Lab 1 Assignment Report

ROBOTICS 41013

Ajal Singh | 12621189

Table of Contents

1. INTRODUCTION 3
2. INSTALL 3
   2.1. UR3 3
   2.2. LINEARUR5 4
   2.3. ENVIRONMENT 4
3. SAFETY & RISK ASSESSMENT 4
4. UR3 – ADVANTAGES AND LIMITATIONS 5
5. MATLAB & ROBOTICS TOOLBOX 5
6. ROBOT PRECISION 6
7. SENSING AND GRASPING CHALLENGES 7
8. ROBOT SAFETY 7
9. REFERENCES 9
10. APPENDICES 10
   10.1. SAFE WORK METHOD STATEMENT 10
   10.2. RISK ASSESSMENT FORM 12

1. Introduction

SafeCo has been tasked with the installation and simulation of two brick stacking collaborative robots (cobots) to build a 3x3 brick wall. We have chosen the UR3 and LinearUR5 to complete this task. Each cobot is to pick-up bricks from a conveyor and build a wall on the platform.

2. Install

2.1. UR3

The UR3 used in this simulation has a custom paint job as well as engraving/embossing. As seen in Figure 1, the end-effector is painted turquoise, the base joint and link are painted grey, the remaining links painted dark red, and the remaining joints painted gold. The base of the UR3 has been embossed with my name and engraved with my student number. The default joint configuration is [0, $-\pi/2$, 0, $-\pi/2$, 0, 0] for joints 1 to 6 respectively (Figure 1). The home position is then set to [$-\pi/2$, 0, $\pi/2$, 0, $\pi/2$, 0] for joints 1 to 6 respectively (Figure 2).

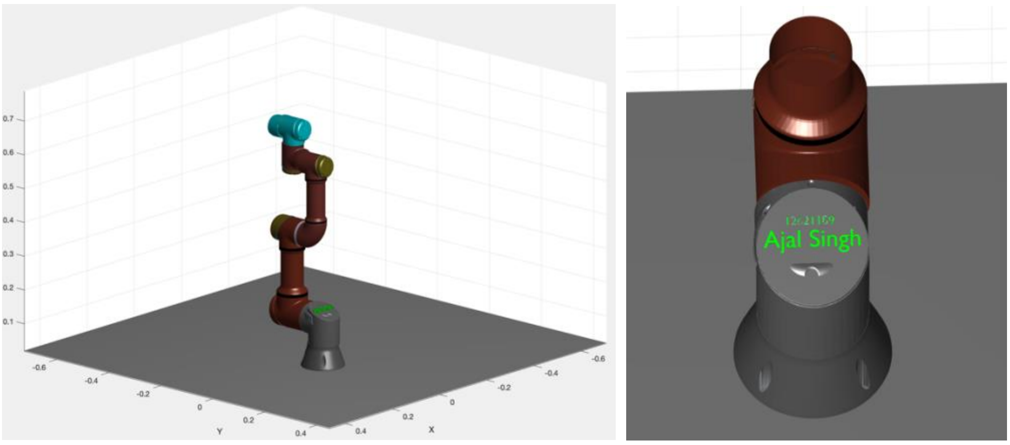


Figure 1: Customised Model of UR3

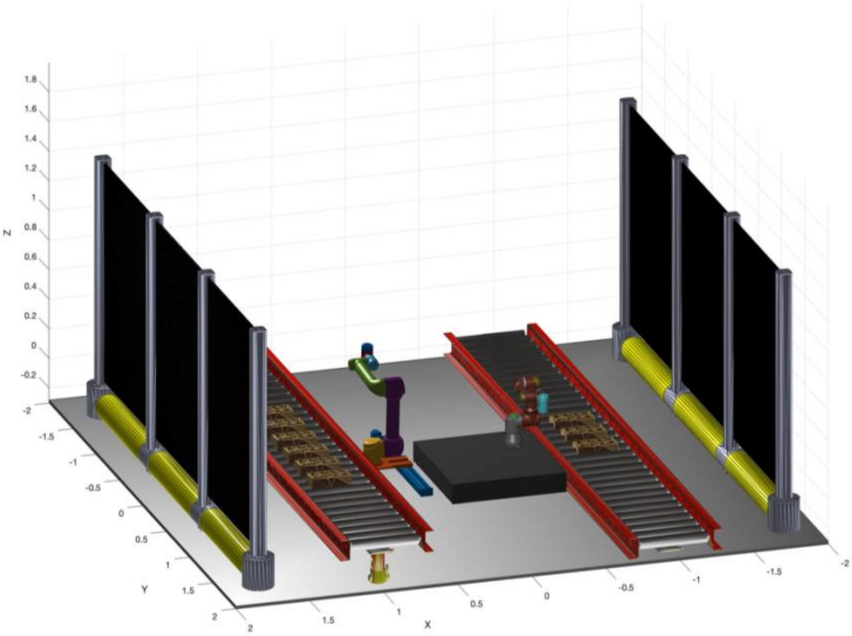


Figure 2: View of environment setup

2.2. LinearUR5

The Linear UR5 is used due to its reach advantage achieved by the linear rail. Aesthetic or default joint configuration modifications are not made. The home position is then set to [0, $-\pi/2$, 0, $\pi/2$, 0, $\pi/2$, 0] for joints 1 to 6 respectively (Figure 2) and is rotated about the x-axis by 90 degrees to stand upright.

2.3. Environment

The environment was created with a conveyor system application in mind. The placement of the LinearUR5 is offset by 450mm in the positive x-direction to service the taller conveyor which stands 370mm high and 500mm in the positive x-direction from the base of the LinearUR5.

The UR3 is mounted on a platform 150mm high and offset 500mm in the negative x-direction from the centre of the environment. The platform is offset 200mm in the negative x-direction and 200mm in the positive y-direction from the base of the UR3. The conveyor serviced by the UR3 stands at a lower height of 150mm and 500mm from the base of the UR3 in the negative x-direction.

Placed on the conveyors are golden bricks with dimensions 267x66x134 mm. They are placed 250mm apart (edge to edge) as depicted in Figure 2. They are then stacked (end-to-end) 300mm from the centre of the platform in the x-direction.

For safety reasons, fences have been erected 1300mm from either side of the conveyor as seen in Figure 2. This prevents access to the area while the robots are in operation.

3. Safety & Risk Assessment

The following risk assessment has been completed based on the simulation created. The fences act as a barrier to humans and it is unlikely anyone will enter the area and collide with the robot while in operation. It is also unlikely that the robot will collide with other fixed objects in the environment. On the other hand, both robots place bricks on the wall at the same time and often cross paths hence the likelihood of collision and tool damage is very high. There is a small chance either robot could break down or perform in an undesired manner. An average brick weighs approximately 3.1kg, while the UR3 has a maximum payload capability of 3kgs. It will most likely struggle to perform this task.

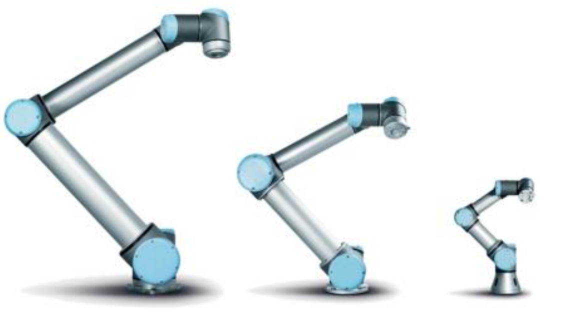
Risk	Likelihood (1-5)
Robot-Human Collision	1
Robot-Object/Hazard Collision	2
Robot-Robot collision	5
Tool/Payload Damage	5
Incorrect End-effector position	2
Incorrect end-effector orientation	2
Robot breakdown	1
Payload too Heavy/Big	3

See Appendix for detailed Risk Assessment form and Safe Work Method Statement.

4. UR3 – Advantages and Limitations

The UR3 is a cheaper and smaller version of the popular and familiar UR5 or UR10 cobots, making it an ideal collaborative robot for use in small spaces or tabletops. It also has 6 degrees of freedom with 360 degrees rotation at each joint, allowing for manoeuvres in tight spaces and has precision of 0.1mm. For safety reasons, the arm is capable of stopping when an overcurrent such as a collision is detected. Ideal implementations of the UR3 robot are soldering, gluing, and printed circuit board tasks.

Conversely, the compact nature of this robot limits its reach (500mm) and can only support a payload of 3kgs. Hence, the UR3 cannot be used for heavy-duty tasks and in the real world, may struggle to perform the brick stacking task as carried out on the MATLAB simulation. During simulation, limited reach of the UR3 prevented the collection of bricks from the conveyor in an acceptable manner. Henceforth, the UR3 was mounted on a 130mm platform as seen in Figure 4. When compared to its bigger brothers, the UR3 does not have many sensors such as the vision sensor with the focus on simplicity and low price.



	UR10 robot arm	UR5 robot arm	UR3 robot arm
Working radius	1300 mm/51.2 ins	850 mm/33.5 ins	500 mm/19.7 ins
Payload	10 kg/22 lbs	5 kg/11 lbs	3 kg/6.6 lbs
Weight	28.9 kg/63.7 lbs	18.4 kg/40.6 lbs	11 kg/24.3 lbs
Footprint	190 mm/7.5 ins	149 mm/5.9 ins	118 mm/4.6 ins

Figure 3: UR range of cobots

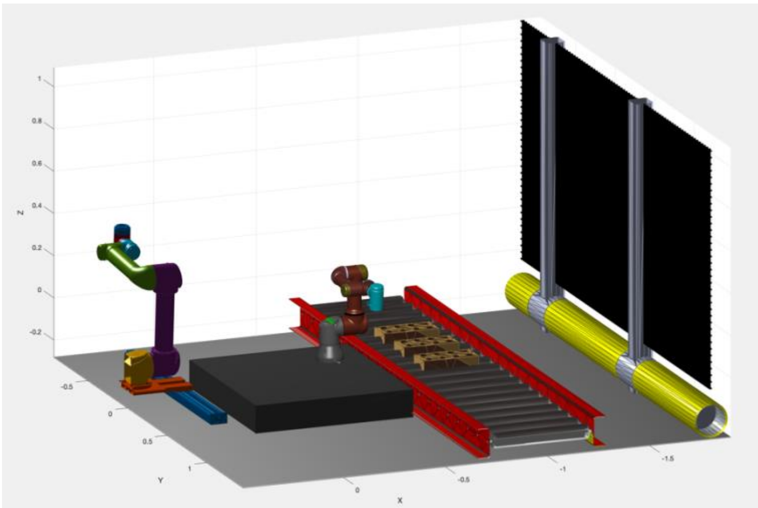


Figure 4: UR3 mounted on a platform

5. MATLAB & Robotics Toolbox

MATLAB is a high-performance numeric programming language allowing integration for visualisation, technical computations and programming in a simple interface. This makes it a quick and useful tool for rapid prototyping to try out different ideas and methodologies. With the inclusion of many different toolbox add-ons and object-oriented programming paradigms, MATLAB can be very useful for programming in highly mathematical applications such as robotics where matrix manipulation is heavily used. Both the MATLAB language and IDE are developed by Mathworks, resulting in seamless integration and performance. Source control such as Git is also built into the application making coding smooth and efficient. On the contrary, some built-in functions are rather ambiguous in terms of naming /syntax and requires testing or visiting documentation to properly understand its operation. It would also be nice to open another command window tab in a similar fashion to Linux Terminals.

An alternative to MATLAB in Robotics applications would be C++ or Python as both support ROS (Robot Operating System). Python programming can take advantage of numPy and matplotlib libraries. However, both languages are more complex when it comes to coding.

The robotics toolbox simplifies the simulation of robots for development and testing. Included functions such as fkine and ikine, forward and inverse kinematics, make calculations for manipulator kinematics significantly quicker and simpler by handling all maths internally. With built-in robots and the ability to develop our custom robots with DH parameters, we can simulate for the real-world on authentic problems. The toolbox also allows us to compare and test algorithms or debug and test robots without the need for down-time. Often simulations aren’t accurate reflections of real-world applications due to multiple different reasons. At this point in time, hands-on time with the UR3 robot is unachievable and hence comments on simulation accuracy cannot be made.

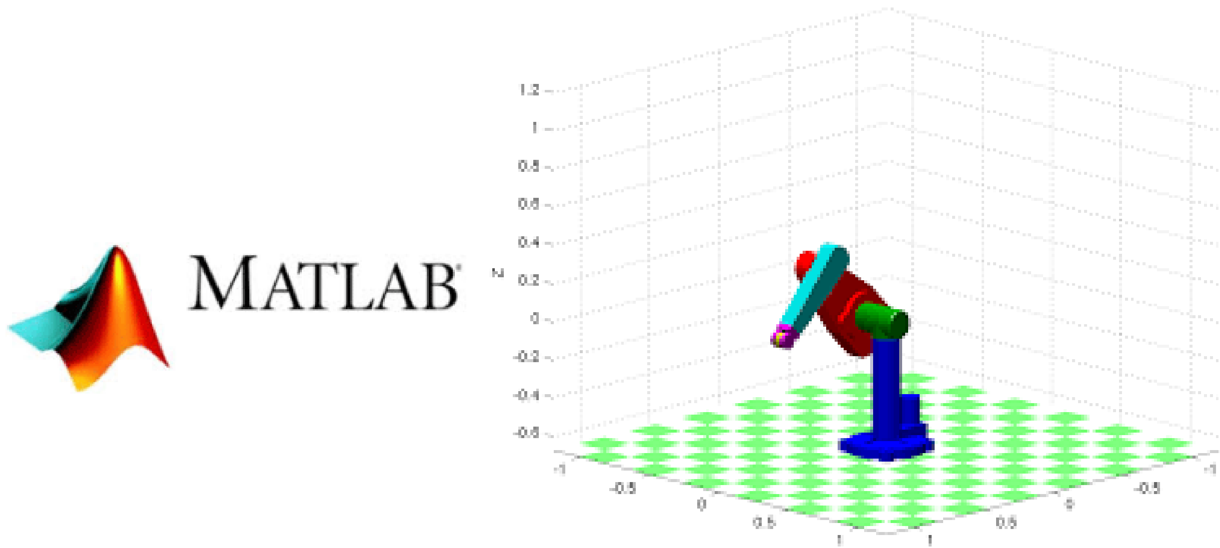


Figure 5: Matlab and robot model from Robotics Toolbox

6. Robot Precision

During simulations, and depending on the use of certain functions, the UR3 did not achieve the required end effector position or produced erratic movements. In a real-world application, precision carries high importance, especially when handling small, fragile or dangerous items. A likely cause for these errors would be the lack of onboard sensors to provide feedback on actual joint angles or end-effector position. Precision can be increased with the use of force-torque, vision sensors, and encoders among some others. Additionally, precision will be the result of mathematical algorithms used and reliable data such as measurements and joint angles.

In this simulation, the Quintic Polynomial method was used to calculate 50 sets of joint states from the initial end-effector position to the goal position. The ‘ikine’ function was first used to determine the final set of joint states but produced erratic movements and incorrect end-effector position on the UR3. Upon further inspection, ‘ikine’ on the UR3 returned joint states that were multiplied by $E*10^3$ or greater as seen in Figure 6. The ‘ikcon’ function was then used instead but still caused erratic movements, although it did reach the correct end-effector position. One solution considered was to mask out the orientation during inverse kinematics. This resulted in smooth motions but was not ideal as the end effector orientation was not correct. Finally, while using ‘ikcon’ both robots (UR3 and LinearUR5) were instructed to return to its home position after every movement. This not only resulted in smooth motions with correct orientations, but this also forced both robots to maintain an elbow up configuration to prevent collisions with the ground.

```
q =

   1.0e+04 *

  -8.8372   -2.1565    0.6477   -0.2729    5.7229   -6.8806
```

Figure 6: Matlab ikine result

This task demonstrates that simple and repetitive motions can easily be programmed or even 'hardcoded' but can become complex when end goals are constantly changing. The best control method would be feeding multiple waypoints towards the end goal instead of returning to the home position after every pass.

Ideally, the most accurate and precise method is physical guiding is where an operator manipulates the robot by hand by effectively 'teaching' it.

7. Sensing and Grasping Challenges

During initial simulations, when the ikine/ikcon functions determined the end-effector orientation for the UR3, the positive z-direction was upwards. This led to erratic movements by the arm before the end-effect finally approached the brick from underneath. In a real-world scenario, such movements will cause imminent collisions. Our aim is for the end-effector to approach from above the brick. To achieve this, the robot's 'tool' transform is rotated about x by 180 degrees making the positive x-axis face downwards.

In practical scenarios, vision cameras are often used to identify the object that is to be grasped. This allows the manipulator to make micro-adjustments based on the feedback being received by the vision sensor. The manipulator may also make use of pressure or photo sensors to determine if an object is being grasped.

8. Robot Safety

During this simulation, both the UR3 and LinearUR5 would place bricks onto the wall at the same time and in a real-world setting, collisions would be inevitable and could cause damage to the robots or payload. A solution to this problem would be to prevent the robots from simultaneously placing bricks on the wall. Instead, a robot should wait for the other to move away from the wall before placing its brick. Although attempted, this was not successfully implemented in the simulation. The manipulators should also be able to avoid all obstacles such as the conveyors or the platform, both of which are successfully implemented in the simulation.

With human safety in mind, fences have been erected either side of the conveyor to prevent anyone from entering the area while robots are in motion. Further devices could have also been added to increase safety. An E-stop button should be implemented to cease operation of robots and conveyors under emergency situations. Light curtains (Figure 8) should also be considered to ensure safety for workers and technicians who enter unsafe areas. In addition, tower lights (Figure 7) could be used to indicate the current operational status of the area.

Figure 7: Tower Lights

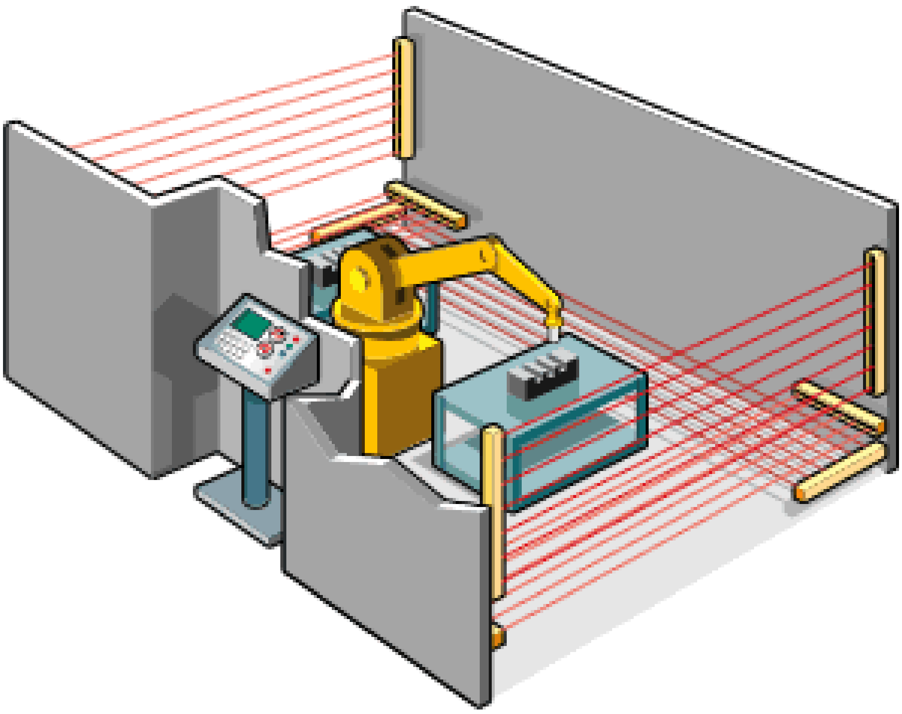


Figure 8: Photo curtains

9. References

Bélanger-Barrette, M. (n.d.). *Universal Robots' UR3 VS Sawyer from Rethink Robotics*. [online] blog.robotiq.com. Available at: https://blog.robotiq.com/universal-robots-ur3-vs-rethink-robotics-sawyer.

Soffar, H. (2019). Collaborative robots, Universal Robots UR3 importance, uses, advantages and disadvantages. [online] Science online. Available at: https://www.online-sciences.com/robotics/collaborative-robots-universal-robots-ur3-importance-uses-advantages-and-disadvantages/.

Robotics Business Review. (2018). How Robot Precision Technology Has Evolved Over the Years. [online] Available at: https://www.roboticsbusinessreview.com/manufacturing/robot-precision-evolves/#:~:text=%E2%80%9CPrecision%2C%20also%20referred%20to%20as [Accessed 4 Sep. 2020].

Vavra, C. (2019). Improving industrial robot accuracy and precision. [online] Control Engineering. Available at: https://www.controleng.com/articles/improving-industrial-robot-accuracy-and-precision/ [Accessed 6 Sep. 2020].

Risk Assessment-Robots-Controls, 18-12-20. (n.d.). [online] Available at: https://www.cesales.com/files/resources/2019-presentation-risk-assessment.pdf [Accessed 6 Sep. 2020].

LLC, exida com (n.d.). *Hazard Analysis and Risk Assessment of Collaborative Robotic Systems*. [online] www.exida.com. Available at: https://www.exida.com/Blog/hazard-analysis-and-risk-assessment-of-collaborative-robotic-systems [Accessed 6 Sep. 2020].

10. Appendices

10.1. Safe Work Method Statement

1. FACULTY/SUBJECT	
Faculty/Subject title	FEIT/Robotics
Subject supervisor/coordinator	Gavin Paul
SWMS prepared by	Ajal Singh

2. WORK ACTIVITY DESCRIPTION	
Describe the work activity E.g. Operating, Handling, Using.. Include names of hazardous equipment, substances or materials used, and any quantities and concentrations of substance(s) or reaction products.	• Correct Installation of 2 robots (UR3 and Linear UR5), conveyors, fence and platform • Programming of robots

3. HAZARDS: Choose those hazard types that will need to have control measures in Section 4			
Work Environment • Working Outdoors/fieldwork • Clinical/Industrial setting	**Plant** • Moving parts (Crushing,friction, cut, stab, shear hazards)	**Chemical** • N/A	**Ergonomic/Manual Handling** • Repetitive or awkward movements • Lifting heavy objects • Over reaching
Electrical • Plug in equipment • High voltage	**Radiation** • N/A	**Biological** • N/A	**Psychosocial** • N/A

4. CONTROLS MEASURES: Choose those that apply for hazards identified		
Eliminate/Isolate/Substitute / Engineering Controls • Remove hazard • Restrict access • Reprogram equipment • Guarding / Barriers / Fencing • Regular maintenance of equipment • Redesign of workspace / workflow • Operational testing	**Admin specific: Licenses/permits Work Methods** • Training Information or Instruction • Licensing or certification of operators • Test and tag electrical equipment • Restricted access	**Emergency Response Systems** • First aid kit • Emergency Stop button • Remote Communication Mechanism
Other controls not listed		

5. PPE REQUIRED (Tick those that apply)

 EYE PROTECTION	 HEARING PROTECTION	 CLOSED-IN FOOTWEAR MUST BE WORN	 HAND PROTECTION	 FACE SHIELD	 PROTECTIVE CLOTHING
✓		✓			
 RESPIRATORY PROTECTION	 LONG HAIR MUST BE CONTAINED	 HEAD PROTECTION			
		✓			

6. EMERGENCY EQUIPMENT

7. WORK ACTIVITY STEPS

BEFORE YOU START:

MEASURE OUT THE ENVIRONMENT AND PLAN ACCORDINGLY.
VIEW ALL REQUIRED INSTALLATION DOCUMENTATION
PROGRAM ROBOTS UNDER SIMULATION ENVIRONMENT

STEPS IN WORK ACTIVITY:
INSTALL PLATFORM
INSTALL UE3 AND LINEARUR5 ROBOTS
INSTALL CONVEYORS
INSTALL SAFETY FENCING
TEST SIMULATED ROBOT PROGRAM
MAKE PROGRAM ADJUSTMENTS IF REQUIRED

REMAIN CLEAR OF WORK AREA WHILE UNDER OPERATION!

EMERGENCY PROCEDURES:

PUSH E-STOP BUTTON
APPLY FIRST AID IF REQUIRED
CALL 000 IF NECESSARY

TRAINING REQUIRED:
ROBOT AND CONVEYOR INSTALLATION
ROBOT PROGRAMMING

8. SIGN OFF

PREPARED BY:	**SUPERVISOR**	**DATE:** 6/9/20
NAME: AJAL SINGH	**NAME:** ~~GAVIN PAUL~~	**REVIEW DATE:**

10.2. Risk Assessment Form

Work area / operation	Plant		**Assessors name**		
Other persons consulted				**Date**	6/9/20

ACTIVITY - Describe hazardous activities related to the work area or operation.	**ASSOCIATED HAZARDS**	**INHERENT RISK** - Harm that could occur from these hazards if controls fail or are not in place.	**EXISTING CONTROL MEASURES**	**PROPOSED CONTROL MEASURES** - Proposed action to minimise risk to an acceptable level.	**TARGET DATE** - To implement proposed controls	**RESIDUAL RISK LEVEL** (H,M,L)
Moving robot	Robot -human collision	Human Injury Robot damage Payload Damage	Fencing	E-Stop button Light Curtains	20/9/20	M
	Robot-object collision	Robot damage Object Damage Payload Damage	Clear workspace of potential obstacles	Collision Detection	20/9/20	L
	Robot- Robot collision	Robot Damage Payload Damage	None	Asynchronous Robot movement at wall	20/9/20	H
Pick and place	Payload too heavy	Robot Damage Payload Damage	None	Assign robot payloads based on robot specs	20/9/20	M
	Incorrect end-effector pose (position and orientation)	Fail to build wall	Thorough testing	Feedback e.g. vision sensor	28/9/20	L