Advanced 6D Sensor Development to Support Utilization of Cobot in High-accuracy Inspection

Guixiu Qiao, *Senior Member, IEEE*

2023 IEEE 19th International Conference on Automation Science and Engineering (CASE) | 979-8-3503-2069-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/CASE56687.2023.10260671

***Abstract* — Collaborative robots, also called Cobots, with the advantages of being safer and more cost-effective, are playing an important role in Industry 4.0. Being more flexible, cobots have been deployed in a wide variety of domains, for example, machine tending, material handling, packaging, assembly, drilling, riveting, welding, inspection, and three-dimensional (3D) printing. However, cobots have their challenges in accuracy and rigidity compared with traditional industrial robots. These challenges have become obstacles when implementing cobots in high-accuracy applications. Thus, developing technologies to support cobots to overcome the accuracy challenges is critical to the success of utilizing cobots in these applications. This paper presents an advanced sensor development project at the National Institute of Standards and Technology (NIST) to support using a Universal Robot (UR) in high-accuracy inspection via accuracy enhancement. The smart target is designed as a motorized target working with a vision-based system to acquire six-dimensional (6D) information (x, y, z, roll, pitch, and yaw) of a robot. Methodologies were developed to support the cobot's accuracy enhancement. A use case was built at the NIST's prognostics and health management lab using real-time sensor feedback and closed-loop control to improve the robot's performance.**

I. INTRODUCTION

Industry 4.0 is driving production lines to be faster and more flexible which requires the response to high-mix, low-volume manufacturing to satisfy the ever-changing consumer demands [1, 2]. The collaborative robot, also called cobot, is playing an important role in this new era with the advantage of safety, flexibility, and easy programming. Compared with expensive machines and industrial robots, the cost of cobots is much lower. The price of a cobot was about $28,000 on average in 2015. It is expected to be $17,500 by 2025 [3]. Moreover, the cost of integration of cobots is lower since no restricted area is required to change the workspace [4, 5]. Recent technology in Artificial Intelligence (AI) also enables faster learning that furthers the expansion of cobot applications [3, 6].

Based on the research [7], "the global collaborative robots market size was valued at USD 1.23 billion in 2022 and is expected to expand at a compound annual growth rate (CAGR) of 32.0%, from 2023 to 2030" [7]. Both small-and-medium enterprises (SMEs) and large enterprises have been prompted to adopt more cobots to enhance their production automation to increase enterprise competition. The automotive market occupied more than 24% of the global collaborative robot market in 2021. The application of cobots in industrial automation will increase significantly over the next five years [7]. Cobots are used in a wide range of industrial utilizations, for example, machine tending, material handling, packaging, assembly, drilling, riveting, welding, inspection, and 3D printing [2, 8, 9].

Compared with traditional industrial robots, cobots still face challenges in rigidity and accuracy, which obstruct cobots' usage in applications that require a higher level of accuracy. Some high-accuracy applications, for example, robot machining, drilling, and riveting, require the robot to apply forces on parts, thus, external forces are added on cobots as well. In this type of application, external forces may deform the robot's structure which influences the robot's performance. For example, if a cobot is programmed to drill a series of holes, if the material is thick and large drilling forces are applied, the position and orientation accuracy of the drilled holes may deviate from the designed positions. In other applications, for example, a cobot carries a 3D scanner to perform an inspection on a large car panel, the cobot is operating as a carrier for the non-contact measurement instrument. Since the 3D scanner usually has a limited field of view, the overall measurement needs to use the cobot's pose information for data registration. In this type of application, cobots' absolute accuracy becomes crucial. However, cobots usually are good with repeatability but not with absolute accuracy [4, 10, 11].

There are three major ways to enhance the accuracy of cobots [12, 13]. One aims to augment the cobot's accuracy through calibration and more advanced low-level control of the robot's joints. The second one focuses on using real-time sensor feedback and closed-loop control to improve the robot's accuracy. The third one measures the cobot's poses in real-time and uses the feedback to replace the robot's pose data, for example, using real-time feedback data to register the inspection data. For the first approach, although calibration may improve the cobot's absolute accuracy, external forces may still deviate the cobot's position and orientation from the designed poses because the cobot's rigidity is weaker compared with the traditional robot. Even for a non-contact inspection application such as a cobot-aided 3D scanner inspection, the change of payload may also influence the accuracy of the cobot. For cobots, adding sensors to enable the second and third approaches may be more effective to enhance the cobot's performance. In recent years, adding sensors to intelligent robotic manufacturing systems to enable real-time monitoring and control of the manufacturing process has shown significant growth [6, 14].

Enhancing a robot's position and orientation accuracy requires a sensor that can capture the robot's 6D information. There are many different types of 6D sensors, including laser trackers, total stations, pose matching, gauges, and coordinate

Guixiu Qiao is with the National Institute of Standards and Technology, Gaithersburg, MD 20899 USA (phone: 301-975-2865; fax: 301-990-9688 e-mail: guixiu.qiao@nist.gov).

measurement machines [15]. These are expensive measurement instruments and some measurement methods are slow. The vision-based system is gaining more attention in recent years. With the advantages of being non-contact and cost-effective, the vision-based 6D sensor provides useful feedback that can be utilized to provide real-time monitoring and control feedback to the cobot, as well as increase their applicability and safety.

The National Institute of Standards and Technology (NIST) has developed a novel smart target (U.S. patent 10885368) to support the precise measurement of a robot's position and orientation. The smart target is mounted on the object of interest (e.g., an end effector or a tool of a robot arm) to measure and track the robot's position and orientation. The smart target consists of fixed-wavelength light pipes and two high-accuracy rotary gimbals. The light pipe structure defines a coordinate frame that contains 6D information. One measurement of the smart target can output the pose of the object. The capture, analysis, and real-time feedback of 6D pose information enables the closed-loop control to improve the robot's accuracy.

The following sections present the hardware and software design of the smart target, the methodology to use smart target data to enhance the cobot accuracy, and a use case to support the utilization of a Universal Robot (UR) in the high-accuracy inspection.

II. DEVELOPMENT OF 6D MEASUREMENT SENSOR

A. *Design theory of the smart target*

A vision-based measurement system usually contains two or more cameras. It can measure 2D or 3D targets, for example, 2D markers or 3D spheres. Common features on the target (e.g., the centroid of the marker or centroid of the sphere's outer circle) are identified from each camera as 2D features; then 2D features are converted to 3D features via triangulation calculations.

To design a 6D target, the target needs to consist of features being measured by a vision-based measurement system. In addition, these features can be used to construct the position (x, y, z) and orientation (roll, pitch, yaw) information. One important rule for the selected feature is that the measurement value should remain the same when measuring the feature from different directions. For example, spheres are the common 3D targets used by the vision-based system for the sphere center feature. Each camera captures a 2D view of the sphere contour (the sphere's outer circle) and its centroid is calculated. After triangulation, the sphere center (x, y, z) is outputted. In an ideal condition, the sphere center will stay the same when measuring from different directions. However, in real applications, the calculation of the 2D centroid has uncertainties. For example, for reflective spheres used by optical tracking systems, center detection is influenced by infrared camera exposures and ambient light [16]. The uncertainty in the 2D centroid calculation contributes to the errors of the final 3D sphere center. When using 2D markers as the target, the uncertainty is larger because different view angles could cause larger bias in 2D marker images (some view angles may fail the measurement) [16, 17].

Multiple spheres or 2D markers are usually combined to make a structured target to represent the orientation. The output is a group of 3D point features. To represent the 6D information, these 3D points can be used to construct a coordinate frame. One point may be used as the origin to trace the position changes. Other points provide information to construct three axes of the coordinate frame. The other way is to use best-fit transformation to detect the point group's pose changes. The shortcoming of this kind of target is the large orientation uncertainty [15]. The uncertainty comes from the limited size of the target. Since an axis direction is only defined by two or a few points spaced by a limited distance, a small error in displacement may be converted to a large direction uncertainty for the target measurement.

To avoid the above-mentioned concerns, a novel way to design a new 6D target is developed. As shown in Fig. 1, line features are used instead of point features to create a coordinate system. To make the line feature measurable by a vision-based system, cylindrical light pipes were made using a high-precision machine with an accuracy of 20 um in roundness. Three narrow-wavelength color-laser LEDs were used to light up the cylindric pipes. The pipes are made of special light-guiding material to create uniform lighting of the light pipe as shown in the top right picture of Fig. 1. The special material guarantees the center line of the light pipe is measurable from different directions. Each center line is represented by hundreds of detected points on the line.

Instead of using a sphere center to represent the origin of the coordinate frame, a cross-shape light pipe is developed. The cross center is defined as the origin, which is represented by the intersection of two lines. The origin created by two line-feature intersections has far better accuracy than the

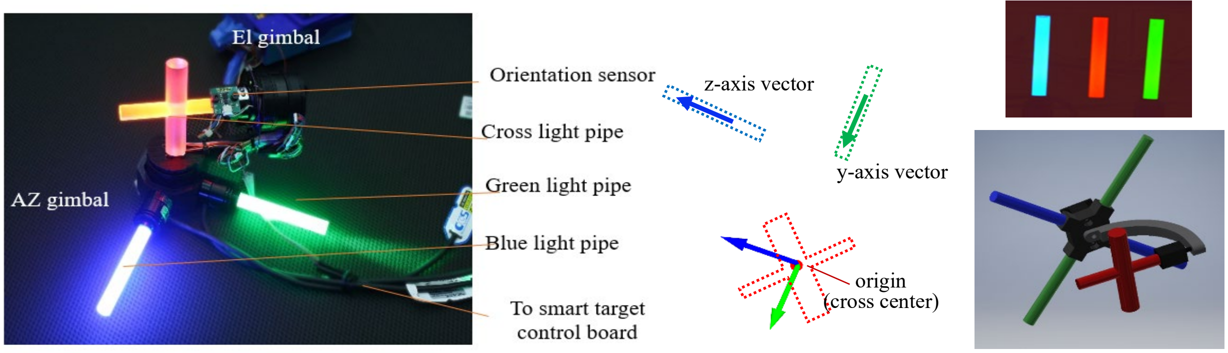


Fig. 1. Smart target – a vision-based 6D sensor

traditional single-sphere center method. The cross-shape light pipe is mounted at the center of the two rotation gimbals, the elevation (EL) gimbal and the azimuth (AZ) gimbal. The purpose is to allow the cross pipe to rotate and always face toward the measurement instrument. Otherwise, when the smart target is mounted on a robot, the robot's motion may carry the smart target to an unmeasurable view, for example, the cross pipe being parallel to the measurement instrument.

The initialization of a coordinate frame needs an origin and two vectors for axis directions as shown in Fig. 1. The blue-color light pipe is used to define the vector for the z-axis direction. The green-color light pipe is used to provide the vector for the y-axis direction. The number of blue/green light pipes could be more than one to create redundancy. The mounting position of the blue and green light pipes can be customized to any convenient location on the target since only vectors of axis-direction information are needed from them.

B. *Hardware development*

Two versions of the smart target were developed. One was a motorized version with an orientation sensor mounted on the EL gimbal. Another one was a lite version without motors and the orientation sensor.

Fig. 2 shows the latest design of the motorized smart target. The motorized version has an orientation sensor mounted on the EL shaft. The first step of using the smart target is to set up a measurement pose. The cross-shape light pipe is manually rotated (by hand or jogged by remote control) toward the vision-based measurement instrument. This will be the good angle for the vision-based system to measure the smart target. Next, a “teach” button will be pressed to teach the smart target to keep this orientation. If the smart target is rotated away (for example, the robot arm carries the smart target and rotates it away from the original pose), the orientation sensor will sense the orientation changes and send the deviation signals to the control board. The control board receives deviation signals and uses them to drive the EL and AZ gimbal motor back to its original orientation. A battery is used to provide power to the control board, motors, sensor, and light pipes.

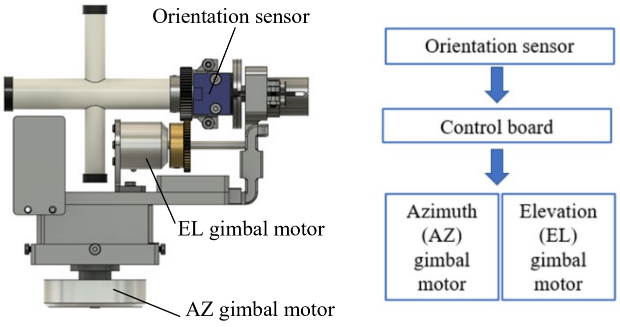


Fig. 2. Motorized versions of Smart target

The motorized smart target is used for measuring an object of interest that is in a continuous motion, for example, mounted on the robot arm, or a 5-axis computer numerical control (CNC) machine. By constantly rotating the smart target toward the measurement instrument, the system can consistently measure the position and orientation of the object of interest, avoiding the issues of bad target angles that could add measurement uncertainties.

Fig. 3 shows the design of the lite version of the smart target. It is a more compact and cost-effective design without motors and a control board. The lite version of smart target is used in conditions where the object of interest is not in a continuous motion or just translations without many orientation changes. One example of the application is cobot user frame calibration.

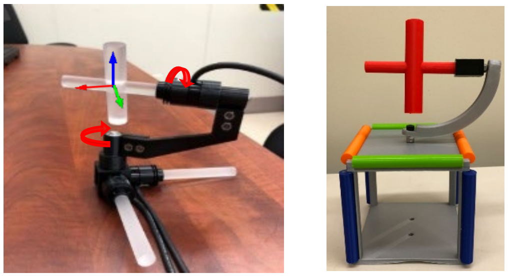


Fig. 3. Lite versions of smart target

Since cobots are designed for easy reconfiguration in a production line to adopt demand changes, their user frame needs to be frequently recalibrated after the working area changes [12]. This kind of calibration is not to improve the arm’s accuracy, but to find the relationship of an object with respect to the robot. There are three different local calibrations [12]: 1) The calibration between the robot base coordinate system to the end-effector is also called hand-eye calibration. This calibration is to adapt changes caused by reconfiguration, thermal impact, vibration, parts wearing, etc. 2) If in the same workspace, there are external tools like a vision system, a tool used for the task, or even another robot to work together, the relationship between them needs to be calibrated. 3) To make programming easier, calibration of the workpiece coordinate system is often taken to find the relationship to the robot base coordinate system. This calibration enables the programming in the workpiece frame regardless of where the robot is positioned.

The lite version of the smart target is suitable for the above-mentioned applications, for example, mounting a lite version of the smart target at a robot base for multiple robot relationship registration. With the clear definition of a coordinate frame from the cross-center origin and colored light pipe axis directions, the target provides an intuitive frame definition that allows users to perform the procedure effectively. Moreover, as shown in Fig. 3, the location, size (light pipe is machinable), and the number of blue and green light pipes can be customized based on the need. The right picture in Fig. 3 shows a more redundant design.

C. *Software development*

The purpose of the software development is to develop algorithms and an image process library to process smart target images, extract features, and output 6D data. The smart target can be integrated into a variety of camera-based vision systems. The image process library will serve as the software development kit (SDK) for developers who would like to adopt the smart target to acquire 6D information. The SDK requires high-speed image processing to generate dynamic, real-time measurements. The development of the SDK implements sub-pixel level image processing for feature extraction, critical for high-precision measurement.

A graphical user interface (GUI) tool was developed to support real-time measurement. The initial effort focuses on the integration of the vision-based system. Multiple threads were used to handle image collection from cameras on the vision-based system to perform 2D calculations. As shown in Fig. 4., a synchronized ring buffer was designed to interface with the cameras to ensure the safe read/write of image pairs (pairs of the image from the left camera and the right camera of the vision-based system). Exception monitoring was added to capture any abnormal failures from cameras. The read procedure always reads from the last most recently finished buffer to ensure thread-safe. When there is the need of integrating with another new vision-based system, the main algorithms stay the same. Only the camera initialization and control parts need to be changed.

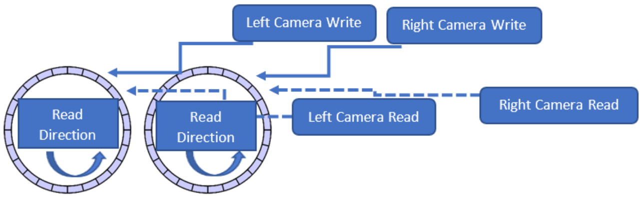


Fig. 4. Ring buffer for the stereo camera setup

The kernel of the SDK is the image processing algorithms. Images are processed as the following:

1) Image undistortion and rectification: Images were undistorted by loading calibration files and rectified to align the image pair horizontally. For camera calibration, a tool for automatic camera calibration was developed and published at NIST (https://www.nist.gov/services-resources/software/automatic-checkerboard-corner-detection-and-data-processing-tool). The tool could improve the accuracy of checker-board corner detection and automate the calibration process and data processing.

2) Segmentation of color: The red, blue, and green light pipes were identified and segmented using color filters.

3) Edge detection: A Canny edge detector was used to detect edges on three color channels, as shown in Figure 5. Edges of the three-color light pipes were detected. They will be used to find the center line for the next step.

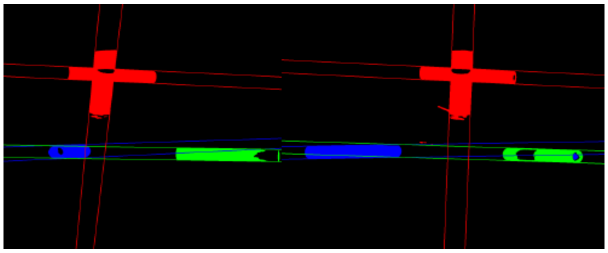


Fig. 5. Result after edge detection

4) Centerline detection and clean up: The Hough line detection algorithm was implemented to find the centerlines of the light pipes [18]. Filters were added with a noise-removal function to remove extra lines. The results are shown in Figure 6. Center lines of the different color light pipes are detected. The cross-center was calculated by the intersection of two center lines on the cross-shape light pipe.

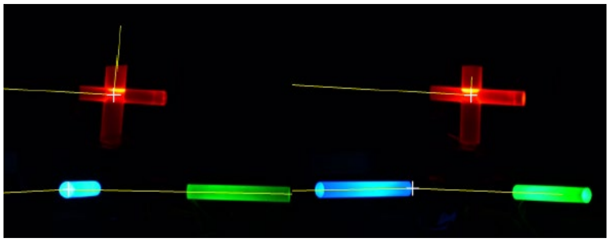


Fig. 6. Result after centerline detection

5) 3D triangulation: In this step, the detected 2D lines were converted into 3D. The final output was the red cross center (x, y, z) and three vectors that represented the direction of three axes. The positions and three axes were converted to a 4 by 4 transformation matrix.

The algorithm can be run on either Central Processing Unit (CPU) or Graphics Processing Unit (GPU). SIMD (Single Instruction Multiple Data) of CPU processors and GPU computation were used with the NVIDIA CUDA library to speed up the calculation. The current calculation speed was 30 fps based on the current hardware we are using. The accuracy of the feature detection was 0.1 mm.

III. METHODOLOGY OF COBOT ACCURACY ENHANCEMENT

One method used for robot accuracy enhancement is robot calibration. NIST has developed a methodology for robot accuracy assessment and calibration. An error model was created to handle both position-independent geometry errors and position-dependent motion errors. Details of this methodology are presented in [18]. Different from traditional robots, cobots have more challenges in accuracy enhancement. With a lighter weight and less rigid design, external forces, payload, and speed may deviate the cobot's accuracy even after a good calibration. The methodology of using real-time sensor feedback could help to address the challenge.

Fig. 7 shows an example of the application of cobot-aided inspection. A cobot is inspecting a workpiece using a non-contact inspection sensor mounted on the robot end-effector. The inspection sensor could be a 3D scanner or other types of sensors. Usually, the inspection sensor needs to be perpendicular to the workpiece surface and maintain a certain distance from the surface. Two smart targets are used in the system. One is mounted on the robot base (SM1, a lite smart target can be used). The other one is mounted on the robot end-effector (SM2). The SM2 is used as the real-time feedback sensor to support the cobot-aided inspection. There are seven coordinate frames in the system as shown in Fig. 7: a) Vision-based measurement instrument frame; b) smart target on base frame; c) cobot base frame; d) smart target on tool frame; e) tool center position (TCP) frame; f) inspection sensor frame; and g) workpiece frame. There are three unknown constant offsets between (b) and (c), (d) and (e), and (e) and (f).

The first step of this methodology is to register the cobot's base frame (c) and the inspection sensor frame (f)

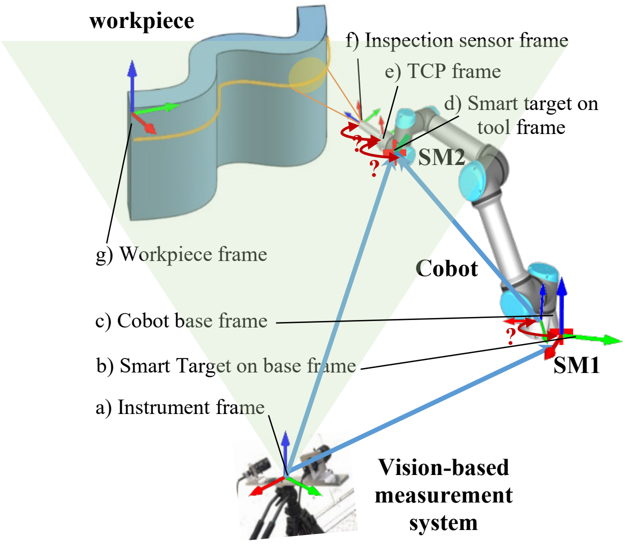


Fig. 7 Cobot-aided inspection

into the measurement instruments frame (a). Procedures are taken to develop the following:

1) Local calibration to register the inspection sensor frame (f) to robot base frame (c). In this step, common targets (usually provided by the inspection sensor vendor) are measured by the inspection sensor and robot TCP. Between the TCP and the inspection sensor mounting position, a constant offset is applied. Using these common target measurements, the unknown constant offset between (e) and (f) is figured out. The inspection sensor frame (f) is registered in the robot's base frame (c).

2) The cobot is programmed to move to a few positions in the cobot's workspace. The TCP position is represented in the matrix between (e) and (c). At the same time, the smart target positions are measured under the vision-based measurement system, represented by a matrix between (a) to (d). Using this set of position measurements, the constant offset between (d) to e) is computed. Given the known (c) to (e) and (a) to (e), the matrix between (c) to (a) is calculated.

3) Given the known of (c) to (a) and (c) to (f), the matrix of (a) to (f) is figured out. Thus the measurement of the workpiece can be converted under the instrument frame (a).

4) An additional step is added to measure the SM2 when SM1 is measured. The purpose is to find the constant offset between (b) and (c). This step creates a measurable robot base frame. A user could monitor the robot base changes by measuring the SM2. In the condition of the robot being relocated, users no longer need to redo the second step calibration to relocate the robot base relative to the measurement instrument, but could directly measure the SM1 and SM2. Then the complete matrix from (a) to (f) can be used directly. This can save a lot of calibration time for setup reconfiguration. In the condition when the cobot base indeed changes during the measurement, this method enables the correction by detecting the changes.

The methodology also developed two approaches of how to use real-time 6D information in cobot applications. One method is to use smart target measurement for closed-loop control to improve the robot's accuracy. As shown in Fig. 8, a robot program is converted to a robot target trajectory in the format of x, y, z, roll, pitch, and yaw in a robot coordinate frame. They are the input to the robot control to drive the robot's motion. At the same time, the vision-based system measures the robot's real-time trajectory positions via the smart target. A real-time compensation algorithm processes the 6D information, performs coordinate transformations, and calculates the deviations to create feedback for the robot control. This method is suitable for applications that need real-time correction for robot control.

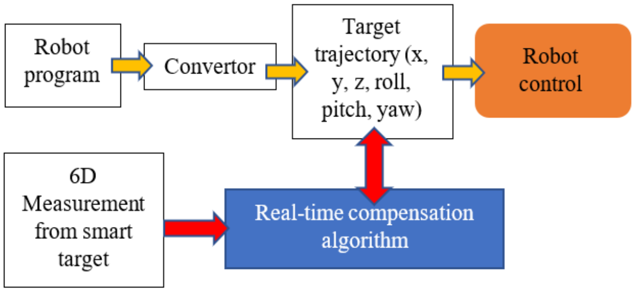
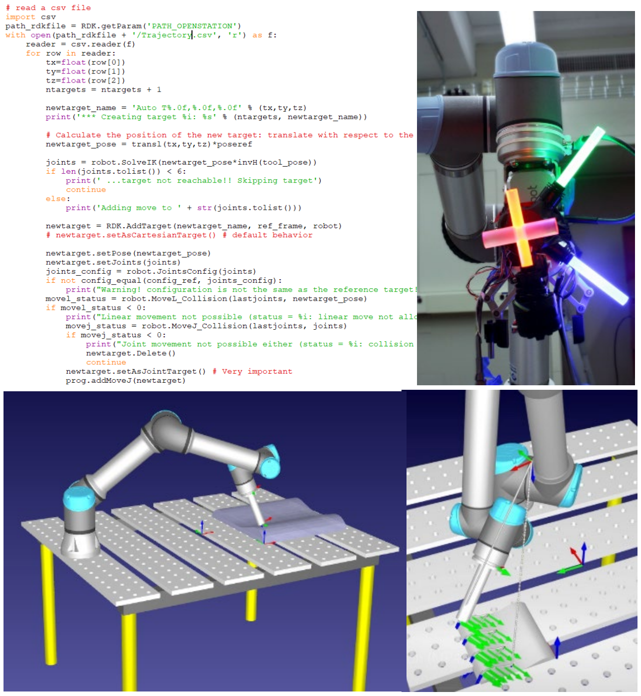


Fig. 8 Cobot accuracy enhancement using close-loop control

Another method is to measure the cobot's poses in real time and use the measured 6D information to register the inspection data. As shown in Fig. 7, the matrix chain is from (a) -> (d) -> (e) -> (f) -> (g). This method is suitable for cobot-aided non-contact applications like inspection, where cobots cannot satisfy the high accuracy requirement for data registration and no feedback for cobot control is needed. Synchronization of the inspection data and 6D measurement data is important in this method. Hardware triggers could be considered to synchronize the instruments. Another way is to use time stamps to align the inspection and 6D data. Fine adjustments may be taken to fine-tune the possible legacy from instruments.

IV. USE CASE DEVELOPMENT

A use case of using smart targets to support cobot-aided inspection was developed at NIST. It is under the Prognostic and Health Management (PHM) for robot system project, developing measurement science to support monitoring, diagnostic, and prognostic technologies to identify performance degradation and minimize unplanned downtime in manufacturing systems. A universal robot UR5 was used to develop the use case. A non-destructive sensor is mounted on the end-effector of the UR5. The smart target was mounted at the last joint next to the non-destructive sensor as shown in Fig. 9. A part to be inspected is mounted on the table. To plan the robot scanning through the part surface, we did not use the traditional method of importing the part's computer-aided design (CAD) model and creating an offline program. Instead, imitation-based teaching is used for robot path generation. The robot is hand-guided to scan through the part surface. The vision-based system measured the smart target to get the robot's trajectory. These measured poses were saved as target positions (in a .csv file) to create a robot program. This imitation-based teaching is fast and very useful when the part has no CAD model to perform offline programming.



```
# read a csv file
import csv
path_rdkfile = RDK.getParam('PATH_OPENSTATION')
with open(path_rdkfile + '/Trajectory.csv', 'r') as f:
    reader = csv.reader(f)
    for row in reader:
        tx=float(row[0])
        ty=float(row[1])
        tz=float(row[2])
        ntargets = ntargets + 1

        newtarget_name = 'Auto T%.0f,%.0f,%.0f' % (tx,ty,tz)
        print('*** Creating target %i: %s' % (ntargets, newtarget_name))

        # Calculate the position of the new target: translate with respect to the
        newtarget_pose = transl(tx,ty,tz)*poseref

        joints = robot.SolveIK(newtarget_pose*invH(tool_pose))
        if len(joints.tolist()) < 6:
            print(' ...target not reachable!! Skipping target')
            continue
        else:
            print('Adding move to ' + str(joints.tolist()))

        newtarget = RDK.AddTarget(newtarget_name, ref_frame, robot)
        # newtarget.setAsCartesianTarget() # default behavior

        newtarget.setPose(newtarget_pose)
        newtarget.setJoints(joints)
        joints_config = robot.JointsConfig(joints)
        if not config_equal(config_ref, joints_config):
            print("Warning! configuration is not the same as the reference target!
        movel_status = robot.MoveL_Collision(lastjoints, newtarget_pose)
        if movel_status < 0:
            print("Linear movement not possible (status = %i: linear move not allo
            movej_status = robot.MoveJ_Collision(lastjoints, joints)
            if movej_status < 0:
                print("Joint movement not possible either (status = %i: collision
                newtarget.Delete()
                continue
            newtarget.setAsJointTarget() # Very important
            prog.addMoveJ(newtarget)
```

Fig. 9. Use case development for cobot-aded inspection

A simulation software RoboDK is used in this development. RoboDK Application Programming Interface (API) for Python is utilized to generate a robot program using the saved target positions to scan through the part surface. When the non-constructive inspection sensor measures the part, the vision-based system measures the 6D information of the smart target as well. The 6D information is used to merge the non-constructive sensor's data. It is also used to compare with the saved trajectory to find deviations coming from different speeds and thermal conditions etc. for performance improvement.

V. SUMMARY

With the growing use of cobots in industrial applications, cobots need to address the challenges in accuracy and agility, which have hindered the use of cobots in high-accuracy applications. This paper presents the development of an advanced 6D sensor to support cobots in high-accuracy applications. A novel 6D sensor based on vision-based measurements is developed (US patent 10885368). The motorized 6D sensor could help manufacturers not only with assessing the accuracy of robots, but also with using the real-time 6D information in robot close-loop control to enhance the cobot's accuracy. Methodologies and a use case were presented. Future efforts are underway to develop additional industrial use cases for the technology.

NIST DISCLAIMER

Certain commercial entities, equipment, or materials may be identified in this document in order to illustrate a point or concept. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

REFERENCES

[1] A. Weiss, A.-K. Wortmeier, and B. Kubicek, "Cobots in Industry 4.0: A Roadmap for Future Practice Studies on Human-Robot Collaboration," *IEEE transactions on human-machine systems,* vol. 51, no. 4, pp. 335-345, 2021, doi: 10.1109/THMS.2021.3092684.

[2] R. Galin, R. Meshcheryakov, S. Kamesheva, and A. Samoshina, "Cobots and the benefits of their implementation in intelligent manufacturing," *IOP conference series. Materials Science and Engineering,* vol. 862, no. 3, p. 32075, 2020, doi: 10.1088/1757-899X/862/3/032075.

[3] A. Borboni, K. V. V. Reddy, I. Elamvazuthi, M. S. Al-Quraishi, E. Natarajan, and S. S. Azhar Ali, "The Expanding Role of Artificial Intelligence in Collaborative Robots for Industrial Applications: A Systematic Review of Recent Works," *Machines (Basel),* vol. 11, no. 1, p. 111, 2023, doi: 10.3390/machines11010111.

[4] P. Akella *et al.*, "Cobots for the automobile assembly line," 1999, vol. 1, no. Conference Proceedings: IEEE, pp. 728-733 vol.1, doi: 10.1109/ROBOT.1999.770061.

[5] A. S. Bisen and H. Payal, "Collaborative robots for industrial tasks: A review," (in English), *Materials today: proceedings,* vol. 52, pp. 500-504, 2022, doi: 10.1016/j.matpr.2021.09.263.

[6] A. Correia Simões, A. Lucas Soares, and A. C. Barros, "Factors influencing the intention of managers to adopt collaborative robots (cobots) in manufacturing organizations," (in English), *Journal of engineering and technology management,* vol. 57, p. 101574, 2020, doi: 10.1016/j.jengtecman.2020.101574.

[7] "Collaborative Robots Market Size, Share & Trends Analysis Report" By Payload Capacity, By Application (Assembly, Handling, Packaging, Quality Testing). Available online: https://www. grandviewresearch.com/industry-analysis/collaborative-robots-market (accessed on 19 November 2022).

[8] B. B. Van Acker, P. D. Conradie, D. D. Parmentier, J. Joundi, J. Saldien, and P. Vlerick, "An explorative study on the acceptability of industrial cobots in (future) operators," vol. 34, no. 1, pp. 13-51, 2021, doi: 10.5117/GO2021.1.003.VANA.

[9] F. Rossi *et al.*, "Effective integration of Cobots and additive manufacturing for reconfigurable assembly solutions of biomedical products," *International journal on interactive design and manufacturing,* vol. 14, no. 3, pp. 1085-1089, 2020, doi: 10.1007/s12008-020-00682-9.

[10] S. El Zaatari, M. Marei, W. Li, and Z. Usman, "Cobot programming for collaborative industrial tasks: An overview," *Robotics and Autonomous Systems,* vol. 116, pp. 162-180, 2019.

[11] M.-A. Bauda, A. Grenwelge, and S. Larnier, "3D scanner positioning for aircraft surface inspection," in *ERTS 2018*, 2018.

[12] R. Pagani, C. Nuzzi, M. Ghidelli, A. Borboni, M. Lancini, and G. Legnani, "Cobot user frame calibration: evaluation and comparison between positioning repeatability performances achieved by traditional and vision-based methods," *Robotics,* vol. 10, no. 1, p. 45, 2021.

[13] T.-P. Phan, P. C. P. Chao, J.-J. Cai, Y.-J. Wang, S.-C. Wang, and K. Wong, "A novel 6-DOF force/torque sensor for COBOTs and its calibration method," 2018, no. Conference Proceedings: IEEE, pp. 1228-1231, doi: 10.1109/ICASI.2018.8394511.

[14] E. Oztemel and S. Gursev, "Literature review of Industry 4.0 and related technologies," *Journal of intelligent manufacturing,* vol. 31, pp. 127-182, 2020.

[15] G. Qiao, "Advanced sensor and target development to support robot accuracy degradation assessment," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019: IEEE, pp. 54-59.

[16] A. D. Wiles, D. G. Thompson, and D. D. Frantz, "Accuracy assessment and interpretation for optical tracking systems," in *Medical Imaging 2004: Visualization, Image-Guided Procedures, and Display*, 2004, vol. 5367: SPIE, pp. 421-432.

[17] R. Khadem *et al.*, "Comparative tracking error analysis of five different optical tracking systems," *Computer Aided Surgery,* vol. 5, no. 2, pp. 98-107, 2000.

[18] G. Qiao and B. A. Weiss, "Quick health assessment for industrial robot health degradation and the supporting advanced sensing development," *Journal of manufacturing systems,* vol. 48, pp. 51-59, 2018.