Chapter 4

Effects of ionizing radiation on robotic trajectory movement and electronic components

4.1 Overview

This chapter focuses on analyzing the impact of radiation (0.09/0.18/1.5 Gy/s) on various electronic components, such as microcontrollers, servomotors, and digital temperature sensors, as well as the movement of a 3-DoF robotic arm over time. In addition, this study examines the impact of radiation on the trajectories of two robotic arms, examining the durability and performance of these components, employing different strategies from the literature to compare radiation effects such as lead shielding and component hardening. Furthermore, the radiation effects on memory Electronic Non-volatile Computer Memory storage (FLASH), Electrically Erasable Programmable Read-Only Memory (EEPROM) and Static Random Access Memory (SRAM) were evaluated. In addition, the robot's durability and degradation were evaluated while managing a robotic arm. The findings offer critical insights for the design and implementation of robotics in hostile environments. The experimental tests were conducted using a linear particle accelerator at Center for Energy, Environmental, and Technological Research (CIEMAT) in Spain.

4.2 Methodology

Three different experiments were performed to examine the effects of ionizing radiation on passive robotic components, including 8-bit and 32-bit RISC microcontrollers [296], digital temperature sensors, and servomotors as shown in Fig. 4.1. The following microcontrollers were utilized: Atmega328P (Arduino Uno) featuring a Dual in-line Package (DIP), ATmega64M1 and ATmegaS64M1 Rad-Hard, both housed in a Quad-flat Package (TQFP), and an RP2040 (Raspberry Pico) with a Micro Leadframe or Quad-flat no-leads (MLF/VQFN). For position control testing, servomotors such as the Parallax Standard Servo 900-00005, micro servo SG90, and Grove micro servo 316010005 were used. For velocity control testing, the Hitech HS-55 was used. Furthermore, temperature sensors included the LM35 with a TO-92 package and an integrated temperature sensor within the Rad-Hard microcontroller.


Atmega328P
AtmegaS64M1
Atmega64M1
RP2040
Parallax
SG90
Grove
LM35

Figure 4.1: Devices exposed to ionizing radiation with and without shielding. Includes 8-bit and 32-bit RISC microcontrollers, servomotors, and digital temperature sensors.

A trio of experiments was conducted to investigate and understand the impact of ionizing radiation on robotic components, both passive (e.g., microcontrollers with RISC architecture [296], digital temperature sensors) and active (e.g., servomotors).

- Experiment #1 Evaluation of two microcontrollers, servomotors, and temperature sensors incorporating mitigation strategies in ionizing radiation conditions.
- Experiment #2 Analisys of robotic trajectories with a low radiation dose utilizing a Non-Rad-Hard microcontroller (ATmega64M1).
- Experiment #3 Comparison of robotic trajectories using Non-Rad-Hard Raspberry Pico (RP2040) and Arduino Uno (ATmega328P) microcontrollers in radiation environments.
- Experiment #4 Comparison of robotic trajectories using Non-Rad-Hard (ATmega64M1) and Rad-Hard (ATmegaS64M1) microcontrollers in radiation environments.

In the first experiment, two approaches to radiation mitigation were evaluated: using a radiation-hardened microcontroller and implementing lead shielding. For their cost-efficiency, general use, and market availability, two Arduino Uno boards (one shielded, the other unshielded) were chosen. Additionally, a custom Printed Circuit Board (PCB) incorporating the ATmegaS64M1, a commercially available Rad-Hard microcontroller, served as a benchmark for radiation resistance up to a Total Ionizing Dose (TID) of 30 Krad. Small Commercial Off-The-Shelf (COTS) servo motors and the LM35 temperature sensor were selected for their space-saving design, cost-effectiveness, and precision. In the second and third experiments, the impact of microcontroller type and radiation dose on the robotic arm's trajectories was investigated. The ATmega64M1 was chosen for its similarity to the ATmegaS64M1 (Rad-Hard) and its cheaper cost; both microcontrollers are available in a thin quad-flat pack format (TQFP). The custom-designed 3-DoF robotic arms were designed to match the dimensions of the accelerator target. Concerning radiation exposure, doses were determined based on the availability and capabilities of the accelerator facility.

4.2.1 Experiment #1

The experiment #1 investigates the performance of microcontrollers during the generation of Pulse Width Modulation (PWM) signals to control servomotors and measures environmental temperature using two digital sensors via the Analog Digital Converter (ADC) ports: the first one, an external digital temperature sensor and the second one an integrated temperature sensor in the Rad-Hard microcontroller.

Fig. 4.2 shows a diagram of the main components of this experiment, which are as follows:

- Control room with the master microprocessor and microPC for saving data and the operator interface.
- Accelerator area with three microcontrollers under different conditions:
  - Uc1: ATmega328P (Arduino Uno) configured with a 16 MHz primary clock speed, 5 VDC power source, 5 VDC 10-bit ADC reference voltage operating at 125 kHz, and 50 Hz PWM ; exposed directly to radiation without any shielding. This microcontroller operates two servomotors (one without protection and one within a 2mm PB lead enclosure) and connects to two unshielded external temperature sensors.
  - Uc2: ATmega328P (Arduino Uno), identical set-up to Uc1 but enclosed in a 2 mm PB lead box for radiation shielding. This unit produces PWM signals for two servomotors (one unshielded and one within a 2mm PB lead enclosure) and links to two external temperature sensors (one unprotected and one housed within a 2mm PB lead enclosure).
  - Uc3: ATmegaS64M1 configured with an 8 MHz main clock speed, 3.3 VDC power supply, 2.56 VDC 10-bit ADC reference voltage at 125 kHz, and 50 Hz PWM ; it is a radiation-hardened microcontroller connected to a servomotor without any shielding and two temperature sensors (one integrated and one external).

4.2.2 Experiment #2

Experiment #2 aims to evaluate the performance of a robot using a Non-Rad-Hard microcontroller when exposed to radiation. This setup employs a 3-DoF robotic configuration (position micro servo SG90), controlled via PWM signals generated by the ATmega64M1 microcontroller (8 MHz primary clock frequency, 3.3 VDC main power supply, 2.56 VDC ADC reference voltage at 125 kHz, and 50 Hz PWM). An integrated temperature sensor reports the microcontroller temperature in ATmega64M1. Figures 4.3 depict the principal components of this experiment, which include: The control room housing the master microprocessor and micro-PC for data storage and operator interface.The accelerator area housing a 3-DoF robotic arm with a Non-Rad-Hard microcontroller.

To develop the robot's trajectory, the kinematics of the robot were initially analyzed using the MATLAB tool created by Peter Corke [297]. This was done to determine the workspace

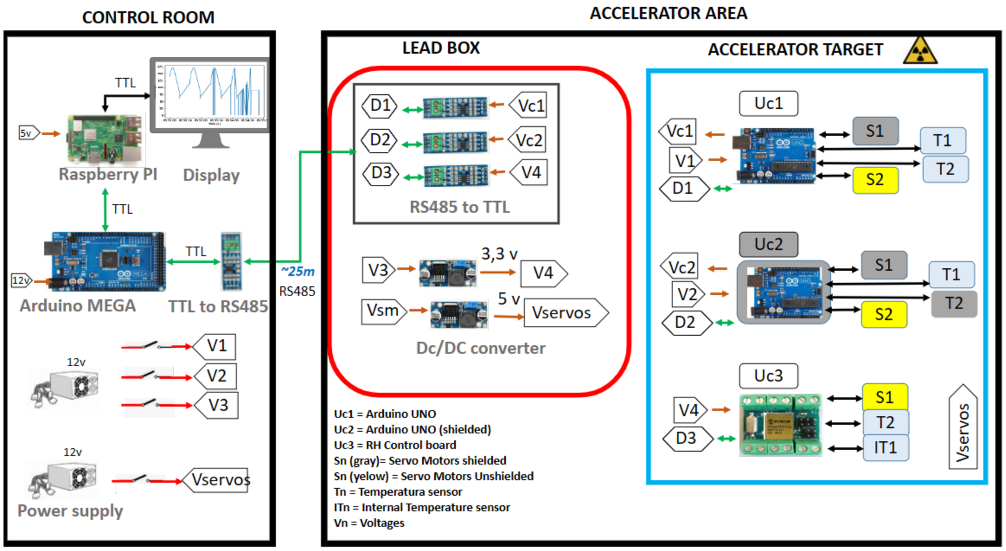


Figure 4.2: Setup for Experiment 1. The irradiated section contains the following key components: a non-protected ATmega328P (Uc1), a radiation-shielded ATmega328P (Uc2), and an ATmegaS64M1 (Uc3). These microcontrollers are responsible for managing several servo motors and temperature sensors, denoted as Sn and Tn.

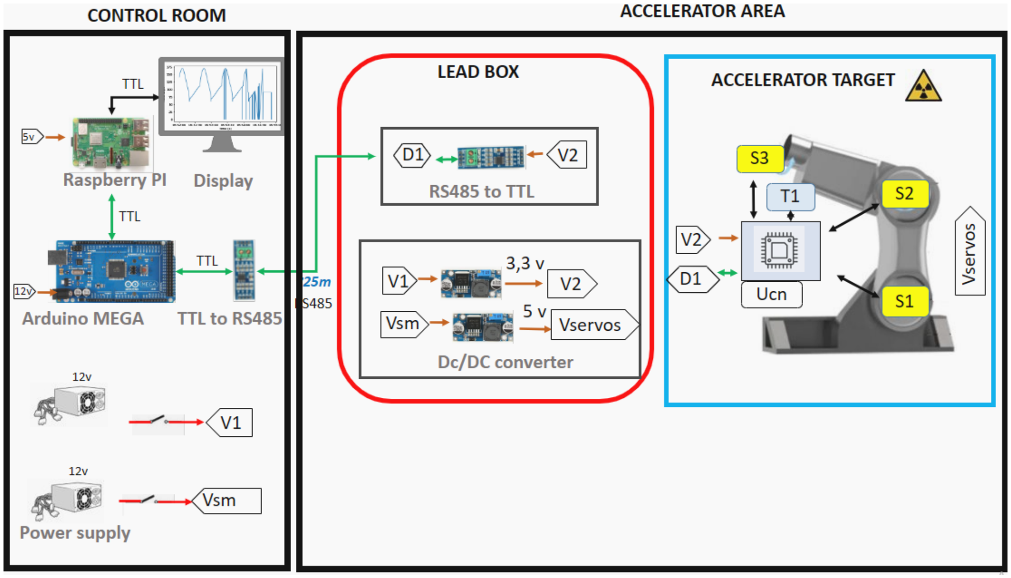
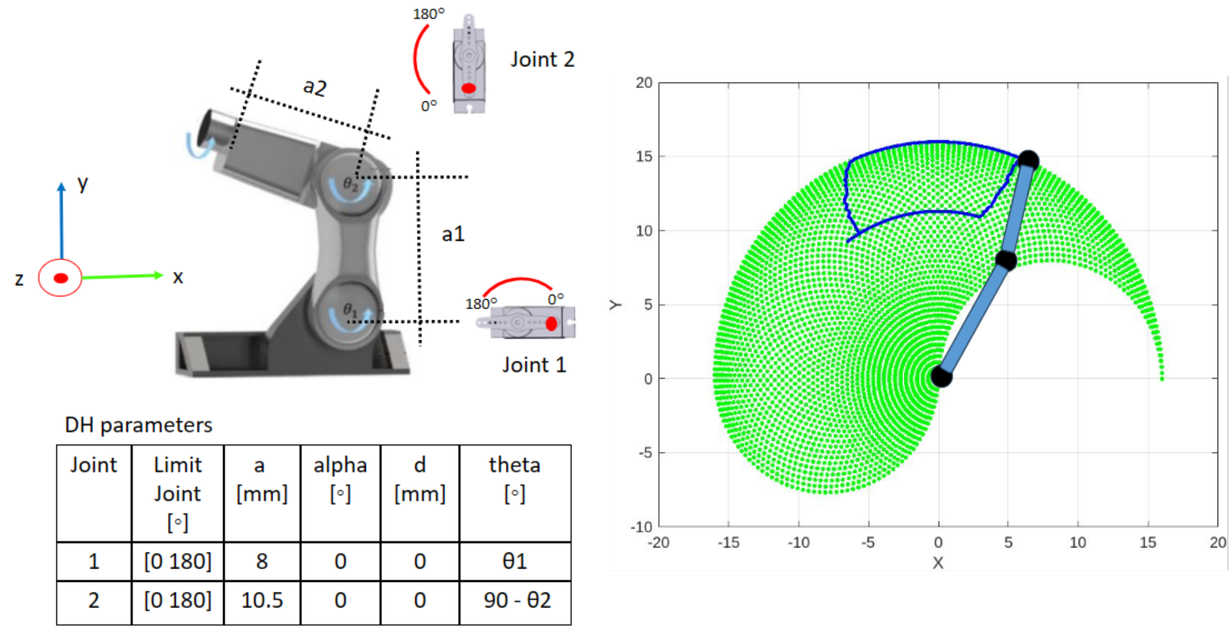


Figure 4.3: Configuration for Experiment #2. The control room has an identical layout to that of Experiment #1, and there is a 3-DoF robotic arm featuring rotational joints situated in the accelerator zone.

based on the robot's kinematics, including its dimensions and the movement limitations of the servomotors. The DH matrix was utilized to construct the robot and to compute the forward kinematics as described in Eq. 4.1, along with the inverse kinematics. Subsequently, a trajectory within the workspace in x, y coordinates was chosen, and the angles were calculated using MATLAB's inverse kinematics tool. Fig. 4.4 illustrates the robot's layout and the placement of the servomotors, along with the DH parameters showing the servomotor limits, the workspace (green) and the selected trajectory (blue).

$$\begin{aligned} x &= a1\cos(q_1) + a2\cos(q_1+q_2) \\ y &= a1\sin(q_1) + a2\sin(q_1+q_2) \end{aligned} \tag{4.1}$$



DH parameters

Joint	Limit Joint [◦]	a [mm]	alpha [◦]	d [mm]	theta [◦]
1	[0 180]	8	0	0	θ1
2	[0 180]	10.5	0	0	90 - θ2

Figure 4.4: A 2-DoF robot designed for experiment #2. This depicts the robot's configuration and its DH matrix. Additionally, It shows the workspace determined by the robot's kinematics (green) and the chosen trajectory (blue).

4.2.3 Experiment #3

Two microcontrollers were used in the experiment #3: Raspberry Pico (RP2040) and Arduino Uno (ATmega328P). RP2040 operates at 125MHz primary clock frequency, a 3.3 VDC main power supply, a 3.3 VDC 12-bit ADC reference voltage at 500 kHz, and 50 Hz PWM. The ATmega328P operates with an 8 MHz primary clock frequency, a 3.3 VDC main power supply, a 2.56 VDC 10-bit ADC reference voltage at 125 kHz, and a 50 Hz PWM at a 120 kHz clock. The communication structure and setup mirror those of the previous experiment #2, depicted in Fig. 4.5. In contrast, this experiment #3 uses a different 3-DoF robotic arm and trajectory, illustrated in Fig. 4.6. The servomotors used were SG90 controlled by position commands.

To select a trajectory for testing, the methodology described in the section below was used. The DH matrix was identified and used to build the kinematics with Matlab. Eq. 4.2 describes the forward kinematics. Fig. 4.6 illustrates the robot configuration, the DH parameters, the workspace (in green) due to robot constraints, and the selected trajectory (in blue).

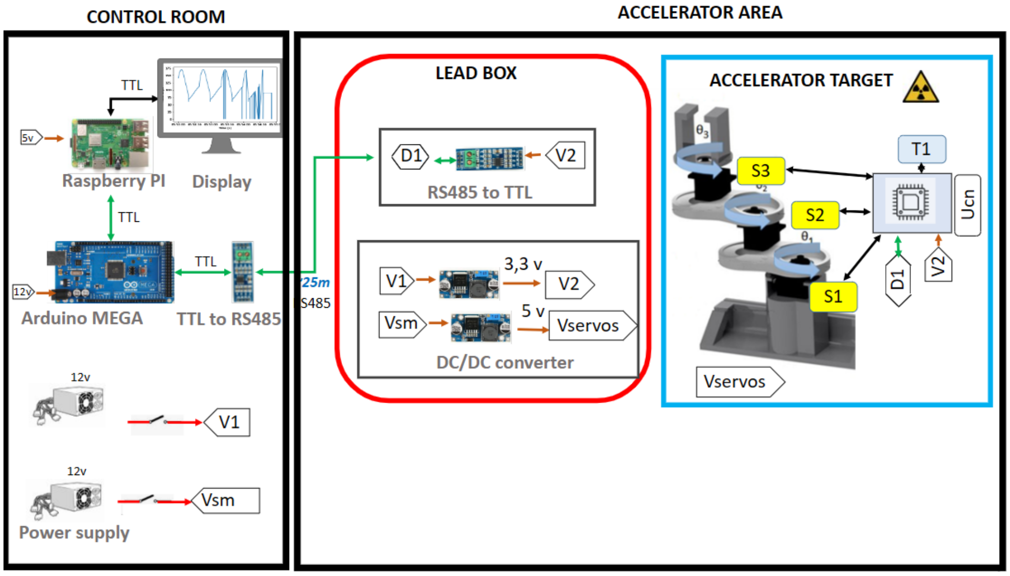
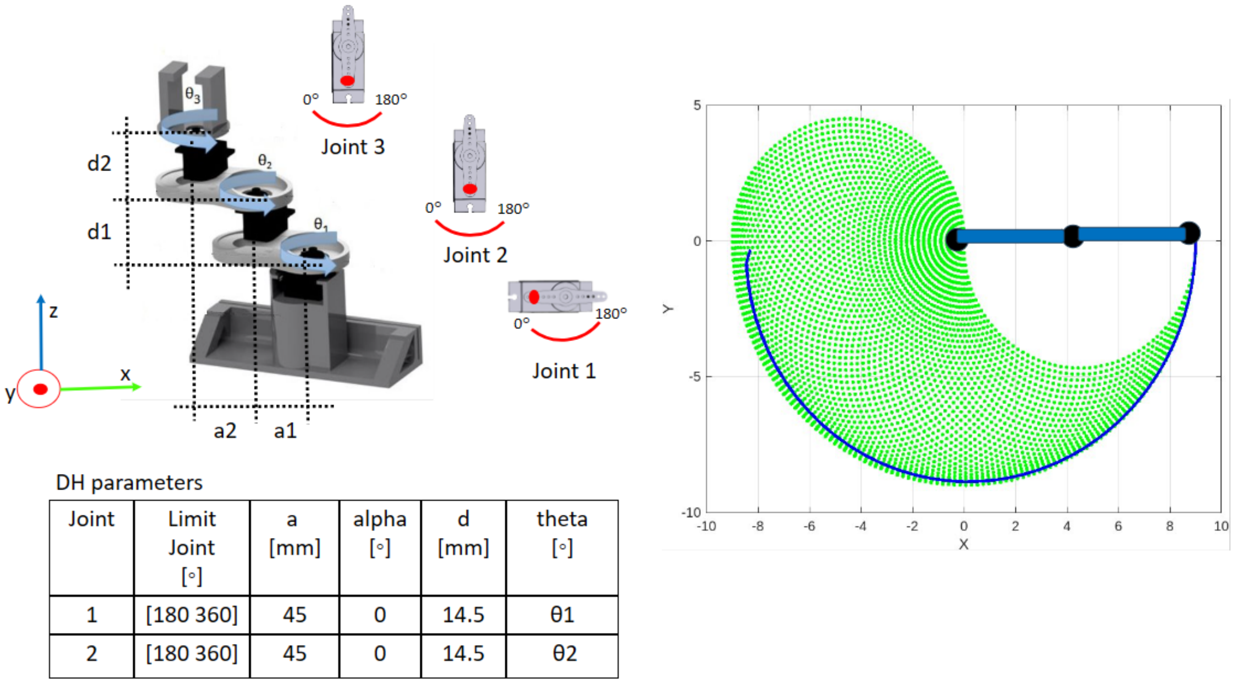


Figure 4.5: Configuration for Experiment #3, #4. The control room has an identical layout to that of Experiment #2, and there is a 3-DoF robotic arm featuring rotational joints situated in the accelerator zone.



DH parameters

Joint	Limit Joint [°]	a [mm]	alpha [°]	d [mm]	theta [°]
1	[180 360]	45	0	14.5	θ1
2	[180 360]	45	0	14.5	θ2

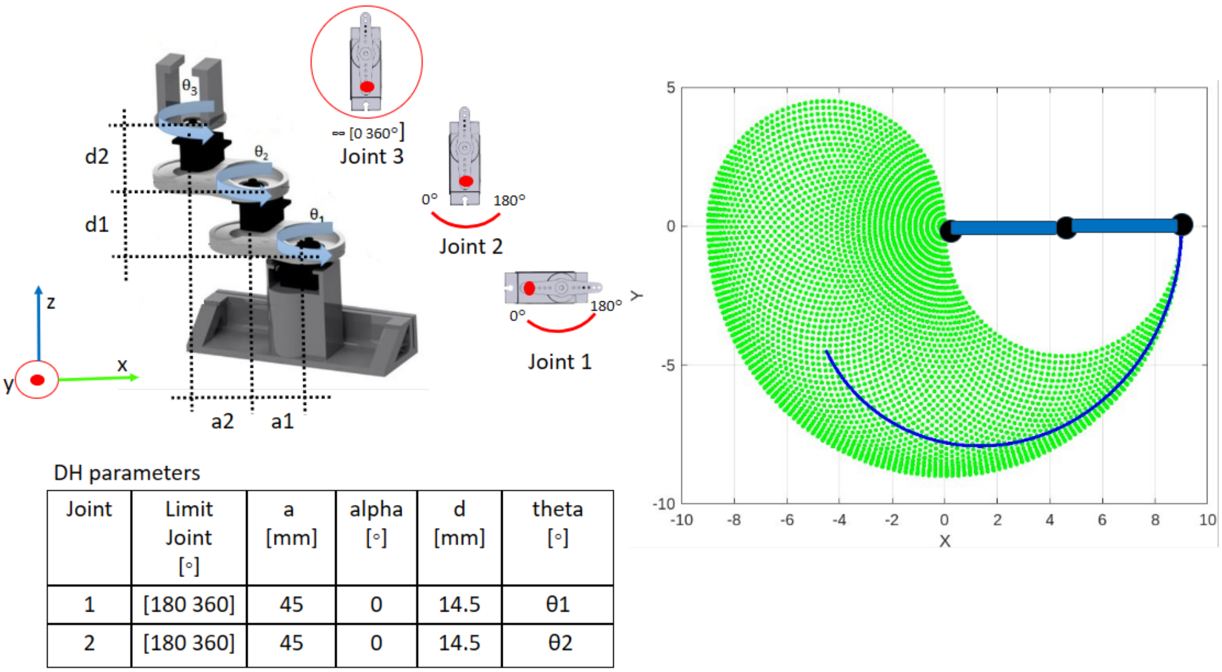
Figure 4.6: A 3-DoF robot designed for experiment #3. This depicts the robot's configuration and its DH matrix. Additionally, It shows the workspace determined by the robot's kinematics (green) and the chosen trajectory (blue).

$$
\begin{aligned}
x &= a1\cos(q_1) + a2\cos(q_1 + q_2) \\
y &= a1\sin(q_1) + a2\sin(q_1 + q_2) \\
z &= d1 + d2
\end{aligned}
\tag{4.2}
$$

4.2.4 Experiment #4

The experiment #4 utilized two microcontrollers: the ATmega64M1 (standard) and the ATmegaS64M1 (Rad-Hard). Both run on an 8 MHz primary clock, powered by a 3.3 VDC main supply, 2.56 VDC ADC reference voltage at 125 kHz, and PWM of 50 Hz. The structure and communication configuration are identical to those in the previous experiment #3, shown in Fig. 4.5. In contrast, this experiment #4 employs different servomotors for the same 3-DoF robotic arm, as shown in Fig. 4.7. Joints 1 and 2 use position servomotors (Grove Servo 316010005) while Joint 3 uses velocity servomotors (Hitech HS-55). Additionally, after completing the test, the FLASH memory data was retrieved to examine any possible byte modifications.

The methodology detailed in the above experiments was used to choose a trajectory for testing. Fig. 4.7 shows the robot configuration, the DH parameters, the workspace (in green) constrained by the robot, and the selected trajectory (in blue).



DH parameters

Joint	Limit Joint [°]	a [mm]	alpha [°]	d [mm]	theta [°]
1	[180 360]	45	0	14.5	θ1
2	[180 360]	45	0	14.5	θ2

Figure 4.7: A 3-DoF robot designed for experiment #4. This depicts the robot's configuration and its DH matrix. Additionally, It shows the workspace determined by the robot's kinematics (green) and the chosen trajectory (blue).

4.2.5 Error Metric

Absolute Trajectory Error (ATE) is relevant for algorithms that estimate robot trajectories [298]. It quantifies the discrepancy between the actual position (ground-truth) and the robot's estimated position at corresponding time instances [299, 300], as shown in Eq. (4.3). ATE was employed to determine the error trayectories in the experiments described in this article, where:

- T_i, is the magnitude of the Euclidean distance along the horizontal plane between the estimated and ground-truth poses at frame i.
- n, number of frames.

$$ATE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} T_i^2} \tag{4.3}$$

In our work, the ATE was calculated for each cycle or lap. The global ATE then corresponds to the mean square value of all the ATE cycles calculated during the experiment.

4.3 Experimental setup

4.3.1 Radiation Facility

The CIEMAT facilities [301] have a high-energy electron accelerator (Van de Graaff) which was used for all the tests in this work. The accelerator had currents of up to 150 uA and energies of up to 2 MeV. The samples were placed directly at the target location to ensure homogeneous radiation and receive direct impact from the ionizing electron beam. The electronic components were situated over a 127 × 200 mm aluminum sheet, which was as close as possible to the emission point of the Gaussian distribution of electrons. Figure. 4.8 shows a general and detailed view of the facility used during the experiments.

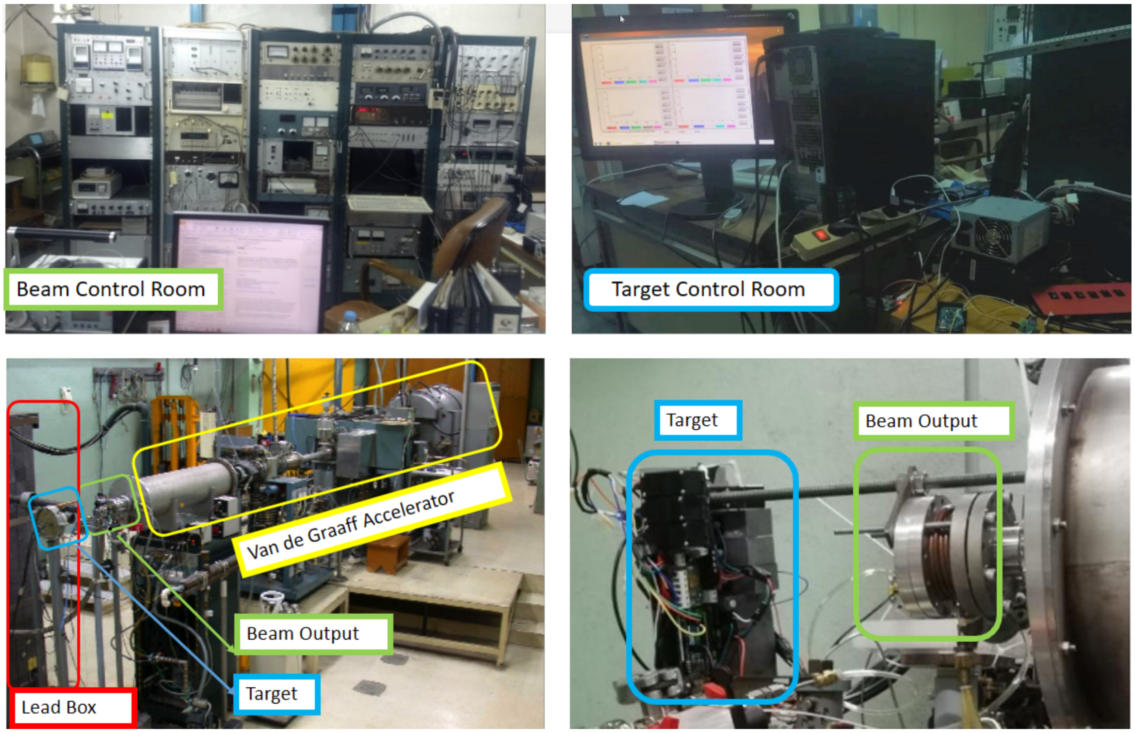


Figure 4.8: Left-hand, Van de Graaff electron accelerator used for the experiments, Right-hand, zoom view of the target and beam output.

4.3.2 Testing set-up

A master-slave network was established on an RS-485 serial line at a rate of 19,200 bauds to facilitate communication between the control room and the accelerator target. The master controller, located in the control room outside the radiation area, sent movement commands to the microcontrollers in the radiation area, which acted as slaves. Figure 4.9 illustrates the communication architecture used for the experiments. The master received information related to the robot's actual position and the status of the irradiated microcontrollers, which was then sent to a data logger to be stored and analyzed.

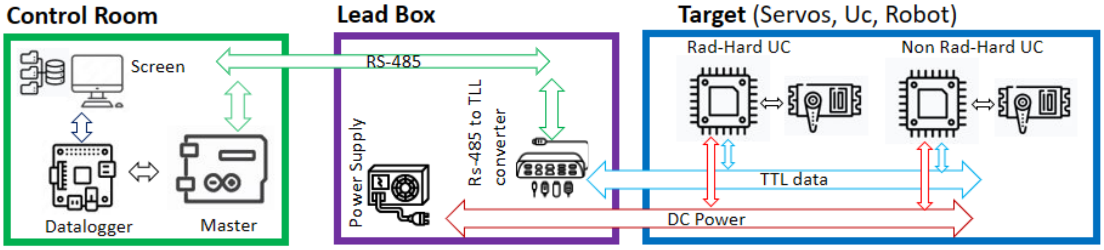
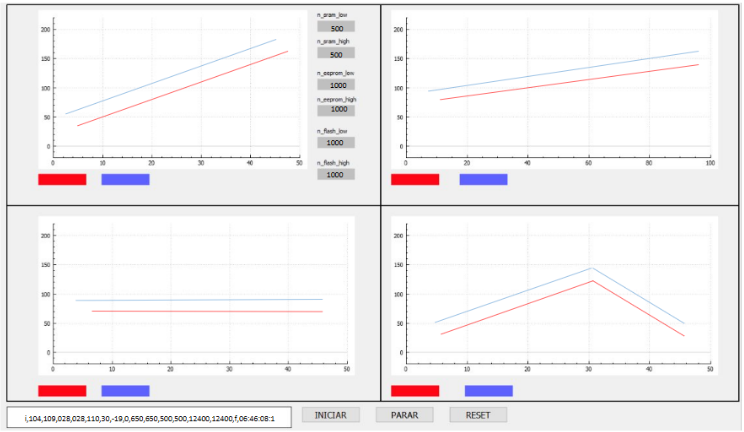


Figure 4.9: The master microcontroller acts as the main point of communication and sends data to a micro-PC-controlled data logger (Raspberry PI). The microcontrollers located in the target work as slaves and process the command received to execute the servomotor movement and send the feedback information to the master microcontroller.

The master unit transmits movement commands for the robot to the slave units, specifying the desired final position. These commands are processed by slaves to create Pulse Width Modulation (PWM) signals for the servomotors, facilitating the corresponding robot joint movements. The discrepancy between the desired robot position and the actual position is analyzed to study the radiation impact on the slave microcontrollers. The actual position of the robot is measured using joint potentiometers, and the temperature of the microcontrollers is collected from digital temperature sensors through an Analog-to-Digital Converter (ADC). All of these data are sent back to the master, which logs it and forwards it to a micro-PC (Raspberry PI). This microcomputer shows the data on a graphic user interface (GUI), enabling the operator to oversee the experiments as shown in Fig. 4.10.

The main program is capable of identifying two types of system errors: resets from the slave microcontroller and communication faults. Resets in the microcontroller slaves occur due to various issues, such as watchdog timer failures or power faults. Communication faults between the master and slave generally occur after multiple resets in the slave microcontroller. The error analysis has been enhanced with a dedicated program designed to detect memory errors in irradiated integrated circuits. For example, when a single particle alters a binary digit in the memory, this is know like SEE occurs (SEE are caused by a single particle that deposits energy in the electronic device, leading to soft or hard errors) [139]. This application relies on an Error Detection and Correction (EDAC) program [302, 303] that reads and writes the content of EEPROM, SRAM, and FLASH microcontroller memories. The program continuously monitors memory and reports any detected changes to the operator for further analysis and decision-making.



n_sram_low 500
n_sram_high 500
n_eeprom_low 1000
n_eeprom_high 1000
n_flash_low 1000
n_flash_high 1000

```
i,104,109,028,028,110,30,-19,0,650,650,500,500,12400,12400,f,06:46:08:1
```

Figure 4.10: GUI designed to show the setpoint sent to the servomotors (blue) and the feedback position (red). It also displays the count of logical ones and zeros in the microcontrollers' memories and all serial data receive by the custom protocol developed. Additionally, it includes buttons to start/stop communication and to software reset all slave devices.

Moreover, a system has been developed to remotely reset microcontrollers either via hardware or software. This system enables the program to restart from the beginning and reconfigure the SRAM and EEPROM according to the identified error. Rewriting the Flash Memory is achievable only by loading the program into the microcontroller, an operation that can be performed when access to the accelerator is available at the conclusion of the trials.

4.4 Results

4.4.1 Experiment #1

The configuration for the experiment also incorporates extra components for power supplies and signal transformation. Fig. 4.11 shows an image of microcontrollers, servomotors, temperature sensors, and PB enclosures used in the irradiated zone.

In this experiment, a gamma radiation flow of 0.18 Gy/s was maintained for 29 minutes before increasing to 1.5 Gy / s until 38 minutes. Results are presented in Fig. 4.12 and Tab. 4.1. The Uc1 microcontroller (ATmega328P without protection) failed due to a hard error after 26 minutes and 34 seconds, reflecting a TID of 286.92 Gy. The Uc2 microcontroller (ATmega328P with PB radiation protection) operated correctly throughout the 38 minutes of the experiment. It tolerated more than 1k Gy of cumulative radiation dose, with partial retention thanks to the PB shielding. The Uc3 with a Rad-Hard microcontroller (AtmegaS64M1) operated for 28 minutes and 6 seconds, corresponding to a TID of 303.48 Gy, before stopping operation due to persistent reset signals, such as Brown-out Reset or WatchDog, designed to prevent malfunction.

The operator attempted to fix the malfunctions in Uc1 and Uc3 using both software and hardware resets, but was unable to restore normal microcontroller function. Furthermore, the memories of the three microcontrollers (SRAM, EEPROM, and FLASH) were examined after retrieving the microcontrollers and recovering the data from the FLASH memory, but no alterations were observed under these experimental conditions.

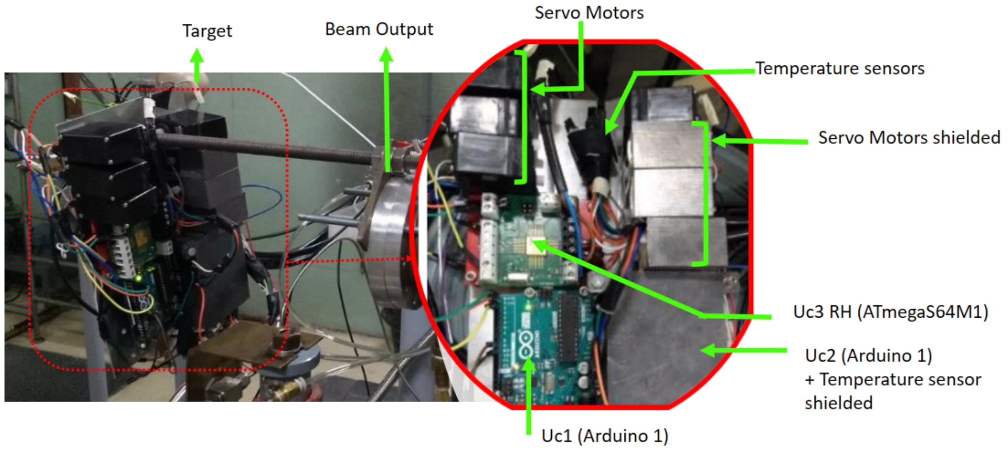


Figure 4.11: Accelerator target with irradiated components in experiment #1.

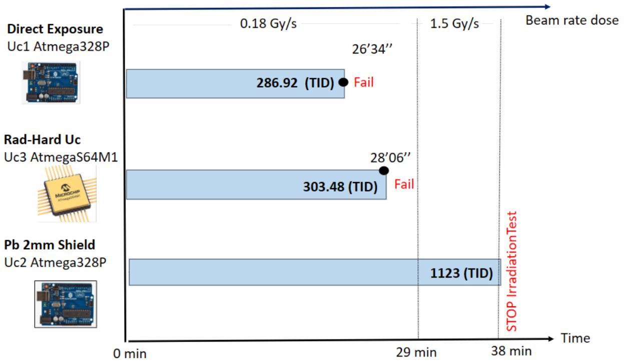


Figure 4.12: Time duration and beam rate dose used in experiment #1.

Table 4.1: Experiment #1. Dose, TID, ATE error, and time working of microcontrollers. Uc1 (ATmega328P without protection), Uc2 (ATmega328P PB protection), Uc3 (ATmegaS64M1 Rad-Hard).

Control Board	**Dose [Gy/s]**	**TID [Grays]**	**Time [min]**	**ATE servos [degrees]**	
				shielded	**unshielded**
Uc1	0.18	286.92	26'34"	0.28	0.29
Uc2	0.18/1.50	1123.00	>38'	0.06	0.10
Uc3	0.18	303.48	28'06"	—	0.10

The evolution and response of the servomotors have also been evaluated. Fig. 4.13 shows the evolution of the position of slave microcontrollers when starting the radiation and before detecting a malfunction (Uc1 and Uc3) or stopping the accelerator (Uc2). In particular, Fig.

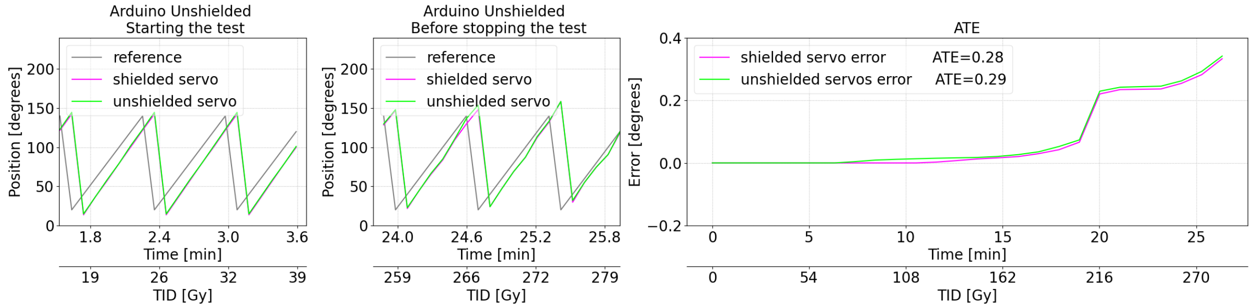
4.13(a) corresponds to the CB1 Arduino UNO without protection. Shows distorted positions on both servo motors before stopping the test due to microcontroller failure. Fig. 4.13(b) corresponds to Uc2 Arduino Uno with protection from PB. The error position is derived from the electronics and the operation of the servomotors. The position values at the end of the test between both servomotors have shown few variations. Therefore, we assume that the servos may be affected by the radiation dose in this experiment. Fig. 4.13(c) corresponding to Uc3 Rad-Hard ATmegaS64M1 shows a minimal distortion in the position values compared to Uc1. Although the microcontroller presents unknown reset faults, it can be observed that the degradation produced by the received dose affects less in the program's operation, and minimal changes exist in the trajectory.

In relation to the behavior of the temperature sensor, an increase in temperature readings was observed during the trials; refer to Fig. 4.15. The results for the Uc1 temperature sensors are illustrated in Fig. 4.15(a). The temperature values were comparable for both the unshielded and shielded sensors, with a notable rise occurring 13 minutes before the microcontroller's failure. This phenomenon indicates electronic degradation of the temperature sensor and could serve as an early warning system before microcontrollers lose communication. Furthermore, we observed that the temperature readings remained constant when the accelerator was stopped and increased again once the irradiation continued.

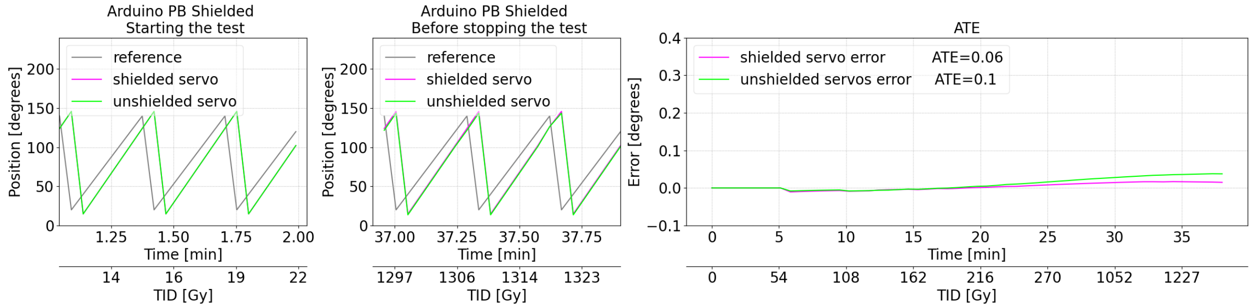
In the case of the Uc2 microcontroller, higher temperature readings were observed for the sensor placed outside the shield; refer to Fig. 4.15(b). A notable increase was detected after the radiation intensity was increased to 1.5 Gy/s at minute 26. The readings reached saturation, indicating they matched the sensor's maximum temperature value. The degradation of the sensor's internal structure results in the DAC circuit of the sensor outputting its highest possible voltage. Consequently, the sensor becomes permanently damaged, maintaining the same voltage even when the accelerator is stopped. In contrast, the temperature sensor with the PB shield did not exhibit any significant deviation from its initial readings.

Fig. 4.15(c) illustrates the temperature readings from external sensors associated with Uc3. The recorded temperatures for Uc3 are lower than those for Uc1 under the same radiation exposure. For example, after 25 minutes of irradiation, the Uc3 temperature sensor reads 25 °C, while the Uc1 sensor shows 42°C. However, the internal temperature sensor of the microcontroller remains relatively stable. Consequently, the Rad-Hard microcontroller minimizes cumulative errors when reading digital temperature sensors.

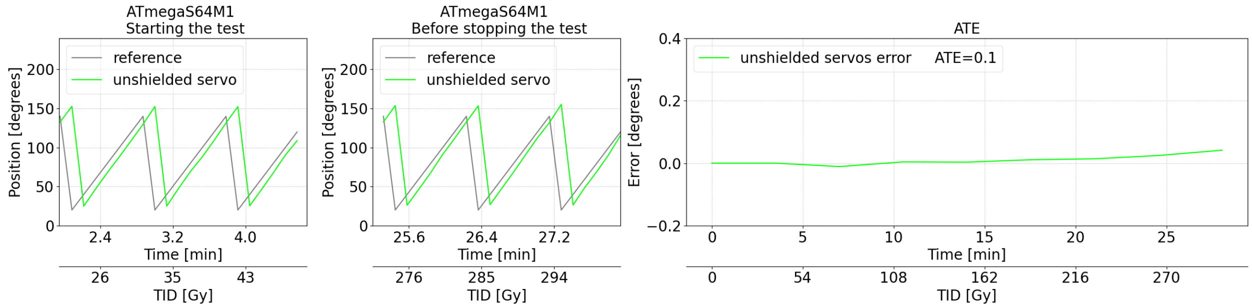
In summary, Fig. 4.14 shows the ATE of servomotors evaluated in the experiment #1. The analysis shows that Uc1 (Arduino Unshielded) ATE is similar in shielded and unshielded servomotors, although ATE should be lower in the shielded servomotor. Therefore, the error source is due to microcontroller functionality. Uc2 (Arduino shielded) ATE error shows that the unshielded servomotor differs slightly from the shielded one. This can be observed from minute 20 with 216 Gy of accumulated radiation. In this case, radiation affectation is on the servomotors. Uc3 (ATmegaS64M1 Rad-Hard) shows a slight increase. The ATE value of 0.1 can be compared with that obtained by the unshielded servomotor of the Uc2 test, so in this case the Uc2 stops the radiation, and most of the error comes from the servomotor. Fig. 4.16 shows the response for all temperature sensors. During the trials, temperature sensors showed increased readings under radiation exposure, indicating potential electronic degradation. For



(a) Servo motor positions of an unprotected Arduino Uno (Uc1: dose rate of 0.18 Gy/s for 26.34 minutes, accumulating a total ionizing dose of 286.92 Gy). The average total error (ATE) is equivalent in both shielded and unshielded servo motors (right image), despite the ATE being expected to be lower in the shielded servo motor. This discrepancy derives from the functionality of the microcontroller.



(b) Servomotors positions of an Arduino Uno with PB shielding (Uc2: exposure of 0.18 Gy/s for 29 minutes and 1.5 Gy/s for 9 minutes, totaling 1123 Gy). The ATE error indicates a slight difference between the unshielded and shielded servomotors. This effect becomes noticeable from minute 20, at which point the accumulated radiation reaches 216 Gy. In this scenario, radiation exposure leads to ATE in the servomotors.



(c) Positions of servomotors for the ATmegaS64M1 Rad-Hard (CB3: received a dose of 0.18 Gy/s over 28.06 minutes, with a total ionizing dose of 303.48 Gy). The ATE indicates a minor increase. An ATE value of 0.1 can be compared with that of the unshielded Uc2 servomotor.

Figure 4.13: Experiment #1. Servomotors position response in different microcontrollers Uc1, Uc2, and Uc3 with their corresponding time and TID. **Left graph:** positions at the beginning of the tests. **Middle graph:** positions until the microcontrollers or accelerator stop working. **Right graph:** error positions in ATE metric during the tests.

Uc1, both shielded and unshielded sensors showed a rise in temperature 13 minutes before the microcontroller failure, suggesting its use as an early warning system. In Uc2, unshielded sensors showed higher readings, reaching saturation after the radiation intensity increased to

1.5 Gy/s, leading to permanent sensor damage. In contrast, the shielded sensor remained stable. Uc3 sensors recorded lower temperatures compared to Uc1 under similar conditions, demonstrating the effectiveness of the Rad-Hard microcontroller in reducing cumulative errors in temperature readings.

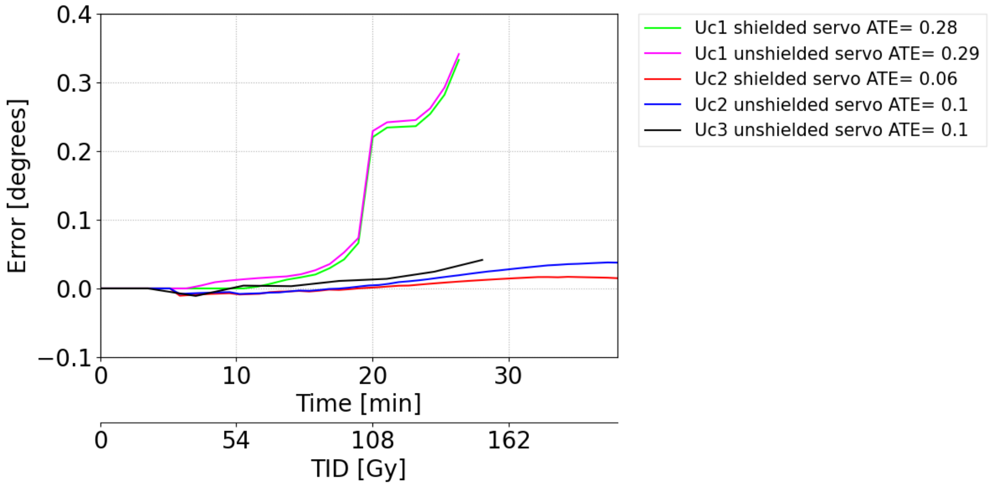
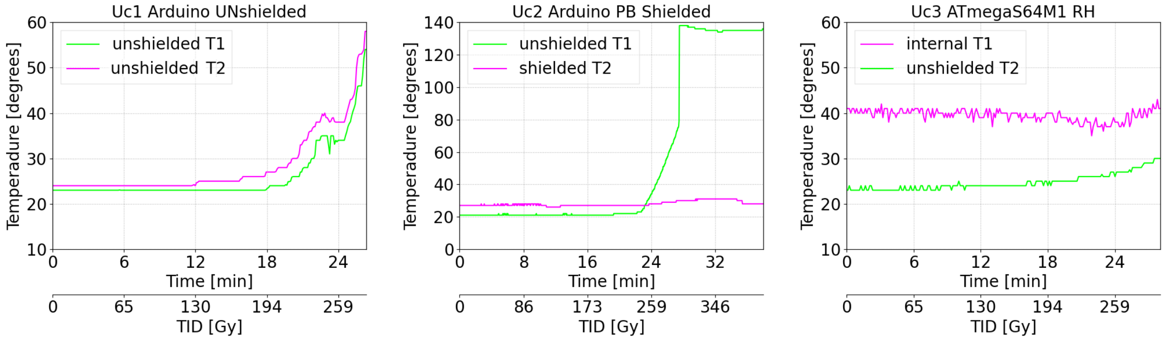


Figure 4.14: ATE comparison between the microcontroller and servos used in experiment #1. Uc1 (Arduino Unshielded) shows similar ATE in both shielded and unshielded servomotors, indicating the microcontroller as the error source. Uc2 (Arduino Shielded) shows a slight difference in ATE between shielded and unshielded servomotors after 216 Gy of radiation, suggesting radiation effects on the servomotors. Uc3 (ATmegaS64M1 Rad-Hard) exhibits a slight ATE increase, comparable to the unshielded servomotor in Uc2.



(a) (b) (c)

Figure 4.15: Changes in temperature sensor readings during experiment #1. **(a).** Temperature variation of Uc1 Arduino Uno Unshielded. Two unshielded temperature sensors are affected in the same magnitude by the radiation from minute 13. **(b).** Temperature variation of Uc2 Arduino Uno shielded. Excessive radiation produces a saturation value in the unshielded sensor (green), while the shielded sensor (magenta) is unaffected. **(c).**Temperature variation of Uc3 ATmegaS64M1 Rad-Hard. The external temperature sensor changed the value from minute 18, but its change is less of magnitude than the Uc1 unshielded sensor. The internal temperature sensor started to change from minute 22, and also its changes are not as high experiment as in Uc1.

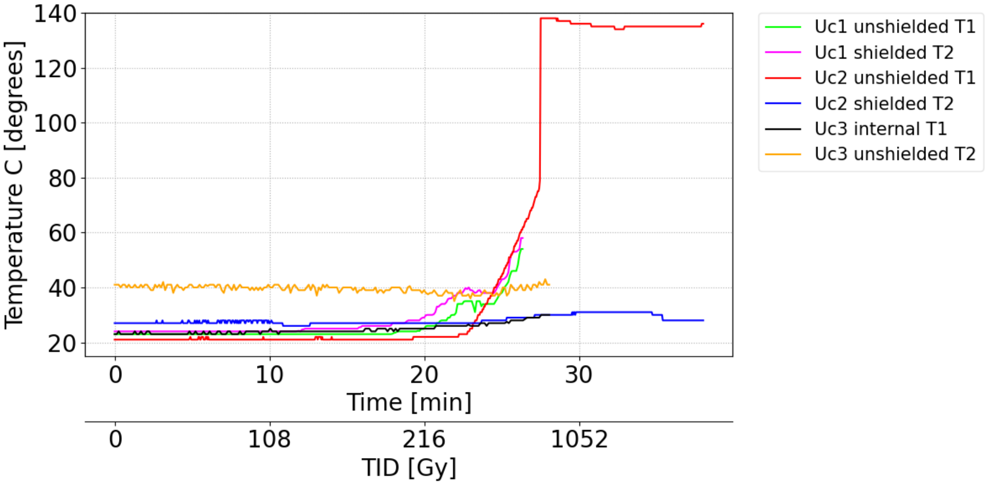


Figure 4.16: Temperature sensor behavior under radiation exposure in experiment #1. Uc1 sensors (green and magenta) showed a temperature increase 13 minutes before microcontroller failure, suggesting early electronic degradation. Uc2's unshielded sensor reached saturation at 1.5 Gy/s after minute 26, resulting in permanent damage, while the shielded sensor remained stable. Uc3 recorded lower temperatures changes compared to Uc1, indicating better stability with the Rad-Hard microcontroller.

4.4.2 Experiment #2

The setup of the experiment within the radiation facilities is depicted in Fig. 4.17. It includes a 3-DoF printer made of Acrylonitrile Butadiene Styrene (ABS) plastic material, mounted on the accelerator target. The microcontroller (ATmega64M1) is located at the upper left position. The ionizing radiation produced by the beam is uniform across the target area. However, joints 1 and 2 have more ABS plastic shielding compared to joint 3.

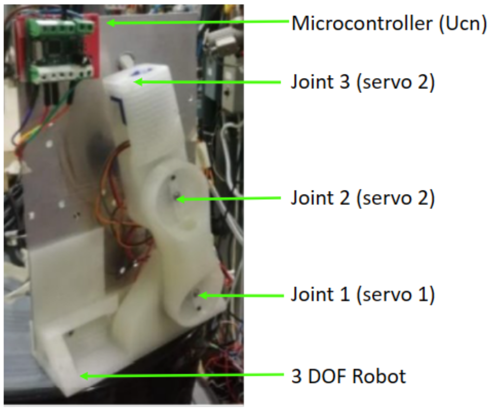


Figure 4.17: 3-DoF robotic arm of Experiment #2.

In this second experiment, a gamma radiation flow of 0.09 Gy/s was generated for a duration of 52 minutes. The ATmega64M1 microcontroller without radiation hardened tolerated a total ionizing dose of 282.15 Gy, with results presented in Table 4.2. Just before failure, the microcontroller emitted a continuous reset signal, disrupting the robot system's correct operation. The current in the microcontroller was recorded before and after the end of the test, increasing from 5.20 to 8.49 mA. This suggests a potential internal short circuit within the electronics, causing a reduction in impedance, current leakage in the semiconductor, and subsequent voltage fluctuations in the microcontroller's structure. During each acquisition period, the microcontroller's SRAM and EEPROM memories were examined, along with the stored data. Additionally, FLASH memory was downloaded after stopping the test and compared bit by bit with the original program to identify any changes. The analysis did not reveal any alterations to the data memories.

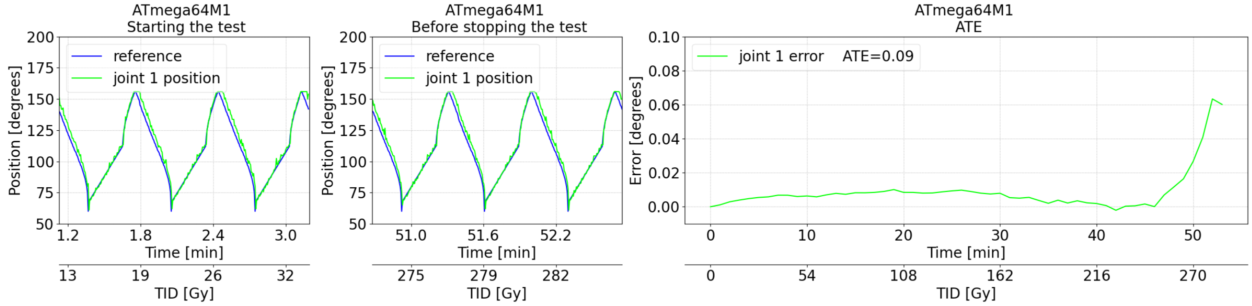
Table 4.2: Results of Experiment #2, representing the microcontroller's radiation dose, TID, ATE and work time.

Control Board	**Dose [Gy/s]**	**TID [Grays]**	**Time [min]**	**Joints ATE [degrees]**			**First & Last Lap ATE [degrees]**
				J1	J2	J3	
ATmega64M1	0.09	282.15	52'15"	0.09	0.12	0.80	0.76

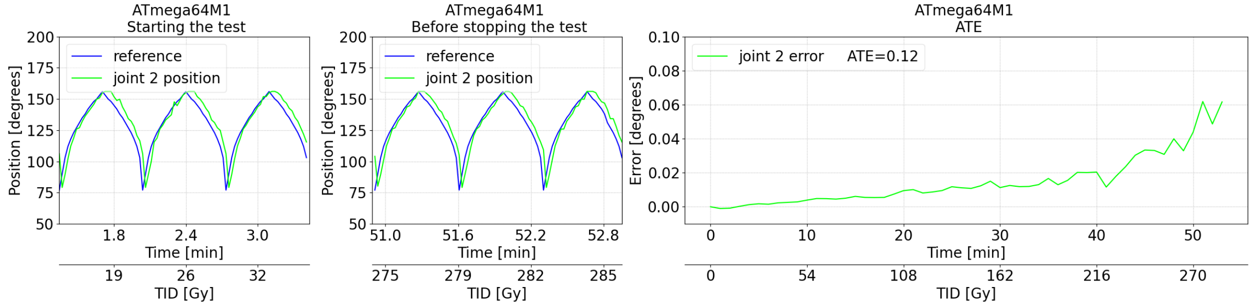
In this experiment, we examined the servomotor performance of each robotic joint. To achieve this, the robotic arm's microcontroller was programmed to execute a repetitive trajectory. This enabled tracking and comparing the deviation using the ATE metric from the ideal trajectory over time. Fig. 4.18 illustrates the position values at the beginning and end of the test, as well as the ATE metric throughout the experiment.

The observations indicate a distinct positional difference between Joints 1 and 2 compared to Joint 3 (Fig. 4.18). In particular, Joint 3 (Fig. 4.18(a)) exhibits significant position changes. This occurred because Joint 3 was enclosed in a thinner housing and the radiation beam hit its electronics more directly due to the servomotor's location, as illustrated in Fig. 4.17, causing it to absorb more radiation. Despite this, the three joints demonstrated variations in ATE throughout the experiment, with Joints 1 and 2 showing a slight increase, as shown in Fig. 4.18. The servomotor of Joint 3 shows clear signs of electronic damage, possibly aggravated by the microcontroller's degradation and the accumulated radiation dose, which might have impacted the performance of the other servos. For a complete analysis of the servomotors' performance, the ATE error was examined and is presented in Fig. 4.19. The findings are as follows: Joint 1: The ATE remains constant until 42 minutes (226.8 Gy), after which there is a slight increase in error with minor deviations from the reference values. Joint 2: The ATE increases steadily up to 40 minutes (216 Gy), then shows a sharp increase in error and a deviation in trajectory from 51 minutes (271 Gy) to the end of the assessment period. Joint 3: The ATE experiences a notable surge in error between 25 and 50 minutes (135-270 Gy), with growing discrepancies between the reference and feedback values.

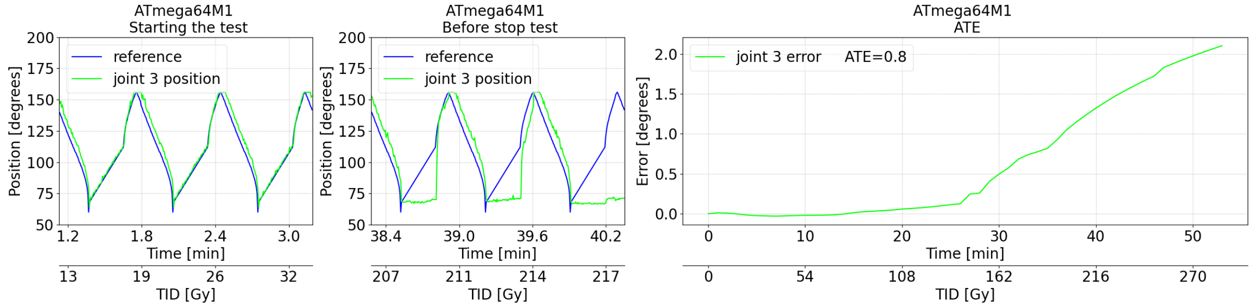
In contrast, the robotic arm has been set up to make a specified reference trajectory. This robot trajectory was derived using the forward kinematics of a 2-DoF system, excluding Joint 3 [304]. The results of this study are illustrated in Fig. 4.20. It can be seen that initial position errors were detected after 30-35 minutes of irradiation, corresponding to an



(a) The graph on the right (ATE representation) illustrates minimal variation between 0 and 42 minutes (accumulating 226.8 Gy of radiation dose). Beyond this point, an increase in error is evident, and the servomotors' positions begin to deviate slightly from the reference values.



(b) The ATE graph indicates a slight increase in error up to minute 40 (216 Gy of the total radiation dose). Subsequently, the error escalates more rapidly within the next 10 minutes. The middle graph demonstrates deviation and ripple effects on trajectories starting from minute 51 (271 Gy of total radiation dose) and continuing to the end of the test.



(c) The ATE graph exhibits a significant increase in error between minute 25 and minute 50 (135 to 270 Gy of the accumulated radiation dose). The middle graph highlights that the error becomes more pronounced, with a greater disparity between the reference and position feedback values.

Figure 4.18: Test #2. Position response of servomotors using the ATmega64M1 microcontroller. **Left graph:** initial positions at test commencement. **Middle graph:** positions monitored until either microcontrollers or accelerators cease functioning. **Right graph:** positions of errors in ATE metric throughout the tests.

accumulated radiation dose of 162 – 189 Gy. This position error increases and becomes more pronounced after 40 minutes of exposure, although communications persisted for an additional 12 minutes, up to a total of 52 minutes of irradiation. The program functioned as expected, with the robot continuously repeated the movement cycle. However, the data transmitted by the microcontroller (Atmega64m1) showed incremental position errors. This was particularly noticeable as the microcontroller began to degrade after 30 minutes of exposure to irradiation. This degradation likely occurs because the processor operates the program's instructions within a longer window time [153], leading to data corruption. Corrupt data (changes in bit values) cause random position errors without abrupt jumps or movements. This may be attributed to the robot's inertia and the smoothness of its trajectories. The positional errors of the robot do not accumulate but result in the robot's trajectory oscillating around the intended path.

For this experiment, the behavior of the internal temperature sensor of the ATmega64M1 microcontroller has been observed. Figure 4.21 illustrates the temperature fluctuations, which become more pronounced at minute 45 (486 Gy of accumulated radiation dose) as a result of radiation exposure and microcontroller degradation.

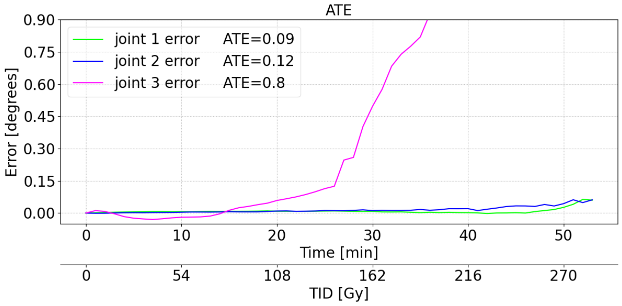


Figure 4.19: ATE comparison for the microcontroller and servos utilized in experiment #2. Joint 1 maintains stability up to 42 minutes with a TID of 226.8 Gy. Joint 2 begins to exhibit increased errors starting at 40 minutes with a TID of 216 Gy. Joint 3 shows substantial error escalation between 25 and 50 minutes with a TID of 135-270 Gy, respectively.

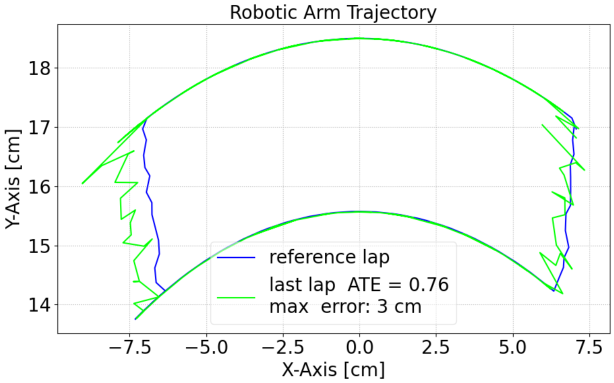


Figure 4.20: Robot trajectory comparison throughout the experiment #2. The blue line indicates the reference trajectory. The green line shows the final trajectory captured at the 50:55 minute mark of the experiment, with a total radiation dose of 275.4 Gy. The max distance error between the first and 115 last lap is 2 cm.

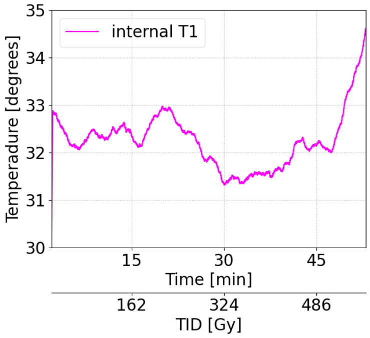


Figure 4.21: Experiment #2 showing the internal temperature progression of the ATmega64M1. The temperature remains relatively stable up to the 45-minute mark (234 Gy of accumulated radiation dose). After this point, the temperature rises steadily until the microcontroller ceases to function.

4.4.3 Experiment #3

The experimental setup within the radiation facilities is illustrated in Fig. 4.22, featuring a 3-DoF printer composed of ABS plastic material, mounted on the accelerator target. The microcontrollers tested were located in the upper left corner. The beam generates ionizing radiation uniformly over the target area. In this experiment, the joints are without ABS plastic shielding, which differs from experiment #2. Consequently, the servomotor is directly subjected to the impact of beam radiation. In this study, the servo motors of the individual joints of the robotic arm responses were examined . The robotic arm microcontroller was programmed to perform incremental executions of set point positions, enabling precise tracking and comparative analysis of deviations from the predetermined optimal trajectory. This comparison was quantitatively evaluated using the Absolute Trajectory Error (ATE) metric over an extended period. Tab. 4.3 summarizes the primary details of this experiment #4, where the flow of ionizing radiation was recorded at 0.18 Gy/s. The Arduino Uno (ATmega328P) operated for 22 minutes, accumulating a TID of 237 Gy. In contrast, the Raspberry Pico (RP2040) operated for 20 minutes with a TID of 222 Gy before showing signs of abnormal operation.

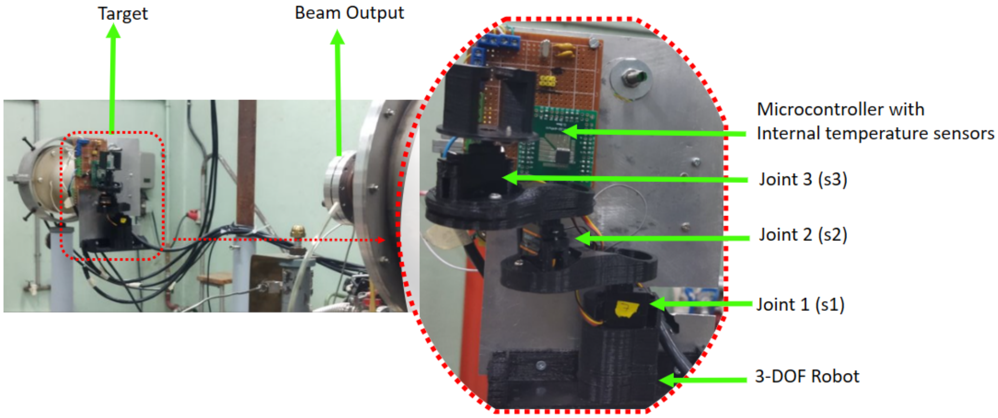


Figure 4.22: Accelerator target with irradiated components in experiment #3, #4.

Table 4.3: Results of Experiment #3, representing the microcontroller's radiation dose, TID, ATE and work time.

Control Board	**Dose [Gy/s]**	**TID [Grays]**	**Time [min]**	**Joints ATE [degrees]**			**First & Last Lap ATE [degrees]**
				J1	J2	J3	
Raspberry pico	0.18	222.30	20'35"	0.79	0.28	0.34	2.86
Arduino Uno	0.18	237.06	21'57"	0.53	0.38	0.74	1.89

The following results were observed related to the ech microcontroller.

Raspberry Pico (RP2040). Fig.4.23 illustrates the position values recorded along with the progression of the ATE metric throughout the duration of the experiment. At the 5-minute mark of the experiment, anomalies were observed in all servos. Specifically, Joint1 exhibited a significant deviation between minutes 5 and 8, during which it struggled to function correctly. By minute 9, Joint1 had completely failed, maintaining constant position feedback for the remainder of the experiment. Joint2 experienced a malfunction from minutes 5 to 7 before recovering its intended trajectory. However, ripples on the trajectory were detected in the feedback data for Joint2 due to radiation. Regarding Joint3, peak values were observed at minutes 3 and 9, coupled with a wrong response between minutes 5 and 7. Post-minute 10, Joint3 resumed its trajectory but displayed feedback ripples similar to those observed in Joint2. The Average Time to Error (ATE) for the joints increased with prolonged exposure to radiation. In particular, Joint1 had a higher ATE compared to the other joints due to its early failure. Furthermore, the ATE for Joint3 surpassed that of Joint2, as Joint3 was closer to the radiation beam, leading to greater radiation absorption.

The response of the trajectory, based on the collected data for Joint1 and Joint2, is depicted in Fig.4.24. The figure shows a suboptimal tracking of the reference trajectory, particularly during the last lap, which can be attributed to an early failure of Joint1. This failure exemplifies a situation in which the execution of a robotic task becomes unfeasible when a servo motor malfunctions. Furthermore, the Absolute Tracking Error (ATE) depicted in the right-hand plot of Fig.4.24 demonstrates a continual increase with each successive lap due to riples induced in individual joints.

The RP2040 microcontroller is equipped with an integrated temperature sensor (diode). During the course of the experiment, temperature readings from this sensor were recorded and are illustrated in Fig.4.25. The recorded data indicate a distinct decrease in temperature as the microcontroller is subjected to increasing levels of radiation. This observation contrasts markedly with outcomes from prior experiments utilizing both external and internal temperature sensors, where the recorded values exhibited an upward trend. Furthermore, the temperature behavior observed in this experiment is entirely different; the values continuously decline once the radiation beam is initiated. This suggests that the microcontroller's packaging may be thinner, consequently leading the sensor to receive more radiation-induced early changes in temperature values.

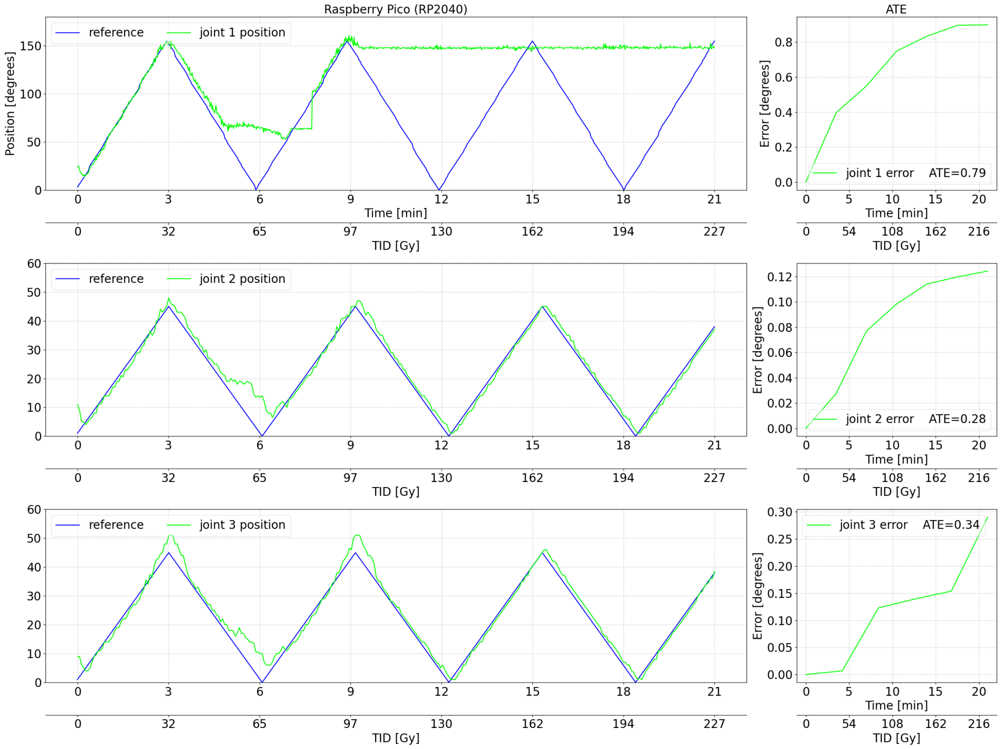


Figure 4.23: Experiment #3, servo motors position response with Raspberry Pico (RP2040). Anomalies were observed at the 5-minute mark. Joint1 exhibited significant deviation between minutes 5 and 8, eventually failing by minute 9. Joint2 malfunctioned from minutes 5 to 7 but recovered, though feedback ripples were detected due to radiation. Joint3 showed peak values at minutes 3 and 9, with malfunction between minutes 5 and 7, followed by recovery and feedback ripples. The Average Time to Error (ATE) increased with radiation exposure, with Joint1 having the highest ATE due to early failure and Joint3 surpassing Joint2 in ATE due to its proximity to the radiation source.

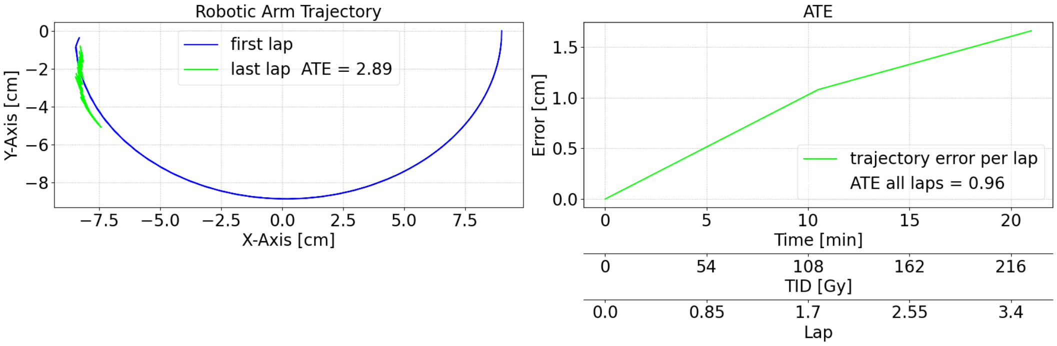


Figure 4.24: Experiment #3, trajectory response for Joint1 and Joint2 and Raspberry Pico (RP2040). The left-hand plot highlighting suboptimal tracking, particularly during the final lap due to the early failure of Joint1. This failure illustrates the impact of a malfunctioning servo motor on the feasibility of robotic task execution. The right-hand plot displays the Absolute Tracking Error (ATE), which increases progressively with each lap due to ripples in individual joints.

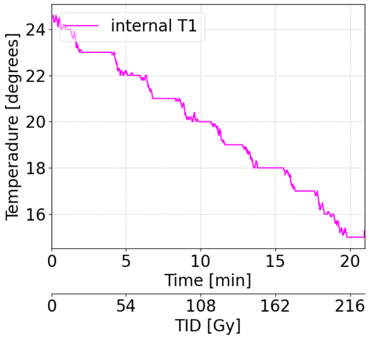


Figure 4.25: Experiment #3, Raspberry Pico (RP2040) onboard temperature sensors response. The data show a distinct decrease in temperature as radiation levels increase, contrasting with previous experiments where temperatures rose. This continuous decline suggests that the microcontroller's thinner packaging may cause the sensor to experience early radiation-induced temperature changes.

Arduino Uno (ATmega328P). Fig.4.26 provides a comprehensive illustration of the positional data logged over the course of the experiment. Joint1 started to deviate from the expected position at minute 7, resulting in a subsequent increase in error magnitude, particularly during peak value instances. Joint2 encountered operational issues as early as minute 1, characterized by oscillations in its trajectory and pronounced errors at peak moments. Similarly, Joint3 demonstrated feedback oscillations post-minute 1 and exhibited peak errors analogous to those seen in Joint1 and Joint2. The ATE for all joints showed a marked increase with extended radiation exposure. In particular, the ATE for Joint3 exceeded that of Joint2 and Joint1. This is attributed to Joint3's proximity to the radiation source, leading to higher levels of absorbed radiation.

The trajectory response, analyzed from the data for Joint1 and Joint2, is illustrated in Fig.4.27. This illustration highlights a significant deviation from the reference trajectory, most noticeably during the final lap as the position approaches the coordinates (-8.5,0). This discrepancy serves as an example of a malfunction in robotic task execution, which occurs when servomotors are subjected to radiation, resulting in positional inaccuracies. Furthermore, ATE, as shown in the right-side graph of Fig.4.27, indicates a persistent increase with each successive lap. This increment arises due to oscillations and inadequate stabilization in the feedback from the servomotors that affect each individual joint.

4.4.4 Experiment #4

The experimental setup used is similar to that of the experiment #3, which is illustrated in Fig. 4.22. Tab. 4.4 summarizes the primary details of this experiment #4, where the flow of ionizing radiation was recorded at 0.18 Gy/s. The ATmega64M1 operated for 20 minutes, accumulating 218 Gy TID. In contrast, the radiation-hardened version, ATmegaS64M1, operated for 28 minutes and reached 311 Gy TID before showing signs of abnormal operation. In contrast to the other experiments, alterations in data memory were identified by examining all accessible memories. The ATmega64M1 exhibited 11160 corrupted bytes in its FLASH memory after 28 minutes (217.7 Gy of accumulated radiation dose). However, no corrupted memory data was observed in the ATmegaS64M1 (Rad-Hard).

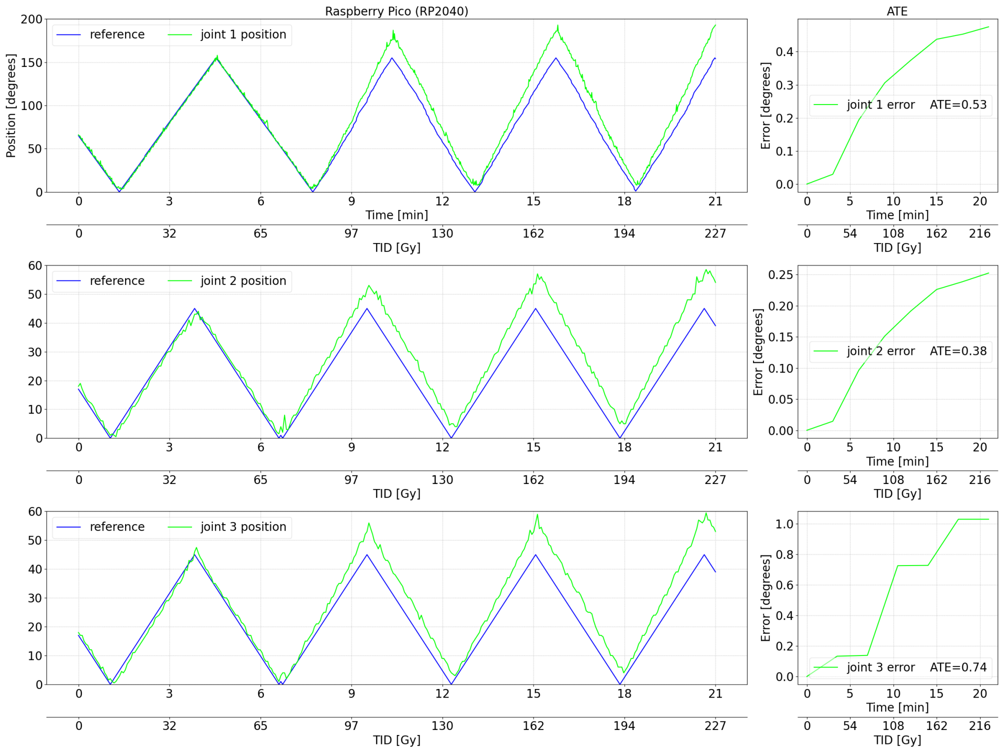


Figure 4.26: Experiment #3, servo motors position response with Arduino Uno (ATmega328P). Graphs show the positional data of joints throughout the experiment. Joint1 deviated from the expected position at minute 7, with increasing error during peak values. Joint2 experienced trajectory oscillations starting at minute 1, and Joint3 showed similar oscillations and errors. The ATE increased for all joints with prolonged radiation exposure, with Joint3 exhibiting the highest ATE due to its closer proximity to the radiation source.

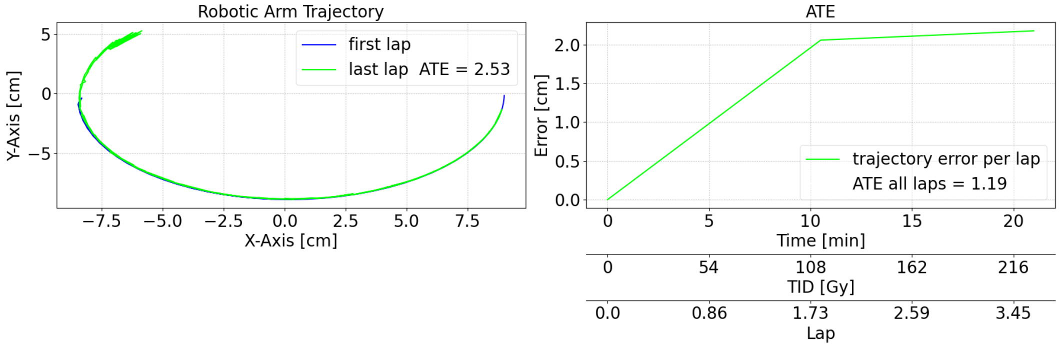


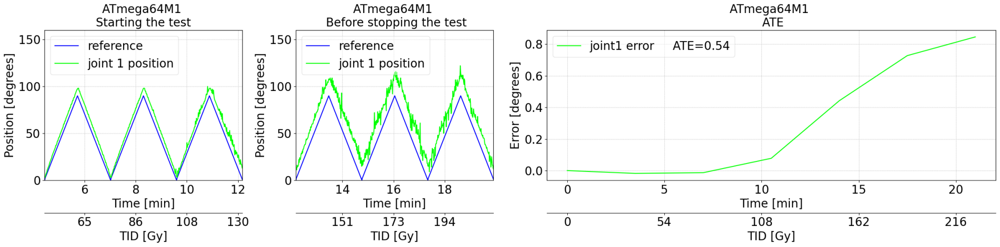
Figure 4.27: Experiment #3, trajectory response for Joint1 and Joint2 and Arduino Uno (ATmega328P). Graphs shows the trajectory response for Joint1 and Joint2, highlighting significant deviations from the reference, particularly near coordinates (-8.5, 0) during the final lap. This demonstrates task execution malfunctions caused by radiation-induced positional inaccuracies. The right-side graph illustrates a persistent increase in ATE due to oscillations and inadequate feedback stabilization from the servomotors.

Table 4.4: Results of Experiment #4, representing the microcontroller's radiation dose, TID, ATE and work time.

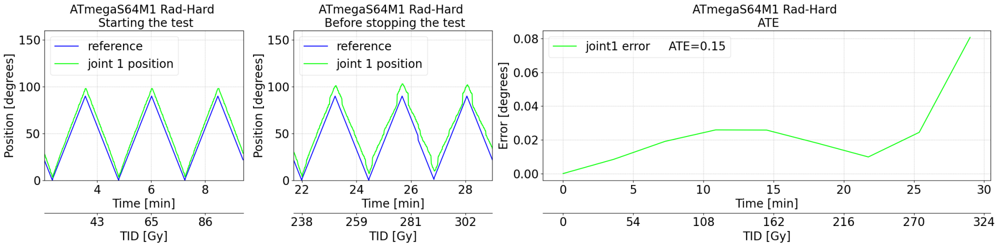
Control Board	**Dose [Gy/s]**	**TID [Grays]**	**Time [min]**	**Joints ATE [degrees]**			**First & Last Lap ATE [degrees]**
				J1	J2	J3	
ATmega64M1	0.18	218.88	20'16"	0.54	0.66	1.21	1.89
ATmegaS64M1 (RH)	0.18	311.22	28'49"	0.15	0.30	0.46	0.95

Both microcontrollers run the same program, which executes a semicircular path (Fig. 4.7). Position-controlled servos traverse a specified route, moving from 0 to 90 degrees and returning to 0 degrees, increasing and decreasing one degree at a time. The velocity-controlled servo operates at 180 degrees per second. The position and velocity values of each servomotor have been verified for the joints in this experiment. *Joint 1 position response.* (Fig. 4.28). The ATmega64M1 trajectory begins to show ripples at minute 8 (76.8 Gy of accumulated radiation dose TDI), while ATmega64SM1 (Rad-Hard) experiences smaller ripples and peaks starting at minute 21 (201.6 Gy). ATE values indicate that ATmega64M1 (ATE = 0.54) exhibits a larger error than ATmegaS64M1 (ATE = 0.15). *Joint 2 position response.* (Fig. 4.29). The trajectory behavior is similar to Joint 1, but the ATE value is higher due to Joint 2 being closer to the radiation beam, causing the servomotor to absorb more radiation. Ripples in the ATmega64M1 trajectory begin at minute 5 (54 Gy), while the ATmega64SM1 (Rad-Hard) shows smaller ripples and peaks from minute 21 (201.6 Gy). The ATE shows that ATmega64M1 (ATE = 0.66) has a higher error compared to ATmegaS64M1 (ATE = 0.3). *Joint 3 velocity response.* (4.30). The ATE value is higher than in Joint 1 and Joint 2 because Joint 3 is closest to the radiation beam. The ATmega64M1 trajectory starts deviating in velocity at minute 7 (67.2 Gy), whereas the ATmega64SM1 (Rad-Hard) shows significant changes beginning at minute 20 (192 Gy). The results show a noticeable deviation compared to the positions of the servomotors in previous experiments (Fig. 4.18). In addition, it implies that a higher radiation exposure results in more data corruption, causing the trajectory to fluctuate around the ideal path. Moreover, Fig. 4.31 presents the ATE values for all joints. It can be seen that the ATmega64SM1 Rad-Hard microcontroller exhibits lower ATE joint values compared to the non-Rad-Hard variant. Additionally, the ATE begins to rise around minute 20 for the Rad-Hard version, whereas it starts increasing from minute 7 for the non-Rad-Hard version. This suggests that the Rad-Hard version provides greater stability in controlling the servomotors.

The trajectory of the robotic arm was evaluated, tracking its evolution before and during radiation exposure. Fig. 4.32 presents the trajectories with these microcontrollers, and the evolution of the robot's trajectory error, ATE, throughout the experiments. It is evident that both microcontrollers malfunction; ATmega64M1 (Non-Rad-Hard) after 20 minutes of radiation, and ATmegaS64M1 (Rad-Hard) post 28 minutes of radiation. Additionally, the robot's position error (ATE) escalates earlier, at minute 7 (67.2 TID) for the ATmega64M1, and minute 20 (192 TID) for the ATmegaS64M1.

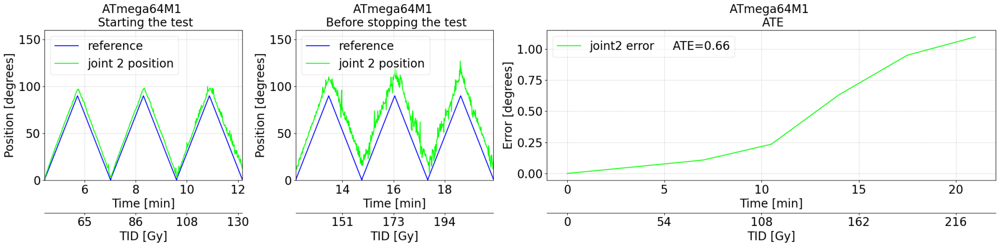


(a) ATmega64M1 Joint 1 response.

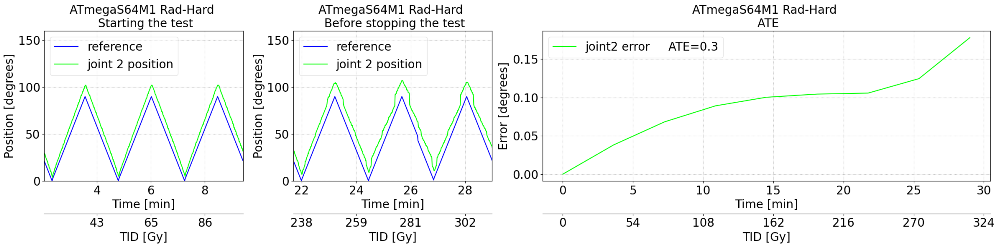


(b) ATmegaS64M1 (Rad-Hard) Joint 1 response.

Figure 4.28: Experiment #4. Position response of Joint 1. The ATmega64M1's trajectory starts to exhibit ripples from minute 8 (TID 76.8 Gy), whereas the ATmega64SM1 (RH) shows smaller ripples and peaks from minute 21 (TID 201.6 Gy). The ATE graph on the right illustrates that the ATmega64M1 (ATE=0.54) has a higher error value compared to the ATmegaS64M1 RH (ATE=0.15).

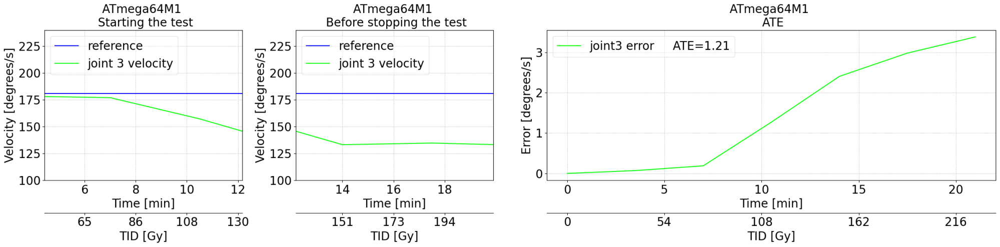


(a) ATmega64M1 Joint 2 response.

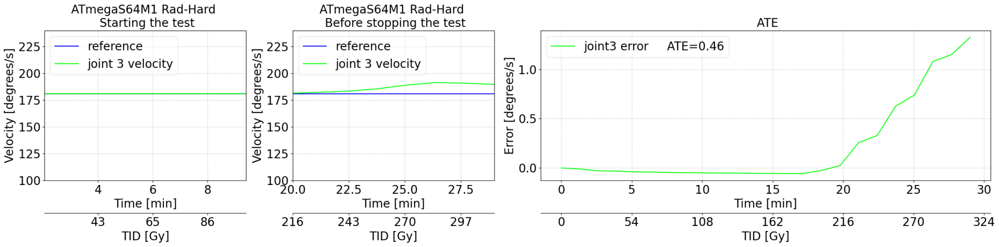


(b) ATmegaS64M1 (Rad-Hard) Joint 2 response.

Figure 4.29: Experiment #4. Position response of joint 2. The trajectory of joint 2 resembles that of joint 1; nonetheless, joint 2 exhibits a higher ATE due to its closer exposure to the radiation source. The ATmega64M1 displays ripples at minute 5 (TID 54 Gy), while the ATmega64SM1 (RH) begins to show smaller ripples and peaks at minute 21 (TID 201.6 Gy). The right graph demonstrates that the ATmega64M1 (ATE = 0.66) has a higher error than the ATmegaS64M1 (ATE = 0.3).

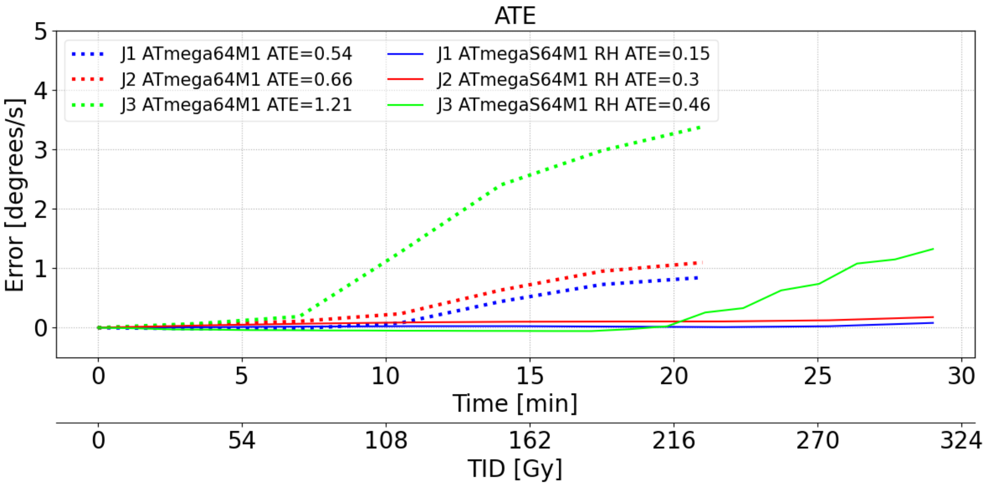


(a) ATmega64M1 Joint 3 response.



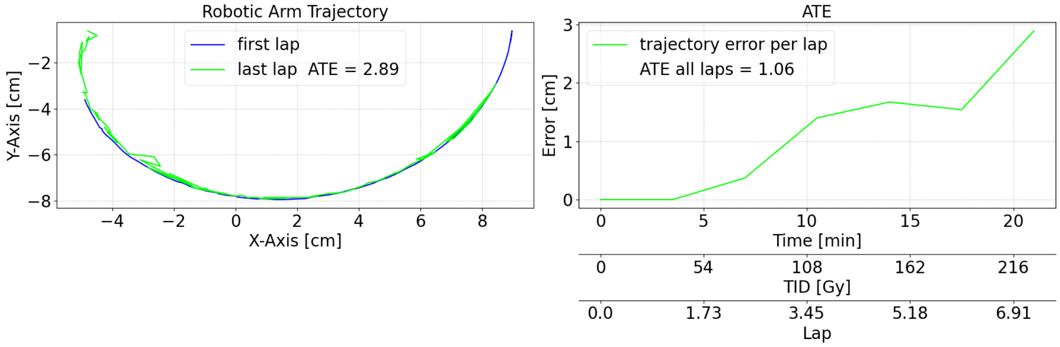
(b) ATmegaS64M1 (Rad-Hard) Joint 3 response.

Figure 4.30: Experiment #4. Velocity response for Joint 3. Joint 3 exhibits a higher ATE value than Joint 1 and Joint 2 due to its proximity to the radiation source. The velocity of the ATmega64M1 changes at minute 7 (TID 67.2 Gy), whereas the ATmega64SM1 (RH) shows a significant alteration at minute 20 (TID 192 Gy). The right graph illustrates that the ATmega64M1 (ATE=1.21) has a greater error value compared to the ATmega64SM1 (ATE=0.46)..

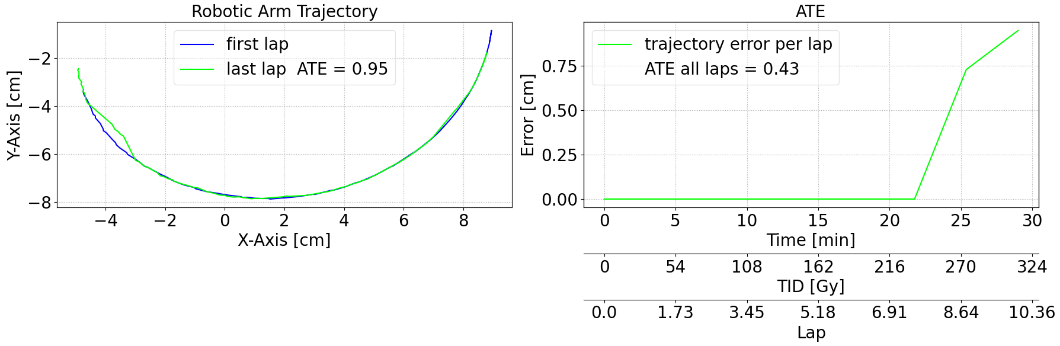


(a)

Figure 4.31: ATE representation of ATmega65M1 and Atmega64SM1 (RH) microcontrollers and joints in experiment #4. The values show that the ATmega64SM1 (RH) exhibits lower ATE joint values compared to the non-Rad-Hard version. The ATE increase starts at minute 20 for the Rad-Hard version, while the non-Rad-Hard version shows an increase starting at minute 7, indicating greater stability in the Rad-Hard microcontroller for servomotor control.



(a) ATmega64M1 trajectory response.

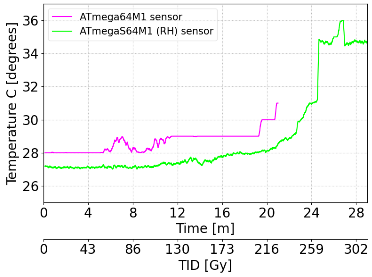


(b) ATmegaS64M1 (Rad-Hard) trajectory response.

Figure 4.32: Comparison of robot paths during the experiment #4. The blue line marks the reference path of the initial lap, completed in 2 minutes and 53 seconds with a TID of 31.26 Gy. Conversely, the green line depicts the final path: at 20 minutes and 16 seconds for the ATmega64M1 with a TID of 218.88 Gy and at 28 minutes and 49 seconds for the ATmegaS64M1 (RH) with a TID of 311.22 Gy. Both scenarios show ripples and deviations from the reference path in the last lap. The ATE for the ATmegaS64M1 begins to rise after 20 minutes, while the ATE for the ATmega64M1 starts increasing after 7 minutes.

The responses of the internal temperature sensor of the microcontrollers are shown in Fig. 4.33. The ATmega64M1 sensor indicates a temperature increase starting at minute 6, corresponding to a TID of 57.6 Gy. This rise becomes more noticeable at minute 19, with a TID of 182.4 Gy, particularly after a brief test interruption due to a malfunction. In contrast, the ATmegaS64M1 sensor starts to show a temperature increase at minute 11 with a TID of 105.6 Gy, with a more significant rise evident by minute 25, just before the test concludes. These observations suggest that the Rad-Hard microcontroller maintains a temperature more consistent than the non-Rad-Hard model until its internal structure starts to degrade.

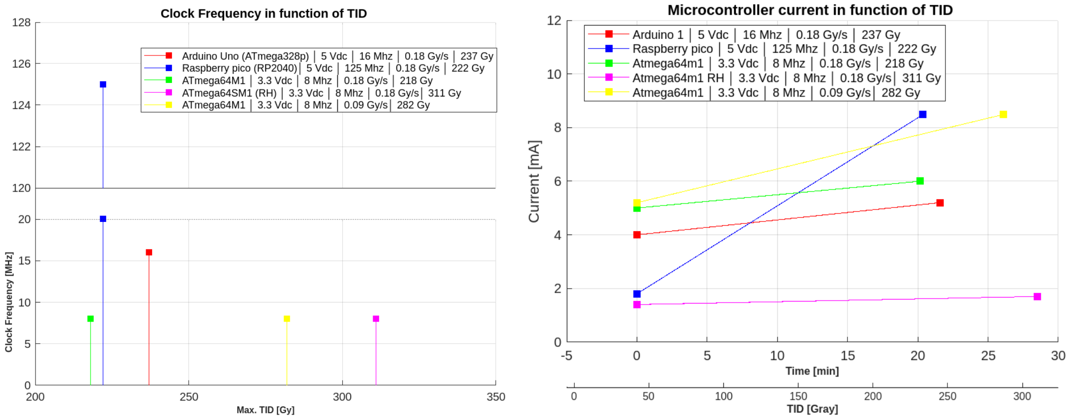
Finally, the voltages, currents and clock frequencies for experiments #3 and #4 were analyzed, as shown in Fig. 4.34. According to [153], an increase in the total ionizing dose (TID) leads to 'Timing Window Violations', where the instruction set fails to execute within the clock cycles, causing microcontroller failures and incorrect data propagation. The relationship between TID, clock frequency and operating voltage was analyzed using data from experiments #3 and #4. Fig. 4.34(a) shows that the Raspberry Pico (RP20240) running at 120 MHz and 3.3V



(a) ATmega64M1 Joint 1 response.

Figure 4.33: Temperature response in ATmega64M1 and ATmegaS64M1 in experiment #4. The graph indicates a similar pattern to previous experiments (temperature rises as radiation exposure increases); however, the Rad-Hard internal sensor exhibited fewer fluctuations until its internal structure started to degrade.

recorded a TID of 222 Gy. The ATmega324P with a 16 MHz clock had a TID of 237 Gy, while the ATmega64M1 at 8 MHz and 0.09 Gy/s reached 282 Gy. The Rad-Hard ATmegaS64M1 at 8 MHz absorbed a TID of 311 Gy. These results align with the literature, showing that higher frequencies result in lower TIDs, except for the non-Rad-Hard ATmega64M1, which tolerated less TID than the RP2040 and ATmega328P. Therefore, further testing is recommended for accurate data. Electrical analysis was performed throughout the experiments, measuring the current before and after irradiation for each microcontroller; see Fig. 4.34(b). After radiation exposure, certain microcontrollers showed increased current consumption, correlated with radiation-induced malfunctions. As confirmed in [153], an increase in TID results in a higher leakage and a higher total current. Significant increases were observed in Arduino UNO, Raspberry Pico and non-Rad-Hard ATmegaS64m1, while Rad-Hard ATmegaS64m1 showed minimal increase. In particular, current sometimes decreased with higher TID, possibly due to failure of architectural components, reducing power draw.



(a)

(b)

Figure 4.34: Experiments #3 and #4 linked voltage, current, and clock frequency. (a) Maximum TID versus voltage and clock frequency for the microcontroller; (b) Current over TID and time.

4.5 Discussion

The main observations from the tests are summarized in the following and on the Tab. 4.5.

- **Mitigation techniques:** Rad-Hard microcontroller and lead shielding have demonstrated advantages in ionizing radiation environments, with the shielding of electronic parts proving to be more effective and cost-efficient.
- **Microcontrollers:** Using a Rad-Hard ATmegaS64M1 was found to be effective in tolerating higher levels of ionizing radiation (up to 311.2 Gy of TID) and maintaining functionality significantly longer between 8% and 35% more compared to the other microcontrollers tested (ATmega328P with 286.92 Gy and ATmega64M1 with 218.88 Gy). Additionally, the lead shielding provided substantial protection to the microcontroller, preventing failure and ensuring continued operation under radiation exposure. Radiation levels critically affect the lifespan of these components, according to the tests.
- **Servo motors:** They experienced positional deviations and errors. Shielding the microcontroller reduced the positioning error, mitigating radiation effects and preserving proper functionality. The placement of servo motors on the accelerator target significantly influences their performance, as the radiation beam impacts the electronics differently based on their location and the material of the servo motor case. Moreover, the radiation's effect on servo motor positioning varied with the type of microcontroller used. Specifically, servo motors managed by Rad-Hard microcontrollers showed improved performance and less degradation.
- **Temperature sensors:** Under irradiation, higher temperature values were observed, indicating potential electronic degradation. However, the values are the same once the irradiation ceased. Sensor temperature errors varied with radiation exposure and dose rate, possibly impacting reliability and long-term performance. Notably, the unshielded ATmega328P showed substantial temperature errors. In contrast, the integrated temperature sensors had fewer errors than the external sensor. The internal temperature sensor of the Raspberry Pico (RP2040), excepted, displayed temperature fluctuations from the beginning of the test.
- **Robotic arm trajectory:** The Rad-Hard demonstrated superior efficacy in robotic arm movement. At a dose rate of 0.18 Gy/s, it maintained trajectory stability for 19 minutes (200 Gy), surpassing the non-Rad-Hard model, which remained stable for only 7 minutes (70 Gy). The Rad-Hard variant also achieved lower trajectory error, with an ATE of 1.51 in the final lap, resulting in a maximum deviation of 3 cm at a TID of 311.22 Gy. By comparison, the non-Rad-Hard version had an ATE of 1.89 and a displacement error of up to 5 cm at a TID of 218.88 Gy. At a reduced dose rate of 0.09 Gy/s, non-Rad-Hard model recorded an ATE of 0.76, causing a trajectory shift of up to 3 cm with a TID of 282.15 Gy. Performance tracking using Arduino Uno (ATmega328P) and Raspberry Pi Pico (RP2040) was inferior. Specifically, the RP2040 experienced a servo motor failure at 7 minutes with a TID of 75 Gy, resulting in a significant trajectory error of up to 17 cm. The ATmega328P exhibited an offset in its trajectory, with an error of up to 7 cm.

Table 4.5: Summary of the microcontroller experiment results, highlighting time, dose, signal reset, failure cause, temperature difference, and ATE for an overall perspective.

Board	**Test**	**Dose** [Gy/s]	**TID** [Gy]	**Time** [min:sec]	**Fails detected**	**Servo motor position ATE** [°]	**Robotic Arm trajectory last lap**		**Temperature init / final** [°]
							ATE [cm]	**Max error** [cm]	
ATmega328P	#1	0.18	286.92	26:34	Fatal error UC stop working	Parallax s1 0.28 (*) s2 0.29	—	—	et1 24->57 et2 23->54
	#1 (*)	0.18 to 1.5	1123	38:00	No fail	Parallax s1 0.06 (*) s2 0.10	—	—	et1 27->30 (*) et2 21->135
	#3	0.18	237.06	21:57	Unknow reset Trajectory with ripples and offset	SG90 s1 0.53 s2 0.38 s3 0.74	1.89	7	—
RP2040	#3	0.18	222.30	20:35	Unknow reset Trajectory with ripples and offset Servo motor 1 break early	SG90 s1 0.79 s2 0.28 s3 0.34	2.86	17	it1 24-8
ATmega64M1	#2	0.09	282.15	52:15	WDT and unknow resetTrajectory with ripples and offset	SG90 s1 0.09 s2 0.12 s3 0.8	0.76	3	it1 33-34
	#4	0.18	218.88	20:16	Trajectory with ripples and offset WDT and unknow resetCorrupted bytes in FLASH memory	Grove s1 0.54 s2 0.66 Hitech s3 1.21	1.89	5	it1 28-31
ATmega64SM1 (RH)	#1	0.18	303.48	28:06	WDT and unknow reset	Parallax s1 0.1	—	—	et1 25->30 it1 41->42
	#4	0.18	311.22	28:49	Trajectory with ripples and offset WDT and unknow reset	Grove s1 0.15 s2 0.30 Hitech s3 0.46	1.51	3	it1 27-35

Chapter 5

Conclusions

5.1 Conclusions

This work outlines the development and testing of a highly robust automated system for detecting components in various industrial applications (in this work, as an example fasteners were used), designed to operate under different environmental conditions, both indoors and outdoors. Traditional fastener detection systems are typically restricted to controlled lighting conditions. However, this system addresses these limitations by integrating a vision-based subsystem, force sensing, and a robotics platform. The proposed solution has been rigorously tested with fasteners ranging from M6 to M24, demonstrating exceptional performance even in semi-unstructured environments.

At the core of the system is a vision subsystem that employs an industrial smart camera with dynamic focus and Near-Infrared (NIR) illumination. This enables the system to handle various fastener sizes and adjust to the specific lighting conditions of the environment. The combination of vision with the collaborative robot (UR5e), which features an embedded force sensor and a custom tool, allows for precise monitoring and interaction force control. The developed algorithms based on Spiking Neural Networks (SNNs) effectively detect the center and rotation of fasteners, ensuring accurate and reliable fastening operations. These algorithms were extensively tested using both synthetic datasets and real-world applications, further verifying their robustness.

A detailed comparison of several SNN models and hardware configurations was carried out to identify the best-performing system. The results showed that Deeplab EdgeMobilenet FP32 achieved the highest accuracy (98.97% mAP IOU) while Deeplab Mobilenet-v2 FP32 was the fastest (88.49 FPS) when running on an Nvidia GTX 1660 Ti GPU. However, for applications that require embedded systems, optimizing these models with pre-quantization techniques was found to significantly enhance performance without a major loss in accuracy. This optimization is particularly important for reducing inference time, which is critical for time-sensitive industrial operations.

The system was further validated through a series of experiments, both theoretical and experimental, focusing on the tolerances of socket wrenches and fasteners. The results revealed that as fastener size increases, translation tolerances become more flexible while rotation tolerances tighten. For larger fasteners, the detection algorithm can allow for more leniency in centering, but rotation detection must be more precise due to increased mechanical forces exerted during the fastening process. These findings underscore the importance of carefully selecting hardware and algorithms based on the specific industrial application and fastener size.

Moreover, the masks generated by SNNs, although effective in challenging lighting conditions, were not always flawless, requiring additional vision algorithms to enhance accuracy. A comparison of several models showed that while Deeplab models performed the best in terms of speed and accuracy, hardware optimization is essential to reduce processing time for embedded systems. Pre-quantization, in particular, emerged as the optimal method for achieving this balance, making the neural networks lighter and faster without compromising performance.

The proposed methodology was rigorously tested with synthetic datasets that varied in brightness, sharpness, and field of view (FOV). The system achieved a rotation accuracy of 99.86% and a center detection rate of 100%, demonstrating the system's ability to perform exceptionally well under controlled lighting conditions. Real-world testing further validated these results, with the system achieving an accuracy of 96.26% and inference time of 37.91 using Deeplab EdgeMobilenet FP32 on Intel Core I9 10-Gen-900H CPU, in tasks performed under various lighting and environmental conditions. The integration of force sensors was a key component in improving the system's dependability. These sensors provided real-time data on task conditions, enabling automated error correction based on the force exerted during the fastening process.

As summary, this topic presents a highly reliable and accurate robotic system for fastener detection and handling, capable of operating effectively in a wide range of industrial environments. The system's ability to adapt to fluctuating lighting, different fastener sizes, and semi-unstructured conditions represents a significant advancement over traditional detection systems, which are typically constrained by controlled environments. The vision system's use of NIR illumination and dynamic focus, combined with advanced SNN algorithms and force feedback, ensures precise detection of fastener position and orientation.

The study highlights the critical role that fastener size plays in defining the system's algorithmic flexibility, particularly in terms of translation and rotation tolerances. As fastener size increases, the detection algorithm must become more rigid with respect to rotation, while force calibration is crucial to account for increased friction and mechanical forces. The optimization of neural networks, particularly through pre-quantization, enables the system to maintain high accuracy while reducing inference time, making it well-suited for embedded systems and time-critical industrial applications. Ultimately, this research provides a robust and scalable framework for implementing automated fastener detection and handling systems in real-world industrial scenarios. The system's ability to address challenges such as unstructured lighting, worn or damaged fasteners, and fluctuating environmental conditions makes it a highly effective solution for modern industrial automation.

Other objective of this study is to develop a framework focused at enhancing the performance and safety of robotic tasks through optimal tool positioning and robot pose configurations. This research particularly emphasizes the design and optimization of robotic tools for opera-

tions that require precise control of forces and torques, such as screwing tasks. By employing forward and inverse kinematics, the study identifies the most advantageous robot poses to redistribute the forces and torques produced by external tools to the robot's robust joints (base, shoulder, elbow). Furthermore, the geometric Jacobian is utilized as an optimization function to determine the ideal pose for screwing applications. The experimental results highlight the critical importance of aligning the external tool axis with the robot's joint axes and selecting suitable tool designs, as these factors significantly optimize joint torques and ensure safe robot operation. The findings also indicate that the proposed force control methodology, coupled with the use of an external bracket, effectively reduces joint stress, allowing safe and efficient operation of high-torque tools in both constrained and standard environments.
The results from the simulations provide substantial observations regarding the design and optimization of robotic tools, particularly concerning the forces and torques acting on the tool flange and the entire assembly during operation. The data show that tool designs characterized by higher centers of gravity (CoG) and smaller tool center points (TCP) typically minimize the torque on the robot's tool flange. In contrast, designs featuring lower CoG and larger TCP values tend to elevate the torque on the flange, potentially increasing stress on the robot's joints. This highlights the importance of careful tool design and positioning to mitigate risk. Additionally, the investigation reveals an inverse relationship between force and torque: as torque increases, force tends to decrease, and vice versa. This dynamic is crucial for selecting tool designs that remain within the robot's operational limits. Notably, a substantial torque consistently manifests along the z-axis, aligning with the screwing tool axis, which is critical for optimizing the robot's positioning.
The study further elaborates on the positioning strategies employed during simulations and real tests. The optimal design and placement of tools are imperative for minimizing torque on robot joints. Testing various tools alongside different robot poses revealed that tool g is the most effective in torque management across both robot platforms. Additionally, the proposed algorithm for pose optimization maximizes torque distribution to stronger joints while minimizing it in weaker joints, ensuring that all poses remain within torque limits.
Experimental data, particularly from the UR5e robot, indicate discrepancies between joint torques derived from the Jacobian and those calculated via the URScript API, primarily due to friction and dynamic influences. The use of mixed control strategies has been shown to improve torque distribution significantly, allowing the robot to operate safely even when facing high torques. Notably, using position control mode without an external bracket resulted in excessive stress on joint 6, which necessitated the application of an external bracket to mitigate risks. Similarly, experiments with the Pilz robot highlighted the significance of tool design in torque distribution across joints, illustrating that appropriate alignment of tool and joint axes is vital for maintaining operational safety up to 60Nm. The experiments conducted in impedance control mode revealed a substantial reduction in stress on joint 6, particularly under free movement conditions. When tested at higher torque set points (45 Nm), the joints remained within safety limits, validating the robustness of the proposed force control technique. Successful applications of the method in constrained environments, such as the ROSE project, demonstrate its practical applicability and effectiveness in real-world scenarios.

The findings of this topic carry substantial implications for the design and implementation

of robotic tools, particularly in scenarios requiring precise force and torque control, such as screwing operations. The research recommends for a systematic approach to optimizing tool design, robot poses, and tool orientations to enhance the efficiency and reliability of robotic systems while ensuring that they operate within safe limits. This study highlights the important connection between tool design and robot posture optimization, offering a solid framework to enhance the effectiveness and safety of robotic operations. The strategies and algorithms proposed here contribute significantly to the field of robotics, offering insights that can be translated into practical applications across various industrial settings.
Overall, the experimental results reinforce the notion that aligning the external tool's axis with the robot's joint axes, coupled with the selection of appropriate tool designs, is crucial for optimizing joint torques and ensuring safe robot operation. Moreover, the comparison of experimental data with theoretical models emphasizes the importance of accounting for dynamic and frictional factors in torque calculations. The proposed force control methodology, alongside the incorporation of an external bracket, effectively reduces joint stress and facilitates the safe use of high-torque tools in both constrained and standard operating environments.

Finally, This work focuses on analyzing the impact of radiation (0.09/0.18/1.5 Gy/s) on various electronic components, such as microcontrollers, servomotors, and digital temperature sensors, as well as the movement of a 3-DoF robotic arm over time. In addition, this study examines the impact of radiation on the trajectories of two robotic arms, evaluating the durability and performance of these components while employing different strategies from the literature to compare radiation effects, such as lead shielding and component hardening. Furthermore, the radiation effects on memory components, including FLASH, EEPROM, and SRAM, were evaluated. The findings offer critical insights for the design and implementation of robotics in hostile environments. The experimental tests were conducted using a linear particle accelerator at CIEMAT in Spain. The main observations from the tests are summarized in the following. *Mitigation Techniques:* Radiation-Hard microcontrollers and lead shielding have demonstrated advantages in ionizing radiation environments, with the shielding of electronic parts proving to be more effective and cost-efficient.
Microcontrollers: Using a Rad-Hard microcontroller (ATmegaS64M1) was found to be effective in tolerating higher levels of ionizing radiation (up to 311.2 Gy of TID) and maintaining functionality significantly longer between 8% and 35% more compared to the other microcontrollers tested (ATmega328P with 286.92 Gy and ATmega64M1 with 218.88 Gy). Additionally, the lead shielding provided substantial protection to the microcontroller, preventing failure and ensuring continued operation under radiation exposure. The tests also indicated that radiation levels are critical in determining the lifespan of these components.
Servo Motors: The servomotors encountered positional deviations and errors due to radiation exposure. Shielding the microcontroller reduced the positioning error, mitigating radiation effects and preserving proper functionality. The placement of servo motors on the accelerator target significantly influences their performance, as the radiation beam impacts the electronics differently based on their location and the material of the servo motor case. Moreover, the radiation's effect on servo motor positioning varied with the type of microcontroller used. Specifically, servo motors managed by Rad-Hard microcontrollers showed improved

performance and less degradation. *Temperature Sensors:* Under irradiation, higher temperature values were observed, indicating potential electronic degradation. However, the values returned to normal once the irradiation ceased. Sensor temperature errors varied with radiation exposure and dose rate, possibly impacting reliability and long-term performance. Notably, the unshielded ATmega328P showed substantial temperature errors. In contrast, the integrated temperature sensors had fewer errors than the external sensor. The internal temperature sensor of the Raspberry Pico (RP2040) displayed temperature fluctuations from the beginning of the test.
Robotic Arm Trajectory: The Rad-Hard AtmegaS64M1 demonstrated superior efficacy in robotic arm movement. At a dose rate of 0.18 Gy/s, it maintained trajectory stability for 19 minutes (200 Gy), surpassing the non-Rad-Hard model, which remained stable for only 7 minutes (70 Gy). The Rad-Hard variant also achieved lower trajectory error, with an Average Trajectory Error (ATE) of 1.51 in the final lap, resulting in a maximum deviation of 3 cm at a TID of 311.22 Gy. By comparison, the non-Rad-Hard version had an ATE of 1.89 and a displacement error of up to 5 cm at a TID of 218.88 Gy. At a reduced dose rate of 0.09 Gy/s, the non-Rad-Hard model recorded an ATE of 0.76, causing a trajectory shift of up to 3 cm with a TID of 282.15 Gy. Performance tracking using Arduino Uno (ATmega328P) and Raspberry Pi Pico (RP2040) was inferior, with the RP2040 experiencing a servo motor failure at 7 minutes with a TID of 75 Gy, resulting in a significant trajectory error of up to 17 cm. The ATmega328P exhibited an offset in its trajectory, with an error of up to 7 cm.

The findings from this topic emphasize the critical importance of radiation mitigation strategies for robotic systems operating in extreme environments. Rad-Hard microcontrollers significantly outperformed conventional ones in radiation-heavy conditions, demonstrating greater tolerance and stability. However, despite their superior performance, even these components eventually succumb to high TID levels above 300 Gy. The inclusion of lead shielding was also shown to effectively prolong the operational lifespan of microcontrollers and servomotors, underscoring its value as a cost-effective radiation mitigation technique.
The robotic arm trajectory data further supports the robustness of Rad-Hard controllers, which not only maintained more precise trajectories but also experienced fewer functional errors compared to conventional controllers. This precision is critical for applications where trajectory accuracy is essential, such as object manipulation, medical robots or space exploration.
Moreover, the study highlights that temperature sensors can serve as early indicators of radiation-induced degradation, with anomalies in their readings signaling potential microcontroller failure. Such signals could be utilized to prompt corrective measures, such as activating backup systems or adjusting robot control algorithms to ensure continued functionality.
This research contributes vital insights into the design of robotics for hostile environments, particularly where ionizing radiation is a significant factor. The experiments confirm that radiation-hardening techniques, such as the use of Rad-Hard microcontrollers and lead shielding, are essential for maintaining the long-term functionality and accuracy of robotic systems in these settings. These findings will inform the development of more resilient robots for applications in nuclear facilities, outer space, and other radiation-exposed areas.

5.2 Achievements

The findings of this thesis have been distributed through several publications in peer-reviewed journals and international conferences. In addition, one manuscript has been submitted and is currently under review, while two others are in the process of being written and are expected to be submitted in the coming months. Tab. 5.1 provides a summary of all these contributions.

Table 5.1: Publications linked to the doctoral thesis.

Title	Type	Journal/Book	Link
Published			
Robust Fastener Detection Based on Force and Vision Algorithms in Robotic (Un)Screwing Applications	Journal	Sensors	1
The effect of ionizing radiation on robotic trajectory movement and electronic components	Journal	Nuclear Engineering and Technology	2
ROSE: Robot for Automatic Spacer Installation in Overhead Power Lines	Conference	Robot 2022 Fifth Iberian Robotics Conference	3
Forces Analysis on Robotics Screwing Tasks	Conference	Robot 2023 Fifth Iberian Robotics Conference	4
Under Review			
Design and evaluation of an advanced robotic bolting tool applied to IFMIF-DONES	Journal	Fusion Engineering and Design	
In process			
Characterization of BLDC Servomotor Parameters and Performance Evaluation with Commercial Driver Controllers (Internship collaboration)			
Framework for the Optimization of Robotic Arm Joint Torques Utilizing High-Torque External Tools (Internship collaboration)			

5.3 Future Work

Enhancing the reliability of the insertion of a socket into a fastener can be achieved through vision-based perception and robot position feedback. By monitoring force and position against set thresholds, one can identify collisions or successful fits. Although effective on rigid surfaces, adapting to diverse surfaces may require new sensors like load cells that measure force when the socket engages the fastener.

The entire procedure, from taking the picture to applying the proposed algorithms, takes 723 ms, with the majority of the time spent on capturing and downloading the image by TCP/IP protocol. To enhance this, cameras with a GIGEVision interface could be utilized, enabling real-time image transfer to the local PC.

The suggested algorithms are designed for 2D perpendicular surfaces. To adapt these algorithms for diverse environments, incorporating deep (RGBD), stereo cameras, or LIDAR sensors can be explored to determine the fastener's orientation on non-perpendicular surfaces.

The research highlights the necessity for thorough examination of torque dynamics to enable accurate torque predictions when subjected to external forces. Conducting a comprehensive dynamic simulation can facilitate the transfer of forces and torques to the tool design, leading to more dependable measurements of forces and torques at the robot tool flange.
The algorithms that have been proposed do not account for collisions between the robot, its tool, and the surrounding environment. To achieve enhanced performance and obtain optimal results, incorporating a collision-detection component would be advantageous.
Furthermore, the introduction of an improved cost function will be integral to refining the proposed algorithms for application on various types of industrial manipulators. In addition to these enhancements, a dynamic framework should be developed for commercial use. This framework would allow users to simply specify the task point, after which the system would autonomously adjust the robot's position to efficiently execute the task.

Enhance serial communication speed and synchronize slave devices based on the main clock frequency to maximize data throughput and detect bit changes in SRAM and EEPROM memories while the microcontroller operates in an ionizing radiation environment.
Develop an alert system that employs cheap active elements like digital temperature sensors to prevent severe damage to robots. Furthermore, use the data collected to implement Artificial Intelligence (AI) algorithms or neural networks for early fault detection, performance enhancement, adaptation to fluctuating conditions, intelligent decision making to extend system longevity, and avoidance of abnormal robot behaviors in radiation-exposed critical zones.
Conduct further testing involving the irradiation of various microcontrollers at different power supply levels, clock frequencies, and radiation doses to determine their limitations and maximum operating lifetime. In addition, examined how ionizing radiation affects the generation and frequency of PWM signals. A single bit alteration in the timer may trigger overflow flags, leading to incorrect set points for the servomotor, thus altering its position or speed. Furthermore, investigated the influence of ionizing radiation on the acquisition of ADC signals and the generation of DCA signals. For ADCs, assess how acquisition time and collected samples are impacted by SEEs, and consider the role of analog filters in signal smoothing. For DCAs, evaluate how filters and PWM are affected, and in the case of embedded DCAs, analyze the impact on operational amplifiers.
Furthermore, investigated the adaptation and practical application of the GEMMA guide for robots used in high-ionizing radiation environments. This guide aims to optimize the design of redundancy-optimized state machines to mitigate microcontroller failures. This approach will improve performance, minimize critical failures, and maximize the operational time of electronic and robotic systems under radiation conditions.

References

[1] A. Rodriguez. “Manipulation Techniques in Robotics: A Comprehensive Review and Future Directions”. In: (2023). DOI: 10.31219/osf.io/49pj2.

[2] M. Vukobratovic and A. Tuneski. “Contact control concepts in manipulation robotics/spl minus/an overview”. In: *IEEE Transactions on industrial electronics* 41.1 (1994), pp. 12–24.

[3] M. Suomalainen, Y. Karayiannidis, and V. Kyrki. “A survey of robot manipulation in contact”. In: *Robotics and Autonomous Systems* 156 (2022), p. 104224.

[4] M. Cutkosky. “On grasp choice, grasp models, and the design of hands for manufacturing tasks”. In: *IEEE Transactions on Robotics and Automation* 5.3 (1989), pp. 269–279. DOI: 10.1109/70.34763.

[5] A. Saudabayev and H. A. Varol. “Sensors for Robotic Hands: A Survey of State of the Art”. In: *IEEE Access* 3 (2015), pp. 1765–1782. DOI: 10.1109/ACCESS.2015.2482543.

[6] M. T. Mason. “Compliance and force control for computer controlled manipulators”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 11.6 (1981), pp. 418–432.

[7] F. L. Lewis, D. M. Dawson, and C. T. Abdallah. *Robot manipulator control: theory and practice*. CRC Press, 2003.

[8] Z. Shang, J. Li, Z. Zhang, G. Zhang, and S. Wang. “Self-Forcing Mechanism of the Braided Tube as a Robotic Gripper”. In: *Journal of Mechanisms and Robotics* (2019). DOI: 10.1115/1.4043686.

[9] H. Liu and D. Geng. “Kinematics Modeling and Grasping Experiment of Pneumatic Four-Finger Flexible Robotic Hand”. In: *Proceedings of the Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science* (2021). DOI: 10.1177/09544062211028266.

[10] E. Brown et al. “Universal robotic gripper based on the jamming of granular material”. In: *Proceedings of the National Academy of Sciences* 107 (2010), pp. 18809–18814. DOI: 10.1073/pnas.1003250107.

[11] “Tactile Sensors for Robotic Applications”. In: (2021). DOI: 10.3390/books978-3-0365-0425-4.

[12] M. Costanzo, G. D. Maria, C. Natale, and S. Pirozzi. “Design of a Force/Tactile Sensor for Robotic Grippers”. In: (2019). DOI: 10.3390/proceedings2019015031.

[13] T. Geng, M. H. Lee, and M. Hülse. “Transferring human grasping synergies to a robot”. In: *Mechatronics* 21 (2011), pp. 272–284. DOI: 10.1016/J.MECHATRONICS.2010.11.003.

[14] Z. Lu et al. “Visual-Tactile Robot Grasping Based on Human Skill Learning From Demonstrations Using a Wearable Parallel Hand Exoskeleton”. In: *IEEE Robotics and Automation Letters* 8 (2023), pp. 5384–5391. DOI: 10.1109/LRA.2023.3295296.

[15] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker. “Human-Inspired Robotic Grasp Control With Tactile Sensing”. In: *IEEE Transactions on Robotics* 27 (2011), pp. 1067–1079. DOI: 10.1109/TRO.2011.2162271.

[16] V. R. Garate, M. Pozzi, D. Prattichizzo, N. Tsagarakis, and A. Ajoudani. “Grasp Stiffness Control in Robotic Hands Through Coordinated Optimization of Pose and Joint Stiffness”. In: *IEEE Robotics and Automation Letters* 3 (2018), pp. 3952–3959. DOI: 10.1109/LRA.2018.2858271.

[17] A. Billard and D. Kragic. “Trends and challenges in robot manipulation”. In: *Science* 364 (2019). DOI: 10.1126/science.aat8414.

[18] B. Zhang, Y. Xie, J. Zhou, K. Wang, and Z. Zhang. “State-of-the-art robotic grippers, grasping and control strategies, as well as their applications in agricultural robots: A review”. In: *Comput. Electron. Agric.* 177 (2020), p. 105694. DOI: 10.1016/j.compag.2020.105694.

[19] T. Yoshikawa. “Multifingered robot hands: Control for grasping and manipulation”. In: *Annu. Rev. Control.* 34 (2010), pp. 199–208. DOI: 10.1016/j.arcontrol.2010.09.001.

[20] Y. Hu, G. Li, P. Yuan, C. Yang, and R. Song. “Development of Sensory-Motor Fusion-Based Manipulation and Grasping Control for a Robotic Hand-Eye System”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47 (2017), pp. 1169–1180. DOI: 10.1109/TSMC.2016.2560530.

[21] R. Calandra et al. "More Than a Feeling: Learning to Grasp and Regrasp Using Vision and Touch". In: *IEEE Robotics and Automation Letters* 3 (2018), pp. 3300–3307. DOI: 10.1109/LRA.2018.2852779.

[22] S.-Q. Ji, M.-B. Huang, and H.-P. Huang. "Robot Intelligent Grasp of Unknown Objects Based on Multi-Sensor Information". In: *Sensors (Basel, Switzerland)* 19 (2019). DOI: 10.3390/s19071595.

[23] J. Huang, Y. Sano, R. Hori, T. Hori, and T. Yabuta. "Method of Grasping Control by Computing Internal and External Impedances for Two Robot Fingers, and Its Application to Admittance Control of a Robot Hand-Arm System". In: *International Journal of Advanced Robotic Systems* (2015). DOI: 10.5772/61106.

[24] A. Naceri et al. "Multidigit Force Control During Unconstrained Grasping in Response to Object Perturbations". In: *Journal of Neurophysiology* (2017). DOI: 10.1152/jn.00546.2016.

[25] A. T. Miller and P. Allen. "Graspit! A versatile simulator for robotic grasping". In: *IEEE Robotics and Automation Magazine* 11 (2004), pp. 110–122. DOI: 10.1109/MRA.2004.1371616.

[26] R. Newbury et al. "Deep Learning Approaches to Grasp Synthesis: A Review". In: *IEEE Transactions on Robotics* 39 (2022), pp. 3994–4015. DOI: 10.1109/TRO.2023.3280597.

[27] S. Yu, D. Zhai, and Y. Xia. "A Novel Robotic Pushing and Grasping Method Based on Vision Transformer and Convolution." In: *IEEE transactions on neural networks and learning systems* PP (2023). DOI: 10.1109/tnnls.2023.3244186.

[28] B. Sundaralingam and T. Hermans. "Geometric in-Hand Regrasp Planning: Alternating Optimization of Finger Gaits and in-Grasp Manipulation". In: (2018). DOI: 10.1109/icra.2018.8460496.

[29] R. Kumar and S. Mukherjee. "Algorithmic Fingertip Repositioning for Enhanced in-Hand Manipulation of the Objects". In: *Journal of Mechanisms and Robotics* (2021). DOI: 10.1115/1.4052323.

[30] Y. Narang et al. "Factory: Fast Contact for Robotic Assembly". In: (2022). DOI: 10.15607/rss.2022.xviii.035.

[31] M. Polishchuk and M. Tkach. "Experimental Studies of Robotic Assembly of Precision Parts". In: *Fme Transaction* (2021). DOI: 10.5937/fme2101044p.

[32] H. Yin, A. Varava, and D. Kragic. "Modeling, learning, perception, and control methods for deformable object manipulation". In: *Science Robotics* 6 (2021). DOI: 10.1126/scirobotics.abd8803.

[33] Y. C. Hou, K. Sahari, and D. N. How. “A review on modeling of flexible deformable object for dexterous robotic manipulation”. In: *International Journal of Advanced Robotic Systems* 16 (2019). DOI: 10.1177/1729881419848894.

[34] G. B. Mahanta, B. Deepak, and B. Biswal. “Application of soft computing methods in robotic grasping: A state-of-the-art survey”. In: *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering* 236 (2021), pp. 712–726. DOI: 10.1177/09544089211039977.

[35] I. KAMBAROV. “From Human to Robot Interaction Towards Human to Robot Communication in Assembly Systems”. In: *The Eurasia Proceedings of Science Technology Engineering and Mathematics* (2023). DOI: 10.55549/epstem.1365802.

[36] S. Liu, L. Wang, and X. V. Wang. “Function Block-Based Multimodal Control for Symbiotic Human–Robot Collaborative Assembly”. In: *Journal of Manufacturing Science and Engineering* (2021). DOI: 10.1115/1.4050187.

[37] Z. Zhu, H. Hu, and D. Gu. “Robot Performing Peg-in-Hole Operations by Learning From Human Demonstration”. In: (2018). DOI: 10.1109/ceec.2018.8674203.

[38] F. Li et al. “Manipulation Skill Acquisition for Robotic Assembly Based on Multi-Modal Information Description”. In: *Ieee Access* (2020). DOI: 10.1109/access.2019.2934174.

[39] B. Nazeer. “Contact-Based Pose Detection Method for Small Components to Optimise the Digital Twin-Driven Robotic Assembly Process”. In: *Matec Web of Conferences* (2024). DOI: 10.1051/matecconf/202440113008.

[40] A. Bilberg and A. A. Malik. “Digital Twin Driven Human–robot Collaborative Assembly”. In: *Cirp Annals* (2019). DOI: 10.1016/j.cirp.2019.04.011.

[41] M. Q. Mohammed et al. “Review of Learning-Based Robotic Manipulation in Cluttered Environments”. In: *Sensors (Basel, Switzerland)* 22 (2022). DOI: 10.3390/s22207938.

[42] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber. “A Survey on Learning-Based Robotic Grasping”. In: *Current Robotics Reports* 1 (2020), pp. 239–249. DOI: 10.1007/s43154-020-00021-6.

[43] Z. Xue, J. Liu, C. Wu, and Y. Tong. “Review of in-space assembly technologies”. In: *Chinese Journal of Aeronautics* (2020). DOI: 10.1016/j.cja.2020.09.043.

[44] A. Brunete et al. “Current trends in reconfigurable modular robots design”. In: *International Journal of Advanced Robotic Systems* 14 (2017). DOI: 10.1177/1729881417710457.

[45] C. Yu and P. Wang. “Dexterous Manipulation for Multi-Fingered Robotic Hands With Reinforcement Learning: A Review”. In: *Frontiers in Neurorobotics* (2022). DOI: 10.3389/fnbot.2022.861825.

[46] F. R. Hogan, J. Ballester, S. Dong, and A. Rodriguez. “Tactile Dexterity: Manipulation Primitives With Tactile Feedback”. In: (2020). DOI: 10.48550/arxiv.2002.03236.

[47] M. Q. Ta, G. D. Khan, X. Li, and C. C. Cheah. “Laser-Actuated Multi-Fingered Hand for Dexterous Manipulation of Micro-Objects”. In: *Ieee Access* (2023). DOI: 10.1109/access.2023.3261558.

[48] S. K. Sampath, N. Wang, H. Wu, and C. Yang. “Review on Human-like Robot Manipulation Using Dexterous Hands”. In: *Cognitive Computation and Systems* (2023). DOI: 10.1049/ccs2.12073.

[49] N. C. Dafle et al. “Extrinsic Dexterity: In-Hand Manipulation With External Forces”. In: (2014). DOI: 10.1109/icra.2014.6907062.

[50] C. Jara, J. Pomares, F. Candelas, and F. Torres. “Control framework for dexterous manipulation using dynamic visual servoing and tactile sensors’ feedback”. In: *Sensors* 14 (1 2014), pp. 1787–1804. DOI: 10.3390/s140101787.

[51] Z. Xia, Z. Deng, B. Fang, Y. Yang, and F. Sun. “A Review on Sensory Perception for Dexterous Robotic Manipulation”. In: *International Journal of Advanced Robotic Systems* (2022). DOI: 10.1177/17298806221095974.

[52] Y. C. Hou, K. S. Mohamed Sahari, and D. N. Tze How. “A Review on Modeling of Flexible Deformable Object for Dexterous Robotic Manipulation”. In: *International Journal of Advanced Robotic Systems* (2019). DOI: 10.1177/1729881419848894.

[53] T. Cheng, G. Zhang, J. Sun, T. Zhang, and F. Du. “Dexterity Enhancement of Continuum Robot for Natural Orifice Transluminal Endoscopic Surgery in the Dual-manipulator Collaborative Space”. In: *International Journal of Medical Robotics and Computer Assisted Surgery* (2023). DOI: 10.1002/rcs.2516.

[54] E. Gerena et al. “Tele–Robotic Platform for Dexterous Optical Single-Cell Manipulation”. In: *Micromachines* (2019). DOI: 10.3390/mi10100677.

[55] C.-Y. Weng, Q. Yuan, Z. J. Lim, and I.-M. Chen. *Applications of Light-Weight Wearable Devices to Online Programming of Industrial Dual-Arm Robots*. 2020. DOI: 10.1142/s2301385020500144.

[56] C. H. Kim, K. H. Mak, and J. Seo. “Planning for dexterous ungrasping: secure ungrasping through dexterous manipulation”. In: (2021). DOI: 10.48550/arxiv.2108.13580.

[57] O. Olukayode. “Design and Analysis of a Novel Light-Weight, Linear Actuator Driven Flexible PET Strips Robotic Gripper”. In: *Uniosun Journal of Engineering and Environmental Sciences* (2023). DOI: 10.36108/ujees/3202.50.0170.

[58] S. Terrile, J. Miguelañez, and A. Barrientos. “A Soft Haptic Glove Actuated with Shape Memory Alloy and Flexible Stretch Sensors”. In: *Sensors* 21.16 (2021). ISSN: 1424-8220. DOI: 10.3390/s21165278. URL: https://www.mdpi.com/1424-8220/21/16/5278.

[59] S. Zaidi, M. Maselli, C. Laschi, and M. Cianchetti. “Actuation Technologies for Soft Robot Grippers and Manipulators: A Review”. In: *Current Robotics Reports* (2021). DOI: 10.1007/s43154-021-00054-5.

[60] G. L. Goh et al. “3D Printing of Robotic Soft Grippers: Toward Smart Actuation and Sensing”. In: *Advanced Materials Technologies* (2022). DOI: 10.1002/admt.202101672.

[61] S. Terrile, M. Argüelles, and A. Barrientos. “Comparison of Different Technologies for Soft Robotics Grippers”. In: *Sensors* 21.9 (2021). ISSN: 1424-8220. DOI: 10.3390/s21093253. URL: https://www.mdpi.com/1424-8220/21/9/3253.

[62] R. Jain, S. Saha, and S. Majumder. “Development of Piezoelectric Actuator Based Compliant Micro Gripper for Robotic Peg-in-Hole Assembly”. In: (2013). DOI: 10.1109/robio.2013.6739689.

[63] R. Jain, S. Majumder, B. Ghosh, and S. Saha. “Micro Manipulation by a Compliant Piezoelectric Micro Gripper Towards Robotic Micro Assembly”. In: *International Journal of Mechatronics and Manufacturing Systems* (2016). DOI: 10.1504/ijmms.2016.075402.

[64] E. M. Faidallah, S. A. Maged, M. Alsamanty, Y. Hossameldin, and S. Abdrabbo. “Design and Control Implementation of Adaptive Soft Gripper Installed on KUKA KR6 R900 Robot Arm”. In: *Engineering Research Journal - Faculty of Engineering (Shoubra)* (2022). DOI: 10.21608/erjsh.2022.156200.1073.

[65] “A Review on Technologies in Robotic Gripper”. In: *International Journal of Advanced Engineering and Nano Technology* (2023). DOI: 10.35940/ijaent.c7232.0511523.

[66] R. Torres and N. M. Fonseca Ferreira. “Robotic Manipulation in the Ceramic Industry”. In: *Electronics* (2022). DOI: 10.3390/electronics11244180.

[67] R. Movassagh-Khaniki, N. Hassanzadeh, A. Makhal, and A. Perez-Gracia. “Design of a Multi-Palm Robotic Hand for Assembly Tasks”. In: (2016). DOI: 10.1115/detc2016-59980.

[68] B. B. Biswal, P. K. Parida, and K. C. Pati. “Kinematic Analysis of a Dexterous Hand”. In: *Advanced Materials Research* (2012). DOI: 10.4028/www.scientific.net/amr.433-440.754.

[69] J. B. Borras, R. Heudorfer, S. Rader, P. K. Kaiser, and T. Asfour. “The KIT Swiss Knife Gripper for Disassembly Tasks: A Multi-Functional Gripper for Bimanual Manipulation With a Single Arm”. In: (2018). DOI: 10.1109/iros.2018.8593567.

[70] F. Chen, L. Carbonari, C. Canali, M. D’Imperio, and F. Cannella. “Design of a Novel Dexterous Robotic Gripper for in-Hand Twisting and Positioning Within Assembly Automation”. In: *Assembly Automation* (2015). DOI: 10.1108/aa-05-2015-046.

[71] L. Chin, F. Barscevicius, J. Lipton, and D. Rus. “Multiplexed Manipulation: Versatile Multimodal Grasping via a Hybrid Soft Gripper”. In: (2020). DOI: 10.1109/icra40945.2020.9196626.

[72] Q. Vu, M. Kuzov, and A. Ronzhin. “Hierarchical Classification of Robotic Grippers Applied for Agricultural Object Manipulations”. In: *Matec Web of Conferences* (2018). DOI: 10.1051/matecconf/201816103015.

[73] D. Vu, O. Y. Solenaya, A. Ronzhin, et al. “Over-view of robotic grippers for physical manipulation with agricultural products”. In: *Tractors and Agricultural Machinery* 84.12 (2017), pp. 14–24.

[74] I. Bonilla et al. “A Vision-Based, Impedance Control Strategy for Industrial Robot Manipulators”. In: (2010). DOI: 10.1109/coase.2010.5584481.

[75] K. Zhou, S. Wang, R. Zhou, and B. Wang. “Admittance Control Design and System Testing of Industrial Robot Polishing Operation”. In: *Proceedings of the Institution of Mechanical Engineers Part B Journal of Engineering Manufacture* (2022). DOI: 10.1177/09544054221136513.

[76] G. Rigatos and M. Abbaszadeh. “Nonlinear optimal control for a five-link parallel robotic manipulator”. In: *Journal of Vibration and Control* 29 (3-4 2022), pp. 714–735. DOI: 10.1177/10775463211051449.

[77] G. Rigatos, M. Abbaszadeh, K. Busawon, and J. Pomares. “Nonlinear optimal control for a 4-dof scara robotic manipulator”. In: *Robotica* 41 (8 2023), pp. 2397–2450. DOI: 10.1017/s0263574723000450.

[78] Y. Ning, Y. Liu, F. Xi, K. Huang, and B. Li. “Human-Robot Interaction Control for Robot Driven by Variable Stiffness Actuator With Force Self-Sensing”. In: *Ieee Access* (2021). DOI: 10.1109/access.2020.3048418.

[79] S. Wen, J. Zhu, X. Li, A. Rad, and X. Chen. “End-Point Contact Force Control With Quantitative Feedback Theory for Mobile Robots”. In: *International Journal of Advanced Robotic Systems* (2012). DOI: 10.5772/53742.

[80] M. P. Polverini et al. “Performance Improvement of Implicit Integral Robot Force Control Through Constraint-Based Optimization”. In: (2016). DOI: 10.1109/iros.2016.7759518.

[81] A. Bajo and N. Simaan. “Hybrid Motion/Force Control of Multi-Backbone Continuum Robots”. In: *The International Journal of Robotics Research* (2015). DOI: 10.1177/0278364915584806.

[82] G. Zheng, J. Lei, L. Hu, and L. Zhang. “Adaptive Variable Impedance Position/Force Tracking Control of Fracture Reduction Robot”. In: *International Journal of Medical Robotics and Computer Assisted Surgery* (2022). DOI: 10.1002/rcs.2469.

[83] A. Ellery. “Tutorial review of bio-inspired approaches to robotic manipulation for space debris salvage”. In: *Biomimetics* 5 (2 2020), p. 19. DOI: 10.3390/biomimetics5020019.

[84] L. L. Paige Gallant. “An Intelligent Force Control Strategy for Soft Robotic Grippers”. In: (2023). DOI: 10.17118/11143/21094.

[85] R. Mei and Y. Cheng-jiang. “Adaptive neural output feedback control for uncertain robot manipulators with input saturation”. In: *Complexity* 2017 (2017), pp. 1–12. DOI: 10.1155/2017/7413642.

[86] C. Sun, W. He, and J. Hong. “Neural network control of a flexible robotic manipulator using the lumped spring-mass model”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47 (8 2017), pp. 1863–1874. DOI: 10.1109/tsmc.2016.2562506.

[87] S. H. Turlapati and D. Campolo. “Towards Haptic-Based Dual-Arm Manipulation”. In: *Sensors* (2022). DOI: 10.3390/s23010376.

[88] M. T. Shahria et al. “A Comprehensive Review of Vision-Based Robotic Applications: Current State, Components, Approaches, Barriers, and Potential Solutions”. In: *Robotics* (2022). DOI: 10.3390/robotics11060139.

[89] J. Wang, L. Li, and P. Xu. “Visual sensing and depth perception for welding robots and their industrial applications”. In: *Sensors* 23 (24 2023), p. 9700. DOI: 10.3390/s23249700.

[90] Z. Jia, M. Lin, Z. Chen, and S. Jian. “Vision-Based Robot Manipulation Learning via Human Demonstrations”. In: (2020). DOI: 10.48550/arxiv.2003.00385.

[91] V. Batra and V. Kumar. “Real-Time Object Detection and Localization for Vision-Based Robot Manipulator”. In: *Sn Computer Science* (2021). DOI: 10.1007/s42979-021-00561-4.

[92] G. Pasquale, C. Ciliberto, F. Odone, L. Rosasco, and L. Natale. “Real-World Object Recognition With Off-the-Shelf Deep Conv Nets: How Many Objects Can iCub Learn?” In: (2015). DOI: 10.48550/arxiv.1504.03154.

[93] E. L. Gouveia. “Implementing a Vision-Based ROS Package for Reliable Part Localization and Displacement From Conveyor Belts”. In: *Journal of Manufacturing and Materials Processing* (2024). DOI: 10.3390/jmmp8050218.

[94] J. Bimbo, P. Kormushev, K. Althoefer, and H. Liu. “Global Estimation of an Object’s Pose Using Tactile Sensing”. In: *Advanced Robotics* (2015). DOI: 10.1080/01691864.2014.1002531.

[95] G. Rigatos. “Applications of Machine Vision to Industrial Systems”. In: (2011). DOI: 10.1007/978-3-642-17875-7_16.

[96] L. Natale and G. Cannata. “Tactile Sensing”. In: (2018). DOI: 10.1007/978-94-007-6046-2_110.

[97] V. Ciobanu, D. Popescu, and A. Petrescu. “Point of Contact Location and Normal Force Estimation Using Biomimetical Tactile Sensors”. In: (2014). DOI: 10.1109/cisis.2014.52.

[98] M. S. Neto et al. “A Soft Tactile Sensor Based on Magnetics and Hybrid Flexible-Rigid Electronics”. In: *Sensors* (2021). DOI: 10.3390/s21155098.

[99] M. Lambeta et al. “DIGIT: A Novel Design for a Low-Cost Compact High-Resolution Tactile Sensor With Application to in-Hand Manipulation”. In: *Ieee Robotics and Automation Letters* (2020). DOI: 10.1109/lra.2020.2977257.

[100] N. F. Lepora. “Soft Biomimetic Optical Tactile Sensing With the TacTip: A Review”. In: (2021). DOI: 10.36227/techrxiv.14703051.

[101] L. Zhu, Y. Wang, D. Mei, and C. Jiang. “Development of fully flexible tactile pressure sensor with bilayer interlaced bumps for robotic grasping applications”. In: *Micromachines* 11 (8 2020), p. 770. DOI: 10.3390/mi11080770.

[102] J.-H. Yang, S.-Y. Kim, and S.-C. Lim. “Effects of Sensing Tactile Arrays, Shear Force, and Proprioception of Robot on Texture Recognition”. In: *Sensors* (2023). DOI: 10.3390/s23063201.

[103] G. Li and R. Zhu. “A Multisensory Tactile System for Robotic Hands to Recognize Objects”. In: *Advanced Materials Technologies* (2019). DOI: 10.1002/admt.201900602.

[104] A. Melnik et al. “Using Tactile Sensing to Improve the Sample Efficiency and Performance of Deep Deterministic Policy Gradients for Simulated in-Hand Manipulation Tasks”. In: *Frontiers in Robotics and Ai* (2021). DOI: 10.3389/frobt.2021.538773.

[105] R. Smith et al. “Robotic development for the nuclear environment: challenges and strategy”. In: *Robotics* 9.4 (2020), p. 94.

[106] K. Zhang, C. Hutson, J. Knighton, G. Herrmann, and T. Scott. “Radiation tolerance testing methodology of robotic manipulator prior to nuclear waste handling”. In: *Frontiers in Robotics and AI* 7 (2020), p. 6.

[107] Y. R. Petillot et al. “Underwater robots: From remotely operated vehicles to intervention-autonomous underwater vehicles”. In: *IEEE Robotics & Automation Magazine* 26.2 (2019), pp. 94–101.

[108] S. Coloma, C. Martinez, B. C. Yalçın, and M. A. Olivares-Mendez. “Enhancing rover teleoperation on the moon with proprioceptive sensors and machine learning techniques”. In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 11434–11441.

[109] M. D. Anand, T. Selvaraj, and S. Kumanan. “Fault detection and fault tolerance methods for industrial robot manipulator based on hybrid intelligent approach”. In: *Advances in Production Engineering & Management* 7.4 (2012), pp. 225–236.

[110] O. Porges, D. Leidner, and M. A. Roa. “Planning Fail-Safe Trajectories for Space Robotic Arms”. In: *Frontiers in Robotics and AI* 8 (2021), p. 710021.

[111] K. A. Szczurek, R. Cittadini, R. M. Prades, E. Matheson, and M. Di Castro. “Enhanced human–robot interface with operator physiological parameters monitoring and 3d mixed reality”. In: *IEEE Access* 11 (2023), pp. 39555–39576.

[112] M. Ferre, S. Coloma, J. Breñosa, and L. Rubio. “Telerobotics Requirements for Remote Handling in Nuclear Facilities”. In: (July 2017). URL: https://www.osti.gov/biblio/22802503.

[113] K. A. Szczurek et al. “From 2D to 3D mixed reality human-robot interface in hazardous robotic interventions with the use of redundant mobile manipulator”. In: (2021).

[114] A. Frantz, S. COLOMA CHACON, and M. A. OLIVARES MENDEZ. “Enhanced Teleoperation for Lunar Exploration Missions through eXtended Reality”. In: *17th Symposium on Advanced Space Technologies in Robotics and Automation*. 2023.

[115] J. Humphreys, C. Peers, J. Li, Y. Wan, and C. Zhou. "High utility teleoperation framework for legged manipulators through leveraging whole-body control". In: *Journal of Intelligent & Robotic Systems* 108.3 (2023), p. 57.

[116] Z. Gong et al. "A soft manipulator for efficient delicate grasping in shallow water: Modeling, control, and real-world experiments". In: *The International Journal of Robotics Research* 40.1 (2021), pp. 449–469.

[117] L. Ye et al. "HeterBot: A heterogeneous mobile manipulation robot for versatile operation". In: *IET Cyber-Systems and Robotics* 5.1 (2023), e12068.

[118] Y. Chen et al. "Intelligent power distribution live-line operation robot systems based on stereo camera". In: *High Voltage* 8.6 (2023), pp. 1306–1318.

[119] V. Girbes-Juan, V. Schettino, Y. Demiris, and J. Tornero. "Haptic and visual feedback assistance for dual-arm robot teleoperation in surface conditioning tasks". In: *IEEE Transactions on Haptics* 14.1 (2020), pp. 44–56.

[120] M. S. Sakore, M. S. Mule, M. A. Dorle, and P. K. Dhenge. "Electrical Inspection and Autometed Robot for Electrical Transmission Line Inspection". In: *International Journal for Research in Applied Science and Engineering Technology* (2023). DOI: `10.22214/ijraset.2023.50619`.

[121] M. J. Ferrer. "Fabrication and Evaluation of a Clamshell Line Inspection Robot". In: *Asean Engineering Journal* (2024). DOI: `10.11113/aej.v14.20822`.

[122] R. S. Gonçalves and J. C. Mendes Carvalho. "Review and Latest Trends in Mobile Robots Used on Power Transmission Lines". In: *International Journal of Advanced Robotic Systems* (2013). DOI: `10.5772/56791`.

[123] J. Zhou et al. "A live-line emergency power restoration robot for 10kv overhead distribution networks". In: *Advances in Transdisciplinary Engineering* (2023). DOI: `10.3233/atde230526`.

[124] Y. Wang, C. Yuan, and Y. Zhai. "Mechanism Design and Analysis of a New Overhead Transmission Line Inspection Robot". In: *Xibei Gongye Daxue Xuebao/Journal of Northwestern Polytechnical University* (2020). DOI: `10.1051/jnwpu/20203851105`.

[125] Y. Zeng. "A Novel Autonomous Landing Method for Flying–Walking Power Line Inspection Robots Based on Prior Structure Data". In: *Applied Sciences* (2023). DOI: `10.3390/app13179544`.

[126] T. Gao, Z. Liu, and C. Ye. "A Novel Unit Mechanism for Serial Head-Tail Alternatively Supported Robot". In: *Proceedings of the Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science* (2020). DOI: `10.1177/0954406220978683`.

[127] A. B. Alhassan et al. "Investigation of Aerodynamic Stability of a Lightweight Dual-Arm Power Transmission Line Inspection Robot Under the Influence of Wind". In: *Mathematical Problems in Engineering* (2019). DOI: `10.1155/2019/2139462`.

[128] L. Chen, Y. Sun, and C. Fang. "UAV Power Line Inspection Based on Multi-Sensor Fusion". In: (2022). DOI: `10.1117/12.2660259`.

[129] X. Jiang et al. "Optimal Design of MFL Sensor for Detecting Broken Steel Strands in Overhead Power Line". In: *Progress in Electromagnetics Research* (2011). DOI: `10.2528/pier11072711`.

[130] B.-E. Byambasuren, D. Kim, M. Oyun-Erdene, C. Bold, and J. Yura. "Inspection Robot Based Mobile Sensing and Power Line Tracking for Smart Grid". In: *Sensors* (2016). DOI: `10.3390/s16020250`.

[131] Y. Tan, Y. Wang, X. Li, and M. Li. "Application of robot automation technology based on machine assisted and artificial intelligence in distribution network overhead line engineering". In: *EAI Endorsed Transactions on Energy Web* 10 (2023). DOI: `10.4108/ew.3717`.

[132] B. Todd and S. Uznanski. "Radiation risks and mitigation in electronic systems". In: *arXiv preprint arXiv:1607.01573* (2016).

[133] J. D. Cressler et al. *Extreme environment electronics*. CRC Press, 2017.

[134] P. R. Villa et al. "Analysis of single-event upsets in a Microsemi ProAsic3E FPGA". In: *2017 18th IEEE Latin American Test Symposium (LATS)*. IEEE. 2017, pp. 1–4.

[135] F. Faccio et al. "COTS for the LHC radiation environment: the rules of the game". In: *LEB 2000 Conference Krakow, Book of Abstracts*. 2000, p. 50.

[136] B. Todd et al. "Radiation risks and mitigation in electronic systems". In: *arXiv preprint arXiv:1607.01573* (2016).

[137] Z. Diggins et al. "Total-ionizing-dose induced timing window violations in CMOS microcontrollers". In: *IEEE Transactions on Nuclear Science* 61.6 (2014), pp. 2979–2984.

[138] F. Faccio et al. "Radiation effects in the electronics for CMS". In: *Tutorial Script, CERN* (1999).

[139] E. Petersen. *Single event effects in aerospace*. John Wiley & Sons, 2011.

[140] S. J. Gaul, N. van Vonno, S. H. Voldman, and W. H. Morris. *Integrated Circuit Design for Radiation Environments*. John Wiley & Sons, 2019.

[141] R. Smith, E. Cucco, and C. Fairbairn. “Robotic development for the nuclear environment: challenges and strategy”. In: *Robotics* 9.4 (2020), p. 94.

[142] I. Tsitsimpelis, C. J. Taylor, B. Lennox, and M. J. Joyce. “A review of ground-based robotic systems for the characterization of nuclear environments”. In: *Progress in Nuclear Energy* 111 (2019), pp. 109–124.

[143] M. S. Muktadir et al. “Development of a radiation detection measuring and monitoring mobile facility with wireless autonomous robots”. In: *The Proceedings of the International Conference on Nuclear Engineering (ICONE)*. 2019.

[144] B. Bird et al. “Vega—A small, low cost, ground robot for nuclear decommissioning”. In: *Journal of Field Robotics* (2021).

[145] B. Bird et al. “A robot to monitor nuclear facilities”. In: *IEEE ROBOTICS & AUTOMATION MAGAZIN* (2019), pp. 1070–9932.

[146] K. Zhang, C. Hutson, J. Knighton, G. Herrmann, and T. Scott. “Radiation tolerance testing methodology of robotic manipulator prior to nuclear waste handling”. In: *Frontiers in Robotics and AI* (2020), p. 6.

[147] F. Chen et al. “The radiation tolerance design and test of a tele-controlled robot”. In: *Nuclear Electronics and Detection Technology* 36.2 (2016), pp. 121–124.

[148] S. Li, H. Wang, Y. F. Zheng, and L. Cao. “Radiation effect on the performance of robot manipulator”. In: *International Journal of Mechatronics and Automation* 6.1 (2017), pp. 10–19.

[149] S. J. Gaul, N. van Vonno, S. H. Voldman, and W. H. Morris. *Integrated Circuit Design for Radiation Environments*. John Wiley & Sons, 2019.

[150] M. Nancekievill, S. Watson, P. Green, and B. Lennox. “Radiation tolerance of commercial-off-the-shelf components deployed in an underground nuclear decommissioning embedded system”. In: *2016 IEEE Radiation Effects Data Workshop (REDW)*. IEEE. 2016, pp. 1–5.

[151] J. Navarrete et al. “Irradiation Measurements of the Hitachi H8S/2357 MCU”. In: (2003).

[152] T. Fried et al. “Radiation testing of low cost, commercial off the shelf microcontroller board”. In: *Nuclear Engineering and Technology* 53.10 (2021), pp. 3335–3343.

[153] Z. J. Diggins et al. “Total-ionizing-dose induced timing window violations in CMOS microcontrollers”. In: *IEEE Transactions on Nuclear Science* 61.6 (2014), pp. 2979–2984.

[154] S. Keller, A. J. Martin, and C. Moore. “DD1: A QDI, Radiation-Hard-by-Design, Near-Threshold 18uW/MIPS Microcontroller in 40nm Bulk CMOS”. In: *2015 21st IEEE International Symposium on Asynchronous Circuits and Systems*. IEEE. 2015, pp. 37–44.

[155] E. Petersen. *Single event effects in aerospace*. John Wiley & Sons, 2011.

[156] G. Foucard. “Handbook of Mitigation techniques against Radiation Effects for ASICs and FPGAs”. In: *Online]. Jan* (2012).

[157] C. Peng et al. “Radiation-hardened 14T SRAM bitcell with speed and power optimized for space application”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27.2 (2018), pp. 407–415.

[158] F. K. Reed, N. Ezell, M. N. Ericson, and C. L. Britton Jr. *Radiation Hardened Electronics for Reactor Environments*. Tech. rep. Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), 2020.

[159] J. Pontes, N. Calazans, and P. Vivet. “Adding temporal redundancy to delay insensitive codes to mitigate single event effects”. In: *2012 IEEE 18th International Symposium on Asynchronous Circuits and Systems*. IEEE. 2012, pp. 142–149.

[160] R. Velazco and F. J. Franco. “Single event effects on digital integrated circuits: Origins and mitigation techniques”. In: *2007 IEEE International Symposium on Industrial Electronics*. IEEE. 2007, pp. 3322–3327.

[161] S. Coloma Chacón. “Methods, strategies and application cases for robotic telemanipulation in hazardous environments”. PhD thesis. Industriales, 2020.

[162] P. Espinosa Peralta et al. “ROSE: Robot for Automatic Spacer Installation in Overhead Power Lines”. In: *ROBOT2022: Fifth Iberian Robotics Conference*. Ed. by D. Tardioli, V. Matellán, G. Heredia, M. F. Silva, and L. Marques. Cham: Springer International Publishing, 2023, pp. 325–337. ISBN: 978-3-031-21062-4.

[163] Y. Zhong et al. “Development of A Robot System Performing Maintenance Tasks on High-Voltage Power Transmission Lines”. In: *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2019, pp. 1344–1349. DOI: `10.1109/ROBIO49542.2019.8961863`.

[164] X. Wei et al. “Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study”. In: *Engineering Applications of Artificial Intelligence* 80 (2019), pp. 66–81. ISSN: 0952-1976. DOI: `https://doi.org/10.1016/j.engappai.2019.01.008`. URL: `https://www.sciencedirect.com/science/article/pii/S0952197619300089`.

[165] G. De Ruvo et al. “A FPGA-based architecture for automatic hexagonal bolts detection in railway maintenance”. In: *Seventh International Workshop on Computer Architecture for Machine Perception (CAMP’05)*. 2005, pp. 219–224. DOI: 10.1109/CAMP.2005.4.

[166] P. De Ruvo, A. Distante, E. Stella, and F. Marino. “A GPU-based vision system for real time detection of fastening elements in railway inspection”. In: *2009 16th IEEE International Conference on Image Processing (ICIP)*. 2009, pp. 2333–2336. DOI: 10.1109/ICIP.2009.5414438.

[167] R. Li et al. “Unfastening of Hexagonal Headed Screws by a Collaborative Robot”. In: *IEEE Transactions on Automation Science and Engineering* 17.3 (2020), pp. 1455–1468. DOI: 10.1109/TASE.2019.2958712.

[168] R. Gerbers, M. Mücke, F. Dietrich, and K. Dröder. “Simplifying robot tools by taking advantage of sensor integration in human collaboration robots”. In: *Procedia CIRP* 44 (2016), pp. 287–292.

[169] A. Rastegarpanah, R. Ner, R. Stolkin, and N. Marturi. “Nut Unfastening by Robotic Surface Exploration”. In: *Robotics* 10.3 (2021). ISSN: 2218-6581. DOI: 10.3390/robotics10030107. URL: https://www.mdpi.com/2218-6581/10/3/107.

[170] M. Sága et al. “Case study: Performance analysis and development of robotized screwing application with integrated vision sensing system for automotive industry”. In: *International Journal of Advanced Robotic Systems* 17.3 (2020), p. 1729881420923997. DOI: 10.1177/1729881420923997. eprint: https://doi.org/10.1177/1729881420923997. URL: https://doi.org/10.1177/1729881420923997.

[171] C. Wei-Yu et al. “Real-Time Instance Segmentation of Metal Screw Defects Based on Deep Learning Approach”. In: *Measurement Science Review* 22.3 (2022), pp. 107–111.

[172] H. Wei, M. Su, and Y. Guan. “Semi-autonomous Robotic Manipulation by Tele-Operation with Master-Slave Robots and Autonomy Based on Vision and Force Sensing”. In: *Intelligent Robotics and Applications*. Ed. by X.-J. Liu, Z. Nie, J. Yu, F. Xie, and R. Song. Cham: Springer International Publishing, 2021, pp. 36–46.

[173] W. H. Chen, K. Wegener, and F. Dietrich. “A robot assistant for unscrewing in hybrid human-robot disassembly”. In: *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*. 2014, pp. 536–541. DOI: 10.1109/ROBIO.2014.7090386.

[174] N. M. DiFilippo and M. K. Jouaneh. “A System Combining Force and Vision Sensing for Automated Screw Removal on Laptops”. In: *IEEE Transactions on Automation Science and Engineering* 15.2 (2018), pp. 887–895. DOI: 10.1109/TASE.2017.2679720.

[175] X. Zhang, K. Eltouny, X. Liang, and S. Behdad. “Automatic Screw Detection and Tool Recommendation System for Robotic Disassembly”. In: *Journal of Manufac-*

turing Science and Engineering 145.3 (Dec. 2022). 031008. ISSN: 1087-1357. DOI: 10.1115/1.4056074. eprint: https://asmedigitalcollection.asme.org/manufacturingscience/article-pdf/145/3/031008/6953322/manu_145_3_031008.pdf. URL: https://doi.org/10.1115/1.4056074.

[176] J. Huang et al. "Strategies for Dealing with Problems in Robotised Unscrewing Operations". In: *Smart Technologies for Precision Assembly.* Ed. by S. Ratchev. Cham: Springer International Publishing, 2021, pp. 93–107. ISBN: 978-3-030-72632-4.

[177] Z. Peng, C. Wang, Z. Ma, and H. Liu. "A multifeature hierarchical locating algorithm for hexagon nut of railway fasteners". In: *IEEE Transactions on Instrumentation and Measurement* 69.3 (2019), pp. 693–699.

[178] E. Yildiz and F. Wörgötter. "DCNN-based Screw Classification in Automated Disassembly Processes." In: *ROBOVIS.* 2020, pp. 61–68.

[179] X. Li et al. "Accurate screw detection method based on faster R-CNN and rotation edge similarity for automatic screw disassembly". In: *International Journal of Computer Integrated Manufacturing* 34.11 (2021), pp. 1177–1195. DOI: 10.1080/0951192X.2021.1963476. eprint: https://doi.org/10.1080/0951192X.2021.1963476. URL: https://doi.org/10.1080/0951192X.2021.1963476.

[180] Y. Feng et al. "Towards Robust Part-aware Instance Segmentation for Industrial Bin Picking". In: *arXiv preprint arXiv:2203.02767* (2022).

[181] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention.* Springer. 2015, pp. 234–241.

[182] W. Abdulla. *Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. 2017.* 2020.

[183] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2018, pp. 4510–4520.

[184] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee. "Yolact: Real-time instance segmentation". In: *Proceedings of the IEEE/CVF international conference on computer vision.* 2019, pp. 9157–9166.

[185] S. Suzuki et al. "Topological structural analysis of digitized binary images by border following". In: *Computer vision, graphics, and image processing* 30.1 (1985), pp. 32–46.

[186] J. Sklansky. "Finding the convex hull of a simple polygon". In: *Pattern Recognition Letters* 1.2 (1982), pp. 79–83.

[187] J. Matas, C. Galambos, and J. Kittler. “Robust detection of lines using the progressive probabilistic hough transform”. In: *Computer vision and image understanding* 78.1 (2000), pp. 119–137.

[188] M. Rosenberger et al. “EMVA 1288 Camera characterisation and the influences of radiometric camera characteristics on geometric measurements”. In: *acta imeko* 5.4 (2016), p. 81.

[189] B. Jähne. “Emva 1288 standard for machine vision: Objective specification of vital camera data”. In: *Optik & Photonik* 5.1 (2010), pp. 53–54.

[190] A. Distante, C. Distante, W. Distante, and Wheeler. *Handbook of Image Processing and Computer Vision*. 1st ed. Springer, 2020, pp. 97–193. ISBN: 978-3-030-38147-9. DOI: https://doi.org/10.1007/978-3-030-38148-6.

[191] S. J. Sangwine and R. E. Horne. *The Colour Image Processing Handbook*. 1st ed. Springer Science & Business Media, 1998, pp. 38–125. ISBN: 978-1-4613-7647-7. DOI: https://doi.org/10.1007/978-1-4615-5779-1.

[192] G. Hastings and A. Rubin. “Colour spaces-a review of historic and modern colour models”. In: *African Vision and Eye Health* 71.3 (2012), pp. 133–143.

[193] S. N. Gowda and C. Yuan. “ColorNet: Investigating the importance of color spaces for image classification”. In: *Asian Conference on Computer Vision*. Springer. 2018, pp. 581–596.

[194] T. Gevers, A. Gijsenij, J. Van de Weijer, and J.-M. Geusebroek. *Color in computer vision: fundamentals and applications*. John Wiley & Sons, 2012. ISBN: 978-0-470-89084-4.

[195] E. H. Adelson and J. Y. Wang. “Single lens stereo with a plenoptic camera”. In: *IEEE transactions on pattern analysis and machine intelligence* 14.2 (1992), pp. 99–106.

[196] J. Weng, P. Cohen, M. Herniou, et al. “Camera calibration with distortion models and accuracy evaluation”. In: *IEEE Transactions on pattern analysis and machine intelligence* 14.10 (1992), pp. 965–980.

[197] Z. Zhang. “A flexible new technique for camera calibration”. In: *IEEE Transactions on pattern analysis and machine intelligence* 22.11 (2000), pp. 1330–1334.

[198] I.-C. Khoo. *Liquid crystals*. John Wiley & Sons, 2022.

[199] S. Kuiper and B. Hendriks. “Variable-focus liquid lens for miniature cameras”. In: *Applied physics letters* 85.7 (2004), pp. 1128–1130.

[200] C.-P. Chiu et al. “Liquid lenses and driving mechanisms: a review”. In: *Journal of Adhesion Science and Technology* 26.12-17 (2012), pp. 1773–1788.

[201] L. Shih. “Autofocus survey: a comparison of algorithms”. In: *Digital photography III*. Vol. 6502. SPIE. 2007, pp. 90–100.

[202] L. Glavin, J. MacKinnon, and S. Varghese. “Autofocus changes the paradigm for camera technology”. In: *Physics and Simulation of Optoelectronic Devices XXV*. Vol. 10098. SPIE. 2017, pp. 305–310.

[203] Y. Zhang et al. “Autofocus system and evaluation methodologies: a literature review”. In: *Sens. Mater* 30.5 (2018), pp. 1165–1174.

[204] D. Sugimura, T. Mikami, H. Yamashita, and T. Hamamoto. “Enhancing Color Images of Extremely Low Light Scenes Based on RGB/NIR Images Acquisition With Different Exposure Times”. In: *IEEE Transactions on Image Processing* 24.11 (2015), pp. 3586–3597. DOI: 10.1109/TIP.2015.2448356.

[205] F. Echtler, T. Sielhorst, M. Huber, and G. Klinker. “A short guide to modulated light”. In: *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*. 2009, pp. 393–396.

[206] I. Shim, J.-Y. Lee, and I. S. Kweon. “Auto-adjusting camera exposure for outdoor robotics using gradient information”. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2014, pp. 1011–1017. DOI: 10.1109/IROS.2014.6942682.

[207] N. Nourani-Vatani and J. Roberts. “Automatic camera exposure control”. In: *Proceedings of the Australasian Conference on Robotics and Automation 2007*. Australian Robotics and Automation Association (ARAA). 2007, pp. 1–6.

[208] A. Hornberg. *Handbook of machine and computer vision: The guide for developers and users*. 2nd ed. John Wiley and Sons, Incorporated., 2017, pp. 70–71. ISBN: 9783527413393.

[209] C. Steger, M. Ulrich, and C. Wiedemann. *Machine vision algorithms and applications*. John Wiley & Sons, 2018.

[210] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis and machine vision*. Springer, 2013.

[211] E. R. Davies. *Computer and machine vision: theory, algorithms, practicalities*. Academic Press, 2012.

[212] R. Jain, R. Kasturi, B. G. Schunck, et al. *Machine vision*. Vol. 5. McGraw-hill New York, 1995.

[213] J. Gu et al. “Recent advances in convolutional neural networks”. In: *Pattern Recognition* 77 (2018), pp. 354–377. ISSN: 0031-3203. DOI: `https://doi.org/10.1016/j.patcog.2017.10.013`.

[214] K. O’Shea. “An introduction to convolutional neural networks”. In: *arXiv preprint arXiv:1511.08458* (2015).

[215] *CNN Overview*. Available online: `https://www.analyticsvidhya.com/blog/2020/10/what-is-the-convolutional-neural-network-architecture/`. Accessed: 2024-27-08.

[216] O. Russakovsky et al. “Imagenet large scale visual recognition challenge”. In: *International journal of computer vision* 115 (2015), pp. 211–252.

[217] H. Yang and I. Patras. “Mirror, mirror on the wall, tell me, is the error small?” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 4685–4693.

[218] S. Xie and Z. Tu. “Holistically-nested edge detection”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1395–1403.

[219] J. Salamon and J. P. Bello. “Deep convolutional neural networks and data augmentation for environmental sound classification”. In: *IEEE Signal processing letters* 24.3 (2017), pp. 279–283.

[220] D. Eigen and R. Fergus. “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2650–2658.

[221] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. “The loss surfaces of multilayer networks”. In: *Artificial intelligence and statistics*. PMLR. 2015, pp. 192–204.

[222] D. Mishkin and J. Matas. “All you need is a good init”. In: *arXiv preprint arXiv:1511.06422* (2015).

[223] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. “On the importance of initialization and momentum in deep learning”. In: *International conference on machine learning*. PMLR. 2013, pp. 1139–1147.

[224] T. Nguyen and F. Pernkopf. “Chapter 9 - Computational lung sound classification: a review”. In: *State of the Art in Neural Networks and Their Applications*. Ed. by

A. S. El-Baz and J. S. Suri. Academic Press, 2023, pp. 193–215. ISBN: 978-0-12-819872-8. DOI: https://doi.org/10.1016/B978-0-12-819872-8.00016-1. URL: https://www.sciencedirect.com/science/article/pii/B9780128198728000161.

[225] N. Srivastava and R. R. Salakhutdinov. "Discriminative transfer learning with tree-based priors". In: *Advances in neural information processing systems* 26 (2013).

[226] N. Qian. "On the momentum term in gradient descent learning algorithms". In: *Neural networks* 12.1 (1999), pp. 145–151.

[227] D. P. Kingma. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[228] I. Loshchilov and F. Hutter. "Sgdr: Stochastic gradient descent with warm restarts". In: *arXiv preprint arXiv:1608.03983* (2016).

[229] T. Schaul, S. Zhang, and Y. LeCun. "No more pesky learning rates". In: *International conference on machine learning*. PMLR. 2013, pp. 343–351.

[230] Y. Yao, L. Rosasco, and A. Caponnetto. "On early stopping in gradient descent learning". In: *Constructive Approximation* 26.2 (2007), pp. 289–315.

[231] L. Prechelt. "Early stopping-but when?" In: *Neural Networks: Tricks of the trade*. Springer, 2002, pp. 55–69.

[232] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. "Understanding deep learning (still) requires rethinking generalization". In: *Communications of the ACM* 64.3 (2021), pp. 107–115.

[233] S. Ioffe. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167* (2015).

[234] K. He, X. Zhang, S. Ren, and J. Sun. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[235] X. Glorot and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[236] A. M. Saxe, J. L. McClelland, and S. Ganguli. "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks". In: *arXiv preprint arXiv:1312.6120* (2013).

[237] S. Hochreiter. “Long Short-term Memory”. In: *Neural Computation MIT-Press* (1997).

[238] K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[239] T. Schlosser, M. Friedrich, T. Meyer, and D. Kowerko. “A Consolidated Overview of Evaluation and Performance Metrics for Machine Learning and Computer Vision”. In: *Tobias Schlosser, Michael Friedrich, Trixy Meyer, and Danny Kowerko–Junior Professorship of Media Computing, Chemnitz University of Technology* 9107 (2023).

[240] A. E. Maxwell, T. A. Warner, and L. A. Guillén. “Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—Part 1: Literature review”. In: *Remote Sensing* 13.13 (2021), p. 2450.

[241] A. E. Maxwell, T. A. Warner, and L. A. Guillén. “Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—Part 2: Recommendations and best practices”. In: *Remote Sensing* 13.13 (2021), p. 2591.

[242] D. L. Pieper. *The kinematics of manipulators under computer control*. Stanford University, 1969.

[243] K. Elashry and R. Glynn. “An Approach to Automated Construction Using Adaptive Programing”. In: Mar. 2014, pp. 51–66. ISBN: 978-3-319-04662-4. DOI: `10.1007/978-3-319-04663-1_4`.

[244] G. Boschetti and T. Sinico. “Performance Comparison of Two Architectures of 6R Articulated Robots”. In: *Machines* 11.2 (2023), p. 306.

[245] L. Sciavicco and B. Siciliano. *Modelling and control of robot manipulators*. Springer Science & Business Media, 2001.

[246] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Dynamics and Control*. John Wiley & Sons, 2004.

[247] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir. “Inverse Kinematics Techniques in Computer Graphics: A Survey”. In: *Computer Graphics Forum* 37.6 (2018), pp. 35–58. DOI: `https://doi.org/10.1111/cgf.13310`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13310`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13310`.

[248] R. Singh, V. Kukshal, and V. S. Yadav. “A review on forward and inverse kinematics of classical serial manipulators”. In: *Advances in Engineering Design: Select Proceedings of ICOIED 2020* (2021), pp. 417–428.

[249] J. T. Gravdahl. “Force estimation in robotic manipulators: Modeling, simulation and experiments”. In: *MS thesis* (2014).

[250] O. Egeland and J. T. Gravdahl. *Modeling and simulation for automatic control*. Vol. 76. Marine Cybernetics Trondheim, Norway, 2002.

[251] D. Hamid. “Robust torque control of harmonic drivesystems”. In: *McGill University, Canada* (1997).

[252] T. Tuttle and W. Seering. “A nonlinear model of a harmonic drive gear transmission”. In: *IEEE Transactions on Robotics and Automation* 12.3 (1996), pp. 368–374. DOI: 10.1109/70.499819.

[253] C. I. Rizescu, C. Udrea, and D. Rizescu. “Experimental Setup for Harmonic Drive Efficiency Determination”. In: *Advanced Materials Research* 463 (2012), pp. 1518–1521.

[254] J. W. Sensinger and J. H. Lipsey. “Cycloid vs. harmonic drives for use in high ratio, single stage robotic transmissions”. In: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 4130–4135. DOI: 10.1109/ICRA.2012.6224739.

[255] T. D. Tuttle. “Understanding and modeling the behavior of a harmonic drive gear transmission”. PhD thesis. Massachusetts Institute of Technology, 1992.

[256] H. Zhang, S. Ahmad, and G. Liu. “Torque Estimation for Robotic Joint With Harmonic Drive Transmission Based on Position Measurements”. In: *IEEE Transactions on Robotics* 31.2 (2015), pp. 322–330. DOI: 10.1109/TRO.2015.2402511.

[257] Z. Shi, Y. Li, and G. Liu. “Adaptive torque estimation of robot joint with harmonic drive transmission”. In: *Mechanical Systems and Signal Processing* 96 (2017), pp. 1–15. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2017.03.041. URL: https://www.sciencedirect.com/science/article/pii/S0888327017301772.

[258] M. Ferre, P. E. Peralta, V. R. Gallego, N. Barbosa, and M. Á. Sánchez-Urán. “Forces Analysis on Robotics Screwing Tasks”. In: *Robot 2023: Sixth Iberian Robotics Conference*. Ed. by L. Marques, C. Santos, J. L. Lima, D. Tardioli, and M. Ferre. Cham: Springer Nature Switzerland, 2024, pp. 553–563. ISBN: 978-3-031-58676-7.

[259] M. Di Castro, M. Ferre, and A. Masi. “CERNTAURO: A Modular Architecture for Robotic Inspection and Telemanipulation in Harsh and Semi-Structured Environments”. In: *IEEE Access* 6 (2018), pp. 37506–37522. DOI: 10.1109/ACCESS.2018.2849572.

[260] *Omron Industrial Smart Camera*. Available online: https://industrial.omron.es/es/products/F430-F000L12M-SWV. Accessed: 2023-08-07.

[261] *P-SERIES Industrial Smart Camera*. Available online: `https://apps.datalogic.com/e-catalog/ec-P-Series.aspx`. Accessed: 2023-08-07.

[262] *Cognex in-sight-2000 Industrial Smart Camera*. Available online: `https://www.cognex.com/products/machine-vision/vision-sensors/in-sight-2000-mini`. Accessed: 2023-08-07.

[263] *Cognex in-sight-7000 Industrial Smart Camera*. Available online: `https://www.cognex.com/products/machine-vision/2d-machine-vision-systems/in-sight-7000-series`. Accessed: 2023-08-07.

[264] B. Pan, X. Zhang, Y. Lv, and L. Yu. "Automatic optimal camera exposure time control for digital image correlation". In: *Measurement Science and Technology* (2022).

[265] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), pp. 834–848.

[266] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. "MobileNetV2: Inverted Residuals and Linear Bottlenecks". In: *CVPR*. 2018.

[267] A. Howard et al. "Searching for MobileNetV3". In: *ICCV*. 2019.

[268] S.-T. Wu and M. R. G. Marquez. "A non-self-intersection Douglas-Peucker algorithm". In: *16th Brazilian symposium on computer graphics and Image Processing (SIBGRAPI 2003)*. IEEE. 2003, pp. 60–66.

[269] R. G. Knudsen, J. W. Price, and G. E. Olson. *Wrench sockets, socket drives and similar couplers*. US Patent 3,495,485, Feb. 17, 1970.

[270] M. Dossier. *Tightening tool for nuts or bolts*. US Patent 4,581,957, Apr. 15, 1986.

[271] W. G. Mader and P. Peppel. *Wrench openings*. US Patent 5,092,203, Mar. 3, 1992.

[272] R. B. Wright and T. M. Vozenilek. *Socket wrench opening*. US Patent 5,284,073 994, Feb. 8, 1994.

[273] *Assembly tools for screws and nuts — Wrench and socket openings — Tolerances for general use*. ISO 691:2005(E), 2005.

[274] *Assembly tools for screws and nuts - Driving squares*. IS0 1174-l :1996(E), 1996.

[275] *Assembly tools for screws and nuts - Square drive sockets*. IS0 2725-1:1996(E), 1996.

[276] *Hexagon head bolts - Product grades A and B*. IS0 4014-1979(E), 1979.

[277] C. Harris, M. Stephens, et al. "A combined corner and edge detector". In: *Alvey vision conference*. Vol. 15. 50. Citeseer. 1988, pp. 10–5244.

[278] A. Buslaev et al. "Albumentations: fast and flexible image augmentations". In: *Information* 11.2 (2020), p. 125.

[279] *Universal Robot CoG payload desig*. Available online: `https://www.universal-robots.com/download/manuals-e-seriesur20ur30/application-guides/how-to-design-for-higher-payloads/`. Accessed: 2024-04-02.

[280] *socket-sets*. Available online: https://www.redboxtools.com/news/socket-sets-101/.

[281] D. Mironov et al. "Haptics of screwing and unscrewing for its application in smart factories for disassembly". In: *Haptics: Science, Technology, and Applications: 11th International Conference, EuroHaptics 2018, Pisa, Italy, June 13-16, 2018, Proceedings, Part II 11*. Springer. 2018, pp. 428–439.

[282] *Denavit Hartenberg*. Available online: `https://docs.mrpt.org/reference/latest/`. Accessed: 2024-04-08.

[283] P. M. Kebria, S. Al-wais, H. Abdi, and S. Nahavandi. "Kinematic and dynamic modelling of UR5 manipulator". In: *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2016, pp. 004229–004234. DOI: `10.1109/SMC.2016.7844896`.

[284] *UR5 Inverse Kinematics*. Available online: `https:https://tianyusongcom.files.wordpress.com/2017/12/ur5_inverse_kinematics.pdf`. Accessed: 2024-01-25.

[285] P. Corke and J. Haviland. "Not your grandmother's toolbox–the Robotics Toolbox reinvented for Python". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 11357–11363.

[286] *Pilz PRBT6 datasheet*. Available online:`https://www.pilz.com/download/open/PRBT_6_Operat_Manual_1004685-EN-04.pdf`. Accessed: 2024-07-13.

[287] M. Di Castro. "A novel robotic framework for safe inspection and telemanipulation in hazardous and unstructured environments". Unpublished. 2019. DOI: `10.20868/UPM.thesis.57127`. URL: `https://oa.upm.es/57127/`.

[288] M. Di Castro et al. "A dual arms robotic platform control for navigation, inspection and telemanipulation". In: *Proceedings of the 16th International Conference on Accelerator and Large Experimental Control Systems (ICALEPCS'17), Barcelona, Spain*. 2017, pp. 8–13.

[289] *Pilz PRBT6 Operating Manual*. Available online:https://www.pilz.com/download/open/PRBT_6_Operat_Manual_1004685-EN-04.pdf. Accessed: 2024-07-13.

[290] K. A. Szczurek, R. M. Prades, E. Matheson, J. Rodriguez-Nogueira, and M. D. Castro. "Multimodal Multi-User Mixed Reality Human–Robot Interface for Remote Operations in Hazardous Environments". In: *IEEE Access* 11 (2023), pp. 17305–17333. DOI: 10.1109/ACCESS.2023.3245833.

[291] J. Rodriguez-Nogueira, K. M. Chrostowski, M. Ferre, M. Di Castro, and E. Matheson. "Multi-user XR reality interface for telerobotic operators at CERN". In: *International Conference on Intelligent Robots and Systems (IROS), series = XR-ROB: Horizons of an extended robotics reality*. 2023.

[292] *ZerroErr servomotors*. Available online: https://docs.mrpt.org/reference/latest/. Accessed: 2024-04-08.

[293] *One Series Gable IMU*. Available online:https://gable-imu.nl/ONE-SERIES_Datasheet.pdf. Accessed: 2024-07-13.

[294] *UR5e datasheet*. Available online:https://s3-eu-west-1.amazonaws.com/ur-support-site/40974/UR5e_User_Manual_en_US.pdf. Accessed: 2024-07-13.

[295] *RoboDK software*. Available online: https://robodk.com/. Accessed: 2024-04-08.

[296] C. Kuhnel. *AVR RISC microcontroller handbook*. Newnes, 1998.

[297] P. Corke. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB®*. 2nd. Cham: Springer, 2017. ISBN: 978-3-319-54413-7. DOI: 10.1007/978-3-319-54413-7.

[298] P. Espinosa Peralta et al. "Performance Analysis of Localization Algorithms for Inspections in 2D and 3D Unstructured Environments Using 3D Laser Sensors and UAVs". In: *Sensors* 22.14 (2022). ISSN: 1424-8220. DOI: 10.3390/s22145122. URL: https://www.mdpi.com/1424-8220/22/14/5122.

[299] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. "A benchmark for the evaluation of RGB-D SLAM systems". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 573–580.

[300] *"Absolute Trajectory Error*. http://www.rawseeds.org/rs/methods/view//9. Accessed: 2021-04-11.

[301] *CIEMAT electron accelerator*. Available online: http://www.fusion.ciemat.es/competitive-access-to-facilities/electron-accelerator/. Accessed: 2023-08-30.

[302] S. Coloma. “Methods, strategies and application cases for robotic telemanipulation in hazardous environments”. PhD thesis. Universidad Politécnica de Madrid, 2020.

[303] F. G. Leite et al. “Ionizing radiation effects on a COTS low-cost RISC microcontroller”. In: *2017 18th IEEE Latin American Test Symposium (LATS)*. IEEE. 2017, pp. 1–4.

[304] S. Kucuk and Z. Bingul. *Robot kinematics: Forward and inverse kinematics.* INTECH Open Access Publisher London, UK, 2006.

Annexes

Annex

A Annex 1. MASK RCNN train parameters

```
"""Configuration for training on the toy shapes dataset.
Derives from the base Config class and overrides values specific
to the toy shapes dataset.
"""
# Give the configuration a recognizable name
NAME = "tornillo"

# Train on 1 GPU and 8 images per GPU. We can put multiple images on each
# GPU because the images are small. Batch size is 8 (GPUs * images/GPU).
GPU_COUNT = 1
IMAGES_PER_GPU = 1

# use small validation steps since the epoch is small
st1=len(os.listdir(os.path.join(ROOT_DIR,"data/perno/val")))//IMAGES_PER_GPU
VALIDATION_STEPS = st1

    # Use a small epoch since the data is simple
st2=len(os.listdir(os.path.join(ROOT_DIR,"data/perno/train")))//IMAGES_PER_GPU
STEPS_PER_EPOCH = st2

# Number of classes (including background)
NUM_CLASSES = 1 + 1  # background + 3 shapes

# Use small images for faster training. Set the limits of the small side
# the large side, and that determines the image shape.
#IMAGE_MIN_DIM = 480
#IMAGE_MAX_DIM = 640
IMAGE_MIN_DIM = 960
IMAGE_MAX_DIM = 1280
#IMAGE_RESIZE_MODE = "pad64"

# If enabled, resizes instance masks to a smaller size to reduce
# memory load. Recommended when using high-resolution images.
USE_MINI_MASK = True
MINI_MASK_SHAPE = (448, 448)  # (height, width) of the mini-mask

    # Number of color channels per image. RGB = 3, grayscale = 1, RGB-D = 4
# Changing this requires other changes in the code. See the WIKI for more
# details: https://github.com/matterport/Mask_RCNN/wiki
#IMAGE_CHANNEL_COUNT = 1
#MEAN_PIXEL = np.array([126])
```

```
BACKBONE = "resnet50"

# Maximum number of ground truth instances to use in one image
#MAX_GT_INSTANCES = 200
MAX_GT_INSTANCES = 50

    # Max number of final detections per image
DETECTION_MAX_INSTANCES = 400
#DETECTION_MAX_INSTANCES = 1000

# Optimizer, default is 'SGD'
OPTIMIZER = 'ADAM'

    # ROIs kept after non-maximum supression (training and inference)
#POST_NMS_ROIS_TRAINING = 2048
#POST_NMS_ROIS_INFERENCE = 2048
# Number of ROIs per image to feed to classifier/mask heads
TRAIN_ROIS_PER_IMAGE = 100

    # You can increase this during training to generate more proposals
                                          .
RPN_NMS_THRESHOLD = 0.99


# Minimum probability value to accept a detected instance
# ROIs below this threshold are skipped
DETECTION_MIN_CONFIDENCE = 0.95

# Non-maximum suppression threshold for detection
#DETECTION_NMS_THRESHOLD = 0.3  # 0.3

# Threshold number for mask binarization, only used in inference mode
#DETECTION_MASK_THRESHOLD = 0.35

# Whether to use image augmentation in training mode
#AUGMENT = True
```

B Annex 2. Mobilenet-v2 train parameters

```
#_________________________________________________________
#### TRAIN CODE ###
###################
python deeplab/train.py \
    --logtostderr \
    --training_number_of_steps=${NUM_ITERATIONS} \
    --train_split="train" \
    --model_variant="mobilenet_v2" \
    --output_stride=16 \
    --train_crop_size="513,513" \
    --train_batch_size=12 \
    --dataset="pascal_voc_seg" \
    --tf_initial_checkpoint=${PATH_TO_INITIAL_CHECKPOINT} \
    --train_logdir=${PATH_TO_TRAIN_DIR} \
    --dataset_dir=${PATH_TO_DATASET} \
    --initialize_last_layer=false \
    --fine_tune_batch_norm=true \
    --last_layers_contain_logits_only=true

#To train all layers the next flags has to be changing:
#--initialize_last_layer=true
#--fine_tune_batch_norm=false
#--learning_rate=0.045
#To train with Quantization aware the flag must be set:
#--quantize_delay_step=0"
#_________________________________________________________

#### EXPORT MODEL CODE ###
###################
python deeplab/export_model.py \
  --checkpoint_path=${PATH_TO_CHECKPOINT} \
  --export_path=${PATH_EXPORT/frozen.pb} \
  --model_variant="mobilenet_v2" \
  --output_stride=16 \
  --train_crop_size="513,513" \
  --num_classes=2 \
  --inference_scales=1.0 \
 #To train with Quantization aware the flag must be set:
#--quantize_delay_step=0"
#_________________________________________________________

#### CONVERT MODEL TO TFLITE ###
###################
python deeplab/convert_to_tflite.py \
  --quantized_graph_def_path=${PATH_TO_QUANTIZATED_MODEL}/frozen.pb \
  --input_tensor_name=MobilenetV2/MobilenetV2/input:0 \
  --output_tflite_path=${PATH_TO_QUANTIZATED_MODEL}/frozen_inference_graph
                                        .tflite \
  --test_image_path=${PATH_TO_TEST_IMAGES} #images to obtain a median
                                        pixels value
```

C Annex 3. EdgeMobilenet train parameters

```
python deeplab/train.py \
    --logtostderr \
    --training_number_of_steps=${NUM_ITERATIONS} \
    --train_split="train" \
    --model_variant="mobilenet_edgetpu" \
    --output_stride=16 \
    --decoder_output_stride=’’ \
    --aspp_convs_filters=256 \
    --train_crop_size="513,513" \
    --train_batch_size=10 \
    --dataset="pascal_voc_seg" \
    --tf_initial_checkpoint=${PATH_TO_INITIAL_CHECKPOINT} \
    --train_logdir=${PATH_TO_TRAIN_DIR} \
    --dataset_dir=${PATH_TO_DATASET} \
    --initialize_last_layer=false \
    --fine_tune_batch_norm=true \
    --last_layers_contain_logits_only=true

#To train all layers the next flags has to be changing:
#--initialize_last_layer=true
#--fine_tune_batch_norm=false
#--learning_rate=0.045
#To train with Quantization aware the flag must be set:
#--quantize_delay_step=0"

#_________________________________________________________

#### EXPORT MODEL CODE ####
###################
python deeplab/export_model.py \
  --checkpoint_path=${PATH_TO_QUANTIZATED_MODEL} \
  --export_path=${PATH_EXPORT} \
  --model_variant="mobilenet_edgetpu" \
  --output_stride=16 \
  --train_crop_size="513,513" \
  --num_classes=2 \
  --inference_scales=1.0 \

 #To train with Quantization aware the flag must be set:
#--quantize_delay_step=0"
#_________________________________________________________

#### CONVERT MODEL TO TFLITE ###
###################
python deeplab/convert_to_tflite.py \
  --quantized_graph_def_path=${PATH_TO_QUANTIZATED_MODEL}/frozen.pb \
  --input_tensor_name=MobilenetEdgeTPU/MobilenetEdgeTPU/input:0 \
  --output_tensor_name=ArgMax:0 \
  --output_tflite_path=${PATH_TO_QUANTIZATED_MODEL}/frozen.tflite \
  --test_image_path=${PATH_TO_TEST_IMAGE}
```

D Annex 4. Parameters of vision algorithms

```
#### Contours aproximation ###
points_wanted = 6
precision = 10000
list_aprox=[]
for x in range(precision):
    epsilon = (x/precision)*cv2.arcLength(cnt[0], True)
    approx = cv2.approxPolyDP(cnt[0], epsilon, True)
    if approx.shape[0] == points_wanted:
        list_aprox.append(approx)
        break

#### Find Contours ###
cv2.findContours(img,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)

#### Find Contours ###
cv2.convexHull(contours,False)

#### Contour to lines ###
cv2.HoughLinesP(binary,1,np.pi/360,50,minLineLength,maxLineGap)

#### Clustering ###
#Kmeans
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10, 0.1)
labels, centers = cv2.kmeans(pts, 3, None, criteria, 10, cv2.
                                     KMEANS_RANDOM_CENTERS)[1:]

#OPTICS
OPTICS(min_samples=2, xi=.5, min_cluster_size=0.25)

#DBSCAN
DBSCAN(eps=0.5, min_samples=1)

#### Fuse lines ###
fitLine(Points, cv2.DIST_L2,0,0.01,0.01)
```

E Annex 5. Simulation forces and torques 1

F Annex 6. Simulation forces and torques 2

Table E.2: Simulation forces and torques obtained related reference of the coordinate system shown in Fig. 3.20. The sizes of the tool represented by the letters a-i are mentioned in the Table. 3.6 and some designs are shown in Fig. 3.22

	Moment load 40 [Nm] / Socket length 0.02 [m]											
	Tool Flange						**Tool Stress**					
	Force [N]			**Torque [Nm]**			**Force [N]**			**Torque [Nm]**		
	Fx	**Fy**	**Fz**	**Tx**	**Ty**	**Tz**	**Fx**	**Fy**	**Fz**	**Tx**	**Ty**	**Tz**
a	0,00	-177,28	0,00	-2,68	0,00	-13,22	-0,68	9,03	0,16	-0,10	-0,05	-34,39
b	0,00	-126,80	0,00	-1,02	0,00	10,22	-0,32	18,40	0,11	-0,07	-0,09	-27,04
c	0,00	-202,68	0,00	-10,31	0,00	-1,48	-0,53	-17,01	-0,21	0,44	0,01	33,77
d	0,00	-89,60	0,00	-4,34	0,00	2,10	-0,56	15,92	-0,20	-0,45	0,01	-29,44
e	0,00	-83,36	0,00	-10,64	0,00	-0,04	-0,17	16,97	0,35	-0,91	-0,11	-27,95
f	-0,45	-317,28	0,00	-17,66	0,02	-5,98	-0,40	15,29	-0,18	-0,69	0,009	-35,39
g	0,00	-300,08	0,00	-39,92	0,00	0,00	-2,20	12,70	-1,01	1,39	-0,31	34,20
h	0,00	0,88	-0,84	0,04	0,11	-28,12	0,84	-1,28	0,03	0,11	-0,10	-35,71
i	0,64	-266,40	0,00	-12,92	-0,02	-13,72	-0,54	16,28	0,15	-0,45	0,00	-35,43

	Moment load 40 [Nm] / Socket length 0.043 [m]											
	Tool Flange						**Tool Stress**					
	Force [N]			**Torque [Nm]**			**Force [N]**			**Torque [Nm]**		
	Fx	**Fy**	**Fz**	**Tx**	**Ty**	**Tz**	**Fx**	**Fy**	**Fz**	**Tx**	**Ty**	**Tz**
a	0,00	-106,36	0,00	-2,41	0,00	-23,25	-0,30	8,48	0,36	-0,09	-0,06	-34,89
b	-0,17	-116,16	0,00	-1,21	0,00	5,84	-0,13	17,77	0,23	-0,08	-0,10	-27,56
c	0,00	-188,52	0,00	-10,30	0,00	-4,26	0,57	-16,71	-0,13	0,45	0,02	34,28
d	0,00	-89,28	0,00	-4,68	0,00	1,61	0,28	15,52	-0,13	-0,44	-0,01	-29,74
e	0,00	-88,40	0,00	-11,44	0,00	-0,01	-0,18	16,12	0,08	-0,89	-0,07	-28,36
f	0,97	-253,92	0,00	-15,84	-0,03	-11,42	0,19	15,03	-0,14	-0,69	0,01	-35,89
g	0,00	-294,72	0,00	-41,68	0,00	-4,44	-3,29	-13,03	-0,84	1,40	-0,34	34,76
h	0,00	0,00	0,00	0,00	0,24	-32,12	0,36	-1,19	-0,11	0,09	-0,11	-36,18
i	1,20	-210,00	0,00	-11,88	0,00	-18,56	-0,21	15,62	0,55	-0,46	0,00	-35,96

	Moment load 40 [Nm] / Socket length 0.078 [m]											
	Tool Flange						**Tool Stress**					
	Force [N]			**Torque [Nm]**			**Force [N]**			**Torque [Nm]**		
	Fx	**Fy**	**Fz**	**Tx**	**Ty**	**Tz**	**Fx**	**Fy**	**Fz**	**Tx**	**Ty**	**Tz**
a	0,00	-52,68	0,00	-1,90	0,00	-30,92	0,37	15,25	0,34	-0,92	-0,06	-35,46
b	0,23	-91,20	0,00	-1,35	0,00	-3,69	-0,16	18,48	0,25	-0,07	-0,15	-27,60
c	0,00	-151,48	0,00	-9,26	-0,02	-10,16	0,06	-17,84	-0,59	0,05	0,06	34,39
d	2,18	-91,20	0,00	-4,88	-0,01	-0,14	-0,12	16,40	0,01	-0,46	-0,03	-29,68
e	0,00	-90,12	0,00	-11,84	-0,01	-0,44	0,15	16,74	0,36	-0,91	-0,12	-28,33
f	0,00	-162,32	0,00	-11,64	-0,02	-19,32	0,37	15,80	0,24	-0,72	-0,01	-36,08
g	0,00	-258,72	0,00	-39,93	0,00	-6,74	-2,89	-13,47	0,59	1,44	-0,36	35,00
h	0,00	0,00	0,00	0,00	0,00	-34,64	0,26	-1,46	0,23	0,12	-0,11	-36,38
i	0,80	-135,64	0,00	-9,52	0,00	-24,76	0,26	15,95	0,31	-0,46	0,00	-36,16

	Moment load 40 [Nm] / Socket length 0.678 [m]											
	Tool Flange						**Tool Stress**					
	Force [N]			**Torque [Nm]**			**Force [N]**			**Torque [Nm]**		
	Fx	**Fy**	**Fz**	**Tx**	**Ty**	**Tz**	**Fx**	**Fy**	**Fz**	**Tx**	**Ty**	**Tz**
a	-12,33	0,00	-1,22	-0,28	0,00	-38,00	0,43	9,33	-0,72	-0,09	0,03	-33,34
b	-12,33	-1,53	0,00	-0,37	0,03	-37,07	0,34	17,95	-0,11	-0,07	-0,06	-26,23
c	-12,36	0,00	0,00	0,05	0,43	-36,50	-0,11	-17,41	0,44	0,45	0,01	32,68
d	-12,59	-5,71	0,00	-0,99	0,50	-34,11	-2,19	22,39	1,05	-0,56	-0,28	-28,30
e	-12,69	-23,01	0,00	-6,57	0,12	-23,32	0,66	23,37	-0,86	-1,09	0,13	-26,70
f	0,00	1,73	0,00	1,32	0,00	-36,98	0,50	15,56	-0,38	-0,67	0,00	-34,31
g	-6,52	2,57	-3,87	-1,68	0,52	-31,08	-3,46	-14,51	0,76	1,458	-0,41	34,27
h	0,00	0,00	0,00	-0,04	0,00	-36,52	0,00	-1,27	-0,14	0,11	-0,08	-34,60
i	0,00	0,00	0,00	-0,56	0,00	-35,04	-0,02	15,61	-0,44	-0,44	-0,01	-34,40

Table F.3: Simulation forces and torques using tool designs ’a,’ ’g,’ and ‘h’ are shown in Fig. 3.22 at the points described in Fig. 3.25. The values represent the joint torques calculated by the best pose provided by Algorithm 8.. The external forces/torques [Fx, Fy, Fz, Tx, Ty, Tz] were taken from the Tab. E.2 column labeled tool stress, where tool ‘a’ = [0.37, 15.25, 0.34, -0.92, -0.06, -35.46], tool ‘g’ = [-3.29, -13.03, -0.84, 1.4, -0.34, 34.76], and tool ‘h’ = [0.26, 15.95, 0.31, -0.46, 0, -36.16]. Each external force was rotated according to the point of application.

		UR5e	**Pilz PRBT6**	**UR5e**	**Pilz PRBT6**
Points	**Tool**	**Joints Forces[N]/Torques[Nm] [Fx, Fy, Fz, Tx, Ty, Tz]**	**Joints Forces[N]/Torques[Nm] [Fx, Fy, Fz, Tx, Ty, Tz]**	**Joints Positions** [°]	**Joints Positions** [°]
P1	**A**	[39.39, -0.28, 0.68, 0.52, 2.46, -35.46]	[31.54, 0.14, 1.51, -25.7, 0.6, -35.46]	[-167.3, -55.0, -139.23, -75.63, 89.95, 13.18]	[-16.62, 7.45, -125.48, -0.0, -46.98, -16.62]
	G	[-38.34, 0.31, 0.2, -0.11, 36.07, -1.17]	[-31.16, -0.5, 1.41, -32.61, -3.67, 1.64]	[-162.15, -57.3, -126.05, -177.04, -72.19, -179.91]	[-20.05, 116.31, 132.93, 20.63, -106.0, 5.73]
	H	[42.22, -0.58, 0.26, 0.25, 1.75, -36.16]	[30.12, 0.55, 0.63, -13.35, 0.38, -36.16]	[-171.31, -73.34, -115.16, -81.93, 89.95, -81.36]	[-11.46, 16.04, -92.82, 0.0, -71.05, -11.46]
P2	**A**	[42.8, -0.21, 0.41, 0.35, -2.09, -35.47]	[28.13, 0.23, 0.62, -17.98, 0.32, -35.46]	[-173.03, -118.03, -80.79, 108.29, -89.95, -173.03]	[-9.17, 37.82, -79.07, -0.0, -63.03, -9.17]
	G	[-41.27, 0.23, 0.71, 0.08, -36.07, -1.11]	[-28.29, -0.64, 1.16, -30.49, 2.2, 1.15]	[-171.89, -106.57, -117.46, 43.54, 81.93, -0.0]	[-10.31, 114.02, 91.1, 10.89, -67.04, -4.58]
	H	[45.78, -0.44, -0.23, 0.19, -1.71, -36.16]	[26.57, 0.49, -0.01, 7.25, 0.25, -36.16]	[-174.75, -170.74, 36.1, 44.12, -89.95, 95.11]	[-7.45, 62.45, -13.18, -0.0, -104.28, -7.45]
P3	**A**	[-10.83, -3.01, -3.24, -3.74, 2.09, -35.47]	[7.03, -9.11, 9.48, 4.26, -0.02, -35.45]	[-174.18, -171.89, 64.17, 108.29, -84.22, 0.0]	[-13.75, 101.99, 105.42, -166.16, 92.82, 0.57]
	G	[5.43, 4.52, 4.45, 5.77, 35.54, -1.31]	[-5.57, 7.32, -6.6, 32.29, -7.74, 1.45]	[-169.6, -62.45, -114.59, -92.82, 89.95, -169.6]	[-13.75, 4.01, -98.55, 0.0, -77.35, 166.16]
	H	[-5.21, -9.74, -11.57, -10.27, -1.99, -36.16]	[5.28, -12.55, 13.56, 9.2, 0.02, -36.16]	[-162.72, -79.64, -151.83, 51.57, 72.77, 89.95]	[-19.48, 122.04, 134.65, -159.86, 101.99, 4.58]
P4	**A**	[-10.83, -3.01, -3.24, -3.74, 2.09, -35.47]	[10.9, -4.8, 4.89, -24.2, 0.33, -35.46]	[-174.18, -171.89, 64.17, 108.29, -84.22, 0.0]	[-7.45, 85.94, 38.96, -169.02, 43.54, -8.02]
	G	[8.76, 1.2, 1.63, 2.91, 35.92, -1.31]	[-8.86, 2.46, -3.5, 24.41, -4.1, 1.45]	[-174.18, -99.12, -84.22, -86.52, 89.95, -174.18]	[-7.45, 76.2, 26.93, -0.0, -130.63, 172.46]
	H	[-9.21, -3.9, -3.55, -4.2, -1.75, -36.16]	[-0.16, 2.01, 1.47, 0.19, -1.04, 3.04]	[-173.03, 141.52, 112.87, -73.91, 82.51, 89.95]	[-9.17, 115.16, 84.22, 10.89, -59.59, 174.18]
P5	**A**	[0.42, -2.15, -1.91, -1.81, -2.08, -35.47]	[-0.42, 21.48, -23.73, -10.53, 0.43, -35.47]	[177.04, -134.07, 68.18, 66.46, 87.09, -179.91]	[-49.85, 59.01, 80.79, -127.77, -75.63, -17.19]
	G	[0.23, 1.76, 2.33, 1.11, 36.05, -1.02]	[0.01, -20.94, 23.49, 10.99, 25.91, 1.62]	[177.04, -68.75, -95.68, 75.06, 89.95, -177.04]	[-49.85, 65.89, 92.82, 0.0, -26.36, -130.06]
	H	[-1.62, -1.31, -1.19, -1.1, -1.69, -36.16]	[1.66, -14.31, 16.88, 3.67, -0.19, -36.16]	[178.19, -110.58, 8.59, 102.56, 88.24, 89.95]	[147.25, -36.1, -25.78, 33.23, -98.55, 5.16]
P6	**A**	[0.38, -2.06, -1.84, -1.81, -2.44, -35.46]	[-0.46, 24.09, -24.1, -14.38, 0.79, -35.46]	[177.04, -159.28, 110.58, 48.7, 87.09, -179.91]	[-49.85, 91.67, 124.9, -124.9, -69.33, -26.36]
	G	[0.24, 1.8, 2.36, 1.11, 36.05, -1.02]	[-0.02, -23.25, 23.68, 12.43, 25.91, 1.4]	[177.04, -69.33, -127.77, 107.72, 89.95, -177.04]	[-49.85, 100.84, 132.35, -0.0, -31.51, -130.06]
	H	[-1.62, -1.31, -1.19, -1.1, -1.69, -36.16]	[1.66, 16.22, -16.93, -9.85, 0.12, -36.16]	[178.19, -110.58, 8.59, 102.56, 88.24, 89.95]	[-32.66, 78.5, 93.97, -146.1, -76.78, -8.59]
P7	**A**	[-37.55, 12.06, 6.56, 2.4, 0.01, -35.46]	[-33.34, -7.62, 3.88, -24.52, 1.61, -35.46]	[-89.95, -123.19, 77.92, -44.69, 89.95, 89.95]	[-46.41, -5.73, -54.43, 0.0, 48.13, -133.5]
	G	[37.4, -9.61, -4.29, -0.14, -36.06, -1.21]	[32.09, 6.42, -3.82, 32.38, -7.45, 1.37]	[-122.61, -105.42, 45.84, 60.16, -32.66, -0.0]	[-59.01, 43.54, 57.87, -120.32, 97.4, 12.61]
	H	[-40.19, 0.01, 0.06, -0.05, -1.66, -36.16]	[-32.07, -4.29, 1.69, -13.58, 0.84, -36.16]	[178.76, -87.09, -76.2, 73.91, 89.95, 91.67]	[-28.65, 0.0, -70.47, -0.0, 70.47, -151.26]
P8	**A**	[-37.49, -0.26, -0.31, -0.12, 2.11, -35.47]	[-33.36, -4.84, 1.29, 3.08, 1.61, -35.46]	[177.04, -154.13, 125.48, 119.18, -89.95, 2.86]	[-46.41, -18.33, -115.74, 0.0, 97.4, -133.5]
	G	[37.4, -1.17, -0.41, -0.09, -36.06, -1.4]	[32.11, 3.97, -3.24, 26.43, -18.34, 1.5]	[174.75, -148.97, 105.42, 43.54, -95.11, 0.0]	[-59.01, 82.51, 119.18, -115.74, 107.72, 32.66]
	H	[-40.22, 0.04, -0.1, -0.05, -1.77, -36.16]	[-32.08, -2.37, -0.22, 16.89, 0.84, -36.15]	[178.76, 164.44, 113.45, -7.45, 89.95, 91.67]	[-28.65, 4.01, -116.31, -0.0, 120.89, -151.26]