Chapter 13
Web-Based Control of Mobile Manipulation Platforms via Sensor Fusion

Ayssam Elkady and Tarek Sobh

13.1 Introduction

The World Wide Web (WWW) eliminates traditional communication problems such as long distance and time constraints. Furthermore, the WWW provides us with an unprecedented comprehensive environment for distributing information over a large network so that people living in distant locates can share their ideas and work together. In this article, we focus on web-based tele-operated mobile platformcontrol. In teleoperation tasks, remote users can control a remote mobile manipulation platform to extend human sensory and control ranges. The tele-operation tasks are potentially useful in dangerous and unpredictable environments such as at a construction site, space, underwater, service environments, and in nuclear power stations.

A mobile manipulator is a manipulator mounted on a mobile platform with no support from the ground. A mobile manipulator offers a dual advantage of mobility offered by the platform and dexterity offered by the manipulator. For instance, the mobile platform extends the workspace of the manipulator. We are developing and constructing a mobile manipulation platform called RISCbot. The prototype of the RISCbot is shown in Fig. 13.1.

Sensor fusion has been an active area of research in the field of computer vision and mobile robotics. Sensor fusion can be defined as a method for conveniently combining and integrating data derived from sensory information provided by various and disparate sensors, in order to obtain the best estimate for a dynamic system's states and produce a more reliable description of the environment than any sensor individually. Sensor fusion algorithms are useful in low-cost mobile robot applications, where acceptable performance and reliability are desired, given a limited set of inexpensive sensors such as ultrasonic and infrared sensors. Depending on the modalities of the sensors, sensor fusion can be categorized into two classes (as described in Yilmaz [1]), sensor fusion using complimentary sensors and sensor

A. Elkady (✉) and T. Sobh
Department of Computer Science and Engineering, University of Bridgeport, Bridgeport, CT 06604, USA
ayssam.elkady@gmail.com; aelkady@bridgeport.edu; http://www1bpt.bridgeport.edu/~aelkady/

S.G. Tzafestas (ed.), *Web-Based Control and Robotics Education,*
Intelligent Systems, Control and Automation: Science and Engineering 38,
DOI 10.1007/978-90-481-2505-0_13,

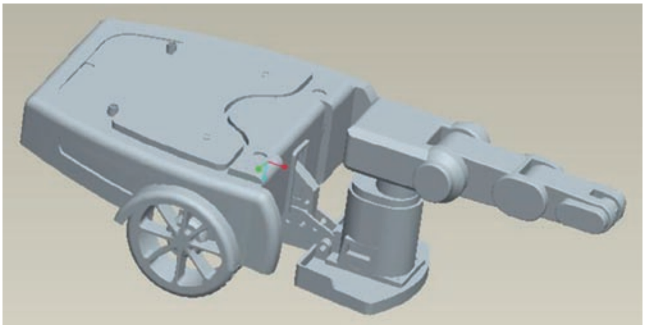

Fig. 13.1 A prototype of the RISCbot

fusion using competing sensors. Complementary sensors consist of sensors with different modalities, such as a combination of a laser sensor and a digital camera. In contrast to complementary sensors, competing sensors are composed of sensors suit which have the same modality, such as two digital cameras which provide photographic images of the same building from two different viewpoints.

Sensor fusion has some critical problems such as the synchronization of sensors. Different sensors have different resolutions and frame rates so the sensors need to be synchronized before their results can be merged by fusing the data from multiple sensors and presenting the result in a way that enables the tele-operator to perceive the current situation quickly. Sensor fusion is used to reduce the workload for the operator to enable him or her to concentrate on only the task itself.

Sensor fusion is commonly used to reduce uncertainty in localization, obstacle avoidance, and map building. Furthermore, sensor fusion may be used to improve tele-operation by creating user interfaces which efficiently facilitate understanding of remote environments and improve situational awareness.

In this article, we discuss sensor fusion for teleoperation, describe our mobile manipulation platform, and present our results.

13.2 Prior Work

Goldberg, Mascha et al. [2] designed a teleoperated robot manipulator via the WWW. Their project consisted of an industrial robot arm fitted with a CCD camera and a pneumatic system. They placed a sandbox filled with buried artifacts in the robot workspace. The WWW users can remotely move the camera to view desired locations or direct a short burst of compressed air into the sand to view the newly cleared region.

In [3], Safaric et al. designed a teleoperated system providing remote users with internet access to a laboratory robotic system and presented a real experiment which enables students to control a six DOF teleoperated robotic manipulator. Furthermore, they developed a virtual reality environment which improves the visualization of the manipulator hardware and associated workspace. In addition, they implemented a simulator to provide realistic collision detection between the virtual robot and its associated workspace.

Xavier [4] can accept commands to travel to different offices within a building, broadcasting camera images as it travels. Minerva [5] is an interactive autonomous robot that moves daily through crowds at the Smithsonian's National Museum of American History. Rhino [6] has been deployed as a tour guide robot at Deutsches Museum in Bonn, Germany.

In [7], a mobile autonomous robot called RISCBOT was designed at the RISC lab of the University of Bridgeport. RISCBOT localizes itself and successfully fulfills WWW user requests and navigates to various rooms.

13.3 Design Specifications

13.3.1 Data Acquisition

In our project, we used a data acquisition module called *Data Translation DT9814* which is a low cost USB data acquisition module that offers 24 analog input channels, 2 analog outputs channels, and one 32-bit counter timer to accommodate most applications. Furthermore, it provides a resolution of 12 bits for both the analog input and analog output subsystems, and input throughput up to 50 kHz. The Analog signal range is from −10 to 10 V. This module also provides the following features (as described in *DT9814 user's manual* [8]):

- One 32-bit counter/timer channel.
- Internal and external A/D clock sources.
- Internal and external A/D trigger sources.
- No external power supply required.
- It supports a 32-location channel-gain list. You can cycle through the channel-gain list using continuous scan mode or triggered scan mode.
- It can be connected directly to the USB ports of a computer.

13.3.2 Sensors

There are various sensor types used for measuring distances to the nearest obstacle around the robot for navigation purposes such as ultrasonic and infrared sensors. The sensors can be classified as proprioceptive/exteroceptive and passive/active [9]. Proprioceptive sensors measure values internal to the robot such as motor speed, wheel load, and battery voltage. Exteroceptive sensors acquire information from the robot environment such as distance measurements.

Passive sensors measure ambient environmental energy entering the sensor; such as temperature sensors, and microphones. Active sensors emit energy into the environment, then measure the environmental reaction.

There are two important concepts to understand when analyzing any sensor; sensitivity and range. A sensing device reacts to varying levels of some physical stimulus by outputting a characteristic voltage (or current, frequency, etc.). Sensitivity is a measure of the degree to which the output signal changes as the measured quantity changes. Let's call the sensor output r and the measured physical quantity z. The sensitivity S can be computed from Eq. (13.1).

$$\frac{\Delta r}{r} = S\,\frac{\Delta z}{z} \tag{13.1}$$

Where Δz is a small change in the measured quantity and Δr is related to a small change in the sensor response.

13.3.2.1 Sonar Sensor

A sonar sensor measures the time of flight of a sonar pulse to travel to the object in font of this sensor and the time to be received again. Given the speed of the sound, one can compute the distance to the object.

The distance d to the nearest object within the sonar cone can be computed from Eq. (13.2). Where t is the elapsed time between the emission of the sonar signal and the reception of its echo and C is the speed of the sonar signal in the medium (the speed of the sound (m/s) in dry air is given approximately by Eq. (13.3) where T_C is the Celsius temperature)

$$d = \frac{Ct}{2} \tag{13.2}$$

$$C \approx 331.4 + 0.6 \times T_C \tag{13.3}$$

There are some uncertainties associated with readings from sonar sensors. The uncertainties are due to:

- The exact position of the detected object is unknown because the computed distance d in Eq. (13.2) could be anywhere within the sonar cone.
- Specular reflections problem occurs when the sonar beam hits a smooth surface at a shallow angle and is therefore not reflected back to the robot.
- Crosstalk can occur when an array of sonar sensors is used.

We used the LV-MaxSonar®-EZ0™ ultrasonic sensors. As described in (LV-MaxSonar®-EZ0™ Data Sheet [10]), they can detect objects from 0 to 254 in. (6.45 m) and provides sonar range information from 6-in. out to 254-in. with 1-in. resolution. Objects from 0 to 6-in. range as 6-in. They are low cost sonar ranger actually consisting of two parts: an emitter, which produces a 42 kHz sound wave; and a detector, which detects 42 kHz sound waves and sends an electrical signal back to the microcontroller. Readings can occur up to every 50 ms, (20-Hz rate)

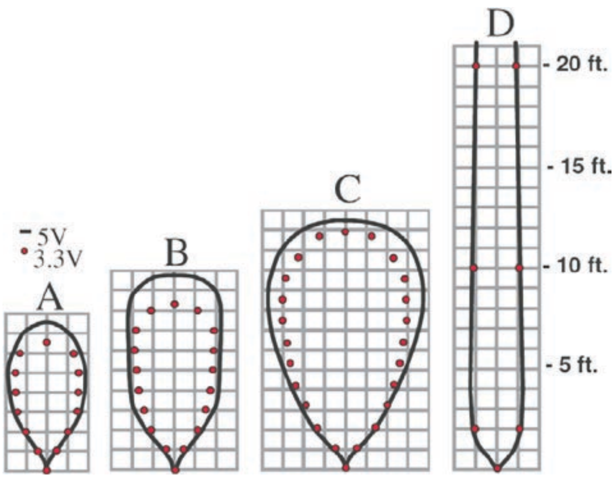

Fig. 13.2 Beam characteristics

and designed for indoor environments. The advantage of using ultrasonic sensors is that they can detect obstacles with high confidence especially when the object is well defined (i.e., located perpendicular to the sonar axis and has good ultrasonic reflectivity).

As described in (LV-MaxSonar®-EZ0™ Data Sheet [10]), the sample results for measured beam patterns are shown in Fig. 13.2 on a 12-in. grid. The detection pattern is shown for;

- [(A)] 0.25-in. diameter dowel, note the narrow beam for close small objects.
- [(B)] 1-in. diameter dowel.
- [(C)] 3.25-in. diameter rod, note the long controlled detection pattern.
- [(D)] 11-in. wide board moved left to right with the board parallel to the front sensor face and the sensor stationary. This shows the sensor's range capability.

13.3.2.2 Infrared Proximity Sensor

Infrared sensors operate by emitting an infrared light, and detecting any reflection off surfaces in front of the robot. If the reflected infrared is detected, it means that an object is detected. On the other hand, if the reflected infrared signal is absent, It does not mean that there is no object in front of the infrared sensor because certain darkly colored objects are invisible to infrared signal. Therefore, infrared sensors are not absolutely safe to use alone in obstacle avoidance applications.

We have used an infrared proximity sensor – Sharp GP20A21YK. As described in (*SparkFun Electronics website* [11]), this sensor has an analog output that varies from 3.1 V at 10 cm to 0.4 V at 80 cm as shown in Fig. 13.3. The analog sensor

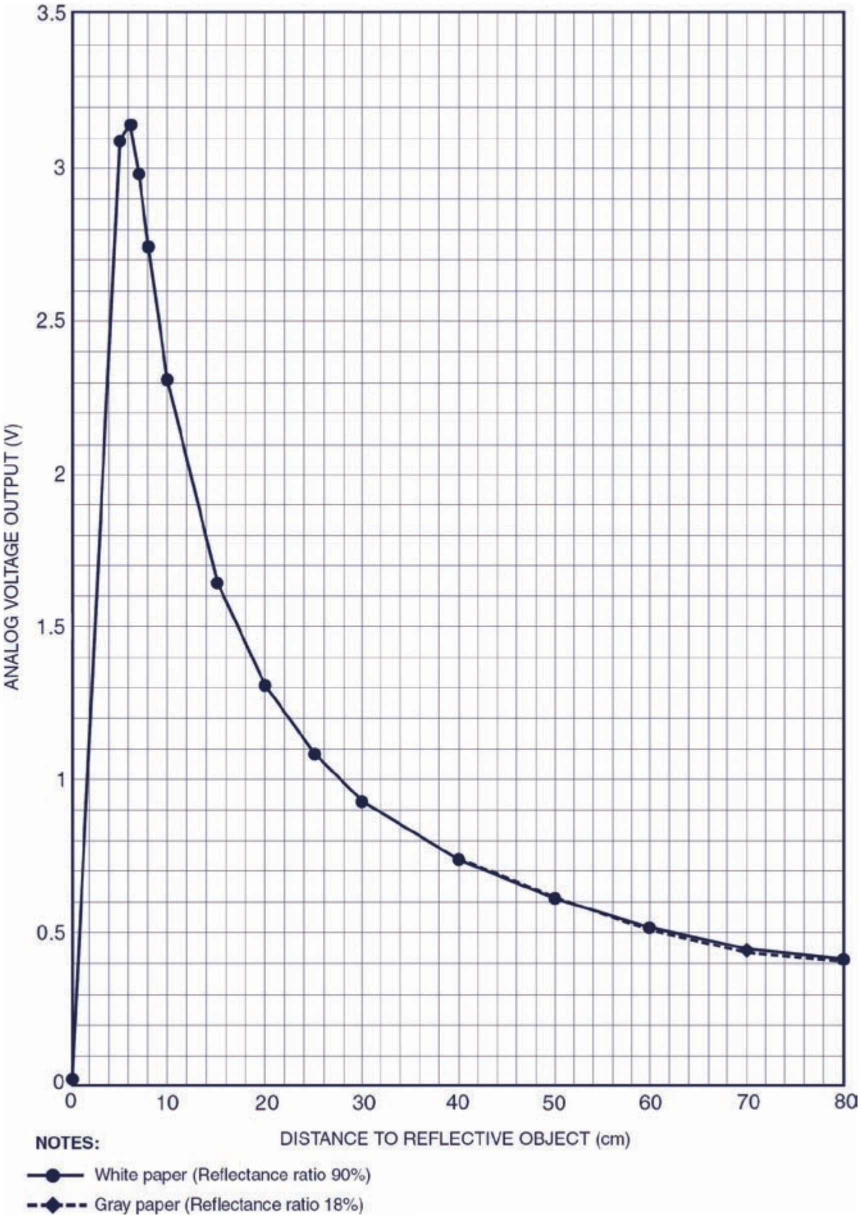

Fig. 13.3 Analog output vs. distance to reflective object

simply returns a voltage level in relation to the measured distance. As shown in Fig. 13.3, it is clear that the sensor does not return a value linear or proportional to the actual distance because the intensity of the infrared signal is inversely

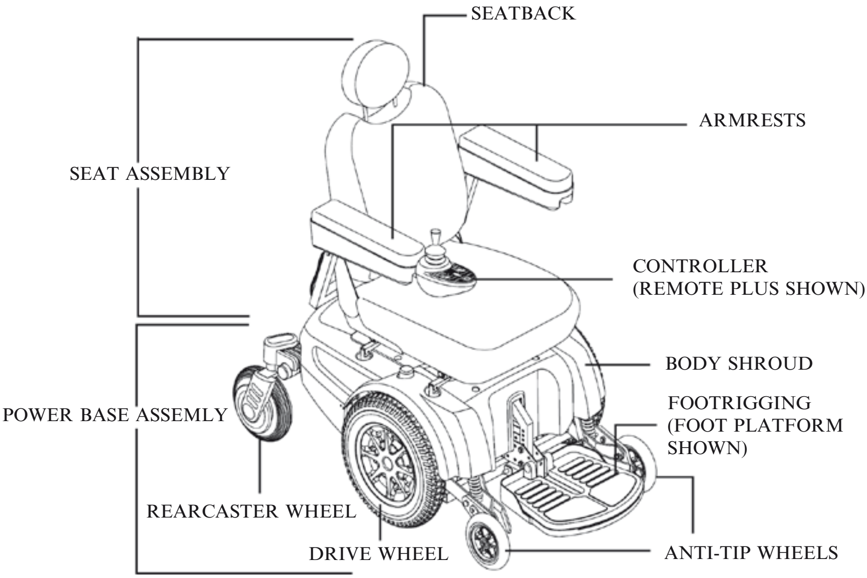

Fig. 13.4 The Jazzy 1122

probational to the square of the distance. Therefore, the infrared signal falls rapidly as the distance increases.

13.3.3 Jazzy 1122 Wheelchair

As described in (Jazzy 1122 the owner's manual [12]), the jazzy wheelchair has two main assemblies: the seat and the power base as described in Fig. 13.4. Typically, the seating assembly includes the armrests, seatback, and controller. The power base assembly includes two drive wheels, two anti-tip wheels, two rear caster wheels, and a body shroud. In our project, we remove the armrests and seatback as shown in Fig. 13.5. The specifications of the Jazzy 1122 wheelchair are described in Table 13.1. The jazzy 1122 wheelchair also provides the following features: (as described in [12])

1. *Active-Trac Suspension*: The wheelchair is equipped with Active-Trac Suspension (ATS) to be able to traverse different types of terrain and obstacles while maintaining smooth operation. With ATS, the front anti-tip wheels work in conjunction with the motor suspension to maneuver over obstacles. As the front anti-tip wheels come in contact with an obstacle, the front anti-tip wheel assembly is drawn upward. At the same time, the motors are forced downward. This allows the motors to push the wheelchair over an obstacle.
2. *Rear Suspension*: The wheelchair is equipped with a rear suspension system to work in conjunction with the ATS and is designed to maintain a smooth ride when driving over rough terrain and up and down curbs.

Fig. 13.5 The RISCbot base

Table 13.1 Specifications of the Jazzy 1122 wheelchair

Suspension	ATS and rear suspension
Drive wheels	14 in., pneumatic, center-mounted
Caster wheels	8 in., solid, rear-articulating
Anti-tip wheels	6 in., solid, front-mounted
Maximum speed	Up to 6 mph
Brakes	"Intelligent braking," electronic regenerative, disc park brake
Drive train	Two motor, mid-wheel
Batteries	Two 12-V, Group 24 batteries
Component weights	Base: 129 lb.
	Seat: 40 lb. (standard seat).
	Batteries: 53.5 lb.

13.4 Applications

13.4.1 Manipulability

Studying the performance characteristics of the robot such as dexterity, manipulability, and accuracy is very important to the design and analysis of a robot manipulator. The manipulability is the ability to move in arbitrary directions while the accuracy is a measure of how close the manipulator can return to a previously taught point. The manipulability index is considered as a quantitative and performance measure of the ability for realizing some tasks. This measure should be taken into consideration in the design phase of a serial robot and also in the design of control algorithms.

In (Mohammed et al. [13]), we presented a new method for measuring the manipulability index, then justify this concept by visualizing the bands of this index resulting from our experiments implemented on different manipulators such as the Puma 560 manipulator, a six DOF manipulator and the Mitsubishi Movemaster manipulator.

In fixed-base serial manipulators, manipulability depends on link lengths, joint types, joint motion limits and the structure of the manipulator. In mobile manipulators, the manipulability depends on the kinematics, geometric design, the payload, mass and mass distribution of the mobile platform. Thus, the manipulability measure in mobile manipulators is very complicated due to the coupling between the kinematic parameters and the dynamics effect. Furthermore, we used the proposed method for measuring the manipulability index in serial manipulators to generalize the standard definition of the manipulability index in the case of mobile manipulators

13.4.2 Navigation and Obstacle Avoidance

A prerequisite task for the autonomous mobile robot is the ability to detect and avoid obstacles given real-time sensor readings. Obstacle avoidance is a crucial issue in robot's navigation. Given partial knowledge about its environment and a goal position or a series of positions, navigation encompasses the ability of the robot to act based on its knowledge and sensor values so as to reach its goal positions as efficiently and as reliably as possible.

The obstacle may be defined as any object that appears along the mobile robot's. The techniques used in the detection of obstacles may vary according to the nature of the obstacle. The resulting robot motion is a function of both the robot's sensor readings and its goal position. The obstacle avoidance applications focus on changing the robot's trajectory as informed by sensors during robot motion. The obstacle avoidance algorithms that are commonly used can be summarized as the following: (as described in [9]).

- **The bug algorithm:** The basic idea is to follow the easiest common sense approach of moving directly towards the goal, unless an obstacle is found. If an obstacle is found, the obstacle is contoured until motion to goal is again possible. In [9], two approaches are described; Bug1 and Bug2. In Bug1 Algorithm, the robot fully circles the object first, and then departs from the point with the shortest distance toward the goal. This approach is very inefficient but it guarantees that the robot will reach any reachable goal. In Bug2, the robot will fellow the object's contour but it will depart immediately when it is able to move directly toward the goal.
- **Tangent Bug:** As described in [14], tangent bug algorithm is a variation of the bug algorithm. The robot can move more efficiently toward the goal also go along shortcuts when contouring obstacles and switch back to goal seeking earlier. In many simple environments, tangent bug approaches globally optimal paths.
- **Artificial Potential Fields**: The artificial potential fields (APF) is proposed by Khatib in [15]. The robot is considered as a moving particle in a potential field generated by the goal and by the obstacles that are presented in the environment. In APF method, the robot immersed in the potential filed is subject to the action of a force that drives it to the goal. This approach uses repulsive potential fields around the obstacles (and forbidden regions) to force the robot away

and an attractive potential field around goal to attract the robot. A potential field can be viewed as an energy field and so its gradient, at each point, is a force. Consequently, the robot experiences a generalized force equal to the negative of the total potential gradient. This force drives the robot towards its goal while keeping it away from the obstacles (it is the action of a repulsive force that is the gradient of the repulsive potential generated by the obstacles). However, there is a major problem with the APF approach because the local minima can trap the robot before reaching its goal. One of the powerful techniques for avoidance of local minima is the simulated annealing approach which has been applied to local and global path planning as described in [16].

- **Vector Field Histogram:** Borenstein and Koren developed the vector field histogram (VFH) [17]. Borenstein and Ulrich extended the VFH algorithm to yield VFH* [18] and VFH$^+$ [19]. As described in [9], the instantaneous behavior of the mobile robot in the Bug algorithms is a function of only its most recent sensor readings which may lead to undesirable problems in cases where the robot's instantaneous sensor readings do not provide enough information for robust obstacle avoidance. The VFH algorithm is computationally efficient, very robust and insensitive to misreading. The VFH algorithm allows continuous and fast motion of the mobile robot without stopping for obstacles. The VFH algorithm [17] permits the detection of unknown obstacles and avoids collisions while simultaneously steering the mobile robot toward the target. This algorithm uses a two-dimensional cartesian histogram grid to represent a local map of the environment around the robot which is updated continuously with the sampled data from range sensors. In contrast, the VFH algorithm generates a polar histogram to represent the relation between the angle at which the obstacle was found and the probability that there really is an obstacle in that direction based on the occupancy grid's cell values. From this histogram a steering direction is calculated. The polar histogram is the most significant distinction between the virtual force field (VFF) and the VFH method as it allows a spatial interpretation (called polar obstacle density) of the robot's instantaneous environment. In the VFH$^+$ algorithm [19], the basic robot kinematics limitations were used to compute the robot possible trajectories using arcs or straight lines. The VFH* algorithm [18] proposes look-ahead verification. The method investigates each possible direction provided by the VFH$^+$ approach, checking their consequences concerning the robot future positions. The experimental results of [18] shows that this look-ahead verification can successfully deal with problematic situations that the original VFH and VFH$^+$ can not handle and the resulting trajectory is fast and smooth.

13.4.3 Path Planning and Map Building

Given a map and a goal location, path planning involves identifying a trajectory that will bring the robot from the initial location to reach the goal location. During the execution, the robot must react to unforeseen events such as the obsta-

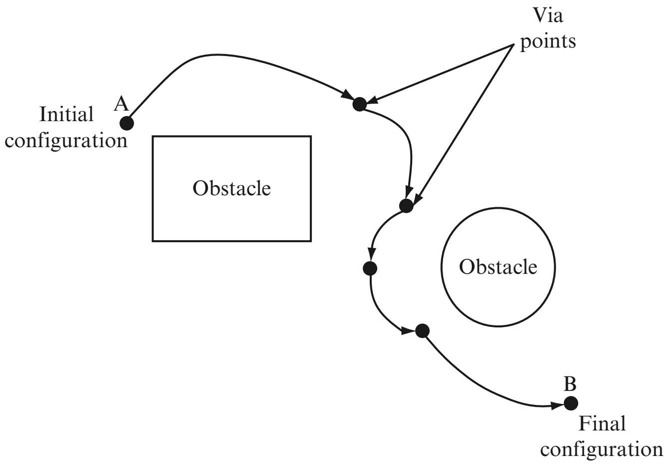

Fig. 13.6 Via points to plan motion around obstacles

cles in such a way to still reach the goal. For some purposes, such as obstacle avoidance, constrained workspace, and time-critical applications, the path of the end-effector can be further constrained by the addition of via points intermediate to the initial and final configurations as illustrated in Fig. 13.6. Additional constraints on the velocity or acceleration between via points can be handled in the trajectory planning.

Depending on the environment surrounding the robot, path planning can be classified as follows: (as described in [20])

- Path planning for static obstacles in a completely known environment
- Path planning for static obstacles in an unknown or partially known environment
- Path planning for dynamic obstacles in a completely known environment
- Path planning for dynamic obstacles in an unknown or partially known environment

The implementation of the path-planning system requires that the continuous environmental model is transformed into a discrete map suitable for the chosen path-planning algorithm. The three general strategies: (as described in [9])

- **Road map:** This approach identifies a set of routes within the free space in a network of 1D curves or lines. In this approach, the path planning is used to connect the start position with the target position of the mobile platform by looking for a series of routes from the initial position to the goal position.
- **Cell decomposition**: This approach distinguishes between the free areas and the areas that are occupied by objects [9].
- **Potential field:** As described in the previous section, this approach considers the robot as a moving particle in a potential field generated by the goal and by the obstacles that are presented in the environment.

In the navigation problem, the requirement is to know the positions of the mobile robot and a map of the environment (or an estimated map). The related problem is when both the position of the mobile robot and the map are not known. In this scenario, The robot starts in an unknown location in an unknown environment and proceeds to gradually build the map of the existing environment. In this case, the position of the robot and the map estimation are highly correlated. This problem is known as *Simultaneous Localization and Map Building* (SLAM) [21, 22]. SLAM is the process of concurrently building a feature based map of the environment and using this map to get an estimation of the location of the mobile platform. In [23], the recent Radio Frequency Identification (RFID) was used to improve the localization of mobile robots. This research studied the problem of localizing RFID tags with a mobile robot that is equipped with a pair of RFID antennas. Furthermore, a probabilistic measurement model for RFID readers was presented in order to accurately localize RFID tags in the environment.

13.5 Implementation and Results

We are developing and constructing the mobile manipulation platform called RISCbot (the prototype of the RISCbot is shown in Fig. 13.1). The RISCbot mobile manipulator has been designed to support our research in algorithms and control for autonomous mobile manipulator. The objective is to build a hardware platform with redundant kinematic degrees of freedom, a comprehensive sensor suite, and significant end-effector capabilities for manipulation. The RISCbot platform differs from any related robotic platforms because its mobile platform is a wheelchair base. Thus, the RISCbot has the advantages of the wheelchair such as high payload, high speed motor package (the top speed of the wheelchair is 6 mph), Active-Trac and rear caster suspension for outstanding outdoor performance, and adjustable front anti-tips to meet terrain challenges.

In order to use the wheelchair as a mobile platform, a reverse engineering process has been used to understand the communication between the joystick of the wheelchair and the motor controller. This process was done by intercepting the continuous stream of voltages generated by the joystick after opening the joystick module and reading the signals within joystick wires that are sent the signals to the wheelchair controller.

We used different types of sensors so that the RISCbot can perceive its environment with better accuracy. Our robot hosts an array of 13 LV-MaxSonar®-EZ0™ ultrasonic sensors. The sensors are suitable for obstacle avoidance applications but their wide beams are unable to distinguish features within the beam angle, making sonars a poor choice of sensor for fine feature extraction within indoor environments. This resolution problem is magnified for objects further away from the robot (i.e., objects appearing at the wide end of the beam). Lastly, our robot is also equipped with an array of 11 Sharp GP20A21YK infrared proximity sensors above the sonar ring. The sonar and infrared sensors were mounted together so that their beams are

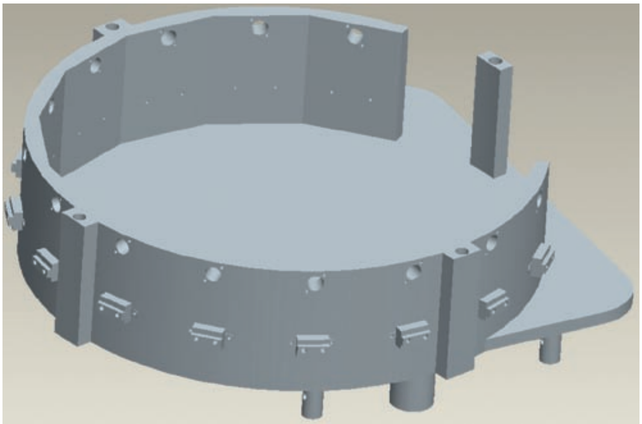

Fig. 13.7 A closeup view of the sonar and infrared

oriented in the same direction. The configuration of sonar and infrared sensors is shown in Fig. 13.7. These sensors allow the RISCbot to obtain a set of observations to provide these observations to the controller and higher decision making mechanisms. The controller acts upon this set of observations to cause the robot to turn in the correct direction. The Integration of these modules together constitutes an intelligent mobile robot.

A main drawback of the infrared sensors is that they can only accurately measure obstacle distances within a range of 0.1-0.8 m. Another drawback of these sensors is that they are susceptible to inaccuracies due to outdoor light interference as well as an obstacle's color or reflectivity characteristics which can be seriously affected by windows and metallic surfaces.

Note that since our sonar and infrared sensors are in fixed positions, our experiments concentrated on performing data fusion on data obtained from a particular fixed height in the environment.

In this project, sonar and infrared sensors are used together in a complementary fashion, where the advantages of one compensate for the disadvantages of the other.

As shown in Fig. 13.8, the RISCbot software which is written in Visual *C#* and runs on a laptop reads the values of all sensors at a rate of 10 Hz gathered in the data acquisition The RISCbot software maps the sensory inputs to a series of actions which is used to achieve the required task. Based on the used algorithm, the RISCbot software responses to the sensor data by generating stream of voltages corresponding to the joystick signals to the wheelchair controller. These voltages control the direction and the speed of the wheelchair to cause the RISCbot to turn in the desired direction.

Figure 13.9 shows an experimental result for the RISCbot navigation in a real hallway environment. In this figure, the rectangles in the map are the positions of the RISCbot and the black regions are the obstacles exiting in the environment.

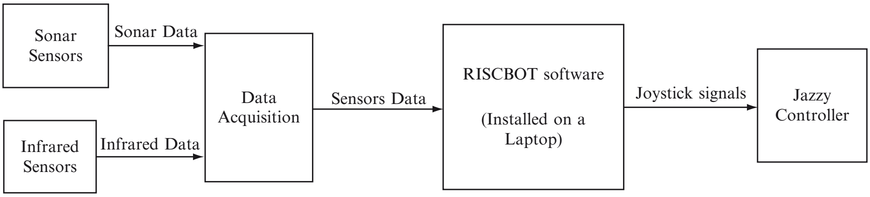

Fig. 13.8 The components of the RISCbot system

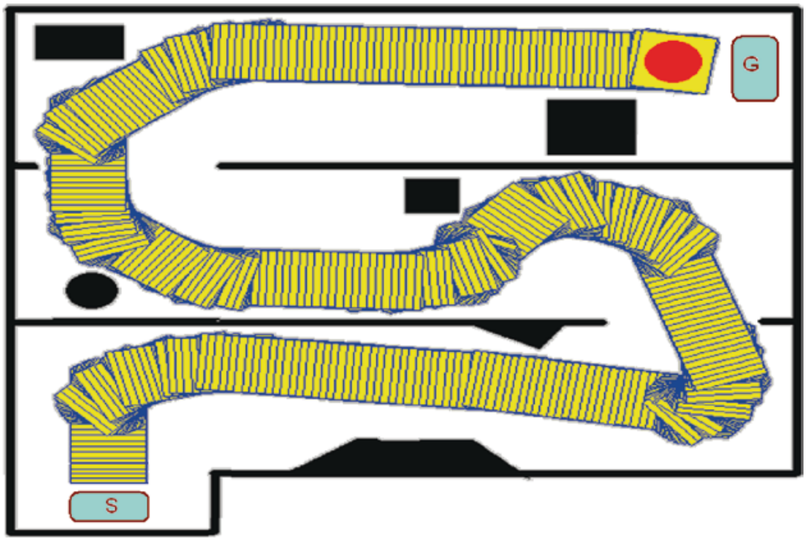

Fig. 13.9 The trajectory of the robot in the hallway environment

The experimental result indicates that the RISCbot can detect any unknown obstacle and avoid collisions while simultaneously steering from the initial position *S* toward the target position *G*.

13.6 Conclusions and Future Work

In this chapter, the mobile manipulation platform has been presented. The RISCbot platform differs from any other robotic platform because its mobile platform is a wheelchair base. Thus, the RISCbot has the advantages of the wheelchair.

Furthermore, the RISCbot consists of a comprehensive sensor suite, and significant end-effector capabilities for manipulation. In addition, we have used infrared and sonar sensors to monitor if any type of obstruction is in the path of the robot.

The results of an experiment for the RISCbot navigation and path planning modules in a real hallway environment has been illustrated. This research aspires to find online real-time collision-free trajectories for mobile manipulation platforms in an unknown static or dynamic environment containing some obstacles, between a start and a goal configurations.

Path planning for mobile robots is one of the key issues in robotics research that helps a mobile robot find a collision-free path from the beginning to the target position in the presence of obstacles. Furthermore, it deals with the uncertainties in sensor data.

The objective for this project is to implement a Web-Based tele-operated mobile manipulator via Sensor Fusion. There are great benefits in using a tele-operated mobile manipulator in dangerous, inaccessible and toxic environments. In teleoperation, a human operator controls the RISCbot from a distance. The teleoperator has some type of display and control mechanisms, and the RISCbot has sensors, an end-effector, and mobility. The teleoperator cannot look at what the remote is doing directly, in most cases, because the robot is physically remote. Therefore, in our project we have three different modules. The first module contains the sensors which gather all the information about the remote environment. The second is the display technology to provide the operator with the chance to see the sensor data. The last module is the communication link between the operator and the RISCbot.

In our anticipated future work, there will be an ongoing effort for the development of multiple mobile manipulation systems and platforms which interact with each other to perform more complex tasks exhibiting intelligent behaviors utilizing the proposed manipulability measure.

References

1. Yilmaz A (2007) Sensor fusion in computer vision. Urban Remote Sensing Joint Event.
2. Goldberg K, Mascha M, Gentner S, Rothenberg N, Sutter C, Wiegley J (1995) Desktop teleoperation via the World Wide Web. IEEE International Conference on Robotics and Automation. Volume 1, Issue, 21-27 May 1995, pp. 654–659.
3. Safaric R, Jezernik K, Calkin DW, Parkin RM (1999) Telerobot control via Internet. Proceedings of the IEEE International Symposium on Industrial Electronics, Volume 1, pp. 298–303.
4. Simmons R, Fernandez J, Goodwin R, Koenig S, O'Sullivan J (1999) Xavier: An autonomous mobile robot on the web. Robotics and Automation Magazine.
5. Thrun S, Bennewitz M, Burgard W, Cremers A, Dellaert F, Fox D, Hahnel D, Rosenberg CR, Roy N, Schulte J, Schulz D (1999) MINERVA: A second-generation museum tour-guide robot. In Proceedings: IEEE International Conference on Robotics and Automation (ICRA '99), Detroit, MI, USA.
6. Burgard W, Cremers AB, Fox D, Hahnel D, Lakemeyer G, Schulz D, Steiner W, Thrun S (1998). The interactive museum tour-guide robot. American Association for Artificial Intelligence (AAAI-98), Madison, WI, USA.
7. Sanyal R, Patel S, Sobh T (2006) RISCBOT: A www enabled mobile surveillance and identification robot. Journal of Intelligent and Robotic Systems, vol. 45, no. 1, January 2006, pp. 15–30(16).
8. DT9814 User's Manual, www.datatranslation.com, Ninth Edition, July, 2007.

9. Siegwart R, Nourbakhsh IR (2004) Introduction to autonomous mobile robots. Intelligent Robotics and Autonomous Agents Series. The MIT Press, Cambridge, MA, USA. ISBN 0-262-19502-X.
10. LV-MaxSonar®-EZ0™ Data Sheet, www.maxbotix.com, Copyright 2005-2007.
11. SparkFun Electronics (www.sparkfun.com), Accessed 24 June 2008.
12. Jazzy 1122 the Owner's Manual (2006) www.pridemobility.com, Pride Mobility Products Corp.
13. Mohammed M, ElKady A, Sobh T (2007) New concept in optimizing manipulability index of serial manipulators, using SVD method. International Conference on Industrial Electronics, Technology and Automation (IETA 07).
14. Kamon I, Rivlin E, Rimon E (1996) A new range-sensor based globally convergent navigation algorithm for mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA.
15. Khatib O (1986) Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics Research, vol. 5, no. 1, pp. 90–98.
16. Park MG, Jeon JH, Lee MC (2001) Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing. Industrial Electronics, 2001 (ISIE 2001). IEEE International Symposium on Volume 3, Issue, 2001, pp. 1530-1535.
17. Borenstein J, Koren Y (1991) The Vector Field Histogram - Fast obstacle avoidance for mobile robots. IEEE Journal of Robotics and Automation, vol. 7, pp. 278–288.
18. Ulrich I, Borenstein J (2000) VFH*: Local obstacle avoidance with look-ahead verification. In Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, USA.
19. Ulrich I, Borenstein J (1998) VFH+: Reliable obstacle avoidance for fast mobile robots. In Proceedings of the International Conference on Robotics and Automation (ICRA'98), Leuven, Belgium.
20. Hui-zhong Z, Shu-xin DU, Tie-jun WU (2006) On-line real-time path planning of mobile robots in dynamic uncertain environment. Journal of Zhejiang University SCIENCE.
21. Dissanayake MWMG, Newman P, Clark S, Durrant-Whyte HF, Csorba M (2001) A solution to the simultaneous localization and map building (SLAM) problem. IEEE Transactions on Robotics and Automation, Volume 17, Issue 3, June 2001, pp. 229–241.
22. Guivant JE, Nebot EM (2001) Optimization of the simultaneous localization and map-building algorithm for real-time implementation. IEEE Transactions on Robotics and Automation, Volume 17, Issue 3, June 2001, pp. 242–257.
23. Hähnel D, Burgard W, Fox D, Fishkin K, Philipose M (2004) Mapping and localization with RFID technology. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2004).