R2D2: Rotating-turret 2D-scanning and Dead-reckoning for Remotely Operated Rovers over Resource Constrained Network Systems

Anandarup Mukherjee
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
India

Sudip Misra
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
India

Parakh Khandelwal
Department of Electronics and Communications Engineering
University of Engineering and Management Jaipur
India

***Abstract*—The use of rovers in navigating dangerous and uncharted terrains for localization, mapping and data-gathering for search and rescue in disaster management is a challenging domain, especially with constrained network connectivity. In this paper, a rover mounted rotating turret-based 2D ultrasonic distance mapping and Inertial Measurement Unit (IMU)-based dead-reckoning (R2D2) mechanism is proposed. The wireless rover is operated over a constrained network, allowing its handler to gather situational awareness of a hazardous site remotely, and without risking lives. A visualization of the path of this rover by means of a low data intensity fusion map generated using a combination of ultrasonic distance measurements and relative positioning information from the dead-reckoning system is performed by the remote operator.**

***Keywords*—** Constrained Networks, Mobile rover, 2D-Ultrasonic Scanning, Dead-reckoning, Situational Awareness

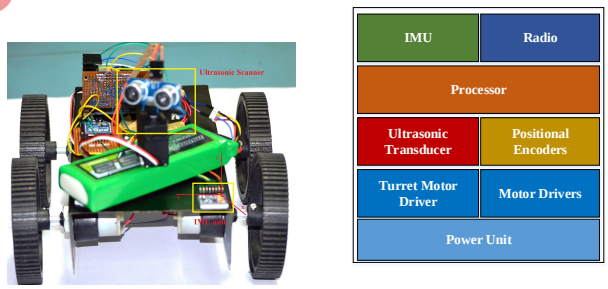
1. Introduction

The challenging domain of disaster management includes manpower-intensive search and rescue operations as well as planning-intensive task allocation and resource management. A disaster zone may be abundant with toxic leakages, fires, poisonous gases, and other hazards. The localization of a rover for its proper guidance, and mapping of the terrain around it are the primary tasks of a rover controlling system. Rovers armed with simultaneous localization and mapping (SLAM), camera-based imaging, 3D-active sensing, ultra-wideband (UWB) sensing, millimeter wave-based localization techniques, and other such technologies are some of the most costly and data intensive, but popularly utilized methods associated with search and rescue operations in global disaster management. The use of rovers as force multipliers in disaster zones is motivated by the increased speed of rescue efforts as well as, reduced exposure of the rescue workers to unforeseen risks. Localization and mapping in mobile rovers is important as it is not always possible to have an inkling of the terrain beforehand, hence networked robots are a popular choice in search and rescue [1]. Biologically inspired robots [2], fire-fighting robots [3] and military rescue robots [4] are some examples of the vast applications of robots and other autonomous rovers in challenging and dangerous environments.

An ultrasonic distance measurement scheme is most widely associated with positional awareness in the immediate vicinity of the rover with respect to its physical surroundings. It operates on physical transmission and reception of sound waves and its corresponding time-of-flight calculation, a method commonly known as echo-location. Typical ultrasonic sensors use a pair of transducers, one acting as a transmitter and the other as a receiver. The distance between the transducer and an obstacle is calculated on the basis of time of flight of the emitted ultrasonic wave using the formula:

$$d_{Tx-Rx} = \frac{v_s \times t_{sound}}{2} \quad (1)$$

Where, d_{Tx-Rx} is the distance between the source and the reflective surface, v_s is the velocity of sound and t_{sound} is the time taken by the wave to travel from the transmitter and receiver.



(a) R2D2 Rover (b) Functional Blocks

Figure 1: The R2D2 mechanism mounted on a remotely operated rover

1.1. R2D2 Rover

A scheme for mapping obstacles in the path of a rover using a rotating-turret based ultrasonic 2D-mapping system has been developed. The R2D2 system is mounted on a mobile rover as shown in Fig. 1(a). Fig. 1(b) shows the functional blocks of the rover and its integrated payload mechanism. The position of the rover is estimated by the direction it is facing and the type of obstacle in front of it, without the operator requiring to know the exact location of the rover at each time-step. The handler sends the directional control to the rover which activates the rotary encoder ensuring that the rover travels the distance given by the user. It works in a closed-loop system which ensures the distance

traveled by the rover is equal to the input given by the user. Once the distance from the starting point, the direction of travel, and the orientation of the rover is ensured, the turret based 2D-scan maps the area at that point. The scan system is designed to ultrasonically scan certain sectors only, depending on the rover's motion command received from the remote handler. The selective scan angles are shown in Figs. 3(a), 3(b) and 3(c) for forward motion, right turn, and left turn, respectively. This selective sectorial scan significantly reduces the obstruction map size as well as saves time as the rover does not have to scan to cover the whole 360^0 around it.

The major advantage of this system is its low, on-site processing requirement. The sensed data is amalgamated and transmitted to a remote server via a constrained network as shown in Fig. 2. Once the data arrives at the server, it is processed to generate distance traveled, direction of travel, and a fusion map. This information gets updated on the remote handler's console, enabling the handler to remotely guide the rover as well as map the terrain in the path of the rover. The mapped information serves as a guide for the local authorities in navigating the terrain and deciding upon the feasibility of choosing a path into the affected zone. A clear and information-rich map can be generated by using multiple such networked rovers. A multitude of sensors can be easily integrated to this system so as to generate a more sophisticated and enriched fusion map. Sensors such as fire detectors, gas sensors, Global Positioning System (GPS), proximity detectors and sound sensors would enhance the operational capability of the rover, and allow for its use in many more scenarios.

1.2. Network Architecture

Fig. 2 shows the architecture of the proposed system. The low-range wireless rovers are connected to an access point (AP) and placed on the exploration site. Each AP handles communication between the rovers under its zone and the outside network. A gateway connects multiple such APs to the backbone network leading to the remote server. The data from the rovers arrive at the remote server which concurrently gets stored in a database, as well as gets processed in the analytical engine. This analytical engine is responsible for generating situational awareness and rendering human-perceivable maps for use by the handler from the rover transmitted data. The rover's position and its surrounding information is displayed on a console which guides the handler in making informed decisions while remotely commandeering the rover through the unknown terrain.

2. Related Works

The task of autonomous localization or mapping uncharted spaces in the absence of traditional guiding technologies such as GPS or other reference markers is a big challenge, especially during cloudy weather and in indoor environments. Various indoor localization and mapping schemes have been proposed, such as the radio-map based localization scheme by Atia *et al.* [5], RSSI-based localization method by Zampella *et al.* [6], fuzzy adaptive extended information filtering (FAEIF) based localization scheme by Lin *et al.* [7], and laser rangefinder-based indoor navigation method by Luo *et al.* [8]. Additional approaches of indoor localization include a smartphone-based indoor pedestrian

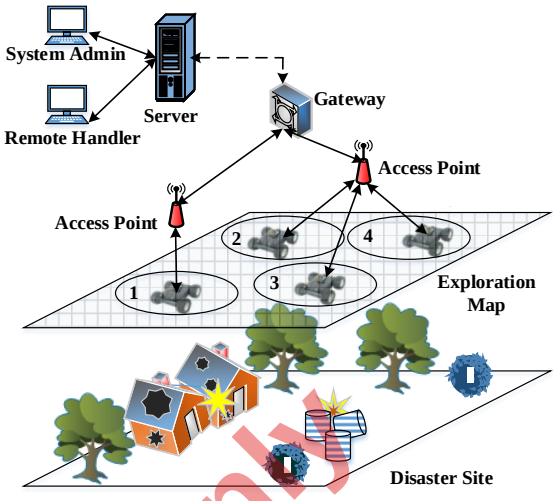


Figure 2: The network architecture for implementing R2D2-rover over a constrained network.

tracking and floor-map generation system by Shin *et al.* [9], a 3D-indoor mobile mapping system using information fusion between a 2D-laser scanner and RGB depth cameras as proposed by Wen *et al.* [10], a Kalmann filter-based localization with an occupancy grid-map for large-scale mapping of indoor environments as proposed by Seongsoo *et al.* [11], crowd-sourcing based indoor map generation by Zhou *et al.* [12], and others.

Echo-location based distance measurement techniques although simple, but have been popularly used for localization tasks. A fixed reference point-based ultrasonic echo-location of a rover, which is both position and orientation sensitive, is proposed by Kim *et al.* [13]. Similar approaches include a UAV-based obstacle detection and collision avoidance system using complementary low-cost sensors is proposed by Gageik *et al.* [14], and a multi-bot localization technique using ultrasonic and vision-based relative positioning as demonstrated by Silva *et al.* [15].

The above discussed approaches rely mostly on data and processing intensive techniques for localization and mapping. Another drawback of these approaches is that they have been designed for normal and peaceful scenarios and none of them consider poor network coverage which is typical to a disaster management or remote operation scenario. Our approach considers the use of networked robots for gathering situational awareness in hazardous areas as well as, the visual map generation at the remote server takes into account the lack of proper network infrastructure at the site of operation.

3. Mathematical Formulation

This section describes the mathematical models designed and applied for the vehicle localization technique discussed in this paper. At every step of the rover, which is adjusted by its handler, the rotating turret mechanism scans the vicinity of the rover using ultrasonic transducer. As the rover is intended for path finding missions, to speeden-up

the scanning process at each step, only the forward half of a representative circle is scanned, as shown in Fig. 3. The representative circle is the top-down view of the rover's position with respect to its surroundings, keeping the rover at the centre of the circle. Figs. 3(a), 3(b), and 3(c) show the angular scan sectors and their directions with respect to the rotating Rx-Tx module for forward, right, and left motion of the rover, respectively. The scanner rotates in the direction of the arrow with a step size of δ for n number of steps. To reduce unnecessary scanning time and minimize data, we have considered $\Theta = 90°$. However, the value of Θ can be easily reset according to the handler's controlling experience.

The obstruction detected by the system is denoted by O_M and is given as shown in equation 2.

$$O_M \supseteq f(\phi_L, \phi_S, \phi_R) \tag{2}$$

The scan module functions can be represented as:

$$\begin{aligned} \phi_L &\supseteq f(\Theta_L, 0, 0) \\ \phi_S &\supseteq f(0, \Theta_S, 0) \\ \phi_R &\supseteq f(0, 0, \Theta_R) \end{aligned} \tag{3}$$

For a given step size δ_Θ, the number of steps n, in a single scan is given by:

$$\begin{aligned} n\delta_\Theta &= \Theta \\ \Rightarrow n &= \frac{\Theta}{\delta_\Theta} \end{aligned} \tag{4}$$

The angle arrays Θ_L, Θ_S and Θ_R corresponding to left scan, straight scan and right scan, respectively are given as:

$$O_M = \begin{bmatrix} \Theta_L \\ \Theta_S \\ \Theta_R \end{bmatrix}^T = \begin{bmatrix} \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \\ \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \\ \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \end{bmatrix}^T \tag{5}$$

The generalized obstruction map data for a time instant t_S is transmitted back to the remote server from the rover, such that, $\forall t = t_S$:

$$Tx_{Map}(t) = \begin{bmatrix} \Theta_L \\ \Theta_S \\ \Theta_R \\ \Delta_O \end{bmatrix}^T = \begin{bmatrix} \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \\ \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \\ \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \\ \Delta_0, \Delta_{\delta_\Theta}, \Delta_{2\delta_\Theta}, \cdots, \Delta_{n\delta_\Theta} \end{bmatrix}^T \tag{6}$$

Equation 6 is dependent on the relations given in equations 2-5. To localize the rover with respect to its starting point, a method of dead-reckoning is used, which relies on IMU sensors for its operation. The dead-reckoning in the rover is achieved by deducing the position of the rover by making use of its previously determined position and traveling velocities over fixed intervals of time. Having previous knowledge of the rover's position and velocity enables the system to determine its current location and position with respect to its last known location. The IMU magnetometer provides x_{mag}, y_{mag} and z_{mag} readings with respect to movement of the rover along x, y and z-axes, respectively. These readings are however, not calibrated. The calibrated magnetic bearing of the rover is denoted by $M(t)$ such that, The obstruction detected by the system is denoted by O_M and is given as shown in Equation 2.

$$O_M \supseteq f(\phi_L, \phi_S, \phi_R) \tag{7}$$

The scan module functions can be represented as:

$$\begin{aligned} \phi_L &\supseteq f(\Theta_L, 0, 0) \\ \phi_S &\supseteq f(0, \Theta_S, 0) \\ \phi_R &\supseteq f(0, 0, \Theta_R) \end{aligned} \tag{8}$$

For a given step size δ_Θ, the number of steps n, in a single scan is given by:

$$\begin{aligned} n\delta_\Theta &= \Theta \\ \Rightarrow n &= \frac{\Theta}{\delta_\Theta} \end{aligned} \tag{9}$$

The angle arrays Θ_L, Θ_S and Θ_R corresponding to left scan, straight scan and right scan, respectively are given as:

$$O_M = \begin{bmatrix} \Theta_L \\ \Theta_S \\ \Theta_R \end{bmatrix}^T = \begin{bmatrix} \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \\ \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \\ \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \end{bmatrix}^T \tag{10}$$

The generalized obstruction map data for a time instant t_S is transmitted back to the remote server from the rover, such that Equation 6 is dependent on the relations given in equations 2-5, and $\forall t = t_S$:

$$Tx_{Map}(t) = \begin{bmatrix} \Theta_L \\ \Theta_S \\ \Theta_R \\ \Delta_O \end{bmatrix}^T = \begin{bmatrix} \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \\ \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \\ \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \\ \Delta_0, \Delta_{\delta_\Theta}, \Delta_{2\delta_\Theta}, \cdots, \Delta_{n\delta_\Theta} \end{bmatrix}^T \tag{11}$$

$$M(t) = \begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} \tag{12}$$

μ_x, μ_y and μ_z are the calibrated readings from the sensor along x, y and z axes, respectively. These values of μ_x, μ_y and μ_z are determined as shown in Equation 13.

$$\begin{aligned} \mu_x &= \frac{x_{mag}}{\sqrt{x_{mag}^2 + y_{mag}^2 + z_{mag}^2}} \\ \mu_y &= \frac{y_{mag}}{\sqrt{x_{mag}^2 + y_{mag}^2 + z_{mag}^2}} \\ \mu_z &= \frac{z_{mag}}{\sqrt{x_{mag}^2 + y_{mag}^2 + z_{mag}^2}} \end{aligned} \tag{13}$$

The IMU gyroscope provides the orientation values of the vehicle with respect to the Earth's inertial frame. It is represented by $G(t)$ such that,

$$G(t) = \begin{bmatrix} \gamma_x \\ \gamma_y \\ \gamma_z \end{bmatrix} \tag{14}$$

The values of γ_x, γ_y and γ_z represent the orientation vectors along x, y and z axes, respectively. The combination of obstruction-map data along with their corresponding dead-reckoning values generate a sensor fusion map which, when visualized all-together gives plentiful information regarding the surroundings and the motion of the vehicle. We denote the server-side generated fusion map by $\Im(t)$, such that,

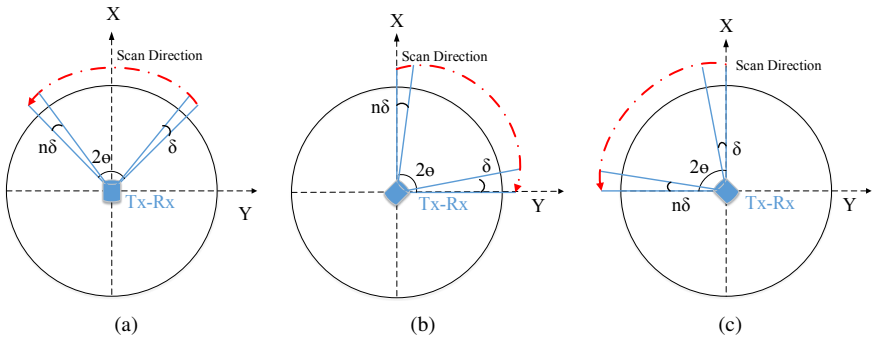


Figure 3: The rotating turret scan angles: (a) Forward motion, (b) Right turn, (c) Left turn

$$\Im(t) = \begin{bmatrix} \Theta_L \\ \Theta_S \\ \Theta_R \\ \Delta_O \\ \mu_x \\ \mu_y \\ \mu_z \\ \gamma_x \\ \gamma_y \\ \gamma_z \end{bmatrix}^T = \begin{bmatrix} \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \\ \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \\ \Theta_0, \Theta_{\delta_\Theta}, \Theta_{2\delta_\Theta}, \cdots, \Theta_{n\delta_\Theta} \\ \Delta_0, \Delta_{\delta_\Theta}, \Delta_{2\delta_\Theta}, \cdots, \Delta_{n\delta_\Theta} \\ \mu_{x0}, \mu_{x\delta_\Theta}, \mu_{x2\delta_\Theta}, \cdots, \mu_{xn\delta_\Theta} \\ \mu_{y0}, \mu_{y\delta_\Theta}, \mu_{y2\delta_\Theta}, \cdots, \mu_{yn\delta_\Theta} \\ \mu_{z0}, \mu_{z\delta_\Theta}, \mu_{z2\delta_\Theta}, \cdots, \mu_{zn\delta_\Theta} \\ \gamma_{x0}, \gamma_{x\delta_\Theta}, \gamma_{x2\delta_\Theta}, \cdots, \gamma_{xn\delta_\Theta} \\ \gamma_{y0}, \gamma_{y\delta_\Theta}, \gamma_{y2\delta_\Theta}, \cdots, \gamma_{yn\delta_\Theta} \\ \gamma_{z0}, \gamma_{z\delta_\Theta}, \gamma_{z2\delta_\Theta}, \cdots, \gamma_{zn\delta_\Theta} \end{bmatrix}^T \quad (15)$$

As the rover is remotely operated by a human handler using resource constrained networks, we consider the time of operation to be k units. Therefore, the server receives $\Im_{Server}$ after k units of time, such that,

$$\Im_{Server} = \Im_0^k = [\Im(0) \quad \Im(1) \quad \Im(2) \quad \cdots \quad \Im(k)] \quad (16)$$

This $\Im_{Server}$, when visualized, generates the server side fusion map. The rotary encoders which keep track of the rover's actual movement in the x-y plane senses the actual rotation of the wheel attached to the rover's motors. If the radius of the wheel, as calculated from the center of the shaft to the periphery of the wheel, is considered as r_w, the distance covered by the wheel during n rotations in time t can be easily computed as shown in Equation 17.

$$d_n(t) = 2\pi r n \quad (17)$$

The actual position of the rover after time t is $\rho(t)$, and is represented as,

$$\rho(t) = f(M(t), d_n(t)) \quad (18)$$

Let, ξ be the error in position of the rover with respect to the real position $\rho_R(t)$. The value of $\xi(t_k)$ can be estimated by:

$$\xi(t_k) = \rho_R(t) - \rho(t_k) \quad (19)$$

The Proportional-Integral (PI) control of the rover positioning is thus,

$$\Psi_{PI}(t_k) = k_P \xi(t_k) + k_I \int_{t_{k-1}}^{t_k} \xi(t_k) dt(k) \quad (20)$$

where, k_P and k_I are tuning gains of the PI controller and $\Psi_{PI}(t_k)$ is the corrected position of the rover at an instant of time t_k.

4. Methodology

This section discusses the working mechanism of the rover. The rover, after being placed and initialized at the remote location links to its operator over the network. The processor on the rover sends raw sensor data to the remote server. This data to the server contains values of the ultrasonic sensor, magnetometer, gyroscope, and the scan angle of the rotating turret servo motor. This data is used to plot the initial map of the mobile rover at stationary scanning points along with the magnetic bearing and orientation for the rover. The map gives the rover's handler a basic idea about the surrounding terrain. The user then sends short bursts of control signals for the motion of the mobile rover which enables it to travel a certain distance as commanded by the attached rotary encoder dependent PI controller mounted along the wheel. The PI controller checks the value from the rotary encoder and matches it with the value of the input given by the handler. If the difference is positive, the PI controller commands the rover to move in the direction specified by the user. Upon getting a negative difference, the PI controller commands the mobile robot to go in opposite direction till an equilibrium is reached. This ensures that the rover actually travels the distance as commanded by the handler. The rover, upon reaching the desired location, localizes itself using the R2D2 and generates a map, which gets updated and appended to the original map as it travels. This enables the rover system to create a continuous obstruction map of its locality, at every time step, as shown in Fig. 4. The processing of magnetic bearing and the orientation values at the server-end gives the user an additional information about the direction and orientation of the rover. The rover operates by receiving subsequent short bursts of command signals from the remote

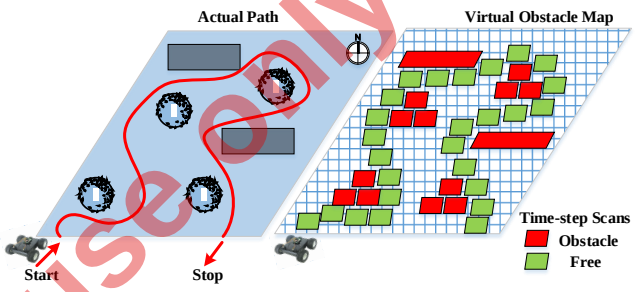


Figure 4: The rover's path bypassing obstacles, and the corresponding virtual obstacle map generated on-board the remote server.

handler and mapping each halting point prior to moving. If a forward command is received, the rover moves forward and then scans the area. But, in case of right or left turn, the rover first scans the right or left sector, as shown in Fig. 3.

5. Results

The rover is put under various conditions for generating remote responses on the handler console. Figs. 5(a)-5(d) show the server-side plot of the sensor readings from the remote rover. The top-most window in each figure shows a plot of the distance of the obstruction (in $cms.$) from the rover versus the scan angle of the Rx-Tx unit. Higher the value on the plot, further away is the obstruction from the rover, signifying that the path is open for traversal. The middle window shows the plot of magnetic bearing of the rover with respect to angle $(0-2\pi)$. The third window shows the plot of orientation of the rover with respect to angle $(0-2\pi)$.

5.1. Obstruction & Motion Maps

Fig. 5(a) shows the server-side output from the R2D2 sensors during straight motion of the rover. The first plot shows that the rover is free to move forward between the scan angle direction of 85° to 135°. The magnetic bearing and the gyroscope readings are more or less uniform in this case. Fig. 5(b) shows the R2D2 motion towards right. The first plot window shows that the rover can move in the direction of $100^\circ - 110^\circ$ without much obstruction. The magnetic bearing shows that the rover is following the designated direction only. Similar to the right turn, the top plot in Fig. 5(c) shows that the rover is free to move between 0° to 85°, if it turns to the left. The magnetic bearing also shows that the rover is headed in the right direction. The plots in Fig. 5(d) shows that the obstruction map is detecting lots of obstacles and there may not be enough path for the rover to pass through. However, the magnetic bearing shows high fluctuation in readings signifying sensor noise, which may be due to electronic fluctuations or mechanical impact on the sensor. Eventually, the gyroscope readings verify that there is uniform change in the rover's orientation along all three axes and, there is a possibility of the rover having flipped over. The plots in Fig. 7(a) show that there are a plenty of obstructions in the rover's path and it may not pass through if it continues in the same direction. The magnetic bearing shows sudden spikes on the z-axis signifying sudden rise and fall of the rover. The gyroscope z-axis also confirms sudden changes in orientation of the rover along z-axis which signifies that the rover has been picked up and its trajectory has been forcibly reset.

5.2. Server-side Fusion Maps

Fig. 6 shows the server-side fusion map generated from the rotating-turret based 2D-scanning and inertial sensor values. The image has been compiled for 6 steps of the vehicle motion – Δ to 6Δ – after receiving control signals from the remote handler. The server keeps on generating these images and updates the older one. Changes in the color of the plot signify obstructions or events.

The color-map is chosen such that blue signifies lesser magnitude and increase in red signifies increase in magnitude. The figure shows that most of the steps are not blue and have a greenish tinge. This signifies that the rover can continue moving freely. The bands formed in some of the images signify obstructions, which are the remnants of the ultrasonic distance measurement values. Blueish bands signify obstacles which are very close and must be immediately avoided. Figs. 7(a) and 7(b) show the server-side parameter plot and fusion map generated from the rotating-turret based 2D-scanning and inertial sensor values for forcibly obstructed or hijacked motion. The image has been compiled for 6 steps of the vehicle, from Δ to 6Δ, same as before. The bands formed in each of these images have mostly, non-bluish colors, signifying absence of nearby obstructions. However, it is evident from the blue color and contrasting colored bands in each of these step images, that the rover has been forcibly obstructed or its orientation changed.

6. Conclusion

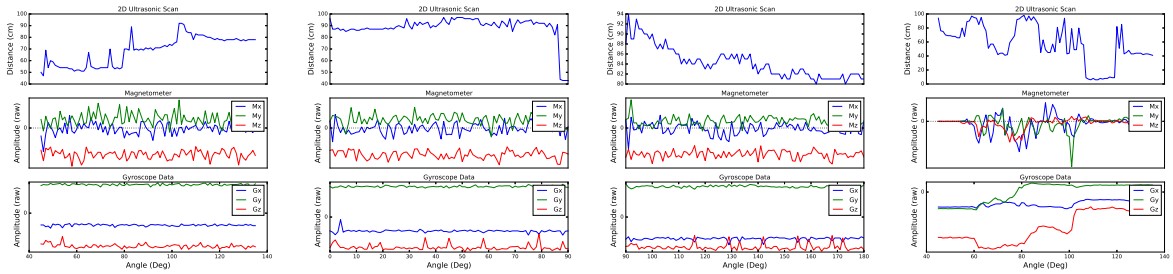
A low cost, low quality camera mounted on a rover would generate a frame size of 320×240 pixels. Keeping the video quality to lowest possible (say, 20 frames per second), a 1 second footage from the camera would generate a data size of approximately 1.53 MB. However, this kind of imaging would be seldom of any use in the task we are proposing to solve. Presence of smoke and gases will distort and inhibit the proper functioning of a camera-based rover. This approach becomes tedious and error-prone while using multiple networked rovers and that too over a network of poor quality.

However, the proposed approach uses selective sectional ultrasonic scanning resulting in a very small data footprint prior to transmission, which can be easily transmitted over poor or resource constrained networks. A single scan of a sector results in a data size of 90×10 Bytes. The rotating turret scans the sector 5 times per second, resulting in a data size of 4.5 kB per second. This data rate is many magnitudes below the rate generated by a low-end visual imaging solution, and can be easily transmitted over a network with poor service quality. The low data-rate also allows the handler to consecutively control multiple rovers by following the same process. This makes our solution a force-multiplier in disaster management scenarios. Additionally, this method is not at all hindered by gases or low visibility. Our proposed mapping and localization method tries to solve the common challenges faced by rovers in various situations, including search and rescues. The use of networked rovers for such tasks always require expensive resources, which may not always be available. Our solution gives an inexpensive method, both in terms of cost of implementation as well as networking and processing needs, for basic exploration of a disaster zone, prior to sending humans or other costlier resources. In the future, we plan on optimizing the fusion-map generation and take it a step further by making it more dynamic and having more easy to interpret visuals.

References

[1] H.-B. Kuntze, C. Frey, T. Emter, J. Petereit, I. Tchouchenkov, T. Mueller, R. Worst, K. Pfeiffer, M. Walter, S. Rademacher *et al.*, "Situation Responsive Networking of Mobile Robots for Disaster Management," in *Proceedings of ISR/Robotik 2014; 41_{st} International Symposium on Robotics*. VDE, 2014, pp. 1–8.

[2] I. Erkmen, A. M. Erkmen, F. Matsuno, R. Chatterjee, and T. Kamegawa, "Snake robots to the rescue!" *IEEE Robotics & Automation Magazine*, vol. 9, no. 3, pp. 17–25, 2002.



(a) Straight (b) Right turns (c) Left turns (d) Flipped over

Figure 5: Generated obstruction maps.

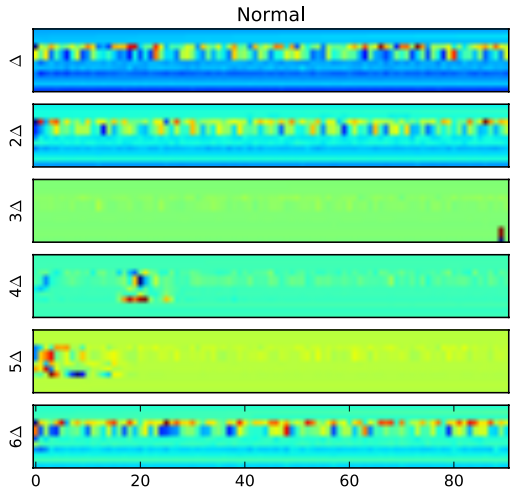
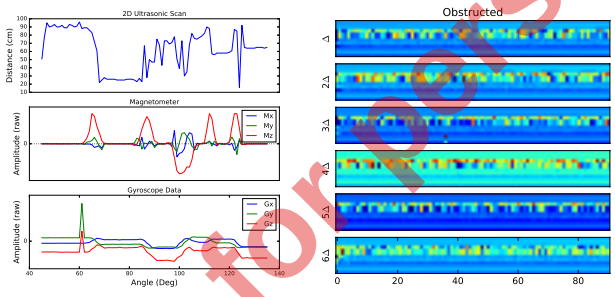


Figure 6: The server-side fusion map, $\Im_{Server}$ for unobstructed rover motion



(a) IMU parameter plots (b) Fusion map

Figure 7: Data and map generated for forcible rover obstruction.

[3] Y. Baudoin, D. Doroftei, G. De Cubber, S. A. Berrabah, C. Pinzon, F. Warlet, J. Gancet, E. Motard, M. Ilzkovitz, L. Nalpantidis *et al.*, "View-finder: robotics assistance to fire-fighting services and crisis management," in *IEEE International Workshop on Safety, Security & Rescue Robotics (SSRR)*. IEEE, 2009, pp. 1–6.

[4] P. Maxwell, D. Larkin, and C. Lowrance, "Turning Remote-Controlled Military Systems into Autonomous Force Multipliers," *IEEE Potentials*, vol. 32, no. 6, pp. 39–43, 2013.

[5] Atia, Mohamed M, Noureldin, Aboelmagd and Korenberg, Michael J, "Dynamic online-calibrated radio maps for indoor positioning in wireless local area networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1774–1787, 2013.

[6] F. Zampella, A. R. Jimenez Ruiz, and F. Seco Granja, "Indoor Positioning Using Efficient Map Matching, RSS Measurements, and an Improved Motion Model," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1304–1317, 2015.

[7] H.-H. Lin, C.-C. Tsai, and J.-C. Hsu, "Ultrasonic localization and pose tracking of an autonomous mobile robot via fuzzy adaptive extended information filtering," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 9, pp. 2024–2034, 2008.

[8] R. C. Luo and C. C. Lai, "Enriched indoor map construction based on multisensor fusion approach for intelligent service robot," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3135–3145, 2012.

[9] H. Shin, Y. Chon, and H. Cha, "Unsupervised construction of an indoor floor plan using a smartphone," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 6, pp. 889–898, 2012.

[10] C. Wen, L. Qin, Q. Zhu, C. Wang, and J. Jonathan Li, "Three-Dimensional Indoor Mobile Mapping With Fusion of Two-Dimensional Laser Scanner and RGB-D Camera Data," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 4, pp. 843–847, April 2014.

[11] S. Lee and S. Lee, "Embedded Visual SLAM: Applications for Low-Cost Consumer Robots," *IEEE Robotics Automation Magazine*, vol. 20, no. 4, pp. 83–95, Dec 2013.

[12] B. Zhou, Q. Li, Q. Mao, W. Tu, X. Zhang, and L. Chen, "ALIMC: Activity landmark-based indoor mapping via crowdsourcing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2774–2785, 2015.

[13] Kim, S Jin and Kim, Byung Kook, "Dynamic ultrasonic hybrid localization system for indoor mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 10, pp. 4562–4573, 2013.

[14] N. Gageik, P. Benz, and S. Montenegro, "Obstacle Detection and Collision Avoidance for a UAV With Complementary Low-Cost Sensors," *IEEE Access*, vol. 3, pp. 599–609, 2015.

[15] O. De Silva, G. K. Mann, and R. G. Gosine, "An ultrasonic and vision-based relative positioning sensor for multirobot localization," *IEEE Sensors Journal*, vol. 15, no. 3, pp. 1716–1726, 2015.