

COTS vision system, radiation sensor and 6 DoF robot for use in identifying radiologically uncharacterised objects

Stephen D. Monk, David Cheneler, Craig West, Manuel Bandala, Nile Dixon,
Allahyar Montazeri, C. James Taylor
Engineering Department, Lancaster University, UK
Email: s.monk@lancaster.ac.uk

Abstract— A novel, semi-autonomous radiological scanning system for inspecting uncharacterised objects is presented. The research utilises commercially available off-the-shelf (COTS) electronic components, intended for use in relatively low radioactive dose environments. To illustrate the concept, a Microsoft Kinect, a Universal Robots UR3 (6 degree of freedom) manipulator and a Kromek RadAngel gamma radiation spectrometer are utilised. The new control software allows the remote operator to select the required object, before the robot arm autonomously scans it, enabling the generation of various radiological spectra shapes. These data inform the operator of any likely radioisotopes present and where they are located in the object.

Index Terms— UR3 manipulator; RadAngel radiation detector; Microsoft Kinect; radiological scanning; vision system; nuclear decommissioning

I. INTRODUCTION

The nuclear industry has typically lagged a few years behind the latest commercial developments when it comes to the utilization of robotic technology. Indeed, it has sometimes been the major disasters suffered by the nuclear industry that have acted as the driver to adopt and develop new robotic systems, such as remote handling techniques. An example came when the Great East Japan earthquake of 2011 damaged the Fukushima Daiichi nuclear power station. The previously developed robots were relatively primitive, at low levels of maturity and not suited to the task [1, 2]. Instead, general use and more mature robotic platforms offered by traditionally non-nuclear companies such as irobot, Qinetiq and Brokk were used [2, 3, 4]. However, research into robotics within the nuclear industry has subsequently increased significantly, with companies such as Toshiba producing bespoke and mature robots for specific use within the stricken reactor at Fukushima Daiichi [5].

Naturally, there is a significant interest in radiation hardened robotics within the industry, hence hydraulically actuated robots [6-11] and platforms that feature radiation hardened components [12] have traditionally been in favour. However, although operations within environments such as hot cells involve very high levels of radiation, other typical nuclear environments may exhibit radiological levels which, despite being too high for daily human exposure, are still far too low to damage conventional hardware. For example, a radiation worker in the UK may not be exposed to more than 20 mSv over the course of a year. Assuming a workplace

exposure of two hours per day over the course of 46 weeks, this equates to a maximum dose rate of 43.4 $\mu\text{Sv/hr}$. In terms of damage to commercially available off-the-shelf (COTS) electronic components, this is a low dose rate. Discussions of radiation hardness tend to begin at around eight orders of magnitude higher [12]. Thus, although commercially available robotic platforms may not be useable in very high radiation environments (e.g. around a stricken reactor), they may be more useable within lower activity areas, for example, in the categorisation of waste. Hence, at present in the UK, an increasing volume of research involves the development of large-scale industrial robot platforms designed for locations with relatively low radiological activity e.g. [6].

The present article concerns the development of a novel, semi-autonomous, radiological scanning system capable of inspecting uncharacterised objects for contamination. The research focuses on low cost solutions where possible i.e. utilizing COTS components that can be purchased cost effectively. Sections II, III, IV and V discuss the various system components, including the overall configuration, the vision system, robot hardware and gamma ray spectrometer, respectively. Furthermore, the authors have developed new identification and characterisation software. This enables the operator to see the radiological spectra shape over a user defined number of locations across the object, so that the type of contamination (i.e. which radioisotope) and position within the object, can both be identified. Preliminary results of the new system in operation are presented in section VI, followed by the conclusions in section VII.

II. SYSTEM COMPONENTS AND SETUP

To illustrate the generic concept, the system developed here consists of a Microsoft Kinect used to determine the position in 3D space of the object, a Universal Robots UR3 to position the radiological instrumentation and a RadAngel radiation detector to determine the gamma ray spectrum emitted by the object. The robot and vision system are linked and both controlled via MATLAB, a software environment capable of rigorous mathematical operation, as well as connection to peripherals via ports such as USB or Ethernet. The spectra generated by the RadAngel features its own bespoke and freely downloadable software (kspect) for creating a radiological spectrum of the incident radiation. In addition, however, the present authors have developed python software to record a series of spectra as the position of the detector changes. The UR3 and vision systems are located on either side of the workspace, as shown in Figure 1.

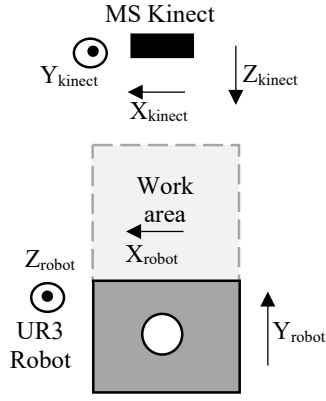


Figure 1: System set-up.

Within the vision system algorithm, X_{kinect} is determined as being from right to left in the camera field of view, Y_{kinect} represents top to bottom and Z_{kinect} is depth from the camera location outwards. The robot co-ordinate scheme differs, in that Y_{robot} is taken as from the robot towards the vision system (analogous to Z_{kinect}), and Z_{robot} is height from the base to the robot (analogous to Y_{kinect}). The X orientation for both the Kinect and robot is the same. Finally, a diagram of the UR3 robot is shown in Figure 2, with the various link lengths indicated.

III. THE VISION SYSTEM

The Microsoft Kinect v2 camera can be purchased at low cost (<£100), due to its popularity and thus mass production. It was discontinued in January 2018 although it is still being used for research due to factors such as the support still available online. Although there are more modern vision systems, such as the Orbbec Astra Pro or the Intel Realsense, the Kinect sensor is sufficient for the present proof of concept study.

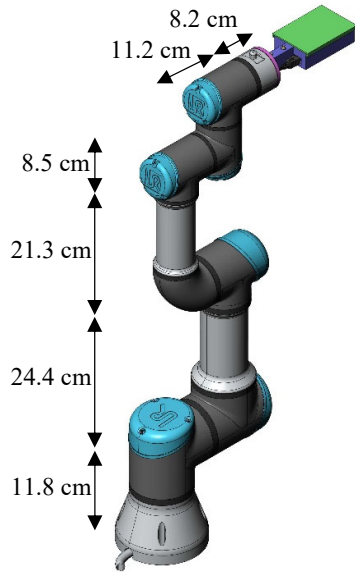


Figure 2: The UR3 robot.

The Kinect v2 was released in 2014 and features an RGB camera with a resolution of 1920 x 1080 pixels, and a field of view of 84.1° by 53.8° [13]. The depth camera operates via a time of flight system [14] and features a resolution of 512 x 424 pixels, a field of view of 70.6° x 60° and a depth range of between 0.4 m and 4.5 m. The MATLAB algorithm used here has evolved from earlier research by several of the present authors [6-10]. In the first instance, the operator chooses when to take a snapshot of the Kinect RGB camera field of view. The algorithm determines the major interfaces between the colours in the RGB image, as recovered via Canny edge detection [15], with the thresholds selected and altered by the operator as required.

The object of interest is selected by clicking on it, whereupon the height, width, orientation and centre of the object is determined via the 'regionprops' function in MATLAB. This information determines the X_{kinect} and Y_{kinect} co-ordinates of the top left and top right of the rectangular front face of the object (known as startX, endX, startY and endY). The Z_{kinect} co-ordinate is determined by interrogating the depth image for its value at the centre position of the object. Within the Kinect universe, the Z co-ordinate is returned to the user as the same regardless of direction, i.e. it is the component along the Z-axis rather than the distance to the object.

The X_{kinect} and Y_{kinect} values are returned as RGB pixel values and thus the determination of the Cartesian robot position values require an additional step, as follows. The Kinect X axis spans an angle of 84.1° and features 1920 pixels i.e. 0.0438 degrees/pixel. The pixel numbers start at the top right-hand corner of the image. Hence, the origin is shifted to the centre of view by subtracting the returned value from 960. This is multiplied by 0.0438 degrees/pixel to obtain the angle of the pixel relative to the centre of view. Hence, visualised as in Figure 3, the cartesian values are determined as follows:

$$Y_{robot} = -1.63 + (Z_{kinect}/1000) \quad (1)$$

$$X_{robot} = (Z_{kinect}/1000) * \tan((960-X_{kinect})*0.0438) \quad (2)$$

$$Z_{robot} = (Z_{kinect}/1000) * \tan((540-startY)*0.0498) \quad (3)$$

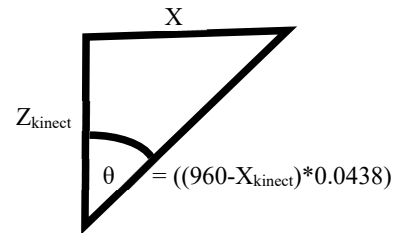


Figure 3: Converting Kinect to UR3 co-ordinates.

IV. THE UR3 ROBOT

The robotic platform is the aluminium constructed Universal Robots UR3, which possesses 6 revolute degrees of freedom (DOF), a payload of 3 kg, weighs 11.2 kg, has a reach of 500 mm and a footprint of 128 mm (Ø). The designed temperature range for operations is between 0 and 50° C which may preclude the robot from being used in some of the higher temperature areas within a nuclear facility. Compared to

other commercially available robots, it is a cost-effective solution at around £15,000. The UR3 is primarily controlled through a teach-pendant where an operator may perform simple operations, such as setting waypoints and programming the robot to move between them using various pathways. However, it is also possible to control the platform using other 3rd party software environments, such as ROS and MATLAB, via an IEEE 802.3u connection (Fast Ethernet). For the present work, a bespoke program utilizes the teach pendant and commands sent through the Ethernet gateway.

The Denavit-Hartenberg parameters of the UR3 robot shown in Table 1 are obtained from the Universal Robots website. The inverse kinematic model of the robot is generated via Peter Corke's Robotics Toolbox [16], using an iterative method. This allows a basic initial pose to be specified and thus ensures the elbow of the arm can be maintained in an upwards position in all configurations. The code used within the MATLAB environment to move the end effector to the required position is shown in Table 2. The system is designed such that the scanning will always occur from above. The Euler angle is $(0, \theta, 0)$ where $\theta = (\pi + a)$ and a is the deviation from horizontal of the top of the object.

The MATLAB function 'eul2rotm' converts these Euler angles to the rotation matrix required before the DH parameters are defined. A thirty-stage loop ensures the robot reaches 30 poses with locations between $(X_{robot}, Y_{robot}, Z_{robot})$ and $(X_{robotF}, Y_{robot}, Z_{robotF})$. It should be noted that it is just the X_{robot} and Z_{robot} parameters which are changing here; Y_{robot} is constant throughout the scan. This is because it was found that identifying the Z_{kinect} values of the corners of the object was inconsistent, with many of the tests returning the correct values, but with other tests returning either the value corresponding to the background and sometimes zeros. Thus, only the Z_{kinect} value which corresponds to the centre of the object was used – this proving more reliable in practice.

For each iteration, the pose matrix is constructed and then the 'ikunc' function is used with an initial suggested joint angle set corresponding to an exaggerated 'elbow up' pose. The set of joint angles 'qq' is sent to the teach pendant via the 'moverobotJoint' function, with the code on the pendant in turn transferring the joint positions to the robot itself.

Table 1: Denevit-Hartenberg parameters for the UR3 robot.

Joint	θ (rad)	a (m)	d (m)	α (rad)
1	0	0	0.1519	$\pi/2$
2	0	-0.24365	0	0
3	0	-0.21325	0	0
4	0	0	0.11235	$\pi/2$
5	0	0	0.08535	$-\pi/2$
6	0	0	0.0819	0

Table 2: MATLAB code used to send the end-effector to the required location with the correct orientation.

```
eul = [0 theta 0]; % Euler angle required of the end effector
R = eul2rotm(eul); % Rotation matrix from Euler angles
L(1) = Link([0 0.1519 0 pi/2]); % Define DH parameters
L(2) = Link([0 0 -0.24365 0]);
L(3) = Link([0 0 -0.21325 0]);
L(4) = Link([0 0.11235 0 pi/2]);
L(5) = Link([0 0.08535 0 -pi/2]);
L(6) = Link([0 0.0819 0 0]);
for = 0:30 % Thirty positions across the object
    XXX = Xrobot + ((XrobotF - Xrobot) * (loop/30));
    ZZZ = Zrobot + ((ZrobotF - Zrobot) * (loop/30));
    P = [XXX; Yrobot; ZZZ];
    C = [R P];
    S = [0 0 0 1];
    F = [C;S]; %% F is now the required pose matrix
    UR3 = SerialLink(L, 'name', 'six link');
    mdl_ur3;
    qq = ur3.ikunc(F, [1.5 -1.3 0.6 -1.2 -1.6 -1.5]);
    addpath('./interface');
    moverobotJoint(s, qq)
    pause(30);
end
```

V. THE RADANGEL

The RadAngel, shown in Figure 4, is a highly portable and affordable (~£650) Cadmium Zinc Telluride (CZT) based entry level gamma ray spectrometer [17]. The crystal is a cube with side length of 5 mm, which features within an overall package size of 90mm x 55mm x 25mm, with a mass of 135g. The output is over 4096 channels and features a spectral resolution of 3.5% over a range of 40 keV to 1500 keV. The bespoke software supplied (kspect) allows the export of the spectroscopic data recorded into a general format such as .txt.



Figure 4: The RadAngel from Kromek [18].

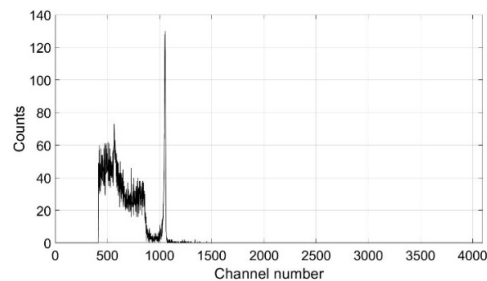


Figure 5: A caesium-137 source used to calibrate the sensor.

The 'kspect' software is utilized here in order to calibrate the sensor, which displays the energy in terms of channel number rather than the more useful electron volts. A 309 kBq cesium-137 source was placed as close as possible to the RadAngel for 5 minutes. The output spectroscopic results are shown in Figure 5, with the signature 661.6 keV Cs-137 sharp peak corresponding to channel number 1053 in this case. This information is used to convert the x-axis from channel number to energy. The authors are unable to determine exactly where the crystal is located within the RadAngel package. To estimate this, however, 10 readings of one minute each were taken at between 1 and 10 cm from a Cs-137 source. The background level is subtracted and total counts are plotted against distance, with a power trendline added to the data as follows:

$$\text{Counts} = 12,944 * \text{Distance}^{-1.157} \quad (4)$$

If the crystal is at 0 cm (i.e. right against the wall of the detector casing) then the exponent would be -2 due to the expectation that radiological flux decreases with the square of the distance. The results are replotted with the position of the crystal assumed to be at -1 cm (i.e. the distance to the 1 cm reading is postulated to be 2 cm) with the fit as in equation (5). This is repeated assuming the crystal is at -2 cm (Eqn. 6) and -3 cm (Eqn. 7):

$$\text{Counts} = 32,399 * \text{Distance}^{-1.518} \quad (5)$$

$$\text{Counts} = 83,489 * \text{Distance}^{-1.874} \quad (6)$$

$$\text{Counts} = 242,125 * \text{Distance}^{-2.267} \quad (7)$$

Interpolation indicates that the crystal is located at a position approximately 2.32 cm from the end of the detector casing.

A resourced python script [18] compiled in a ROS catkin package was initially used to read each detection event and store the total counts for each channel in a spectrum file. This code is further edited to allow for repeat measurements and to print all radiation data to a .CSV file in a clearer format. This package is started using a single command line which allows the user to select the time over which a measurement should be taken and the total number of measurements. The package is compiled in both melodic and kinetic versions of ROS and tested on a Raspberry Pi running an Ubuntu core (accessed over a private network).

This arrangement provides an opportunity to use the package for autonomous robot and remote radiation mapping applications, running the ROS operating system. Further edits are, therefore, made to the code so that the system posts detection events to ROS topics, allowing it to be integrated with other ROS packages for processing and mapping. This allows a spectrum to be constructed for 30 seconds over each of the 30 different positions taken by the robot.

Finally, note that the RadAngel was attached to the robot head via a 3D printed holder. The parts were created in SolidWorks 2019 CAD software before conversion to .STL files and further processing in slicer software Ultimaker Cura

4.4.1. They were printed on an Ultimaker 2+ with a 0.4 mm nozzle using 2.85 mm diameter black Polylactic Acid (PLA) filament (Ultimaker Tough PLA). Layer thickness was set to 0.1 mm, with 2 mm side walls and 50% infill.

VI. EXPERIMENTS AND RESULTS

A white cardboard box with dimensions of 161 mm tall, 132 mm wide and 72 mm deep was placed in view of the Kinect sensor. The box was tilted using a marker pen under one side of the box. A Cs-137 source with an activity of 309 kBq was placed on the box at a random location. The software algorithm was run as described above, initially producing a frame as in Figure 6, where the Canny edge detection algorithm in MATLAB was used to determine the edges depending on the colour interfaces.

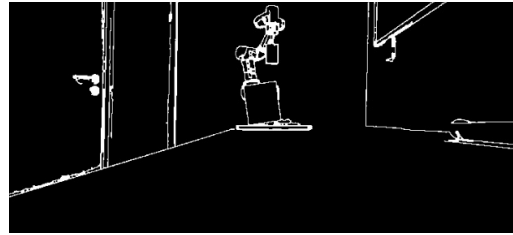


Figure 6: The Canny edge algorithm finds the edges.

Once the operator has clicked on the object of interest, the original RGB image is overlaid with the outline of this object (in red), as illustrated in Figure 7. The centre of the object is also determined and displayed (green circle) along with the upper corners of the object (blue and red crosses). These blue and red crosses represent the required start and finish locations of the radiation detector. Here, the system has been designed to record thirty second scans over thirty different positions evenly distributed between these start and finish locations. The values of scan time and number of scan locations can be changed to suit the individual needs of the operator.



Figure 7: The object is selected, and the centre is found, together with the locations of the start and finish scanning points.

In order to ensure the sensor does not physically impinge on the uncharacterised object, a 20 mm safety margin has been added within the software to ensure no contact between the sensor and the object occurs. Although this may be a trivial consideration here, contact between the detector and a potentially contaminated object would obviously not be favoured within a real-life radiological environment.

The RadAngel is operated via a separate computer running LINUX and thus is controlled manually – i.e. the operator is

required to run the python code when the robot arm begins in its first scanning position. The data acquired by the RadAngel is saved as a 4096 x 30 grid within a .csv file. Part of this file is displayed in Figure 8, with the spectra between channels 400 and 1200 (corresponding to energies of approximately 250 keV and 750 keV), displayed over the 30 positions taken up by the sensor. It can clearly be seen in Figure 8, that the spectra produced in positions 1-20 indicate a higher fluence rate than in positions 20-30, with a peak at around reading six.

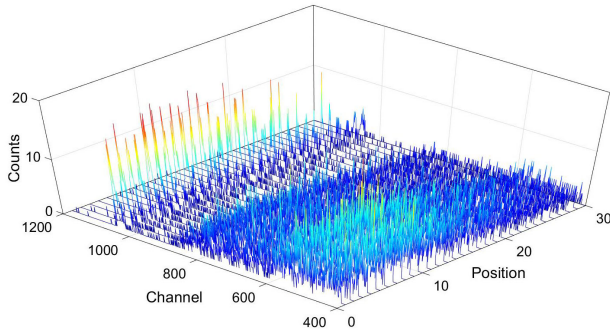


Figure 8: Cs-137 spectra across the top of the object selected by the user.

The experiment was repeated (Figure 9) using two smaller sources, a Cs-137 source with an activity of 24.7 kBq and a Co-60 source with an activity of 2.6 kBq. It should be noted that the gamma peaks of the Co-60 occur at energies of 1.17 MeV and 1.33 MeV corresponding to higher channel numbers than in the Cs-137 case. Here, despite the significantly lower fluence rate, the caesium (located close to position 6) can still be noticed compared to the rest of the positions.

The case of using the 24.7 kBq Cs-37 and 2.6 kBq Co-60 source was re-run using a smaller safety margin of 5 mm. The overall fluence increased as expected although the low activity of the Co-60 made it hard to validate the presence of that source. However, if only the channel data between 1200 and 4000 was utilised, then it was easier to verify the presence of Co-60 in around position number 24 (Figure 10). The Total counts over the 30 second time segments for each scenario are shown in Figure 11.

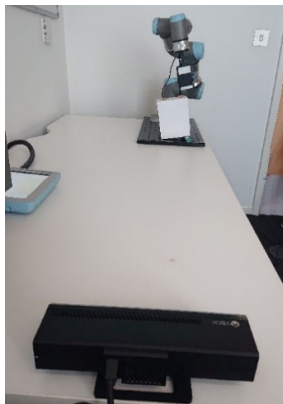


Figure 9: The MS Kinect and the robot holding the RadAngel detector and scanning.

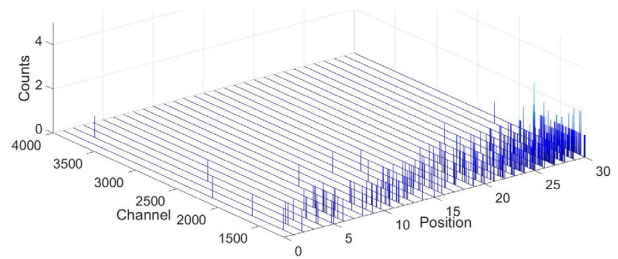


Figure 10: Counts for energies between approximately 750keV and 2.5 MeV.

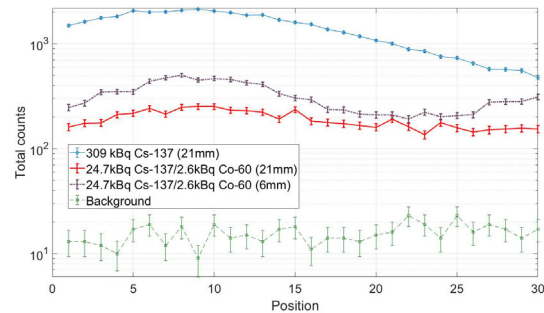


Figure 11: The total fluence recorded at every location.

VII. DISCUSSION AND CONCLUSIONS

A series of commercially available devices have been utilised to develop a novel radiological scanning system capable of inspecting uncharacterised objects for possible contamination. The software algorithm also developed here, enables the operator to see the radiological spectra shape over a user defined number of locations across the object. This potentially enables the determination of the type of contamination (i.e. which radioisotope) as well as the position within the object. Preliminary results reported above, indicate that the system can successfully autonomously scan the selected object.

To date, these results have only involved preferential, as opposed to realistic scenarios, although the intention is to expand the work into more complex and realistic work environments. In this regard, there are several limitations to the present system, which become obvious when any sub-optimal parameters are applied:

- In the experiments undertaken here, a standard box shape was used as the object in all experiments. This improved the reliability of the IR time of flight-based depth sensor, and offered a simple geometry which rendered it trivial to find the corners regardless of object orientation. If a non-standard object were to be used here, the algorithm applied would not operate correctly. A solution may be an algorithm which chooses start and stop points at the positions where the object ends, on the vectors at $+45^\circ$ and -45° relative to vertical.
- Another limitation is that the thickness of the object was pre-known (72 mm) and this was considered when programming the algorithm, i.e. the RadAngel would scan 36 mm behind the front edge of the object. However, if the object was 200 mm deep, the current system would have no way of knowing this, and the scan would occur again, 36 mm behind the front face. One way to overcome this limitation would be to install a second sensor located

orthogonally to the work area, with the depth data achieved from this.

- The range of the UR3 robot is limited to 500 mm, thus constraining the range possible with this system.
- Anything located behind the robot (relative to the vision sensor) is also not considered to be within the work area here, due to the robot itself obscuring the view.
- The Canny edge detection algorithm works here using the interface between colour segments. Hence, if the object colour roughly matches the background colour then the algorithm breaks down. Similarly, the work has assumed a monochromatic object where there are no colour interfaces within the object. A multi-coloured object would be expected to confuse the system and thus reduce effectiveness.

These limitations will be addressed in future research. Nonetheless, the preliminary system reported in this article, helps to demonstrate the concept i.e. the robotic control system autonomously scans the selected object and the software algorithms successfully identify the source.

ACKNOWLEDGMENTS

This work is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/R02572X/1.

REFERENCES

- [1] Kawatsuma, S., Mimura, R., & Asama, H. (2017). Unitization for portability of emergency response surveillance robot system: experiences and lessons learned from the deployment of the JAEA-3 emergency response robot at the Fukushima Daiichi Nuclear Power Plants. *Robomech Journal*, Vol. 4, No. 1, pp. 1-7.
- [2] Tsitsimpelis, I., Taylor, C. J., Lennox, B., & Joyce, M. J. (2019). A review of ground-based robotic systems for the characterization of nuclear environments. *Progress in Nuclear Energy*, Vol. 111, pp. 109-124.
- [3] irobot at Fukushima (accessed 11/02/20) <https://www.asianscientist.com/2011/04/topnews/remote-controlled-robots-investigate-fukushima-nuclear-plant/>
- [4] Kawatsuma, S., Fukushima, M. and Okada, T. (2012), "Emergency response by robots to Fukushima-Daiichi accident: summary and lessons learned", *Industrial Robot*, Vol. 39, No. 5, pp. 428-435.
- [5] Watanabe, H., Kikuchi, S., Narui, K., Toyoshima, K., Suzuki, H., Tokita, N., & Sakai, M. (2019). Japan's Fukushima Daiichi nuclear power plant accident. In *An International Perspective on Disasters and Children's Mental Health* (pp. 167-190). Springer, Cham.
- [6] Bandala, M., West, C., Monk, S., Montazeri, A. and Taylor, C.J. (2019) Vision-based assisted tele-operation of a dual-arm hydraulically actuated robot for pipe cutting and grasping in nuclear environments, *Robotics*, Vol. 8, No. 2, ID: 42, pp. 1-24.
- [7] West, C., Monk, S. D., Montazeri, A., & Taylor, C. J. (2018). A vision-based positioning system with inverse dead-zone control for dual-hydraulic manipulators. In *UKACC 12th International Conference on Control (CONTROL)* (pp. 379-384). IEEE.
- [8] West, C., Montazeri, A., Monk, S. D., & Taylor, C. J. (2016). A genetic algorithm approach for parameter optimization of a 7DOF robotic manipulator. *IFAC-PapersOnLine*, Vol. 49, No. 12, pp. 1261-1266.
- [9] Montazeri, A., West, C., Monk, S. D., & Taylor, C. J. (2017). Dynamic modelling and parameter estimation of a hydraulic robot manipulator using a multi-objective genetic algorithm. *International Journal of Control*, Vol. 90, No. 4, pp. 661-683.
- [10] West, C., Montazeri, A., Monk, S. D., Duda, D., & Taylor, C. J. (2017). A new approach to improve the parameter estimation accuracy in robotic manipulators using a multi-objective output error identification technique. In *26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (pp. 1406-1411). IEEE.
- [11] Marturi, N., Rastegarpanah, A., Takahashi, C., Adjigble, M., Stolkin, R., Zurek, S., ... & Bekiroglu, Y. (2016). Towards advanced robotic manipulation for nuclear decommissioning: A pilot study on tele-operation and autonomy. In *International Conference on Robotics and Automation for Humanitarian Applications (RAHA)* (pp. 1-8). IEEE.
- [12] Nancekievill, M., Watson, S., Green, P. R., & Lennox, B. (2016). Radiation tolerance of commercial-off-the-shelf components deployed in an underground nuclear decommissioning embedded system. In *IEEE Radiation Effects Data Workshop (REDW)* (pp. 1-5). IEEE.
- [13] Amon, C., Fuhrmann, F., & Graf, F. (2014). Evaluation of the spatial resolution accuracy of the face tracking system for Kinect for windows v1 and v2. In *Proceedings of the 6th Congress of the Alps Adria Acoustics Association* (pp. 16-17). Australian alps.
- [14] Lun, R., & Zhao, W. (2015). A survey of applications and human motion recognition with microsoft Kinect. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 29, No. 5, 1555008.
- [15] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 6, pp. 679-698.
- [16] Robotics toolbox for MATLAB (accessed 13/02/20) <http://petercorke.com/wordpress/toolboxes/robotics-toolbox>
- [17] RadAngel (accessed 04/02/20) <https://www.kromek.com/product/radangel-czt-gamma-ray-spectrometer/>
- [18] Python software (accessed 19/02/20) <https://github.com/thinrope/kromek-PROTO>