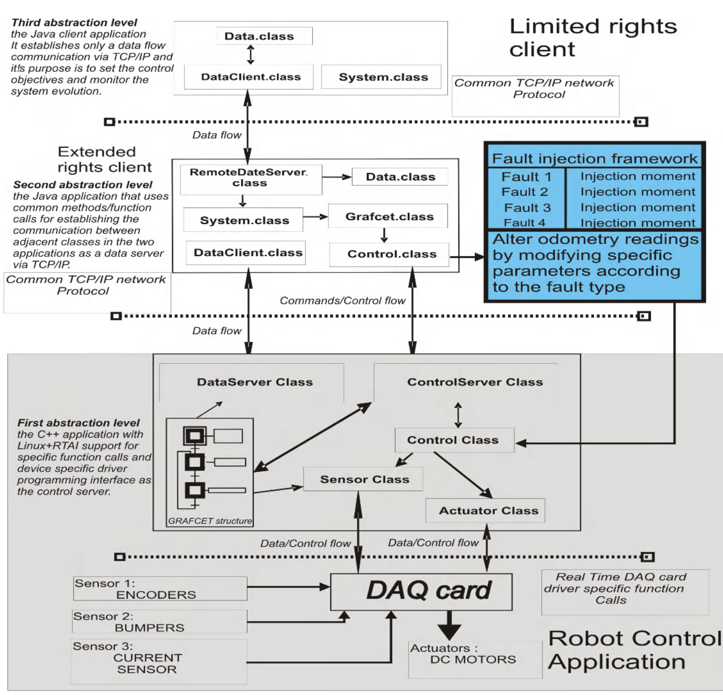
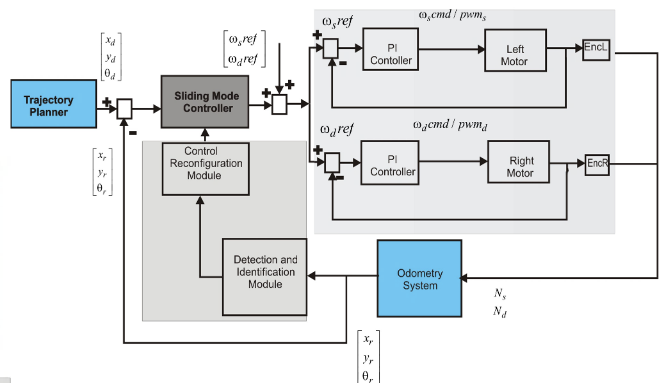
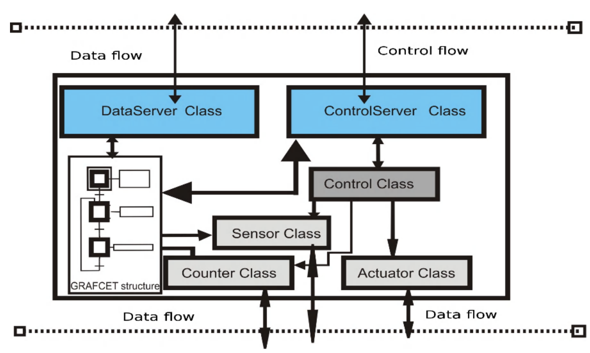
# MSc. Thesis Report
# Environment mapping for limited sensor mobile robots

The current report introduces a novel implementation for limited sensor mobile robots environment mapping. The developed algorithm is an offline SLAM implementation. It uses real time data acquired from the sonar ring and uses this information to feed the mapping module for offline mapping. The latter is running on top of a real time fault tolerant control application for mobile robot trajectory tracking operation. The designed application, that includes both hardware and software design, was developed to fulfill the MSc Thesis requirements.

As mentioned earlier the mapping module is based on a real time fault tolerant control application. This application was developed on a distributed pattern containing two main components, an embedded server running on the robot and a series of distributed clients. A synthetic description of the base application description is next given, focusing on the main software architecture, the distributed client architecture, the real time control loop and the interconnection logic.



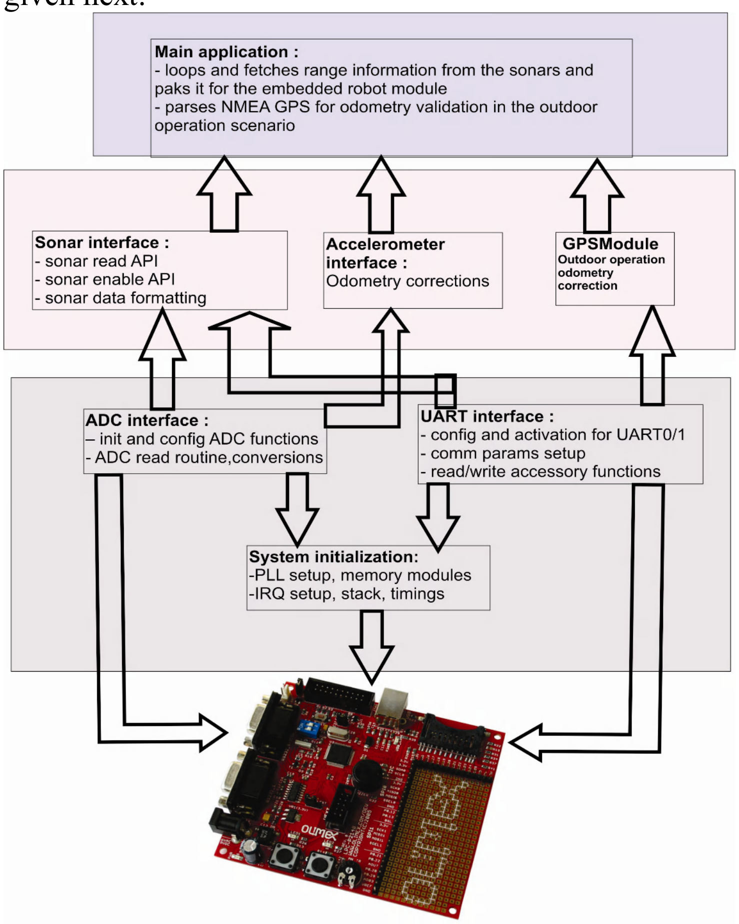



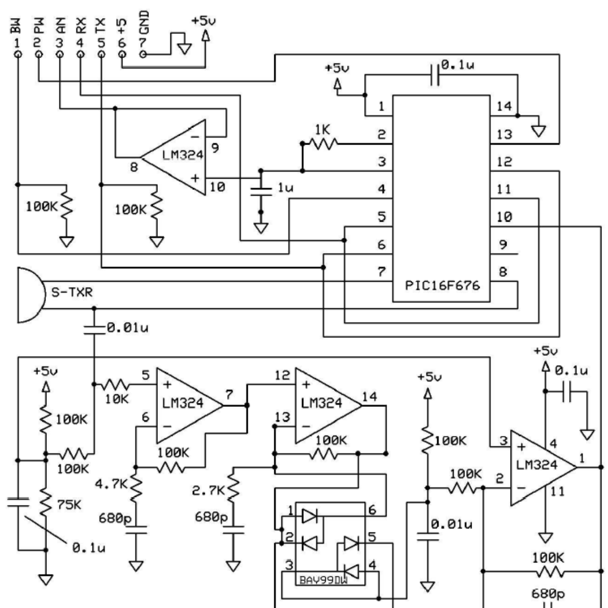




The server application implements a real time control loop to compute the control signal for a compound control loop (PI and Sliding Mode). The control task is executed concurrently with a monitoring and diagnosis task for fault detection and control reconfiguration. The monitoring and diagnosis task synthesizes five Kalman filters for residual computation and fault discrimination based on probability sequences. On the client side the main application enables only manual operation specific commands to access/set the control loop parameters and inject the faults for base platform testing procedure. The distributed architecture is using a typical TCP/IP network and is using message based upper level wrappers.

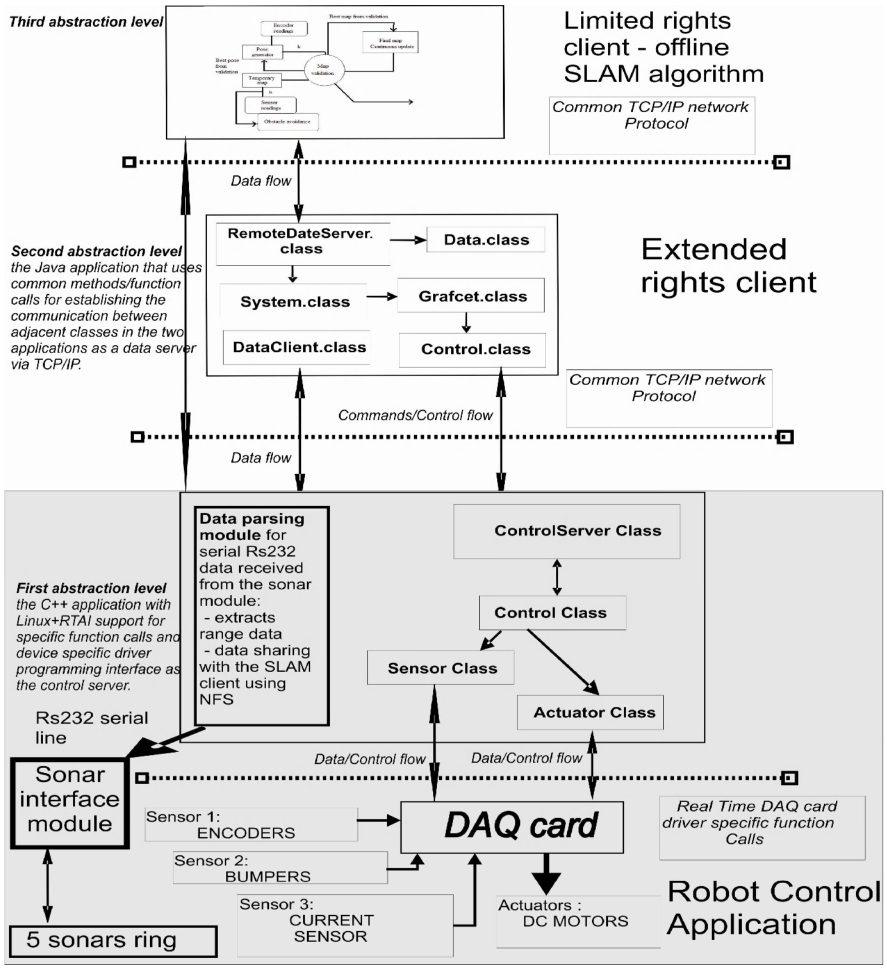
On top of this layered base level platform a mapping module was synthesized. The glue logic with the base level application is ensured by the main task concurrency mechanisms and by network file system distributed application support. The developed approach is a metric mapping algorithm, using Bayes rule and filter synthesis to implement an occupation matrix based offline SLAM algorithm. Starting from the Bayes rule a Bayes filter is synthesized to implement sensor fusion between odometry information and range information from the ultrasonic sensors. Some analysis details are next given to properly determine how the proposed implementation fits in the mention category of algorithms. By implementing an occupation matrix based algorithm uncertainty representation in the map is minimized by using continuously generated auxiliary temporary maps. This short term maps are containing a snapshot of the sensor readings at a certain moment in time. To minimize the error accumulation a specific time interval for data acquisition was defined. This interval will define in fact the amount of information that the temporary map will contain. A second important aspect is regarding the algorithm convergence that is good and it is ensured by the fact that the solution is based on incremental sensor fusion steps. The third aspect that was considered in the design stage was the dimension limits for the maps and the link between the environment features and the acquired data to give a valid solution in a static environment. Another important aspect was based on the quality of robot model, sensor model and environment model. The sensor model was useful to convert the raw measurements into a generic form to carry the conditional probability for each cell of the occupancy matrix given for a sensor reading.

To better emphasize the developed module specific a separate (hardware/software) description is given. So, starting with the low level sonar data acquisition level, an important aspect is to consider the definition of the embedded components. A new board to interface with the sonars was considered. This additional module reads voltage information from the sonar ring and packs it in specific RS-232 packets. The packaged range information is then sent using a serial line to the server application where an additional packet parsing and conversion and serial device initializing routines were added. A synthetic depiction of the range acquisition module along with a schematic of the sonar module is given next.

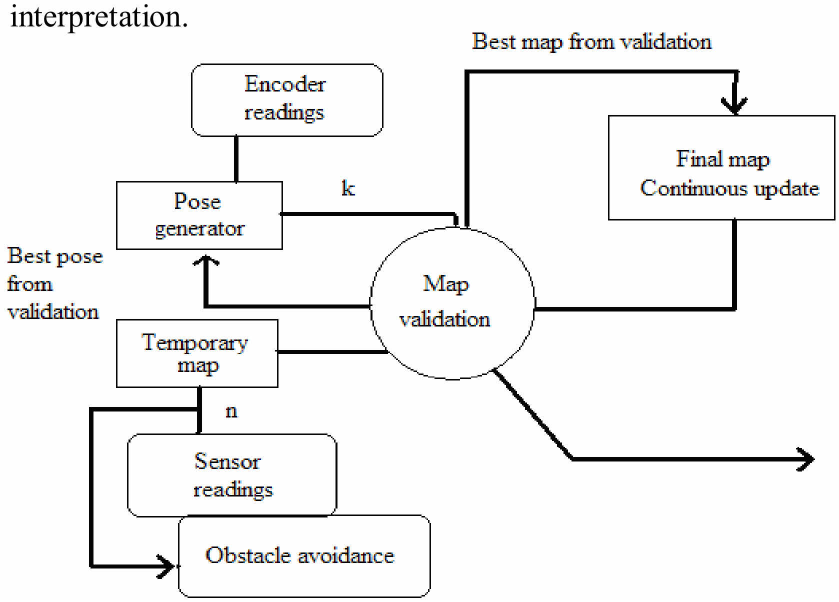


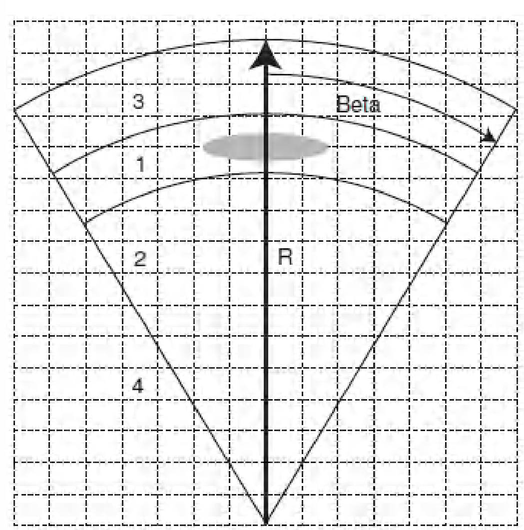




The main application running on the embedded linux machine on the robot includes an adapting layer to support the interface with the sonar interface module and the support for data acquisition used in the sensor fusion stage for the SLAM algorithm. The main architecture of the real time control application was enhanced to include the new functionality. The next image synthesizes the final application layers.




As mentioned earlier the SLAM implementation relies on multiple components. The first component is the sonar interface module found onboard the robot. This module gathers range data from the sonar modules and sends it to the real-time server application over the serial line. The server makes the sonar and odometry data available to the distributed client over a network file system. After the trajectory tracking operation is completed the acquired data is accessed by the client from the shared raw data files. The client then runs the offline SLAM algorithm. To synthesize the SLAM implementation a functional diagram is given along with auxiliary information about region separation used in data interpretation.

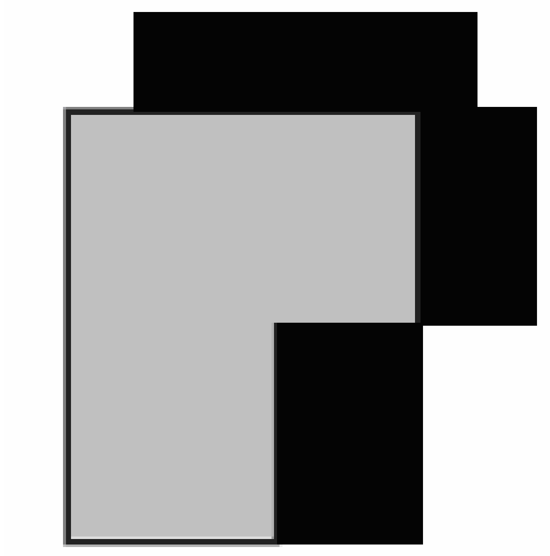


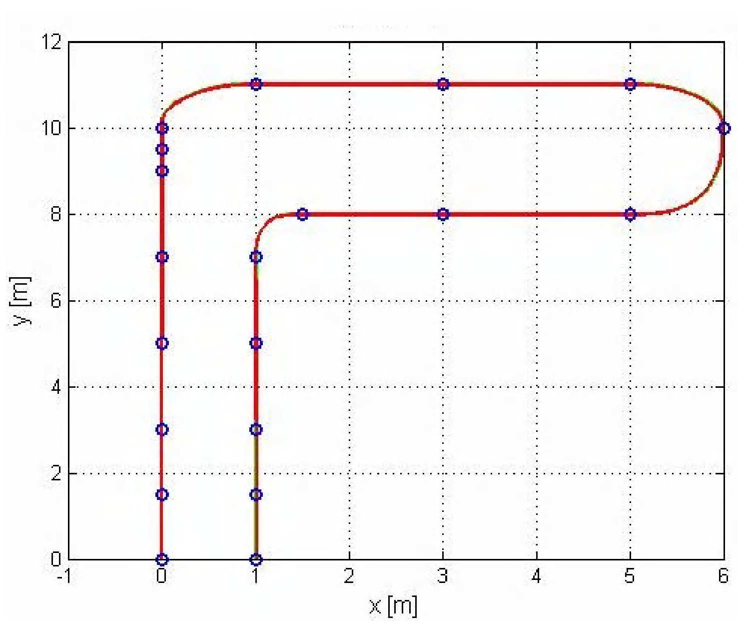



The defined regions are representing the way acquired data is interpreted. Hence, region 1 contains environment elements partially occupied, region 2 contains those environment elements that are probably empty, region 3 defines the uncertain elements (found behind the occupied elements) and finally region 4 defines the sonar out of range elements. In the first stage in map building will use the data acquired from the sonar ring. This data is later interpreted as probability profiles to define occupied and empty regions in the environment. Odometry data is integrated with the sonar data based map. Data integration is composed of multiple steps starting with the initialization of the temporary map to an empty state. Next, the superposition of the empty and occupied zones is done for the existing sensor readings. Following, the threshold computation is made by comparing relative power values for the empty/occupied matrix cells. The Bayes sensor fusion is used to complete the data integration stage by converting the measurements from the sensors into conditional probabilities used in the algorithm to make a generic representation in the map validation stage.

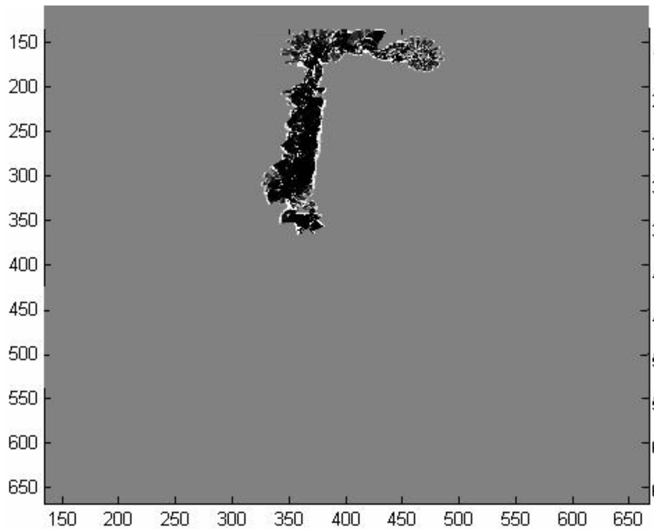
The offline SLAM application is built using multiple functional sub-modules. The first sub-module is responsible with acquiring data from the environment and the navigation logic. The robot shall navigate (trajectory tracking) in the environment and shall acquire data from the sonars and compute odometry data. The current pose of the robot is used also in the real time control algorithm. As one can see in the algorithm synthetic representation the best validated pose will be used to correct odometry data. The temporary map is incrementally built using sensor information until it becomes mature and then a validation stage is required. The validation stage assumes that the temporary map is compared on a per pose basis with the final map. Practically, the algorithm just compares the poses from the two maps and the difference between them is considered to be the odometry error. After updating the robot pose the temporary map is fused along with the final map resulting a new final map and a new iteration will start.

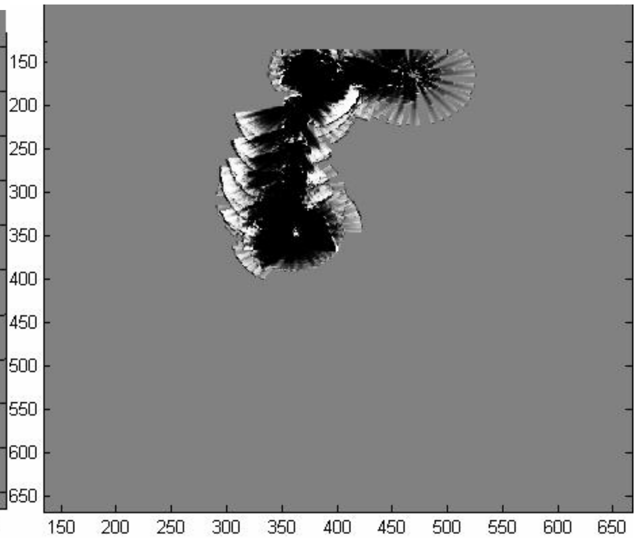
To validate the developed algorithm a test scenario was defined and the results are next presented. The environment configuration and the trajectory tracking profile are depicted next.

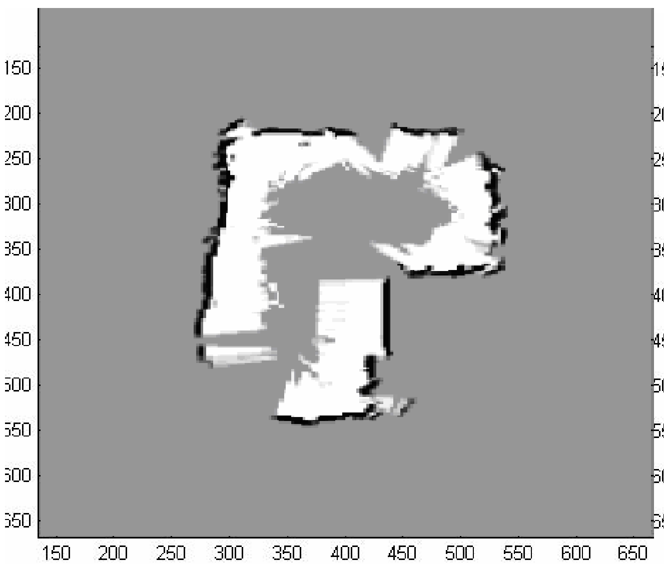


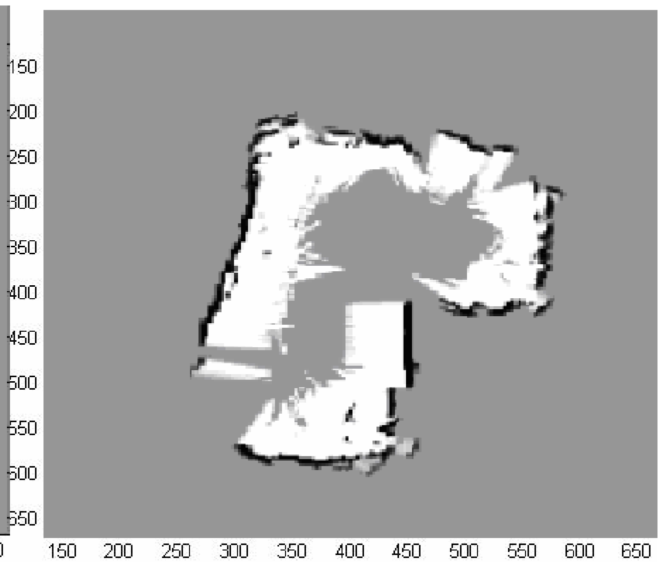



The developed algorithm refines the room representation incrementally, so by adjusting the algorithm parameters one can achieve good quality index in the context of a good and justifiable computational effort. For the given environment configuration the offline SLAM algorithm computed the next representation obtained for various parameters setup (continuous refinement). The upper images are depicting intermediary readings form the temporary maps and the lower image are introducing the final maps generated by the algorithm.




On can observe that the representation is proper and by adjusting the algorithm parameters can obtain a better performance index. The presented environment profile was defined in the real case of validation when the robot had to navigate to half the hallway in the faculty 5$^{th}$ storey. As future work the main focus will be on implementing the same concepts for online real time mapping and also to add an adaptive parametrization stage according to the input environment profile (empty areas, occupation matreix dimension, sensor number). The presented application was a minimal justifiable implementation of SLAM for limited sensor mobile robots that included hardware and software design. Next some internal robot structure details are depicted.