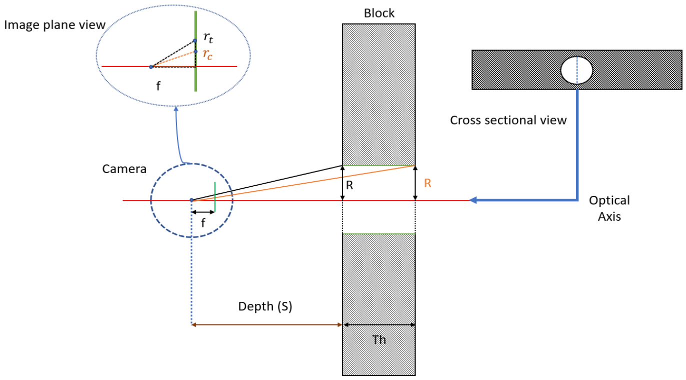

Fig. 5. Pin hole model.

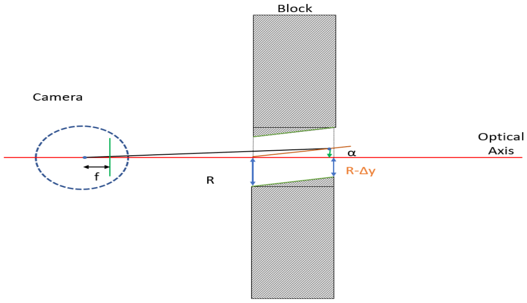

Fig. 6. Tilt drill model.

$$\frac{R}{S} = \frac{r_t}{f} \tag{2}$$

Applying the same approach to the back circle (r_c), with a drill length equal to the thickness of the material (Th):

$$\frac{R}{S + Th} = \frac{r_c}{f} \tag{3}$$

Solving for f in (2) and (3) and combining them:

$$S = Th * \frac{r_c}{r_t - r_c} \tag{4}$$

For a tilted drill (Fig. 6), the relationship between the difference in drill centers in pixels (y_c-y_t), focal length, depth, thickness, and the difference between the drill centers (Δ y) in mm, is given by (5). To convert the focal length to pixels, it is multiplied by the camera's physical resolution (K, in pixels/mm):

$$\frac{y_c - y_t}{f * K} = \frac{\Delta y}{S + Th} \tag{5}$$

Also, from Fig. 6, the angle α of drill tilt on the y-axis is given by:

$$\alpha = \tan^{-1}(\frac{\Delta y}{Th}) \tag{6}$$

Solving for the thickness in (5) and replacing it in (6) leads to:

$$\alpha = \tan^{-1}(\frac{(y_c - y_t)}{f * K} * \frac{r_t}{(r_t - r_c)}) \tag{7}$$

- r_t: front circle radius (pixels)
- r_c : back circle radius (pixels)
- Δy : difference between circle centers (mm)
- y_t : front circle coordinate (pixels)
- y_c : back circle coordinate (pixels)
- f : equivalent focal length (mm)
- K: physical resolution (pixels/mm)

Notice that, thanks to the particular geometry happening when the centre of the hole is placed on the image axis, the drill radius and part thickness are no longer intervening, replaced by the difference in front and back circle radiuses. An equivalent equation is obtained for the tilt along the rows (y) direction. Equation (7) allows for the system to function without the need for camera calibration, although the effective focal length (fx, fy) will deviate from the nominal value if the camera focus has been brought closer in. In that case, a standard camera calibration can be used to determine the new focal length [38] Finally, a result image showing the measured tilts in degrees is produced (Fig. 7). if the system has detected the first drill, it reorients the camera until the back and front circle of the hole align with the optical axis so it becomes the reference hole used to measure the tilt of subsequent holes

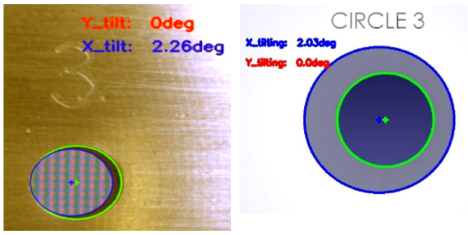

Fig. 7. Image processing results on real (left) and simulated (right) drills.

F. Restart

After the analysis, the system returns to the home position it repeats the sequence to scan all of the remaining references yet to be analysed . If there are no more circles, the system returns to the home position and then translates to a new position further along the element to scan. The new position is set to be the new home position.

V. Results and discussion

The system has successfully detected all the drilled circles on the block in the simulation and the real experiment, the results are shown in Table I. When the system is tested in real conditions, additional issues appear as the block produced light reflections on the surface together with dark background through the hole (that the simulated images were not replicating correctly), which complicates the detection of the circles used to estimate the drill tilt. This was mitigated by introducing diffuse frontal lighting when detecting the top circle for robot guidance; once the robot is positioned to start the tilting analysis, one image is taken with frontal lightning followed by another image with backlighting instead to highlight the back circle (both images are taken in quick succession without moving the robot between them). Both front and back drill circle detections are then combined to perform the same tilt calculations, as shown in Fig.7, left. It can be noted that the system can't measure tilts greater than 7 degrees in the real images, as the radius does not allow the drill's back circle to fully appear in the image; in any case, this large tilt is unlikely to appear in practice unless intentional. The values measured are nearly accurate to the true results but present some error (less than 0.5 degree). To further understand the sources of this error, an error propagation analysis based on partial derivatives is performed on (7).

TABLE I
Simulation and experimental results versus ground truth.

Drill reference number	Ground truth (degrees)	Simulation tilt (degrees)	Experimental tilt (degrees)
1	0	0	0
2	1	0.72	1.50
3	2	2.08	2.26
4	3	2.84	2.79
5	4	3.97	3.65
6	5	4.8	4.88
7	6	6.05	5.54
8	7	6.71	7.21
9	8	7.41	11

$$\frac{d\alpha}{dy_t} = \frac{f * K * r_t(r_c - r_t)}{{r_t}^2 * (y_t - y_c)^2 + f^2 * K^2 * (r_c - r_t)^2} \tag{8}$$

$$\frac{d\alpha}{dr_t} = \frac{f * K * (y_t - y_c) * r_c}{f^2 * K^2 * (r_t - r_c)^2 + (y_t - y_c)^2 * {r_t}^2} \tag{9}$$

In terms of experimental control, the focal length and the physical resolution are assumed to contribute with no error as no action can be taken to improve them except for the camera calibration to obtain the practical focal length (K*f). The radiuses (r_t and r_c) and the center coordinates are subject to the error from the circle detection; in CHT it is largely dependent on the number of points and the distribution of the points used in the calculation [39]. The modified circle optimization algorithm improves the circle centre and radius determination to an average of 0.4 pixel error, this is validated by creating synthetic images of circles in matlab and testing it on our modified algorithm . Also, (8) and (9) show that the larger the apparent radiuses r_t and r_c, the smaller the error; this justifies the need for zooming into the circle as much as possible to increase the accuracy.

VI. Conclusion

In this paper, we proposed a finite state machine-based autonomous system to perform an inspection in a simulated environment; once simulation results were adequate the system was tested on a real setup. The system integrates robot control and object detection to complete the assigned task (inspection of drill alignment) with success. The precision obtained (below 0.5 degree) is analysed to find future improvements. The inspection application can be adapted to other similar inspection problems as well by replacing the image processing algorithms used to feedback positional error to the robot, and an adequate final analysis algorithm. The finite states introduced should allow easy repurposing for new applications.

References

[1] Poor, Peter & Tomas, Broum & Basl, Josef. (2019). Role of Collaborative Robots in Industry 4.0 with Target on Education in Industrial Engineering. 42-46. 10.1109/CRC.2019.00018.

[2] R. Horatiu, A. C. Trasculescu, I. Cristian Resceanu, L. C. Bazavan, R. Dumitru Antohi and N. G. Bizdoaca, ”Collaborative Application Between Robots for Industrial Environments,” 2021 International Conference on Applied and Theoretical Electricity (ICATE), Craiova, Romania, 2021, pp. 1-4, doi: 10.1109/ICATE49685.2021.9464979.

[3] F. Sherwani, M. M. Asad and B. S. K. K. Ibrahim, ”Collaborative Robots and Industrial Revolution 4.0 (IR 4.0),” 2020 International Conference on Emerging Trends in Smart Technologies (ICETST), Karachi, Pakistan, 2020, pp. 1-5, doi: 10.1109/ICETST49965.2020.9080724.

[4] J. Wan, M. sXia, J. Hong, Z. Pang, B. Jayaraman and F. Shen, ”IEEE Access Special Section Editorial: Key Technologies for Smart Factory of Industry 4.0,” in IEEE Access, vol. 7, pp. 17969-17974, 2019, doi: 10.1109/ACCESS.2019.2895516.

[5] M. Javaid, A. Haleem, R.P. Singh, and R. Suman, ”Substantial capabilities of robotics in enhancing industry 4.0 implementation,” Cognitive Robotics, vol. 1, pp. 58-75, 2021.

[6] M. Dragic and M. Sorak, ”Simulation for improving the performance of small and medium sized enterprises,” International Journal of Simulation Modelling, vol. 15, no. 4, pp. 597-610, 2016.

[7] C. S. Timperley, A. Afzal, D. S. Katz, J. M. Hernandez and C. Le Goues, ”Crashing Simulated Planes is Cheap: Can Simulation Detect Robotics Bugs Early?,” 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), Västerås, Sweden, 2018, pp. 331-342, doi: 10.1109/ICST.2018.00040.

[8] R. Almadhoun, T. Taha, L. Seneviratne, J. Dias, and G. Cai, ”A survey on inspecting structures using robotic systems,” Int. J. Adv. Robot. Syst., vol. 13, no. 6, pp. 1-14, 2016, doi: 10.1177/1729881416663664.

[9] H. M. La, N. Gucunski, K. Dana, and S.-H. Kee, ”Development of an autonomous bridge deck inspection robotic system,” J. Field Robotics, vol. 34, pp. 1489-1504, 2017, doi: 10.1002/rob.21725.

[10] W. Zhu, B. Mei, G. Yan, and Y. Ke, ”Measurement error analysis and accuracy enhancement of 2D vision system for robotic drilling,” Robot. Comput.-Integr. Manuf., vol. 30, no. 2, pp. 160-171, 2014, doi: 10.1016/j.rcim.2013.09.014.

[11] J. Costa et al., ”Autonomous Robotic System for Pipeline Integrity Inspection,” in Digital Human Modeling. Applications in Health, Safety, Ergonomics, and Risk Management: Health and Safety. DHM 2017, V. Duffy, Ed. Springer, Cham, vol. 10287, 2017, doi: 10.1007/978-3-319-58466-9_30.

[12] T. Olsson, M. Haage, H. Kihlman, R. Johansson, K. Nilsson, A. Robertsson, M. Björkman, R. Isaksson, G. Ossbahr, and T. Brogårdh, ”Cost-efficient drilling using industrial robots with high-bandwidth force feedback,” Robot. Comput.-Integr. Manuf., vol. 26, no. 1, pp. 24-38, 2010, doi: 10.1016/j.rcim.2009.01.002.

[13] Y. Zhang, Z. Dai, Y. Xu, and R. Qian, ”Design and adsorption force optimization analysis of TOFD-based weld inspection robot,” in Journal of Physics: Conference Series, vol. 1303, no. 1, p. 012022, Aug. 2019, doi: 10.1088/1742-6596/1303/1/012022.

[14] Y. Gao, D. Wu, Y. Dong, et al., ”The method of aiming towards the normal direction for robotic drilling,” Int. J. Precis. Eng. Manuf., vol. 18, no. 6, pp. 787-794, 2017, doi: 10.1007/s12541-017-0094-4.

[15] M. Summers, ”Robot Capability Test and Development of Industrial Robot Positioning System for the Aerospace Industry,” SAE Technical Paper 2005-01-3336, 2005, doi: 10.4271/2005-01-3336.

[16] L. F. Furtado, E. Villani, and R. Sutério, ”A perpendicularity measurement system for industrial robots,” in Proc. of the 20th International Congress of Mechanical Engineering (COBEM), 2009.

[17] A. Garbev and A. Atanassov, ”Comparative Analysis of RoboDK and Robot Operating System for Solving Diagnostics Tasks in Off-Line Programming,” 2020 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2020, pp. 1-5, doi: 10.1109/ICAI50593.2020.9311332.

[18] M. Summers, ”Robot Capability Test and Development of Industrial Robot Positioning System for the Aerospace Industry,” SAE Technical Paper 2005-01-3336, 2005, doi: 10.4271/2005-01-3336.

[19] A. Cirillo, P. Cirillo, G. De Maria, A. Marino, C. Natale, and S. Pirozzi, ”Optimal custom design of both symmetric and unsymmetrical hexapod robots for aeronautics applications,” Robotics and Computer-Integrated Manufacturing, vol. 44, pp. 1-16, 2017, ISSN: 0736-5845, doi: 10.1016/j.rcim.2016.06.002.

[20] R. DeVlieg, K. Sitton, E. Feikert, and J. Inman, ”ONCE (ONe-sided Cell End effector) Robotic Drilling System,” SAE Technical Paper 2002-01-2626, 2002, doi: 10.4271/2002-01-2626.

[21] Y. Gao, D. Wu, Y. Dong, X. Ma, and K. Chen, ”The method of aiming towards the normal direction for robotic drilling,” Int. J. Precis. Eng. Manuf., vol. 18, pp. 787-794, 2017.

[22] C. C. Eguti, L. G. Trabasso, E. Villani, G. K. Coracini, and L. F. F. Furtado, ”Development of a robotic end-effector of drilling and fasteners inserter for aircraft structures,” in SAE Technical Paper, 2012, no. 2012-01-1858.

[23] H. Hu, J. Chen, H. Liu, et al., ”Natural Language-Based Automatic Programming for Industrial Robots,” J Grid Computing, vol. 20, no. 26, 2022, doi: 10.1007/s10723-022-09618-x.

[24] A. Grami, ”Finite-State Machines,” Discrete Math, pp. 373-388, 2023, doi: 10.1016/B978-0-12-820656-0.00020-4.

[25] A. Afzal, D. S. Katz, C. Le Goues, and C. S. Timperley, ”Simulation for Robotics Test Automation: Developer Perspectives,” in 2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST), Porto de Galinhas, Brazil, 2021, pp. 263-274, doi: 10.1109/ICST49551.2021.00036.

[26] B. Jakubiec, ”Application of simulation models for programming of robots,” in SOCIETY. INTEGRATION. EDUCATION. Proceedings of the International Scientific Conference, vol. 5, pp. 283-292, May 2018.

[27] A. Garbev and A. Atanassov, ”Comparative Analysis of RoboDK and Robot Operating System for Solving Diagnostics Tasks in Off-Line Programming,” 2020 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2020, pp. 1-5, doi: 10.1109/ICAI50593.2020.9311332.

[28] A.S. Hassanein, S. Mohammad, M. Sameer, and M.E. Ragab, ”A survey on Hough transform, theory, techniques and applications,” arXiv preprint arXiv:1502.02160, 2015.

[29] S.H. Chiu and J.J. Liaw, ”A proposed circle/circular arc detection method using the modified randomized hough transform,” Journal of the Chinese Institute of Engineers, vol. 29, no. 3, pp. 533-538, 2006.

[30] T.C. Chen and K.L. Chung, ”An efficient randomized algorithm for detecting circles,” Computer vision and image understanding, vol. 83, no. 2, pp. 172-191, 2001.

[31] B. Lamiroy, L. Fritz, and O. Gaucher, ”Robust circle detection,” in Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), vol. 1, pp. 526-530, IEEE, September 2007.

[32] A.O. Djekoune, K. Messaoudi, and K. Amara, ”Incremental circle Hough transform: An improved method for circle detection,” Optik, vol. 133, pp. 17-31, 2017.

[33] E.A. Sekehravani, E. Babulak, and M. Masoodi, ”Implementing Canny edge detection algorithm for noisy image,” Bulletin of Electrical Engineering and Informatics, vol. 9, no. 4, pp. 1404-1410, 2020.

[34] H. Zhang, R. Li, Z. Cai, Z. Gu, A. A. Heidari, M. Wang, H. Chen, and M. Chen, ”Advanced orthogonal moth flame optimization with Broyden–Fletcher–Goldfarb–Shanno algorithm: Framework and real-world problems,” Expert Systems with Applications, vol. 159, p. 113617, 2020.

[35] G. G. Wu, D. Qiu, Y. Yu, W. Pedrycz, M. Ma, and H. Li, ”Superior solution guided particle swarm optimization combined with local search techniques,” Expert Systems with Applications, vol. 41, no. 16, pp. 7536-7548, 2014.

[36] S. Li and M. Tan, ”Tuning SVM parameters by using a hybrid CLPSO-BFGS algorithm,” Neurocomputing, vol. 73, no. 10-12, pp. 2089-2096, 2010.

[37] A. M. Nezhad, R. A. Shandiz, and A. E. Jahromi, ”A particle swarm-BFGS algorithm for nonlinear programming problems,” Computers & Operations Research, vol. 40, no. 4, pp. 963-972, 2013..

[38] Y. M. Wang, Y. Li and J. B. Zheng, ”A camera calibration technique based on OpenCV,” The 3rd International Conference on Information Sciences and Interaction Sciences, Chengdu, China, 2010, pp. 403-406, doi: 10.1109/ICICIS.2010.5534797..

[39] Q. Li and Y. Xie, ”Randomised Hough transform with error propagation for line and circle detection,” Pattern Analysis & Applications, vol. 6, pp. 55-64, 2003.