Project Deliverable C - Design Criteria

GNG 1103 Group 3

Benoit Tremblay (300236751)

Rebeca Poulin (300236741)

Jess Beardshaw (300227707)

Isabelle Barrette (300228564)

Sarah Alkadri (300245117)

University of Ottawa

February 6th 2022

TABLE OF CONTENTS

Wrike link:
https://www.wrike.com/workspace.htm?acc=4975842#path=folder&id=824968763&c=list&vid=64521612&a=4975842&so=11&bso=10&sd=0&f=&st=nt-1&sfi=1

Introduction

John Faurbo and the Royal Canadian Navy need a robotic arm that has the ability to detect, remove and paint over corroded areas to ensure proper maintenance on the Halifax class warships. Cost, ease of use, proportions and safety of product are key to creating the best product for the client's needs.

To fulfill the client's needs, this deliverable contains a list of prioritized needs, target specifications, and similar existing products for benchmarking.

This deliverable contains a list of the user's interpreted needs, including function and non-functional requirements, as well as metric constraints. The target specifications include the ideal and marginal values provided by John Faurbo. In addition to the design criteria, benchmarking and research of other solutions that satisfy our interpreted needs are included in this deliverable. The information in this document will be used to create the conceptual design in the future.

Translating Needs into Design Criteria

Rank	Needs	Design Criteria
1	Safety system	Shut off sensors for passing people/stability of arm
2	Ability to move and complete the 3 tasks with inverse kinematics	1. Draw a shape or logo 2. Sand blast or water blast a grid area 3. Inspect a low oxygen space in 360deg Uses inverse kinematics in code to indicate to robot where to go The arm uses 3 degrees of freedom
3	Holds and withstands the end effectors (weight and pressure)	Weight of end effectors(grams)/ Water pressure (psi)/Paint pressure(psi)
4	Easy switchable end effectors for different tasks	End-effector shape/fragility/End-effector weight(grams)
5	Lightweight and compact	Weight of end-effectors/arm (grams/lbs)
6	Simple/familiar code, files and hardware	Arduino Uno/Readable code/Common programming language(C, C++, Python)
7	Easy to learn to operate	Time to learn to operate the robot(min)/Operable by someone with little technical experience
8	Infrequent repairs/replacement pieces	Lifespan before minor repairs(months)/Lifespan before major repairs(months)
9	Efficiency/speed	The area of sight/vision/range to spray and observe with a camera per hour.(m^2/hour)
10	Water resistant/able to work in near water environment	Operating conditions: water

11	Light up a dark area to take picture scan	Light system connected to camera/ brightness of lights(cd)

List of Prioritized Criteria

	Functional Requirements	Relation	Value	Units	Verification Method
1	The end effectors can hold and operate camera, paint and anti-corrosion sprayer.	=	Yes	N/A	Test
2	The arms supported weight.(i.e.the end effectors weight)	>=	750	g	Final test, weighing, analysis
3	The end effectors are easily interchangeable.	=	Yes	N/A	Test
9	The arm and end effectors remain stable and withstand pressure from hose and paint.	>=	180	psi	Test
4	The robot is easy to learn to operate.	<	45	minutes	Test
5	The end effector and parts are 3D printable	=	Yes	N/A	Analysis
6	The arm is powered by 120-volt outlets	=	Yes	N/A	Test, final check
8	The arm and end effector can be assembled quickly.	<	25	minutes	Test
10	The code uses a common programming language such as C or C++ or Python.	=	Yes	N/A	Test
11	The code uses inverse kinematics.	=	Yes	N/A	Test
12	The design of end effectors and code are open source.	=	Yes	N/A	Test
13	The arm is controlled using an Arduino Uno.	=	Yes	N/A	Test

	Constraints	Relation	Value	Units	Verification Method
1	The weight of the end effectors.	<=	750	g	Analysis
2	The dimensions of the end effectors.	<	60	mm	Analysis
3	The cost of the project.	<	50	$	Estimate, final check
4	The weight of the arm.	=	20	lbs	Analysis
5	The dimensions of the arm.	<	1	m^2	Analysis
6	The lighting of the arms surroundings.	>	Yes	N/A	Analysis
7	The area of sight/vision/range to spray and observe with camera.	=	1	m^2	Analysis
	Non-Functional Requirements	**Relation**	**Value**	**Units**	**Verification Method**
1	The robot is safe to operate and be around while working.	=	Yes	N/A	Test
2	The robot is operated by someone with limited technical experience. (High School Education)	=	Yes	N/A	Ask non-expert
3	The parts, robot and code are easily repairable.	=	Yes	N/A	Can be 3D printed
4	The robot is compact and transportable.	=	Yes	N/A	Analysis of dimensions
5	The robot's lifespan before minor repairs needed.	>=	2-3	months	Estimate, analysis
6	The robot's lifespan before major repairs needed.	>=	6	months	Estimate, analysis
7	The robot is aesthetically pleasing.	=	Yes	N/A	Client Meeting

Target Specifications

Importance (1-10)	Criteria	Target	Description
10	Degrees of Freedom	3	The degrees of freedom allows for optimal movement of the end effector to perform the required tasks. The use of inverse kinematics is required as well.
10	Weight supported by end effector	≥ 1 kilogram	The more weight supported by the end effector, it will be able to perform jobs at a higher complexity, while decreasing the risk of failure due to exceeding the maximum supported weight. Ensuring the end effector is properly stabilized at the connecting joint
7	Weight of robot	≤ 20 lbs.	The greater the weight, the more difficult the crew will have transporting the robot to different locations. This is essential as the robot is required to perform jobs in a variety of locations among the ships. The lower the weight, the easier manufacturing of the robot pieces via 3D printing on the ship.
7	Weight of end-effector	≤ 750 grams	The weight of the end-effector should be very inferior to that of the entire arm to assure that the centre of mass is within the dimensions of the arm. It will also be easier to handle if it is lightweight.
8	Pressure from water hose	140psi ≤x≤ 180psi	The pressure from the water hose cannot be too high or it risks applying too much force to the end-effector and the arm. A water pressure of 140 to 180 psi is the ideal pressure to rinse off the sides of the boat prior to applying paint. Ensuring the base and the end effector is stable to withstand a pressure preferably greater than 180psi to eliminate the risk of failure.
9	Volts Resistance	= 120-volt	The robot will be powered via 120 volt electrical outlets scattered throughout the ship. Utilizing the power source already present and accessible aboard prevents creating an alternative power source costing resources and money.
6	Painting Speed	≥ $1m^2$/hour	The client requests that they need to be able to leave the robot for periods of time, and return to a section fully painted. It will be fast enough to strip the surface of corrosion, let it dry and then apply a layer of paint in a $1m^2$ radius of the robot within an hour. At this rate, it can be supervised maybe once every hour and moved every few.

Changes Since Deliverable B

There have been no significant changes to the development of the project since deliverable B. Our group has discussed the possibility of programming the previous Thor solution using inverse kinematics, as well as developing our own unique solution.

Technical Benchmarking

Rank	Source	Technical Requirements	Similar design criterias/needs (Table 1)
1	https://hackaday.io/project/12989-thor	- Open source - Designed with FreeCAD - Arduino Mega - 3 different end effectors shown (claw, screw and scoop claw) - Powered by stepper motors for arduino in descending sizes to reduce weight needed to support motors - GT2 pulley and belt for 360 rotation	-Easy switchable end effectors for different tasks -Arduino/Readable code/Common programming language(C, C++, Python) -Holds and withstands the end effectors (weight and pressure) -Lightweight and compact -3D printable
2	https://blog.floydhub.com/localize-and-detect-corrosion-with-tensorflow-object-detection-api/	- Describes how to detect corrosion using program that scans images - Explains process of how to code task using python - Program will say if there is/is not presence of rust in the image	-Ability to scan a photo and detect rust/corrosion -Coded on python, open source
3	https://hackaday.io/project/12989-thor/log/44018-inverse-kinematicsb	- This website demonstrates formulas to understand and begin to program the Thor arm with inverse kinematics. - Physics formulas that explain logic of inverse kinematics in robotics	-Inverse kinematics
4	Generate Code for Inverse Kinematics Computation Using Robot from Robot Library - MATLAB & Simulink (mathworks.com)	- Algorithm and basic code in different languages that we can base off of and compare to some of the others we find. - It does use a robot arm and inverse kinematics which are helpful to our situation.	-Inverse kinematics -Arduino/Readable code/Common programming language(C, C++, Python) -Lightweight and compact
5	Inverse Kinematics - Nikita Lukhanin	- Inverse kinematics code for a three joint robot arm programmed and run with Arduino - Made for an open source arm - Gives a good idea of the format and algorithm needed	-Arduino Uno/Readable code/Common programming language(C, C++, C#, Python) -3D printable -Inverse kinematics

6	https://www.sunfounder.com/products/arduino-robot-arm-kit#:~:text=The%20Robot%20Arm%20Kit%20for,them%20move%20on%20your%20computer.	-Open source MCU Arduino UNO -Servo expansion board - Can be controlled by 4 potentiometer buttons - Flexible Arm with a gripper that opens to 3.54 inches and 260 degrees as well as a radial wrist motion of 180 degrees - It has an extensive elbow with range of motion of 180 degrees	-Arduino Uno/Readable code/Common programming language(C, C++, Python) -Lightweight and compact -Easy to learn to operate
7	https://www.alanzucconi.com/2018/05/02/ik-2d-2/	- Inverse kinematics code in C# - 2 dimensional - Explains in depth the different components involved in inverse kinematics for each joint	-Uses inverse kinematics in code to indicate to robot where to go -Arduino Uno/Readable code/Common programming language(C, C++, C#, Python)
8	https://hackaday.com/2014/03/27/3-dof-open-source-robot-arm-is-just-the-beginning/	- Open source palletizing robot - Arduino Uno firmware - 3DOF - Inverse Kinematics - Holds a pencil and writes message	-Draw a shape or logo -Arduino Uno/Readable code/Common programming language(C, C++, Python)
9	https://www.kickstarter.com/projects/ufactory/uarm-put-a-miniature-industrial-robot-arm-on-your	- 4DOF - Different end effectors for different tasks - Possible idea for light (design criteria)	-Light up a dark area to take picture scan(with light attachment) -Easy switchable end effectors for different tasks
10	https://github.com/MarginallyClever/Arm3	- Arduino Uno firmware for inverse kinematic robot arm	-Arduino Uno/Readable code/Common programming language(C, C++, Python) -Inverse Kinematics

Conclusion

In order for our team to develop a strong solution to the problem initialized by John Faurbo, a clear set of interpreted needs and design criteria are essential. Functional and non-functional needs, as well as constraints and target specifications must be thoughtfully planned in order to create the best solution to the problem.

To work on deliverable D, the information gathered in this document will be used to develop and create potential concepts for solutions that follow the constraints and requirements.