Implementation of a Task-motivated Controller, for a Mobile Robot

Adam Hubble (P17175774)

Abstract – Fundamental to the existence of mobile robots, autonomy is renowned for its facilitation of tasking; where mobile robots are typically assigned series of complex, predefined tasks to complete, without cause of human intervention. As the description of this document reveals, the configuration of a task-motivated robot controller will be explored, for the series of tasks that it is purposed for; said controllers' suitability for the tasks to be described, is detailed in scope of its architectural, behavioural, and experimental designs, and is reinforced by the relevant testing features.

Keywords – mobile robots, autonomy, predefined tasks, human intervention, task-motivated, robot controller, architectural, behavioural, experimental, testing features.

I. Introduction

For the layout of the structured environment provided and for the continued support of the Pioneer P3-DX mobile robot [1], it was expected of the robot controller proposed, to expedite the robot's completion of a series of complex, predefined tasks, which partake in a linearised progression. These tasks were as listed:

- *Locate the centre of the inner-most room*
- *Exit the inner-most room without collision*
- *Locate and station at the beacon object*
- *Chart the robots subjected environment*
- *Return to the inner-most rooms centre*

For all the tasks specified, it was anticipated for the controller to feature a random wandering, edge following, avoiding, exploration and mapping strategy, each purposed for the robot's navigation, exploration, and obstacle avoidance aptitudes; when roaming, target finding and whilst charting the environment. Which as an independent strategy, environment charting was purposed for graphically representing the intricacies of the robots subjected environment, for which, is assumed unknown by the robot; thereby, obstacles situated in and around the robot in the environment, as well as its bordering structures, can be understood and visualised from the sensory perspective of the robot. This is renowned as Simultaneous Localization and Mapping (SLAM) [2], which incorporates the robot charting an unknown space, relative to its existence and currency of its location in said space; this is achieved via the combination of "sensor signal processing", which regulates the sensory data inferred by the robot, and "pose-graph optimization" (PGO) techniques, which are used to validate and represent sensory data, through "sampling the robot's trajectory" [3].

Available in Appendix A

For the behavioural strategies mentioned, it is notable that most of said strategies were designed to function with the distance metrics returned by the array of "range-finding Sound Navigation and Ranging (SONAR) sensors", that the robot employs [4] as its primary sensory capability. Whereas a vision-based sensor is also equipped by the robot, as its secondary sensory capability, which is entirely purposed for appropriating its "method of navigation" [5], when target finding behaviours are invocated. This is achieved through the interpretation and manipulation of image colour buffer data [6], which promotes the visualisation, approach, and stationing of the robot, at the designated beacon objects location within the environment.

Available in Appendix B

A. Software development

As the software application operated for fulfilling a robot controller that encompasses the behavioural strategies abovesaid, CoppeliaSim [8] was reclaimed for its ongoing support for robot controller development, as realised for both controller submissions prior. Moreover, in surplus of the software applications involvement in the 'Mobile Robotics' undergraduate module and for its contemporary customs for the module in focus, remote application programming interface(s) (API's) were not applicated once more, given their absence in prior controller developments and their resultant foreignness in native programming language and featured user interfaces (UI's). Hence both task-motivated and unmotivated behavioural strategies of the robot controller proposed, were developed using CoppeliaSim (Lua), as the prominent integrated development environment (IDE) for the controller's development; whereas a secondary IDE, Microsoft Visual Studio (C++) [9], was purposed for compiling the graphical representations of two offline map variants, which feature Random Sample Consensus (RANSAC) [10], to defer from the visualisation of plot points, strictly (linearised line series alternatively).

In continued mention of environment charting, Microsoft Excel [11] was also adapted for the purpose of writing environment charting data (including RANSAC derived data), to external comma-separated values (CSV) files, and reading said data from said files, for populating a series of offline map variants within separate spreadsheet documents. Where each of the variants were purposed for representing the linear sub-processes of the data validation technique, that is adopted for filtering the outlying detections of the robot, which thereby legitimises the remaining positions of obstacles and structures in the environment, that were detected throughout the course of simulation runtime.

B. Pioneer P3-DX composition

Given the known mechanical composition and sensory capabilities of the Pioneer P3-DX mobile robot, as declared in prior subsections and in previously submitted robot controller developments, it is not necessary to detail its composition once more.

However, it should be acknowledged that the sensory inferences of the robot, both the transmission, reception, and transducing of ultrasound waves, as well as the generation, interpretation, and manipulation of image colour buffer data, can be utilised for actuating the robots featured, two-wheel two-motor differential drive [1], for the ultimate purpose of task completion. Where said inference techniques can be applied to address either motivated or unmotivated natures of the robots tasking, which are dependent on the concepts derived from odometry, a dead-reckoning sub-domain [12], that concerns the measurement of the robots "steering orientation" and "relative positioning" [13] overtime, using odometric sensors, to handle the extent at which the robot executes each actuation pattern.

Thereby, the techniques abovesaid purpose to condition the invocation of wheel displacement and the proportionality of each wheel's motor velocity, to enable the robot to adhere to the demands of the active task(s); thus, in continued appliance of the subsumption architecture [14] previously submitted, behavioural states possess the means to subsume one another through the application of finite-state machines (FSM's), where unmotivated tasks assume higher precedence for the priority of obstacle evasion, which enables the robot to autonomously appropriate its behavioural invocation and resultant movement patterns, to near motivated task completion, whilst maintaining a reactive state of awareness to sensed stimuli (unmotivated), interchangeably. This was conceptually derived from Robin Murphy's "hybrid deliberative/ reactive paradigm" [15].

Available in Appendix C

II. Architectural Design

Given the application of the behavioural subsumption architecture, revealed for the robot controller submitted, each behaviour of both motivated and unmotivated tasking qualities, is represented by a succinct series of Boolean variables, which represent both the major and subsidiary states, or "modules" [17] of the controller. Said variables exist to enable a "basic control system to be established", when complemented by a series of 'if-else' statements, that are used to "disable specific behaviours where their activity at a particular time or circumstance is undesirable"; thus, allowing behavioural states to subsume one another, where behavioural functionality can be regulated and therefore exhibited by the robot, autonomously. It is notable that each of the behaviours implemented, represents a "level of competence" for the controller, as Rodney Brooks would describe.

Collectively, these syntactical elements of the controller formulate the FSM's mentioned prior, which are used to "model the robots' behaviours as a series of states, transitions and associated actions" [18]; through the alternation of each behaviours binomial state (Boolean), enables any but only one behaviour of the robot, to be transitioned to and exhibited at any given period. Thereby for the controller's architecture submitted, it is proposed for the robot to achieve and actuate said tasks in isolation, where it is accepted as a single-tasking application. This technique is recognised as behavioural inhibition [17], which provides bettered error-handling whilst dictating less computational expense, as opposed to its counterpart: behavioural suppression, given its unilaterality.

Available in Appendix D

A. Locating the inner-most rooms centre

From an architecturally-driven standpoint, locating the centre of the environments inner-most room required a medium for the robot to sense its containing structure; this was appropriated by applicating the full-range of the robot's primary sensory capability: sixteen ultrasonic sensors, which collectively function to provide an omnidirectional spatial awareness for the robot, and resultant ability to factionalize its surrounding space, for determining a centre point. Where said centre point is inevitably the objective of the robots preliminary tasking.

When applicating the range of ultrasonic sensors detailed, it was known that gradual displacements within the robot's wheel motor velocities could be issued, proportional to the distances, or error, at which each regional pair of its sensors detects the rooms structural arrangement and transduces to populate appliable, distance-oriented data. In use of said data, the error between the robot and all sides of the room could be nullified, to an equilibrium of all sensor detections, indicating the centre. In continued discussion of the regional pairing of sensors, the sensors returning the closest detection to an object for each region of sensors: front-most (sensors '4' and '5'), right-most (sensors '8' and '9'), back-most (sensors '12' and '13') and left-most (sensors '16' and '1'), were referred to for determining the invocation of wheel motor velocity displacements, that would eventually navigate the robot to establishing the centre point of the space. Whereby, it was observed only relevant to applicate the sensors yielding the closest detections, for allowing the robot to continually evade regions of the surrounding structure that were situated closer by, until equilibrium could be achieved; thus, being less computationally expensive compared to multifaceted calculations.

Available in Appendix E

As the initial and abandoned strategy trialled for such tasking, a multitude of proportional-derivative-integral (PID) feedback controllers [19] were applicated, for the robot's front-most, right-most, left-most, and back-most sensor region pairs, for enabling the robot to appropriate its centering behaviours, relative to "minimising the variability of calculated control error" [20]. A PID controller was allocated to the robot for each of the four subdivisions figured for the subjected space, representing each perpendicular arrangement of the spaces structure (corners); however, as a behaviourally trialled and ineffectual method, which resorted in the robot spiralling, continually, the appliances of PID controllers were neglected and a further manual, experimental-focused design was elected. This design alteration was supported by the elected methods behavioural desirability and consistency in the task's completion, regardless of the robots starting point in the space; this is believed to be the result of increased control of behavioural tuning passed.

Available in Appendix F

B. Exit the inner-most room

Featured as the secondary objective of the robots tasking, exiting the inner-most room of the subjected environment also anticipated the application of the robot's ultrasonic sensors, for both appropriating the robots heading alignment, to face the exit (doorway) and the robot's resultant rotary adjustments, required for exiting the space in a linear motion, without the apparency of collision occurrences with the spaces structural arrangement; this concerned the wall objects, which constitute to the rooms doorway, or exit.

Dissimilar to the requirement of the robot's ultrasonic sensors, when establishing the centre point of the inner-most room, the exiting procedures of the robots tasking, only demand the front-most facing sensors, ranging from the front-left most (sensor '1') and front-right most (sensor '8') indexes, given the robots forward traversal pattern, required to near and overcome the exit. Whereby, the front-most facing sensors of the robot are regionally sectioned by their relativity to the robots left and right sides, as well as their angular offsets from the robots heading orientation; this is obtainable by the robots supporting gyroscopic hardware [1]. Partitioning the sensors in said way is justified for the appropriation of wheel motor velocity displacements, that promote angular traversal patterns in the robots exhibited behaviours, to marginalise the error between the robots facing direction and the boundaries of the exit; thereby enabling the robot to exit the space with a linear motion pattern, and thus, nullifying the collision occurrence potential.

Available in Appendix G

Meanwhile, for the alignment behaviours of the robot, it was figured to retarget all sixteen ultrasonic sensors for determining the aligning direction of the robot, omnidirectionally; the direction in which would enable the robot to establish an agreeing relation with the exit, within a shorter duration, comparatively. For which, the sensors were segregated by their relativity to the robots left and right sides once more, ranging from the front-left-most sensor (sensor '1') and back-left-most sensor (sensor '16') indexes, to acknowledge the side(s) at which the structural arrangement of the room situates closest to the robot; this metric could then be applied to determining the robots aligning direction, which poses a negated relation to the side(s) of the structural arrangement, which are cumulatively detected and therefore known to reside closest to the robot. In result of this determination, wheel motor velocity displacements could then be invocated to pivot the robot around its own axes, for sensing and appropriating an alignment with the rooms exit. To note, when handling identical detections, numerical randomisation is defaulted to, for resolving the dispute in the robots aligning direction.

Available in Appendix H

C. Locate and station at the beacon

Instructed as the tertiary motivated tasking for the robot, it was expected for the robot to explore its subjected environment, whilst avoiding obstacles that obstruct its capabilities within establishing, navigating towards, and stationing at, the designated location of an object that identifies to be the beacon, within the radius of '0.5' metres. Given the inability of ultrasound transmission and reception, in the determination of a detected objects characteristics and its resultant non-distinguishment of said objects, the robot's primary sensory capability could not be applicated for the tasking, explicitly dedicated to the beacon, abovesaid.

Thereby, a singular visual sensor was elected for the tasking in mention, given its capabilities of "extracting visual features for positioning" [21] the location of objects, which translates to the beacon's location being positionally known, amongst the robots subjected environment; as previously declared, said sensor became the secondary sensory capability of the robot, in result of this taskings requirements.

Uniquely, the sensor is configured to be explicitly handled, for the purpose of permitting the generation of and access to, each rendered image's colour buffer data, that is sampled by the robot. For which, it can utilise the auxiliary values [22] of said sampling, being the multitude of minimum, maximum, and average colour intensity data, alongside the depth buffer values, attained through depth testing [23] targeted colour intensity variants, in each of the images rendered by the sensor. That are then applicated, as emulated distance metrics, at which an object of a targeted colour resides from the sensor (average intensity value). Whereby, for the robots tasking, the beacon objects renderable material property, 'adjust colour', is assigned the maximum green intensity value '1', from the RGB colour model [24], for which the sensor is configured to sample the average intensity value of, for enabling the robot to identify and pursue the beacon.

Fundamentally, the robot possesses the capacity of acknowledging the beacons location and detected state, for which its wheel motor velocities can be displaced overtime in use of said data, to autonomously navigate itself towards and eventually station at the location of the beacon, within the radius instructed for its tasking (within '0.5' metres). This is functionally achieved by the enablement of the robot progressively pivoting around each of its wheel's positions, whilst maintaining the beacon object in the renderable boundaries of the sensor's perspective frustum, which occurs up until the distance threshold is surpassed. For the boundaries mentioned, the robot alternates said pivoting actuation patterns that it invocates, at the time of which the beacon is visualised at an offsetting alignment, relative to the facing direction of the robot; thereby said boundaries can be referred to as the robot's points of oscillation, or adjustment. Undoubtably, this implementation factors the visual sensors resolution ('512 x 512'), field-of-view (FOV of the sensors perspective frustum) ('80' degrees), rendering mode (OpenGL) and near and far clipping plane range ('0.0001' to '10' metres), that all are configured for said enablement of the robot and its tasking, to be consistently accomplished.

Available in Appendix I

Moreover, for addressing the robot's exploration strategy proposed for enabling the visual sensors' acknowledgment, of the beacon's presence in the environment. The robot invocates the following behavioural strategies, interchangeably:

- Random wandering
- Edge following

Given the applications of said behavioural strategies in the controllers previously developed, submitted, and purposed for the Pioneer P3-DX mobile robot, their existing functional implementations were also figured to be well-adapted for the environment exploration tasking of the robot, presented. This was in knowing of a series of "continuous movements" [25], the random wandering behaviours, bettering the robot's "target searching" capability in relation to locating the beacon, given the exploration potential that increased variance in movement patterns yield, exponentially. Whereas, edge following behaviours as recognised navigation candidates, within “navigation systems” [26], provide the supporting functionality used to facilitate the robot’s navigation, around structural arrangements in its subjected environment. Which inevitably enhances the visual sensors detection potential if the beacon is positioned nearby other objects, with edges that the robot detects as being followable; this is assumed by the resultant panning motions that are exhibited by the robot, some of which even cyclical, that provide vast coverages of the robots surrounding proximity.

In this relation, given the “transient and steady-state responses” [27] to error that each of the PID “feedback control systems” [26] provide (each side of the robot), for the robot’s edge following strategy, it was imperative for its prior implementation to be refactored, for appropriating the presence and trackability of cylindrical-based objects, which were not catered for previously. Whereby, when not managed, the resultant behaviours exhibited by the robot, would resemble the continuity of a parallel relation with the sides of said objects, that exist in its environment. This was addressed by invocating an avoidance strategy, evasive sequence, that defers the robot from the object being followed, beyond one revolution around the object being actuated by the robot, which integrates heading computations of the robot to determine such and allows for the surrounding proximity of the robot to be surveyed in advance of evasive behaviours being invocated. Aside from this implementation, additional functionality was not required and nor was a refurbishment for the random wandering strategy, purposed for the environment’s exploration. Notably, both avoidance and edge following behavioural strategies, applicate the robot’s array of ultrasonic sensors, to achieve their functional purposes.

Available in Appendix J

Hereby, in continuation of the avoidance strategy mentioned and proposed for the obstacle evasion requirement of the tasking in focus, a series of subsidiary states were compiled, that collectively comprise said strategy, for enabling the robot to self-manage problematic scenarios that it could, as an autonomous mobile robot (AMR), expose itself to. These subsidiary behavioural strategies exist as:

- Valentino Braitenberg avoidance [28]
- Reversing
- Stuck

For each of the avoidance states listed above, provides the robot with separable functionality for nullifying the frequency of collision occurrences, if any, in a range of possible and probable scenarios encouraged by the randomised nature of the robots wandering capability, which aims to maximise the robot’s exploration potential (close relations to objects), for environment charting purposes.

As the strategy’s fundamental behaviour, customary avoidance provided by Valentino Braitenberg’s avoidance algorithm, is applicated using the robots front-most facing SONAR sensors (sensors ‘1’ to ‘8’), where each’s “output directly affects the movement” [28] of the robot, through proportionally displacing each wheel’s motor velocity. This technique establishes “obstacle avoidance between other static objects in the environment, such as: walls” and is achieved by applicating coefficients to each of the robots detecting sensors, that are proportional to the angular offsets of detected objects, accrued from the facing direction of the robot. Where objects detected ahead of the robot (sensors ‘4’ and ‘5’) stimulates a “faster avoidance”, when compared to objects detected by the robot’s side-most facing sensors; this transpires the urgency of frontal avoidance, to preserve the continuity of the robots tasking. For the behavioural domains of the algorithm, the robot only considers its front-most range of sensors given its forward mode of traversal, which is prevalent across most behavioural strategies featured by the controller. Whereby, in use of said sensors, the robot can appropriate leftward, rightward, and forward modes of traversal, reactively, to evade detected objects. For which the robots front-left sensors (sensors ‘1’ to ‘4’) and front-right sensors (sensors ‘5’ and ‘8’) are parted into logical regions, as previously determined, for proportioning the velocity displacements issued to each wheel’s motor, determining the resultant mode of traversal actuated by the robot, that is directionally calculated optimal, for obstacle evasion.

Available in Appendix K

Less significantly, the strategies reversing behaviour facilitates the robot’s ability to appropriate backward traversal motions, in correspondence to objects being detected at equivalent distances (within ‘0.005’ metres), using any two-symmetrically-opposed front-facing SONAR sensors (sensors ‘1’ to ‘8’). Said behaviour exists to resolve disputes in evasive direction, that the Braitenberg avoidance algorithm cannot determine, without the invocation and exhibition of resultant oscillatory motions, which within a behavioural respect, is undesirable and further hinders the robot’s manoeuvrability within restricted spaces of a given environment. Regarding the negated displacement of the robot’s wheel motor velocities, used to achieve backward traversal, said velocities are actuated by the robot for a randomised duration, unless all sixteen ultrasonic sensors (omnidirectionally surveyed) determine when an object is no longer detected in its robot’s frontal face, or inversely, when object(s) are suddenly detected within its rearward face. Whereby, the controller then proceeds to invocate the turning subsidiary state; the fundamental behaviour of the stuck subsidiary state of the avoidance strategy, to avert said behaviours from immediately recurring.

Available in Appendix L

In extension of the subsidiary behavioural state mentioned, stuck, as the final avoidance strategy behaviour for the robots tasking, the robot is functionality provided with the capability of exploring narrow spaces amongst its subjected environment, whilst evading potential incidents of collision, with the structural arrangements of said spaces, for the preliminary purposes of attaining collision-free navigation and environment charting maximisation. These behaviours of the robot are supplemented by its aptitude to appropriate a departure space, which is instrumented by the reuse of all sixteen ultrasonic sensors onboard the robot. Whereby, like the implementation described for the room exiting task of the robot, the robot can cumulatively determine an alignment direction, relative to the proximity of each side of the spaces structure, to the robot’s sensors; this can also be arbitrarily settled, to resolve cumulatively equal detections for finding a given spaces exit, nonetheless.

As the robot’s realisation mechanic for recognising the exit point of a given space of entrapment, a non-detection status returned by the robot’s front-most facing sensors (sensors ‘4’ and ‘5’), is relied upon for invocating forward traversal patterns; these are actuated by the robot to enable it to emerge from said spaces of entrapment. As previously detailed, said behaviours of the state can be invocated upon the robot transitioning from the reversing subsidiary state, however, it is principally configured for when the robot detects objects using six of more of its available sensors, simultaneously. This infers that the robots entire frontal face can be obstructed by the presence of objects, which demands the invocation of motor velocity displacements, to enable the robot to actuate a pivoting motion around its own axes, in attempt of re-establishing the spaces entry point, that the robot can repurpose as its point of exit.

Available in Appendix M

D. Environment Charting

As the robots fourth recognised tasking, it was further expected of the robot to chart its subjected environment, through applicating its primary sensory capability yet again, which is fundamental to the acquisition of “a global overview map that integrates all of the data collected by the robot” [30]. Given the nature of the robots tasking, a visual representation of the “robots’ environment from a top-view perspective” offline, and online in “real-time” is compiled, using said sensory data of the robot, which requires to be globally translated from the local coordinate space that each point-of-detection originates; this encompasses the localization process of SLAM, as mentioned within the preliminary sections of this document.

Available in Appendix N

Whereby, both offline and online map variants of the robot’s environment are populated, for live demonstration purposes, as well as for passing charting data through PGO techniques [2], that then enables offline map variants to become increasingly more accurate and proficient in representing the structural arrangements and positions of obstacles, surrounding the robot in its subjected environment. Both methods are supported by the enablement of the robot “knowing its position at all times relative to the environment” [32], which is accomplished via surveying the robots front-most facing sensor detections (sensors ‘1’ to ‘8’), given the robots dominant traversal pattern, forwards, as discussed previously. In which, the robot invocates the random wandering, avoidance and edge following strategies, for the robot’s exploration, avoiding and object edge-tracking behaviours, that are each obligatory to the environment charting task and for adhering to the robot’s behaviourally unmotivated taskings, intermediately.

E. Return to the inner-most rooms centre

Advancing from both the environment charting and beacon-orientated taskings of the robot, the robot is then instructed to navigate and return to the centre-point of the environments inner-most room, which is discovered and recorded preceding the robots exit-alignment process, which comprises the robots preliminary tasking. Unlike all other implementations compiled for combating the tasks presented to the robot, for the task in focus, the robot does not require the application of its sensory capabilities, primary nor secondary, for supervising its navigation, directly. Whereby, the robot alternatively appropriates the appliance of odometry [12], as integrated within the controllers random wandering strategy, to continually regulate its heading orientation in attempt to maintain a facing direction with the centre-point of the given room; this is irrespective of the robots' position in the environment and is simply calculated as the angular offset between the robots and centre-points position vectors, in custom of the arctangent operation [33].

For this purpose, a subsidiary behavioural state, explore, was integrated into the controllers random wandering strategy, which enables the strategies arbitrarily determined angular traversal behaviours, to be subsumed by premeditated angular traversal behaviours; these are used to regulate the robots heading orientation, odometrically, through the previously disclosed wheel velocity displacements, that can be issued to the robot's differential-motor components [1]. Given the architecture revealed for the task's enablement, the robot remains within the capacity of being able to reactively invocate other unmotivated taskings, for their obstacle avoidance and navigation purposes, which when exhibited, synthesizes the robot to maintain a continuity of its tasking.

III. Behavioural Design

A. Locate the inner-most rooms centre

Behaviourally, implementing the calibration behaviours of the robot for establishing the centre-point of the environments inner-most room, targeted the full-range of SONAR sensors available to the robot, for harnessing an omnidirectional spatial awareness, as previously mentioned. This purposed for granulating the robot's traversal pattern as a recurring series of forward and oscillatory motions, which the robot in sequence, utilises to navigate to a middle, or equilibrium. As previously revealed, the robot utilises a sensor pairing scheme for each of the robot's regional faces, where the sensor with the smallest detection of either region, is applied as the evasive manoeuvre determinant of the robot; this is relative to the distances that each wall object constituting to the structural arrangement of the room, resides at. Inevitably, these objects situate closer to or further from each of the robot's sensor pairs, overtime, thus enabling the robot to react proportionally to its detections.

For the invocation of the navigation and traversal patterns mentioned, a series of functional conditioning is used, this can be simplified by the given pseudo notation:

If difference between *side-most sensors is significant*
and *front-most sensor detects an object close by* **then**

If *left and right-most sensor detections are similar to their previous detections* **then**

If *objects are detected closer to the robots left side* **then**

Robot turns right

Elseif *objects are detected closer to the robot's right side* **then**

Robot turns left

Else *prevent the robot from statically oscillating*

Robot oscillates with a forward mode of traversal

Elseif *no object detected on front or back sensors when side sensors aligned*
or *side-most sensors aligned but front and back sensors are not*
or *front and back sensors are not aligned* **then**

Robot traverses forward

Elseif *all sensors detect a wall* **then**

Robot remains stationary
Robots position is stored as the rooms centre-point
Robot identifies readiness for exit-alignment

In correspondence to the pseudo notation, numerous constant and randomly determined displacements of velocity are issued to the robot's wheel motor components, for enabling the robot to actuate turning motions, which can be pivot-focused or modelled as angular traversal modes, as well as forward traversal motions, which can be momentary or continuous. Collectively, each of the movement patterns abovesaid constitute to the robot's establishment of the rooms centre-point, at which the stationary idle pattern is invocated, to avert further traversal-based movements from being actuated by the robot; this prepares the robot for the rooms exit-alignment.

For handling the exit space of the room in mention, given that no object resides within said space reserved as the rooms exit, or doorway, the robot in relation to the pseudo notation presented, would not invocate turning motions and would resultingly near the walls bordering it, until an eventual collision occurrence; this was apparent as the robot could not maintain a constant detection with one of two adjacent faces of the room's walls, that occupy its exit space. Given the robots oscillatory motions that are predominantly actuated for achieving the task in focus, a side-most sensor adjacent to the rooms exit, would alternate between returning a positive and negative detection status, for every other frame that is passed, which is relative to the rotary adjustment actuated by the robot; thus, a persistent detection could not be attained for either of its side-most sensors. Thereby, to overcome this behavioural deficiency, the robot's side-most sensors when not detecting the structural arrangement of the room, are passed a distance metric that is half as great as its opposing sensor, which assumes a detection; this enables the robot to turn into the centre of the room space, upon establishing the exit, before then detecting a face of an oncoming wall. Providing the less significant value is passed to the non-detecting sensor, the robot will acknowledge that object(s) are positioned closer-by for the corresponding side, in which the controller will invocate evasive manoeuvres as realised by the pseudo notation, to enable the robot to avoid incidences of collision.

Available in Appendix O

Supplementary to the scenario put forth, upon the robot's frontal face establishing an opposite or adjacent relation to the perpendicular arrangements of the rooms structure (comers), the robot would also result in inevitable collision with the walls constituting to the rooms arrangements described. Whereby, similar to the scenario populated previously, the apparency of collision was also resultant from the oscillatory motions actuated by the robot, in which derive from the robots heading orientation being directed at the cornering boundaries of the room; this promoted the robots side-most sensors to establish equality in their detections through the controllers invocation of continual, alternating turning motions, at which the robot actuates forward traversal patterns in the eventuality that said detections are relatively equal (difference less than or equal to '0.2' metres).

Available in Appendix P

Deviating from the specification of the scenario given, upon a side-most sensor not detecting one of the rooms borders (walls) but its opposite achieving such, whilst the robot's front-most facing sensor also detects a wall ahead (cornering), the robot would similarly collide with the bordering walls, in result of the robot actuating angular traversal patterns, as opposed to being pivot-focused. Thereby, for the pseudo notation and each of the scenarios presented, the robot possessed no means for evading collision and achieving its tasking. Consequently, to combat the scenarios submitted to the robots tasking, the side-most sensor not detecting the rooms structural arrangement, could again be assigned a distance metric, that is instead twice as large as the detection returned by its opposing sensor; this instructs the robot to turn into the centre of the room once more, for when the robot recognises its presence within a cornered space, using the detection from its front-most sensor (when within '0.5' metres). Provided the more significant value is assigned to the robots non-detecting side-most sensor, the robot will evade the corner spaces and sides of the room's structural arrangement, positioned oppositely and adjacently to it, given their closer, emulated dispositions to the robot, as it acknowledges.

Available in Appendix Q

This configuration enables the robot to accomplish said task, regardless of where it is initially positioned in the room; upon the robot recognising its central alignment in the room, its current position is stored as the middle-most coordinates obtained for the room, where the robot then transitions to its next tasking, via alternating the binomial value of the tasks completion state (Boolean).

B. Exit the inner-most room

For implementing the behaviours concerned for the robot when exiting the inner-most room of its subjected environment, an omnidirectional spatial awareness was also required, initially, in determination of its alignment direction to the rooms exit space; this inevitably required all the SONAR sensors onboard the robot, as previously detailed, to fulfil. Meanwhile, purposed only for the exiting procedure of the robots tasking, the robot surveys its front-most facing sensors, given its forward mode of traversal that is expected for the enablement of the robot exiting the room, in the motion-linear format instructed for its tasking.

In continuation of the robot's alignment process and in use of all the robot's sensors, an alignment direction can be determined for conditioning the robots wheel motor velocity displacements, that enable it to actuate pivot-focused motions around its axes, in attempt to establish the exit space of the room, which is determined by the eventuality of the robot's front-most facing sensors returning non-detection statuses ('0'); this infers that no object, or wall, resides ahead of the robot, signalling the robot to then actuate forward and angular traversal patterns, for approaching and overcoming the rooms exit. For this tasking, the robot's sensors are sectioned by their relative position to each side of the robot, thereby formulating a left and right partition of sensors that are applicated in the determination of its alignment direction, that is elected from its bidirectional capacity. The process in which the robot's alignment process is functionally determined, derives from the following pseudo notation:

If *objects are detected closer to the robots left side* **then**

Robot aligns left with the inner-most rooms exit

Elseif *objects are detected closer to the robots right side* **then**

Robot aligns right with the inner-most rooms exit

Else *objects are detected at equal distances to either side of the robot*

Randomise an aligning direction for the robot

If *the robot's alignment direction is recognised as left* **then**

Robot aligns left with the inner-most rooms exit

Else *the robot's alignment direction is recognised as right*

Robot aligns right with the inner-most rooms exit

Upon the alignment direction and tasking being settled by the robot, forward and angular traversal patterns are actuated, as abovesaid. In which, through the application of the robot's front-most facing sensors, enables the robot to further adjust its heading orientation for positioning the robot central, in relation to the walls constituting to the room's doorway. Whereby, if the robot is offset for the centre of rooms exit, angular traversal patterns are invocated by the robot for realignment purposes; this simplifies a collision avoidance strategy for evading the walls bespoken, when approaching the exit of the room. Otherwise, the robot is then capable of exiting the room in a motion-linear traversal pattern. For the offset in the robot's alignment discussed, this can be determined by the returned differences between the sensor's detections, that are regionally sectioned and accumulated as left and right partitions, similarly to its prior sub-tasking. The robots room exit strategy can be further understood by the accompanying pseudo notation:

If *the absolute difference between the robots accumulated sides of sensor detections is significant* **then**

If *the robot detects the rooms walls further away on its left side* **then**

Robot turns left

Elseif the robot detects the rooms walls further away on its right side **then**

Robot turns right

Else the absolute difference between the robots accumulated sides of sensor detections is insignificant

If *the robots front-most facing sensors detect an object close-by*
and *the robots front-most facing sensors do not detect an object* **then**

Robot traverses' forwards

Elseif *the robots front-most facing sensors detect an object close-by*
or *the robot's side-most sensors detect the walls constituting the doorway*
then

Robot identifies the inner-most room as being exited
Robot identifies the beacon object as being searchable

As unprecedented by the pseudo notation provided, the robot also presents the capability of exiting the inner-most room, irrespective of an obstacle being positioned in or around the proximity of the rooms exit space. For which, the robot achieves by exiting the taskings behavioural strategy, upon an object being detected close-by (within '0.3' metres) using any of its front-most facing sensors, which the walls constituting to the room's doorway, cannot determine; this is appropriated by the priority of the robot's realignment behaviours, that attempt to invocate wheel velocity displacements to the robot's motor components, for maintaining a central relation with said doorway up until the room is exited. Thereby, to then determine the robots exit-alignment, given that the robots front-most facing sensors would no longer simultaneously return non-detection statuses ('0'), two Boolean variables are configured, where for either sensor that returns a non-detection at any period whilst the robot is actuating pivot-focused motions, its binomial state is alternated. Upon both states being alternated, the robot can then identify its exit-alignment process as complete, where the resultant heading orientation of the robot situates between the obstructing object and corresponding wall, that collectively border the robots exit space; this method sensibly prevents the robot from continually pivoting around its own axes whilst it senses for its exit-alignment. Therefore, the tasks behavioural design can be considered well-adapted, for quashing problematic scenarios that the robot could be subjected to, both naturally and forcibly.

Available in Appendix R

C. Locate and station at the beacon

As the premise of robot's capability in locating, navigating towards, and stationing at the beacon object designated to its tasking, the robot equips and utilises a visual sensor, which through the computational competences of colour and depth buffer data extraction from computerized images, the robot can support a means for accomplishing the task in focus. As the robot's mobility and "target-searching" [25] capability, the robot as previously revealed, invocates its unmotivated tasking behavioural strategies, that are: random wandering, edge following and object avoidance (discussed later), for their fundamental purposes of exploration, navigation and evasion; generally, each of said strategies enables the robot to establish a surveillance of the beacon object (random wandering), avoid obstacles in and around the proximity of the beacon object in the environment (object avoidance), as well as enhancing the prospect of surveying the beacon object (edge following).

In correspondence to the behaviours specific to the robot and beacon objects tasking, images rendered by the visual sensor can be interpreted and extrapolated for their auxiliary values (when explicitly handled), for which the robot purposes the average 'green' intensity value of, to appropriate and invocate oscillatory motions of the robot, that enables it to near and recognise its distance displacement to the beacon object, overtime; this was configured in parallel with the material colour applied to the beacon object, as was realised in the strategies architectural design. When sampling the given average intensity value of the sensors generated images, a distance metric is emulated, where the closer the beacon situates to the visual sensor, the higher the average intensity value becomes; this is given that the objects profile appears larger and thereby occupies more dimensions (pixels) of said images, when rendered in the following render pass [34]. This value is applicated for signalling when the robot is within the instructed range of the beacon object (within '0.5' metres), where it would then actuate stationing behaviours for a specified period ('20' seconds). For understanding the values application for the tasking said, refer to its pseudo notation:

If *the beacon object has not been seen by the robot*
or *the robots front-most facing sensors detect an object close-by*
and *the beacon object is seen far away from the position of the robot* **then**

If *the beacon object is seen far away from the position of the robot*
and *the robots front-most facing sensors detect an object close-by*
and *the robot acknowledges surveillance of the beacon object* **then**

Robot enters avoidance, evasive manoeuvre

Elseif *the beacon is seen closely ahead of the robot*
and *the robots front-most facing sensors detect an object close-by* **then**

Robot identifies as being stationed at the beacon object

Elseif the beacon is seen by the robot **then**

If *the robots initial aligning direction of the beacon has not been set* **then**

Robot enters avoidance, evasive manoeuvre

Robot identifies the beacon object as been seen

Else in the case of miscellaneous errors occurring with the robots sensors

Default the values of the tasks behavioural variables

For settling the robot within the proximity instructed for its stationing task, alongside the visual sensor, the robot applicates its front-most facing ultrasonic sensors additionally, for identifying the distance at which the beacon object or alternatively, other objects (obstacles), reside at relative to the current position of the robot. In compliance with the notation above, the robot is then also able to appropriate evasive behavioural invocation when objects other than the beacon, obstruct the robot's pathing to it; this is executed as part of the avoidance strategy configured, which is purposed for the robot to avoid collision incidents with obstructing objects and re-establish beacon surveillance from a better-suited location. For which the visual sensor cannot acknowledge nor appropriate, as simply.

Upon the robot establishing beacon surveillance, the robot then functions from the use of a series of sampling filters [35], that are each configured to identify the location of the beacon, relative to the 'green' intensity value offsets depicted in the images rendered by the visual sensor. For their implementation, two filters are configured for the equally distributed, horizontal split of the images that are rendered, where one filter is used to sample the bordering region of the images and the other, offsets from their absolute centre; this counteracts an error of the functionality used, where one filter cannot be configured to sample the entirely of an images side, thus two must be applicated alternatively. When sampling from either filter configured, the beacon objects location can be determined by a positive 'green' intensity value (larger than '0'), which in knowing of the filter's occupancy relative to the dimensions of the image, can be used to align the robots heading orientation with the beacon objects position, in the subjected environment.

Available in Appendix S

In continuation of the behavioural implementation for the robot's navigation to the beacon object, for explanation purposes, assume that the beacon object is initially detected within the right portion of an image rendered by the robot's visual sensor, the robots initial alignment to the beacon assumes a rightward angular traversal motion to be actuated; this alike other behavioural implementations, is achieved by proportioning the velocity displacements issued to the robots wheel motor components, and is determined by binomial state (Boolean). The robot's initial direction of alignment is assumed, given the robots prior traversal pattern(s) that cause the visual sensor to pan the environment, resulting in the beacon objects profile typically emerging at the borders of the sensors perspective frustum, that correlate to the trajectory of the motion actuated by the robot.

Available in Appendix T

Advancing from the initial direction of alignment being appropriated, the robot continues to angularly traverse until the beacon object is sampled as being at an alternative offset by the filters configured for the contrary portion of the images generated; this enables the robot to actuate angular traversal motions that are inversely proportional to its traversal pattern prior. Given both traversal patterns forward progression (accelerative and not pivot-focused), the robot is gradually able to oscillate towards the beacon object, until within the stationing range; for the traversal patterns mentioned, the robot at each alternation of angular traversal, applicates the wheel with a non-accelerative motor as its anchoring-point (velocity of '0'), this enables a forward advance to be achieved. Functionally, this behavioural strategy can be articulated by the proceeding pseudo notation:

If *the robot's initial alignment direction to the beacon has not been set* **then**

If *the beacon has been seen closer to the robot's right side* **then**

Robot aligns right with the beacon object

Else *the beacon has been seen closer to the robots left side* **then**

Robot aligns left with the beacon object

Elseif *the robot's initial alignment direction to the beacon has been set* **then**

If *the robot is aligning right with the beacon*
and *the beacon has not been detected closer to the robots left side* **then**

Robot aligns right with the beacon object

Elseif *the robot is aligning right with the beacon*
and *the beacon has not been detected closer to the robots left side* **then**

Robot aligns left with the beacon object

Elseif *the robot is aligning left with the beacon*
and *the beacon has not been detected closer to the robots right side* **then**

Robot aligns left with the beacon object

Elseif *the robot is aligning left with the beacon*
and *the beacon has been detected closer to the robot's right side* **then**

Robot aligns right with the beacon object

I. Random Wandering

Fundamental to the robot's target searching capability, random wandering is integrated for mobilising the robot and resultingly, nearing the robot to the beacon object, for which the robot can survey and navigate towards, overtime. For the strategy's implementation, the subsequent behaviours are featured:

- Forward traversal
- Angular traversal

Given the triumph of the strategies applications in developments of the prior controllers submitted, its configuration remains identical to its predecessors, in which appropriates the robot's exploration potential, as a continual series of forward and angular advances through its environment. To note, each of the traversal patterns mentioned, integrate metric randomisation and odometry to appropriate the extent at which each pattern is actuated by the robot, before then being alternated; this behaviour is subsumed upon an object being detected, for which edge following or obstacle avoidance behaviours are invocated, to aptly address.

II. Edge Following

As a navigation agent adapted to error marginalisation, edge following which advocates PID feedback control, is purposed for enhancing the robot's surveillance prospects, that assisted routing around the edges of objects encourages, for expediting the robots beacon tasking. Also featuring as a reproduced, efficacious behavioural strategy of past controller developments, edge following behaviours have been reclaimed for enabling the robot to maintain a set distance, or set-point, from the edges of objects that are detected adjacently, to the robot's side-most SONAR sensors (sensors '1' and '8'); this is provided that an object does not obstruct the robots forward advance around the environment, whereby, obstacle avoidance behaviours are instead invocated. Fundamental to said behaviours exhibition, the strategy applies error computations to the robot's wheel motors, as proportional displacements in each's velocities; these are actuated by the robot, to then regulate a stipulated distance from the faces of objects that it encounters, in a paralleled motion ('0.25' metres).

As previously revealed, the strategy was revised for the presence of cylindrical-based objects, which were previously unnoticed for the strategies application; this accommodates the deterrence of the robot from continuing to actuate edge following traversal patterns, around said objects, beyond their circumnavigation. Which is achieved by invocating the avoidance strategy's evasive sequence, 'edge end reached', that instructs the robot to evade objects that it detects, for a randomly passed interval. This sequences invocation proceeds an objects circumnavigation, for the maximisation of proximity surveillance and resultant beacon establishment. A given objects circumnavigation is determined by accumulating the robots heading orientation adjustments, overtime, which is odometrically derived and when surpassed, enables the robot to resume its motivated tasking(s).

III. Avoidance Strategy

Significant to the continuity of the robots tasking, obstacle avoidance as previously unseen within prior controller developments, is integrated as a multitude of subsidiary behavioural strategies, for combating an array of scenarios likely to the robot's encounter; as previously realised, the randomisation yielded by the robot's traversal, when wandering, potentializes the robot to unforeseen circumstances, which require to be addressed, behaviourally. For the strategies implemented, the following conditions can be surmounted by the robot:

- Enclosed, confined areas for exploration
- Frontally positioned obstacle evasion

As the predominant avoidance architecture of the strategy, Braitenberg's avoidance algorithm is applicated for an imbalance in frontal detections of objects, that cannot be appropriated by the controller's edge following strategy, as well as for the evasive sequence referred to prior, which enables the robot to evade object-dense areas of its environment, for a given duration. Providing a weight coefficient is passed to each of the robot's front-most facing sensors as detailed before, the robot is capable in evading objects at proportioned rates, which is reflected by its issued motor velocity displacements, for appropriating the positioning of objects that present varied resultant obstruction to the robot's continuity of traversal. Whereby, objects positioned ahead of the robot demand larger coefficients, given the vaster angular offset required for the robot to avoid collision incidents, when compared to objects positioned and detected parallel to the robot's sides. Dependant on the detection offset of said objects, the weight coefficients applied for the robot's evasive manoeuvres are configured to be inversely proportional to the side they are detected upon; this enables the robot to actuate leftward and rightward angular traversal motions, or alternatively, forward traversal motions given the scenario that an object is not detected (evasive sequence) or is detected equally. For the functional breakdown of the algorithm, refer to the following pseudo notation:

If *the difference between the robot's front sensor detections, regionally, is insignificant* **then**

Robot traverses' forwards

Elseif *the robot detects objects closer to its front-right side* **then**

Robot angularly traverses' left

Elseif *the robot detects objects closer to its front-left side* **then**

Robot angularly traverses' right

Else *the robots detects objects on both of its front sides equally* **then**

Robot traverses' forwards

Meanwhile, superseding the forward traversal invocation presented in the pseudo notation of the Braitenberg algorithm, for when objects are detected at equal lengths, the controller features the subsidiary behavioural strategy, reversing, for preventing collision incidents with objects that the robot, when invocating the algorithm abovesaid, would cause from forward advancements being actuated in the subject setting. Thereby, the robot is instead able to reverse away from said situations and offset its heading orientation (stuck state), upon the reversing motion climaxing, thus allowing the robot to then continue with its motivated tasking(s). For the behaviour's implementation, the robot surveys its front-most sensors forming symmetrically-opposed relations, every frame, which when their detections are compared, can be applied for the robot's acknowledgement, regarding whether an object is detected equally or unequally ahead (within a '0.005' metre threshold); provided the comparison returns an equal detection, the robot would then invocate reversing behaviours, which is determined by binomial state alternation (Boolean). Said climax of the behaviour is determined by the expiration of an arbitrarily configured timer, or as described previously, when objects are no longer detected ahead of the robot, or inversely, when objects are suddenly detected behind it; the robot appropriates all sixteen SONAR sensors (sensors '1' to '16') to accomplish this, which can also be realised by the behaviour's pseudo notation:

If *the robots front-most facing sensors detect an object* **then**

If *the robots back-most facing sensors do not detect an object* **then**

Robot traverses' backwards

Else *the robots back-most facing sensors detect an object*

Robot identifies readiness for its heading orientation offset
Robot identifies as being distanced from the object ahead

Else *the robots front-most facing sensors do not detect an object*

Robot identifies readiness for its heading orientation offset
Robot identifies as being distanced from the object ahead

For the robots heading orientation adjustment, proceeding the climax of its reversing motion, an additional, separate subsidiary behavioural strategy, stuck, is invocated for its primary functional purpose of seeking exit spaces, where objects do not reside at. Said strategy alike the reversing strategy, utilises all sixteen SONAR sensors onboard the robot, to determine a rotary direction for which it appropriates as the robots nearest path of evasion, or exit space, away from the object positioned ahead; this is processed by partitioning the robot's sensors relative to their positional offset from the robots heading orientation, as instrumented for the robots exit-alignment task, when subjected to the environments inner-most room (see ***Appendix H***). Each sensor partitions detections are accumulated for comparative sake, whereby, the rotary direction of the robot can be determined, inversely to the side of the robot of where objects are situated closest (cumulatively lower value). In compliance with the rotary direction determined, wheel velocity displacements are then invocated to the robot's wheel motor components, that appropriate and enable the robot to actuate pivot-focused motions, which offset the robots heading orientation, overtime, for an arbitrarily determined interval (ranging between '0.5' and '2' seconds). This strategy can be acknowledged by the pseudo notation featured below this passage:

If *the robot is rotating left* **then**

Robot rotates left
Deduct the timestep from the rotation timer

If *the rotation timer has depleted* **then**

Robot identifies its exit space as being found

Elseif *the robot is rotating right* **then**

Robot rotates right
Deduct the timestep from the rotation timer

If *the rotation timer has depleted* **then**

Robot identifies its exit space as being found

Given the scenario that the robot is exposed to an enclosed, confined area within its subjected environment, the robot possesses the ability to alternatively orientate until its front-most facing sensors (sensors '4' and '5') return non-detection statuses, inferring that an exit space (no object) resides ahead of the robots facing direction; where the Braitenberg avoidance algorithm is resultingly invocated, provided that the robot detects objects on all of either of its sideward and backward faces, allowing a forward advance of the robot to be progressively actuated, to overcome the spaces exit. This variant of the strategy is invocated upon the robot's sensors, omnidirectionally, returning six or more positive detection statuses simultaneously, which enables the robot to also motion-handle perpendicular arrangements of its environment (corner spaces), when further factoring edge following behavioural invocation. This behavioural strategy is primarily appropriated for the robot's exploration in the specifications of spaces said, for the purpose of environment charting maximisation, in which the robot may require to navigate back through the space's entry point, as its point of exit also. For this alteration of the behavioural strategy, refer to the following pseudo notation:

If *the robot is rotating left* **then**

If *the robots front-most facing sensors detect an object* **then**

Robot rotates left

Else *the robots front-most facing sensors do not detect an object* **then**

Robot identifies its exit space as being found

Elseif *the robot is rotating right* **then**

If *the robots front-most facing sensors detect an object* **then**

Robot rotates right

Else *the robots front-most facing sensors do not detect an object* **then**

Robot identifies its exit space as being found

D. Environment Charting

For charting the robot's discoveries in its subjected, unknown environment, the charting strategy utilises the detections returned by the robot's front-most facing SONAR sensors, to populate a series of coordinates relative to CoppeliaSim's global coordinate space, that can be plot and sampled, for a graphical series of online and offline-based map variants, as revealed previously. Purposed for the robot's mobilisation, exploration, navigation, and obstacle avoidance capabilities whilst charting its environment, all the controllers other existing, unmotivated behavioural strategies (taskings) are invocated simultaneously with the charting functionality, for acquiring the robots inference data, that is then used to settle said variations of maps. To interpret the robots inference data for the application of charting, said data is translated from its local coordinate space to the simulators global space, for transmuting the data to be relative to the robot's position in the environment, at the timestep a given object is detected.

For this calculation, the point of object detection is initially calculated using the Cosine ('X' dimension) and Sine ('Y' dimension) operations, that each summarise the angular offsets of the robots sonar sensors, from its heading orientation, that then applies as a multiplier to the accumulation of the sensors detection (distance from the robot), and the sensors positional offset from the robots origin; this formulates the 'X' and 'Y' coordinates at which an object is detected, relative to the robots position, locally. To calculate the positional offset of each sensor from the robot's origin, the magnitudinal difference [36] between each sensor's position and the robot's position is returned, for every timestep that is passed.

Available in Appendix U

Upon reasoning the object detection(s) as a two-dimensional coordinate, relative to the robot's position in local coordinate space, said coordinates are then passed into a rotation matrix, that translates each dimensional value from local to global coordinate space (see ***Appendix N***); this provides the point in the environment, at which an object is detected by the robot's sensor(s). Purposed as a preliminary data reduction technique of the strategy, each point populated by the process abovesaid is then numerically rounded, to the nearest decimal place ('1' decimal place); this secondarily aims to linearise the graphical output of each map variant also, thus being increasingly presentable. Meanwhile, as the strategies secondary data reduction but primary validation technique, points populated by the process described, are only passed to each of the map variants for plotting, upon multiple sensors detecting the same point in space or another point being known to adjacently border it (a resulting difference smaller or equal to '0.5'); this technique is used to filter outlying data from being plot and thereby being applicated to represent the robot's environment, inaccurately. Which is known to be the causation of "ultrasonic wave propagation" [37], that derives from the SONAR technology embedded by the robot's corresponding sensors.

I. Online Charting

As the online-based application of environment charting, an 'XY' format graph object within the CoppeliaSim IDE, is configured to handle the inference data parsed by the transformation process described prior. In which, for every sensor that detects an object during the robot's course of simulation, a plot point within said graph, is designated; duplicate points are not plot however, for computational resource management purposes. Given by name, the graph object updates its plots live and therefore continually with the data transduced by the robot's front-most facing sensors, which jointly, represents the structural arrangements of the robot's environment. Additional to the robot's object detections, the current position of the robot is also plot, for each timestep succeeded, which is purposed for demonstrating the robot's exploration and resultant pathing actuated, throughout the course of simulation runtime; these plots are coloured uniquely, to enable a vivid distinguishment.

Available in Appendix V

II. Offline Charting

In reuse of the data processed for the online map variants plot points, a series of offline-based maps could also be populated, where each occupies a varied approach to data reduction and or validation. For appropriating the apparency of offline map variants, the environment is orchestrated as a two-dimensional grid, that bares dimensional relevance to the size of the environment that the robot is subjected to, as well as the degree of numerical rounding inflicted to each plot; this would present to border the robot's detectable capability, which all the environment's objects reside within. The resultant dimensions used to plot points for each offline map variant is: '200' x '200' ('20' squared metres respectively), which dictates two-hundred potential indexes for either dimensional space, 'X' or 'Y'. In principle of this configuration, each of the parsed points detected by the robot can be used to index the array in mention, where each index can be identified as a neutral number ('0'), representing no object, or a positive number ('1' or more), representing an objects residence and its count of detections, over the course of the robot's simulation routine. For which the majority of the offline map variants configured, utilise as a validation metric, that is applied to determine outlying data by affirming an objects detection at the same location, on multiple accounts (twice or more); this proves an objects existence at a given position in the environment, or index in the array. Given the majority of the offline map variants in mention, each of said maps feature as a separate Microsoft Excel worksheet, that incorporate a series of conditional formatting of its cells, for the dimensions said, to colour coordinate where objects reside at, in the robot's environment; each of the variants submitted follow a linear data validation progression, that is presented by the worksheets tab-wise order and labelling.

Available in Appendix W

Founding the final offline-based map variants, RANSAC derived data validation was applicated for the purpose of replacing plot-based illustrations, with linearised line series, which were believed to represent the robot's environment more accurately, as it's known resolve for distinguishing inlying and outlying data supposed. Fundamental to its calculation, each of the robot's object detections throughout a given simulation routine, are stored into a separate array, which exponentially increases in size, for every detection submitted by the robot. For the robots tasking, a target detection count is arbitrarily generated at the start of simulation runtime, which purposes as the task's capacity, for which the robot should attain map-related data until; this is used to indicate when the robot should progress with its motivated taskings, and thereby being to return to the inner-most rooms centre. This is signalled by the controller through binomial state alternation (Boolean), upon the taskings target, being amassed, also indicating that the environment has been charted, entirely.

For the calculations entailed by the algorithm, it was imperative that the target set for RANSAC was of a non-remaindering amount (even), and was significant enough in size so that the dataset could be sectioned whilst returning a satisfactory amount of data per section determined (ranged '12000' to '18000'), which could then be used for computing a series of line coordinates, representing the data's best fit, or as otherwise known, lines that consider the greatest amount of agreeing data (points), verified by each's Euclidean distance [38] from gradients of said lines (distance of '1'); these are output for plotting and drawing, as interpolation lines. For computing the coordinates that produce said lines of best fit, a stop condition is passed to the algorithm, which determines when each sections line is to be settled; this is configured as one-thousand iterations, to enable each line settlement to be sensitively determined. Upon the coordinates being ratified, they like other sections computations, are then appended to another distinct array, for which they all remain to reside within. To note, RANSAC computations are invocated upon simulation runtime termination.

When visualising the coordinates compiled, the Simple and Fast Multimedia (SFML) library package [39] is applicated for its known graphical capabilities, in representing numerical data. This is used in conjunction with Microsoft's Visual Studio IDE, as a familiar software application, to cooperate in the rendition of two offline-map variants; one of said variants produces the RANSAC derived output untampered, whereas the remaining variant is purposed for visualising the validated output of the RANSAC algorithm. The data validation technique approached, concerns the abandonment of coordinates from being processed through the libraries graphics pipeline, that when normally interpolated to render lines, surpass the length threshold configured (beyond or at '8.0' relative to the coordinate space); this aims to render the environments visualization, without the apparency of cross-sectional lines, given that the coordinates computed, are processed irrespective of their positioning in the robot's environment (randomly assorted).

Available in Appendix X

E. Return to the inner-most rooms centre

Purposed for the robots final tasking, the controllers random wandering strategy was refactored for capacitating the addition of an exploration-centred subsidiary state, explore. The nature of said behavioural subsidiary, encompasses the robot's ability to maintain its heading orientation, focused on the centre-point of the environments inner-most room, for which is known by the robot, since its preliminary taskings; notably, this behaviour subsumes the random wandering, angular traversal subsidiary state by binomial state (Boolean) alternation, for purposing an established means of exploration, as opposed to being randomised, like originally purposed. For its implementation, the robots heading orientation requires continual translation from CoppeliaSim's orientation space, which regulates 'Z' dimensional orientation, within the '180' to '90' degree boundaries, both positively and negatively. Whereby, in focus of the taskings odometric implications, heading adjustments invocated by the robot via wheel motor velocity displacement, require to be proportionally calculated, accumulated, and distributed overtime, for achieving the degree of exactness instructed for the robots tasking. This is appropriated by normalizing the robots heading orientation between the '0' to '360' degree standard alternatively, which incorporates a series of accumulation and numerical clamping operations to preserve; this procedure resultingly revokes negative orientations of the robots heading, from being factored in the odometry derived calculations, that concern the robots heading adjustment, overtime.

Available in Appendix Y

For regulating the robots heading alignment to the inner-most rooms centre-point, the angular difference between the robot's current position and the inner-most rooms centre, is calculated using the arctangent operation, which computes the angular difference between the two points in space, and is used to identify the robots angular offset, from its supposed target. This metric is calculated in advance of the robot determining its alignment direction to the rooms centre, for which is determined as the direction of angular traversal, that requires a less significant orientation adjustment, to nullify the angular difference between the robot and the inner-most rooms centre. This can also be realised by the provided pseudo notation:

If *the robots heading orientation is smaller than the angular offset* **then**

> **If** *the difference between robots heading and angular offset is smaller than its inverse (made positive)* **then**
>
> > ***Robot angularly traverses left***
>
> **Else** *the difference between robots heading and angular offset is larger than its inverse (made positive)*
>
> > ***Robot angularly traverses right***

Elseif *the robots heading orientation is larger than the angular offset* **then**

> **If** *the difference between robots heading and angular offset (made positive) is smaller than its inverse* **then**
>
> > ***Robot angularly traverses left***
>
> **Else** *the difference between robots heading and angular offset (made positive) is larger than its inverse*
>
> > ***Robot angularly traverses right***

Upon the robots angular offset and aligning direction to the inner-most rooms centre being determined, the robot continually invocates angular traversal patterns, that maintain the robots heading orientation aligned with the rooms centre-point, this as previously mentioned, is achieved by invocating wheel velocity displacements, to the robot's motor components, over time; in which, the robot actuates accelerative-based motions instead of pivot-focused motions, to maintain a forward advance to the rooms centre. As measured, the robots heading adjustment is accumulated overtime, to odometrically signal when the robot should no longer invocate angular traversal behaviours; this is determined by the robots cumulative heading adjustment, climaxing to its angular offset from the rooms centre, in which, through Boolean state alternation, the robot can resulting invocate the forward traversal behaviours of the random wandering strategy, until another offset in its alignment is realised. Supporting the robots tasking, the controllers edge following, and obstacle avoidance strategies are also invocated for collision evasion and navigation resolutions, that enable the robot to gradually appropriate the rooms point of entry and avoid incidents of collision with it, as instructed to do so. Upon the robot being contained within the bordering structure of the inner-most room once more, the robot continues to navigate towards and maintain its heading alignment with the centre-point of the room, until their magnitudinal difference in position, becomes relatively equal (within '0.05' metres). At which point, the simulation routine is terminated, in acknowledgement of all the robot's tasks being completed.

IV. Experimental Design

A. Locate the inner-most rooms centre

For evaluating the configuration submitted for the robot's room centering capabilities, the robot was placed in an array of positions offsetting the rooms centre, that remained it to locate within the borders of said room's structural arrangement; this process enables the robot's competency in task completion to be acknowledged and appropriated in the future adaptations applied to the controller developed. Where it was expected of the robot to be able to appropriate the literal or near centre-point of the inner-most room, for its application to be regarded as successful.

To execute the investigation detailed, the robot was subjected to a series of simulation routines, for yielding definitive results that could confidently model the robot's performance for task in focus; this was approached in knowing that the robot could be subjected to any of the scenarios populated, in lifelike domains. For the scenarios submitted to the robot for the purposes of this experimentation, see ***Appendix Z***.

B. Exit the inner-most room

In examination of the robot's performance for its exit-alignment tasking, which concerns its competencies in directional alignment with the inner-most rooms exit, or doorway, and overcoming it, objects of varied profiles were placed in and around the proximity of the rooms exit space, to analyse the robot's resultant ability to appropriate an exit, when subject to obstruction. Separate experimentation has also been conducted for the same interest of the robot, however, for when the robot starts at different positions in the room and when obstacles are not factored in its tasking. For a successful behavioural implementation to be realised for its tasking, the robot was expected to exit the inner-most room of its environment, without cause of collision with the walls constituting to the room's doorway and the obstacles placed around their location, if factored.

Like the previously led experimentation for the robots preliminary tasking, the robot is further exposed to a series of simulation cycles, which are also purposed for acquiring results that can be used to reason its performance, in attempt of completing the task relevant. For the scenarios derived for the robot's experimentation said, see ***Appendix AA*** for obstacle factored tasking and ***Appendix AB*** otherwise.

C. Locate and station at the beacon

For exercising the robot's capability within locating, navigating towards, and stationing at the beacon object designated for its tasking, it was deemed sensible to variate the beacon objects location in the environment, as well as to disrupt the robot's navigation and stationing abilities by introducing more and less obstacles around said environment, arbitrarily, to identify its resultant ability to continue with the tasks said. Given the nature of the robots tasking, it was expected of the robot to establish a means of stationing at the beacon object, regardless of its location in the environment; for which the strategy submitted relies upon. to be regarded as an accomplished approach.

As settled by the investigations led for prior taskings of the robot, the robot is once again subjected to a series of simulation routines, to ensure that the behavioural observations recorded about the robot are fair and therefore valid, for determining the strategy's resultant compatibility and suitability, in the enablement of task completion. To visualise the scenarios formed for the study purpose, see ***Appendix AC*** for beacon displacement tasking and ***Appendix AD*** for obstacle augmentation tasking.

D. Environment Charting

Aligning with the experimentation conducted for the robot's beacon tasking, to examine the accuracy and or presentability of the environment charting methods disclosed, objects with various profiles were arbitrarily placed in the robots subjected environment, to determine whether the map outputs rendered, draw similarities and or resemblances to the objects placed in the scene. Given each map variants purpose, to provide a structural-focused overview of the robot's environment, said experimentation was believed appropriate for exercising their suitability for this application; where clear-cut representations of the robot's environment, determine each's achievement as an appropriate charting candidate.

Established by the investigations led for the robot's previous tasks, the robot was yet again exposed to a series of simulation cycles, for yielding confident analysis regarding the performance of the robot's environment charting strategy, However, given the strategy's demand in data volume and thereby resultant computational expense and time consumption, for gathering said data and then compiling renditions of it, the capacity of experimentation led for the environment charting strategy, is seemingly superficial when compared to the investigations conducted for the robots other taskings. This remains appropriate to determining the strategies performance, nevertheless. For the renditions output by the strategy, and for the scenarios mentioned, see ***Appendix AE***.

E. Return to the inner-most rooms centre

To evaluate the robot's competence in relocating the centre-point of its environments inner-most room, the environment as configured in previous task examinations, was subjected to the placement of obstacles, varying by profile and quantity, to determine whether the robot could appropriate a means of navigation to the rooms centre, regardless of the conditions it is exposed to (within reasonable measures). To further examine the strategies suitability for the task in focus, the robots starting position is also offset, for every trial, in which dictates that the behaviour is to be tested in isolation, given that the robot invocates mobilisation behaviours, in its prior taskings. Provided that the robot could return to the centre-point of the inner-most room, without incidents of collision being observed, the behavioural strategy implemented for the robots tasking, could be recognised as a successful methodology.

As the investigative process involved in the determination of the strategy's compatibility, for the tasks said, the robot's performance is again subjected to a series of simulation cycles, to verify the observations constituting to its resultant suitability. For the scenarios designed for scrutinising the robot's behavioural strategy, see ***Appendix AF***.

V. Results

A. Locate the inner-most rooms centre

In accordance with the behaviours exhibited by the robot for the scenarios submitted, it is apparent that the robot is mostly able to appropriate the centre-point of the environments inner-most room, at which the robot invocates its exit-alignment tasking. For the scenarios acknowledged successful, it was noticed that the robot would initially pose with a facing orientation directed towards the centre of the room, where it was inevitable that the robot could locate the centre of the room, from all of its positional offsets trialled. However, it is also recognized that the robot is unable to appropriate the centre of the room, when initially, directly facing away from the rooms centre, whilst being positioned closely to a bordering wall, for any offset in position. At which the robot collides with said walls, as can be seen from the scenarios visualised.

Given the behavioural deficiency outlined, it appears that the strategy designed for the robots tasking is not entirely suited to its accomplishment. Whereby, the robot demonstrates behavioural incompetence within the determination and actuation of an evasive manoeuvre, that preserves its tasks continuity; this infers that the robots potential starting position can be less varied, for enabling it to climax to the task's completion. However, to overcome said insufficiency, the behavioural strategy should support a backward or pivot-focused traversal sequence as previously implemented for the controller, that enables the robot to actuate a reversing or turning motion for evading a threshold of closeness to the walls bordering it, thus avoiding further incidents of collision, whilst expanding the possible start position of the robot in the rooms space, that permits it to complete its task, nonetheless.

B. Exit the inner-most room

Evidenced by the experimentation conducted, the behavioural implementations for the robots exit-alignment and exit emergence, are sufficient for when obstacles are factored in the robots tasking and when obstacles are not but the robots starting position is offset also. Firstly, relating to the accomplished scenarios populated when obstacles are factored, it is inevitable that when objects are positioned at, beyond, within (offset) or offset from the exit spaces centre, the robot can appropriate an exit strategy, at which it emerges from the space without any incidents of collision with the spaces bordering structure. However, upon obstacles being centrally aligned with the rooms exit space, when at or within the proximity of it, the robot is unable to emerge from the room, due to its evasive sequence being invocated, in result of the robot gradually positioning too close to the obstacle, as it attempts to near the space already aligned to; the robot exits the tasks strategy and invocates unmotivated taskings alternatively. Meanwhile, for the scenarios not factoring obstacles but varied starting positions, the robot is capable of emerging from the room, regardless of its positional offset initially.

Proven that the robot is unable to emerge from the room, when an obstacle is positioned within the central alignment of its exit space, the strategy compiled for the robots tasking cannot be recognised as wholly effective, although the robot's fundamental priority to evade, is addressed sensibly. To potentially ensure that the robot emergences from the room consistently, for each simulation routine executed, the threshold at which the evasive sequence is invocated should be reduced, as well, the velocity displacements applied to the robot's motor components overtime, could also be fine-tuned to ensure that the robot maintains an absolute central alignment with its exit spaces borders. At which the evasive sequence said, would not be invocated, unless the room poses a restrained margin for the robot to emerge from.

C. Locate and station at the beacon

In direct correspondence to the results yielded by the relevant investigations, the behavioural strategies implemented for the robot's beacon tasking, were evidently well-adapted for their purpose. In which the robot, regardless of the beacon's location and the presence of obstacles in its environment, was capable in locating, navigating towards, and stationing at. Inevitably, the robot's adherence to locating the beacon was expedited by the presence of obstacles, which through the invocation of the controllers unmotivated taskings, the robot was able to navigate around and evade, to enhance its surveillance potential. Undoubtedly, the robot's random traversal behaviours were also sufficient in the robot's mobilisation and resultant beacon establishment, for which, is generally achieved quickly; this is determined by the length of the routing exhibited by the robot. In continued mention of the robots unmotivated taskings, from their invocation, the robot was also able to maintain behavioural continuity for its task's completion, whereby, no collision incidents were recorded throughout the simulation routines executed. Thereby, the strategies implementation can be considered exemplary.

However, in the rendition of behavioural desirability, the robots means of navigating towards the beacon object via alternating angular traversal patterns, could be bettered by PID feedback control alternatively, where the oscillatory motions of the robot would become insignificant overtime, from the applications known error marginalisation process. This would project an even more so decreased likelihood of collision occurrence, given the more direct forward advance that the robot would resultingly exhibit, from the forward-focused traversal patterns advocated by PID feedback control (correction to error being the robots heading alignment offset to the beacon).

D. Environment Charting

As can be determined by the renditions of the robot's environment that are compiled by the environment charting strategy implemented, each of the map variants featured, present the robots environment in a comprehensive format, where the profiles of objects can mostly be determined and the volume of outlying data exported, is relatively insignificant; this thereby determines that the environment charting strategy approached, is tailored to environmental depiction, for a two-dimensional plane. For the renditions of the environment that encompass object placements for the robot's encounter, it is obvious that the faces of objects were not all detected by the robot, given the apparent absence of plot points and neutral counts of detection ('0'); this infers that the robot's exploration was somewhat limited, which is supported by the robots routing, that is also rendered by the online map variant. This does not degrade the capability and resultant performance of the charting strategy directly, however, this is given by the robot being mobilised by the random wandering, edge following and obstacle avoidance strategies, which are not affected by the progressions entailed within the environment charting strategy. Meanwhile, in correspondence to the secondary renditions of the environment, where the robot was not exposed to obstacle confrontation, the precision achieved within the visualisation of the environment's structural arrangements, is considered detailed, where each variant is recognised as approximately representing the entirety of the environments outline, from an above perspective. Inevitably, the environment charting strategy, forwards an accomplished approach to map construction.

For enhancing the strategy submitted for the robots tasking, the values of metrics applied to the RANSAC derived data calculations, could be fine-tuned for improving the RANSAC renditions of the robot's environment by simplifying its output further, through using fewer lines to represent it; this would also require an improved line redaction technique, assuming that each line length would exponentially increase, with the decrease in lines drawn. Furthermore, the offline map variants could also incorporate cell-filling algorithms, for better indicating the presence of objects and unreachable boundaries of the robot's environment, as solid-colour-filled entities.

E. Return to the inner-most rooms centre

From the results attained by the experimentation conducted, it is inevitable that the behavioural strategy configured for the robots tasking, is utterly compatible for its completion, as observed in both populated and unpopulated variants of the robots subjected environment. Whereby, it was noticed for the robot to not collide with any of the objects placed in the environment, as the controllers edge following, and obstacle avoidance strategies precluded, upon their invocation. Indisputably, the robot also demonstrated competency in navigating to the centre-point of the inner-most room, regardless of its positional offset that it resided at, upon the task being instructed for resolution; where the robot was observed to stop at the exact centre of the room in mention, for every simulation routine recorded, as well. However, it can also be noticed that when objects are populated in the robot's environment, its routing typically develops to be increasingly elongated, when compared to its routing in the object depopulated variant; the strategy does not consider distance, which disadvantages the efficiency of the robots tasking.

For the behavioural deficiency mentioned, an artificial intelligence (AI) orientated path-finding algorithm, could instead be integrated into the strategy developed for the robots tasking, to accommodate efficient routing to the inner-most rooms centre, which would enable the tasking to be completed considerably faster than what the current implementation could compute; this would require the use of charting data, to appropriate an understanding of where objects have been located or detected it, for compiling paths to the inner-most rooms centre, that target the shortest traversal distance(s). This process can be resolved by a heuristic function [40], to achieve said behaviours.

VI. Conclusion

In summary of the controller configuration submitted, for the application of the Pioneer P3-DX mobile robot, it is evidenced throughout the scrutiny of robot's behavioural strategies, that the controller is well-established for its instructed purpose, to enable the robot to complete a series of complex, predefined tasks; each of which concerning alignment, emergence, surveillance, and navigation behaviours, for their accomplishment. Where it has been acknowledged throughout the passages comprising this document, that the robot when equipping said strategies and more, is able to complete each of the tasks purposed for its routine; the robot achieves this in use of its onboard ultrasonic sensors, that collectively derive its omnidirectional spatial awareness, as well as a vison sensor, that complements its visual recognition capabilities. With relevance to the experimentations led, in determining each strategy's suitability for the tasks instructed, it is accepted that the robot is mostly capable in overcoming obstructive and densely populated environments, for which the robot can supplement its task-agnostic behavioural strategies, to overcome; this is inevitably achieved by the controllers robust architecture and resultant behavioural state transition configuration, that enables either of the behaviours to subsume one another, in accordance with the state priority, that the controller regulates. Thereby, the configuration of the controller submitted, is definitively compatible for the purposes of its development and should therefore be regarded as a successful candidate, in this domain.

Throughout the controller's development cycle, both theoretical and practical applications of visual recognition instruments, has been realised, which stretches to the domain of graphics rendering pipelines. As well, the practical appliance of exploration behaviours for mobile robots, has been introduced and investigated theoretically, for the potential, future controller developments, that purpose target-finding functionality, for objective assignments. Moreover, the algorithms incorporated within environment charting processes have been taught, alongside their approaches to assembling numerical data, for devising graphical visualisations of a subject's environment. Inevitably, a diverse basis of knowledge has been attained, over the duration of the controller's development.

Provided that more time and computational resources were available, to further the development state of the controller submitted, many of the non-algorithmic implementations of the robot's behavioural strategies, would be replaced with well-established algorithms that are matched to a task's requirements; in anticipation of yielding increased behavioural stability and efficiency, in respect of reducing each tasks timing for completion.

References

[1] Génération Robots (2020) *Pioneer P3-DX mobile robot*. [Online] Génération Robots. Available from: https://www.generationrobots.com/en/402395-robot-mobile-pioneer-3-dx.html [Accessed: 05/01/21].

[2] MathWorks (2021) *What is SLAM?* [Online] MathWorks. Available from: https://uk.mathworks.com/discovery/slam.html [Accessed: 05/01/21].

[3] HARSANYI, K. and KISS, A. and SZIRANYI, T. and MAJDIK, A. (2020) MASAT: A fast and robust algorithm for pose-graph initialization. *Pattern Recognition Letters*. [Online] 129 (2020). Available from: https://www.sciencedirect.com/science/article/pii/S0167865519303241 [Accessed: 05/01/21].

[4] ESPINOSA, F. and SALAZAR, M. and PIZARRO, D. and VALDES, F. (2010) *Remote and Telerobotics*. London: IntechOpen.

[5] OHYA, A. and KOSAKA, A. and KAK, A. (1998) Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing. *IEEE Transactions on Robotics and Automation*. [Online] 14 (6). Available from: https://ieeexplore.ieee.org/document/736780 [Accessed: 05/01/21].

[6] O'REILLY (2021) OpenGL Programming Guide. [Online] O'REILLY. Available from: https://www.oreilly.com/library/view/opengl-programming-guide/9780132748445/ch04lev2sec1.html [Accessed: 05/01/21].

[7] KIM, K. and KIM, M. and CHONG, N.Y. (2010) RFID based collision-free robot docking in cluttered environment. In: *Progress in Electromagnetics Research.* Massachusetts: EWM Publishing, pp. 199-218.

[8] Coppelia Robotics (2021) *CoppeliaSim from the creators of V-REP*. [Online] Coppelia Robotics. Available from: https://www.coppeliarobotics.com [Accessed: 05/01/21].

[9] Microsoft Corporation (2019) Microsoft Visual Studio Community. Version: 16.5.5 [Software] Washington: Microsoft Corporation

[10] NIELDFELT, P.C. and BEARD, R.W. (2013) Recursive RANSAC: Multiple Signal Estimation with Outliers. In: *9th IFAC Symposium on Nonlinear Control Systems*, *Toulouse, France, September 2013*. Amsterdam: Elsevier, pp. 430-435.

[11] Microsoft (2021) *Microsoft Excel*. [Online] Microsoft. Available from: https://www.microsoft.com/en-gb/microsoft-365/excel [Accessed: 06/01/21].

[12] SEO, W. and BAEK, K-RYAL. (2017) Indoor Dead Reckoning Localization Using Ultrasonic Anemometer with IMU. *Journal of Sensors*. [Online] 2017. Available from: https://www.hindawi.com/journals/js/2017/3542354/ [Accessed: 06/01/21].

[13] ZAKI, J.A.M. and ARAFA, O. and AMER, S.I. and BASHTA, A.H. (2012) MOBILE ROBOT POSITIONING USING ODOMETRY AND ULTRASONIC SENSORS. *Journal of Cybernetics and Informatics*. 13 (2012). Available from: https://www.researchgate.net/publication/274894432_MOBILE_ROBOT_POSITIONING_USING_ODOMETRY_AND_ULTRASONIC_SENSORS [Accessed: 06/01/21].

[14] TOAL, D. and FLANAGAN, C. and JONES, C. and STRUNZ, B. (1995) Subsumption Architecture for the Control of Robots. [Online] Available from: https://www.researchgate.net/publication/244321030_Subsumption_Architecture_for_the_Control_of_Robots [Accessed: 06/01/21].

[15] MURPHY, R.R. (2000) Introduction to AI Robotics. [Online] Available from: https://www.researchgate.net/publication/238699045_Introduction_to_AI_Robotics [Accessed: 06/01/21].

[16] KELASIDI, E. and MOE, S. and PETTERSEN, K.Y. and KOHL, A.M. and LILJEBACK, P. and GRAVDAHL, J.T. (2019) Path Following, Obstacle Detection and Obstacle Avoidance for Thrusted Underwater Snake Robots. [Online] Available from: https://www.frontiersin.org/articles/10.3389/frobt.2019.00057/full [Accessed: 06/01/21].

[17] SIMPSON, J. and JCOBSEN, C.L. and JADUD, M.C. (2006) Mobile Robot Control the Subsumption Architecture and occam-pi. In: The 29th Communicating Process Architectures Conference, CPA 2006, organised under the auspices of WoTUG and the Napier University, Edinburgh. Berlin: ResearchGate, pp. 225-236.

[18] ZHANG, P. (2008) Application Software for Industrial Control. In: ZHANG, P. (ed) *Industrial Control Technology.* Norwich: William Andrew Publishing, pp. 569-673.

[19] KUMAR, D.U. and NISHA, N.M. and MATHIVANAN, N. (2018) Tracking of a PID Driven Differential Drive Mobile Robot. *International Journal of Mechatronics, Electrical and Computer Technology*. [Online] 8 (27). Available from: http://www.aeuso.org/index.php?volume=48&issue=46 [Accessed: 06/01/21].

[20] Control Soft Incorporation (2021) PID Tuning Objectives and Considerations. [Online] Control Soft Incorporation. Available from: https://www.controlsoftinc.com/pid-tuning-objectives-and-considerations/ [Accessed: 06/01/21].

[21] FRONTONI, E. (2012) *Vision Based Mobile Robotics: Mobile Robot Localization using vision sensors and active probabilistic approaches*. Berlin: ResearchGate.

[22] Coppelia Robotics (2021) *Regular API function*. [Online] Coppelia Robotics. Available from: https://www.coppeliarobotics.com/helpFiles/en/regularApi/simHandleVisionSensor.htm [Accessed: 07/01/21].

[23] Learn OpenGL (2017) *Depth testing*. [Online] Learn OpenGL. Available from: https://learnopengl.com/Advanced-OpenGL/Depth-testing [Accessed: 07/01/21].

[24] EDUCBA (2020) *RGB Color Model*. [Online] EDUCBA. Available from: https://www.educba.com/rgb-color-model/ [Accessed: 07/01/21].

[25] XIE, Y. and YAN, X. and CHEN, M. and CAI, J. and TANG, Y. (2019) An autonomous exploration algorithm using environment-robot interacted traversability analysis. In: *2019 IEEE/ RSJ International Conference on Intelligent Robots and Systems (IROS), Macau.* New York: IEEE, pp. 4885-4890.

[26] LEKKALA, K. and MITTAL, V.K. (2014) PID Controlled 2D Precision Robot. [Online] Available from: https://www.researchgate.net/publication/288424236_PID_controlled_2D_precision_robot [Accessed: 07/01/21].

[27] ANG, H.K. and CHONG, G. and LI, Y. (2005) PID Control System Analysis, Design, and Technology. In: *IEEE Transactions on Control Systems Technology*, *20th June 2005.* New York: IEEE, pp. 559-576.

[28] GOCHEV, I. and NADZINSKI, G. and STANKOVSKI, M. (2017) Path Planning and Collision Avoidance Regime for a Multi-Agent System in Industrial Robotics. Machines. Technologies. Materials. [Online] Available from: https://www.semanticscholar.org/paper/PATH-PLANNING-AND-COLLISION-AVOIDANCE-REGIME-FOR-A-Gochev-Nadzinski/69faf19aaf406796377141ed6ba76a4dd6641f23 [Accessed: 07/01/21].

[29] P-LUJAN, J-LUIS. and P-YAGUE, J-LUIS. and TEN-S, ENCRIQUE-J. and SIMARRO, R. and BENET, G. (2015) Distributed Sensor Architecture for Intelligent Control that Supports Quality of Control and Quality of Service. *Sensors*. [Online] 15 (3). Available from: https://www.mdpi.com/1424-8220/15/3/4700/htm [Accessed: 07/01/21].

[30] LAKAEMPER, R. and LATECKI, L.J. and SUN, X. and WOLTER, D. (2005) Geometric Robot Mapping. In: *Discrete Geometry for Computer Imagery, 12th International Conference, DGCI 2005, Poitiers*. Berlin: ResearchGate, pp. 11-22.

[31] The University of Edinburgh School of Informatics (2021) Transformation. [Online] The University of Edinburgh School of Informatics. Available from: http://www.inf.ed.ac.uk/teaching/courses/cg/lectures/cg3_2013.pdf [Accessed: 08/01/21].

[32] RIAZ, Z. and PERVEZ, A. and AHMER, M. and IQBAL, J. (2010) A fully autonomous indoor mobile robot using SLAM. In: *2010 International Conference on information and Emerging Technologies, Pakistan, June 2010*. New York: IEEE, pp. 1-6.

[33] Math2 (2006) *LuaSearch*. [Online] Math2. Available from: http://math2.org/luasearch/math.html [Accessed: 08/01/21].

[34] arm developer (2021) *Understanding Render Passes*. [Online] arm developer. Available from: https://developer.arm.com/solutions/graphics-and-gaming/developer-guides/learn-the-basics/understanding-render-passes/single-view [Accessed: 09/01/21].

[35] Coppelia Robotics (2021) *Regular API function*. [Online] Coppelia Robotics. Available from: https://www.coppeliarobotics.com/helpFiles/en/regularApi/simGetVisionSensorImage.htm [Accessed: 09/01/21].

[36] Math Is Fun (2017) *Distance Between 2 Points*. [Online] Math Is Fun. Available from: https://www.mathsisfun.com/algebra/distance-2-points.html [Accessed: 10/01/21].

[37] MAJCHZRAK, J. and MICHALSKI, M. and WICZYNSKI, G. (2009) Distance Estimation with a Long-Range Ultrasonic Sensor System. *IEEE Sensors Journal*. [Online] 9 (7). Available from: https://www.researchgate.net/publication/224492926_Distance_Estimation_With_a_Long-Range_Ultrasonic_Sensor_System [Accessed: 10/01/21].

[38] ROSALIND (2021) *Euclidean Distance*. [Online] ROSALIND. Available from: http://rosalind.info/glossary/euclidean-distance/ [Accessed: 10/01/21].

[39] SFML (2021) *Simple and Fast Multimedia Library*. [Online] SFML. Available from: http://rosalind.info/glossary/euclidean-distance/ [Accessed: 10/01/21].

[40] FERGUSON, D. and LIKHACHEV, M. and STENTZ, A. (2007) *A Guide to Heuristic-based Path Planning*. [Online] Available from: https://www.cs.cmu.edu/~maxim/files/hsplanguide_icaps05ws.pdf [Accessed: 11/01/21].

Appendices

Appendix A:

Figure 1: Pioneer P3-DX mobile robot [1].

Appendix B:

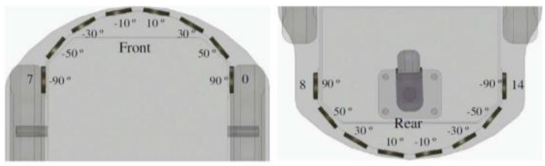


Figure 2: Pioneer P3-DX SONAR sensor arrangement [7].

Appendix C:

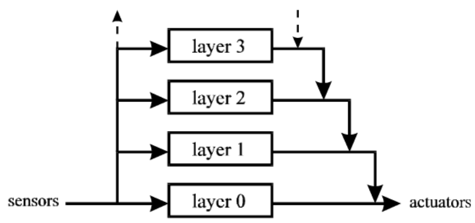


Figure 3: Rodney Brooks' behavioural subsumption architecture [14].

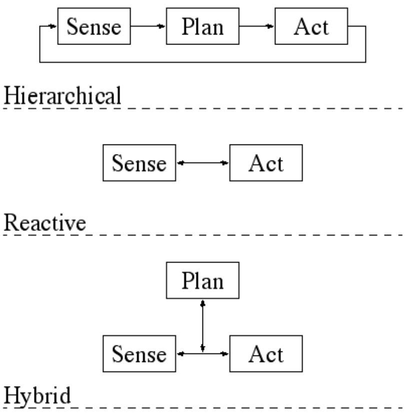


Figure 4: Robin Murphy's behavioural architecture paradigms [16].

Appendix D:

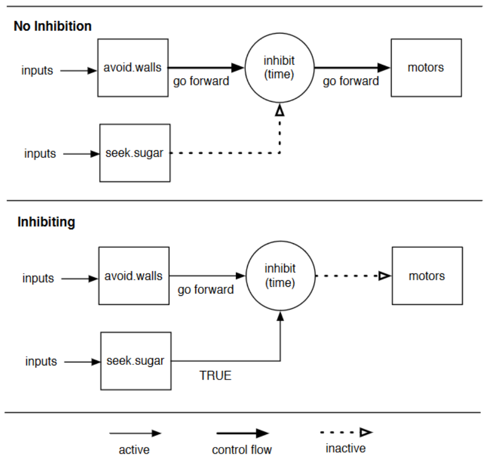


Figure 5: Visualisation of the behavioural inhibition process [17].

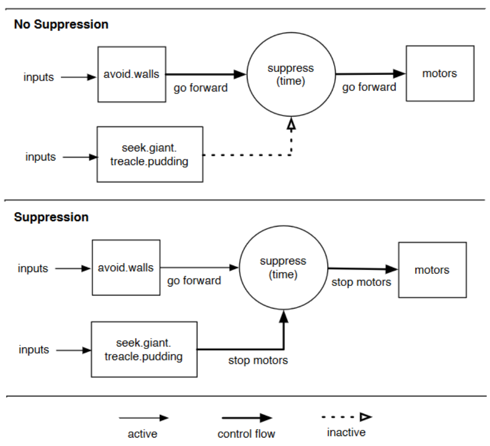


Figure 6: Visualisation of the behavioural suppression process [17].

Appendix E:

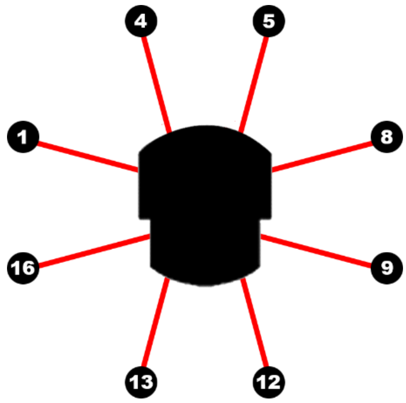


Figure 7: Pioneer P3-DX ultrasonic sensor pairs, for each facial region of the robot. Used to determine the centering behaviours exhibited by the robot, when establishing the centre point of the inner-most room, within its subjected environment.

Appendix F:

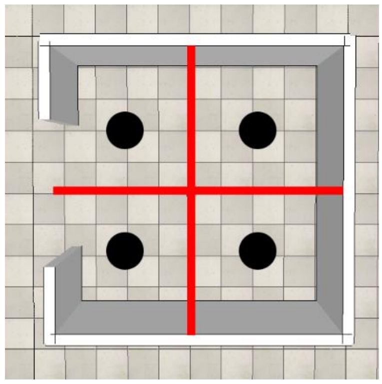


Figure 8: Inner-most rooms subdivisions, for the trialled application of four PID controllers, to enable the robot to establish the spaces centre point.

Appendix G:

Figure 9: Pioneer P3-DX, front-most facing SONAR sensor partitioning visualization, used to applicate the robot exiting, alignment behaviours. When exiting the inner-most room of the robots subjected environment.

Appendix H:

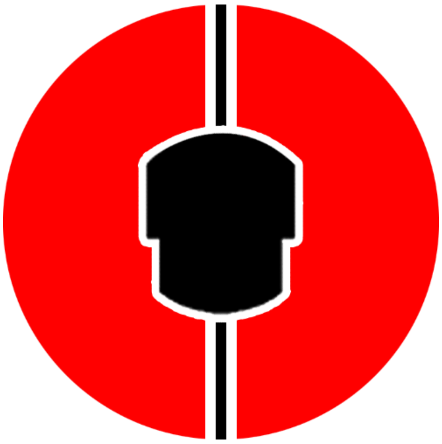


Figure 10: Pioneer P3-DX, ultrasonic sensor sideward partitioning visualisation, used to determine and applicate the aligning direction of the robot, when establishing the exit of the inner-most room, of its subjected environment.

Appendix I:

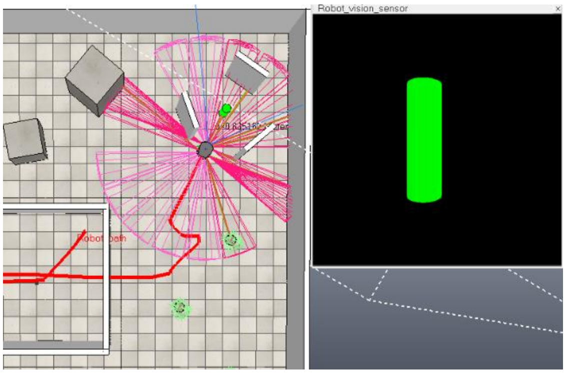
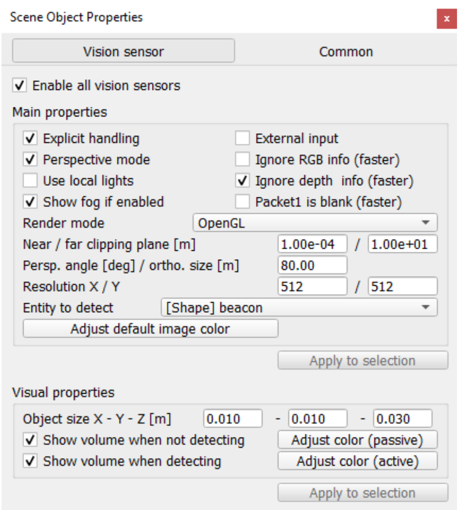


Figure 11: Vision sensor equipped by the Pioneer P3-DX, floating view visualisation of the beacon object, when the robot is navigating towards its location in the robots subjected environment.


Scene Object Properties

Vision sensor | Common

☑ Enable all vision sensors

Main properties

☑ Explicit handling ☐ External input
☑ Perspective mode ☐ Ignore RGB info (faster)
☐ Use local lights ☑ Ignore depth info (faster)
☑ Show fog if enabled ☐ Packet1 is blank (faster)
Render mode: OpenGL
Near / far clipping plane [m]: 1.00e-04 / 1.00e+01
Persp. angle [deg] / ortho. size [m]: 80.00
Resolution X / Y: 512 / 512
Entity to detect: [Shape] beacon
Adjust default image color
Apply to selection

Visual properties

Object size X - Y - Z [m]: 0.010 - 0.010 - 0.030
☑ Show volume when not detecting — Adjust color (passive)
☑ Show volume when detecting — Adjust color (active)
Apply to selection

Figure 12: Vision sensor equipped by the Pioneer P3-DX; configuration used for the vision sensors application in the beacon tasking of the robot, for achieving task compliance.

Appendix J:

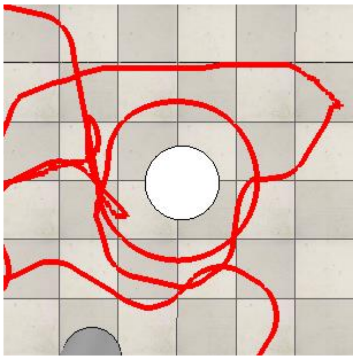


Figure 13: Edge following behavioural strategy refactoring, implementation supporting cyclical surveying of objects with followable, adjacent faces; alternatively, cylindrical-objects, as exemplified by the spherical 'red-coloured' pathing of the Pioneer P3-DX.

Appendix K:

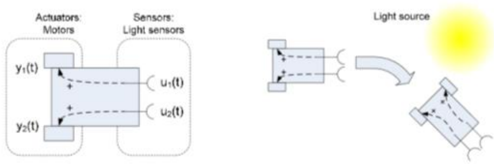


Figure 14: Valentino Braitenberg avoidance algorithm visualisation, for a light-sensitive sensor application. Mobile robot manoeuvrers away from the light source in the environment, that it is subjected to [29].

Appendix L:

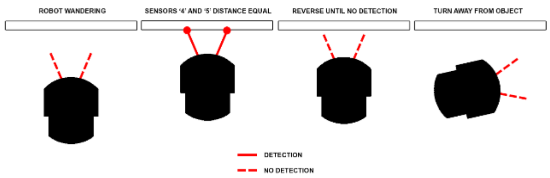


Figure 15: Avoidance behavioural strategy, reversing subsidiary state visualisation of the Pioneer P3-DX. Transition to the turning subsidiary state proceeding, is also visualised in the later development of the illustration.

Appendix M:

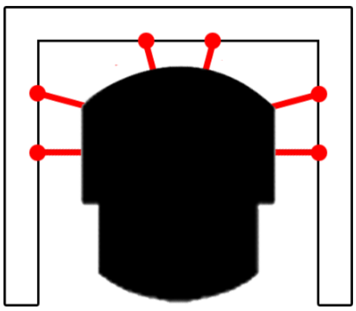


Figure 16: Avoidance behavioural strategy, stuck subsidiary state visualisation of the Pioneer P3-DX. Sensor detection exemplification, that would result in the invocation of the state's evasive behaviours.

Appendix N:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Figure 17: Rotation matrix notation [31], used to translate local object detections into global coordinate space. Said detections are applicated as two-dimensional coordinates, for the two-dimensional plane used to visualise environment charting, during and proceeding the robot undertaking the corresponding tasking (offline and online map variants).

Appendix O:

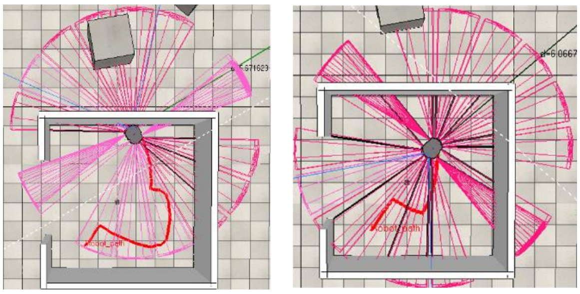


Figure 18: Finding the centre of the inner-most room tasking, before emulating a detection distance for the robots non-detecting side-most sensor (left illustration) and after emulating the robots non-detecting side most sensor (right illustration).

Appendix P:

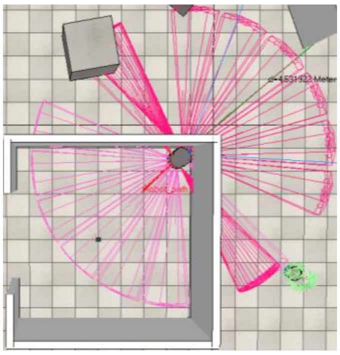


Figure 19: Finding the centre of the inner-most room tasking, robot colliding with one of four perpendicular arrangements (corner) of the rooms structure, resulting from side-sensor disputes and oscillatory motions exhibited, that led to being quashed and surfacing forward traversal patterns actuated by the robot.

Appendix Q:

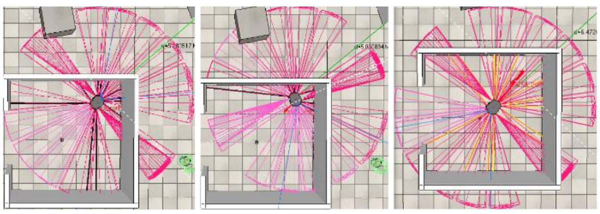


Figure 20: Finding the centre of the inner-most room tasking, robot handling perpendicular arrangements (corners) of the rooms structure, with an initial, opposing relation. Pivot-based evasive manoeuvres are actuated by the robot for its accomplishment.

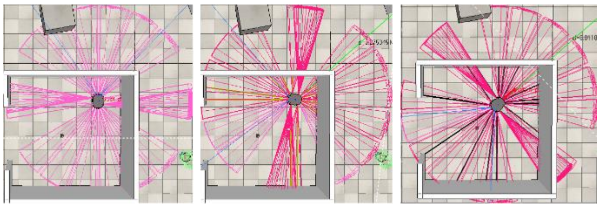


Figure 21: Finding the centre of the inner-most room tasking, robot handling perpendicular arrangements (corners) of the rooms structure, with an initial, adjacent relation. Pivot-based evasive manoeuvres are actuated by the robot for its accomplishment.

Appendix R:

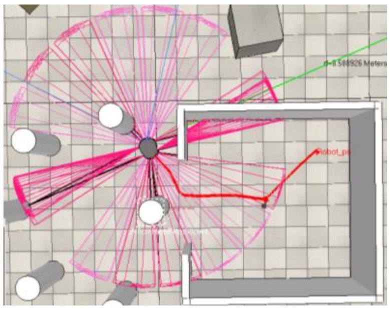


Figure 22: Exiting the inner-most room tasking, robot handling objects placed in the rooms exit space, for which it can appropriate an exit-alignment for, as well as an exit strategy that overcomes the obstacle evasively; if required to, for completing its task.

Appendix S:

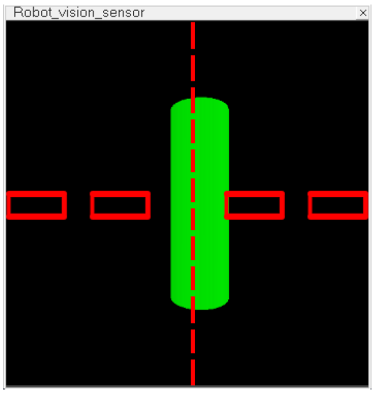


Figure 23: Locating the beacon tasking, floating view visualisation of the robot's visual sensor surveying the beacon object. 'Red-coloured' boxes representing the sampling filters for either side of the image, where the horizontal split is represented by the 'red-coloured' dashed, horizontal line.

Appendix T:

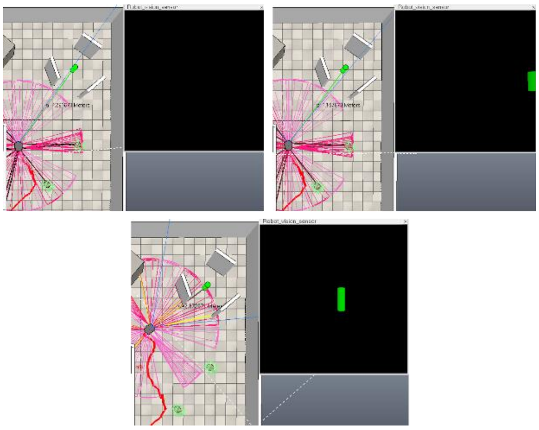
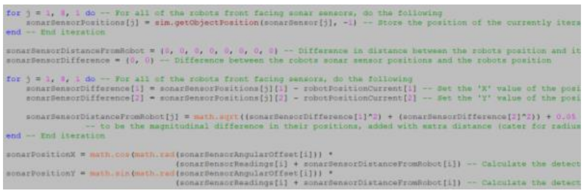


Figure 24: Locating the beacon tasking, floating view visualisation of the robot's visual sensor surveying the beacon object. Robot aligns right with the beacon object upon initialising the detecting the beacon within the right portion of its rendered image.

Appendix U:



```
for j = 1, 8, 1 do -- For all of the robots front facing sonar sensors, do the following
    sonarSensorPositions[j] = sim.getObjectPosition(sonarSensor[j], -1) -- Store the position of the currently itera
end -- End iteration

sonarSensorDistanceFromRobot = {0, 0, 0, 0, 0, 0, 0, 0} -- Difference in distance between the robots position and it
sonarSensorDifference = {0, 0} -- Difference between the robots sonar sensor positions and the robots position

for j = 1, 8, 1 do -- For all of the robots front facing sensors, do the following
    sonarSensorDifference[1] = sonarSensorPositions[j][1] - robotPositionCurrent[1] -- Set the 'X' value of the posi
    sonarSensorDifference[2] = sonarSensorPositions[j][2] - robotPositionCurrent[2] -- Set the 'Y' value of the posi

    sonarSensorDistanceFromRobot[j] = math.sqrt((sonarSensorDifference[1]^2) + (sonarSensorDifference[2]^2)) + 0.05
              -- to be the magnitudinal difference in their positions, added with extra distance (cater for radius
end -- End iteration

sonarPositionX = math.cos(math.rad(sonarSensorAngularOffset[i])) *
                          (sonarSensorReadings[i] + sonarSensorDistanceFromRobot[i]) -- Calculate the detect
sonarPositionY = math.sin(math.rad(sonarSensorAngularOffset[i])) *
                          (sonarSensorReadings[i] + sonarSensorDistanceFromRobot[i]) -- Calculate the detect
```

Figure 25: Environment charting tasking, preliminary calculations used for populating two-dimensional coordinates from the robot's object detections and its current position, in local space.

Appendix V:

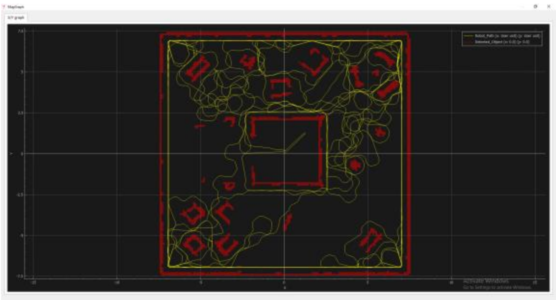


Figure 26: Environment charting tasking, visualisation of the online-based map variant, that applicates a graph object within the CoppeliaSim IDE and is used for illustrating the structural arrangement of the robots subjected environment, that it detects. As well as its exploration pattern during the simulation runtime routine (refer to the images legend).

Appendix W:

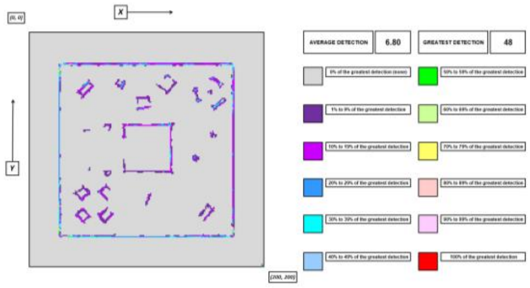


Figure 27: Environment charting tasking, visualisation of the primary offline-based map variant, representing the robots object detections over the course of its simulation routine. The map variant is compiled in Microsoft Excel, where the cells of the worksheet are conditional formatted by colour, to indicate positive detections and their recurrency in the routine submitted.

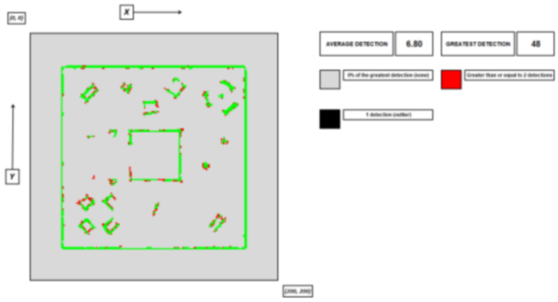


Figure 28: Environment charting tasking, visualisation of the secondary offline-based map variant, representing the robots object detections over the course of its simulation routine. The map variant is compiled in Microsoft Excel, where the cells of the worksheet are conditional formatted by colour, to indicate inlying and outlying detections.

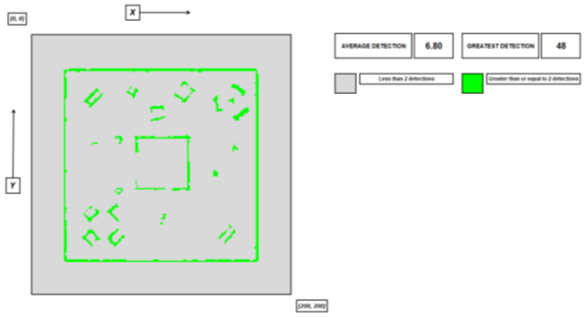


Figure 29: Environment charting tasking, visualisation of the tertiary offline-based map variant, representing the robots object detections over the course of its simulation routine. The map variant is compiled in Microsoft Excel, where the cells of the worksheet are conditional formatted by colour, to indicate inlying detections only.

Appendix X:

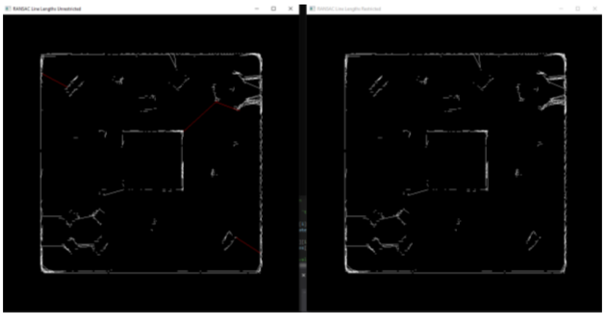


Figure 30: Environment charting tasking, RANSAC derived renditions of the robot's object detection data, visualised within separate SFML graphics windows. Left rendition visualises the non-validated RANSAC output, where outlying lines are depicted 'red'. Whereas right rendition visualises the validated RANSAC output, where only inlying lines are drawn.

Appendix Y:

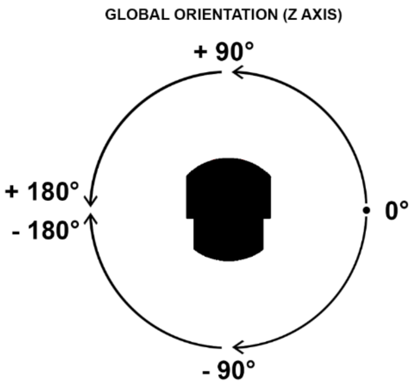
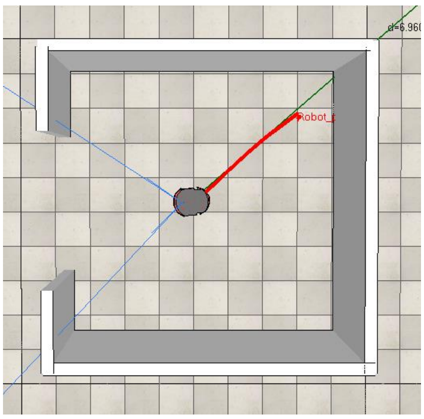
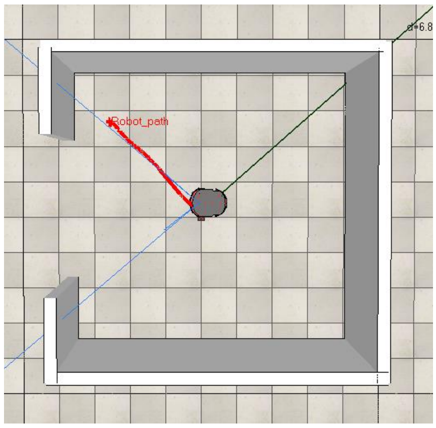


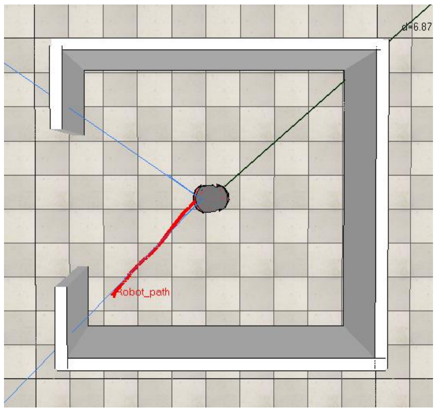
Figure 31: CoppeliaSim IDE, 'Z' dimensional orientation (heading) visualisation.

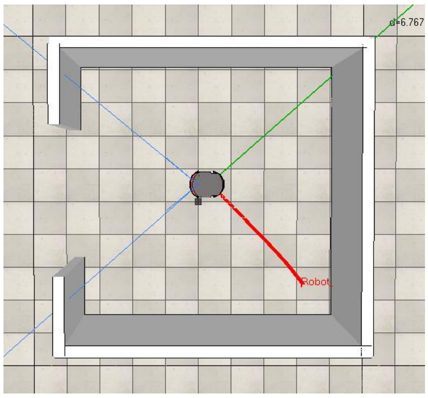
Appendix Z:

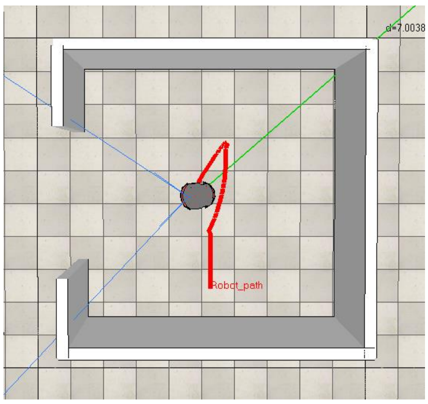
Successful scenarios











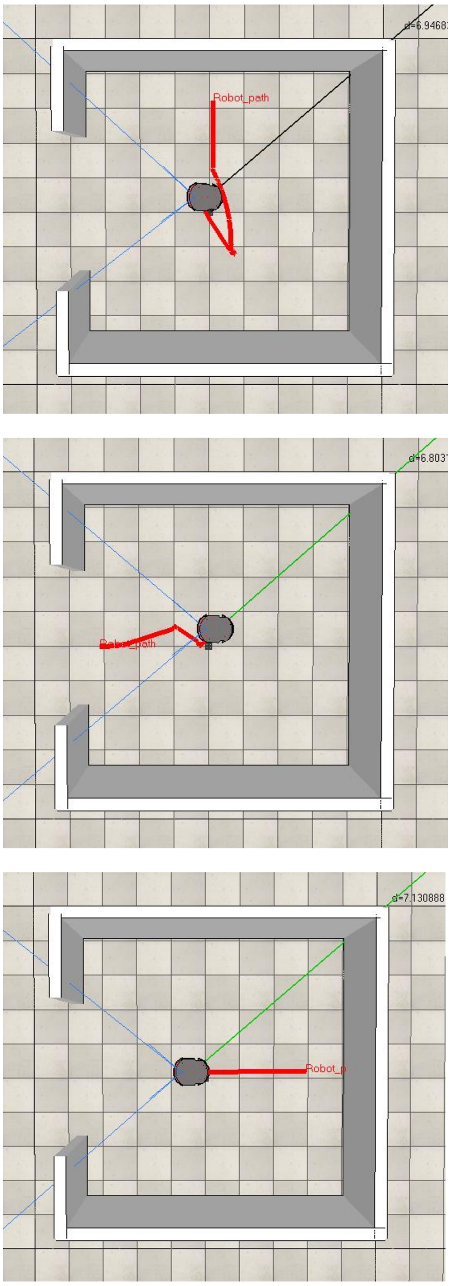
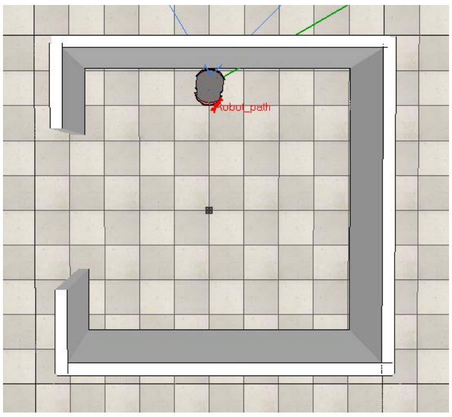


Figure 32: Locate the centre of the inner-most room tasking, successful test scenarios.

Unsuccessful scenarios



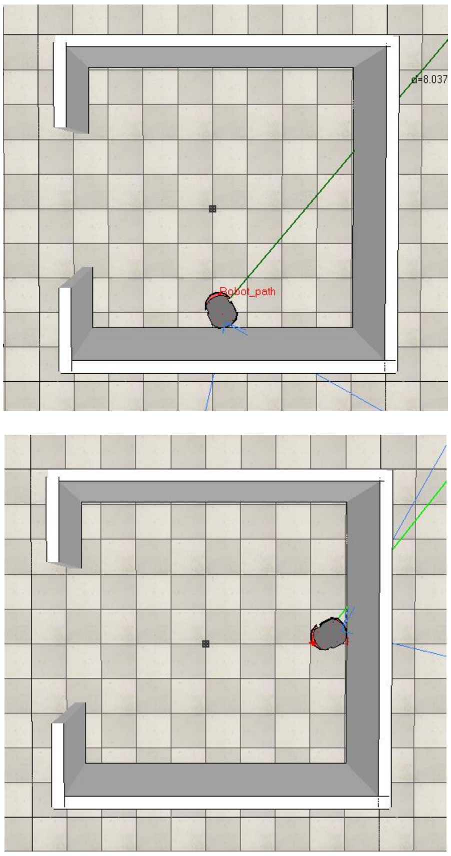
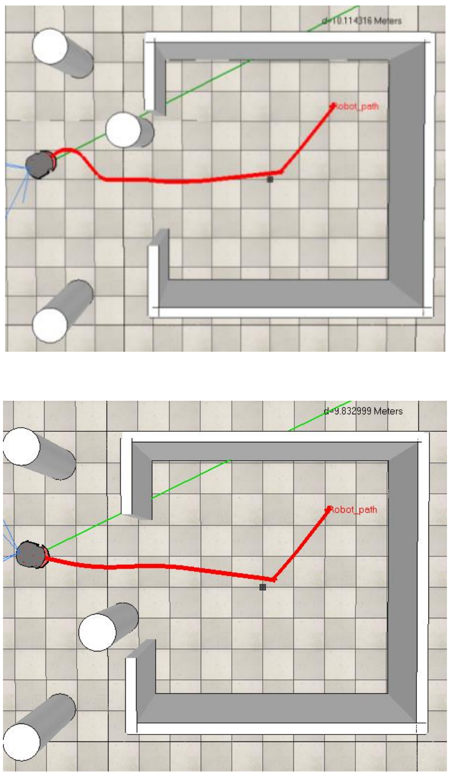


Figure 33: Locate the centre of the inner-most room tasking, unsuccessful test scenarios.

Appendix AA:

Successful scenarios



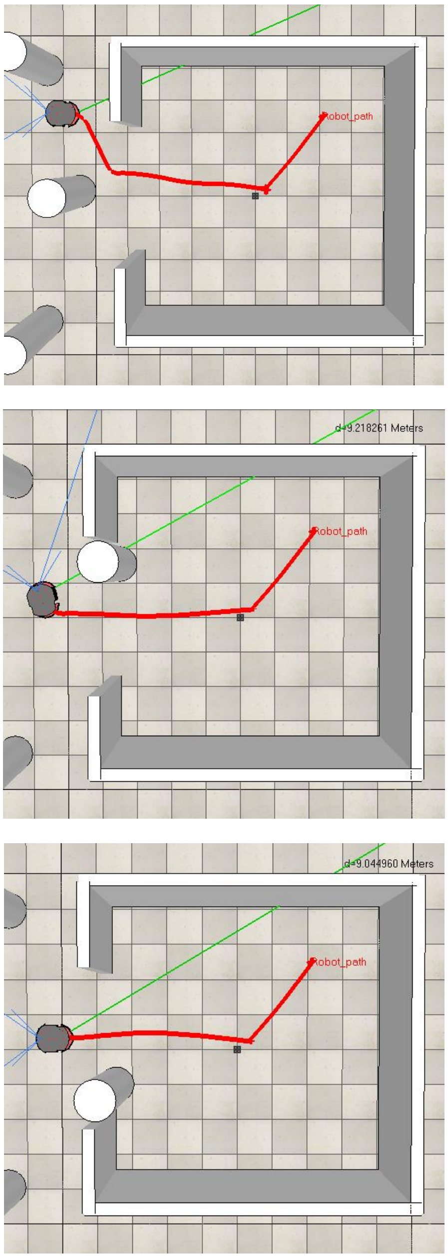
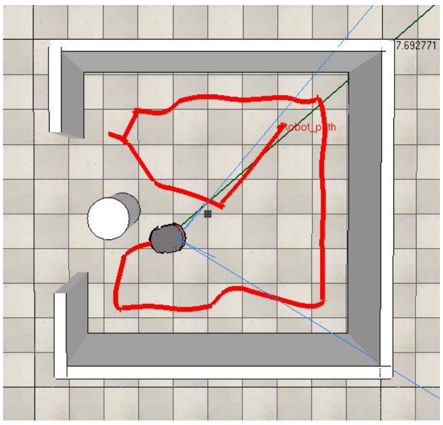


Figure 34: Exit the inner-most room tasking, successful test scenarios, obstacles factored.

Unsuccessful scenarios



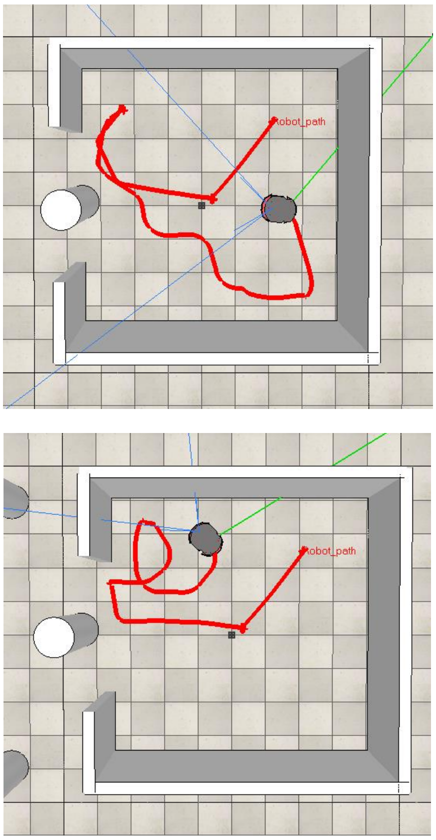
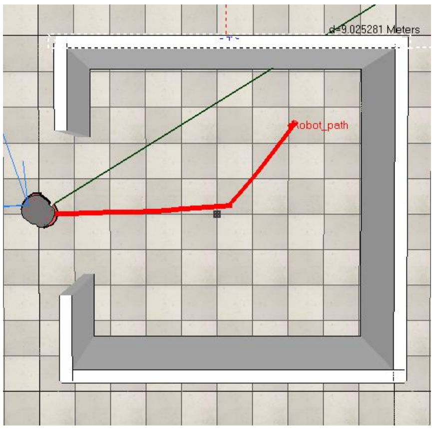
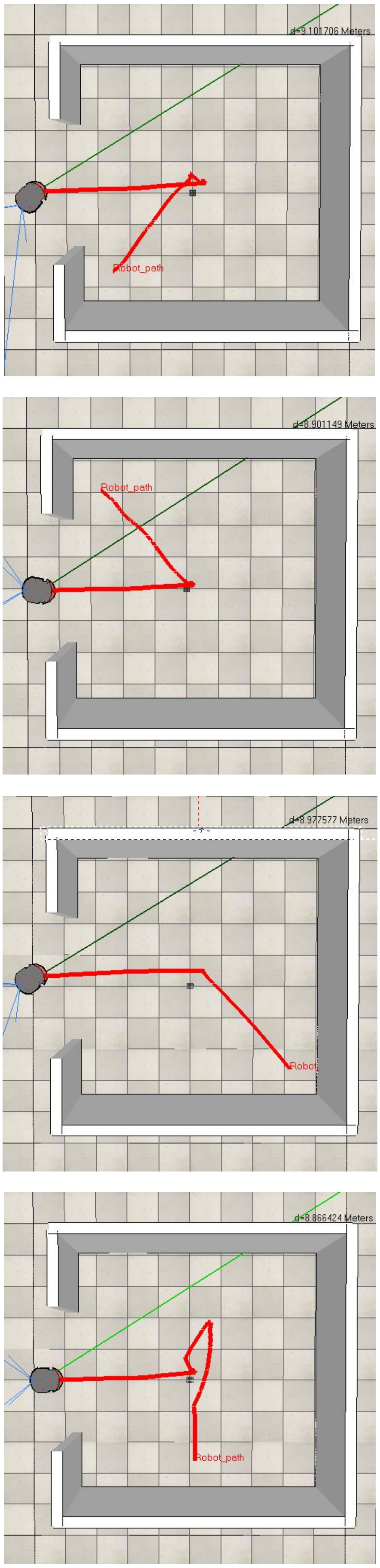


Figure 35: Exit the inner-most room tasking, unsuccessful test scenarios, obstacles factored.

Appendix AB:

Successful scenarios





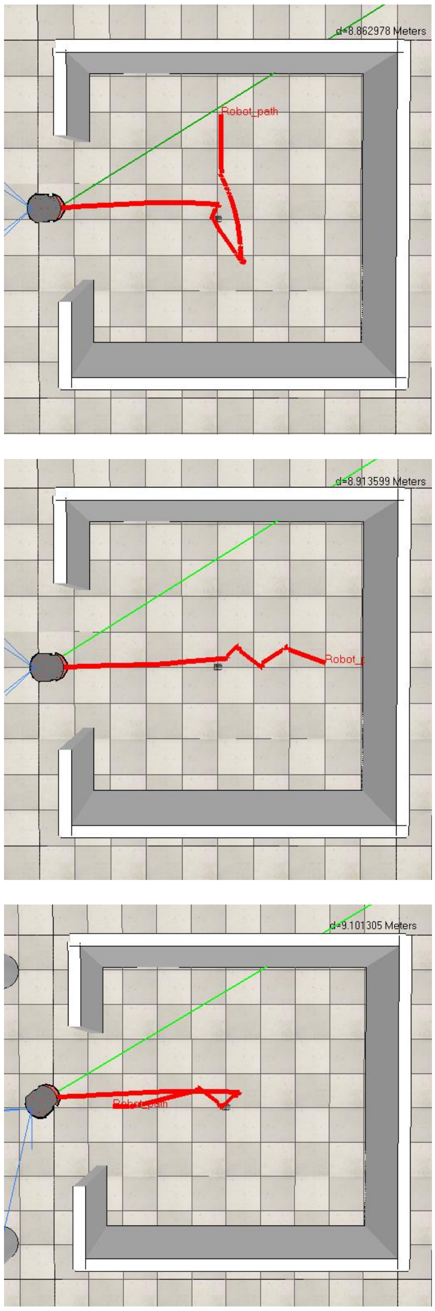
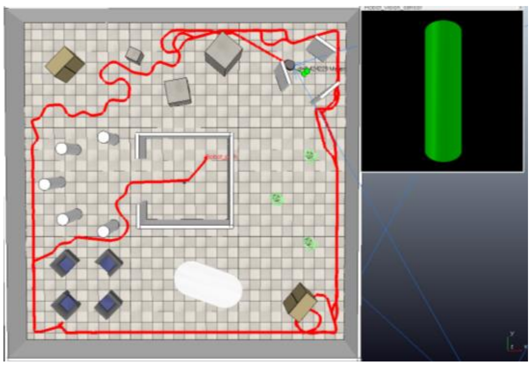
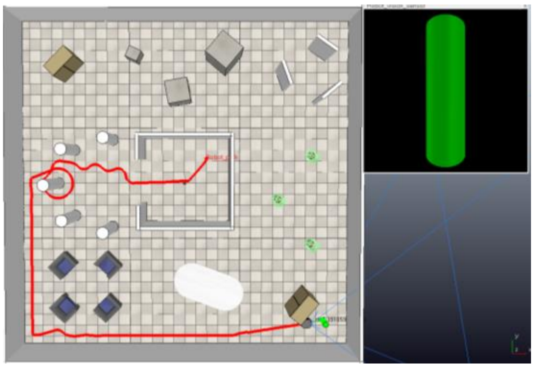


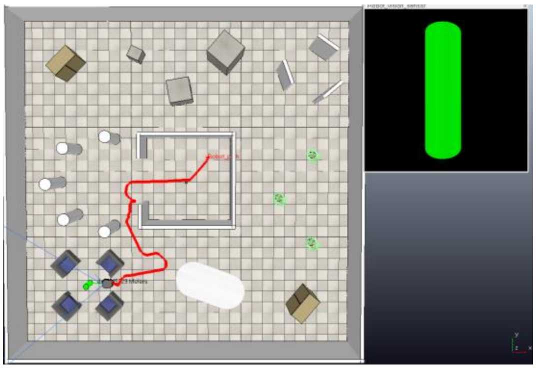
Figure 36: Exit the inner-most room tasking, successful test scenarios, obstacles not factored, varied start position.

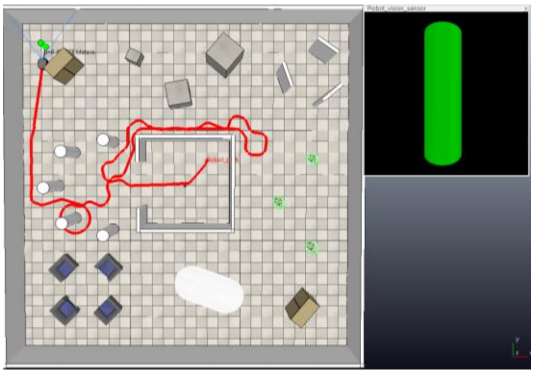
Appendix AC:

Successful scenarios









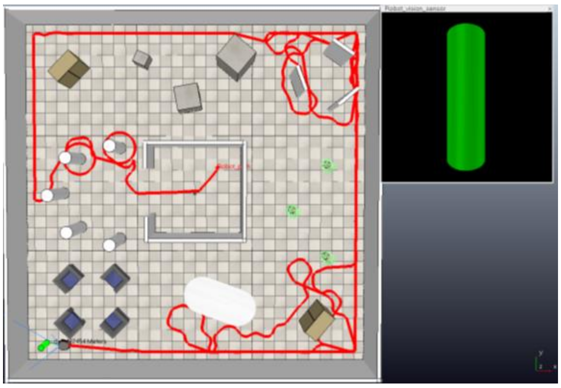
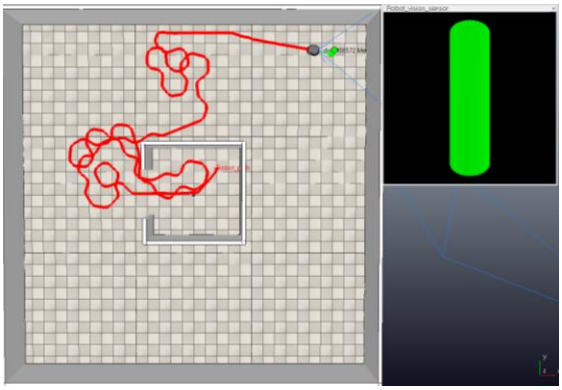
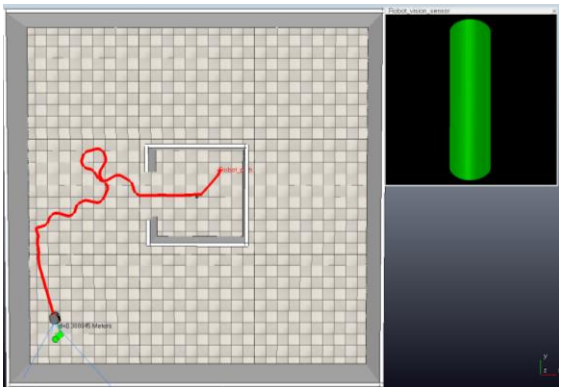


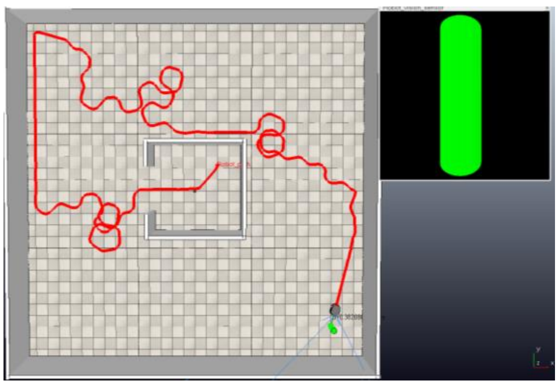
Figure 37: Locating the beacon tasking, successful test scenarios, varied beacon location.

Appendix AD:

Successful scenarios







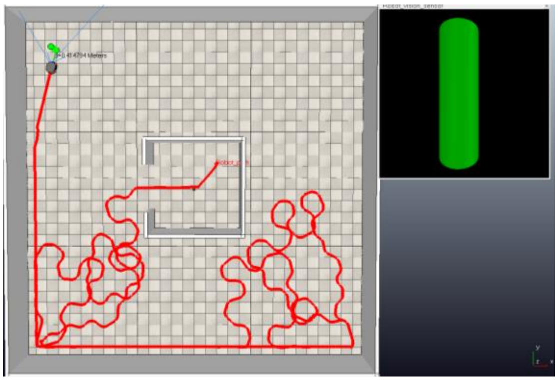
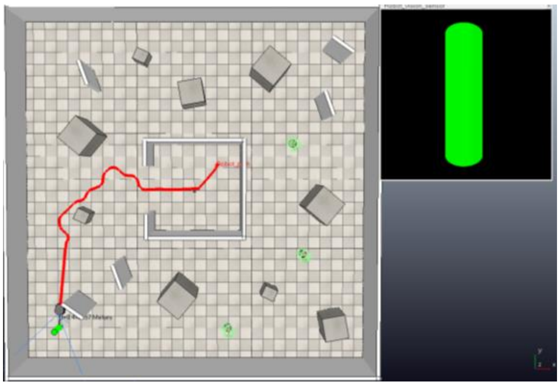
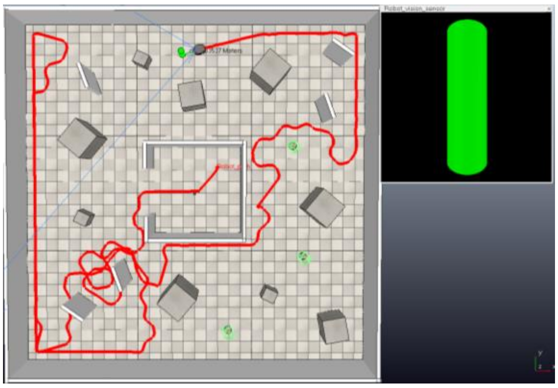
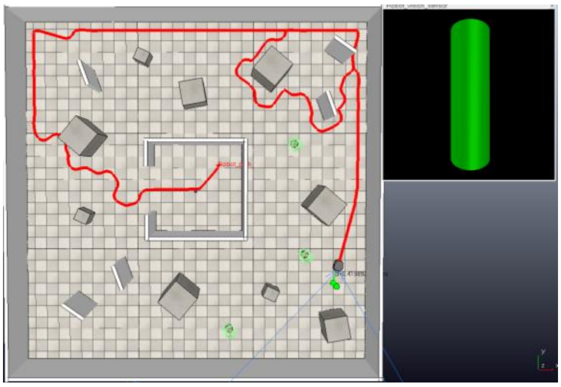


Figure 38: Locating the beacon tasking, successful test scenarios, varied beacon location, no obstacles.







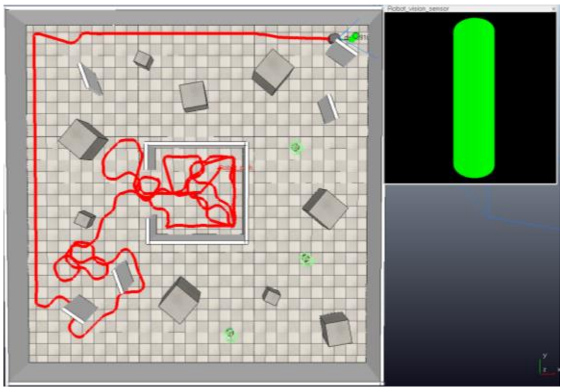


Figure 39: Locating the beacon tasking, successful test scenarios, varied beacon location, higher presence of obstacles.

Appendix AE:

Object placement in the environment

Figure 40: Environment charting tasking, robots subjected environment populated with objects.

Figure 41: Environment charting tasking, visualisation of the robots routing over the course of the simulation routine.

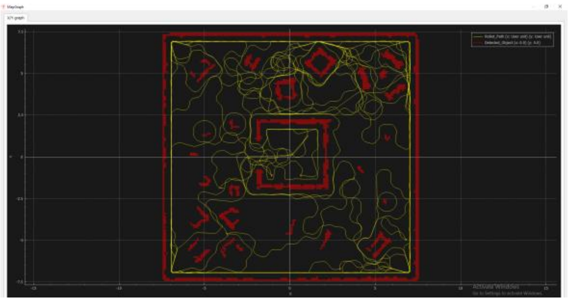


Figure 42: Environment charting tasking, online-based map rendition of the robots subjected environment and routing, concluding the simulation routine executed.

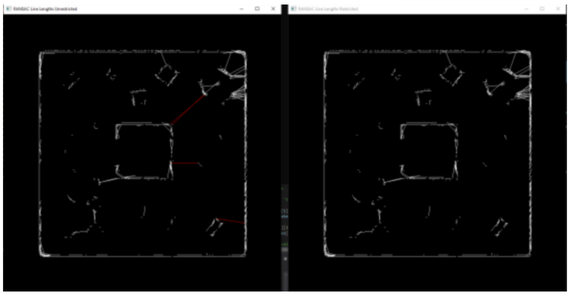


Figure 43: Environment charting tasking, RANSAC derived renditions of the robots subjected environment. Left illustration visualises non-validated RANSAC output, whereas the right illustration visualises validated RANSAC output.

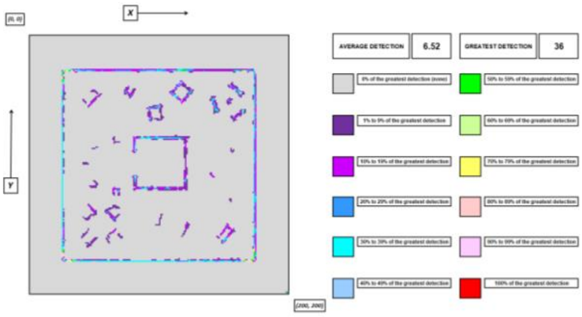


Figure 44: Environment charting tasking, offline-based map variant, visualisation of the robots subjected environment, illustrating the locations of objects and the counts of each part of an object detected throughout the course of the simulation routine.

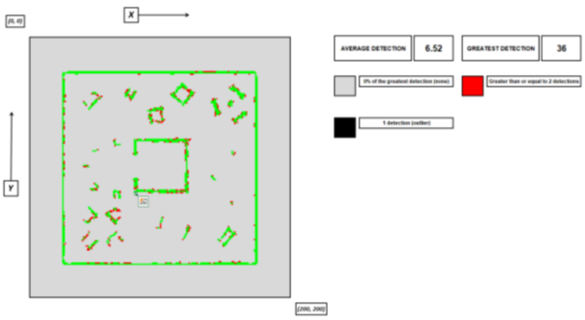


Figure 45: Environment charting tasking, offline-based map variant, visualisation of the robots subjected environment, illustrating the inlying and outlying object detections, gathered throughout the course of the simulation routine.

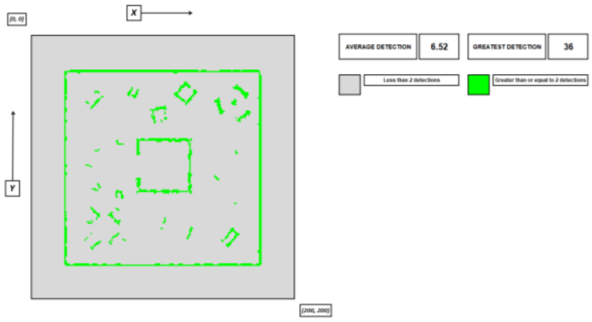


Figure 46: Environment charting tasking, offline-based map variant, visualisation of the robots subjected environment, only illustrating the inlying object detections gathered throughout the course of the simulation routine.

No object placement in the environment

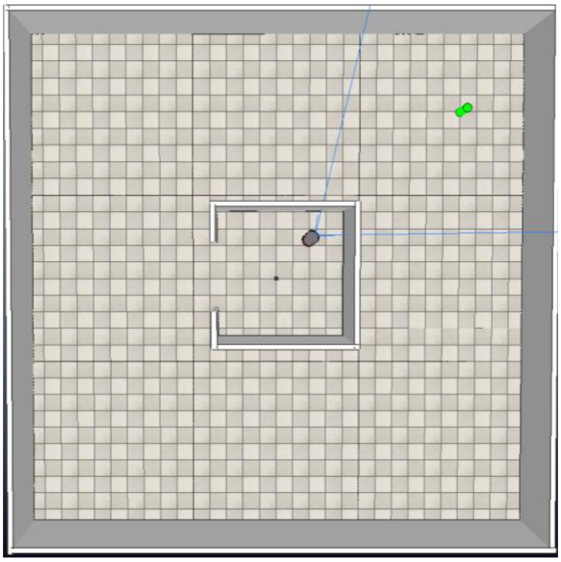


Figure 47: Environment charting tasking, robots subjected environment not populated with objects.

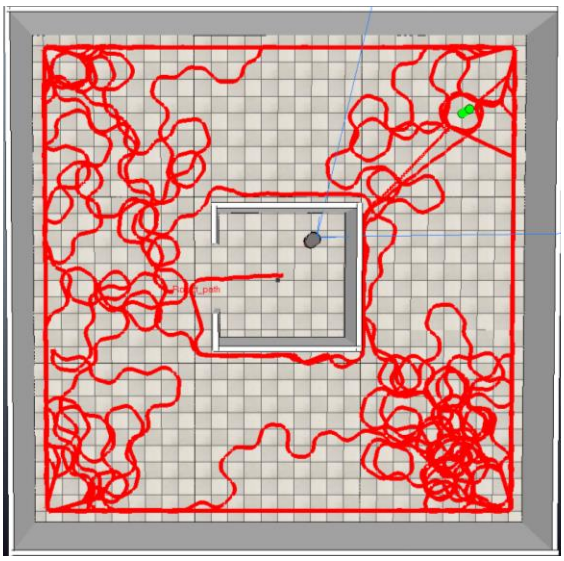


Figure 48: Environment charting tasking, visualisation of the robots routing over the course of the simulation routine.

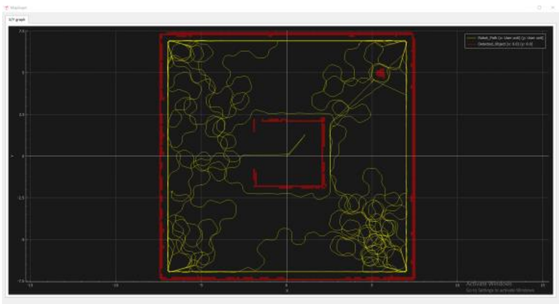


Figure 49: Environment charting tasking, online-based map rendition of the robots subjected environment and routing, concluding the simulation routine executed.

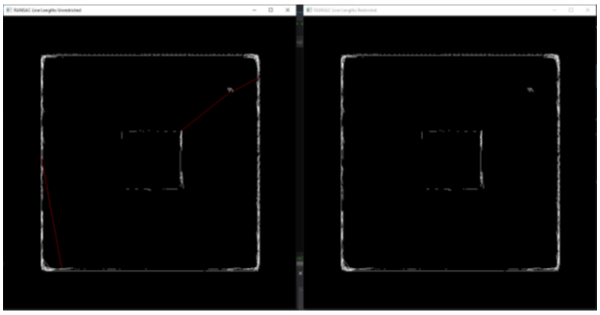


Figure 50: Environment charting tasking, RANSAC derived renditions of the robots subjected environment. Left illustration visualises non-validated RANSAC output, whereas the right illustration visualises validated RANSAC output.

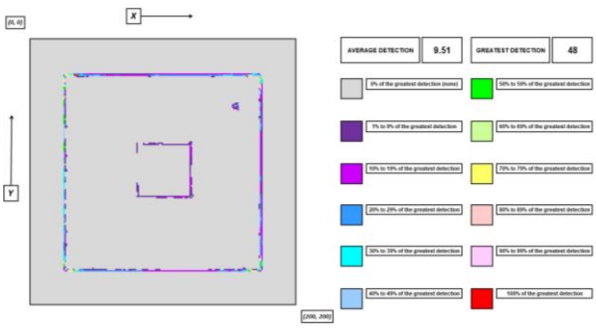


Figure 51: Environment charting tasking, offline-based map variant, visualisation of the robots subjected environment, illustrating the locations of objects and the counts of each part of an object detected throughout the course of the simulation routine.

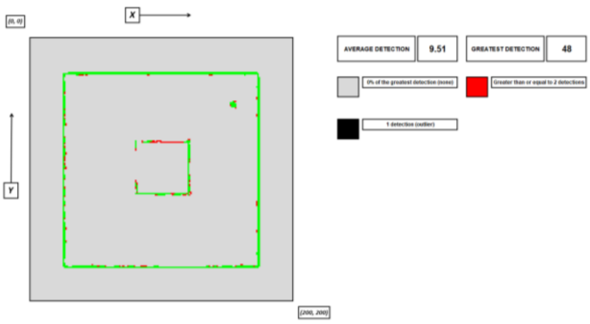


Figure 52: Figure 45: Environment charting tasking, offline-based map variant, visualisation of the robots subjected environment, illustrating the inlying and outlying object detections, gathered throughout the course of the simulation routine.

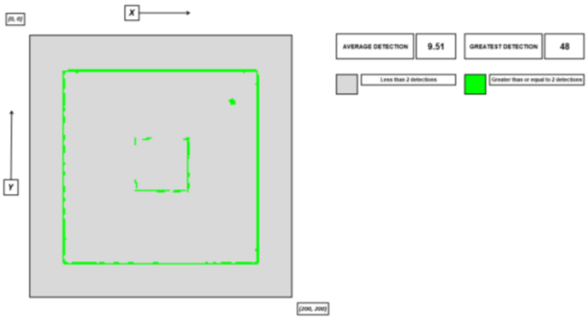


Figure 53: Environment charting tasking, offline-based map variant, visualisation of the robots subjected environment, only illustrating the inlying object detections gathered throughout the course of the simulation routine.

Appendix AF:

Object placement in the environment

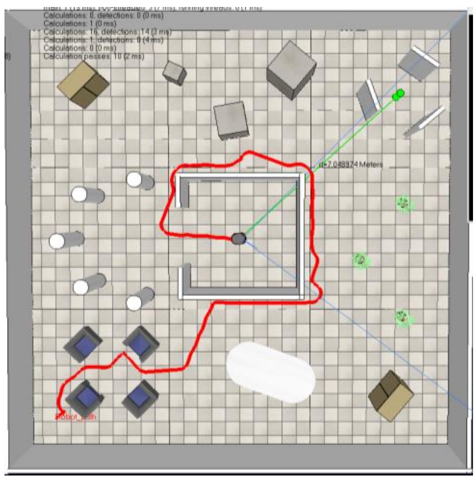
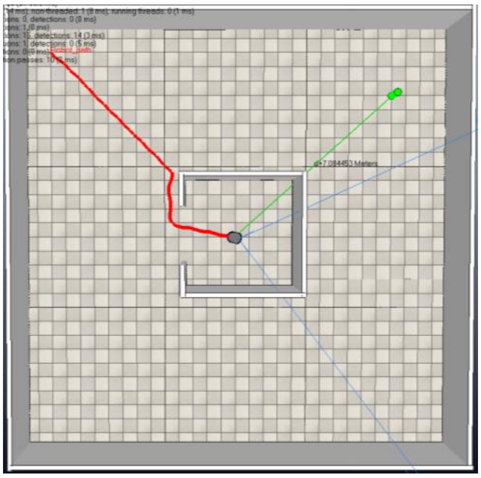
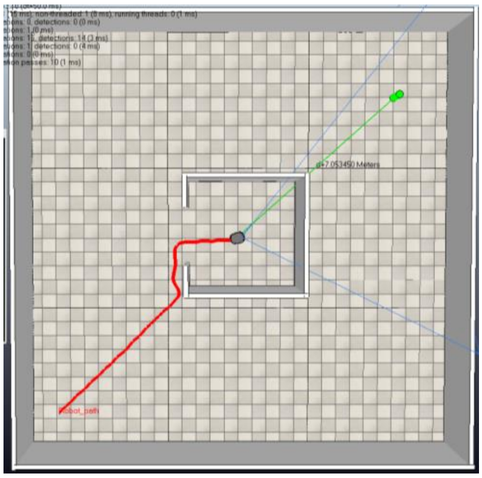
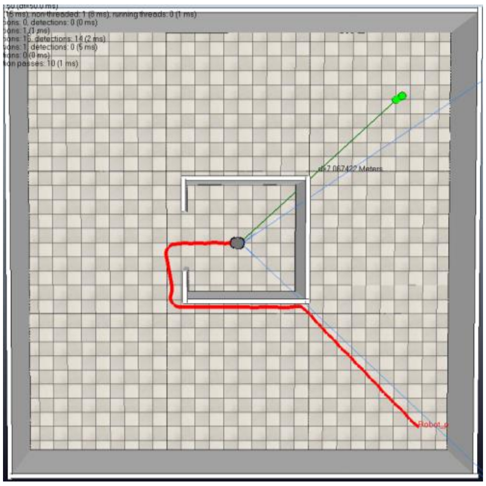


Figure 54: Return to the centre of the inner-most room tasking, visualisation of the robot returning to the centre of the inner-most room, environment populated with objects.

No object placement in the environment







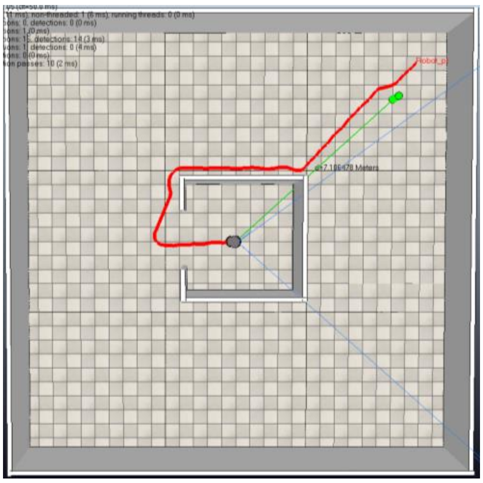


Figure 55: Return to the centre of the inner-most room tasking, visualisation of the robot returning to the centre of the inner-most room, environment not populated with objects.