Robotic Applications Assignment REPORT

# Table of Contents

[1 Table of Contents 2](#_Toc121802731)

[2 References 3](#_Toc121802732)

[3 Robot Strategy 4](#_Toc121802733)

[3.1 Introduction 4](#_Toc121802734)

[3.2 Modelling 5](#_Toc121802735)

[3.2.1 State machine 5](#_Toc121802736)

[3.2.2 ROS Nodes 6](#_Toc121802737)

[3.2.3 ROS Services 8](#_Toc121802738)

[3.3 Implementation 10](#_Toc121802739)

[3.3.1 Nodes which run independent to the state of the robot 10](#_Toc121802740)

[3.3.2 Task Nodes 11](#_Toc121802741)

[3.3.3 droppingIntoBin 15](#_Toc121802742)

[3.4 Discussion and Conclusion 16](#_Toc121802743)

[4 Ethical Consideration 17](#_Toc121802744)

[4.1 First full-time robot teacher employed 17](#_Toc121802745)

[5 Self Assessment 19](#_Toc121802746)

[5.1 Programming assignment 19](#_Toc121802747)

[5.1.1 Completing scenario 19](#_Toc121802748)

[5.1.2 Screen cast 19](#_Toc121802749)

[5.1.3 ROS usage 19](#_Toc121802750)

[5.2 Report 20](#_Toc121802751)

[5.2.1 Strategy 20](#_Toc121802752)

[5.2.2 Ethics 20](#_Toc121802753)

[5.3 Overall marking 20](#_Toc121802754)

# References

|  |  |
| --- | --- |
| [1] | A. Blasdel, "wiki.ros.org," Open Robotics, 13 August 2015. [Online]. Available: http://wiki.ros.org/Robots/fetch. [Accessed 12 December 2022]. |
| [2] | I. Satio, "ROS/Tutorials/CreatingMsgAndSrv - ROS Wiki," Open Robotics, 30 August 2022. [Online]. Available: http://wiki.ros.org/ROS/Tutorials/CreatingMsgAndSrv. [Accessed 12 December 2022]. |
| [3] | R. Zickerl, "vision\_opencv - ROS Wiki," Open Robotics, 10 March 2020. [Online]. Available: http://wiki.ros.org/vision\_opencv. [Accessed 12 December 2022]. |
| [4] | "MoveIt Motion Planning Framework," PickNik Robotics, 2019. [Online]. Available: https://moveit.ros.org/. [Accessed 12 December 2022]. |
| [5] | Z. Chang, S. Lijuan, T. Tong, Z. Ziwei, P. Xiyuan, S. Yidong, L. Yamin and O. Jianjun, "Associations Between Academic Stress and Depressive Symptoms Mediated by Anxiety Symptoms and Hopelessness Among Chinese College Students," 5 March 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8906854/. [Accessed 12 December 2022]. |
| [6] | A. Khanlari, "Teachers perceptions of using robotics in primary/elementary schools in Newfoundland and Labrador," Memorial university of Newfoundland, June 2014. [Online]. Available: https://core.ac.uk/download/pdf/33556818.pdf. [Accessed 12 December 2022]. |
| [7] | H. Groarke, "The Impact of Smartphones on Social Behaviour and Relationships," Department of Psychology, Dublin business School, March 2014. [Online]. Available: https://esource.dbs.ie/bitstream/handle/10788/2096/hdip\_groake\_h\_2014.pdf?sequence=1&isAllowed=y. [Accessed 12 December 2022]. |

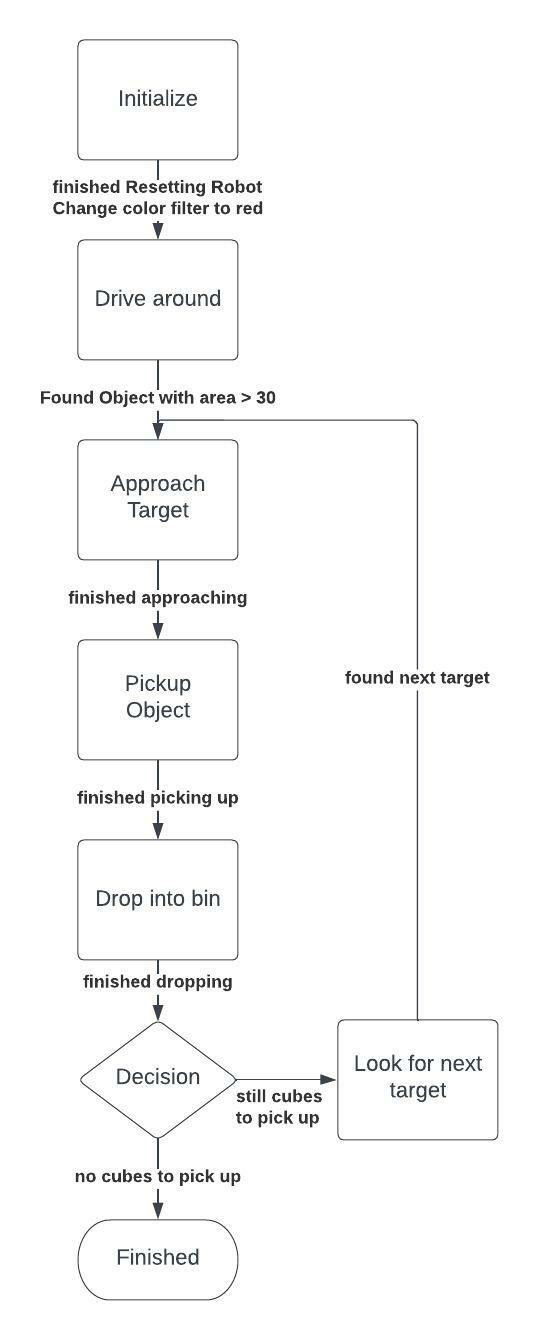
# Robot Strategy

## Introduction

The task for this assignment, using a fetch robot in ROS [1], was to navigate through a maze to find two red cubes on top of a pillar. These two cubes have to be picked up and dropped inside a green bin. In the next section I will be explaining how the task at hand has been broken down and how everything connects to each other. In the section implementation I will give an in-depth explanation on how each broken down step works. After that I will be reflecting on my approach and talk about how it performs.

## Modelling

### State machine

The above explained task has been broken into seven smaller steps. These can be seen in the diagram below:

First, we start with an Initialization state which resets the robot to its standard values. After this has been done and the colour filter was switched to filter for red objects we start driving around. We finish the driving around state once we find a target with an area value greater to 30. Then the robot will start approaching this target and get into a final approach position. Once this approach position has been reached, we attempt to pickup the object. After the object has been picked up, we change the colour filter to filter for the green bin. We approach this bin and drop the target into it. After we finished dropping the target we decide if we still need to pickup objects or if we finished dropping all objects. If we finished dropping all objects we go into a finished state where the robots stays where it is. Otherwise, if we still have objects to be picked up, we look for the next target. Once we found the next target, we repeat everything after driving around.

Important to notice is that there is a program constantly running which looks for a target. This does not get started by the state machine or stopped by the state machine.

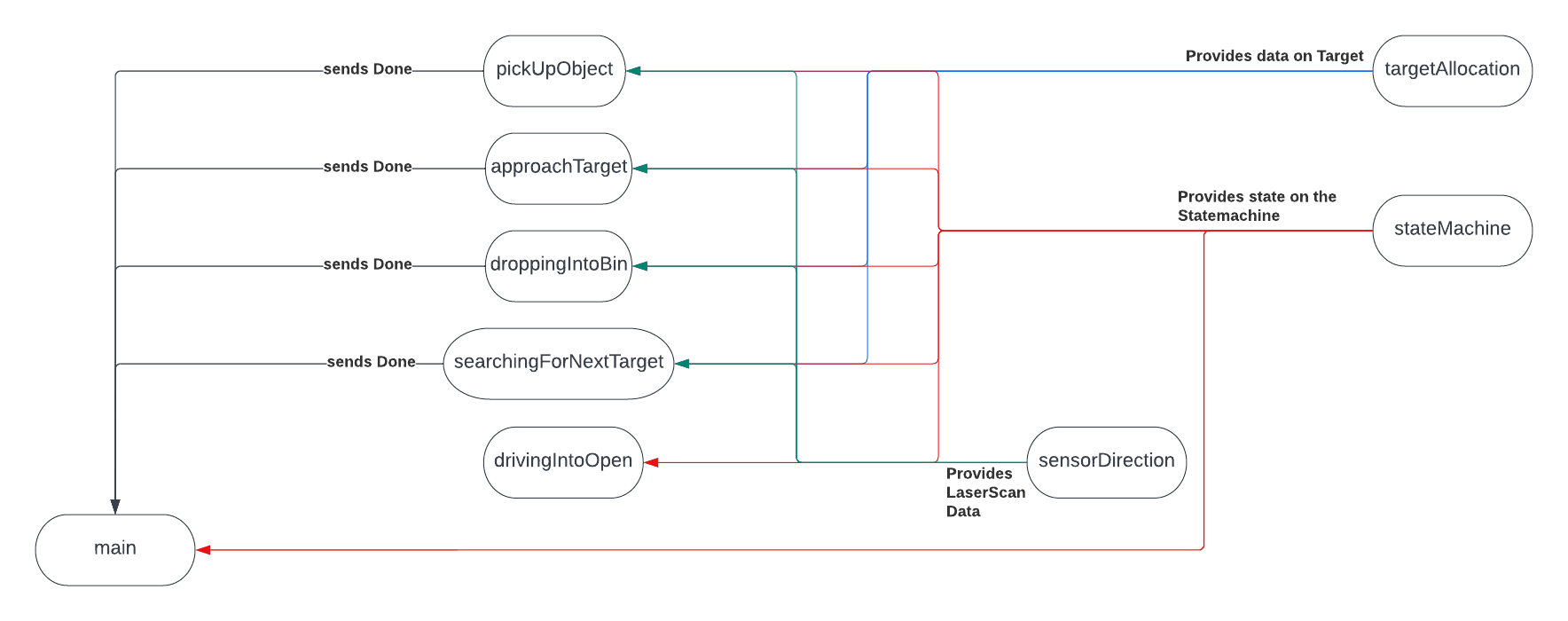
Everything excluding initialization and finished has its own ROS node. These nodes are being started at the creation of the ROS environment and will be executed depending on the state we are in. This enables us to have a big modularity for each module. Especially if we want to change the order of the execution of these smaller tasks, we just change by which state they are being triggered. Or we are able to export these on their own.

I will now be going into my nodes and the services that are being used.

In the following section I will be talking about the communication between nodes. I have used ROS msg and srv for this. To achieve this, I used ROS wiki on msg and srv [2].

### ROS Nodes

In my approach I create multiple nodes where some nodes are just used to get access to values which are being constantly updated and broadcast. Other nodes are used to execute the tasks major tasks of the state machine.

Following the Diagram how the data flow is between the nodes

Due to the fact that it is hard to tell how the flow is I will be listing this. I will be explaining what kind of data is being send and why in the implementation section.

#### targetAllocation

This node provides data for:

* pickUpObject
* approachTarget
* droppingIntoBin
* searchingForNextTarget

Msg type being used: ObjectPosition.msg

#### sensorDirection

This node provides data for:

* pickUpObject
* approachTarget
* droppingIntoBin
* searchingForNextTarget
* drivingIntoOpen

Msg type being used: SensorDirection.msg

#### stateMachine

This node provides data for:

* pickUpObject
* approachTarget
* droppingIntoBin
* searchingForNextTarget
* drivingIntoOpen
* main

Msg type being used: StateMachine.msg

#### Done with task

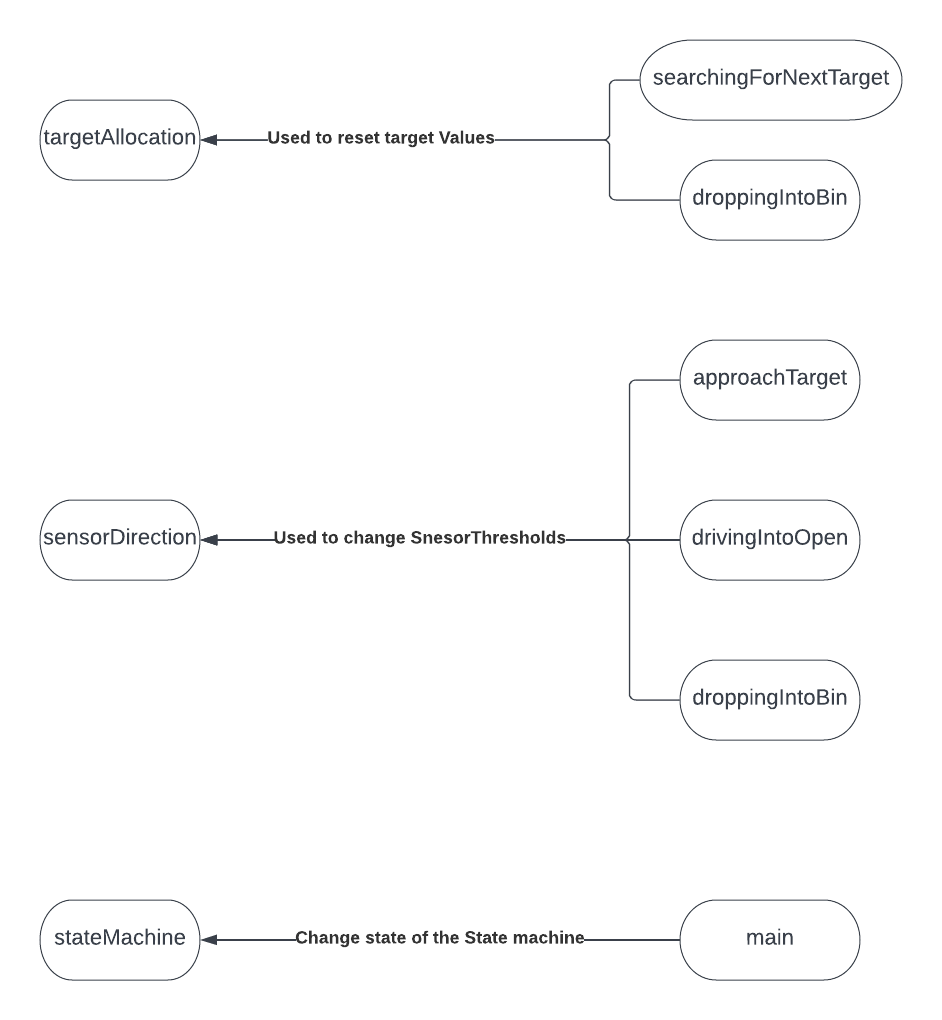
Following nodes send a done state to main:

* pickUpObject
* approachTarget
* droppingIntoBin
* searchingForNextTarget

Msg type being used: DoneWithState.msg

### ROS Services

I do use three ROS services which are ResetTargetService, SensorDirectionService and StateMachineService.



#### ResetTargetService

This service is used for resetting target values. To ensure that no old values are being published anymore.

* Started by
  + targetAllocation
* Used by
  + droppingIntoBin
  + searchingForNextTarget
* Input
  + bool resetTarget
* Response
  + bool response

#### SensorDirectionService

This service is used to change the threshold lasers for forward, left. To tell which lasers are used to calculate each average direction

* Started by
  + sensorDirection
* Used by
  + approachTarget
  + drivingIntoOpen
  + droppingIntoBin
* Input
  + int16 thresholdLeft
  + int16 thresholdForwardMin
  + int16 thresholdForwardMax
  + int16 thresholdRight
* Response
  + String[] response

#### StateMachineService

This service is used to change the state of the state machine.

* Started by
  + stateMachine
* Used by
  + Main
* Input
  + Int8 changeState
* Response
  + bool response

## Implementation

In this section I will be talking about how I was able to implement the above strategy for completing the task at hand and how each smaller task has been implemented.

### Nodes which run independent to the state of the robot

#### Main

Main implements the logic for the state machine. It also sets the state which is being published by the stateMachine into the right state. It further does all the logic checking, such as if a task is done and what else we need to do. Main also implements the resetting of the robot joints.

#### targetAllocation

Target allocation uses the OpenCV library [3] of ROS to filter for objects set by a colorFilter launch file. With the OpenCV library we are able to get the position of a filtered object. These positions are being used to constantly broadcast its positions for every node that needs it. It further implements the ResetTargetService which allows nodes to reset the target values. Which is really helpful after switching colorfilters. Moreover, targetAllocation also creates the transform for the transform tree for the object. The transform that is being broadcasted to the transform tree is only being broadcasted if we have a depth value to the target and if we have a target area above a certain threshold. Otherwise, the previous target position will be broadcasted. This allows us to have a transform to use even if an object gets in the way of the target or if the target becomes too small because something is blocking the camera. Also important to mention is that the target allocation looks through all contour\_moments. If the contour\_moments have a length greater 0, then we also check through each moment and choose the target with the lowest centre in Y. Thus always allowing us to target the cube which should be stationed higher in the world.

#### sensorDirection

Sensor direction constantly broadcasts average values for left, forward and right based on the thresholds for which lasers should be used for what. This again allows nodes to use different thresholds for these while still calculating average distances for a range of sensors from the base\_scan\_Laser. It is possible to change these thresholds using the SensorDirectionService.

#### stateMachine

The stateMachine is used to broadcast the current state the robot is in and should be in. These allows nodes to only run their code based of the state. This node implements the StateMachineService to change the state.

### Task Nodes

Now I will be talking about the nodes which each complete a task in the state and how they achieve this task.

#### drivingIntoOpen

Driving into open is used to go through the maze and drive into the open. This code keeps being run until the conditions of the next state have been met.

##### How does it achieve this

The program achieves this by having two states: inMaze and inOpen.

Depending on the state the robot behaves differently but share some common logic

##### Common logic

Keep the distance left and right the same otherwise turn to balance this out. Move forward unless there is a object closer than a certain value, then we turn left or right depending on where we have more space.

###### inMaze

If still inside the maze the robot looks at a set max range for left and right. If the value exceeds this max range, then we just say that we are at max range. This way we more likely only react to the objects which create a path and do not take into account the space between them which leads to walls further away. Otherwise, we would be trying to stay in the middle of the further away walls and not follow a set path.

This logic gets changed into an inOpen state, once the left or right distance exceedes a certain threshold meaning that there is a huge open space to our right or left. This value has been set to 6m.

##### inOpen

Once we enter this state, we do not set max ranges for left or right and thus we will be moving into the open space and stay in the middle while moving down.

#### approachingTarget

When we enter this state, we have found a target we want to approach to. The goal of this state is to achieve a final position to the target to ensure a predictable and replicable endPosition to the target. To achieve this, we go through multiple states:

##### First State

We reset all global variables to ensure on that we do not have any saved values from a previous run

##### Second State

We approach the target using until we are at a certain distance to it. While we do this, we align ourselves such that the target is in the horizontal centre of the camera. We also move our camera such that the target is in the vertical centre of the camera.

##### Third State

We close in to the target to a certain distance from the camera while still horizontal and vertically aligning to the target.

##### Fourth State

We check if we are not looking down enough on the target. If this is the case, we readjust our height to ensure a similar height difference from the camera to the target. This is done by increasing the torso height until the object is at the Vertical bottom of the camera and then we realign to the vertical centre using the camera tilt.

##### Fifth State

We again check if our vertical and horizontal alignment of the object to the camera centre has changed in anyway. Further depending on the torso height, we will also move closer to the target than before as the height difference means a bigger difference in direct distance to the target.

Once this state is done, we publish a message saying we are done.

#### pickUpObject

After the approach to the target, which is really consistent and almost every time a precise replica it is easier to create the same conditions for the robot to pick up the target.

To pick up the target and move the arm of the robot we use a library called move\_it [4]. This helps us to set goal positions for the robot to go into gripping position and any kind of arm movement.

The strategy of this task is to recreate the same movement of the arm as much as possible to recreate the same grip almost every single time. This ensures a really good grip of the target. After picking the target up and ensuring the grippers are closed to a certain distance, we understand that the target has been picked up well. This means we move backwards through the pre-sets to a certain position. After that we distance ourselves from the pillar. After the distancing has been done, we move our cube to a neutral position and rotate the wrist. This neutral position enables us to make sure the arm and the cube are not in the way for the next states. The rotation of the wrist helps us to make sure we do not loose the cube through gravity while moving around with it. After the neutral position we reset our torso height and our camera position.

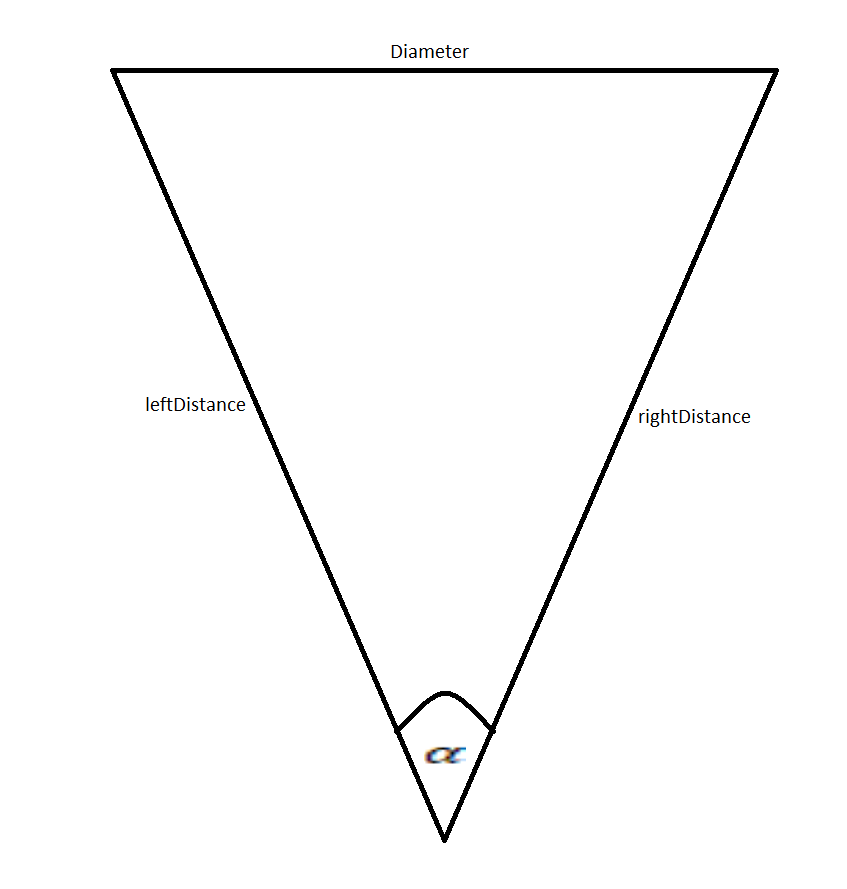
To guarantee we do not hit the pillar the cube is stationed on, a cylinder collision object is being created around that pillar. The height of that pillar for the first pre-set of position is set to the cube height. This allows us to avoid the cube itself and not hit it. The height will then be lowered after the arm arrives at the first position to make sure we reach the end position to grip.

The radius of the cylinder is being calculated using the LaserScan values. An explanation on how this is done:

First, we take the distance exactly in front of us and look at all lasers which give us a value within that distance multiplied by 2. With the number of lasers, we use and the distance furthest left and right, we are able to calculate the diameter of the cylinder with following formula:

220° is the maximum radius the laser scanner covers. 662 represents the number of lasers which cover 220°

With this radius and the distance to the most left and right we create a basic triangle:



Using this information and following Equation we can calculate the Diameter:

This helps us get the Radius:

I used a buffer value of 12cm in the program due to the fact that my lasers don’t touch the full diameter of the pillar. 12cm was chosen due to it being quite reliable in testing.

Now that we have the height and the radius of the cylinder we just need to find out where to put it. It would be possible to put the cylinder centre underneath the target but we will be really inaccurate with that.

The following has been done to find out where to place the cylinder in the world:

We place the x, y and z centre based of base\_link and calculate each of them.

##### Cylinder centre in z

The height centre for the cylinder was calculated using following formula:

##### Cylinder centre in x

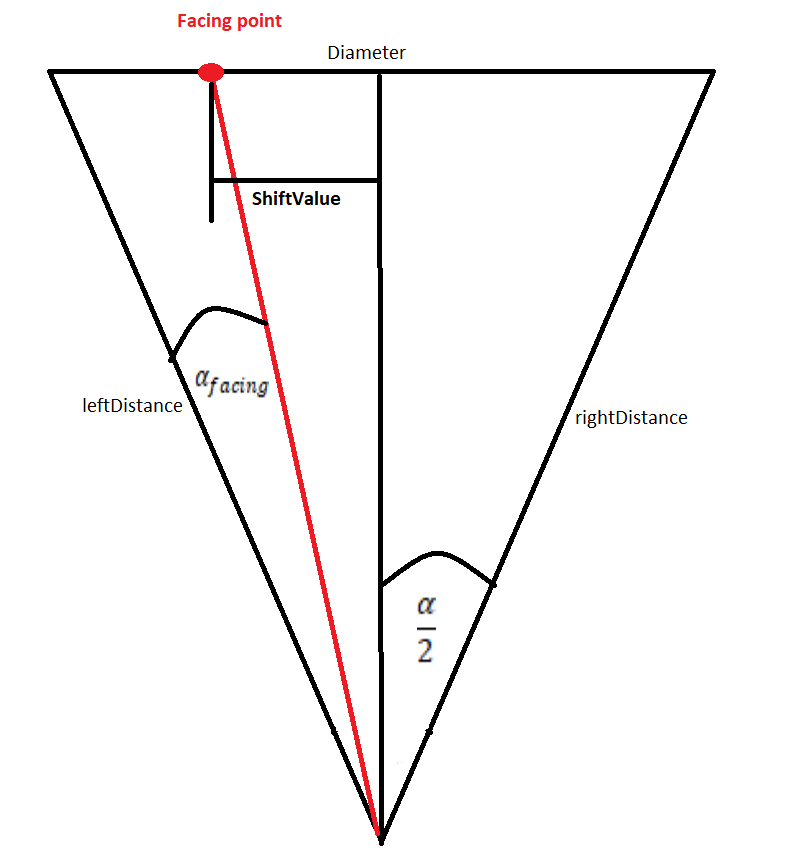
This reflects the distance of the centre to base link. Thus, we use following formula:

0.235 represents the distance difference between base\_link and laser\_link

Due to the fact that the smallestDistanceOfSensorValues seems to not properly represent the actual distance a buffer of 1cm was used, which again works really well.

##### Cylinder centre in y

How much we need to shift in y was quite hard to find out. But doing the following works really well.



First, we can assume that the centre of the pillar is at exactly and then we calculate the angle we are facing at. So, we calculate the facing angle:

sensorDistanceMiddle represents the distance for laserValueIndex = 331. The indexOfLaser at this distance gives us the number of lasers we use from the leftMostDistance to straight forward.

Now knowing that represents the full diameter and knowing what value is, we can calculate the shiftValue:

This shift value represents our cylinderYCentre. To ensure we are shifting into the right direction we do need to multiply this by -1

### droppingIntoBin

After we picked up the cube, main changes the colour filter to filter for green objects.

Strategy used to drop the cube in the bin is following:

We turn towards the side that has more space and keep turning till we see the bin. Then we approach the bin while keeping it in horizontal and vertical centre of the camera. We close the distance to the bin until its below a certain threshold of the camera. A position based of the transformed target of the bin and offsets was set to get the arm into a position to drop the cube. This position has an orientation that represents the gripper still holding the cube from above and below. This actually helps making the arm movement more predictable and not having the need to create a collision object around the bin.

#### searchingForNextTarget

If a cube was dropped and we still have more to pick up, main changes the color filter to red for red object. We then turn to the side which has more space and keep turning until we see the next object. After we found it we reset the robot arm to its original position and make sure torso is also at joint position 0.

## Discussion and Conclusion

I personally think my implementation does really well and is really flexible. Major changes such as:

* Way bigger distance between the walls and maze obstacles might need a different left and right value for inMaze maxLeftRight
* Way smaller distance to the back of the “cube room” might ensure that I do not go into the inOpen state. This again can be adapted by changing the switchToOpenRange.
* Cubes are not sitting at the front part of the pillar facing the robot. This will result in a problem where I will keep driving into the pillar.
* A height change of the bin will result in my arm most likely hitting the bin and also not aligning properly to it, because I do not change my torso height or put a collision object around the bin.

Other than those changes I do think my robot strategy can adapt to any environment, without the need of minor or major adjustments. I also think taking all of this into account would be out scope of the assignment. Because a robot is made to work in a certain environment and will do best in the environment it is intended to work in. If the environment changes to much different approach might be needed. Being able to recreate same joint configurations using a good offset approach, enables me to minimize a lot of random and unpredictable arm movements.

So far testing the program 20 times in a row I only dropped the cube on the higher pillar ones. I have a problem when touching the cubes, the first time the smaller pillar jumps up at me due to it being clipped slightly into the ground. This results in my arm touching the pillar. For the taller pillar it seems that in my environment on first touch the cube is glued to the pillar. So the pillar is moving to the sides without my arm ever touching the pillar itself.

I do think I am making use of a lot of the ROS middleware such as topics, msg and srv. I could have implemented ROS parameter server for states and actions for finishing smaller tasks. But overall I do think I use a lot of ROS’s middleware.

# Ethical Consideration

## First full-time robot teacher employed

A robotic teacher might sound astonishing to students and could be seen as a nightmare for teachers. They can have a great deal of upsides to institutions and be surprisingly difficult to work with for students. I will be exploring the impact a robotic teacher could have on students, teachers, the institution that is running it and society itself. I also want to inform that I will be viewing this topic from a high school perspective. To consider each stage of institution would blow this article out of proportions.

To students a robotic teacher would be a welcome idea. The possibility of endless knowledge and not having to deal with teachers that are not passionate about their subject. For students it would be possible to ask any kind of question that is answerable. The ability for the robotic teacher to access the internet and have a factual answer ready for any kind of question that has been answered before is a huge upside for the student. Further the robotic teacher, if done in this way, has the ability to refer the student to the right literature on the internet. Moreover, the robot would be able to give an unbiased view on the subject. This could be tampered with, but a regulation on the design on robots could hinder this problem.

But we also need to take a look on the other side. A robotic teacher would be one source less for students to have a professional social environment with someone that is in charge of the class. This could lead to students, with a damaging behaviour towards the class environment, to go out of control. Furthermore, who would take responsibility if anything happens to a student? The parents are giving their children into the care of the institute and the institute gives this responsibility to the teacher that is teaching the class. What if a fire breaks out and the students are not able to get to the fire exit, or if there is a medical emergency. This could be solved by having medically trained staff on standby but it could become impossible to deal with every situation if there are too many at once.

Another solution could encompass the requirement of more training on the student side. If every student has the ability to provide first aid and how to deal with almost every possible scenario this could be beneficial to society itself in the long run. Students do need to be more mature. But how realistic is this. We are already putting loads of academic pressure on our children and this correlates to a higher likelihood of depression. This was shown by a study from PubMed Central [5]. Wouldn’t this extra pressure and responsibility make it worse?

So far, we have been talking about this topic from the students view. Now I would like to take this time to consider the teachers, institutions and societies views in this. For teachers a robotic teacher could become quite damaging. It would make the job market for teachers really competitive and a lot of teachers could end up loosing their jobs. We should also consider if robotics would actually help students achieve better grades and motivate them. A human being could have more motivational impact on a student than a robot could ever have due to the reason of having no emotional or psychological impact on any kind of subject. A study based on teachers’ perception of using robotics by Ahmad Khanlari [6] ,shows that teachers do not believe that robotics has a positive effect on students’ academic achievements.

Even if robotics has a good impact on economical value for institutions, they could also be quite expensive to run. Based on current energy prices.

For society robotic teachers could have more of a negative impact than we think. The absence of supervision and guidance could have long term damage. Further removing another social opportunity for students could be quite damaging. We can already see how mobile phones affect our daily live and more and more children have problems in a social environment, which leads to more damage to society. A study by Hilary Groarke on the impact of smartphones on social behaviour and relationships [7], shows how more usage of smartphones has a strong coloration to higher use of technology rather than engaging with co-present others.

To summarize I do believe that robotic teacher can have a good impact for students as well as the institute, but there can also be a lot of bad by products. I believe we need to be really cautious on how we approach this and how we implement it.

# Self Assessment

## Programming assignment

### Completing scenario

#### Assessment

I personally think my robot is achieving this part of the assignment really well. It is really rare that it fails. It should be really adaptable to a lot of environments which have a base line to the assignment world. The way I dealt with the pillar has been done really well and should work with higher and wider pillars too. There might be a possibility if the pillar is too wide, I get to close and I won’t be setting the width properly. As mentioned in the report earlier on I use margins for deciding when to exit the inMaze logic. This could become problematic if the distance is too small or if the distances in the maze become too big. Further if the bins height gets changed too much this could result in me hitting the bin. This could become problematic for the actual robot and damage it. I do think otherwise I am completing the task under the assignment’s conditions really well. Thus, I would mark myself [29/30].

#### Feedback I would like

* General coding practice and how I approached it
* How I deal with my arm movements and making them consistent if that is good or not
* The cylinder calculation
* General strategy for each part

### Screen cast

#### Assessment

I do personally think the screencast explains very short and briefly how each tasks work. I do not go into too much detail, but it should explain what is happening to someone that is not familiar with coding how I am approaching each task. Maybe it is a bit too short and I could have stopped the video and explained a bit more in depth. Could have also spoken a bit more clearly and did stutter here and there. I would mark myself [8/10]

#### Feedback I would like

* Anything I could improve on generally speaking

### ROS usage

#### Assessment

I do think I use quite a lot of ROS techniques with srv and msg. I could have used global parameters and maybe actions for certain things. But it works really well. My usage of using srv and msg means I have got a modular program that can be changed by adding and removing modules and keeps reuse of code to as much of a minimum as possible, without creating too many topics and services. Because too many could become a problem. I would mark myself [7/10]

#### Feedback I would like

* How I could have used actions and global parameters and for what

## Report

### Strategy

#### Assessment

I think my report is really well structured. I added references where needed. I explained everything in detail and thus just by reading this report it should be quite clear on what the program is doing and how I went to solve the task at hand. My wording could need improvement and maybe how to explain certain things a bit more clearly and better. I would mark myself [18/20]

#### Feedback I would like

* Maybe a few hints on how to improve writing
* Feedback on overall structure

### Ethics

#### Assessment

The article is written well. I go into different scenarios and try to look at it from different point of views and also try to take some problems into consideration. I do add a few references to literature, which help me build a case. Maybe I could have been better on the overall flow of the article and added a few more positives for it. Furthermore, I do think more reference could have been added to build upon my arguments. I would mark myself [14/20]

#### Feedback I would like

* Overall flow of the article
* Did I forgot to consider certain things?

## Overall marking

Here I will list my overall mark

|  |  |
| --- | --- |
| Topic | Mark |
| Completing scenario | 29 |
| Screen cast | 8 |
| ROS usage | 7 |
| Strategy | 18 |
| Ethics | 14 |
| Overall | 76/100 |