AVITRA

Surveillance, Disaster Mitigation and Mobile Rescue Robot with SLAM and human movement imitation

Technical Description :

India and many other countries have faced situations where expedited human intervention for rescue or mitigation, could not be realized. A very relevant example being the 2011 nuclear accident at the Fukushima Daiichi nuclear power plants, where the situation turned too dangerous for humans to enter in order to inspect the damage, owing to the risk of exposure to radioactivity. Despite tremendous progress in robotics and its allied fields over the decade, the mobile rescue robots that existed during the crisis, were not technically versatile to carry out disaster response missions. We thereby propose an economical and efficient robotic solution implemented using Robot Operating System (ROS) on the Nvidia Jetson Tx2 module that combines the robustness of robots and intelligent sensing systems with a human in the loop feedback system to perform exploration and surveillance. The robot will navigate through the terrain, implementing SLAM(Simultaneous Localization and Mapping) powered by Nvidia Jetson.

We propose Human-in-the-Loop feedback to increase the problem solving capability of our system while maintaining the performance during intermittent communications. We aim to develop a robot capable of replicating human actions in real-time so as to achieve the same effect that would be caused by a human being if they were to be sent for the mission. This process of using imitation as a substitute to conventional coding and control of the robot will result in accurate human-like motor skills that are often required in critical situations. Such a real-time imitation process can be extended to form a learning model resulting in a better decision making process in case of communication failure.

Subsystems Overview:-

Mechanical:

The robot is primarily a humanoid mounted on an Omni-Driven base. The key advantage of omni-drive systems is that translational and rotational motion are decoupled for simplified navigation. This will allow the robot to move around swiftly. The structure would be having two manipulators (arms) which would mimic a remote human operator. The manipulators would compose of aluminium links housing servo

motors which allow their movement in space.The robot has 3 degrees of freedom in both arms (shoulder adduction, shoulder flexion, elbow flexion) along with pneumatic system for elevating the robot's manipulators.

Electronics and Sensing:

An operator will perform an action in the control station which will be sensed by the Microsoft Kinect depth sensing camera. The movement of the operator will be tracked and sent to the robot for replication, thus the robot's manipulators would be completely dependent on the action of the person. Such a Human-in-The-Loop system would enable the robot to carry out complex actions at the disaster site. The human would be able to control the robot's actions based on the visual feedback received from the robot. Another Kinect camera is placed on-board to perceive the environment in 3D.

This will facilitate the generation of the sitemap. The robot will be equipped with two on-board computers i.e. one embedded controller and the Nvidia Jetson. The embedded controller (atmega 2560) will be used for controlling the motion of the robot's base. The Jetson is the brain of all the computation. The Jetson receives manipulation data from the remote centre. It also generates the 3-D map from the camera feed from Kinect obtain and relay them to the command centre. The live camera feed will help the remote human operator to visualize and understand the robot's environment allowing him to swiftly control the robot's actions.

Software:

The various entities involved in the project exchange data through ROS (Robot Operating System). ROS is based on publish-subscribe architecture which allows different nodes to publish data to a master and subscriber data from a master. The master has registration API's which allow the nodes to register as publisher, subscriber or both. ROS uses standard TCP/IP sockets to transport the message data. ROS allows us to run various subsystems on separate nodes that are independent of each other. This architecture is of utmost importance in critical tasks like defence operations. This has an advantage over conventional robotic hardware coding such that even if one subsystem fails, only that particular node is affected and the rest of operations running on other nodes continue without interruptions.

The project can be divided into the following major parts:

1. Mimicking the human operator
2. Teleoperation of the robot
3. Map generation

BIOMIMICRY

- **Sensing**

  Human Body above the torso will be mapped to a skeleton frame using the Microsoft Kinect (Depth Camera) and Openni software package from the ROS repository. The camera will be linked via a USB port to a computer running a linux OS (Ubuntu) and a ROS node (**Sensing node**). This node captures the data returned by the camera and relays it to the robot which hosts the Jetson Tx2 GPU. The camera returns the coordinates of shoulder, elbow and wrist of both the hands along with the rotation quaternions of the said frames which allows us to compute the direction along which the links (bones) point. The GPU operates multiple ROS nodes of which one captures the data sent by the sensing node. This node (**Computing node**) calculates the possible joint angles based on the skeleton coordinates.

- **Computation**

  The joint angles are calculated using the Denavit-Hartenberg convention (D-H parameters) for a *n* DOF manipulator. Iterative and Analytical methods were used for calculating the joint angles, but the latter was finalized as it resulted in faster and accurate results. The method of solving for joint angles (Inverse Kinematics) yielded multiple solutions for the joint angles as the end effector of the manipulator had the same position for different poses of the manipulator (Singularities). Therefore, the motion of the manipulator was restricted using different algorithms to avoid certain conflicting conditions. These algorithms involved mapping other coordinates of the body and sensing relative positions.

- **Instructing**

  The computed angles are written to the servo motors (Dynamixel AX-12) using serial communication between the Nvidia Jetson Tx2 and a microcontroller (Atmega2561). The above model of servo motors have built-in controllers that provide real time feedback of the motor position, torque, temperature and current. They also allow the user to configure the motors for variable speed, torque and positional limits. The computing node uses the Dynamixel-SDK (ROS Package for servo motors) and controls the motors individually.

  The above procedure successfully allows us to control the robot's actions with a human-in-the-loop control paradigm. The next step involves maneuvering the motion of the robot at the concerned site.

TELEOPERATION OF THE ROBOT

- **Remote control**
  An operator from the control station can maneuver the robot using a standard keyboard / joystick input. The remote computer connected to the Kinect sensor is used for the same by running a second ROS node on the computer. The input from the operator is relayed to the GPU which further sends commands to the microcontroller serially. The microcontroller runs a virtual ROS node which receives the input command and controls the motors accordingly. The motors are coupled with rotary encoders that allow us to estimate the position of the robot relative to the starting position of the robot. The encoder data is used to estimate the robot position along with its orientation.The encoder data is then floated through the microcontroller to Jetson Tx2 for further processing.

MAP GENERATION

Localization of the robot and generation of the site map along with it is a problem which has been intensively studied and still worked upon. We achieved SLAM by using a Kinect Sensor mounted on the robot's top which allows us to generate a 3-D Map of the concerned site and also a 2-D Map along with the path that the robot traverses.
Slam being computationally heavy, could not be achieved on the go with other processors (RaspberryPi) and this required a lot of post-processing to be done after the process. But Nvidia Jetsons 256 CUDA cores allowed the generation of the site map and live stream of the site visuals during the operation.