

METHODS AND SYSTEM DESIGN FOR INDOOR NAVIGATION OF AN AUTONOMOUS ROBOT WITH LOCALIZATION USING SEMANTIC MAPS AND THE USE THEREOF

BACKGROUND

Field of the Invention:

This invention relates to the field of navigation and localization strategies for an autonomous robot deployed in a structured indoor environment, for example a hotel, hospital, or other enterprise buildings.

Description of the Related Art:

- Current methods for indoor navigation and localization of an autonomous robot primarily rely on depth or image sensors to create a map of the environment, followed by matching and sensor fusion to determine the robot's location within this map.
- One method is to create occupancy grid map of the environment at a certain distance resolution. This method may either involve simultaneous localization and mapping (SLAM) or localization using a global map built only once. In another implementation of this popular method, a global occupancy grid map is created to represent mostly the static elements of the environment. A local occupancy grid map is then used along with the global map to register the presence of dynamic obstacles in the environment. Localization using these maps is performed through scan matching in some derived feature space of the environment. The scan can be features derived from a RGB image like SIFT or SURF, planar lidar data, lidar data with varying field of views (FOVs), or in recent implementations, data from a low cost 3D sensor like Kinect, PrimeSense or RealSense.
- However, a problem with such implementations is the need for constant scan matching and map data maintenance. Certain indoor environments like a hospital, university or enterprise, have highly structured environment as compared to an outdoor environment or other indoor environments. The structure of the environment can be used for localization by data collected through cheap sensors like a low or mid range laser distance sensor.
- Hence, there is a need to develop robust and universal navigation and localization algorithms that can be scaled for any robot in an indoor environment. Moreover, this setup should use the minimum possible sensors and computing units for a viable and profitable commercial solution to robotics based solutions deployed in indoor environments like hotels, hospitals or other enterprise buildings.

SUMMARY

- The present invention provides methods and system design for navigation and localization of an autonomous service robot deployed in indoor environments. Indoor environments have the property of being highly symmetric across floors and have certain landmarks that can be used by various sensor modalities to distinctly identify and hence localize themselves.
- In any embodiment of the present invention, the robots follow a semantic navigation strategy where a set of policies are followed in appropriate steps triggered by certain events raised by the robot. A policy is any of the following three instructions provided to the robot: forward, left or right. Under any given policy, the robot manipulates the velocity of each wheel in order to successfully complete the policy while also avoiding any obstacles at the same time.
- In any embodiment of the present invention, the robots define an action zone around themselves based on some mathematical formulation relevant to the indoor environment. Action zone readings of the sensors used in the robot is used to maneuver the robot around obstacles and also facilitates policy following.
- In one embodiment of the present invention, the action zone is defined as an ellipsoid with major axis in front of the robot and minor axis on the robot sides. The ellipsoid can be realized through the use of a distance ranging device, for instance a lidar. This ellipsoid based action zone is used to define the arc and radius at which the robot must navigate in order to avoid all the obstacles within the zone.
- In any embodiment of the present invention, the robot keeps track of the number of intersections registered by the sensors. Intersections are defined as regions of the indoor environment having two almost perpendicular planes of areas greater than a particular defined threshold. Almost perpendicular planes can be calculated by evaluating the dot product or other mathematical methods. The robot uses passive landmarks like room numbers or through RFIDs, a better estimate of its location within the indoor environment when needed.
- In an embodiment of the present invention, the robot has two distinct modules, one each for navigation and user interaction. The modules can establish a successful communication link between themselves through wireless methods or wired connections.
- In any embodiment of the present invention, a multiplicity of robotic service agents can be deployed through an API, allowing monitoring and control of their locations, service status, and path correction through manual mode by a remote operator upon receipt of complete navigation failure flag.
- This Summary was provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

- FIG. 1A shows the top view of the robot's navigation module and sensor placements.
- FIG. 1B shows the side view of the robot with the navigation module and other sensors for user interaction.
- FIG. 2 is a diagrammatic view of the cross section of an indoor environment where the robot must navigate.
- FIG. 3 logical flow that describes the operation of the application framework running in the navigation module.
- FIG. 4 is a diagrammatic description of the functionality of the navigation unit when an intersection is detected and the system needs to change policies.
- FIG. 5A is a diagrammatic description of the action zone defined by the robot based on some mathematical formulation.
- FIG. 5B shows how a robot uses the action zone to calculate local trajectory information.
- FIG. 6A is a diagrammatic description of the manner of decision taking when an object spanning both the regions around the robot, like a wall is encountered.
- FIG. 6B is a diagrammatic description of the manner of decision taking when an object spanning both the regions and another obstacle is present.
- FIG. 7A and FIG. 7B are descriptions of manner of decision making in scenarios when objects are present at certain angle with respect to the robot.
- FIG. 8 is a diagrammatic description of a situation where a robot has been instructed to reach a particular location.
- FIG. 9 shows the various services offered by the navigation API.

- FIG. 10 is a drawing of the robot design and sensor placement.

DETAILED DESCRIPTION

- [001]** For the purposes of providing an understanding of the principles of the invention and the scope of applications involved therein, reference will now be made to the embodiments illustrated in the drawings and specific language will be used to describe the same. It will nevertheless be understood that no limitation of the scope is thereby intended. Any alterations and further modifications in the described embodiments, and any further applications of the principles as described herein are contemplated as would normally occur to one skilled in the art.
- [002]** In the following description, circuits and functions may be shown in block diagram form in order not to obscure the present invention in unnecessary detail. Conversely, specific circuit implementations shown and described are exemplary only and should not be construed as the only way to implement the present invention unless specified otherwise herein. Additionally, block definitions and partitioning of logic between various blocks is exemplary of a specific implementation. It will be readily apparent to one of ordinary skill in the art that the present invention may be practiced by numerous other partitioning solutions. For the most part, details concerning timing considerations and the like have been omitted where such details are not necessary to obtain a complete understanding of the present invention and are within the abilities of persons of ordinary skill in the relevant art.
- [003]** The invention describes methods and system design for an autonomous service robot for control, navigation, and localization in an indoor environment. The definition of indoor environment encapsulates any enterprise setup with structured environment, for instance hotels, hospitals, or university buildings. Specifically, the invention directs a robot to utilize the presence of certain structural parameters in the environment to navigate and localize itself.
- [004]** FIG. 1A shows the top view of the robot's navigation system module developed as a part of the present invention. In one embodiment, the robot consists of a circular base with two differentially driven wheels 106A and 106B. A motor controller 102 is the circuit responsible for executing the various maneuvers of the robot through control of

the rotational velocity of each wheel. Additionally, encoders may be used on the wheels to maintain an estimate of the amount of rotation of each wheel. The wheels may have diameter as: 10cm, 15cm, 20cm, 25cm, or 30cm as required. A computing unit 100 serves as the main computer for collecting information from various sensor modalities and performing calculations relevant to the functionalities of the robot. The navigation unit is also equipped with at least 3 distance sensors. In one embodiment of the present invention, the navigation module has 5 sound based distance sensors with a 45-degree angular spread 104. The number of such sound based sensors can vary depending on their spread angle. The direction of sensor placement as such defines the front facing direction of the robot. The computing unit samples data from the ultrasonic sensors ten times in a second or at a higher frequency. The distance sensor readings can be used by the navigation module to bring the robot to a dead-stop if needed or perform appropriate sensor fusion and path calculation steps to circumvent the obstacle. The distance sensors can be connected to the computing unit through a wired or wireless protocol.

- [005]** In one embodiment of the present invention, the distance sensors can also be placed at the back-facing direction of the robot in order to avoid any sudden dynamic obstacle collision. The distance sensors are placed primarily at a low height so as to avoid small obstacles also. The height of these sensors from the ground base level can vary as 5cm, 10cm, 15cm or 20cm. In other embodiments, the distance sensors can also be oriented at other angles with respect to the ground plane.
- [006]** In one embodiment, the motor controller 102 can communicate with the computing board 100 with a bi-directional serial connection interface where the computing unit sends commands to the motor controller for execution. A successful execution program flag may be indicated upon completion of tasks as directed.
- [007]** In another embodiment, the motor controller 102 can communicate with the computing board 100 through wireless communication protocols including but not limited to local WiFi, Bluetooth, ZigBee, MQTT, websockets, socket.io, or radio. Wireless interaction methods can make the system increasingly modular though also at the same time, increasing the complexity of the setup.
- [008]** In most implementations of the navigation module, the robot has at least one caster wheel to support the robot's weight. In other implementation of the system, the robot can have three omnidirectional wheels oriented at 60-degrees apart. In yet another implementation of the system, the robot can have four motorized wheels, each wheel being independently controlled or maintained as a setup where two back wheels are

connected through a cable to follow each of the front wheels. In any case, the degree of freedom of robot's movement can be decided based on the tradeoff between power requirement of the resulting system and the profile obtained.

[009] FIG. 1B shows the side view of the entire robot system including the navigation module 114. The view shows another distance sensor like a laser based device 108, for providing a 360-degree data about the robot's surroundings. In one embodiment, the laser sensor provides planar data instead of a full 3D or some field of view of the environment. The laser device can be mounted at an appropriate height from the robot's base at 10cm, 15cm, 20cm, 25cm, 30cm, or 35cm. Data from the laser sensor is sampled by the computing unit at a rate of 10Hz or more, as desired.

[0010] In other embodiments, the lidar can be replaced by an equivalent depth sensor that is infrared based, stereo camera image based, RGB-D camera, and/or others. Data from these sensors can be plane filtered to generate lidar like readings for processing. An additional computing unit may be installed in some embodiments to allow dedicated processing of such data.

[0011] In most embodiments, the laser device 108, is placed parallel to the ground plane. In other embodiments, this angle may be varied to obtain data from other planes of the environment. In all embodiments of the present invention, data from various distance sensors 108 and 104 are updated asynchronously by program threads on appropriate program variables. These program variables are used by other threads for data processing.

[0012] In any embodiment of the present invention, a touch screen display 112 is provided. This system can be used for user interaction, voice, or information display purposes. The screen can be connected to the computing board on the navigation module through wireless or wired methods to facilitate communications.

[0013] Another depth sensor 110 may also be provided in some embodiments for path planning and navigation. This depth sensor can be another lidar, stereo based, infra-red based, monocular camera, or RGB-D camera. In case of more than one distance sensors as in the presence of a laser based distance sensing unit, ultrasonic sensors and depth sensor 110, the algorithm assigns a priority flag to each modality whose readings take priority over others for path planning and navigation.

[0014] FIG. 2 shows a diagrammatic description of the problem statement. A robot 124 as described in FIG. 1 is present at the location as depicted in the figure. The indoor environment can have various structural elements like a closed intersection 128, an

open intersection 126, an obstacle 120, and a curved path 130. In a particular situation, the robot is required to navigate through the curved path to a further destination zone. Moreover, the robot must simultaneously localize itself in this process in order to reach the desired destination zone.

[0015] The complexity of the above setup is understood from the fact that the robot must reach its destination while making decisions based on the semantic map provided. Zones such as 126 and 128 represent points at which the robot must make a decision in order to complete the task assigned. Moreover, dynamic obstacles like 120 might have motion of their own, have various shapes and size, and must be avoided accordingly.

[0016] The present invention utilizes the count of the number of intersections to understand a robot's current location with respect to its starting location. Readings from wheel encoders are also used to maintain a partially accurate estimate of the robot's location. The navigation algorithm is essentially partially blind as the robot starts from an initial position and propels itself in a certain direction using encoder readings.

[0017] At any time, the robot can be provided with a semantic map that contains instructions for the robot on how it should move and what is expected in the future. The term semantic is understood to refer to instructions as: forward, left or right.

[0018] A semantic map of the indoor environment can be constructed from its floor plan by using an image-processing algorithm provided by this invention. Semantic map creation involves extraction of intersections from the scaled floor plan image. The semantic map is stored in appropriate format such as XML file, text file, array, and/or other formats. Intersections are defined as regions in the environment where a wall perpendicular to another wall is found. This definition arises from the fact that in structured environments as those where the invention is meant to be deployed, intersections are always present. Such structural elements are important as the robot can reach its desired destination by traversing such paths.

[0019] The presence of clutter, for example people, may pose a problem if intersections are unable to be detected by the robot. As a measure to prevent such misses, the algorithm resets an estimated time of next intersection's expectation variable each time an intersection is encountered. Following overflow of this pre-defined time variable, the robot may enter path recovery mode that involves it turning 180-degrees and traversing back in the environment to find the missed intersection. Following failure to localize after path recovery mode instantiation, the robot may choose to stop and enter manual mode where a remote operator is required in part to orient the robot in a particular

direction with respect to its current location and update its semantic map provided earlier.

[0020] FIG. 3 shows a logical process flow when the robot is desired to reach a particular destination within the indoor environment. The robot receives destination information at step 140 from a remote central management operator within the indoor environment. Destination information may itself be in terms of the semantic map generated previously or in a language understood by the robot's algorithm and parsed into its semantic language. Following receipt of destination, the robot generates a semantic map of the possible paths that can lead to the desired destination at step 142. The generation of multiple paths is useful as the robot may not need to enter path recovery mode frequently. Moreover, since the indoor environment is itself represented in a semantic form, it can be queried by the robot at any time to reform its current navigation strategy.

[0021] In most implementations, each robot deployed in the indoor environment stores locally, a copy of the semantic map. This permits it to perform computations upon receipt of instructions to go towards a certain destination. In another implementation, the robots receive final semantic policies to reach their destination that are computed remotely on a central management server on the cloud.

[0022] Following this step, the navigation module is initiated at step 144 where the robot is prepared to deliver appropriate items, if any, to the desired destination. In most implementations of the robot service setup, the robot may be needed to move in a hall or its equivalent structure towards a lift in order to reach specific floors. This may be achieved by sending the robot in a particular hard coded direction followed by a strict policy traversal to reach the lift. In other embodiments, the robot can follow certain infrared spectrum visible markers to localize itself with respect to the lift system and move towards it. In other embodiments, the robot may use certain visual features derived from distinct parts of the hall (or its equivalent), to localize itself with respect to the lift system. Following successful lift exit, the semantic algorithms of the navigation module are utilized to help the robot navigate in the environment while avoiding any static or dynamic obstacles encountered, at step 146.

[0023] After every intersection observed by the sensors of the robot at decision step 148, the robot turns to its provided semantic map to acquire the next policy to be followed. This policy represents whether the robot must go straight or turn at the particular intersection. In most embodiments of the present invention, the intersection registration

and detection process is an asynchronous thread of the navigation stack running on the robot.

[0024] In other embodiments, the intersection registration and detection process is performed as a time looped synchronous step within the main navigation module program. In case of intersection detection, the robot updates its location update in the system and queries the semantic map for the next policy at step 150. After this query execution, at step 152, the result of 150 is set as the current policy the robot must follow until the next decision point, that is, another intersection.

[0025] Another thread of the main process checks whether the robot has reached its desired destination at step 154. If the result of this decision step is true at any time, the robot stops and performs appropriate functions as relevant to its particular indoor environment's deployment purpose; else, the navigation module keeps repeating the steps as defined in the logic flow.

[0026] In certain embodiments of the present invention, the robot may need to localize on a greater accuracy detail than obtained through the semantic maps. For instance, for a robot deployed in a hotel setup, the robot may need to localize itself with respect to the doors on each corridor. This may be achieved through use of optical character recognition methods to read specific floor and room numbers, color based segmentation of markers, infrared based segmentation of markers, RFID tags, Bluetooth tags, and/or others.

[0027] Turning now to FIG. 4, a robot's policy following method based on the semantic maps is shown. A process of the navigation module checks asynchronously for the presence of intersections as depicted in the figure. Intersections are detected by the presence of almost perpendicular lines or areas in the scan data of depth sensor of the navigation module. Almost perpendicular planes or lines are identified after plane/line filtering of scan data followed by their dot product calculation if the area/length of the plane/line is within defined thresholds. By this definition, the robot is able to register the presence of upcoming intersections before their occurrence and perform its path planning accordingly.

[0028] At one particular topology of an intersection, the robot has three options: follow a forward policy 160, follow a left policy 164, or follow a right policy 162. At this step, the navigation module queries the semantic map provided for the next policy. Consequently, the robot changes its current policy to one of the above three.

[0029] In case when the robot misses the detection of any intersection in the indoor environment due to presence of dynamic clutter, the robot becomes aware of this state through the error raised by a time to next intersection flag. Following this, the robot may choose to use certain environment features to localize itself and reform semantic traversal policies for destination, or raise a flag for remote operator on the central management system to guide and indicate its current location. The remote operator can be a human or a program on the central management server.

[0030] Turning now to FIG. 5A, the obstacle avoidance and navigation method is described. The robot defines an area of action 178, called the action zone around itself, based on some mathematical formulation derived from the indoor environment. In one embodiment of the invention, the action zone is defined on the front facing part of the robot. The zone has an ellipsoid shape governed by its major axis 174 and minor axis 176. The major axis 174 is defined on the front facing side of the robot while the minor axis is defined on robot sides. The action zone is further divided into left zone 170 and right zone 172 identifying the left and right sides of the robot. The values for any shape of the action zone may be pre-calculated on algorithm initialization in order to speed up further calculations. The major axis in a typical setup can be set as 1000mm, 1500mm, or 2000mm while the minor axis can be set as 400mm, 500mm, or 600mm depending on the robot's size and the corridor/path space available. Data from the action zone is constantly used by the navigation module of the robot to actively avoid obstacles and move in a smooth manner.

[0031] In another implementation, the action zone 178 is defined as a cone with a certain radius and height parameter on the front of the robot. In yet another implementation, the action zone may be defined as a circle of a certain radius. The shape of action zone may also represent contours of similar risk areas of the robot. A circular implementation of the action zone is more relevant to an omnidirectional robot platform embodiment with 36-degrees of movement freedom.

[0032] As shown in FIG. 5B, the robot uses the defined action zone to identify potential obstacles 120. In one implementation, the navigation module may maintain an estimate of the heading of the robot or the amount of turn a particular obstacle may cause. This estimate can be used to limit the amount of turn the robot can take so as to maintain a more strict policy following paradigm.

[0033] Turning now to FIG. 6A, a scenario is depicted where the robot while following a certain policy encounters a wall obstacle 190. In this case, the navigation module

calculates an envelope of the obstacles encountered. In one embodiment of the present invention, the envelope is calculated in the following manner as described. Assuming the action zone to be an ellipse, with 0-degrees as the major axis and counting counter-clockwise for a 360-degree rotation, the navigation module finds the minimum of readings in the left zone, represented as θ 198. Note that this minimum is the first minimum of readings ordered through angles from 90-degree to 0-degree. Similarly, the minimum of right zone is the first minimum of readings ordered from 270-degree to 359-degree. After the above operations, minima of both zones are obtained. Using these two values, a line is fit and its distance from robot's major axis 196 is obtained. In the case as depicted in FIG. 6A, the minimum of left zone is at an angle of θ , while the minimum of right zone is at an angle of 0. Hence the envelope is same as the straight wall and its angle with respect to the robot is the angle of the wall itself.

[0034] In another implementation of the above calculation process, the envelope can be found by fitting a second, third or higher degree polynomial, splines, and/or other methods instead of using a straight line. In this case, the envelope may represent the curvature of obstacles and their distribution in the environment.

[0035] FIG. 6B depicts a scenario where another obstacle 192 in the right zone is present along with a wall obstacle. The robot must navigate in a manner as to avoid the wall as well as the dynamic obstacle 192. By following the above-mentioned algorithm, the minimum of left zone is θ , 200, and minimum of right zone is the radial distance of the obstacle itself from the robot ϕ , 202. By taking the envelope as the straight line joining these two points, the robot can effectively avoid both these obstacles. The distance of robot from this envelope is taken as the line 196. Following calculation of distance 196, the radius of turn is defined as its half, the angle of turn is defined by the slope value of the line hence obtained (line 196). In some embodiments, the robot may perform this maneuver for a defined period of time or until the next obstacle happens. Additionally the robot may perform these maneuvers at a certain fixed differential velocity of the wheels or at any unknown velocity if the angle of arc can be defined through use of encoder readings.

[0036] FIG. 7A and FIG. 7B depict two scenarios where the robot registers the presence of one or more than one obstacles in either the left or right action zones only. In such case, the envelope of the obstacles is the nearest obstacle itself 214. The angle of 214 with respect to the robot's major axis θ , 210 and ϕ , 212 is then used to decide the radius and turn

angle of the arc. In some embodiments, the distance function can be modified by another factor in order to incorporate risk assessment of a particular obstacle profile. For instance, obstacles immediately in the front of the robot have a greater propensity or risk for collision than those on the sides at a distance greater than the robot's diameter. This is assuming the obstacles are static or moving at a smaller/equivalent relative velocity with respect to the robot's major axis. In case of fast moving obstacles or in cases where the robot computes non-feasibility to circumvent, the robot may temporarily change its policy to a dead stop and wait for a pre-defined amount of interval for the dynamic obstacle to pass. In case of standby for a duration longer than a set threshold, the robot may again raise a flag for manual operation through a remote operator.

[0037] In some embodiments, the robot may perform corrective maneuvers in cases where it is completely obstructed by dynamic obstacles. In any embodiment, a maximum set limit on the velocity of the robot might be set as a safety measure to avoid sudden collisions and dead stop cases. For instance, if the indoor environment is a hospital, the robot will have a considerably lower set maximum velocity than if the indoor environment was a hotel.

[0038] Turning now to FIG. 8, a case is shown where the robot starts from its source 220 under a policy of forward. The task of the robot is to reach the destination area depicted as 222 in the figure. For this task, the robot will need to make decisions at intersection points 224 and 226 while avoiding obstacles. By using the computational principles defined in the earlier sections and figures, the robot will ideally follow a policy of forward until the intersection point 226 where the semantic map will direct it to follow a policy of right followed by forward. The destination area 222 can be localized and identified at a finer distance resolution by the robot through use of methods like optical character recognition, RFID tags, Bluetooth tags, color markers, infrared spectrum markers, and/or others. After successful completion of this task, the robot may choose to invert its supplied semantic policy map to reach the source or query the system, onboard or remote, to obtain the next set of semantic policies in order to reach back to the source or its next set destination.

[0039] In some embodiments of the invention, the robot has an active microphone and audio processing system that listens to commands in any defined format. For instance, there may be a case where the robot is needed to stop and move back certain distance in order for an obstacle to move away. Such cases may happen commonly in hospital buildings

or even hotel buildings. In other embodiments, the robot may have a dedicated speaker module that may allow it to respond in a calculated manner when needed, for instance, to receive any feedback from people.

[0040] In some embodiments of the invention, the robot is integrated with a face recognition system that allows it to identify people registered within the indoor environment and interact with them as required. For instance, if the robot recognizes a checked-in person in a hotel setup, it can appropriately greet the person. This feature is intended to add a greater social value and user friendliness for the robot through social interactions as needed.

[0041] Turning now to FIG. 9, various functions of the navigation module API 252 are depicted as blocks. In preferred embodiments, these functions are implemented programmatically as asynchronous processes in a multi-threaded paradigm. In other embodiments, the functions might be implemented in part as sequential processes within a main process. The API provides a function 240, that allows a remote operator to start or stop the laser based depth sensor functionalities. The remote operator may be a human in the central management system, another computing module or another thread of the API itself. Instructions to perform such a functionality may be relayed through the use of wired methods or wireless methods, including but not limited to, websockets, REST, MQTT, socket.IO. Decision to start or stop the lidar scans may be primarily motivated from power consumption optimizations or debugging.

[0042] The navigation module API can also provide a function 242, that results in the application sending raw scan data from the depth sensors to a remote operator. The remote operator can be a human at the central management system for viewing, or another computing unit onboard for sensor fusion and further decision making. The scan data can be sent as raw data in decimal format, raw data in hexadecimal format, data encoded according to some existing compression schemes or any compression scheme whatsoever.

[0043] The navigation module API can also provide functions 244 that enable a remote operator of types as described above, to start or stop the navigation sub-process. This may be done in response to robot boot-up or module re-initialization for debugging purposes.

[0044] The navigation module API stores the robot's current state in an appropriate variable 246. The robot state can represent its current working status for instance, serving destination zone <destination zone id>, faulty, manual assistance needed, and/or others.

These robot states are made accessible to remote operators through wireless protocols including but not limited to those defined earlier.

[0045] The navigation module API may store a semantic map 248 of the environment. This map is stored in appropriate file formats as XML, program array, text file, and/or others.

[0046] The navigation module API has a sub-process 250 that is responsible for detecting intersections in the indoor environment. This sub-process constantly checks for presence of intersections at a certain update frequency by using the depth sensor's scan data. A counter keeps track of the number of intersections passed under given policy instructions to the robot. At each intersection, the sub-process calculates an estimated time to next intersection based on the robot's current velocity and data from the semantic map of the particular region of the indoor environment.

[0047] FIG. 10 is a sketch of the robot showing its design and placement of various sensors.

WHAT IS CLAIMED

1. A semantic navigation and localization strategy for an autonomous robot in an indoor environment without needing detailed geometric map of the environment, provided through the steps of:
 - creating a semantic map of the said environment by identification of junctions or intersections in its structure and storing it in an appropriate media format.
 - storing the above said semantic map of the indoor environment in an appropriate format for future retrieval.
 - generating a semantic traversal map in form of instructions or policies, when needed. The semantic traversal map necessarily represents states as a combination of three instructions: forward, left or right, based on desired location of the robot, without any other information.
 - localizing using knowledge of the number of junctions detected by the robot while following the policies.
 - a method of detecting and compensating sudden deviations from the current policy that the robot is programmed to follow.
2. A means of claim 1, providing for a method to generate the semantic maps from the floor plan of the said indoor environment using image analysis tools.
3. A method of claim 1, wherein the robot is programmed to strictly follow a set of policies specified. For instance, under a policy of forward, the robot keeps moving forward avoiding obstacles, and making adjustments to avoid obstacles. In a curvilinear path, the robot can easily traverse the arc under control by a simple policy of forward.
4. A method of claim 1, wherein the robot is programmed to autonomously make decisions to follow an arc or move straight based on current policy.
5. A method of claim 1, wherein the robot avoids circular loops while traversing, by keeping track of its heading direction with respect to time elapsed.
6. A method of claim 1, wherein the robot performs sensor fusion of data from other distance sensors like ultrasonic or infrared with the primary sensor used for navigation and localization.
7. A method of claim 1, wherein the various modalities of distance sensors installed on the robot are assigned a priority number that allows one sensor to override decisions made by the others.
8. A method of claim 2, wherein the semantic instructions containing the policies to be followed is calculated on the robot through a computing unit onboard.

9. A method of claim 2, wherein the semantic instructions containing the policies to be followed is calculated on a remote central management system. The map is sent through some wireless protocol in a proper format to the robot.
10. A navigation planner module that facilitates policy following and localization through the steps of:
 - defining an action zone for the robot based on some mathematical formulation suitable to the indoor environment. This region is necessarily pre-defined without needing repeated calculations by the robot.
 - checking for presence of obstacles within the above mentioned zone and deciding whether it is present only on the left side of the robot, only on the right side of the robot, or both sides.
 - in case of only on either side of the robot, directing the robot to follow an arc in order to avoid the obstacle. The radius of this arc being a function of the distance of the obstacle's nearest point from the center of the robot.
 - in case when more than one obstacles happen on either side of the robot, the robot calculates an envelope followed by distance estimation to nearest point in the envelope.
 - in case when an object is on both sides of the robot, the module calculates a slope value by calculating the envelope of the obstacles. This value is further used to decide the radius of arc to be traversed.
11. A method of claim 10, where the robot implements the above functionalities necessarily as asynchronous blocking or non-blocking functions.
12. A method of claim 10, wherein the safety zone is defined as an ellipse with major axis in front of the robot and minor axis on the robot sides.
13. A means of claim 10, wherein a planar scanning lidar is used to obtain a two dimensional map of area around it.
14. A method of claim 10, wherein the navigation module application generates an arc for the robot to follow by calculating the circle center and radius resulting from the topology of the obstacles detected in the safety zone.
15. A robot platform, comprising:
 - a navigation module having:
 - at least three ultrasonic distance sensors.
 - two differential drive motors with encoders.
 - a planar range finding device with a range of at least 5 meters.
 - a computing unit to facilitate calculations and relay it to other modules installed in the robot.

interaction with a central management server located within the indoor environment or on the cloud.

a detachable module to be placed above the navigation module providing additional functionalities to the robot as:

user interaction through a visual display.

user interaction through gestures.

16. An implementation of claim 15, wherein the robot can be switched from autonomous control to manual control by a remote operator after indication through a program variable.
17. An implementation of claim 15, where the robot's detachable module is provided with an empty space for transporting items if needed.
18. An application programming interface (API) of claim 15, provided through a cloud service or stand-alone service, allowing instantiation of robots dynamically and exposing functions to allow immediate deployment and management.