DESIGN AND SIMULATION OF A MOBILE TOUR-GUIDE ROBOT FOR MUSEUM GUIDANCE

by

Serdar Sarı

B.S. in Electrical and Electronical Engineering, Bilkent University, 2002

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Systems and Control Engineering
Boğaziçi University
2006

ACKNOWLEDGMENTS

Many people have helped me along the way; First of all, I would like to thank all of them.

Furthermore, I wish to thank my promoter Prof. Levent Akın, for proposing this challenging project, for his confidence in my abilities and keeping me on the right track during the two years.

Many thanks go out for the Boğaziçi University AI Lab members for their guidance and patience with me.

Special thanks to Assoc. Prof. Eşref Eşkinat and Assist. Prof. Pınar Yolum Birbil, for their kindly attendance to examination jury.

Finally I would like to thank my family and friends for being so supportive. I would not manage without them.

ABSTRACT

DESIGN AND SIMULATION OF A MOBILE TOUR- GUIDE ROBOT FOR MUSEUM GUIDANCE

This thesis aims at building a realistic design of a fully autonomous social mobile robot act as a tour guide to the visitors in museums and that can give information about the museum and the exhibits. It should define all the parts robot need such as locomotion mechanism, sensors, batteries, motors, body, control algorithms and methods required to accomplish its objective. Three dimensional modeling of the robot according to this design, building a virtual museum and dynamic agents, based on a real museum, in a simulation environment for further research and progress are also accomplished. Also, to verify the design, a fuzzy controller based on a layered behavior-based approach, has developed to control the robot. The behaviors used by the robot include going to target point, avoiding obstacle, avoiding human in front of it by speaking and waiting the visitors behind who are following him.

ÖZET

MÜZELERDE TUR REHBERİ OLARAK KULLANILABİLECEK OTONOM HAREKETLİ BİR ROBOT TASARIMI VE BENZETİMİ

Bu tezde, müzelerde tur rehberi olarak, gelen ziyaretçilere yön gösterebilen, onlarla konuşabilen, sergilenen eserler hakkında bilgi verebilen, otonom, hareketli, sosyal, bir robot için gerekli donanım ve yazılımın tasarımını yapmak ve yapılan bu tasarım doğrultusunda üç boyutlu benzetimi gerçekleştirmek amaçlanmıştır. Bu tasarım olabildiğince gerçekçi olmalı, robotun sahip olması gereken hareket mekanizması, algılıyıcıları, aküsü, motorları, vücut yapısı, kontrol mimarisi, robotun yapması gerekenler doğrultusunda, ortaya konmalıdır ve bunların uygunluğu test edilmelidir. Üretici firmaların yayınladığı bilgiler doğrultusunda tasarımı yapılan robotun kontrolü için tepkisel robot mimarisi ve bulanık mantık kullanılmıştır. Dört farklı davranış, hedefe yönelme, engelden sakınma, insanla konuşma ve takipçileri yönlendirme davranışları tanımlanmış ve üç boyutlu modellenen ortamda test edilmiş, yapılan tasarımın ve geliştirilen kontrol programının yeterli olduğu gösterilmiştir. Test ortamı için gerçek müze ortamı ve çoklu etmenlerin modellenmeside yapılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

$P_{M\max}$	Maximum power motor could deliver
P_m	Motor power
τ_m	Torque of motor
t_{op}	Operation time of robot
τ_s	Stall torque of motor
$v_{\max}$	Maximum velocity of robot
v_{cut}	Cut-off voltage of batteries
ω_m	Angular velocity of the shaft of the motor
ω_f	Free speed of motor
W_{robot}	Weight of the robot

2D	Two dimensional
3D	Three dimensional
AC	Alternating current
AFSM	Augmented finite state machines
AI	Artificial Intelligence
API	Application programming interface
DC	Direct current
DOF	Degrees of freedom
FOG	Fiber optic gyroscope
FSA	Finite state acceptors
GND	Ground
GPS	Global Positioning Systems
LED	Light emitting diode

MTBF	Mean time between failures
MTR	Mean time to repair
IR	Infra-red
RADARS	Radio detection and ranging sensors
SONAR	Sound wave sensor
RF	Radio Frequency
TOF	Time of flight

1. INTRODUCTION

Complex social mobile robots which could serve and get into contact with people are always of great interest. Until late 1990's they were only the topic of novels and films. But growing technology and academic research made it possible to realize this now. There are many studies on both mobile robot navigation and interaction. These techniques and industrial high quality production for mobile robots take the autonomously navigating vehicles out of the university offices and put them into real-world applications. Service robotics is one of such applications where tour guides [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15], maids [16] and assistants [17] for impaired or old people had been very well implemented.

In this study, the aim is to implement an interactive tour-guide robot which will be the Turkey's first social robot. The focus of this thesis is: On the design of the system and implementation of the low level behaviors. Implementation of high level control, social interaction and tour guide functionalities are beyond the scope of this work.

Design and implementation of autonomous mobile robots needs the integration of many different bodies of knowledge, making mobile robotics a challenging science. Kinematics, dynamics, control theory and mechanisms are required to solve the locomotion problem. Signal analysis, sensor technologies and computer vision are required to create perceptual systems. Computer algorithms, information theory, artificial intelligence, probability theory needed to solve localization and navigation problem of robots. Mechanics, art, architecture knowledge is unavoidable for physical appearance. So in this thesis, a design that is as reliable as possible according to the information gathered from all the sciences mentioned above has been done.

Autonomous vehicle navigation is often implemented within a behavioral architecture which has excellent real-time execution properties. A reactive behavior tightly couples a perceptual input with a control action. In this work, such a reactive navigation is developed based on a fuzzy control system. First of all fuzzy control systems are suitable in that fuzzy rules provide a natural framework to describe this kind of stimulus-response behavior. Secondly, this robotic behavior is constituted by set of rules which can be designed without requiring any mathematical or logical models that, control algorithms could be developed in laboratory without a real robot. Moreover, modular structure facilities the integration of new behaviors without requiring modification to the existing ones. And finally, logic representation in linguistic form which permits the designer to define highly abstract behaviors in an intuitive fashion motivated us.

A museum robot project is an expensive project for the university conditions that a good design and lots of preliminary work had to be done before implementation. Because, there would be no prototype chance before implementing actual robot that you can make trials on it than improve. For this reason, to test, create, modify and improve such a robot project a simulation environment, which had to be as real as possible was required. The robot, museum and visitors are modeled as part of this thesis, in a three dimensional simulator for future development and studies on control and design. The designed robot and control algorithm developed were tested in this modeled simulation environment as the final work in this thesis.

In this thesis, chapter two gives brief information on robots implemented in museums to serve people. In chapter three, behavior based robotics and subsumption architecture is discussed. In chapter four, a requirement analysis done, the necessities for the robot defined and robotic parts investigated. In chapter five, the software design and reactive control program that developed to run the robot based on fuzzy-control explained. Later on, in chapter six according the design and implementation details of three dimensional models of robot, museum and agents stated. Chapter seven is the part where simulations and their results demonstrated that had been done to evaluate how well the control program and the museum robot. Chapter eight is the conclusion chapter of this thesis. Finally, we conclude the thesis with chapter nine where future works will be done is stated.

2. MUSEUM ROBOTS

In the literature there are many museum robots, with different architectures. The mobility systems of these robots and the types of sensors they use do not differ much, however the navigation methods and algorithms, and the tools and methods they use for human-robot interaction are numerous. The museum robots or tour-guide robot's task can be divided into two issues: interaction and navigation. In this context, when we examine the museum robots, each newly developed museum robot has introduced a new functionality over its predecessors. This is not only because of the experiences learned but also because of the developments in artificial intelligence that always introduces new algorithms and methods, which simplifies the development of more efficient and fast software architectures.

2.1. Rhino

The first museum robot in literature is, Rhino [1, 2], implemented in Deutches Museum Bonn in 1997. It was a joint project started in 1994 between the "Institute für Informatik III" of the Rheinische Friedrich-Wilhelms-Universität and the Carnegie Mellon University. The central scientific goal of the Rhino project was the analysis and synthesis of complex adaptive and learning systems in the area of mobile robotics. Rhino was able to create metric maps of the environment by the data from its range sensors. Then from the metric maps, topological graphs are constructed that describe the environment in an abstract and compact way. To find out the answer for the question: "Where am I?" Rhino tries to match the sensor readings with a model of the environment by a probability density function. Rhino's path planning is based on value iteration, propagates values through the map, to find minimum cost paths to a goal. With its reactive collision avoidance method, dynamic window approach, Rhino was able to navigate safely though the rapidly moving people. The key feature of this algorithm is reducing the search space to the dynamic window, which consists of the velocities reachable within a short time interval. Being a sophisticated tour guide, requires careful task planning, taking into account issues such as the amount of time available, choosing appropriate exhibits to visit on that tour, and planning the shortest path covering those exhibits. Rhino's mission planning, coded by a declarative language based on situation calculus which is implemented in Prolog, allows the user to concentrate on the high-level specification of complex actions without having to worry about how these are actually carried out by the robot that is used.

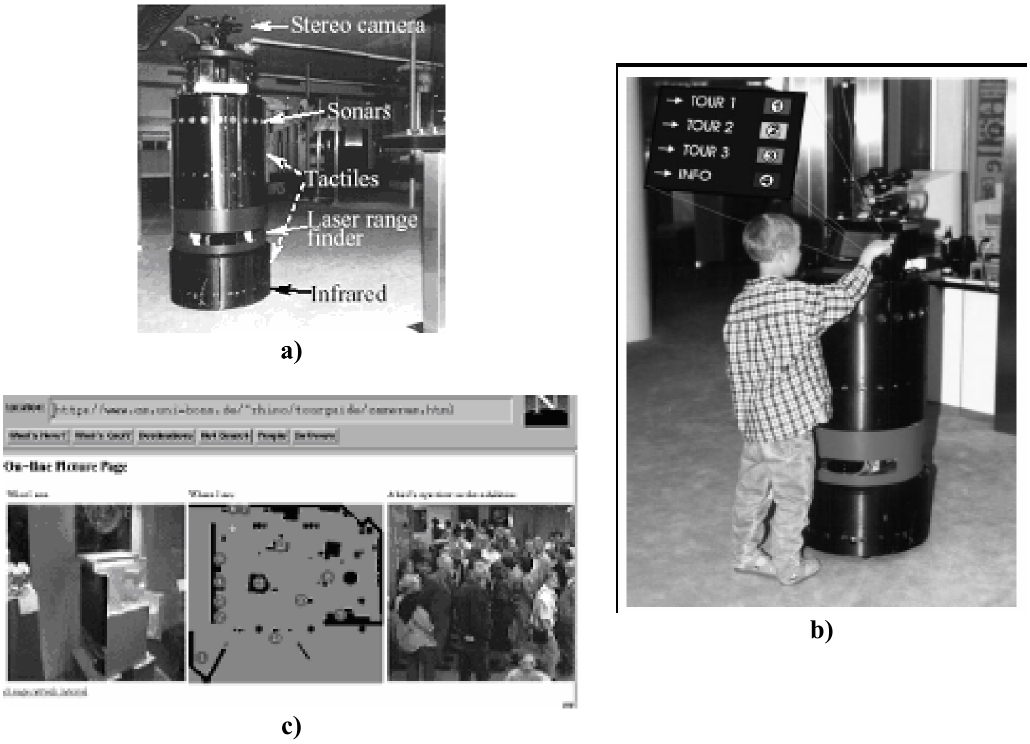


Figure 2.1. Rhino's a) Sensors b) Interaction menu c) Web-page [1]

Rhino had been heavily equipped by sensors and electronics that could be seen in Figure 2.1 (a); 32 infrared range sensors on the base, 32 contact sensors on the base, 24 ultrasonic sensors on the enclosure, 24 contact sensors on the enclosure, 24 infrared range sensors on the enclosure, two Pentium 200 inside the enclosure, one on board computer, two video cameras on the console, one radio link Ethernet (10Mbit/s). The computation was so intensive that to be able to run the software program of Rhino three on-board, two off-board computers had to work together. Dependency of external resources could be counted as a drawback for a mobile robot. However, Rhino was successful in navigating autonomously. The main deficiency of Rhino was lack of interaction capability except a menu shown in Figure 2.1 (b) where a visitor could select a predefined tour. For this reason Rhino is accepted as an interesting machine in the museum but not a social tour-guide. Beside all of these, Rhino was able to transfer the images it captures by its camera, and the location of itself by a wireless link that, from the web page of Rhino shown in Figure 2.1 (c), a person could able to visit the museum by Internet. See what Rhino see, look at the location of Rhino and also watch the robot from museum cameras.

2.2. Minerva

The experiences gained from the Rhino robot led the creator team to implement a second museum robot, Minerva [3, 4, 5], which had been employed in Smithsonian Institution's National Museum of American History for two weeks. Minerva, which could be seen in Fig 2.2 (a), was rather different from its predecessor. It had a novel design, a rectangular body and a face as a human. The numbers of sensors were reduced when compared with Rhino.

Minerva was controlled by a distributed network of computer programs. Globally we can divide them like Rhino. First one is mapping and localization. Minerva used two types of maps to orient itself: One was the occupancy grid maps like the Rhino and the second was the texture map of the museum's ceiling. Both maps are learned from sensor data, -laser scans, camera images, and odometry readings- by manually maneuvering the robot through its environment. Minerva employed a modified version of Markov localization [3, 5] to locate itself by using these two maps. Markov localization maintains a probability distribution over all possible poses and estimates the most likely location on the map. Since Markov assumes the environment is static, the researchers modified the Markov localization by adding a filter to eliminate some sensor readings. One of the key issues of success of Minerva for localization was made use of the texture on the ceiling that was perceived by a camera towards to floor. Second software module was collision avoidance; Minerva used an extended version of Dynamic Window Approach [3, 5]. Because of two differences first one is Minerva had differential drive (a non-holonomic drive system) where Rhino had synchronous drive (a holonomic drive system). Second reason was the rectangular shape of the Minerva. Third software module was, path planning, Minerva's path planner was called a coastal planer, keeps the distance constant to known obstacles. For this reason the path planner generates paths which minimize the path length and maximize the information content. The forth module was task planning; Minerva's task planner was basically supposed to do two important tasks. The first one was scheduling the tours and monitoring their execution. The second one was changing the course of actions when an exception occurs. This high level controller of Minerva was implemented on the top of a reactive plan language. Minerva's high-level planner learns a model of the time required for moving between pairs of exhibits to plan optimal tours. The final module was interaction; Minerva was also able to speak, she (its voice was a female voice) was able to ask visitors to get out of her way when it is blocked. Minerva had a face that consists of a movable mouth and eyebrows, and two video camera eyes, mounted on a pan/tilt neck that the main feature was her ability to exhibit different emotions via this mechanical face. In Figure 2.2 (a) neutral and (b) angry face appearance could be seen.

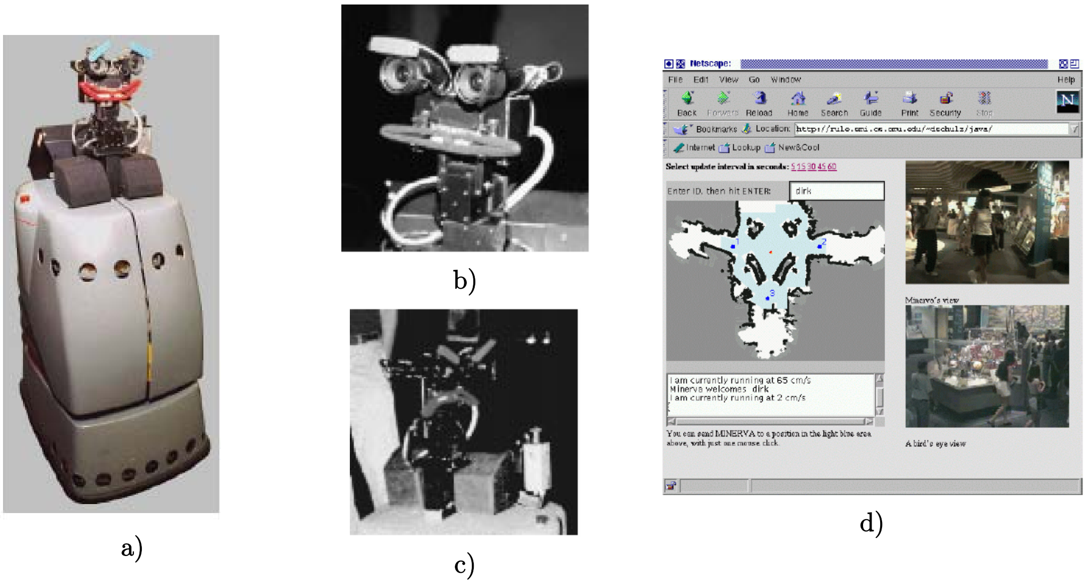


Figure 2.2. Minerva's a) Itself b) Neutral face c) Angry face d) Web-page [3]

When compared with Rhino, the major difference of Minerva was the interactive capabilities. People were more satisfied, more amused and more interested in it. Most of the kids thought she was alive and almost all the visitor's thought that she was very intelligent. Minerva was not only successful at navigation but also successful at being accepted by people. However dependency on off-board resources like Rhino was Minerva's drawback. An improved web-interface was also developed shown in Figure 2.2 (d), for the online visitors who could direct the robot when the museum is closed.

2.3. Mobot Robots

After the success of Rhino and Minerva, the Mobot Company decided to invest for the museum robots. With the academicians from Stanford and Carnegie Mellon Universities, started a project in 1998 and that was finished at the end of 2001. In the project four museum robots were implemented; Sage [6, 7], Sweetlips [6], Joe Historybot [6], Adam 40-80 [6] which could be seen in Figure 2.3. The first robot, Sage (Chips) began work at the Carnegie Museum of Natural History in 1998 and operated for almost three years, covering a total travel distance greater than 323 kilometers. The second robot, Sweetlips, conducts tours in the Hall of North American Wildlife, also at the Carnegie Museum of Natural History, started to operate from April 1999 and covered a total distance greater than 145 kilometers autonomously. The third robot, Joe Historybot, was operated in the atrium of the Heinz History Center. Joe started to operate in July 1999, covering a total distance greater than 130 kilometers autonomously. The latest Mobot robot was which was used as an information desk (not as museum-tour guide) and operated in a variety of venues, including the Republican National Convention, the Democratic National Convention, a shopping mall, the National Aviary and finally the Pittsburgh International Airport. Each robot had been the products of a complete re-design based on lessons learned from the prior robots. Although mechanical aspects have remained unchanged, all of them evolved in an effort to improve the autonomy and interactivity of the robots. Like the others, Sage, had the ability to provide predefined tours presenting audiovisual information via a screen and speakers about exhibits during the tour. However, Sweetlips was also able to attract human attention by demonstrating an awareness of human presence such as saying “Hello”. Joe is able to answer many different classes of questions and even asks humans limited questions. Adam can play trivia games with humans and taking polls such an exchange, where both the human and the robot can initiate the next part of the conversation.

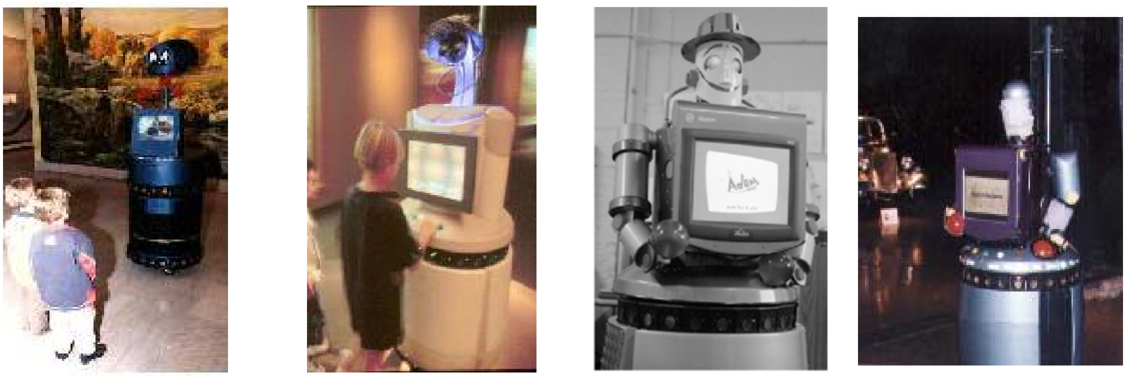


Figure 2.3. Mobot robots Sage, Sweetlips, Joe Historybot, Adam 40- 80 [6]

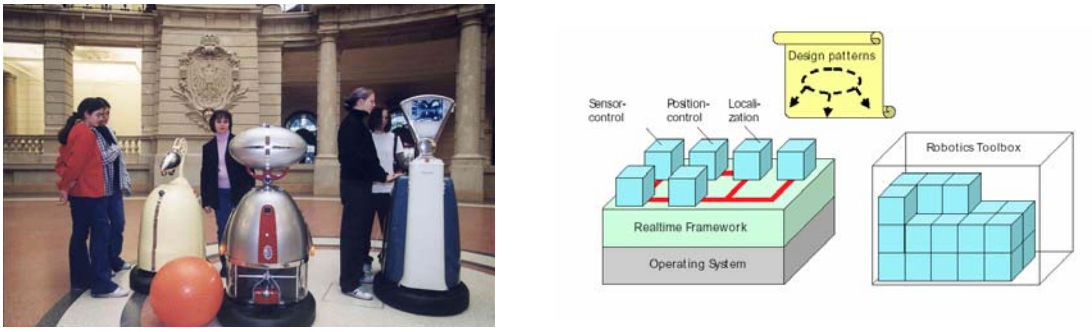
Sage has very simple but reliable computer architecture. It does not have localization or mapping module. It keeps only a topological representation of environment, the areas where it is allowed to move. It is simply adapted for only give predefined tours in a specified place. Its navigation module was based on reliable visual landmark recognition scenario. As seen in Fig 2.4, high-contrast two dimensional (2D)/three dimensional (3D) artificial landmarks which were installed on walls can easily be recognized by its camera. The robot was also limited to a predefined set of unidirectional, safe routes and zones. This means if the path is blocked, the robot could not continue with another available path beyond the edge of the safety zone. Besides these there was a risk that if one landmark failed, the robot could fail completely since there was no internal representation of the obstacle ridden environment, to locate itself. Sage, in every cycle, computes the most free direction of travel, within a pre-specified window of the desired direction of travel, purely as a function of the current sensor readings. Translational speed is computed independently from the direction of travel. One of the most important features of Sage was finding out the way to the docking station when it starts running out of power. The docking station was also specified by artificial landmarks that without any human intervention it could recharge itself.

Figure 2.4. Artificial land markers for a turn point and docking station [7]

Mobot Project's fundamental goal was creating a lifelong autonomous robot. Sage, who worked 174 days (135 days are error free) between 1998 –1999, 4000 hours in operation, 1000 hours in motion, 6 days mean time between failure (MTBF), 1 hour mean time to repair (MTR), showed that the project was very successful. The experiments during the four year implementation showed that: 1) People learn more if they spend time with the robot 2) Robot interactivity makes people to spend more time with the robot. 3) No matter how appealing is the robot, people attention is between 10-15 minutes consequently short tours and short info are more influential.

2.4. Inciting, Instructive and Twiddling

In 2000, three robots [8, 9] were installed in Communication Museum in Berlin by Fraunhofer University. They were, *Inciting,* that can able to speak and recognize visitors. *Instructive*, which gives explanations underlined by picture and video sequences and with its swinging head shows the exhibit about which it talking. And the last one Twiddling which only follows a ball, no ability to speak, ability to show its emotional moods (happy, grumpy or angry), according to the availability of ball, by different types of sounds and lights it can emit. The most exciting feature the robots had was they could interact with each other via wireless link and coordinate an action. For example, when Twiddling becomes angry Inciting helps him to find the ball or asks visitors to give it him back. These three robots could be seen in Figure 2.5 (a).



a) **b)**

Figure 2.5. a) The robots: Twiddling, Inciting, Instructive [8] and b) Their software library design [9]

One of the aims of the project was developing a software library that could be used in any kind of mobile robot. The designers of the robots established an object oriented ‘Real-time Framework’ and the software library ‘Robotic Toolbox’ for software control. The structural representation could be seen in Figure 2.5 (b). Robots used odometric data to locate themselves and positions are corrected by Kalman Filter which compares segments found in the natural environment (walls, doors...), matches the data acquired by laser scanner and predefined modeled map. Each robot used different types of motion planning according to the situation they are in: 1) Program Controlled Navigation: Navigation and act according to the predefined routes in the program. 2) Reactive Navigation: Current target position for a robot is constantly recalculated in reaction to its environment. 3) Preplanned Path: For the certain target position, it plans the shortest path to this position based on a statistic map. For safety of the robot and visitors, robots obstacle avoidance had three layers: 1) Laser scanner based collision detection, reduce according to vicinity, stop if necessary and wait till area again clear. 2) Activating bumper results with an immediate stop. 3) They do not leave the operational area. Since the robots operating area defined by white magnetic material on the surface, they used magnetic sensors to detect.

Even if the appearance of Twiddling, Inciting and Instructive are very different compared to each other (and any other social robots) their basic platform (undressed) was identical. The idea of dividing the museum tour guide task into entertainment, giving info and making interaction and solving it with a multi-agent cooperative approach was also very successful. Their total 1000 kilometer traveling score and amused visitors leaving showed their success clearly.

2.5. RoboX

The latest museum tour-guide robot, RoboX [5, 16], which can be seen in Figure 2.6 (a), had been applied at the Swiss National Exhibition Expo 02 during five months in 2002. It was the final product of cooperation between the researchers from the Autonomous Systems Lab, Swiss Federal Institute of Technology Lausanne and the BlueBotics Inc. It is still commercially available and employed in many places.

a) **b)**

Figure 2.6. a) The museum robot RoboX [10] b) Light emitting diode (LED) matrix on one of eyes [11]

When we examine the software architecture, we see that novel methods and architectures are used. RoboX keeps the map of the environment as a weighted directed graph. It consists of nodes, representing [x y φ] positions, and edges between the nodes. Every node refers to geometric primitives such as lines, segments and points that could be called natural landmarks. It doesn't track its pose with odometry. It reacts directly to the environment in the sense that features tell us when and where to place a location hypothesis. So the method used, which is categorized as a feature-based approach, gives the robot ability to find its way, in the case the robot totally lost: It can generate hypotheses about its current position and therefore relocate itself. RoboX plans its path to a target point with a search algorithm that generates a length-optimal path. Since the path is global and no sensor readings are taken into account, later on, dynamic path modification done on the path, during RoboX is navigating. It uses a navigation function that takes into account the current sensor readings and is not limited to nodes of the a-priori map. However, the paths generated have a very poor geometry, consisting of linear segments that lie on angles which are multiples of 45 degrees. Smoothing the path and adapting it to dynamic surroundings is done by another function. Dynamic window approach was also their choice for obstacle avoidance. The RoboX has introduced two new features for interaction; first one is face tracking, by using the color camera mounted on the left eye the robot can look at the person it is interacting with. It uses the Intel Image Processing Library and detects and follows skin colored regions. Second one is motion tracking, which is for distinguishing between moving and static elements in the environment that it was able to aware of the people around. To make the robot more interactive a led matrix on the right eye used to show emotional state and other figures for non-verbal communication. In Figure 2.6 (b) the negative, happy and confused moods could be seen.

Success of RoboX proven by serial manufacturing that makes the tour-guide robot a “product” rather than a prototype. All the previous examples of museum robots were a platform for researcher to study robotics but RoboX, with ability to speak four different languages (English, Spanish, German and French), is now a product that can traded worldwide.

2.6. Others

There are also other robots implemented in museums such as Kapros [12], Tourbot and Webfair [13] but they were only tele-presence robots for the web users. There are also lots of other robots implemented as a tour-guide in trade shows or exhibitions but not in museums. Such as: Diligent [14] and Blacky [15].

3. ARTIFICIAL INTELLIGENCE FOR ROBOTS

Robotics deals with the practical application of many artificial intelligence (AI) techniques to solving real-world problems. This combines problems of sensing and modelling the world, planning and performing tasks, and interacting with human beings and other robots. There are currently three paradigms to solve these problems of a robot where a paradigm is a philosophy or a set of assumptions and/or techniques which characterize an approach to a class of problems [16]. The oldest one is Hierarchical Paradigm which had been also the only one till 1990. Under the Hierarchical Paradigm; robot first collects data from all sensors to build a global world model. This model had to include all the details of the objects in the robot's world. Secondly, the robot plans a series of actions that will achieve the goal by using the world model. Finally, the plan is executed by appropriate commands to actuators of the robot. Second paradigm which was very popular between 1988 and 1992 and led to exciting advances in Robotics is Reactive Paradigm. Under Reactive Paradigm each sensor data is linked to an action where these sense-act couplings called behaviors. Several behaviors, even conflicting ones, could be active at the same time, however robot's final behavior will be combination of these active behaviors. This type of control inspired from simple animals in the nature and most of the time called as Behavior-Based Robotics. The final paradigm is Hybrid Deliberative/Reactive Paradigm, which overcomes all the disadvantages of previous ones. As understood from the name it is mixture of Reactive and Hierarchical paradigms. Under Hybrid Deliberative/Reactive Paradigm, sensor data not only routed to a behavior but also is available to the planner for the construction of a task-oriented global world model. The reactive layer fires the behaviors, and an arbitration mechanism decides the final behavior of robot by combining active ones.

3.1. Behaviour Based Robotics

Behavior based robot architectures decompose overall control system into set of behaviors. Each behavior is a coupling between sensor information and actuator command. All behaviors use limited use of symbolic representations or knowledge. However, all architectures differ in the granularity of behavioral decomposition and coordination of behaviors for final action. The most popular architectures are: *Potential fields* [17, 18], based on imaginary forces acting on the robot from the environment, such as repulsive forces from the obstacles and attractive force from the destination point, that the sum of all forces determines the subsequent direction and speed of travel. *Motor schema theory,* was an attempt to create a more general, behavior-based, conception of the potential fields method. A motor schema consists of an activation portion that examines the sensory data, and effector's portion that computes an action vector based on that stimuli. The effector's portion gives an action vector based on a function that corresponds to the particular goal of the motor scheme [19]. A robot operating on a motor schema based system may have many schemes, each of which examines a part of the sensory data and gives an action vector. There can be an infinite number of action vectors for each motor scheme. In this way sensory information from a variety of different modalities, such as vision, infrared, sonar, and bump sensors, can be integrated into motor command decisions. The behavior of the robot is computed by summing all of the action vectors together and acting according to the result. Thus the entire schema is blended together in the end. *The Distributed Architecture for Mobile Navigation (DAMN)* is a planning and control architecture in which collections of independently operating modules collectively determine a robot's actions. DAMN consists of a group of distributed behaviors communicating with a centralized arbiter, either by sending votes in favor of actions that satisfy its objectives, or by indicating the utility of various possible world states. The arbiter is then responsible for combining the behaviors' votes and generating actions which reflects their objectives and priorities. The distributed, asynchronous nature of the architecture allows multiple goals and constraints to be fulfilled simultaneously [20]. *Subsumption* is the reactive architecture that sensor-driven behaviours are represented as separate layers. Individual layers work on individual goals concurrently and asynchronously. Layers are organised hierarchically allowing higher layers to inhibit inputs or suppress outputs of lower layers. This constitutes a coordination method called priority-based coordination. The architecture can be built incrementally adding layers in different phases. Each layer is composed of one or more Augmented Finite State Machines (AFSM). Details of the architecture are explained in [21, 22]. Detailed information about behavior based robotics could be found in [23] and also brief explanations on the behavior based architectures proposed in the literature according to their behavior coordination mechanisms could be found in [24].

3.2. Subsumption Architecture

Subsumption architecture is one of the best methods for robot control which has a well defined principles and procedures for development [21, 22] which has already many successful implementations documented [25]. A list of the characteristics of a system that could best be approached using subsumption architecture stated in [26] that are very well fits for our robot control system that:

- The system is required to operate in real time,
- It needs to pursue multiple goals simultaneously,
- It exploits a parallel architecture,
- It needs to be aware of real time events external to it,
- It needs to process sensor data,
- There are conditions involving details not important to the system's central goals that need to be met and maintained in order for those goals to be pursued.

Subsumption provides a way to design a complex mobile robot control system in a straightforward, intuitive way that the design procedure such a system divided into four steps: Behavioral choice, behavior expression, behavior coordination and behavior encoding.

3.2.1. Behavioral Choice

There are many approaches to choose and design behaviors. First of the most known is *ethologically guided/constrained design* that is based on animal behavior. Behaviors are designed by consulting biological literature and searching animal behaviors, which can be used for a robotic system. An animal model, which accomplishes the wanted behavior, is translated to a more suitable model to be implemented in the robot. Second one is *experimentally driven design that* is also known bottom-up approach. The basic premise is to build up a set of competences and add them to the behavior system after a test process. By repeating this process iteratively, the behavior-based architecture is built. However, the one we have used is the, *situated activity based design* that robot actions are predicted upon the situations in which the robot is located. The design requires a solid understanding of the relationship between the vehicle and its environment. The problem is reduced to recognizing the situation of the robot.

3.2.2. Expressions of Behaviors

Expressions of behaviours could be done by either stimulus-response diagrams where behaviours are represented using stimulus(input)-response(output) blocks or functional notation where behaviours expressed as mathematical functions. But the finite state acceptors (FSA) which is used to describe aggregations and sequences of behaviours during the accomplishment of high-level goals mostly used in Subsumption architectures. They make the active behaviors and the transitions between them explicit. A finite state acceptor M is specified by:

Q : set of allowable behavioural states

δ : transition behavioural configuration

q_0 : starting behavioural configuration

F : set of accepting states

3.2.3. Behavior Coordination

Coordination method used in subsumption architecture is priority-based. Behaviors organised hierarchically into layers that behaviors at higher layers have the priority by suppressing outputs of lower layers.

3.2.4. Behaviour Encoding

To encode the behavioral response it is necessary create a functional mapping from the stimulus plane to the motor plane. A behavior can be expressed as (S, R, **β**) where:

S: Stimulus Domain. S is the domain of all perceivable stimuli. In our case it is range values from range-sensors and info from cameras.

R: Range of Responses. In our case responses are velocity of left-wheel and right-wheel

β: Behavioral Mapping. For each behavior we have defined a mapping function **β** relates the stimulus domain with the response range by using fuzzy logic: **β**(s)→ r.

4. HARDWARE DESIGN

The hardware design of autonomous mobile robots capable of intelligent motion and action involves the integration of many different bodies of knowledge. This includes mechanics, kinematics, dynamics, perception (sensor suites), computation, interaction (media devices), power & control system. The aim of hardware design is to make a preliminary work for the realizable autonomous museum tour guide robot and getting the parameters for the reliable simulation.

4.1. General Design Issues

There is no unique approach or methodology to design a system. There is not a unified approach to creativity. Given a particular need, each individual designer would probably design something different [27]. There are however, some common guidelines which can be useful in general. We have accomplished the hardware design under the guidance of following issues:

Reliability: We should consider the entire equipment: the robot will possess, to be able to achieve its goals under any circumstances.
Simplicity: We should avoid the unnecessary complexity that can arise during the design process
Availability: Proposed components should be readily available off the shelf
Empirically Successful: Proposed equipments have to be used in a similar projects that their success must be undisputable.
Cost Effective: Price, size, power consumption, weight and maintenance cost of the proposed equipment should be as low as possible.
Generality: Design process could be used in any other mobile robot application.
Modularity: Every component should have its own design. As much as possible solutions should be stated that particular design solutions could be alterable when necessary.
Alterability: Instead of directly addressing unique, specific equipment, we should keep the design open to any further changes.

4.2. Requirement Analysis

Several different behaviors are expected from our museum tour guide robot. First of all, it will safely navigate inside the museum. The exhibits, visitors and the robot itself must always be safe. For this reason, a robust stable locomotion mechanism, actuators to move robot in any condition, necessary sensors and a program works in real-time are required. Secondly, given an initial point, the robot should be able to plan, start, guide and finish a tour. The robot should be able to answer the following questions: “Where am I?” and “Where am I going?” at any time. To be able to do it, it needs appropriate sensors, a static map of environment and the necessary software. During the tour it should give info about the exhibits to the visitors. The agent should also be able to adapt any customized paths by the visitors. Then there have to be some media devices and related sensors to recognize visitors. The robot will operate in a manner as long, robust and durable as possible. Human intervention should be minimal. This could be accomplished by battery tanks and a docking station where the robot could recharge itself autonomously. Finally, the robot will appeal people, interact with them and keep the interest as long as possible. The visitors must accept it as a social being. And also, our tour guide robot could able to progress itself autonomously from experience and observation of people’s reaction. These could be done by high level coding, with powerful processors. Mechanics and electronics, to assembly the robot, without any doubt also required.

To be able to find more precise solutions (for the hardware part) to the requirements stated above some variables had to be defined for the museum robot. They are: Environment, maximum speed ($v_{\max}$) robot must have, operation duration (t_{op}) and also weight of the robot (W_{robot}). The required systems and variables have an influence diagram as depicted in Figure 4.1. We will first define the environment where robot will operate. Environment will bring us some constraints that will be useful for further design. The second step is deciding the locomotion mechanism. After this step, for different values of $v_{\max}$ and t_{op} different solutions could be found for the actuators and batteries where they are mostly depend on each other. For this reason we will make a look up table for motors for possible assumptions on weight and continue the design process by deciding on sensors. In this part only the sensors which are used on the tour-guide robots will be explored and the ones we will use be decided. Also they are investigated in detailed manner for the simulation part. The design of possible electronics such as computational resources, interactivity equipment will not be explored in detail, only the power consumptions, weight and dimensions for our reliable design will be considered. Mechanical parts will not be explored. Because, we assume that, Robot will be constructed by a professional company in this area. Finally, batteries will be explored to be able to make a good design. Decision of batteries will be done after making the power analysis on the part-list constructed according to design considerations. After making the cost, weight analysis for possible different solution, we will finally shape the body roughly. However, due to the high probability of the future changes and contradiction between the assumptions we will as little oversize the system deliberately.

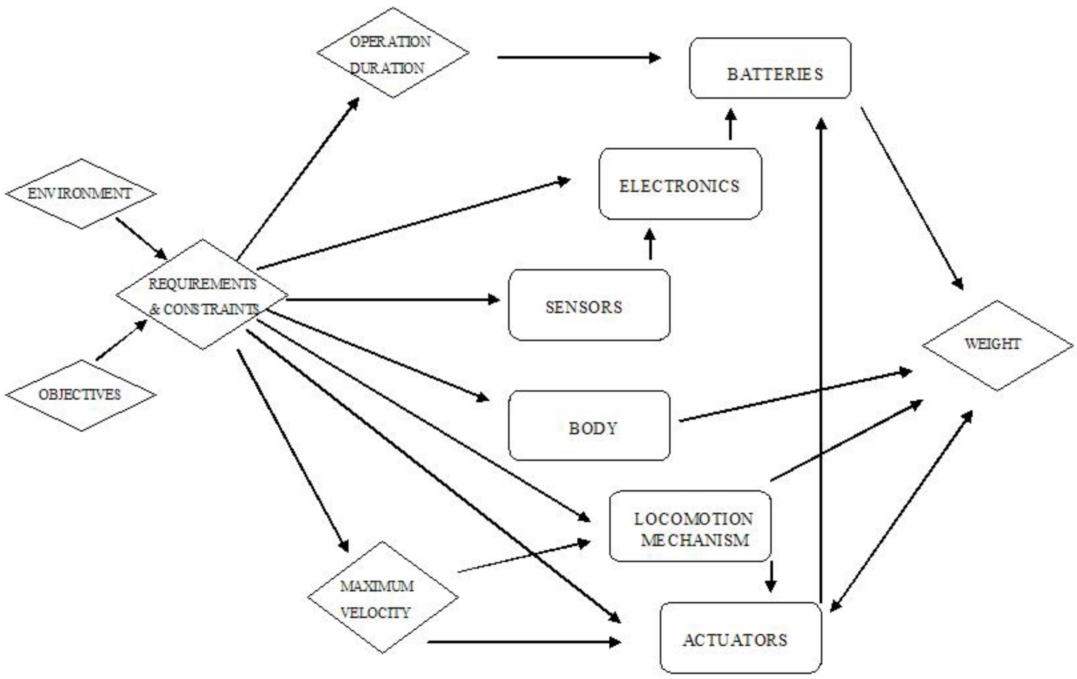


Figure 4.1. Systems and variables for hardware design and their dependencies

4.3. Environment

Museums are environments composed of walls, exhibits in show boxes, gateways, doors and columns and so on. Also the visitors especially children are the other components of the environment that deserves special consideration. In general, the museum environment is partially observable (sensors can not access the complete state of the environment), stochastic (next state of the environment could not be predictable), dynamic (environment continuously changes). However, this info is not adequate for a reliable robot design. There could be special constraints for a specific museum that could certainly change the design. For this reason it is better to address the museum. After a small search in Istanbul, we have decided on the big hall, in the second building of the Rahmi M. Koç Museum, located on the shore of the Golden Horn. The Rahmi M. Koç Museum is the first major museum in Turkey dedicated to the history of Transport, Industry and Communications. Big hall is the place where exhibits related to marine, aviation and rail road transportation are shown.

Big hall of the Rahmi M. Koç Museum is 20 meters wide, 55 meters long. The ceiling is more than three meters high. The floor is concrete and smooth, parceled with faiences that we could able to extract the topological map using the real metric values of the environment. This map could be seen in Figure 4.2. The narrowest path robot must pass through is 130 centimeters. If we give a number for identical structures, we can roughly say that there are ten different structures required to be recognized by the robot. They are:

1 – Entrance footsteps with 30 cm height 80 cm width and 3.5 m length

2 – Walls of rooms with three meters height in different sizes

3 – Columns of the museum with a 35 cm radius 3 m height

4 – Show-box type1 with a 90 cm height 65 cm width 1.7 m length on the four legs with five cm radius and 100 cm height

5 – Show-box type2 with a 70 cm height 60 cm width 1.9 m length on the four legs with 2.5 cm radius and 100 cm height

6 – Show-box type3 with a 65 cm height 50 cm width 1.2 m length on the four legs with 2.5 cm radius and 65 cm height

7 – Show-box type4 with a 90 cm height 50 cm width 1.2 m length on the four legs with 3.5 cm radius and 70 cm height

8 – Train simulators with the same logic as show-boxes 45 cm height 2 m width 4.92 m length on the four legs with 5 cm radius and 80 cm height

9 – Real airplane assumed to be box with 2 m height, 4 m width and 7 m length

10 – They are restricted areas where big exhibits lie on the floor and nobody permitted to enter.

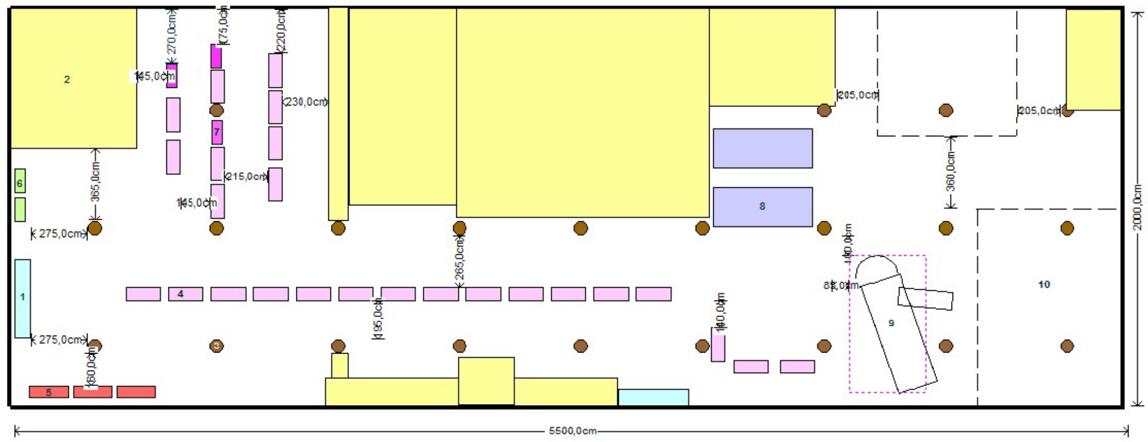


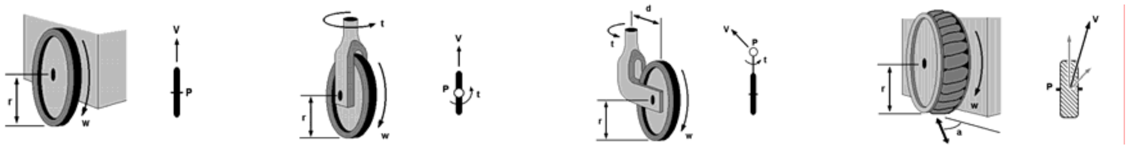
Figure 4.2. Topological map of the big hall in the second building of the Rahmi M. Koç Museum

4.4. Locomotion

A mobile robot needs a locomotion mechanism that enables it to move throughout its environment. There is a large variety of possible ways to move which are adapted to different environments and purposes. Many biologically inspired robots walk, crawl, roll, swim or hover. The types and information could be found in [28, 29]. However, for a museum environment there are two options: Legged or wheeled. Wheels are extremely well suited for the flat and hard surfaces. When the surface becomes soft, wheeled locomotion accumulates inefficiencies due to the rolling friction so that no matter how the legged locomotion mechanically complex, it is preferred. For instance, it could be the solution for a museum with soft carpets. But in our case, wheels are the best without any doubt.

4.4.1. Drive Type and Wheel Configuration

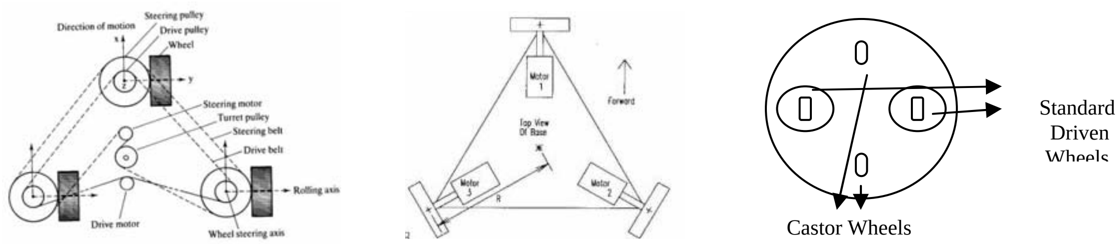
There are four major wheel classes which can be seen in Figure 4.3. (In the Figure ‘v’ shows the angular velocity, ‘w’ shows the rotational velocity). First one is standard (fixed) wheel which has a roll axis parallel to the plane of the floor that degrees of freedom (DOF) of robot motion is two. The second is steered wheel which is the specialized of the standard wheel in that mounted on a rotational link with the axis of rotation that DOF is three. Third one is the castor wheels which has the same type of operation with steered one with a key difference that castor wheels rotates around an offset axis causing a force to be imparted to the robot chassis during the steering. The last one is omni-directional (Swedish), which is mounted like standard but act like castor that has three DOF. Actually the angle of the peripheral wheels may be changed to yield different properties.



a) b) c) d)

Figure 4.3. Types of wheels: a) Standard b)Steered c)Castor d)Omni-direction [20]

The decision of the type, number and arrangement of the wheels has a great importance on the dynamics and kinematics of the mobile robot that affects further hardware and control software. Anyway, if the wheel arrangement designed is a movable configuration it will fall into one of the five different drive systems which are comprehensively stated in [30]. Two of them are tricycle and car-drive (Ackerman Steering). They are non-holonomic drive types and widely used in outdoor environments that they are out of our scope. The other three drive systems are differential, synchronous and Omni-directional. Possible wheel configuration of these types could be seen in Figure 4.4. Synchronous drive, even though there are different typical configurations for this type of mechanism, generally consists of three steered wheels arranged as vertices of an equilateral triangle often surmounted by a cylindrical platform used. In this type of drive system all the wheels are controlled by two motors (two for each or two for all) one for translation one for rotation so that all of the wheels turn and drive in unison. It is a holonomic drive mechanism. Omni-drive systems operate by having individual wheels apply torque in one direction in the same way as a regular wheel, but are able to slide freely in another direction (often perpendicular to the torque vector). This type of drive system is also holonomic.



a) b) c)

Figure 4.4. Drive mechanisms a) Synchro [21] b) Omni- Direction [21] c) Differential

Omni-drive and Synchro-drive robots have the kinematics advantage of allowing continuous translation and rotation in any direction. This advantage is apparent for transport systems in confined spaces and in competitive high-speed environments. However their complex design and implementation, high power requirement and complex control for motors contradicts with our design guidelines. Consequently we have decided to have differential drive. In our museum robot drive mechanism, two fixed driving wheels powered by two motors (one for each) should be used. Two parallel wheels execute straight-lines by moving with the same velocity or curves with differences in velocity. Differential drive system is mechanically easy to implement, cheaper and requires less power consumption. But the robots with this type are non-holonomic. For this reason controllability is more complex. And also since speed and direction are coupled this makes straight line motion difficult. For this reason, two castor wheels should be added for balance. We will assume a wheel configuration depicted in Figure 3.4 c).

Up to now, we have decided to buy two castors and two fixed wheels but the problem is deciding the type. To be able to select the type, we will use the wheel selection table, which could be seen on Appendices A, taken from [31]. It shows the ratings of different wheel type according to the floor surface and its operational characteristic. Also for each wheel there is a unique datasheet which shows the available radius, maximum payload and other parameters of the wheel. After exploring all of them we could say that; polyolefin, cast iron, forged-steel and durasten are for the tough environments that their resiliency is poor and rolling is difficult but very robust and bear heavily load. Urethane on iron has the noise problem in operation. Otherwise it is also good enough for our mobile application. Urethane on plastic has low endurance. Soft rubber small in size but load bearing is only up to 100 kg. Pneumatic wheels and rubber on iron wheels are also excellently fit to our application but high diameter, relatively low load bearing makes them second choice. Urethane on aluminum wheels roll easier than rubber wheels and seems to be the best fit to our museum environment. Available wheel diameters are 10 - 12 cm and width are 3.5 - 5 cm which could bear 400 – 500 kg of load. We will continue our further design taking wheel diameter 11 cm and width 4 cm.

4.5. Actuators

Actuators are mechanical devices for moving or controlling something. Electric motors and drives, hydraulic drives, pneumatic drives, internal combustion hybrids are some examples of it. Lots of useful document could be found at [32]. For our museum mobile robot only available actuators are electric motors that we will focus on them.

4.5.1. Motors

An electric motor converts electrical energy to mechanical energy. Mainly there are two kinds of motors, electromagnetic alternating current (AC) motors and electromagnetic direct current (DC) motors. Because our mobile robot's power supply is typically a DC battery, we will concentrate on DC motors only.

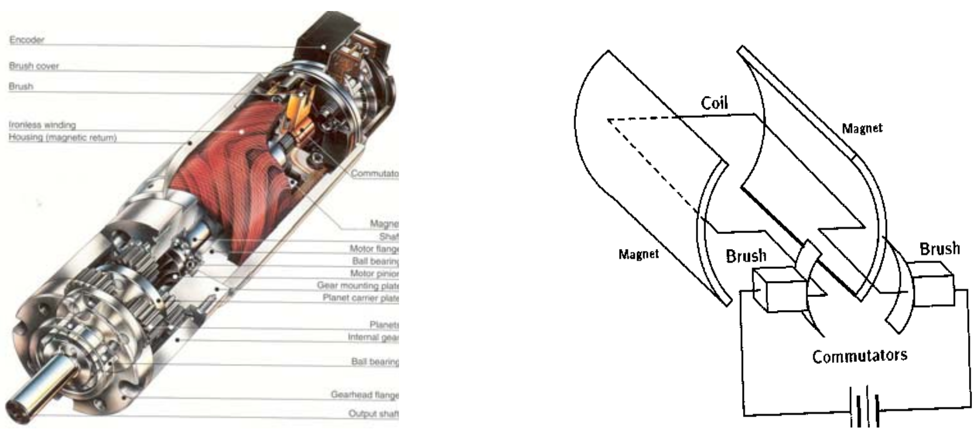
According to the working principle DC motors could be divided into three different groups. 1 - DC motors with two electrical terminals. Applying a voltage across these two terminals will cause the motor to spin in one direction while a reverse polarity voltage will cause the motor to spin in the other direction. The polarity of the voltage determines motor direction, while the amplitude of the voltage determines motor speed. However, some DC motors have more than two electrical terminals, often up to six or eight, 2- Stepper motors, devices that process digital signals in order to obtain a specific angular displacement. As the name implies, the motor accomplishes its displacement by rotating in steps. Each signal input into the motor results in the incremental rotation of the rotor. 3 – Servo motors that use feedback and controllers to achieve a specific angular location. Servo motors can achieve the same levels of accuracy as stepping motors more rapidly, but must use a closed loop feedback system whereas stepping motors operate using open loop systems. Both stepper motors and servo motors are used in applications where precise angular location of the shaft is necessary. They are not preferred for the applications where continuous rotation required.

Because of the drive system we have chosen for our robot is differential drive we will focus on DC motors with two electrical terminals that they are:

1) Series-wound motors: They generate very high torque at time of initial start time but changes in load dramatically change the rotational speed that controlling speed is very hard.
2) Shunt-wound motors: Good speed regulation with slight variations with a changing load but low start torque.
3) Compound-wound motors: The best characteristics of the series-wound and shunt-wound motors, good starting torque and constant operating speed but compared to them they are very expensive
4) Permanent-magnet motors: They are the cheapest motor widely available all around the world with different shapes, size and characteristics. In general they produce low torque with very high speed that could be controlled well. Their main

advantage is they are up to %50 lighter than other motors that best suits to the mobile robot applications.

Basically, permanent-magnet dc motors are based on two components, one is the stator, the stationary outside part of a motor, and other is the rotor, the inner part which rotates the shaft. According to the type of rotor-stator implementation and related working principle, permanent magnet motors categorized by brushed or brushless. In Figure 4.5 a) and b) inside of a full featured brushed DC motor (a gear head and an encoder suit added which will be explored later) and basic motor components could be seen respectively. Here magnets are used as stator and the coils wound by wires (also called windings) on which current flows are used as stator. In motors, keeping the reverse polarity between the magnets and windings result a continuous magnetic flux in one direction that rotation of shaft achieved. To keep the reverse polarity, change of current direction on the windings required. The process of switching current is called commutation. In brushed motors, the brushes move across the commutator contacts and energize the next winding for commutation. However, a brushless DC motor has a rotor with permanent magnets and a stator with windings. It is essentially a brushed motor turned inside out. The control electronics replace the function of the brushes and energize the proper winding. Since brushless motor usually has a longer life in that friction is reduced, also controlling motor is finer and less radio frequency interference produced that we will choose a permanent-magnet brushless dc motor for our museum mobile robot.



a) b)

Figure 4.5. a) Inside view of a full featured brushed DC motor [23] b) Basic motor components of brushed DC motor [24]

4.5.2. Motor Characteristics

There are lots of specifications to define a motor such as nominal supply voltage, armature resistance, efficiency, no-load speed, no-load current, friction torque, stall torque, velocity constant, torque constant, armature inductance... However, to be able to decide specifically which motor to use, we need the torque/speed curve which is specific for every motor. Figure 4.6 shows a torque/speed curve of a typical DC motor. As seen from the graph, torque of a motor (τ_m) is inversely proportional to the speed of the output shaft of the motor (ω_m) that, there is a tradeoff between how much torque a motor delivers, and how fast the output shaft spins. Stall torque (τ_s), is the highest amount of torque a motor can generate, the motor will be stalled with this much load. Free speed (ω_f), is the maximum angular velocity of the shaft of a motor which could be seen when there is no load. For the design of energy supplies, two characteristics of motor are required. The first one is, nominal supply voltage required to run motor and the second is stall current, the amount of current drawn when motor is stalled. It is the maximum amount of current that a motor can require. The final characteristic we will use is the maximum mechanical power of the motor ($P_{M\max}$) it could deliver where P_m is the amount of power delivered from the motor, which depends on the instantaneous torque and speed of the motor.

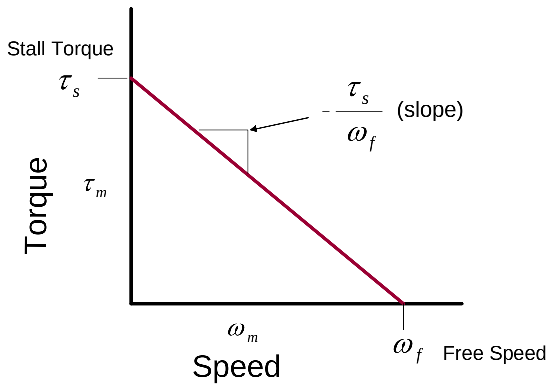


Figure 4.6. Torque-speed curve and related equation

We can easily derive the torque and speed equations of the motor from the Figure 4.6:

$$\tau_m = \tau_s - \frac{(\omega_m \times \tau_s)}{\omega_f} \tag{4.1}$$

$$\omega_m = \frac{(\tau_s - \tau_m) \times \omega_f}{\tau_s} \tau_s - \frac{(\omega_m \times \tau_s)}{\omega_f} \tag{4.2}$$

We know from basic physics that:

$$W = F \times d \tag{4.3}$$

$$v = \frac{d}{t} \tag{4.4}$$

$$P = \frac{W}{t} \Rightarrow P = \frac{(F \times d)}{t} = F \times \frac{d}{t} \tag{4.5}$$

$$P = F \times v = \tau \times \omega \tag{4.6}$$

So combining 4.1 and 4.6 we can derive the power equation for a motor:

$$P_m(\tau_m) = \omega_f \times \tau_m - \left(\frac{\omega_f}{\tau_s}\right) \times \tau_m^{\ 2} \tag{4.7}$$

If we take derivative of $P_m(\tau_m)$ and equate it to zero we could find the τ_m value to maximize P_m:

$$\frac{\partial(P_m(\tau_m))}{\partial \tau_m} = \omega_f - 2\left(\frac{\omega_f}{\tau_s}\right) \times \tau_m = 0 \Rightarrow \tau_m = \frac{1}{2}\tau_s \tag{4.8}$$

By the same way combining equation 4.2 and 4.6, we can derive power equation for a motor $P_m(\omega_m)$ and then the value of ω_m which maximize P_m that result will be $\omega_m = \frac{1}{2}\omega_f$ which is the same value when we take $\tau_m = \frac{1}{2}\tau_s$ in the equation (4.2) that we can conclude max power $P_{M\max}$ could be delivered from a motor when its torque is half of the stall torque and speed is if half of the free speed that from equation 4.6 it is:

$$P_{M\max} = \frac{\tau_s \times \omega_f}{4} \tag{4.9}$$

4.5.3. Gears

As mentioned before permanent magnet DC motors are lacking in torque, or in other words, they cannot push very hard but spin at a rate of thousands of revolutions per minute. However for our mobile robot application we need the reverse of it: high torque, slow spin. For this reason we need gears. A gear is a wheel with teeth that mesh together with other gears. And gears have the ability to change speed, torque and also direction of rotating axles. Even if gears have different types and specifications, we will concern only gear-boxes made of spur gears (could be seen on Figure 4.5) supplied by motor-producers to fit their motors. Gear-boxes characterized by gear ratio such as five to one (5:1) which means for every five turn of motor shaft, shaft of gear-box will turn one times that total system will able to apply a torque five times greater than the motor could do itself. If the ratio is greater to smaller it is called geared down, this is exactly what we want. If it is smaller to greater, then it is called geared up. Hereafter, in our design we will assume that gear box shaft is directly connected to wheels of robot. If say the gear ratio n_{gear} then the output torque of gearbox (τ_m^*) and (ω_m^*)

$$\tau_m^* = n_{gear} \times \tau_m \tag{4.10}$$

$$\omega_m^* = \frac{\omega_m}{n_{gear}} \tag{4.11}$$

Actually, the biggest enemy of any gearbox is friction. Every place where something rubs, energy is lost and this lost grows bigger in time. Even if the gears specialized to maximize efficiency in gear boxes, there is always a lost.

4.5.4. Vehicle Model

We have decided to use differential wheels with two wheels, each will be driven with one motor. Choosing necessary motors is one of the key issues of design part. Because a high overestimate of motors will result in high power consumption, high footprint, more weight and high cost that affects the whole design drastically. On the other side under-estimate of motor will make the whole design useless. To determine the needed power, torque and the gear ratio required we will use a simple vehicle model as depicted in Figure 4.7 stated in [44].

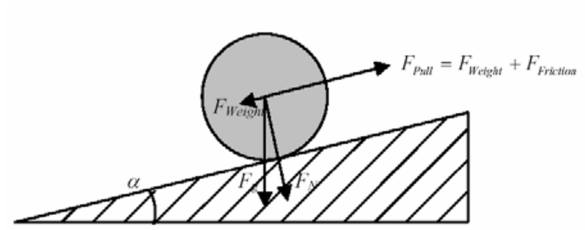


Figure 4.7. Simple vehicle model [24]

Assume a wheeled vehicle moving with a constant velocity (v) on a surface with an incline **α**. Since acceleration is 0 then:

$$F_{Net} = F_{Pull} - F_{Weight} + F_{Friction} = 0 \tag{4.12}$$

$$F_{Pull} = mg\sin(\alpha) - \mu_f mg\cos(\alpha) \tag{4.13}$$

In the equations μ_f is the friction coefficient of a rolling wheel. Required power (P_{Req}) will be then:

$$P_{Req} = F_{Pull} \times v \tag{4.14}$$

4.5.5. Selecting the Motor

Since the motor is directly related to weight and velocity that instead of predicting them we will find alternative solutions for different scenarios. For this reason, we will take the weight of the robot, maximum velocity and surface angle as variable:

Weight of the robot (W_{robot}) (kg) = **{**125, 150, 175, 200**}**

Velocity of the robot (v_{robot}) (m/sec) = **{**0.3, 0.5, 0.7**}**

Surface angle **(**α**)** (Degrees) = **{**0, 5,10**}**

To find out the power we will seek on the data sheets we have used the following equation**:**

$$P_{M\max} \geq \frac{\left(\frac{P_{Req}/e_{gear}}{2}\right)}{0.7} \tag{4.15}$$

First, we divided P_{Req} to e_{gear}, efficiency of the gear box, which depends on the ratio rate. For example, from ratio 5:1 to 20:1 it is generally between 0.9 and

0.95. However when the case is 200:1 then it decreases to 0.5. Then divide the equation into two because we have two motors to supply workload. Then finally to 0.7 because we can not run a motor at 100% power it would probably burn. According to the equation 4.15 and the variables stated above we have constructed the Table 4.1 which shows required motor power for different cases. The required parameters, constants and simplification done based on [34]. As seen from the table in the worse case we will need motors with a max power rating 450 watts which is not a big problem. When we explore the [35, 36, 37] we find out that there are lots of geared dc brushless dc motors with a power rating between 300 to 3000 Watts. They are also small, their radiuses are not more than five cm and lengths are not more than 18 cm with a total weight at most four kg. The problem is their high power consumption must be suspended from the robot for a long time. To find out the gear ratio of the gear-box for our motor, if the motor is not geared, we could use following equation by the same logic above:

$$n_{gear} >= \frac{F_{req} \times r_{wheel}}{0.9 \times \left(\frac{\tau_s}{2} \times 0.7 \right)} \tag{4.16}$$

For the gear ratio instead of purchasing a preassembled one, we will assume that it will be prepared for us to easily adapt for the final robot shape defined and requirements.

Table 4.1. Look-up table for required motor power for different cases

W(kg)	V(m/s)	Angle (deg)	Wheel r (m)	W wheel (rpm)	T(Oz-in)	P Motor Gear Eff 0.9	P Motor Gear Eff 0.7	P Motor Gear Eff 0.5
125	0.3	0	0.055	52	7117	15	19	26
150	0.3	0	0.055	52	8541	18	23	32
175	0.3	0	0.055	52	9964	20	26	37
200	0.3	0	0.055	52	11388	23	30	42
125	0.5	0	0.055	87	19770	24	31	44
150	0.5	0	0.055	87	23724	29	38	53
175	0.5	0	0.055	87	27678	34	44	61
200	0.5	0	0.055	87	31632	39	50	70
125	0.7	0	0.055	122	38749	34	44	61
150	0.7	0	0.055	122	46499	41	53	74
175	0.7	0	0.055	122	54249	48	61	86
200	0.7	0	0.055	122	61999	54	70	98
125	0.3	5	0.055	52	19490	40	51	72
150	0.3	5	0.055	52	23388	48	62	86
175	0.3	5	0.055	52	27286	56	72	101
200	0.3	5	0.055	52	31184	64	82	115
125	0.5	5	0.055	87	54139	67	86	120
150	0.5	5	0.055	87	64966	80	103	144
175	0.5	5	0.055	87	75794	93	120	168
200	0.5	5	0.055	87	86622	106	137	192
125	0.7	5	0.055	122	106112	93	120	168
150	0.7	5	0.055	122	127334	112	144	201
175	0.7	5	0.055	122	148557	130	168	235
200	0.7	5	0.055	122	169779	149	192	268
125	0.3	10	0.055	52	31715	65	84	117
150	0.3	10	0.055	52	38057	78	100	140
175	0.3	10	0.055	52	44400	91	117	164
200	0.3	10	0.055	52	50743	104	134	187
125	0.5	10	0.055	87	88096	108	139	195
150	0.5	10	0.055	87	105715	130	167	234
175	0.5	10	0.055	87	123334	152	195	273
200	0.5	10	0.055	87	140953	173	223	312
125	0.7	10	0.055	122	172668	152	195	273
150	0.7	10	0.055	122	207201	182	234	328
175	0.7	10	0.055	122	241735	212	273	382
200	0.7	10	0.055	122	276269	243	312	437

4.6. Mechanics

Actually not only the gears all the mechanics assumed to be prepared for us by a company. These mechanics are: Whole body (skeleton) construction, Transmission mechanism from motors to wheels and housings for all equipment Bearings, screws, hubs…

4.7. Sensors

Robots are no different than humans that without perception of environment, they are merely machines, incapable of adapting to any change in the environment. Sensors give a robot the ability to collect information about the world around it and to choose an action appropriate to the situation. Sensors are merely transducers that convert some physical phenomena into electrical signals that the processors can read. So while designing a sensor suite for a robot, we have to decide, first what the robot must recognize around the environment according its duties then make a selection among possible ones to achieve it.

4.7.1. Position and Orientation

The first major duty of a museum tour guide robot is simple is to be able to guide a tour. To be able to achieve the robot must always know its present location and direction of itself and also the positions of the objects around and paths. An environment model with different degrees of abstraction could be established offline, however, the problem is, matching this model with the one gathered while working online and estimating the location of robot. We can say that there is no truly unique solution for the problem. However, there are many partial solutions that can be grouped and categorized in [21, 29] as follow:

A) Relative Position Measurements (also called Dead-reckoning)

Odometry: It is based on advancing some previous position and direction through known course and velocity information over a given length of time. In wheeled mobile robots rotational displacement sensors derive navigational parameters directly from wheel rotation.

Inertial Navigation: It is developed for deployment on aircraft but also adapted for use of missiles and nuclear submarines that depends on continuous sensing of minute accelerations in each of the three directional axes, and integrating over time to derive velocity and position.

Doppler Navigation: It is employed in maritime and aeronautical applications to yield velocity measurements with respect to the earth itself

B) Absolute Position Measurements (Reference-based systems)

Active Beacons: In this type of navigation systems there are usually three or more transmitters mounted at known locations in the environment and one receiver on board the robot. Conversely, there may be one transmitter on board and the receivers are mounted on the walls. Using time-of-flight information, the system computes the distance between the stationary transmitters and the onboard receiver. Active beacons can be detected reliably and provide accurate positioning information with minimal processing. As a result, this approach allows high sampling rates and yields high reliability, but it does also incur high cost in installation and maintenance.

Global Positioning Systems (GPS): These systems are revolutionary technology for outdoor navigation. They are based on the satellites, which are transmitting encoded RF signals from the space and a receiver on the ground. These RF signals include information about the satellites momentary location. Then finding out the latitude, longitude and the altitude of the receiver becomes computation of the distance between the satellites and receiver that could be calculated easily by travel time of the RF signals.

Landmark Navigation: It is based on recognizing the landmarks that are distinct features in the environment and whose location is known by robot. Landmarks can be geometric shapes (e.g., rectangles, lines, circles), and they may include additional information (e.g., in the form of bar-codes). There are two types of landmarks: “artificial” and “natural” landmarks. Natural landmarks are those objects or features that are already in the environment and have a function other than robot navigation; artificial landmarks are specially designed objects or markers that need to be placed in the environment with the sole purpose of enabling robot navigation. Before a robot can use landmarks for navigation, the characteristics of the landmarks must be known and stored in the robot’s memory. The main task in localization is then to recognize the landmarks reliably and to calculate the robot’s position.

Most of the landmark navigation systems are based on computer vision and to be easily identified by the robot, there must be sufficient contrast between the landmark and the background. For artificial ones it could be designed to reach optimum recognition. For natural ones generally long vertical edges, doors, wall junctions and ceiling lights are chosen that could be recognized easier by a camera. Range sensors can also be used for landmark navigation to identify corner or edge or long straight walls. Or if we use a bar-coded reflectors as an artificial landmark than the corresponding sensor would be a laser scanner.

Map-Based Positioning: This technique is based on comparing the map, created locally by the current data from the sensors of the robot, with the global map of the environment. Global map includes all the operating area reachable by the robot and stored in the memory. This global map could be either extracted by robot itself or uploaded to memory exclusively.

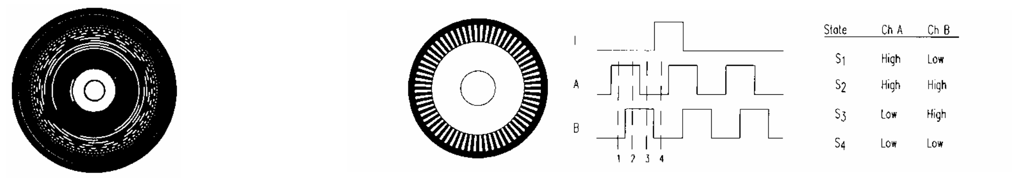
According to the previous work on museum robots we will focus on the sensors used to position by odometry, landmark navigation or map matching.

4.7.1.1. Encoders

A common means of odometry instrumentation involves encoders directly coupled to the motor armatures or wheel axles. Other widely used odometry sensors are: Potentiometers best fit with low-cost, low-speed, medium accuracy applications not involving continuous rotation. They are widely used at the manipulator joints. Synchros and Resolvers best fit with high cost, high accuracy, high power applications that not widely preferred in mobile robotic applications. The best odometry sensor that fits for our wheeled museum tour guide robot is encoders.

There are several types of encoders. We can divide them into different categories. The first is, according to type of application they are used. 1 – Rotary: Used to determine angular position of a rotating actuator and 2 - Linear: used to convert linear position information into an electrical output signal. The second is according to the working principle, 1 – Optical: usually either a LED or a stimulated emission of radiation (Laser) is projected through thin slits in a rotary disc. The beam of light from the emitter is matched by a photo-detector on the opposite side of the rotating disc. The coded opaque/transparent pattern on the rotating disc results with a digital encoded electrical output at the detector side that used for position estimation. 2 – Magnetic: For the magnetic encoders, instead of a light emit-detect pair a read head and a special magnetic disc used. The read head contains a magneto resistive sensor, which is basically an inductor that detects changes in the magnetic flux. The disc is magnetically coded. The magnetic code is interpreted by the sensor as a series of on and off states.

Finally, we can divide encoders into two different categories according to the type of information sent as an output from encoder that are absolute and incremental which are both ideal for our robot. Absolute encoders output a digital word that represents a specific angular location of the disc. This makes each angular position unique that even in case of a power failure the actual position will be transmitted to the evaluation electronics. They are characterized by the resolution. There are different options from eight bit to 13 bit in the market. An eight bit encoder ring could be seen in Figure 4.8(a), however an incremental encoder outputs a number which is the total counts of detected light on the receiver part. Incremental encoders usually, will come with three channels, as depicted in Figure 4.8 (b), referred to as A, B, and Z. A and B are placed 90 degrees out of phase which are called quadrature. With these two channels, the processor determines the distance traveled by the number of steps, and the direction traveled by the leading wave form. The third channel is the reference. Usually the Z channel will have only one pulse per revolution or per length of the encoder, so it can be used to determine an actual location, rather than just an incremental number. In the market there are encoders whose resolution differs from 96 to 2045 pulses per revolution. We will also use incremental encoders. Because if an error occurs in absolute encoders such as a bit value read wrongly, this will strictly affect results. Availability of incremental ones are better. For this reason we will use an incremental encoder with a resolution 1000.



a) **b)**

Figure 4.8. a) Absolute encoder ring b) Incremental encoder ring and working principle

4.7.1.2. Angular Rate Sensors

Very common sensors to this type are compasses which measure the orientation of the earth's magnetic field (most often only the horizontal component) relative to the vehicle and hence can be used to estimate the attitude of the vehicle. There are many compasses for different applications in the market, specified by their resolution 0.1 degrees to even 15 degrees. However, this kind of sensors is very susceptible to local variations in the ambient magnetic field. Calibration is therefore essential. Also, compasses are known to be of limited value close to large metal objects and machine equipment that is not preferred much in indoor applications. An alternative to compass is a gyroscope, a sensor for measuring orientation, based on the principle of conservation of angular momentum. There are different types, commercially available, according to their implementation. Mechanical and optical gyros had been widely used in airplanes, ships, helicopters, racing cars... After the invention of the fiber-optic gyros (FOG), which makes them very small and cheap that for the mobile robot builders, they have particular importance for navigation to compensate odometry errors. Like all of the types, optical-gyros are designed to measure angular rate using the Coriolis force (if an object is moving in a straight line, and it is subject to a rotation, you will see a deviation from the original straight line due to the Coriolis force). Working principle of FOG is depicted in Figure 4.9. The laser split beam of light into two that is perfectly in phase. And a detector recombines it with adding two beam of light. If the FOG sensor is rotated, the laser-detector moves a tiny amount. This shortens the distance one beam of light travels, and lengthens the distance the other beam of light travels. When the two beams are recombined, they will form an interference pattern that depends on the rotation rate. For example if light remains in phase, there will be a constructive interference, which makes a bright spot; if the light is out of phase, there will be a destructive interference, which makes a dark spot. FOGs are sold with control circuitry that the name in the market is angular rate sensors. They are specified by the detection range of from ﬃ 75 °/s to ﬃ 300 °/s and sensitivity from five mV to 15 mV. Another important feature respects consideration of them is one gyroscope can only sense the angular deviation of one axis. Therefore we will need two gyroscopes, for x and z axis.

Figure 4.9. Working principle of a fiber optic gyro (FOG)

4.7.2. Range Sensors

There are many different ranging techniques that use different sensor technologies. The most commonly used technology is time of flight (TOF). Based on the detection of reflected pulse of transmitted signal and time measurement between emitted and received. The second frequently used methodology is triangulation ranging technique is based on the simple trigonometry that says when a base and two angles are known the third point can be calculated. The third one is, phase-shift measurement and frequency modulation, generally used in outdoor navigation, a continuous waveform is transmitted towards a target. The phase shift between transmitted and reflected waveform is proportional to distance between the target object and the source. There are also other possible application specific ranging techniques which are: Interferometry, swept focus and return signal intensity.

Here, we will focus on TOF sensor technologies which are widely available, reliable and without any exception, used in all tour guide robots.The first one is infrared (IR) distance sensors. It is composed of three main parts; IR emitter + focusing lens + position-sensitive detector. As shown in Figure 4.10. IR rangers are specified by the minimum and maximum distance they can measure. There are different alternatives in the market such as 10 – 80 cm or 4 – 30 cm or 20 – 150 cm. They return the measured distance value when they are enabled by the processors. Since they are simple, cheap and reliable sensors that without no doubt we will use the ones which has a range 10 – 80 cm

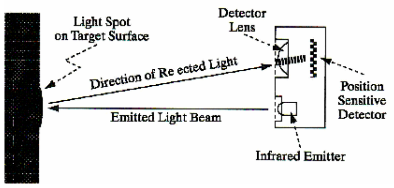


Figure 4.10. Infra-red ranging sensor

Another sensor technology widely used is sound wave sensors (Sonar). It's working principle is the same as IR based on a transmitter and receiver part, the only difference is the signal, sound, that is transmitted and received. They are also specified by min-max distance they measure. There are alternatives from 3 cm – 6m to 20 cm – 10 m. Another specification for the sonar is beam width of the transmitted signal. It is twice of the beam angle of sonar specified in datasheets. Beam width determines the sensing arc of the sonar. In Figure 4.11 beam angle of a sonar sensor is shown and relative effects on the arc of detection specified. There are different models from 15° to 45° in the market. So we can locate 16 sonars with a beam width 22,5° or 24 sonars with a beam width 15° to achieve 360° environment scan of the robot.

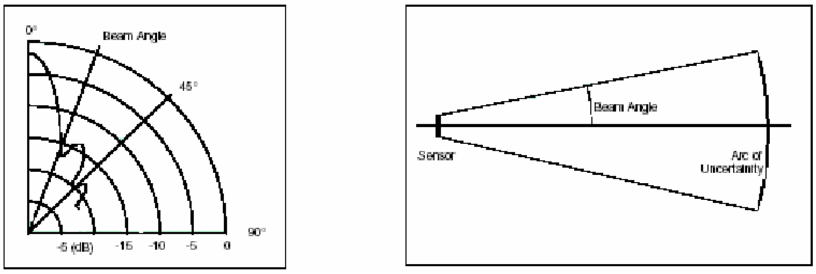


Figure 4.11. Beam-width of a sonar and relative effects on arc of detection

Although there are range sensors radio detection and ranging sensors (RADARS) based on the radio frequency (RF) signals, the last sensor technology will be light amplification by laser range finder or scanners. Laser scanner is a sensor that measures distances to surrounding profiles in a two dimensional plane. The laser energy is emitted in a sequence of very short bursts over a scanning angle of 100 to 180 degrees. Most common laser sensors used mobile robotics is the one which scan 180 degrees with 0,5° resolution, one centimeters accuracy. Beam width is not more than 1° degrees. They can read from 40 cm to 50 m or even 150 m. Scan times differ 13 to 40 msec. Laser scanner needs 200 ns to detect an object 30 m far away where this is 200 ms for sonar. However, the show-boxes in the museum have transparent showcases that sonar sensors are better for this situation. On the other hand laser scanners are very expensive and scarcely available that contradicts with our guidelines. We will use two rings of sonar sensors where each ring includes 24 pcs.

4.7.3. Tactile and Proximity Sensing

Robot has to avoid collisions with surrounding obstacles not only for its safety but also for the safety of people and objects around. Tactile sensors are based on direct mechanical contact that they are the last-resort indication of collisions with the surrounding obstructions. Even if there are many different types of force sensors and tactile feelers according to technology used to feel touch, according to size and appearance. Typical one for mobile robots is a bumper into which momentary switches are attached. The momentary switch and the bumper implementation showed in Figure 4.12. Working principle of switches are very simple. Vcc, supply voltage, usually represents a logic one, and grounf (GND) is zero in software coding. So that in normal position switch sensors are connected to the ground and when the sensor is activated Vcc connection established that Vcc voltage could be read.

Figure 4.12. Switches and their implementation as a bumper

Proximity sensors sense a signal from the target which is near. They do not give information about the range. There are many examples of these sensors in the market with different type: Magnetic, piezoelectric, capacitive, photoelectric, magnetoresistive, pietzoresistive, optical, microwave... etc. Since we use ir range finders for sensing the obstacles nearby we will not use any proximity sensors. But we will use a bumper which will consist of a 12 contact sensors.

4.7.4. Proprioceptive Sensors

The Robot's awareness of internal state is as much important as the external world. Any kind of circuitry could be used to detect internal state of the robot such as; Battery level sensing, by sensing its battery voltage, a robot can determine the time to return to the charging station or curtail power-draining operations. Stall current sensing, used to drive motors go maximum when the robot get stuck, wheels do not rotate. Temperature sensing, it is good idea using discrete temperature sensors to avoid the electronics, motors and batteries get too warm, which will be harmful.

4.7.5. Vision

Machine vision utilizes cameras, model of biological eyes, mounted to robot to provide any kind of information about the environment. Machine vision is such a challenging topic that it has historically been a separate branch of Artificial Intelligence (AI). Light, scattered from objects in the environment goes through an iris/lens, impinges on the image plane (a photo-sensitive pixel array) and then a matrix data is further processed. Vision sensors are generally specified by pixels (resolution) and frame rate. Cameras could be used as range sensors, color tracking sensors, motion or optical flow sensors.

4.8. Electronics

The robot will need computation resources to be able to achieve anything. Since the museums are dynamic environments processing must be as rapid as possible. General procedures for museum robots are using two notebooks both for redundancy and share of workload. One is for high level operations such as mission planning, path planning and vision and so on. Other is for interfacing with sensors and obstacle avoidance.

The robot will also need some equipment for interaction with people:

- A monitor for visual info
- Speakers for audial info
- An electronic or mechanic face to show emotional mood
- Some control mechanism for visitors such as buttons
- There will be also driver boards, interface cards and other necessary circuitries between processors and other component's hardware drivers.

4.9. Batteries

Simply, a 'battery' is an electrochemical device that converts chemical energy into electricity. Since any gas combustion is not allowed, batteries are the only solution for the problem of energy storage for indoor mobile robots. From our robot a long autonomy expected that design on batteries have a crucial importance.

4.9.1. Type of Battery

There are various batteries with different chemistries, however practical considerations dictate that a heavy mobile robot application will use lead-acid cells [30, 31]. Because, for our mobile robot application the batteries:

1) Must be rechargeable that we can eliminate all primary (disposable) batteries which are: Zinc carbon, heavy duty zinc chloride, alkaline (primary types), lithium, silver, mercury oxide, zinc air used mostly in house-hold items such as flashlights, toys, radios, recorders, watches…

2) Must have high capacity (stored energy) and high discharge rate (amount of current when discharging) because of the high amount of power requirement and because of the fluctuating current characteristics of the motors. For this reason we can not use li-ion batteries even if which they have the best energy density (max. amount of energy per unit mass)

3) Must have long cycle time (number of recharge/discharge cycles that battery can sustain) to lower the maintenance cost of the batteries that we can not use alkaline (rechargeable type) batteries which have the worse cycle time.

4) Must have no memory-effect (losing of the discharged part when starting to recharge before totally discharged). At this stage we eliminate Nickel cadmium (Ni-Cd) and Nickel metal hydride (NiMH) that have the disadvantage.
5) Must cost low and must be widely available. Actually only these two features are enough to decide for lead-acid batteries in that they are the cheapest and widely available with various size and capacity all around the world. They are mostly used in cars, cordless electric lawn mowers, electrical transporters, forklifts, mobile robots and telecom applications. Long charge time, heavy and somewhat huge size, sensitiveness to environment (performance degradation at very hot and very cold climates) is the disadvantages of lead-acid batteries we have to bear with. In Table 4.2, comparison of three different types, widely available batteries could be used for an application, which requires 12V 5A/h for six hours. Data is gathered from the datasheets of manufacturers [32, 33, 34].

Table 4.2. Comparison of NiCd, li-ion and lead acid batteries

Battery	**Dimensions(H*W*D cm)/ Volume**	**Weight**	**Price(euro)**	**Cycles(to %80)**	**Charge Time**
10 pcs 1.2V 25A/h NiCd	10*(26*12*8) / 24960	18 kg	350	1500	1 hour
5 pcs 14V 6.6 A/h Li-ion	5*(2*12*8) /1500	2,5 kg	750	1000	3 hour
1pcs 12V 33A/h Lead acid	15*13*20 / 3900	11 kg	50	2000	12 hour

4.9.2. Sizing the Battery

The easy part was deciding the battery type. The difficult part is, deciding the capacity and the number of batteries that will be used. When we examine the datasheets of lead-acid batteries, two tables, discharge characteristics under constant current and power are given which is shown on Table 3.3 and 3.4 taken from datasheet of a 12V 33A/h from the 6FM33 model of Vision company [35]. A battery is characterized by the cut-off potential ($Vcut$) at which the cell is considered discharged. This must be decided by the user which must be between 1.6V – 1.8V. We will use 1.70 Volts. Since there are 6 cells in a 12V lead-acid battery that when voltage is 6 * 1,7 = 10,2 V we will assume battery is drained. Actually the battery could continue to supply till decreasing to 6 * 1,6 = 9,36 V however if they are forced to dis-charge after this level, deep-discharge occurs and the batteries fail in other words become useless. In our further design we will consider the discharge under constant power tables stated in the data sheets and only the data given for the $Vcut = 1.7$V.

Table 4.3. Time table of discharge under constant current power of 12V 33A/h lead acid battery [35]

Discharge Constant Current (Amperes at 77°F25°C)

End Point Volts/Cell	5min	10min	15min	30min	1h	3h	5h	10h	20h
1.60V	115	73.9	57.7	34.4	21.6	9.08	6.10	3.30	1.70
1.65V	109	70.2	54.8	33.3	21.4	8.89	6.04	3.30	1.70
1.70V	101	66.5	52.6	32.3	21.1	8.70	5.93	3.25	1.70
1.75V	93.4	62.8	49.7	31.2	20.8	8.50	5.82	3.20	1.65
1.80V	86.2	59.1	47.6	30.9	20.5	8.31	5.68	3.15	1.60

Table 4.4. Time table of discharge under constant power of 12V 33A/h lead acid battery [35]

Discharge Constant Power (Watts at 77°F25°C)

End Point Volts/Cell	5min	10min	15min	30min	45min	1h	2h	3h	5h
1.60V	215	150	112	71.8	53.6	43.7	24.2	17.7	11.7
1.65V	200	142	108	69.6	52.5	43.0	23.8	17.5	11.6
1.70V	185	133	103	67.5	51.4	42.3	23.3	17.3	11.5
1.75V	170	125	99.2	65.3	50.4	41.5	22.9	17.0	11.4
1.80V	160	116	95.0	63.0	49.3	40.8	22.4	16.8	11.4

Batteries are composed of cells. The cells are small batteries. A 12V battery is a serial connection six cell with a voltage rating two volts. In our robot we will use 12V batteries which are widely available. Therefore we have collected the data, discharge under constant power at $Vcut = 1.7V$ from all datasheets and then multiply them with six to create Table 4.5. It shows how many minutes does a particular battery supply a fixed load.

Table 4.5. Battery discharge time under constant power

Battery Type		5 min	10 min	15 min	30 min	45 min	60 min	120 min	180 min	300 min
VOLT	AH									
12	7	289	193	155	86	65	51	29	21	14
12	9	389	257	187	111	82	65	37	25	18
12	12	491	340	272	157	121	95	50	38	25
12	17	738	485	390	223	172	138	77	54	36
12	20	852	606	487	268	206	162	90	62	42
12	24	966	642	481	296	218	173	110	75	49
12	33	1110	798	618	405	308	254	140	104	69
12	40	1290	930	744	482	363	302	160	125	83
12	65	2022	1512	1182	708	567	463	280	199	133
12	80	2340	1800	1476	906	702	576	323	223	162
12	100	2868	2188	1830	1098	822	726	407	291	205

For our design we need a function which will take the required power as an input and outputs a table, which shows how many batteries you need, for each type to supply required power. For this reason we have to first approximate a function for each battery. According to points for 12V 33A/h, the approximated function is $y = 150379x^{-1,\ 4388}$ which could be seen in Figure 4.13. But this function overestimates for smaller values which could be seen in Table. 3.4. For this reason we divide the points at 60 minutes and define two approximate functions for 12V 33A/h that they are: $y = 777564x^{-1,\ 6949}$, $y = 56433x^{-1,\ 2393}$ can be seen in Figure 4.14. Since we have gather better very good results, which can be seen on Table 4.6 that we have found two approximate functions for each battery. Another good point is two function approximations have a small error and generally the error is due to the underestimation that would not be a problem for us.

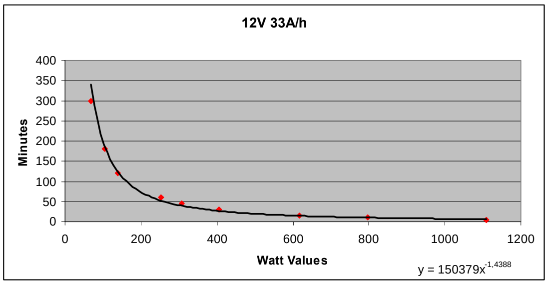


Figure 4.13. Approximate function and real values for 12V 33A/h

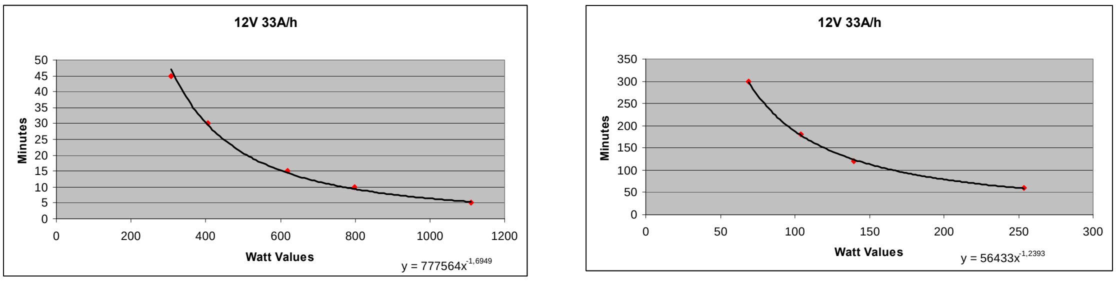


Figure 4.14. Two approximate functions and real values for 12V 33A/h

Table 4.6. Comparison of two approximations

Watt Value	Real Time(min)	Approximated time with one function	Error	Approximated time with two function	Error
1110	5	6	20%	5	0%
798	10	10	0%	9	-10%
618	15	15	0%	14	-7%
405	30	27	-10%	30	0%
308	45	39	-13%	46	2%
254	60	52	-13%	58	-3%
140	120	123	3%	123	3%
104	180	189	5%	179	-1%
69	300	340	13%	298	-1%

4.10. Final Analysis

According to the information gained up to now and investigation from [36, 37, 38, 39, 40, 41, 42] we have prepared Table 4.7 where latest innovative, widely available products, their price, weight and power consumption exists. All the data related to products taken from data-sheets. For the power consumption value of the sonar and infra-red range finders on the table we have taken one of each ring since at time t only one of them is active.

Table 4.7. Cost, power and volume analysis of the products of the museum robot

Type	Name	Number	Total Cost($)	Total Weight(kg)	Total Avarage Power(Watt)	Dimensions (mm) H	W	D
Ultrasonic Range Finder	Polaroid6500	48	2.400,00 $	8,88	43,2	10	32	15
IR (Infra-red) Sensor	GP2D05	24	600,00 $	0,48	1,2	15	30	14
Camera	CMUcam1	1	130,00 $	0,27	1	50	57	45
	CMUcam2	1	190,00 $	0,3	1	50	57	45
	Sony Evi-D30 Pan/Tilt	1	1.000,00 $	1,2	11	164	142	109
Face	Led Matrix	1	30,00 $	0,2	1	50	100	20
Computation	TECRA M4-103	1	1.200,00 $	2,9	75	29	332	270
	SATL L20-102	1	1.399,00 $	2,67	96	29	332	270
Gyroscope	ADXRS150	2	60,00 $	0,02	0,06	3	7	7
Screen	Mini TFT LCD	1	459,00 $	1,5	18	229	251	99
Encoder	TRD-NH2500-RZWD	2	328,00 $					
Speech	SP03 Text to Speech Synthesizer	1	102,00 $	0,22	0,5	37	37	
	Exper Speakers	1	30,00 $	1,5	10	250	60	150
Bumper	Switch Sensors	12	24,00 $	1,2	0,12	15	7,5	20
Keypad	Daventech 3x4 Matrix	1	10,00 $	0,05	0,01			
Electronic Boards	Motor Controller	2	580,00 $	0,3	2	25	50	70
	H-Bridge	2	550,00 $	0,4	2,4	25	50	75
	Encoder Driver Board	2	90,00 $	0,1	0,4	25	60	60
	Sonar Multiplexer(3 units)	8	480,00 $	0,8	4	25	45	50
	Infra-Red Multiplexer(6 units)	4	240,00 $	0,4	2	25	50	50
	Camera Interface Boards	3	360,00 $	0,24	3	30	70	100
	Charger Board	1	200,00 $	0,1	5	30	80	200
	Power Distribution	1	250,00 $	0,15	2,5	30	60	250
Housings, Screws ,cables	For all material		1.500,00 $	20	0			
Wheels	Castor with suspension	2	70,00 $	1,2	0	r = 55 length= 40		
	Polyurethene	2	40,00 $	1	0	r = 55 length= 40		
Mechanics	Body, transmission,screws,housing		5.000,00 $	50	0			
Docking Station	Triton Battery Charger	1	130,00 $					
	Station Construction	1	1.000,00 $					
Further Development	Anything		1.000,00 $	10	15			
	SEMI - TOTAL		19.452,00 $	106,08	294,39			
Motors	200 Watt Power (Leeson M11200)	2	240,00 $	8	300	r = 60 length= 200		
	SEMI - TOTAL2		19.692,00 $	114,08	594,39			
Batteries	12V 80A/h Vision	4	260,00 $	96	0			
	FINAL TOTAL		19.952,00 $	210,08	594,39			

On Table 4.7, there are two semi totals for design of batteries and motors. In the first semi total we look at the weight of the robot which is 106 kg. Then multiply it by two to approximate the weight of the whole robot with batteries. This is the heuristic we have developed during the investigation of commercially available mobile robot bases. For this reason we have taken the weight of the robot 212 kg. Actually there are no slopes on the operation area. But there would be a small gradient that can not be recognized by eyes that taking the slope 5° is a good approach. For the speed of the robot 0.5 m/s or 0.7 m/s is good enough. However, it seems that we will force the producers of gears to keep efficiency at least 0.7 that maximum power rating of two motors required for our robot is 200 watts according to the Table 4.1 created before.

There is another issue related to motor power of the robot we must consider in software design. We have designed no mechanical brakes on the robot that will be stopped by the motors which will be forced to turn reverse according to direction of motion. For the designed motors above, we can construct the Table 4.8 by using basic kinematics equations for motion. It shows the maximum required time to stop and maximum distance traveled by the robot during the stopping time. They are said to be maximum because any small slope of the floor or small frictions ignored which decreases the required time and traveled distance. So according Table 4.8, the velocity of 0.3 m/sec seems the best nominal velocity to move around in the museum which requires a 3.4 cm free distance to stop. However, the robot could be programmable to run up to 0.7 m/sec based on the info from range sensors .

Table 4.8. Stop time and distance requirement for the museum robot

		Max. Time Required (sec)	**Max. Distance Travelled To Stop (cm)**
Velocity of Robot (m/sec)	0.1	0.08	0.4
	0.2	0.15	1.5
	0.3	0.23	3.4
	0.4	0.30	6.0
	0.5	0.38	9.4
	0.6	0.45	13.5
	0.7	0.53	18.4

To calculate the batteries, we first decided the operation time for the robot. Rahmi M. Koç Museum is open between 10:00 – 17:00 during work days and 10:00 – 19:00 during weekend. So at most robot must work for nine hours. We take the operation duration 10 hours to oversize the batteries because batteries will probably loose their power rating after the first recharge. For this situation, the program we developed according to the considerations stated in section 4.9, we get the results which is depicted on Table. 4.9. We have chosen four pieces of 12V 80A/h which weights 96. So total robot weight than is 210 kg that my assumptions on weight seem correct. And then, after calculating cost of batteries, we could say that total cost for the products required for our museum robot will be 19 952 $.

Table 4.9. Cost, power and volume analysis of the products of the museum robot

Battery Type	**Quantity (pcs)**	**Duration (min)**	**Dimensions(D*W*H mm)**	**Total Weight (kg)**
12 V 7 A/h Vision Lead Acid	52	614	151 * 65 * 94	126.4
12 V 9 A/h Vision Lead Acid	45	619	151 * 65 * 94	126.0
12 V 12 A/h Vision Lead Acid	25	604	151 * 98 * 95	97.5
12 V 17 A/h Vision Lead Acid	20	642	181 * 77 * 167	114.0
12 V 20 A/h Vision Lead Acid	21	624	181 * 77 * 167	123.9
12 V 24 A/h Vision Lead Acid	15	629	166 * 175 * 125	129.0
12 V 33 A/h Vision Lead Acid	9	620	195 * 130 * 155	91.8
12 V 40 A/h Vision Lead Acid	7	635	197 * 165 * 170	94.5
12 V 65 A/h Vision Lead Acid	5	678	350 * 167 * 179	117.0
12 V 80 A/h Vision Lead Acid	4	871	350 * 167 * 179	96.0
12 V 100 A/h Vision Lead Acid	3	634	330 * 171 * 215	96.0

5. SOFTWARE CONTROL

The second step after the hardware design is the software design of the robot. There is no need to mention the importance of software that runs the robot. It has to be as reliable and robust as hardware and also able to accomplish all the objects stated in second chapter. The control problem of a museum tour guide robot could be divided into two separate issues, which are interaction and navigation. Autonomous navigation in a structured, dynamic, partially observable, multi-agent environment like a museum is the first problem that can be broken into four steps: 1) Perceiving and modeling the environment, 2) localizing the robot within the environment, 3) planning and deciding the robot's desired path and 4) executing the robot's motion safely in the desired path.

Human-centered and social interactive robotics is a comparatively young field in mobile robotic research. Interaction is simply to acquire the attention of the visitors and communicate with them efficiently in order to give a tour. We could subdivide it into four separate problems for our case: 1) Human recognition, 2) speech synthesis and media playing, 3) emotions expression, 4) face and motion tracking.

5.1. Robotic Architecture

Conventionally the best paradigm for a museum robot we have designed and want to implement is Hybrid Deliberative/Reactive Paradigm. However, the architecture to implement a Hybrid Deliberative/Reactive Paradigm differs according to the application and designer. Robotic architectures are software systems and specifications that provide languages and tools and their interaction between each other in order to control the robot. For a museum tour-guide robot we could build a system depicted in Figure 5.1 to solve the control problem of a museum tour guide robot. There will be five main systems and software modules inside them as building blocks. Communication between the modules inside the same system is allowed. Communication between modules which are located in different systems is achieved by a centralized network of communication. This is because, requires different type of data and the data representations that an interpreter is unavoidable for their communication.

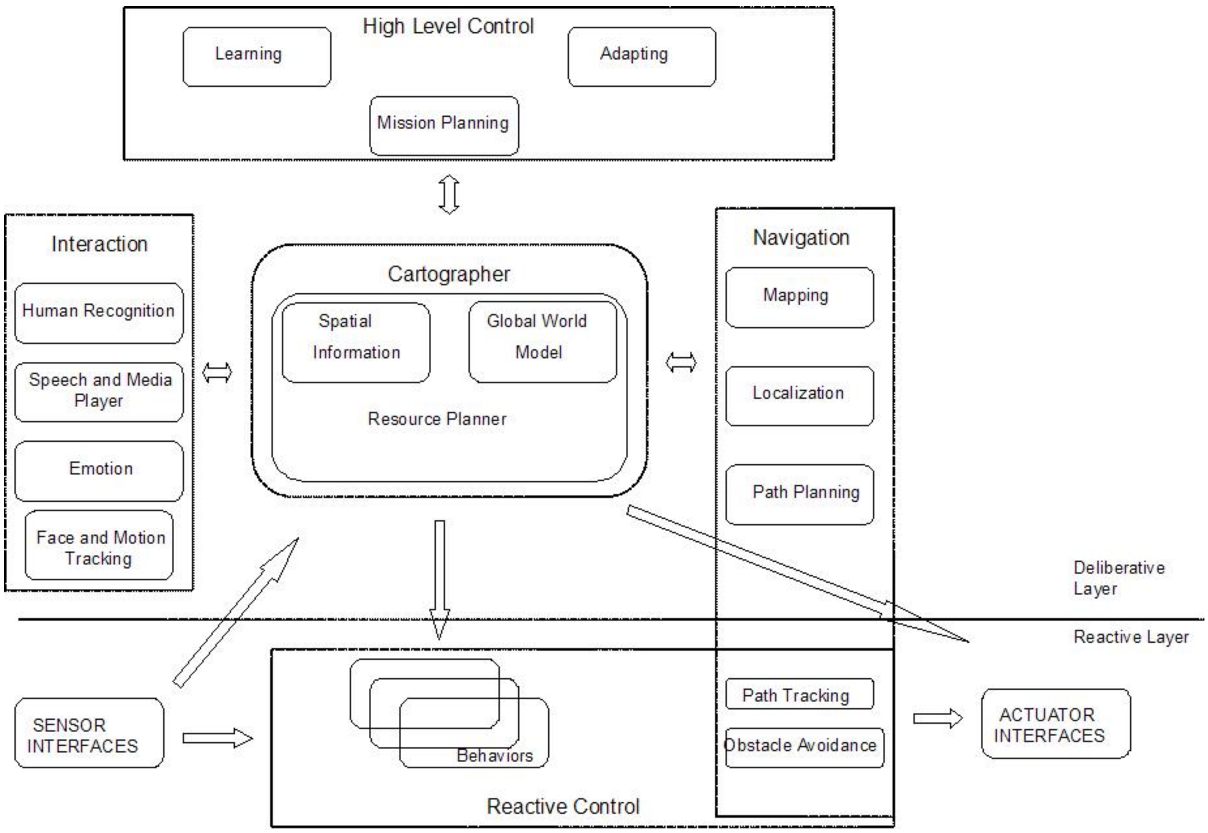


Figure 5.1. Software architecture of museum robot

According to our software design, there is a need for a High-Level Controller which must be composed of three modules: Mission Planning, Learning and Adaptation. The High Level Controller will be responsible for: Constructing a mission plan according the state that robot is in, being aware if the robot making progress or not in that mission, evaluating the results and learn to be more efficient, reasoning and adapting to the changing environment and people, modifying overall behaviors consciously according to the internal and external needs.

The Navigation subsystem which is composed of the modules Mapping, Localization, Path Planning, Path Tracking and Obstacle Avoidance, is the indispensable part of a goal-oriented mobile robot. This subsystem is responsible for: locating people, exhibits and the robot itself on the predefined museum-map, find a free path towards the goal point. The path has to be constituted by considering all dynamic and kinematics constraints of the robot and the tour-mission defined. Path tracking and avoiding dynamic agents around the robot are the last two modules of the Navigation subsystem, however they lie between Deliberative layer and Reactive layer in the architecture. This is because in dynamic environments such as museums, crowded with children, the path planner may not find a collision free path in real-time. However for the safety of all it must be in real-time. For this reason these modules has to be implemented such that they can be run by both components of the reactive layer and deliberative layer.

The interaction subsystem will be able to recognize people around, distinguish them (who is in contact, who is following, who is an obstacle for him in front). The module has to achieve face tracking, motion tracking and logical speeches with people. Telling and playing videos about the exhibits and museum are also done by this subsystem. Keeping an internal emotional mood according to the current situation and expressing it by using via voice and face is also a duty of this system.

The cartographer subsystem is used for storing and maintaining the global world model and spatial information such as videos and speeches. A resource planner module inside the cartographer is required for the data conversion and transmission between the systems. The behavior Controller system is used to run the robot in real time.

All these modules are actually different research areas of AI and Robotics. There are many papers and materials for the implementation of each module that writing a whole software program for a museum robot requires an expert team and time. For this reason in this thesis we will implement only the reactive control layer of the museum robot and assume other modules are working in a way that, we supply the data directly which has to be retrieved from other modules of software.

5.2. Reactive Navigation

Our one of objectives in this thesis: Develop an intelligent navigation system which could find collision-free trajectories in static or dynamic environments containing some obstacles between a start and goal configuration given. Different architectures, to solve such autonomous robot navigation problem, have been proposed. This problem is sometimes called as local path planning, but generally obstacle avoidance in hybrid architectures. State of the art solutions for obstacle avoidance: Bug Algorithms, Vector Field Histogram, Bubble Band, Curvature Velocity, Dynamic Window Approach, Schlegel Approach, behavior-based architectures, fuzzy control and more are presented briefly in [60].

5.3. Subsumption Architecture

We are motivated to use subsumption architecture one of the behavior based architectures discussed in this thesis. Since, it could achieve the whole control of the robot by combining several simple behavior-producing units without any deliberative module. And also it could be an execution layer, in layered hybrid architecture we have designed.

Behaviors used in mobile robots applied in structured, dynamic, indoor environments could be categorized into four groups: *Obstacle avoidance behaviors*, which are intended to avoid collisions with obstacles that are in the vicinity of the robot. It is a mandotory behavior for all behavior based mobile robots that we have defined "avoid-obstacle" behavior to protect robot from any collisions. *Goal reaching behaviors* are intended to drive the robot to the current goal. We have defined "go-to-target" to move the robot to target point successfully. And also two more goal reaching behaviors, "avoid-human" and "wait-visitors" to achieve the mission of robot which is tour guidance. *Alignment behaviors* are intended to maintain the robot along the centre of corridors. Since our robot will follow a path which is defined by a path planner that we do not need any allignment behavior. Path could be tracked reasonably with the cooperation of "avoid-obstacle" and "go-to-target" behaviors. *Escape/emergency behaviors* drive the robot to the left or to the right when it encounters a U-shaped obstacle (local minima) or stops whenever an exception or error occurs for safety. We will use one of the defined behaviors for the states where robots need escape/emergency behaviors such as: stoping which is actually "wait-visitors" or escaping from local-minima which could be accomplished by "avoid-obstacle" . FSA representation of these behaviours could be seen in Figure 5.2. and layered organization of these behaviors according to their priority are depicted in Figure 5.3.

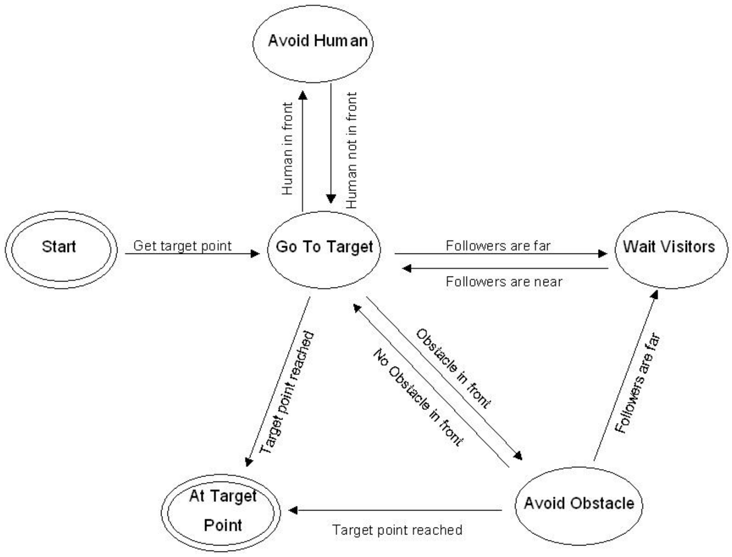


Figure 5.2. FSA representation of the behaviours

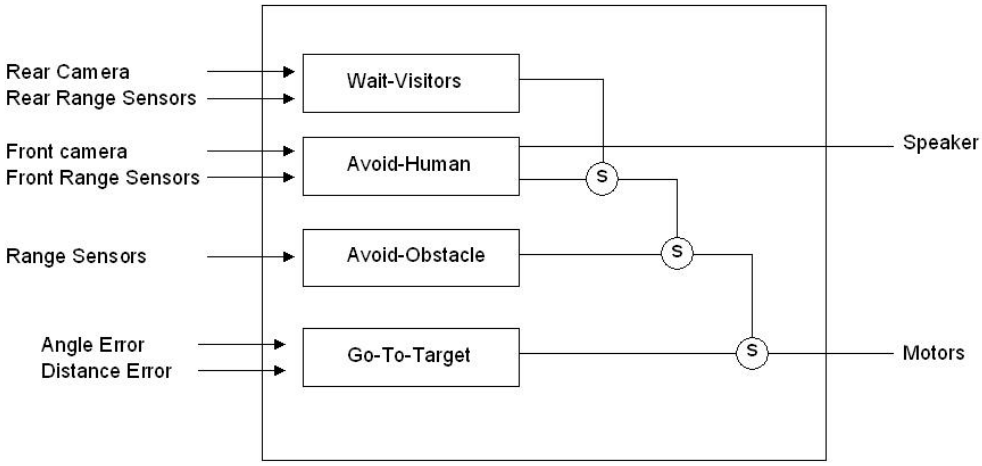


Figure 5.3. Layered behaviors in subsumption architecture

5.4. Fuzzy Control

The reactive navigation by subsumption architecture does not involve any type of world representation or reasoning on them. In response to sensed data, it simply chooses control actions. We have used the fuzzy logic to define this behavior encoding between the sensor space to command space. It requires no mathematical model only an expert knowledge. Beside, easy and flexible structure based on the linguistic variables allows us to add knowledge and update the system at any time. In literature there are several successful examples for fuzzy control of mobile robots [55, 56, 57, 58, 59] stated. In [60] state-of-the-art fuzzy logic solutions are presented and their pros and cons are discussed. However, all the proposed solutions are different from each other due to the different as in robots, sensors both in type and arrangement on the robot, different behaviors, different goals robot must achieve, different environments robots employed and of course different designers.

In the design of fuzzy systems following have to be clearly defined:

- Input and output data
- Linguistic values for each input variable
- Associated membership functions
- Fuzzy rules describing the fuzzy algorithm from available knowledge

The general tendency of fuzzy controllers implemented in mobile robots is to find a directional velocity and a turn angle at the output or angular velocity. These are sufficient parameters for synchronous or omni-directional drive mechanisms. However, for the differential drive mechanism you need some extra interface (generally hardware comes with the robot) to move robot in desired direction. Since our robot uses differential drive and simulation environment requires the robot wheel velocities independently. We have implemented a controller in that; one of two outputs of our fuzzy controller will be linear velocity of right wheel where the other is linear velocity of left wheel.

5.4.1. Go-To-Target Behavior

It is the basic behavior and dominant only when no other behaviors are activated. Since this behavior does not need sensor activation that always directs the robot to target point (if any available). General fuzzy structure for the Go-To-Target behavior implemented has depicted in Figure 5.3.

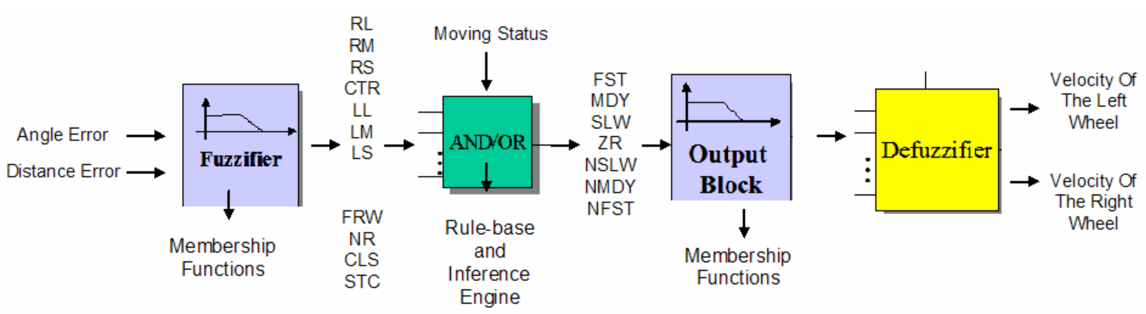
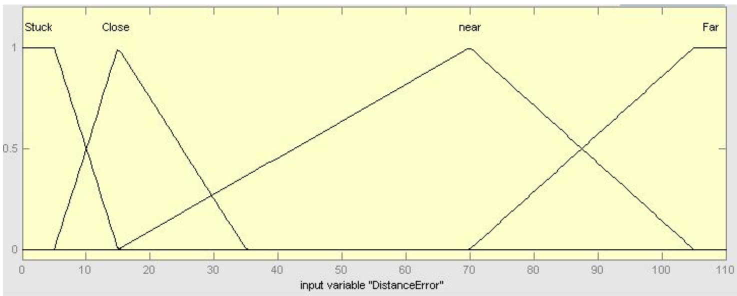
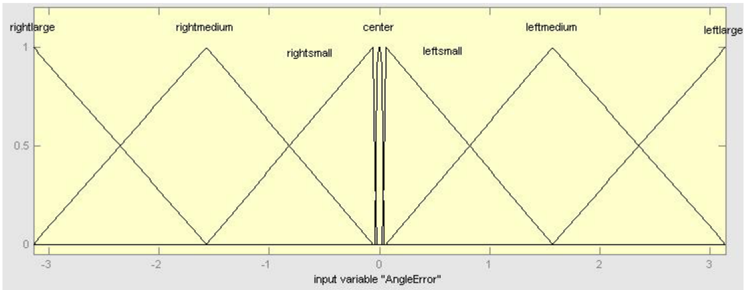


Figure 5.4. The Structure of the fuzzy controller

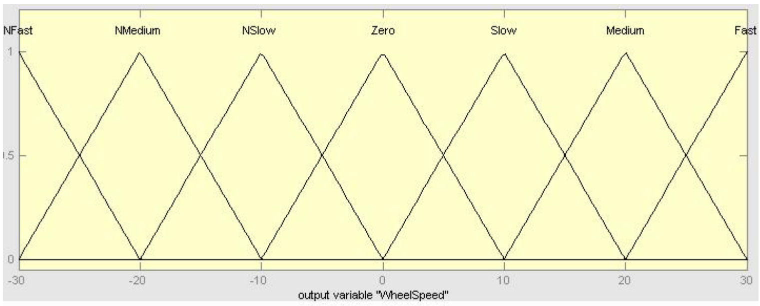
There are three input values which are moving status, distance-error and angle-error and two outputs velocity of right wheel and velocity of left wheel. The first input, moving status, is a boolean-function used only in rule-base of fuzzy controller. If the wheels not turning or turns with the same speed but opposite directions than means, robot is standing, in any other cases robot is moving. The second input is distance error. We assume that the robot always knows its current position and orientation, [$x_{current}$, $z_{current}$, $\boldsymbol{\sigma}_{current}$], according to the reference point which is the left corner of the museum. Obviously, there has to be a target point [$x_{target,}$ z_{target}] where the robot wants to reach, determined by the path planner of the robot. First the robot draws a line from the center of itself to the target point. Then the length of this line is the distance error that is one of the inputs fuzzified. The linguistic values used to identify distance are, simply: *far, near, close, stuck.* Membership functions of these values are depicted in Figure 5.4. (a). The second input the robot calculates is the angle error. To identify it, the first angle of the line drawn, $\boldsymbol{\sigma}_{line}$,to the reference point is calculated. Then the narrow angle between the $\boldsymbol{\sigma}_{line}$ and $\boldsymbol{\sigma}_{current}$ and direction of motion to make it zero as fast as possible are calculated. So if turning right is the optimum action to reach the target orientation, this means the gap on the left side of the target orientation must be closed. For the left side it is vice versa. The narrow angle is always between [0 $\boldsymbol{\pi}$]. To identify which direction to turn we have coded angle error such as that if minus signed added to narrow angle between the robot and target line then left turn is required. For the right turn it is taken as positive. Smiliarly, we have defined seven linguistic values to identify angle error. They are: *right large, right medium, right small, center, left small, left medium, left large.* The membership functions of these values are depicted in Figure 5.4. (b). The two output velocities of wheels are also identified with seven linguistic values which are the same for each: *negative fast, negative medium, negative slow, zero, positive slow, positive medium, positive fast.* The membership functions related to these variables are depicted in Figure 5.4. (c).



a)



b)



c)

Figure 5.5. Membership functions for a) Distance error b) Angle error c) Speed of wheels

The final step in designing a fuzzy controller is, constituting the rule base and inference engine. There are two different possible inputs from the mobility status, four from distance error and seven from angle error. So there are 56 different possible cases for which we have to decide output values. At first, an expert could define outputs for each case however it requires modifications after the controller is tested. The 56 different cases and the corresponding output values are shown on Table 5.1 which are constructed after several simulations done. 22 different “if then” rules are used to include all the cases in the rule-base.

Table 5.1. Knowledge base of fuzzy controller used in go-to-target behavior

States	Rule	Mobility Status	Angle Error	Distance Error	Left Wheel Speed	Right Wheel Speed
1	**1**	Moving		stuck	Zero	Zero
7	**2**	Moving	LeftLarge	Close	Slow	Zero
8	**2**	Moving	LeftMedium	Close	Slow	Zero
9	**2**	Moving	LeftSmall	Close	Slow	Zero
10	**3**	Moving	RightLarge	Near	Slow	Medium
11	**3**	Moving	RightMedium	Near	Slow	Medium
12	**3**	Moving	RightSmall	Near	Slow	Medium
13	**4**	Moving	center	Close	Slow	Slow
14	**5**	Moving	LeftLarge	Near	Medium	Slow
15	**5**	Moving	LeftMedium	Near	Medium	Slow
16	**5**	Moving	LeftSmall	Near	Medium	Slow
17	**6**	Moving	LeftMedium	Far	Fast	Medium
18	**6**	Moving	LeftSmall	Far	Fast	Medium
19	**7**	Moving	RightMedium	Far	Medium	Fast
20	**7**	Moving	RightSmall	Far	Medium	Fast
21	**8**	Moving	RightLarge	Close	Zero	Slow
22	**8**	Moving	RightMedium	Close	Zero	Slow
23	**8**	Moving	RightSmall	Close	Zero	Slow
24	**9**	Moving	LeftLarge	Far	Fast	Slow
25	**10**	Moving	RightLarge	Far	Slow	Fast
26	**11**	Moving	center	Far	Fast	Fast
27	**12**	Moving	center	Near	Medium	Medium
28	**13**	Standing	LeftLarge		Fast	Nfast
33	**14**	Standing	LeftMedium		Medium	Nmedium
37	**15**	Standing	LeftSmall		Slow	Nslow
41	**16**	Standing	RightLarge		Nfast	Fast
45	**17**	Standing	RightMedium		Nmedium	Medium
49	**18**	Standing	RightSmall		Nslow	Slow
53	**19**	Standing	center	Far	Fast	Fast
54	**20**	Standing	center	Near	Medium	Medium
55	**21**	Standing	center	Close	Slow	Slow
56	**22**	Standing	center	stuck	Zero	Zero

Mamdani type of implication is used to represent the meaning of the "if-then" rules and Center-of-Area method is applied in the defuzzification.

5.4.2. Avoid-Obstacle Behavior

This is the second behavior and suppresses the go-to-target behavior whenever it is activated. Activation of this behavior occurs whenever the robot senses an object nearby. The range sensors that perceive the distance to the objects, are located in the front such a way that we can define virtual sensors as the minimum of one sonar at the upper, one sonar at the middle, one infra-red and one tactile at the bottom which are on the same column. We have defined ten virtual sensors. Four of them are located on the north-west sides of 12 sided polygonal named fronleft4 (FL4), fronleft3 (FL3), fronleft2 (FL2), fronleft1 (FL1). Four of them are located on the north-east sides of the base named frontright4 (FR4), frontright3 (FR3), frontright2 (FR2), frontright1 (FR1). And four of them are located on the north. Two of four, front2 (F2) and front3 (F3) on the north side of the polygon, the other front1 (F1) is the same as frontleft4 and the front4 (F4) is the same as FR4. The value of the virtiual sensors are the minimal range sensed in that group. The minimal range of these 10 virtual sensors is the distance error used as an input to the avoid–obstacle module. The other eight inputs to this module are: FR4, FR3, FR2, FR1, FL4, FR3, FR2 and FR1. Each sensor of these eight, like the angle-error in the previous behavior that each make a coupling with distance error and leads to an action command in the defuzzification stage that final behavior is calculated by taking the average of them. The sensor arrangements and the fuzzy architecture depicted in Figure 5.6.

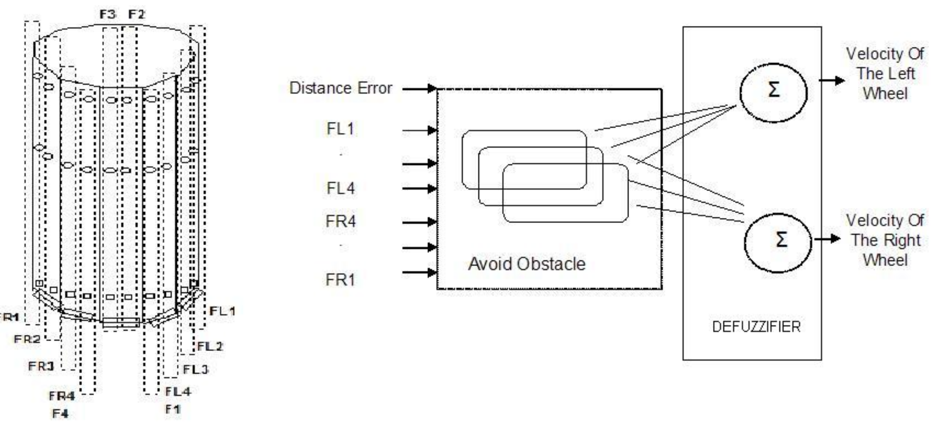
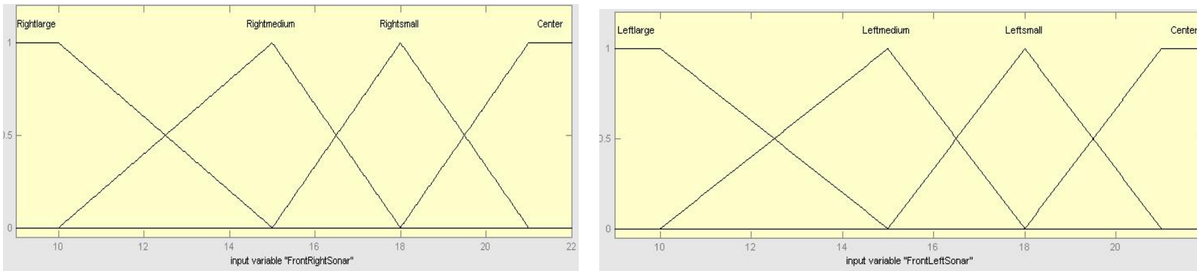


Figure 5.6. Sensor arrangement and fuzzy structure of avoid-obstacle behavior

Membership functions and the fuzzy values of distance error is the same as the one defined in the previous section. Membership functions for the sonar range sensors on the front right shown in Figure 5.7 (a) and for the range sensors on the front left are shown in 5.7 (b). Crisp values are right large, right medium, right small, center for the right sided where left large, left medium, left small and center for the left sided ones which are exactly the same as the angle error that directs the robot to the free path, since, for example, in the angle error, linguistic value rightlarge means there is a wide gap between the robot's current orientation and the target angle on the right side that robot turns left to orient itself right. In this way, the robot avoids the obstacle away.



a) **b)**

Figure 5.7. Membership functions for sonar sensors a) Front-right b) Front-left

The fuzzy values of avoid-obstacle fuzzy controller which are inputs to the inference engine are the same as fuzzy controller of go-to-target behavior whom works analogously. For this reason the knowledge base and the inference engine of both are expected to be the same. However, narrow corridors, dynamic agents and a non-holonomic drive system requires very slow motions in order not to hit any obstacle. For this reason the rule base has been changed in that almost all the rules are based on slow motion. This lets us define, the 56 possible case, with only nine "if-then" rules.

Table 5.2. Knowledge base of fuzzy controller used in avoid-obstacle behavior

States	Rule	Mobility Status	Angle Error	Distance Error	Left Wheel Speed	Right Wheel Speed
1	**1**	Moving		stuck	Zero	Zero
7	**2**	Moving	LeftLarge	Close	Slow	Zero
8	**2**	Moving	LeftMedium	Close	Slow	Zero
9	**2**	Moving	LeftSmall	Close	Slow	Zero
10	**3**	Moving	RightLarge	Near	Zero	Slow
11	**3**	Moving	RightMedium	Near	Zero	Slow
12	**3**	Moving	RightSmall	Near	Zero	Slow
13	**4**	Moving	center	Close	Slow	Slow
14	**4**	Moving	LeftLarge	Near	Slow	Slow
15	**4**	Moving	LeftMedium	Near	Slow	Slow
16	**4**	Moving	LeftSmall	Near	Slow	Slow
17	**2**	Moving	LeftMedium	Far	Slow	Zero
18	**2**	Moving	LeftSmall	Far	Slow	Zero
19	**3**	Moving	RightMedium	Far	Zero	Slow
20	**3**	Moving	RightSmall	Far	Zero	Slow
21	**3**	Moving	RightLarge	Close	Zero	Slow
22	**3**	Moving	RightMedium	Close	Zero	Slow
23	**3**	Moving	RightSmall	Close	Zero	Slow
24	**2**	Moving	LeftLarge	Far	Slow	Zero
25	**3**	Moving	RightLarge	Far	Zero	Slow
26	**8**	Moving	center	Far	Fast	Fast
27	**7**	Moving	center	Near	Medium	Medium
28	**5**	Standing	LeftLarge		Slow	Nslow
33	**5**	Standing	LeftMedium		Slow	Nslow
37	**5**	Standing	LeftSmall		Slow	Nslow
41	**6**	Standing	RightLarge		Nslow	Slow
45	**6**	Standing	RightMedium		Nslow	Slow
49	**6**	Standing	RightSmall		Nslow	Slow
53	**9**	Standing	center	Far	Fast	Fast
54	**7**	Standing	center	Near	Medium	Medium
55	**4**	Standing	center	Close	Slow	Slow
56	**6**	Standing	center	stuck	Nslow	Slow

No changes on the membership functions on the defuzzification stage are done. For this behavior also Mamdani type of implication to represent the meaning of the "if-then" rules inside the inference engine and Center-of-Area method for the defuzzification stage are applied.

5.4.3. Avoid-Human Behavior

This behavior is activated by the camera sensor in front of the robot. Actually it has to recognize people using image processing but in our case it receives radio signals emitted by the agents who are able to talk to the museum robot via sending messages. Emitted signals have a short range so that the robot could only recognize them whenever these agents are nearby. This behavior uses the controller constructed for the go-to-target behavior in such a way that: For the distance error it calculates the minimum value of the virtual front sensors (F1, F2, F3, and F4). For the angle error it gives simply zero so that the robot does not change its path. First the robot approaches the opposite agent, then speaks (sends a message via emitter) with the agent who could understand (receive the message) and finally moves away.

5.4.4. Wait-Visitors Behavior

The behavior is active whenever the camera sensor located on the rear of robot perceives that there are followers on the backside and the range sensors gives that the distance to them is increasing. Like the front ones six virtual rear sensors are defined. If the minimum value of these sensors are more than a threshold limit, this behavior slows down the robot by decreasing the speed in each run untill it stops. Also it regularly considers the front-range sensors, and if there is something nearby it directly stops.

6. 3D MODELING

Implementation of a robot requires detailed mechanic design. However, for a simulation environment we only need a solid object to control. A barely realistic, 3D Virtual robot in appearance is enough. So without considering details only taking into global constraints following museum robot will be designed.

Constraints:

- In the museum the minimum distance that the robot must pass through is 80 cm. If we assume a secure passage for the robot to the 10 cm from one edge and also a 10 cm from other edge then the maximum diameter of the robot could be 60 cm
- The base of the robot must be longer than 100 cm due to the show boxes which are depicted in Figure 6.1. To be able to recognize all the show boxes the robot has to have range sensors above this limit.
- The robot's total height must be lower than 1.70 cm. since big robots do not appeal people but frighten them. Actually for Turkish people who are relatively short and especially for children, a total height less than 1.65 is better.
- The robot's body should be big enough to place the entire equipment robot requires.

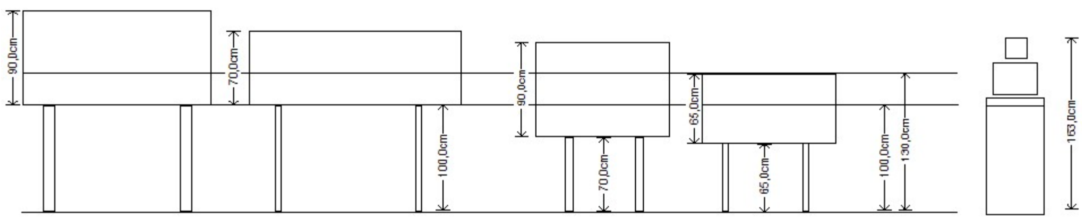


Figure 6.1. Show boxes and robot

6.1. 2D Design

Before starting to construct a 3D model of a robot we need a 2D design according to those restrictions stated in the previous section. We have designed a robot which has a polygonal base with 12 sides, a rectangular prism body and a spherical face. The whole robot fits into a cylindrical shape with a diameter of 56 cm and a height of 163 cm. The arrangement of the body is modular as depicted in Figure 6.2. Since we have 56 cm diameter, we could fit a square prism with a side length 38 cm. As seen in the figure robot is divided vertically into 12 parts, the shelves on which products will stand is designed with a height 1.5 cm that the shelves will be used as follows:

0 - Reserved for wheels and the base of the robot, its height is 10 cm, which will carry the weight of the robot. Also the arrangements of wheels will be on the sides of the square as depicted.

1 - Reserved for motors and transmission mechanisms with a 15 cm height.

2 - Reserved for boards which will drive and power the motors with a height 4.5 cm.

3 and 4 - Reserved for batteries where 12V 80A/h batteries, whose dimensions are 350 mm depth * 167 mm width * 179 mm height, could fit.

5 - Reserved for boards used to distribute power.

6- Reserved for computers and other necessary equipments with a height 13 cm.

7- Reserved for sensors such as sonar's driver boards. We can put here a laser scanner also by increasing the height of the robot.

8- Will be the solids to distinguish the face, body and base.

9- Body would be rectangular prism 200 mm depth * 400 mm width * 300 mm height for screen and computer.

10- The face, an ellipse which will fit into a 110 mm depth * 300 mm width * 150 mm height rectangular prism, will for cameras and led matrix is wide enough. The space between the cylinder and square prism is wide enough for cables and any edges due to the mechanics.

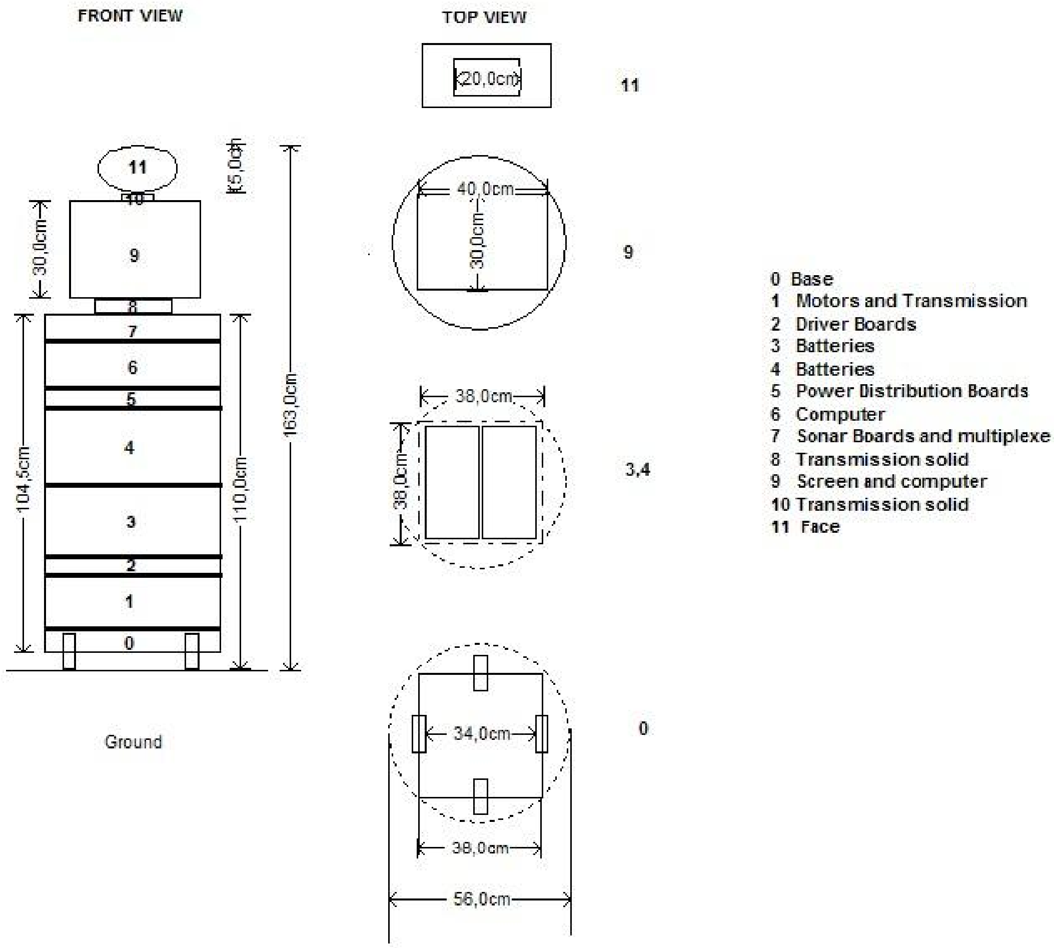


Figure 6.2. Robot's body

Another important point for the body design for our project is the arrangement of the sensors. This arrangement is depicted in Figure 6.3. We have used two sonar rings where 24 sonar sensors exist on each. The first ring is 110 cm above the ground to be able to recognize all the show-boxes in the museum. The second sonar ring is 75 cm above the ground to recognize visitors, especially children, who are shorter than 110 cm. The 24 infra-red distance sensors are located 15 cm above the ground. These are proximity sensors for the last indication about the obstacles around. These are used for the foot steps at the entrances which have a height of 20 cm. Tactile sensors are located 5.5 cm to sense a collision in the case where distance sensors fail. While deciding the locations of 24 sensors on the 12 sided polygonal prism, the only criteria we used is the symmetry. We keep the distance between two sensor's origin on one edge of the polygonal 7.5 cm. Because the different side, which sees 15° of an isosceles triangle (where the equal sides are 28 cm), is 7.4 cm that we simply take 7, 5 cm. Sensor's dimensions are taken from datasheets [45 46 47].

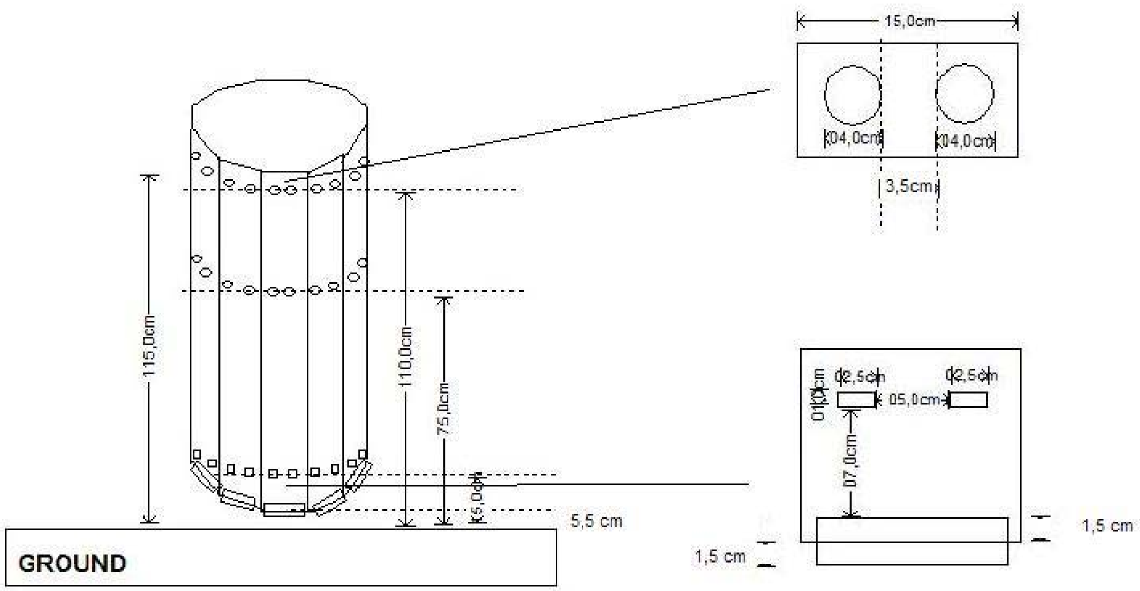


Figure 6.3. Sensor arrangement

6.2. Simulation Environment

To be able to model and a simulate we need a simulation environment we have used Webots [61, 62]. Webots is professional, commercial, three-dimensional mobile robot simulation software. Originally, it was developed as a research tool for investigating various control algorithms in mobile robotics. It contains a rapid prototyping tool allowing the user to create 3D virtual worlds, simple inert objects or active objects called mobile robots. These robots can have different locomotion schemes (wheeled robots, legged robots or flying robots). Moreover, it also allows these robots to be equipped with a number of sensor and actuator devices, like distance sensors, motor wheels, cameras, servos, touch sensors, grippers, emitters, receivers, etc. Finally the user can program each robot individually to exhibit a desired behavior.

Webots is composed of nodes and controller application programming interface (API) related to these nodes which are explained in [62] comprehensively. The User can define anything applicable by the nodes and control them by predefined classes and methods defined in controller API.

6.3. 3D Virtual World

The world where our museum robot and others run is the 3D model of the Rahmi Koç Museum. It does not include small accessories such as pictures on the walls, exhibits inside the show boxes, objects hanging on the columns. However, all the corridors, open areas, corners and all the big solid objects such as show-boxes, walls, columns, in the museum are modeled with the same size ratio in the museum. Photos from the real museum and screen shots from my virtual museum can be seen in Appendix B. As we said that Webots has a nodal structure, the *solid* node in Webots used to define objects that sensors can perceive them as physical features. To shape these objects all the nodes related to geometry such as: *Appearance, box, background, color, cylinder, sphere, point light, coordinate, etc* are also used.

6.4. 3D Virtual Robots

We have modeled the museum robot based on the design we had done. Also we added some other agents representing the visitors in the museum. Actually, a 3D model of a visitor and representing a typical visitor behavior is a challenge. We assumed that the agent could only exhibit one behavior of a visitor. The museum robot and the agents are depicted in Figure 6.4. Our 3D modeled museum robot is in the middle and the other agents stand on the right and left of it.

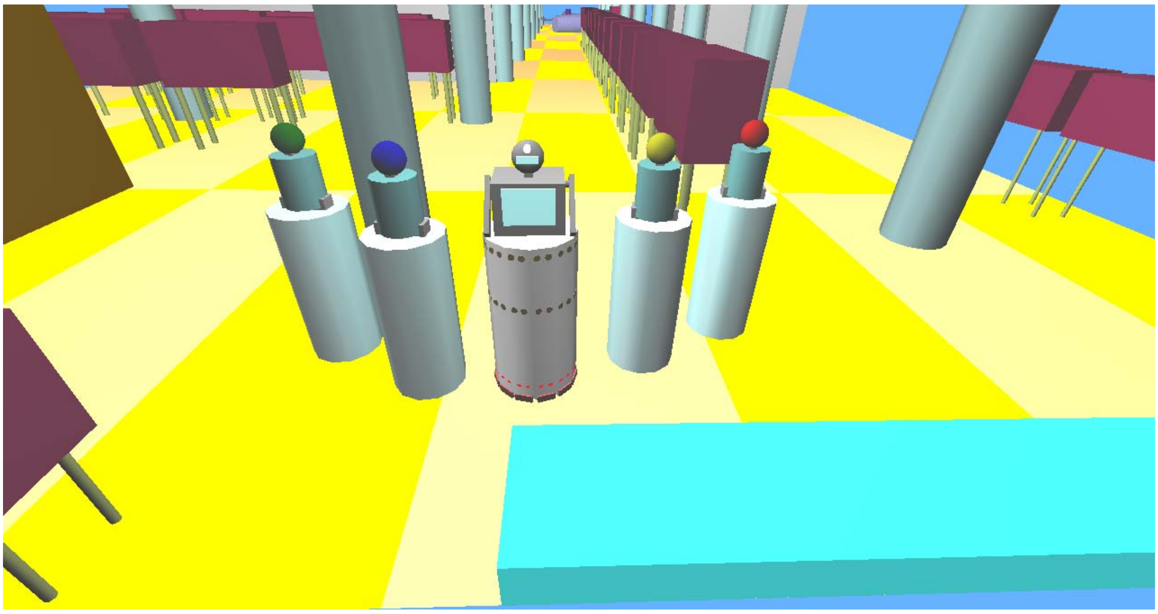


Figure 6.4. Museum robot and other agents

6.4.1. Museum Robot

To model the museum robot in Webots the *Differential Wheels* node is used which represents any robot with two-wheel differential steering. The two specific fields which are essential for the simulation are *axleLength* and *wheelRadius* that are adjusted to 34 cm and 55 cm respectively as designed. The *Distance Sensor* node is used to model sonar sensors and infra-red sensors. The important field of this node is *LookupTable* field where you define the characteristics of the sensor in response to a measured distance. Our model for the infra-red distance sensor and the sonar distance sensor can be seen in Figure6.5. Since there is no specific model for any sensor the model for infra-red in [61] adjusted and the model for sonar-sensor is created intuitively. Since the infra-red sensors in the market have 8 bit resolution with a 0 to 80 cm range, the sensor returns a value between 0 and 256. On the other hand the sonar sensor returns a value between 0 and 1024 due to the 10 bit resolution. The *TouchSensor* node is used to model tactile sensors. A tactile sensor will detect the collision with any Solid object in the world, including other *DifferentialWheels* nodes. Normally, it returns nothing (0) however if it is activated it returns some value other than 0. *Supervisor* node is used to get the robot's current position and orientation instead of a localization module. *Transmitter* and *Receiver* nodes are used for communication.

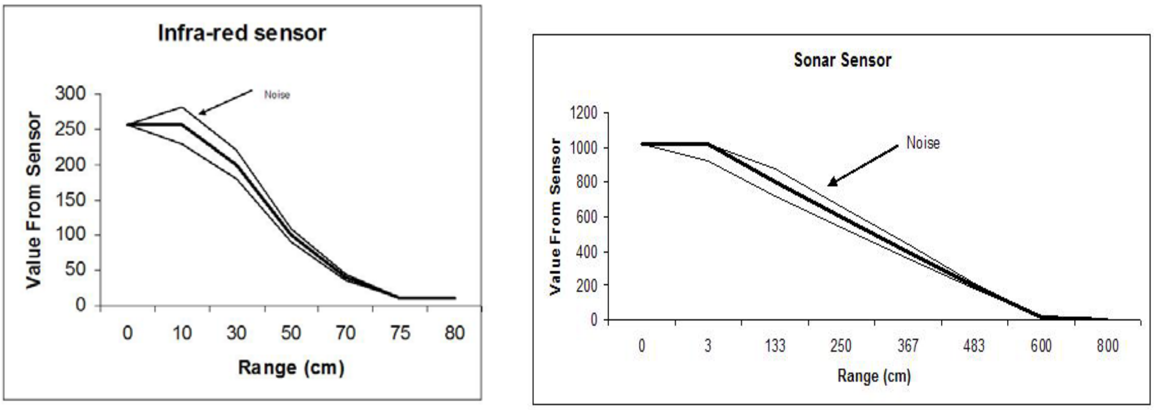


Figure 6.5. Distance sensor models

6.4.2. Other Robots

There are four different robots that are modeled to simulate a museum environment. The first agent, seen in Figure 6.4 at the left side, which has a green head, only moves to right and left. If there is something on its way it stops and waits. The second agent between the first and the museum robot with blue head has the same features with the first one, except it moves back and forward. These represent the visitors who are not aware of the museum robot and just walk around. They are simple agents implemented by *custom robot* node in Webots. The agent that stands to the right of the museum robot with a yellow head is also implemented by *custom robot.* But it also includes a *receiver* node which allows him to perceive the messages that come from the museum robot and a *transmitter* node to send messages. It stands without any movement unless a message that comes from the museum robot. Whenever a request comes it moves to vacate the path of museum robot. This agent represents people who are aware of the museum robot. The final agent that stands on the right edge uses the same nodes as the museum robot and a control algorithm with a different mission. It follows the museum robot to represent visitors who are interested in a tour with the museum robot.

7. SIMULATION AND RESULTS

We have done several experiments, as a simulation, during the development process of the control algorithm. In these experiments the modeled museum environment and the designed and modeled museum robot are used. To represent dynamic environments agents who move left-right and south-north, who wait a message from museum robot to move and who only follows the museum robot are used. To be able to demonstrate their location when they met with museum robot, we have added a feature of drawing line when the museum robot 1.5 meters away from them.

7.1. Path Tracking

We have defined 24 nodes on the 2D museum map where each node is a [x, z] point on either the intersection area of two (or more) corridors or areas where the corridors end. A path is an array of these nodes. The only restriction is that the consequent nodes in the array must be the adjacent nodes (There must be no other nodes between them). In the literature this type of representation called "Road-map representation". In Figure 7.1 path tracked by the robot is shown. The nodes on the map are the points where the robot makes a 90° turn. We assume that the robot first reaches the target node, it stops then gets the next target point and then moves there. The robot makes sharp turns at the nodes since the control program does not make a forward motion but only hard turns while the robot is not moving and there is an inevitable angle error.

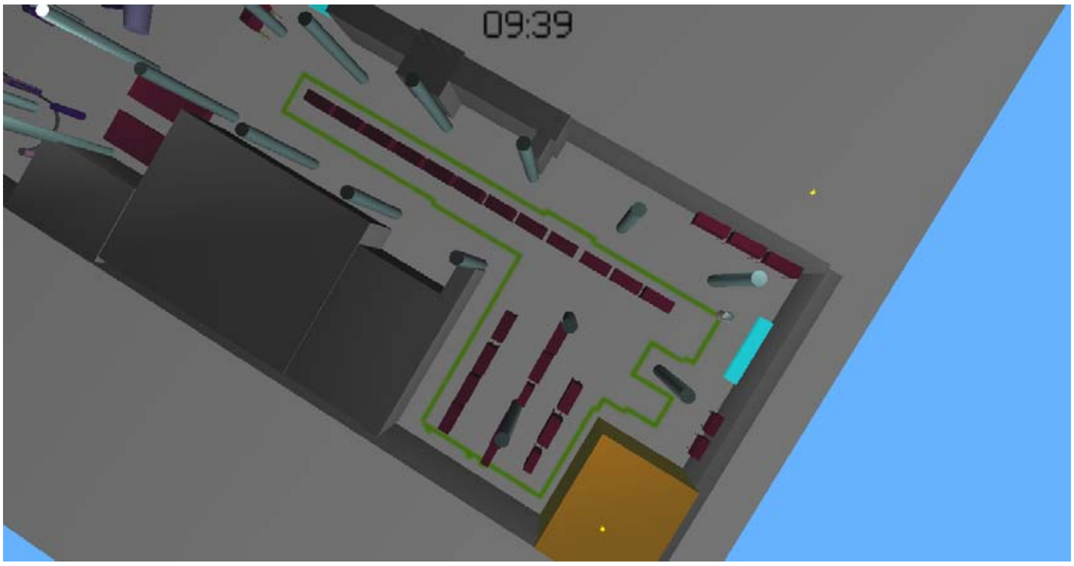


Figure 7.1. Path tracking with hard turn

However, if we change our program which controls the robot such that, we give the target node after reaching the next node but before stopping a path has resulted as depicted in Figure 7.2. The path, the robot follows changes such that we observe soft turns. This shows that our robot could follow any path linear or curved if it is planned under the necessary dynamic and kinematics constraints the robot has. The robot completes the path in a shorter time using this method which was also expected. Shorter time and curved turns also means less power consumption because motors work less and also the high currents motor consumes during the initial start are avoided. Since power is vital for an autonomous robot, the control algorithm is based on this type of turn is used. Further experiments are done and the control code is developed in this way.

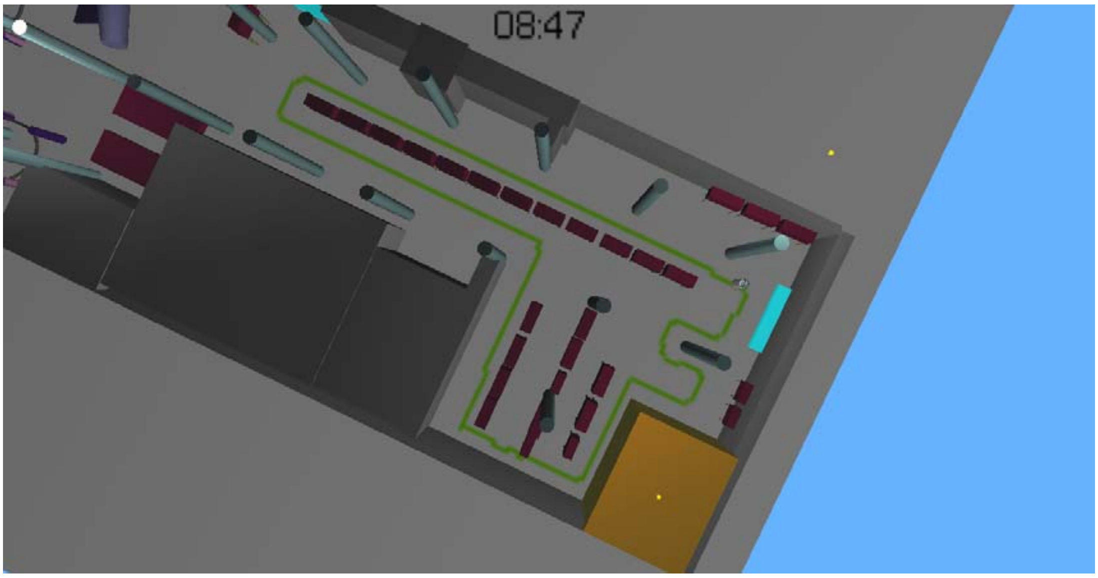


Figure 7.2. Path tracking with curved turns

We have planted some static obstacles on the way of the robot to test the congruity between the path tracking and obstacle avoidance. Two different obstacles are used: A box to represent sharp corners and a cylinder to represent curved corners. In the previous tests the corridors, corridor ends and open areas were tested. These were the challenges that the robot has to cope. The resulting path the robot follows is depicted in Figure 7.3. The robot moves in the desired places and avoids the obstacles successfully but as seen, simple path representation we have defined is not sufficient. Because after an obstacle is avoided, gaps between the desired path and the robot's path followed appeared. Robot's path tracking without having any collisions in a dynamic environment is shown in the Figure 7.4.

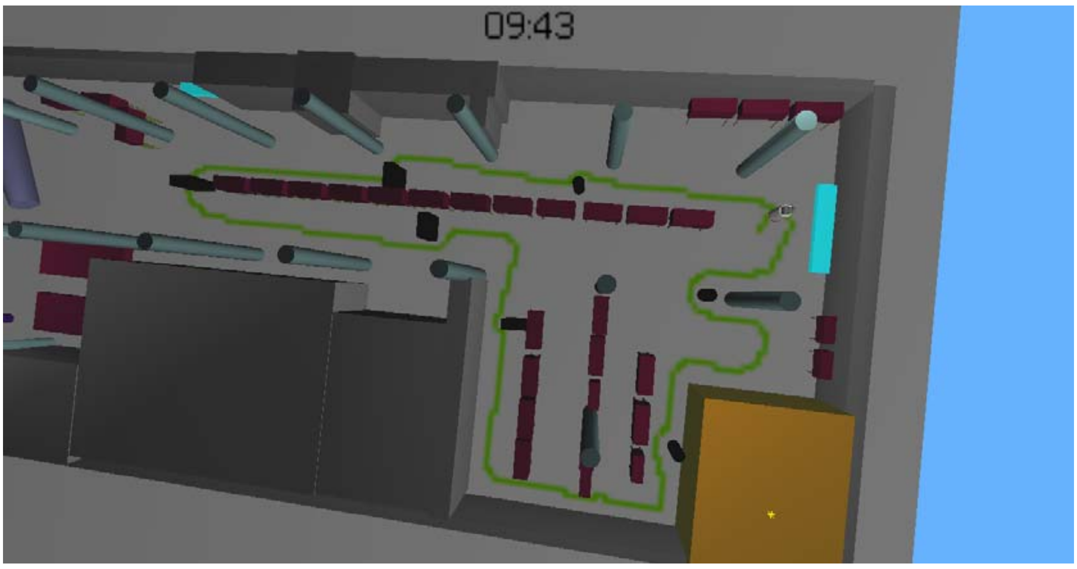


Figure 7.3. Path tracking with static obstacles on the way

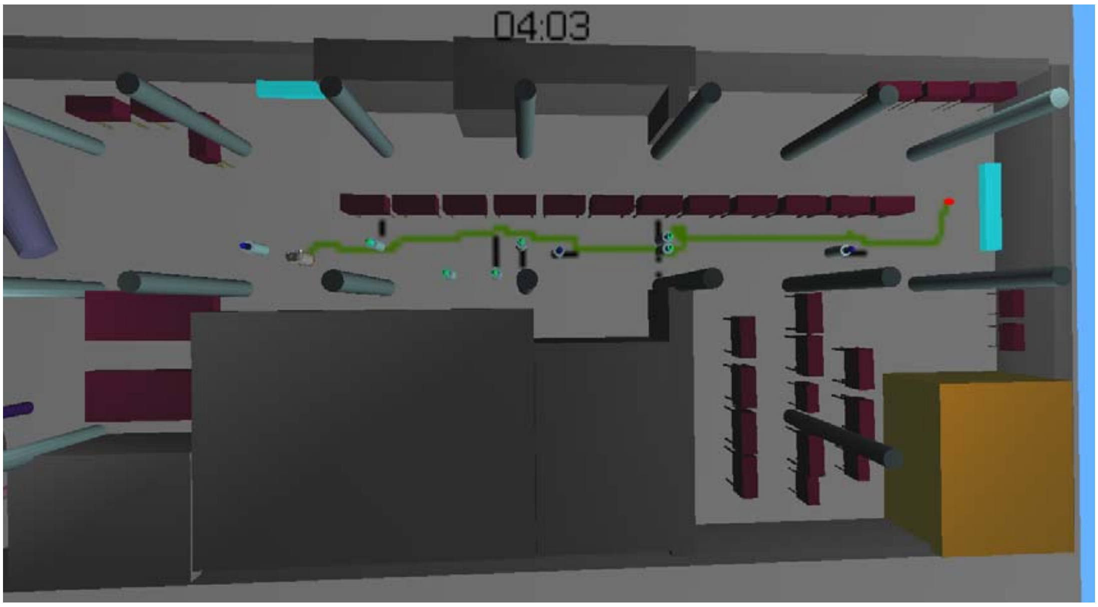


Figure 7.4. Path tracking with static obstacles on the way

7.2. Reactive Navigation

Reactive navigation is finding the way to the desired destination without a world representation or a planned path. In Figure 7.5 robot's path is shown from various different starting points to one final target point with no obstacles around and in Figure 7.6 with dynamic obstacles shown. Since it was done without a path plan robot could go anywhere in the museum. This makes us to visit the entire museum to test for deficiencies in robot's perception and software design. During these experiments we have recognized that at some places the robot fails to avoid obstacles. This was due to noisy and distorted sonar data. Distances less than 3 centimeters could not be detected. When the robot approaches this close, it fails to measure the distance and behaves like there is no obstacle. For this reason we have changed the code so that the calculated range values are decreased by five centimeters before processing. This simple change results with a very reliable and successful control algorithm which does not suffer from the local minima problem. However to make sure that the robot wanders around safely, some touch sensors canbe placed on the body to feel the show-boxes as an additional precaution in case the sonar sensors fail.

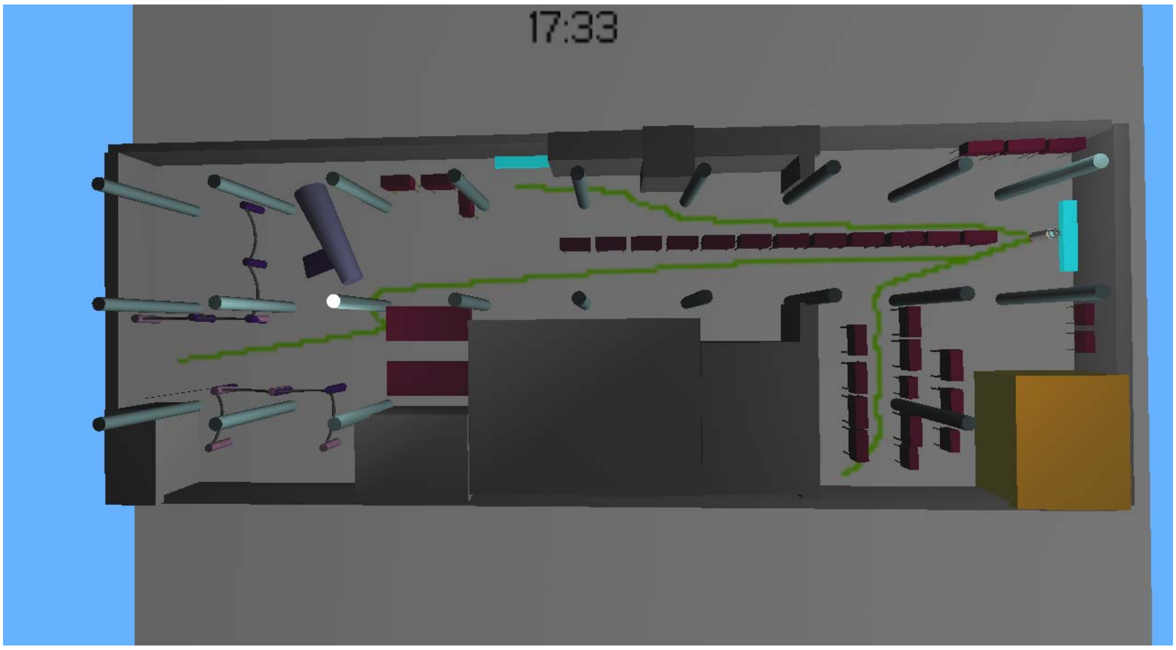


Figure 7.5. Robot's paths to reach a target point without a planned path

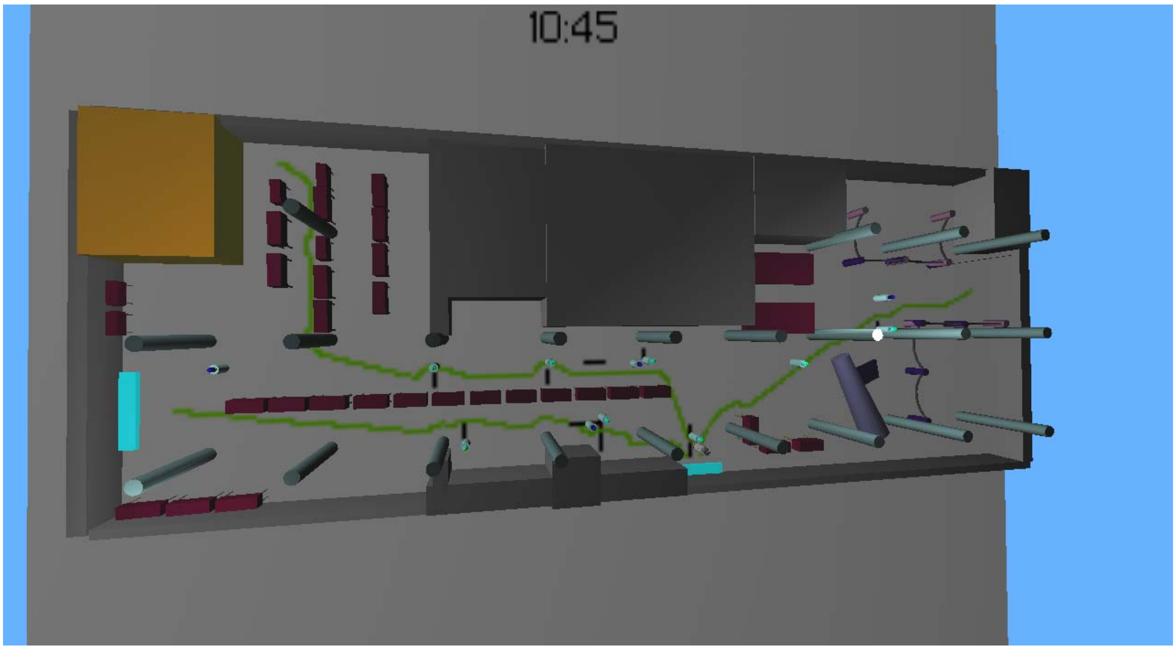


Figure 7.6. Robot's paths to reach a target point without a planned path with dynamic obstacles

We assume that the full featured museum robot can always locate itself. For this reason up to now in these experiments, the robot is assumed to know its current location. However it could not always be the case. The robot could lose the current location or can be moved by external forces, which is called as the kidnapping problem. The recovery of the current location is the job of the localization module but until the module relocates the robot wandering around without knowledge of current location is the responsibility of reactive navigation. Our control algorithm which includes avoid-obstacle behavior which can direct the robot based only on the information from sonar sensors successfully accomplishes this job. The route tracked by the robot is shown without any knowledge from localization or path planning in Figure 7.7 without dynamic obstacles and in Figure 7.8 with dynamic obstacles. This type of navigation actually is not desirable for the long time. The localization module must recover as soon as possible because robot has the probability of accessing restricted areas.

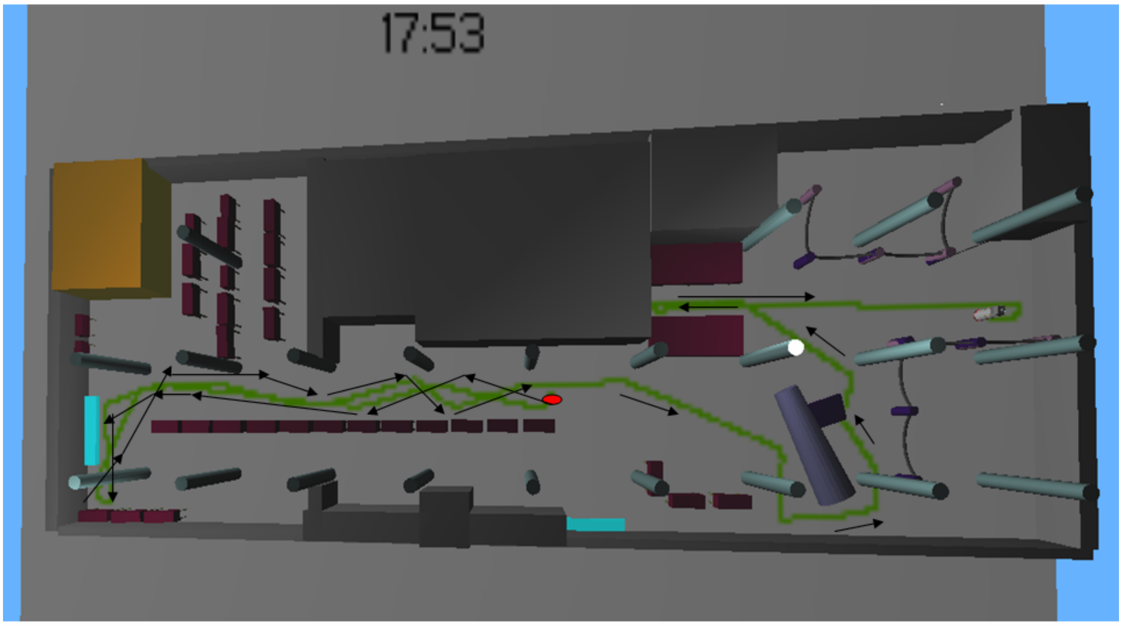


Figure 7.7. Robot wandering without localization or path planning info

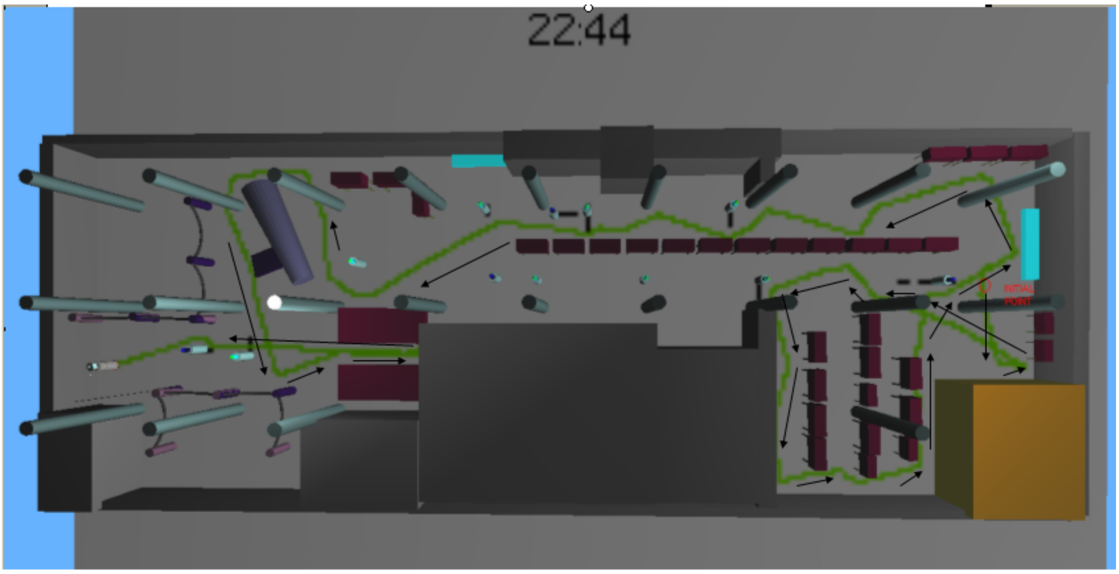


Figure 7.8. Robot wandering without localization or path planning info in dynamic environment

7.3. Goal Oriented Navigation

In [60] several different fuzzy approaches were stated for navigating vehicles autonomously with fuzzy logic. According to verification examples used in those approaches to show their fuzzy controller successful, the experiments we have done up to now seems sufficient. However, our museum robot has objectives such requesting a human to make the path free and guiding visitors keeping the distance between them at a traceable range. Actually these behaviors require high level controllers; here we simply demonstrate our reactive navigation programing since the whole system is modular in future behaviors implemented could be easily integrated with our reactive implementation. In Figure 7.9 the path tracked by the robot is shown. There are two agents, one of them is on the path of museum robot, and the other is not. As can be seen, the robot forces the agent who is on the way, to move and does not changed its path. The robot ignores the other agent who is out of its way. In the second step we have added a follower agent and the results are shown in Figure 7.10. Our museum robot again does not change its path but finishes 12 seconds late since it stoppes to wait the follower who was obstructed by the first agent. Finally we added two more dynamic agents to test all the behaviors, go-to-target, avoid obstacle, avoid-human, and wait-visitors. The path generated is shown in Figure 7.11. The coordination of behaviors was satisfactory and our museum robot accomplished all the expected actions.

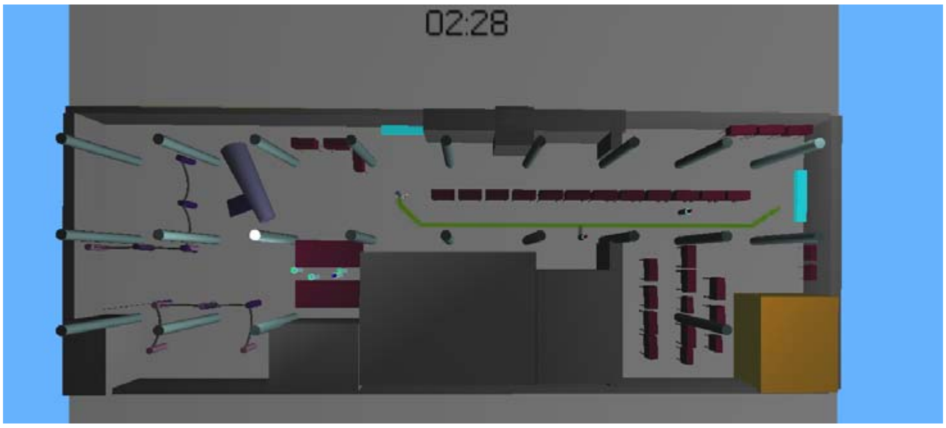


Figure 7.9. Avoid-human behavior

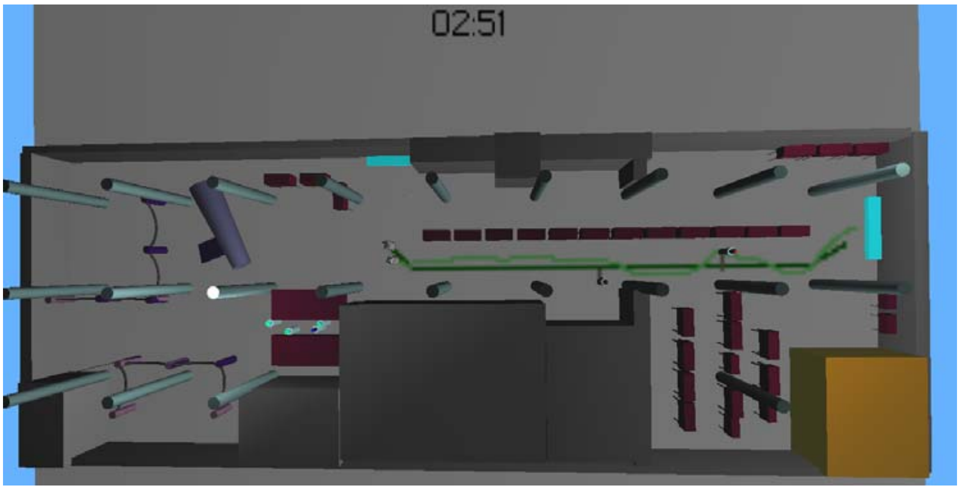


Figure 7.10. Avoid-human behavior and wait-visitors

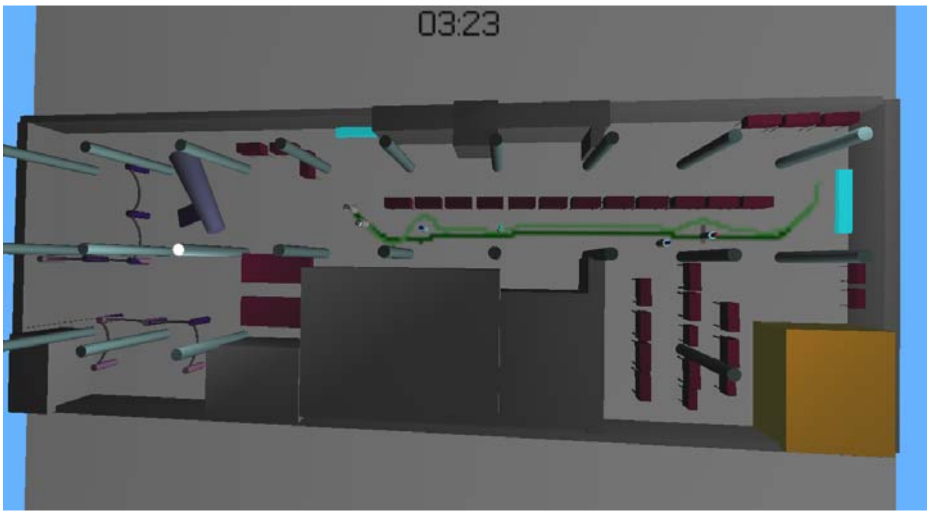


Figure 7.11. Robot's final control

8. DISCUSSIONS AND CONCLUSIONS

In this study, we have designed and modeled an autonomous mobile robot which can be implemented as a tour guide. First, the mechanical design of the robot was made, by developing a hierarchical framework between the systems. In the requirements analysis we have elicited the objectives of the robot and the environment it will operate. Decisions and constraints raised here directly affect further design in that any change could change the whole design. For example, if the surface of the Rahmi Koç museum was not solid but was carpet covered, instead of wheeled locomotion we would have to use legged locomotion so that every sensor, motor, batteries, shape, and electronics would be different. If we decided to implement two robots that work in shifts, half of the day one, and other half of the day other, then again the battery size, motor power and body of the robot would be considerably different. We made the hardware design by deciding on a reliable locomotion mechanism. Differential drive was chosen although there are other suitable ones due to easy implementation, popularity in robotic applications and proven success for our tour guidance. According to the datasheets proposed by the manufacturers, urethane on aluminum wheels were chosen due to easy rolling, high load bearings and relatively small sizes. Actuators which are DC motors and gears in our case are discussed and their characteristics studied. In motor power selection stage instead of addressing a particular motor we have made a look-up table due to the lack of information about the total weight of the robot. Sensors are comprehensively investigated due to the importance on further design, model and analysis. Batteries which are also a vital part of the design due to the long autonomy expected from our robot and high influence on robot shape and motor size are extensively investigated. A software tool is developed according to the datasheets of manufactures, which gives the required capacity and number of batteries for given power requirement. Electronics and mechanics of the robot were not investigated in detail but only the required information for further design and analysis were gathered. Before getting into software design, a final analysis on cost, power and volume was done.

After we have finished the preliminary hardware design we have continued with the software design. We first built a software architecture based on the hybrid deliberative/reactive paradigm. Systems and modules that must exist and their relations to control robot are defined. One of the subsystems, reactive control is designed based on the subsumption architecture and implemented based on the fuzzy control. The design of a fuzzy logic based navigation system for our museum mobile robot basically consists of two behaviors - an obstacle avoidance behavior and a goal seeking behavior. The advantage of using fuzzy logic for navigation is that it allows for the easy combination of various behaviors. For this reason we have added two more, simple, behaviors to simulate a goal-oriented museum tour guide robot. They are avoiding humans in front by speaking and waiting the followers who tracks the robot. Developing fuzzy control with only a human expert without any training set, and accessibility whenever required to change flow of control are the strongest parts of the fuzzy approach. Also there is no need of mathematical modeling of the environment or the robot which is very hard in our case. Hence for us it is a better solution. The inputs to the fuzzy controller are: the errors in distance and angle towards the desired direction of motion and the readings from the sensor array. Behavior activation mechanism was priority based that at each time only one behavior was active however, actions activated inside the behaviors are fused at the defuzzification stage.

To test and develop the fuzzy controller we have designed and to verify our robot design, we have modeled one big hall of Rahmi Koç museum and the designed the museum robot in 3D environment. We have also modeled four different agents to make the museum environment dynamic. The solid objects in the museum and the distances between them represented as realistically as possible. This museum model could be improved to make it look like much more to real one however, for the navigation of the robot this representation is sufficient, since all the corners, edges, crossings, passages and solid objects on the way are modeled. The robot's shape and appearances are modeled like the one developed in the design part. The sensors used for ranging are modeled according to the information stated in datasheets of the vendors.

Several simulations made to demostrate that our robot design and reactive navigation implemented is suitable for our guide robot. The cases shown are: path tracking of the robot, on a collision free path, on a path with static obstacles and on a path with dynamic agents. In all cases the robot followed its path without collision. Secondly, we have shown the path finding ability of the robot with only a target position known. This ability is shown for both dynamic and static environments. This is also required for a robot because the path planner could not be working in real time to feed the robot with a path all the time or path planning module could fail. Thirdly, the robot's navigation inside the museum without any localization and path information given showed. This is the case when robot is kidnapped. In this situation, localization module needs time and data from the environment to relocate robot. If the robots stuck or stop in somewhere the localization could not be done or done incorrect. Our robot was successful to be able to move all around the museum without any collision. Finally, the simulation of a tour with followers and the agents, who could obey the robot when it requests to empty the way, has shown. The implementation of this simulation is not realistic it does not involve any image processing or speech synthesis but it is a good example to show that our reactive fuzzy navigator could be easily integrated with the goal oriented behaviors.

9. FUTURE WORKS

We have made a design for a tour-guide robot and develop a control program for reactive navigation. Then, by modeling a 3D realistic world and robot, we have demonstrated that both our program and design is reliable and suitable for a real-word application. This is surely a nice starting point to implementation of a museum robot in real world. Our long-term objective is to improve the design proposed in this work and implement a real tour-guide robot by addressing the following issues:

- **Mechanical Design.** We have made a design for the drive mechanism, motors, gear ratio and wheels. However, this design will be extended in that transmission mechanism from motors to wheels, housings for all equipment, bearings, screws, hubs and all their location in 3D will also be designed. The hardware interface required to control the motors is also part of this design that will be achieved.

- **Power.** We have find out how much power we need and required batteries to supply. The next step will be design of: Power distribution from these batteries to components of the robot, recharging interface and a docking station suitable to that interface.

- **Electronic Design.** Hardware interfaces of sensors, interactive equipment and their connection and communication between each other and between computational resources will also be very well addressed.

- **Body Design.** Interior body design will be improved according to the requirements of mechanical, electronic and power designs (above) done. To be an attractive and esthetic appearance of the robot, exterior body design will be done by professionals.

- **Mechanical Implementation**. After all these design done, the robot will be constructed by a professional company according these designs.

- **Software Design.** The reactive navigation proposed in this thesis will be tested on a real robot in real environment to improve the algorithm during the construction process of the museum robot. Then finally, it should be tested, modified and improved on the real museum robot when the robot is ready. The other software modules discussed in this thesis (section 5.2) mission planning, learning, adaptation, mapping, localization, people recognition, face and motion tracking, media playing, resource planner and necessary data and spatial information will also be implemented. First these modules will be developed in 3D simulation environment proposed in this thesis. Later they will be tested and improved on a real robot in real environment. Then finally, on the museum robot for final modifications and improvements.

APPENDIX A: WHEEL SELECTION TABLE

Table A.1 Wheel properties

Wheel Type:	Cast Iron	Durastan	Forged - steel	Polyolefin	Pneumatic	Rubber on Iron	Urethane on Alum	Urethane On Iron	Urethane On Plastic or Urethane on Solid	Groove Forged	Groove Iron
Floor Surface	Rating: E=Excellent, G= Good, F= Fair, N= Not recommended										
Finished Wood	F	G	F	G	E	E	E	E	E	N	N
Finished Concrete	F	G	F	G	E	E	E	E	E	N	N
Ceramic Tile-Terrazo	N	F	N	F	E	E	E	E	E	N	N
Vinyl-Asphalt-Rubber-Tile	N	F	N	F	E	E	E	E	E	N	N
Oil-Grease-Solvents	G	G	G	G	F	F	G	G	G	G	G
Alkalines-Organic	F	N	F	F	F	F	G	G	G	F	F
Acids-Inorganic	F	G	F	G	F	F	F	F	G	F	F
Metal chips on Floor	G	F	G	G	N	F	F	F	F	G	G
Wheel Characteristic	Rating: E=Excellent, G= Good, F= Fair, N= Not recommended										
Ease of Rolling	E	E	E	E	F	F	G	G	G	E	E
Load Carrying Ability	E	E	E	E	F	F	E	E	E	E	E
Noise in Operation	F	F	F	F	E	E	G	G	G	F	F
Floor Protection	N	G	N	G	E	E	E	E	E	N	N
Resiliency	N	N	N	N	E	G	F	F	F	N	N
Impact Resistance	F	F	E	G	E	E	E	E	E	E	F
Moisture/Chem. Resistance	E	E	E	N	F	F	F	F	E	E	E
Abrasion Resistance	E	G	E	G	F	F	E	E	E	E	E
Temp. Range (High)	800	270	800	230	200	200	175	175	175	800	800
Temp. Range (Low)	-65	-65	-65	-40	-40	-40	-40	-40	-40	-65	-65
Durometer Rating	n/a	85D	n/a	65D	n/a	70A	95A	95A	95A	n/a	n/a
	n/a		n/a		n/a		60D	60D		n/a	n/a
	n/a		n/a		n/a		85A	85A		n/a	n

APPENDIX B: SCENES FROM MODELED MUSEUM AND REAL MUSEUM

Table B.1. Table of figures from real museum and modeled museum

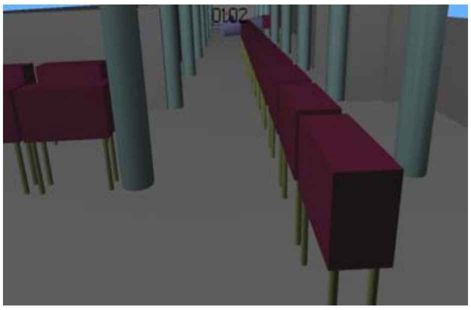
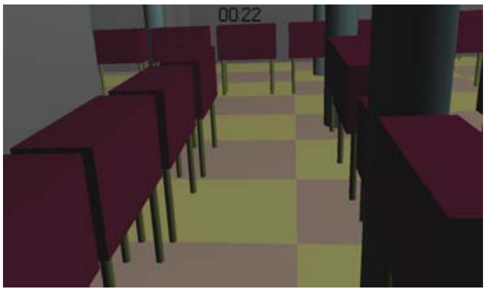
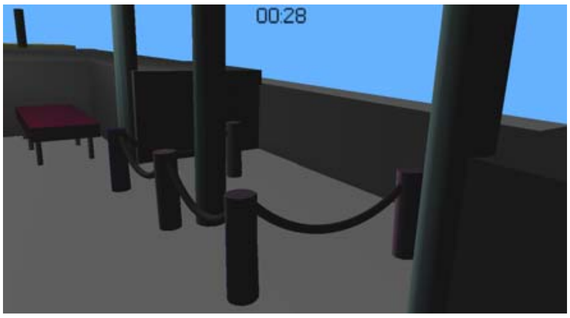
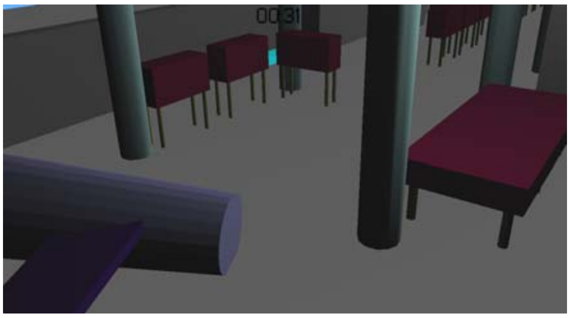
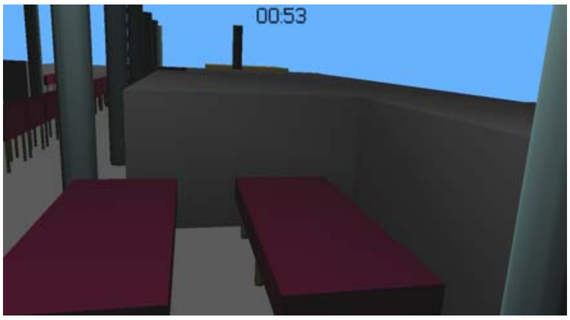
	
	
	

Table B.1. Table of figures from real museum and modeled museum continued

REFERENCES

1. Burgard, W., A.B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner and S. Thrun. "The interactive museum tour-guide robot", *AAAI-98*. Madison, Wisconsin, 1998.

2. Burgard, W and A. B. Cremers, et al. "Experiences with an Interactive Museum Tour-Guide Robot." *Artificial Intelligence*, Vol. 2, pp. 1-53, 1999.

3. Thrun, S., M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert and D. Fox, "MINERVA: A Second-Generation Museum Tour-Guide Robot", *in Proceeding of the IEEE International Conference on Robotics and Automation*, Detroit, Michigan, USA, pp. 1999-2005, 1999.

4. Thrun, S., J. Schulte and C. Rosenberg, "Spontaneous, Short-term Interaction with Mobile Robots", *Proceeding of the IEEE International Conference on Robotics and Automation*, Detroit, Michigan, USA, pp. 1999-2005, 1999.

5. Thrun, S. and M. Beetz, "Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva". *International Journal of Robotics Research* 19(11): 972-99, 2000.

6. Nourbakhsh, I., T. Willeke and C.Kunz, "The History of the Mobot Museum Robot series: An Evolutionary Study", *Proceeding of FLAIRS 2001 Conference*, 2001.

7. Nourbakhsh, I., J. Bobenage, S. Grange, R. Lutz, R. Meyer and A.Soto, "An affective mobile robot educator with a full-time job", *Artificial Intelligence*, vol. 114, pp. 95-124, 1999.

8. Graf, B. and R. D. Schraft, "A Mobile Robot Platform for Assistance and Entertaiment". *International Symposium on Robotics*, Montreal, Canada, 2000.

9. Graf, B. and O. Barth, "Entertainment Robotics: Examples, Key Technologies and Perspectives", in proceedings of *IEEE/RSJ International Conference on Intelligent Robots and Systems – Workshop Robots in Exhibitions*, 2002.

10. Arras, K., R. Philippsen, N. Tomatis, M. Battista, M. Schilt and R. Siegwart, "A Navigation Framework for Multiple Mobile Robots and its Application at the Expo.02 Exhibition", in *Proceeding of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, pp. 1992-1999, 2003.

11. Jensen, B., G.Froidevaux, X.Greppin, A.Lorotte, L.Mayor, M.Meisser, G.Ramel and R.Siegwart, "The Interactive Autonomous Mobile System RoboX", *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1221-1227, Lausanne, Switzerland, 2002.

12. Matsubara, K., K. Kato, T. Maekawa, S. Maeyama, S. Yuta and A. Harada, "Asynchronous Robot Teleoperation via the Internet: Mobile Agent Approach", *Proceedings of SSGRR*, 2000.

13. Burgard, W., P. Trahanias, D. Hahnel, M. Moor, D. Schulz, H. Baltzakis and A. Argyros "TOURBOT and WebFAIR: Web-Operated Mobile Robots for Tele-Presence in Populated exhibitions", *in proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, EPFL Lausanne, Switzerland 2002.

14. Rachid, A., "Diligent : Towards a Personal Robot", *in proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, EPFL Lausanne, Switzerland, 2002.

15. Rodriguez, D., F. Matia, R. Galan and A. Jimenez. "Blacky, an interactive mobile robot at a trade fair", *in proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

16. Murphy, R. R, *Introduction to AI Robotics,* Massachusetts Institute of Technology Press, Massachusetts, 2002.

17. Koren, Y. and J. Borenstein, "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation", *Proceedings of the IEEE Conference on Robotics and Automation*, Sacramento, California, April 7-12, pp. 1398-1404, 1991.

18. Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots". *IEEE International Conference on Robotics and Automation*, St. Louis, Missouri, March 25-28, pp. 500-505, 1985.

19. Arkin, R.C. "Motor Schema Based Navigation for a Mobile Robot: An Approach to Programming by Behavior", *IEEE International Conference on Robotics and Automation*. pp.92-112, 1987.

20. Rosenblatt, J., "DAMN: A Distributed Architecture for Mobile Navigation", *Proceedings of the 1995 AAAI Spring Symposium on Lessons Learned from Implemented Software Architectures for Physical Agents*, AAAI Press, March, 1995.

21. Brooks, R. A, "A Robust Layered Control System For A Mobile Robot", *IEEE Journal Of Robotics And Automation*, RA-2, April. pp. 14-23, 1986

22. Brooks, R. A. "Intelligence without representation". *Artificial Intelligence*, 47:139–159, 1991.

23. Arkin, R. C, *Behavior-Based Robotics*, MIT Press, CA, 1998.

24. Pirjanian, P. "Behavior coordination mechanisms - state-of-the-art". *Technical Report IRIS-99-375*, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California, 1999.

25. Maes, P., *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and back*. MIT Press, Cambridge, MA, 1990.

26. Bryson, J., A. Smaill and G. Wiggins. "The reactive accompanist: Applying Subsumption architecture to software design." *Research Paper 606, Dept. of Artificial Intelligence,* Edinburgh, 1992.

27. *Design Handbook, Online Materials & Application Notes For Robot Construction*, http://pergatory.mit.edu/2.007/Resources/index.html , 2005.

28. Akin, H. L., *Lecture nodes from Autonomous Robots course,* Bogaziçi University, Computer Science, http://www.cmpe.boun.edu.tr, 2004.

29. Siegwart, R. and I. Nourbakhsh, *Lecture nodes from Autonomous Mobile Robots course,* Carnegie Mellon, School of Computer Science, http://www.cs.cmu.edu, 2004.

30. Borenstein, J., H.R. Everett and L. Feng, *Navigating Mobile Robots: Systems and Techniques*, Wellesley, Mass.: AK Peters, 1996.

31. *User Guide: Wheel Selection Guide*, http://www.faultlesscaster.com/, 2005.

32. Jizhong, X., *Lecture notes from Robotics,* the city college of New York, department of electrical engineering, http://www-ee.ccny.cuny.edu/newpage/, 2005.

33. Jones, J. and Flynn A., *Mobile Robots Inspiration to Implementation,* A K Peters, Ltd., Massachusetts, 1993.

34. *Application Notes: Motor Calculations*, http://www.faulhaber-group.com, 2005.

35. Homepage of Robot Combat, Robotic Part Vendor, http://www.robotcombat.com/ , 2005.

36. Homepage of Dynetic, Motor and Gear Manufacturer, http://www.dynetic.com/, 2005.

37. Homepage of Maxon Motor, Motor and Gear Manufacturer, http://www.maxonmotor.com/index˙a.cfm, 2005.

38. Everett, H. R., *Sensors for Mobile Robots: Theory and Application*, A K Peters, Ltd., Wellesley, MA, ISBN 1-56881-048-2, 1995.

39. *American National Standard Specification for Dry Cells and Batteries*, American National Standards Institutes, Inc.ANSI C18.1M-1992.

40. *Application Notes & Product Data Sheet, Primary Batteries—Alkaline, Heavy Duty & General Purpose*, Rayovac Corporation, January 1996.

41. Homepage of Saft, Battery Manufacturer, www.saftbatteries.com, 2005.

42. Homepage of Creup, Battery Vendor, http://www.creup.com/, 2005.

43. Homepage of Ultralife, Battery Manufacturer, http://www.ultralifebatteries.com/, 2005.

44. *Product Guide: FM Series General Purpose Applications*, http://www.vision-batt.com/, 2005.

45. Homepage of Acroname, Sensor and Electronics Vendor, http://www.acroname.com/, 2005.

46. Homepage of Active Robots, Robotic Part Vendor, http://www.activrobots.com/, 2005.

47. Homepage of Zagros Robotic, Robotic Part Vendor, https://www.zagrosrobotics.com, 2005.

48. Homepage of Mesanet, Motor Controller Vendor, http://www.mesanet.com/, 2005.

49. Homepage of Xenarc, Monitor and Screen Vendor, http://www.xenarc.com/, 2005.

50. Homepage of Cisco Eagle, Wheel and Conveyor Vendor, http://www.cisco-eagle.com/, 2005.

51. Homepage of Active Robots, Robot and Robotic Part Vendor, http://www.activerobots.com/, 2005.

52. Siegwart, R. and Nourbakhsh I. R, *Introduction to Autonomous Mobile Robots,* Cambridge, Mass.: MIT Press, 2004.

53. Maxwell, B. A., L. A. Meeden, N. Addo, L. Brown, P. Dickson, J. Ng, S. Olshfski, E. Silk and J. Wales, "Alfred: The Robot Waiter Who Remembers You," in *proceedings of AAAI Workshop on Robotics*, July, 1999.

54. Montemerlo, M., J. Pineau, N. Roy, S. Thrun and V. Verma, "Experiences with a Mobile Robotic Elderly Guide for the Elderly", *National Conference on Artificial Intelligence*, AAAI, August, 2002.

55. Surmann, H. and L. Peters, "MORIA - A Robot with Fuzzy Controlled Behaviour", *Studies in Fuzziness and Soft. Computing*, vol. 61, pp 343-365, Springer, Berlin, 2000.

56. Saffiotti, E. H. Ruspini and K. Konolige, "Using Fuzzy Logic for Mobile Robot Control", *Practical Applications of Fuzzy Technologies*, Kluwer Academic Publisher, pages 185-206, 1999.

57. Yen, J. and N. Pfluger, "A Fuzzy Logic Based Robot Navigation System", *in proceedings of AAAI Fall Symposium,* 1992.

58. Yung, H. C. and Ye C., "An Intelligent Mobile Vehicle Navigator Based on Fuzzy Logic and Reinforcement Learning", *IEEE Top Reference. BibRef 9904 SMC-B(29),* No. 2, pp. 314., April 1999.

59. Castellano, G., G. Attolico and E. Stella, "Reactive Navigation By Fuzzy Control" , *Via Amendola,*Vol. 166, pp. 5 -20, Bari , 1998.

60. Driankov, D. and A. Saffiotti, *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*. Springer-Verlag, Berlin, Germany, 2001.

61. *Webots User Guide*, http://www.cyberbotics.com/, 2005.

62. *Webots Reference Manual*, http://www.cyberbotics.com/, 2005.