GOVERNMENT POLYTECHNIC

PURNEA, BIHAR

Name : Aditya kumar

Branch : C.S.E. Sem. : 5th

Class : 5th semester Roll No. : 511191821602

Subject : Robotics Basic lab

Session : 2021-2024 Year : 3rd

Board Roll No. : 511191821602

NAME: Aditya Kumar YEAR: 3rd

STD: DIV: ROLL NO.: 51119182 1602 **YOUVA**

SUBJECT: Robotic Basic Lab SEMESTER: 5th

INDEX

SR. NO.	EXPERIMENT	DATE OF EXPERIMENT	DATE OF SUBMISSION	REMARKS / SIGNATURE
1.	Identify components and different configuration of Robots.	06-11-2023	07-11-2023	
2.	Use different types of Robotic sensor for specific situation.	07-11-2023	08-11-2023	
3.	perform Robot control with stepper motor interfacing.	27-11-2023	28-11-2023	
4.	perform pick and place operation simulation control Software.	28-11-2023	29-11-2023	
5.	programme 5-axis robotic arm to control various motion.	04-12-2023	05-12-2023	[illegible] 20-01-24

Q.(1) Identify Components and different configurations of robots.

Introduction: Robots are complex machine that can have various components and configurations depending on their intended purpose and application. Here are some key components and different configurations of robots.

(i) Actuators:-

Electric Motors: Commonly used for tasks requiring precise and controlled momements.

pneumatic actuators:- use compressed air for high speed movements, often see in industrial robots.

Hydraulic Actuators:- use pressurized fluids for heavy-duty applications, like construction or manufacturing.

(ii) Sensors:-

Vision system:- Cameras and image processing software for recognizing objects and environments.

Proximity sensors:- Detect nearby objects and obstacles.

Range sensors: Measure distances to objects.

Inertial sensors: Accelerometers and gyroscopes for balance and orientation

Force/Torque sensors: Measure force and torque applied robot's end effector.

Touch sensors: Detect contact with objects or surfaces.

Environmental sensors: Measure temperature, humidity and other environmental conditions.

(iii) Control system:

Microcontroller or Microprocessor:

The brain of the robot, responsible for processing sensor data and generating control signals.

1. Motor Drivers:

Control the movements and speed of actuators.

Feedback Loops:

Mechanism for comparing desired and actual states and adjusting robot actions accordingly.

Programming Interface:

Software or hardware used to program and control the robot.

End-Effector:

The tool or attachement at the robot's end used to interact with the environment.

Examples include grippers, welding tools, or 3D printers.

(v) Mobility :-
Robots can have various mobility configurations, including :

wheeled Robots : use wheels for movements, suitable for flat and smooth surfaces.

Legged Robots :
Mimic animal like walking for more complex terrains.

Tracks or Treads :
provide stability and traction in rough environments.

Aerial Robots :- fly in the air.

Underwater Robots.
operate underwater.

Robotic Arms :
fixed-base or mobile arms with multiple joints for precise manipulation.

(vi) power source:

Robot's can be powered by batteries, electrical cables, internal combustion engines, or other energy sources depending on their application and mobility.

(vii) Robot configurations:

Industerial Robots: used in manufacturing and assembly processes often have multiple degrees of freedom and precise control.

Mobile Robots:
Designed for autonomous navigation in environments like warehouses, hospital, or outdoor areas.

Humanoid Robots:
Mimic the human form and are used in research, entertainments and assistive applications.

Medical Robots:
used in surgery, diagnostics, and patient care.

Scanned with OKEN Scanner

classmate
Date :
Page :

Agricultural robots:
Designed for tasks like harvesting designed on tending to livestock.

Space exploration robots:
Designed for missions to other plants and celestial bodies.

Search and Rescue Robots:
used in disaster scenarious to locate and assist survivors.

underwater robots:

Explore and perform tasks in marine environments.

Q.(2) Use different types of robotic sensor for specific situation.

Certainly! Let's explore how different types of robotic sensor can be used in a specific situation. autonomous agricultural farming. In modern agriculture, robots and sensors are increasingly employed to improve crop yield, reduce labour costs, and optimize resource usage.

There are some sensors used in this context:

(i) GPS and GNSS (Global Navigation Satellite System):
These sensors provides precise location data, allowing autonomous tractors and drones to navigate fields accurately. Farmers can plan planting, irrigation, and harvesting routs more efficently.

(ii) Cameras (vision sensors):-
Cameras, often mounted on drones or ground-based robots, capture images or videos of crops. Computer vision algorithms analyze these visual for various purposes.

Crop Monitoring: Detecting plant health, identifying diseases, and tracking growth.

Weed Detection: Recognizing and targeting weeds for precision herbicide application.

Crop Counting: Determining the number of plants for yield estimation.

(3) LIDAR (Light Detection and Ranging)

LIDAR sensors use laser pulses to measure distances accurately. In agriculture, LIDAR can be used for

Terrain Mapping: Creating high resolution elevation maps of fields to aid in water flow management.

(4) Ultrasonic Sensors: These sensors use sound waves to measure distances.

(3) perform robot control with stepper motor interfacing :-

Controlling a stepper motor with a microcontroller is a common task in robotics and automation. Stepper motors are used for precise position control in applications like 3D printers, CNC machines, and robotics. To perform robot control with a stepper motor interfacing you'll need the following components.

(i) Stepper Motor
(ii) Microcontroller (e.g. arduino, Raspberry pi)
(iii) stepper Motor Driver (e.g. A4988, DRV8825).
(iv) power suply for the Motor
(v) jumper wires and a breadboard if (necessary).

here's a general outline of the steps to control a stepper motor.

(1) wiring :-
connect the stepper motor to the stepper motor driver. stepper motors have four or more wires which are usually color. coded connect them to the appropriate pins on the driver.

(vii) Install the required libraries:
If you're using an Arduino, you might need to install a stepper motor library like the "AccelStepper" library to simplify moto control. You can install libraries through the Arduino IDE's Library Manager.

(viii) Write the code:
Write the code the control the stepper motor. Below is a basic example using an Arduino and the "AccelStepper" library.

```
#include<Accelstepper.h>

Accelstepper stepper(1,2,3);
void setup() {
stepper.setMaxspeed(1000)
stepper.set Accelnation (500)
void loop(){
stepper.MoveTo(2000);
stepper.runToposition();
delay(1000);
stepper.MoveTo(0);
stepper.runToposition();
delay (1000);
}
```

This code is initializes the stepper motor, sets its maximum speed and acceleration, moves it to a specified position. and then moves it back to the starting position.

(iv) Upload the code.

(v) Test and fine tune.

(4) perform pick and place operation simulation control software.

performing a pick and place operation using simulation control software is a common task in robotics, especially in the context of industrial automation and robotics development simulation software allows you to test and validate your robot's pick and place task's without the need for physical hardware. one popular tool for this purpose is ROS (Robot operating system) and it's simulation environment, Gazebo, Here's a high-level overview of how to perform a pick and place operation using ROS and Gazebo.

Requirements :-

(i) ROS Installation : Ensure you have ROS installed on your computer. The installation process can very depending on your operating system and ROS distribution. you can find installation installation instructions on the ROS website.

ii) Gazebo: Install Gazebo, a widely used robotic simulation environment along with its ROS integration package.

iii) Robot Model:

iv) End-Effector Model:

Steps to perform a pick and place operation.

i) Launch the simulation environment: Use ROS and Gazebo to launch the simulation environment with your robot model and the objects you want to pick and place.

ii) Robot control: Write or use existing ROS node's and controllers to control the robot's movements. ROS provides package like 'MoveIt!' for motion planning and control.

Here's a simplified example using ROS and Gazebo (assuming you have already setup the have already and robot model):

```
# import ROS packages and libraries
import rospy
import moveit_commander
from moveit_commander import
    MoveGroupCommander

rospy.init_node("pick_and_place_demo)
arm = MoveGroupCommander("arm")

pick_pose = [0.5, 0.0, 0.1]
place_pose = [0.5, -0.2, 0.1]

arm.set_pose_target(pick pose)
arm.go()

arm.set_pose_target(place_pose)
arm.go()

rospy.signal_shutdown("pick and place completed.")
```

Q.(5) programme 5-axis Robotic arm to control various motion.

programming a 5-axis robotic arm to control various motions typically Involves using a combination of hardware software, and mathematical Calculations. Below, I'll provide you with a high-level guide on how to approach programming a 5-axis robotic arm. please note that this is complex task and can vary depending on the specific hardware and software you are using.

Requirements:

(i) 5-axis Robotic Arm: Ensure you have a 5-axis robotic arm with associated hardware and controllers.

(ii) programming Environment:

(viii) Inverse Kinematics:

Steps to program a 5-axis Robotic arm:

(i) Hardware setup:
(ii) Software setup.
(iii) Kinematics Model.
(iv) Control algorithms.
(v) User Interface
(vi) Motion planning.
(vii) Testing and Debugging
(viii) Safety Measures.
(ix) Integration.
(x) Documentation.