Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

A Mobile Hyper Redundant Mechanism for Search and Rescue Tasks

A. Wolf, H. B. Brown, R. Casciola, A. Costa, M. Schwerin, E. Shamas, H. Choset
Department of Mechanical Engineering,
Carnegie Mellon University
Pittsburgh, PA 15213, USA
Email: alon.wolf@cmu.edu, choset@cs.cmu.edu

Abstract - In this work we introduce a new concept of a search and rescue robotic system that is composed of an elephant trunk-like robot mounted on a mobile base. This system is capable not only of inspecting areas reachable by the mobile base but also to inspect unreachable areas such as small cracks, and pipes, using the camera mounted on its elephant trunk robot. In the report we describe the mechanical structure of the elephant trunk robot, the kinematic analysis of the structure, the robot control, and its human interface systems.

I. INTRODUCTION

Urban search and rescue (USAR) robots are not just an academic study anymore. Unfortunate past events such as the Mexico City earthquake in 1985, the 1995 Oklahoma City bombing, and the September 11 attack on the World Trade Center stress the need for these robots in assisting rescue workers in unreachable or unsafe places. In Mexico City earthquake, 135 rescue workers were killed, 65 of the 135 died while trying to search and rescue while going through confined spaces which were flooded trapping the rescue workers inside[1] [2].

The September 11 terror attack on the World Trade Center (WTC) was the first known time where robots were actually used in an USAR effort. In the WTC, robots were used to explore unreachable spaces by delivering real-time photographic information from an environment unreachable and hostile both to the rescue workers and their trained dogs. This unfortunate event provided an opportunity to test the existing USAR robotic systems in real rubble piles, to explore the robot human interaction, and to define new requirements and mechanical needs from these systems. Numerous works related to USAR robots were published. Some confront the challenge of platform development [3] [4] [5] [6] [7] [8] [9], and some address the software/human interface development issues [3] [10] [11] [12] [13] [14]. In [3], the researchers introduce the idea of marsupial and shape-shifting robots for USAR. This idea was further investigated in [4], where a team of marsupial robots were deployed to cover a large search area. Shape-shifting qualities are used to overcome obstacles, or get a better view point. Other robotic structure is studied in [7], where the researchers introduce a fire-fighting robot, capable of climbing rails such as those found on high-rise apartment complex balconies. Software/human interface development questions address control issues, multi-agent collaboration in the field [11], data flow from a multi-sensor system [12], and human interface which enables the operator to focus on driving the robot and less on understanding the incoming information [4].

The present work introduces a new concept for USAR robotic systems which addresses the requirements which were defined following the WTC experience. The present system is composed of an elephant-trunk like robot (ETR) mounted on a mobile robot (Fig. 3). Both the mobile base and the ETR are equipped with cameras that provide the operators with real-time images of the environment. The main goal of the ETR is to extend the sensing ability of the USAR robotic system by enabling the operator to take advantage of the high maneuverability of the ETR and to steer the camera around obstacles into small cracks and to places not accessible to the mobile base.

Elephant-trunk and snake robots are the main subject of several robotic researchers who seldom address them as hyper-redundant manipulators [13]. The maneuverability inherent in these types of biological structures and their compliance (i.e. their ability to conform to environmental constraints) allow them to overcome obstacles of significant complexity compared to conventional robots. Hence they became a challenge for imitation in robotics [15][16]. One of the pioneer works in this area was introduced by Hirose [14]. In this work he developed an impressive device that mimicked the locomotion of real snakes on the ground. This research continued in the early 1990's at Caltech with the planar *hyper-redundant* manipulator by Chirikjian and Burdick. Their contribution focused on novel end-effector placement algorithms for these robots [17] [13]. Recently, other researchers, such as Yim [18] at Xerox PARC, Miller [19] on his own, and Haith at NASA Ames [20], have duplicated Hirose's pioneering work on snake locomotion, where Yim and Haith used Yim's polybot modules to form modular hyper-redundant mechanisms. Modularity clearly has its benefits, but comes at a high cost, which manifests itself in a loss of strength and maneuverability. The electro-mechanical connection is Polybot's innovation, yet it also provides a point of weakness to the mechanism and makes the robot more bulky hence reducing its maneuverability. Modularity has ore value when the target configuration of the robot is unknown *a priori*. The challenge for a snake robot

0-7803-7860-1/03/$17.00 © 2003 IEEE

mechanism is to be strong enough to lift itself in three dimensions but be small and light enough to be useful to demonstrate even basic planning. The Pacific Northwest Labs developed a three-dimensional mechanism which was incredibly strong but moved too slowly and was too large. This robot moved too slowly because it was intended to be used for surgical bomb disarming, so that a technician could tele-operate this robot to probe the internals of a bomb without accidentally detonating it. Kinematically, the mechanism is a sequence of linearly actuated universal joints stacked on top of each other. Takanashi developed at NEC a new two-DOF joint for snake robots that allowed a more compact design (Figure 1a). This joint uses a passive universal joint to prevent adjacent bays from twisting while at the same time allowing two degrees of freedom: bending and orienting. This universal joint enveloped an angular swivel joint, which provided the two degrees of freedom. The universal joint being installed on the outside rendered the joint too bulky. Researchers at JPL "inverted" Takanashi's design by placing a small universal joint in the interior of the robot (Figure 1b). This allowed for a more compact design, but came at the cost of strength and stiffness (backlash). A small universal joint cannot transmit rotational motion at big deflection angles nor can it withstand heavy loads. Other known designs use cable/tendon actuation systems for driving the robot, yet this design is somewhat cumbersome and requires quite a big external driving system [14] [15] [21] [22].

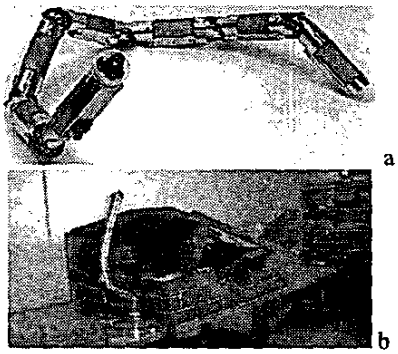

Figure 1 NEC Snake Robot JPL Serpentine Robot

In the following sections, we introduce the elephant trunk robot which was designed in our lab. Its mechanical structure (section II), low level control system (section III), high level control (section IV), image representation (section V), and motion planning (section VI).

II. MECHANICAL STRUCTURE OF THE ETR

Prior to the actual mechanical design, several design goals for the USAR-ETR were defined based on the experience acquired in the WTC and reports by CRASAR (center of robot-assisted search and rescue). These goals include: maximum torque-to-weight ratio to allow cantilever support of the snake, minimum envelope diameter to fit through small cracks, minimum achievable radius of curvature resulting from short links with maximum angular travel between links (for high maneuverability), and rugged construction. Secondary design goals include minimum backlash and compliance in the structure and "reasonable" speed of motion. From the outset, a modular design with all links identical was chosen for simplicity of design, fabrication, assembly, and maintenance. This requirement is sub optimal in the sense that the joints near the fixed end of the snake will generally have higher loads to carry than those near the ends.

A. Mechanical structure

We refer the reader to Fig.2 for a descriptive schematic of one stage of the ETR. The ETR is composed of fourteen actuated universal joints (U-joint) connected in a serial kinematic chain (Fig.3). An actuated universal-joint design was selected for its simplicity and ruggedness. In this design, U-joint "crosses" are connected to one link with a pitch pivot joint, and to the next with a yaw pivot joint. The pitch and yaw joints are always orthogonal, and intersect along the link centerlines; this leads to a relatively simple kinematic system. The pitch and yaw joints are actuated by linear actuators placed within the link's envelope. The links are configured such that the axes at each end of any link are parallel; thus, one link has pitch joints at both ends actuated by its two linear actuators; the next link has two yaw joints. This arrangement facilitates packaging of the two linear actuators side-by-side within the link.

Ball screws were chosen for the linear actuators because of their high efficiency (compared to lead screws) and effective speed reduction. The screws are fixed in bearings mounted to the links, while the nuts drive clevises connected to the crosses of the U-joints. The screws are driven by brush-type, permanent-magnet, DC motors which can be operated with simple, pulse-width-modulated (PWM) control. For compactness, the gearmotor and ball screw are placed side-by-side with a small toothed-belt drive connecting them. Each actuator is mounted to the link through a steel flexure that accommodated the slight lateral movement of the screw as the joint angle changes.

A novel feature of this design is the overload mechanism or "snubber". It is designed to absorb the kinetic energy of the links and motors when the mechanical stops are reached, and to accommodate imposed loads on the snake without damage to the actuators or structure. Belleville spring washers-4 series sets of 3 parallel-stacked washers are mounted in the

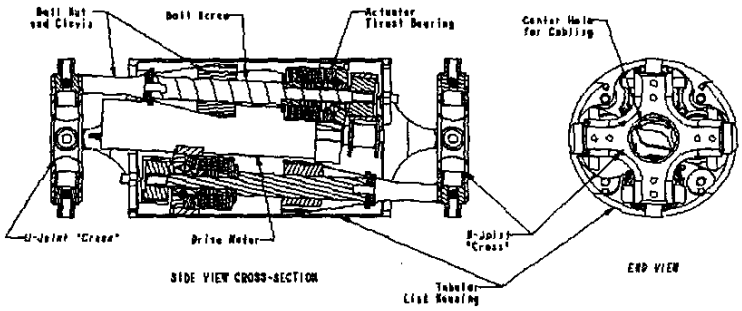

Figure 2: Link details

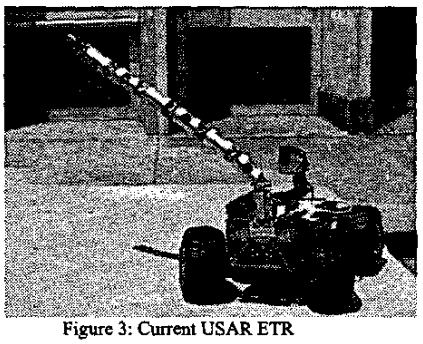

Figure 3: Current USAR ETR

"snubber housing" such that the ball screw can move axially by 1mm if the preload value is exceeded. The thrust load of the screw is taken by a custom-made, 4-point-contact bearing integrated into the snubber housing.

B. Mechanical properties

The ball screws are 6mm diameter with 1mm lead, rated at 700N, and are connected to the crosses at 14.7mm from the pivot. The motors used are Maxon RE-13 (13mm diameter) gearmotors with 16.58:1 planetary gear reducers and 16-count encoders (64 counts per revolution with quadruature encoding). The motors develop about 38mNm of continuous torque; this translates to 380N of force at the ball screw (well below the rated load), considering the 2:1 belt drive and transmission efficiencies. The snubber mechanisms are preloaded to about 600N to protect the ball screws and bearings from overload; no displacement occurs until this load value is reached, so the normal stiffness of the structure is not compromised. The motor no-load speed at the nominal 12V input is 8900RPM, which corresponds to 5s time to travel the full 22.4mm of screw travel resulting a ±55 degrees of joint angular travel.

Tests of the joints indicate that the actuators can produce 4.5Nm of torque at 12VDC (0.40A). That is, each ball screw produces 307N at 14.7mm radius on the U-joint cross. Based on the expected 5.08mNm at 0.40A, theoretical output would be 1060N with 100% transmission efficiency. This indicates that overall drive efficiency is only (307N/1060N) 29%, much lower than predicted (48%). This point will need further investigation to see if significant increases in efficiency and output torque are possible.

The torque about a joint needed to "cantilever-lift" (lift when extended horizontally) a single joint, assuming its center-of-mass (COM) to be at its geometric center, is 0.113Nm. The torque to lift n joints, in series, is n-squared times this. Given 4.5Nm available joint torque, the snake should then be able to cantilever-lift 6 joints. Tests on the complete snake robot confirm this capability. This ability is important to allow the snake to achieve arbitrary configurations working against gravity.

III. LOW LEVEL CONTROL

Hard wiring to all 14 actuators and encoders would require (14 x 6) 84 conductors to run inside the ETR envelope. This solution was deemed unfeasible; hence we decided to use an I^2C control bus (by Philips) in order to reduce the number of wires by connecting microcontroller circuit boards, placed inside the ETR, to the main control computer (Fig. 4). Although I^2C technology is an available technology, packaging the required components (H-bridge, decoder chip, PIC microcontroller plus passive components) to fit within the link envelope, and providing interconnects between controllers, are very challenging problems.

The current electronics design includes one custom circuit board per actuator (two per link). The boards are mounted on both sides of each U-joint, and (electrically) connected to each other through a hole in the cross via individual pin and receptacle contacts. This attempts to maintain modularity at the "link" level, and allows disassembly of the links at the crosses without desoldering wires. Boards on either end of each link are connected to each other via hard soldered wires. To remove individual boards (i.e., to disassemble a link itself), some desoldering of wires is necessary. Also, the motors are hard soldered to their controlling boards.

Each circuit board includes a Microchip PIC16F876 microcontroller, an LSI/CSI LS7166 24 bit (quadrature) counter, an Allegro A3953 H-bridge amplifier IC, a linear voltage regulator for logic power supply, and several passive components. The overall physical size of the board is 36mm x 23mm (Fig. 5).

The "bus" that runs through the robot consists of a total of nine wires: two supply voltages (~14V, and ~7V), separate grounds for each voltage, I^2C communication lines (SCL; SDA), a reset line (present on the connector but not currently wired through), a camera signal and ground for it. The board is equipped with a 19 pin intracross connector, three of which are dedicated to programming the microcontroller. The remaining pins can be assigned functions in the future.

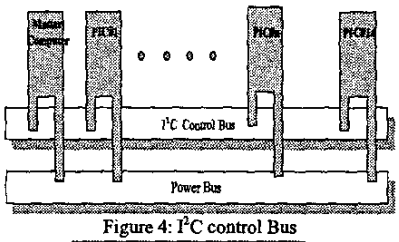

Figure 4: I²C control Bus

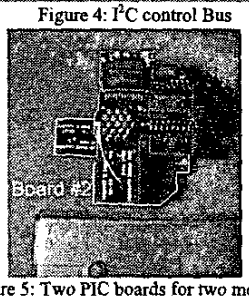

Figure 5: Two PIC boards for two motors

The firmware running on the PIC microcontroller includes support for the I²C protocol, both send and receive mode. Moreover, the code also perform PID control loop for the motor, connected to it, locally. For this use, the PIC reads the joint angle (quadrature encoder counter), and drive the H-bridge amplifier with a PWM signal according to the data received on the I²C bus. It also has the ability to communicate with the host computer over the bus and provides, upon request, the current joint angle. The I²C bus is capable of a 400 kBits/s. Currently only 120 KB/s are used for a 50 Hz data update rate on the bus. The PID control loop on board the PIC microcontroller runs at a 1 KHz rate. This frequency is limited by the current microchip that is being used and can be increased using a different model.

IV. HIGH LEVEL CONTROL, AND USER INTERFACE

When designing a USAR system which is operated at a disaster scene which does not resemble any lab condition, it is very important for the system to provide data which is as simple as possible for interpretation, and simultaneously, have a system which is easy to control. Based on this assumption we gave a lot of thought as to how to simplify the operation of such a complex multi degrees of freedom system, and came up with a user interface system which includes an on-board computer, a command computer and a single joystick. Next is a short description of the system.

The command interface for the USAR ETR system is written in Java; hence the system can be teleoperated either using a peer-to peer network or over the internet. We refer the reader to figure 6 for a block diagram of the system. As was described before, the USAR ETR system contains two cameras, one mounted on the end-effector of the ETR, and the second mounted on a pan-tilt servo system on the mobile base. One of the key requirements for the system was providing low latency, high frame rate video from both the ETR camera and the mobile base camera this was made possible using the Java Media Framework (JMF) to send video capture streams over RTP. JMF also provided a convenient way to add image rotation through the use of a custom codec. Image rotation helps to prevent the operator from becoming disoriented due to ETR orientation. With the current rotation scheme, the operator can trust that a vertical line in the image is in fact vertical in the real world. The rotation angle calculation depends on the ETR instantaneous configuration and is described in section V.

The command interface uses a joystick (SideWinder® Precision 2 Joystick) to control the motion of both the mobile base and the ETR. Switching between driving the mobile base and the ETR is done by pressing the trigger button on the joystick. When driving the mobile base, one can steer the base by pushing the joystick forward/backward and to the sides. While driving the base, the operator can simultaneously pan and tilt the camera, which is mounted on the base, using the hat-switch.

By switching to ETR control, the operator controls the ETR motion. By moving the joystick forward/backward and to the sides, the operator "aims" the snake (see section VI); while doing this he is also capable of curving/curling the snake using the hat-switch. The interface also provides joystick control for homing the ETR, and for toggling of the image rotation effect.

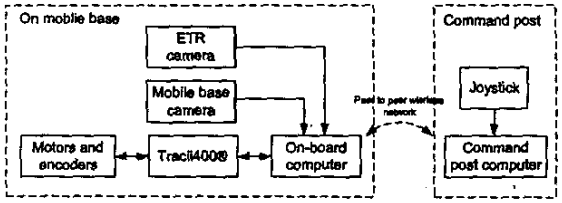

Figure 6: System block diagram

The mobile base (Fig.3) is equipped with an on board computer which captures video from the two cameras over USB, and then compresses these raw video streams into JPEG over RTP streams using JMStudio, which is part of JMF. This base computer also hosts a server which receives ETR and mobile base motion instructions from the command station. The command station sends the desired joint angles to this server, which then delivers this information over the I²C bus to the local controllers. The signals to the motors and from the encoders are transferred over the I²C control bus using IPORT™ as an interface. IPORT™ is a hardware-based 120 kBit/s full-speed I²C bus interface featuring the I²C master and slave-mode.

As was mentioned before, the PID control loop runs on the local PIC microcontroller. The "Command Post Computer" (CPC) reads the joystick position and system's configuration and sends accordingly a packet to the "On-board Computer" (OBC), which contains commands addressed to one of the fourteen motors driving the ETR or the four servos driving the mobile base system. When receiving a packet on the local network, the OBC sends a position command, over the I^2C control bus, to the addressed microcontroller. The local PID control loop then generates a value to the local ETR motor which represents joint velocity; this value is then translated, on-board the microcontroller, to a PWM signal to drive the motor. In the case that the packet is addressed to one of the mobile-base servos, the local PIC microcontroller does not initiate a PID control loop, but generates a pulse whose length is proportional to the required servo position.

V. IMAGE REPRESENTATION

One of the most challenging issues for USAR robots operators is the on-line interpretation of the images received by the cameras on board the robot [13]. This is challenging because objects in a disaster scene usually do not appear in their natural orientation and location. Therefore, crucial questions like "is the rod on top of the survivor or under?" are sometimes hard to answer. This issue is even more challenging when one of the cameras is mounted on the end-effector of an ETR for maneuverability extension. This is because the camera is oriented in space as a function of the joint parameters, i.e. the instantaneous robot configuration.

In this section, we present an algorithm that solves this problem by resolving the angle of rotation of the image as a function of the robot instantaneous configuration. Using this angle the image is then rotated, on-line, and the viewer is provided with an image as if is taken by a human eye, such that a vertical line in the real world will appear also as vertical in the image. Next is a description of the algorithm.

Given an instantaneous configuration of the ETR, i.e. a set of kinematic parameters of the robot's joints, one can derive the transformation matrix, ${}^0A_T$, from world coordinate frame (WCF) to tool coordinate frame (TCF), which is, in our case, also the camera coordinate system such that the image is facing the positive $\hat{Y}_T$ direction. Figure 7 describes the kinematic skeleton of the ETR and the local coordinate frames used for the DH (Denavit Hartenberg) method. Given the DH parameters, one can calculate the ${}^{i-1}A_i$ homogeneous transformation matrixes between coordinate frames i and i-1, and then calculate ${}^0A_{14}$ as:

$$ {}^0A_{14} = \prod_{i=1}^{14} {}^{i-1}A_i = \begin{bmatrix} R_{3\times3} & p_{3\times1} \\ 0_{1\times3} & 1_{1\times1} \end{bmatrix} \quad (1) $$

Let us define the vectors $\hat{X}_{14}, \hat{Y}_{14}, \hat{Z}_{14}$ in coordinate system 14 (tool coordinate system: T). $\hat{X}_T, \hat{Y}_T, \hat{Z}_T$ can be expressed in world coordinate frame (WCF) i.e. coordinate system 0, as:

$$ \begin{aligned} {}^0\hat{X}_T &= [x_1, x_2, x_3] = R \cdot \hat{X}_T \\ {}^0\hat{Y}_T &= [y_1, y_2, y_3] = R \cdot \hat{Y}_T \\ {}^0\hat{Z}_T &= [z_1, z_2, z_3] = R \cdot \hat{Z}_T \end{aligned} \quad (2) $$

Also, let us define a plane, P, which is normal to the ground and contains $\hat{X}_T$ (Fig. 8), a normal vector, $\hat{N}$, to P, (in WCF) has its y component equal to zero and is given by:

$$ \hat{N} = [n_x, n_y, n_z] = \frac{[\frac{-n_z \cdot x_3}{x_1}, 0, n_z]}{\left\| [\frac{-n_z \cdot x_3}{x_1}, 0, n_z] \right\|} \quad (3) $$

Substituting $n_z = 1$ in Eq.3, one can determine $\hat{N}$ for every given instantaneous configuration of the robot.

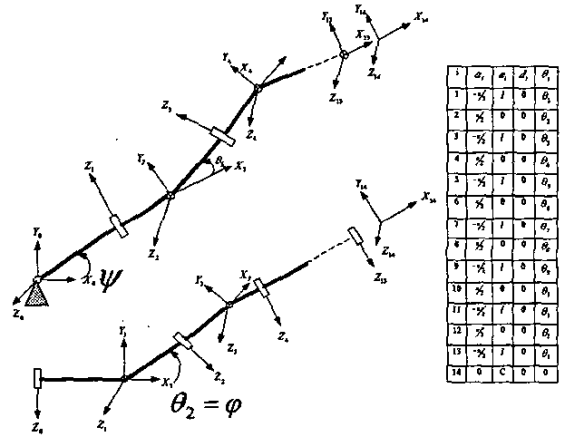

Figure 7: Kinematic skeleton of the ETR, and its DH parameters

Given that the camera is mounted on the ETR such that, in the home position, the image is facing the positive $\hat{Y}_T$ direction, we want $\hat{Y}_T$ to be contained in P such that the image is always facing the "up" direction and is horizontal. In order to accomplish this, one needs to rotate the image β degrees (Fig. 8) with respect to $\hat{X}_T$ such that both $\hat{Y}_T$ and $\hat{Y}_P$ are collinear. Let us define $\hat{Y}_P$; a vector in P and normal to ${}^0\hat{X}_T$ such that:

$$ \hat{Y}_P = [yp_1, yp_2, yp_3] = \hat{N} \times {}^0\hat{X}_T, \; {}^0\hat{X}_T \times \hat{N} \;\; | \;\; yp_2 > 0 \quad (4) $$

Observing Fig. 8 and Eq.3 and Eq.4 one can detect that $\hat{Y}_P \perp \hat{X}_{14,0}$ and ${}^0\hat{Y}_T \perp {}^0\hat{X}_T$, hence one can rotate ${}^0\hat{Y}_T$

β degrees, with $\hat{X}_T$ as the axis of rotation, such that it is collinear with $\hat{Y}_P$, where β is given by (Fig.8):

$$\beta = a\cos\left({}^{0}\hat{Y}_T \bullet \hat{Y}_P\right) \quad (5)$$

Remark: in case that Eq.5 reveals $\beta < 0$ $\beta = \pi/2 - \beta$. An example of the effect of this algorithm is given in Fig.9.

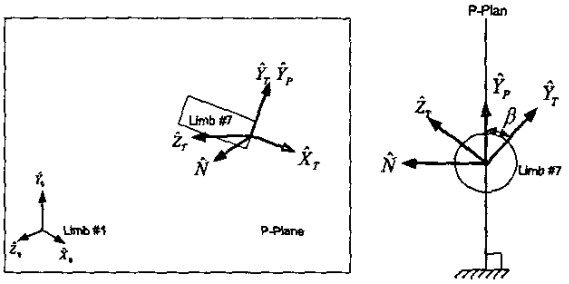

Figure 8: Tool coordinate frame

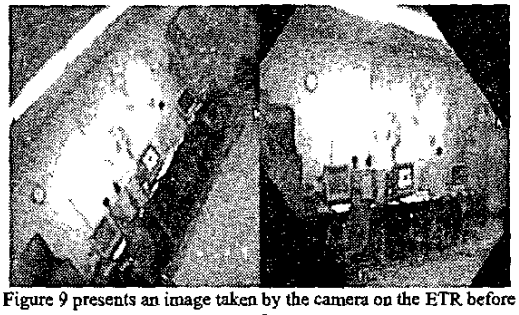

Figure 9 presents an image taken by the camera on the ETR before rotation (left) by β and after (right).

VI. MOTION PLANNING

Motion planning for ETR, snake robots and hyper redundant robots, in general, has been always a great challenge, due to the redundancy of joint parameters and limbs one has to control in order maneuver in a confined 3D environment [17] [18]. This issue is a greater challenge when motion planning takes place "on the fly", while driving a real system. The main challenge here is to have an algorithm that can be implemented without introducing much of a lagging to the system.

Taking into account that our system runs on real time, we introduced two modes of motion for the USAR ETR. In the first mode, the operator "aims" the snake in a given direction. This is achieved by moving joint 1 and 2 so that the snake can pan φ degrees, and tilt ψ degrees (Fig. 7) as a straight line, i.e. $\theta_{3-14} = 0$. While performing this motion, the end-effector (camera) moves on the work envelope of the robot.

After panning and tilting the ETR to a chosen direction, the robot can perform its second mode of motion, i.e. curve/curl with respect to that direction. When curving, the robot forms a parabolic shape (fig. 10). When curling, each limb is rotating separately until reaching its limit, and only then the successive limb follows the same motion and so-on.

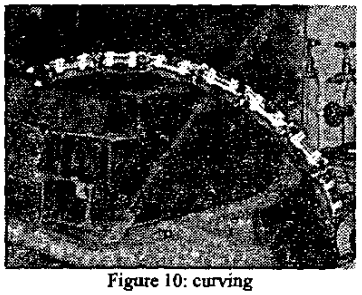

Figure 10: curving

The curving/curling motion is restricted to 4 planes: vertical (P-Plane) horizontal (P2) and tilted in ±45 degrees (P1, P3). These planes are oriented relative to the pan and tilt angle such that the snake (limbs 2-7) are contained in the P-plane of figure 8. We realize that this motion seems somewhat restricted, yet as one can detect from the following equations (Eq.6-8), the calculations . The restriction is affected by hardware γ are general in limitation. i.e. the hat-switch which controls the curving/curling motion (see section IV) is restricted to eight configurations. However, we would like to point out that these motions cover large portions of the ETR workspace. Moreover, after operating the system, we find this restriction helpful when maneuvering the ETR while trying to avoid obstacles. Figure 11 provides a general case of curving/curling for each of limb of the ETR (in is a $\hat{V} = (vm_1, vm_2)$ local coordinate frame) where normalized velocity vector and M_1 and M_2 stand for the horizontal and vertical motors normalized velocity components respectively. vm_1 and vm_2 are the projection , stands for γ on M_1 and M_2 axis respectively, and $\hat{V}$ of curving/curling direction such that in a general case of curving/curling:

$$\hat{V} = \begin{bmatrix} vm_1 \\ vm_2 \end{bmatrix}^T = \begin{bmatrix} V \cdot \cos(\gamma) \\ V \cdot \sin(\gamma) \end{bmatrix}^T \quad (6)$$

In case of curving/curling in P-Plan: $\gamma = \pm 90^\circ$, in case of P2: $\gamma = 0^\circ, 180^\circ$, in case of P1: $\gamma = 45^\circ, 225^\circ$ and in case of P2: $\gamma = -45^\circ, 135^\circ$.

Curving/curling in all planes when either $\psi = 0, \varphi = 0$ or both equal to zero is a relatively simple motion for our ETR due to its kinematic structure ($\beta = 0$). In these cases all uneven joint axes are parallel to the ground and all even are normal to the ground such that curving/curling in P-plan is achieved by rotating all uneven joints in equal speed (CCW, or CW). Curving/curling in case of $\psi = 0$ or $\varphi = 0$ in p1 plane is

achieved by rotating all uneven and even joints together in a constant velocity (CCW, or CW) and for p1 plane all uneven and even joints are rotated together in a constant velocity when one group is rotated CCW and the other CW and visa-versa.

Curling/curving the ETR becomes a little tricky when both ψ and φ are not equal to zero. In this case all uneven joint axes and even axes are tilted relative to the horizontal and vertical direction by β degrees, where β is given in Eq.5 (see Fig.8). In those cases, motion in all planes is achieved by simply multiplying $\hat{V} = (vm_1, vm_2)$ by:

$$\begin{bmatrix} \cos(\beta) & \sin(\beta) \\ -\sin(\beta) & \cos(\beta) \end{bmatrix} \quad (7)$$

such that:

$$\begin{bmatrix} v'm_i \\ v'm_1 \end{bmatrix} = \begin{bmatrix} \cos(\beta) & \sin(\beta) \\ -\sin(\beta) & \cos(\beta) \end{bmatrix} \begin{bmatrix} vm_1 \\ vm_1 \end{bmatrix} \quad (8)$$

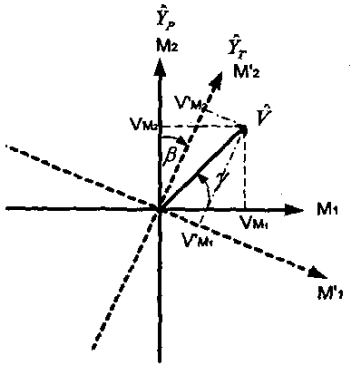

Figure 11: Projection of motors velocity

VII. CONCLUSION

In this work we introduce a new concept of a search and rescue robotic system that is composed of an elephant trunk-like robot mounted on a mobile base. This system is capable not only to inspect areas reachable by the mobile base but also to inspect unreachable areas such as small cracks, and pipes, using the camera mounted on its elephant trunk robot. The report introduces both low level as well as high level control of the USAR-ETR and its user interface. All of these components were designed to serve the system purpose, search and rescue. The system has been presented and tested at the IEEE Workshop on Safety, Security and Rescue Robotics held in Tampa, Florida in February 2003.

VII. REFERENCES

[1] United States Fire Administration and National Fire Association. *Rescue Systems I*, 1993.

[2] J. Casper, M. Micire, and R. Murphy, Issues in Intelligent Robots for search and Rescue, *SPIE Ground Vehicle Technology II*, 4: 41-46, 2000.

[3] R. Murphy, Marsupial and Shape-Shifting Robots for Urban Search and Rescue, IEEE Inteligent Systems, 15(3): 14-19, 2000

[4] R. Murphy, and J. J. Martinez, Lessons Learning from the USF REU Site Grant: Multiple Autonomous Mobile Robots for Search and Rescue Applications, In *Frontier in Education*, 1997.

[5] S. Hirose, Snake, Walking and Group Robots for Super Mechano-System. *IEEE SMC'99 Conference Proceedings*, Pages 129-133, 1999.

[6] A. Kobayashi and K. Nakamura, Rescue robots for Fire Hazards, *Proceedings of the 1983 International Conference on Advanced Robotics*, pages 91-98, 1983.

[7] H. Amano, K. Osuka, and T. Tran, Development of Vertically Moving Robot with Gripping Handrails for Fire fighting, *Proceedings of the 2001 International Conference on Intelligent Robots and Systems*, pages 661-662, 2001.

[8] A. Castano, W. M. Shen, And P. Will, CONRO: Toward Deployable Robots with Inter-Robot Metamorphic Capabilities, *Autonomous Robots*, 8(3): 309-324, 2000.

[9] R. M. Voyles, TerminatorBot: A Robot with Dual –Use for Manipulation and Locomotion, *Proceedings of the 2000 International conference on Robotics and Automation*, Pages 61-66, 2000.

[10] J. S. Jennings, G. Whelan, and W. F. Evans, Cooperative Search and Research with a Team of Mobile Robots, In *1999 8th International conference on Advanced Robotics*, pages 193-200, 1997.

[11] R. Murphy, J. Casper, J. Hyams, and M. Micire, Mixed-Initiative Control of Multiple Heterogeneous Robots for Search and Rescue, *Technical report, University of South Florida*, 2002.

[12] R Masuda, T. Oinuma, and A. Muramatsu, Multi-Sensor Control System for Rescue Robot. *In 1996 IEEE/SICE/RJS International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 381-387, 1996.

[13] Chirikjian, G. S. and Burdick, J. W, A modal approach to hyper-redundant manipulator kinematics, In *IEEE Transactions on Robotics and Automation* 10: 343-354, 1994.

[14] Hirose, S. Biologically Inspired Robots: Snake-like Locomotors and Manipulators. Oxford University Press: Oxford 1993.

[15] W. M. Kier, K.K. Smith, Tongues, Tentacles, and Trunk: The Biomechanics of Movement in Muscularhydrostas, *Zoological Journal of the Linnean Society*, 8: 307-324, 1985.

[16] M.A. Hannan and I.D. Walker. A Novel "Elephant's Trunk" Robot. *In the proceedings of the 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Atlanta, Georgia, September*, pp. 410-415, 1999.

[17] Chirikjian, G. S. and Burdick, J. W, Kinematically optimal hyper-redundant manipulator configurations, *IEEE Transactions on Robotics and Automation* 11: 794-806, 1995.

[18] http://robotics.stanford.edu/users/mark/bio.html (06/2003)

[19] http://www.doctorgavin.com/ (06/2003)

[20] Haith, G.L., Thomas, H., Wright, A., A Serpentine Robot for Planetary and Asteroid Surface Exploration. Oral Presentation at the Fourth IAA International Conference on Low-Cost Planetary Missions, May 2-5, 2000, Laurel, MD.

[21] R. Cieslak, A. Morecki, Elephant Trunk Type Manipulator-A Tool for Bulk and Liquid Materials Transportation, *Robotica*,.17: 11-16, 1999.

[22] G. Immega, K. Antonelli, The KSI Tentacle Manipulator, *IEEE Conference on Robotics and Automation*, pp. 3149-3154, 1995.