

मौलाना आजाद राष्ट्रीय प्रौद्योगिकी संस्थान,
भोपाल-462003

यांत्रिक अभियांत्रिकी विभाग



स्नातकोत्तर प्रयोगशाला

सामग्र

क्रम संख्य	प्रयोगशाल
1	रोबोटिक्स प्रयोगशाल
2	उत्पाद विकास प्रयोगशाल
3	नैनो-स्नेहक प्रयोगशाल
4	नैनो-कंपोजिट प्रयोगशाल

1. रोबोटिक्स प्रयोगशाला

Maulana Azad National Institute of Technology Bhopal

Department of Mechanical Engineering



Lab Manual

Robot Programming Lab

M. Tech (Automation and Robotics)

Robot Programming Lab

List of Experiments

1. Introduction to programming in Python, ROS, C, C++, MATLAB, etc.
2. Creation of arrays.
3. Mathematical operations with arrays.
4. Creation of 2D plots.
5. Creation of 3D plots.
6. Curve fitting with polynomials.
7. Programming applications in numerical analysis.
8. Robot programming and path planning.
9. Controlling of robotic manipulator using Arduino/Robot programming.
10. Do-It-Yourself (DIY) experiments (Students should take a real-world issue, decide on a solution, and implement it).

प्रयोगों की सूची

1. पायथन, आरओएस, सी, सी++, मैटलैब आदि में प्रोग्रामिंग का परिचय।
2. सरणियों का निर्माण.
3. सरणियों के साथ गणितीय संक्रियाएँ।
4. 2डी प्लॉट का निर्माण.
5. 3डी प्लॉट का निर्माण.
6. बहुपदों के साथ वक्र फिटिंग।
7. संख्यात्मक विश्लेषण में प्रोग्रामिंग अनुप्रयोग।
8. रोबोट प्रोग्रामिंग और पथ योजना।
9. Arduino/रोबोट प्रोग्रामिंग का उपयोग करके रोबोटिक मैनिपुलेटर का नियंत्रण।
10. डू-इट-योरसेल्फ (DIY) प्रयोग (छात्रों को वास्तविक दुनिया का मुद्दा लेना चाहिए, समाधान पर निर्णय लेना चाहिए और इसे लागू करना चाहिए)।

Experiment No. 1

Title: Introduction to Programming in Python, C, C++, MATLAB, and ROS

1. Introduction and Overview

1.1 Objective

This experiment aims to provide an introductory understanding of programming languages and environments commonly used in robotics: Python, C, C++, MATLAB, and ROS (Robot Operating System). The session will help students familiarize themselves with basic syntax, programming constructs, and applications relevant to robotics.

1.2 Overview of Programming Languages and Environments

- **Python:** A versatile, high-level programming language widely used for scripting, data processing, and AI applications in robotics. Known for its simplicity and large libraries.
 - **C:** A procedural programming language foundational to robotics firmware and embedded systems.
 - **C++:** An extension of C that supports object-oriented programming, essential for performance-critical robotics applications.
 - **MATLAB:** A numerical computing environment and programming platform often used for simulations and algorithm prototyping in robotics.
 - **ROS:** An open-source framework for writing robot software, providing tools and libraries for building robotic systems.
-

2. Software Environment

- **Python:** Python 3.8 or later; IDE: PyCharm or VS Code.
- **C:** GCC or MinGW Compiler; IDE: Code::Blocks or VS Code.
- **C++:** GCC/G++ Compiler; IDE: Code::Blocks, VS Code, or CLion.
- **MATLAB:** MATLAB R2022b or later.
- **ROS:** ROS 2 Humble Hawksbill; Ubuntu 22.04.

System Requirements:

- **Operating System:** Windows 10, Linux (Ubuntu), or macOS.
 - **Processor:** Intel Core i5 or equivalent.
 - **RAM:** Minimum 8 GB.
 - **Storage:** At least 50 GB of free space.
-

3. Methodologies to Perform the Experiment

3.1 Setup and Initialization

1. **Python:**

- Install Python from the official website.
- Set up an IDE (PyCharm or VS Code).
- Verify installation using `python --version` in the terminal.

2. **C and C++:**

- Install GCC or MinGW Compiler.
- Set up the IDE (e.g., Code::Blocks) and link the compiler.
- Verify installation using `gcc --version` in the terminal.

3. **MATLAB:**

- Install MATLAB and activate the license.
- Launch MATLAB and ensure all toolboxes are available.

4. **ROS:**

- Install ROS 2 Humble on Ubuntu.
- Source the ROS environment using `source /opt/ros/humble/setup.bash`.
- Verify the installation using `ros2 --version`.

3.2 Writing and Executing Basic Programs

1. **Python:**

- Write a basic "Hello, World!" program and execute it.
- Example:

```
print("Hello, World!")
```

2. **C:**

- Write a "Hello, World!" program and compile it using GCC.
- Example:

```
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

3. **C++:**

- Write a "Hello, World!" program and compile it using G++.

- Example:

```
#include <iostream>

using namespace std;

int main() {

    cout << "Hello, World!" << endl;

    return 0;
}
```

4. MATLAB:

- Write a script in the Command Window or Editor.
- Example:

```
disp('Hello, World!');
```

5. ROS:

```
import rclpy

from rclpy.node import Node

class HelloWorldNode(Node):

    def __init__(self):
        super().__init__('hello_world_node')
        self.get_logger().info('Hello, ROS 2 World!')

def main(args=None):

    rclpy.init(args=args)

    node = HelloWorldNode()

    rclpy.spin(node)

    rclpy.shutdown()

if __name__ == '__main__':
    main()
```

3.3 Running and Debugging

- Compile and execute programs in respective environments.
- Debug any errors using built-in debugging tools.

-
4. Expected Outcomes
 5. Understand the basic syntax and structure of Python, C, C++, MATLAB, and ROS programs.
 6. Execute and debug basic programs in each environment.
 7. Develop a foundational understanding of programming tools used in robotics.
-

5. Common Errors and Troubleshooting

6. **Python:**

- Error: "Module not found."
- Solution: Install the required module using `pip install module_name`.

7. **C/C++:**

- Error: "gcc: command not found."
- Solution: Ensure the compiler is installed and added to the PATH.

8. **MATLAB:**

- Error: "Function not recognized."
- Solution: Ensure the required toolbox is installed.

9. **ROS:**

- Error: "ROS environment not sourced."
 - Solution: Source the environment using `source /opt/ros/humble/setup.bash`.
-

6. Thought-Provoking Questions

7. What are the differences between Python and C++ in terms of usage in robotics?
 8. Why is MATLAB widely used for robotics simulations?
 9. How does ROS simplify the process of building robotic applications?
 10. What are the challenges in debugging programs across different environments?
 11. Which language would you choose for real-time robotic applications and why?
-

7. Applications of Programming in Robotics

8. **Python:** Data processing, AI integration, and scripting.
9. **C:** Embedded systems and firmware.
10. **C++:** Performance-critical algorithms and control systems.

11. **MATLAB:** Simulation, prototyping, and numerical analysis.

12. **ROS:** Developing modular robotic systems.

This concludes the experiment on the introduction to programming in Python, C, C++, MATLAB, and ROS.

प्रयोग संख्या 1

शीर्षक: Python, C, C++, MATLAB और ROS में प्रोग्रामिंग का परिचय

1. परिचय और अवलोकन

1.1 उद्देश्य:

यह प्रयोग रोबोटिक्स में उपयोग की जाने वाली सामान्य प्रोग्रामिंग भाषाओं और पर्यावरणों जैसे Python, C, C++, MATLAB और ROS (Robot Operating System) का प्रारंभिक परिचय प्रदान करता है। सत्र छात्रों को मूल सिंटैक्स, प्रोग्रामिंग संरचनाओं, और रोबोटिक्स से संबंधित अनुप्रयोगों से परिचित कराएगा।

1.2 प्रोग्रामिंग भाषाओं और पर्यावरणों का अवलोकन:

- **Python:** एक बहुउद्देश्यीय, उच्च-स्तरीय प्रोग्रामिंग भाषा जो स्क्रिप्टिंग, डेटा प्रोसेसिंग, और रोबोटिक्स में AI अनुप्रयोगों के लिए उपयोग की जाती है। इसकी सादगी और विशाल लाइब्रेरीज़ के लिए जानी जाती है।
 - **C:** एक प्रक्रियात्मक प्रोग्रामिंग भाषा, जो रोबोटिक्स फर्मवेयर और एंबेडेड सिस्टम्स के लिए आधारभूत है।
 - **C++:** C का विस्तार, जो ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग का समर्थन करता है, और प्रदर्शन-आधारित रोबोटिक्स अनुप्रयोगों के लिए आवश्यक है।
 - **MATLAB:** एक संख्यात्मक कंप्यूटिंग पर्यावरण और प्रोग्रामिंग प्लेटफॉर्म, जो रोबोटिक्स में सिमुलेशन और एल्गोरिदम प्रोटोटाइपिंग के लिए उपयोगी है।
 - **ROS:** रोबोटिक सॉफ्टवेयर लिखने के लिए एक ओपन-सोर्स फ्रेमवर्क, जो रोबोटिक सिस्टम्स के निर्माण के लिए टूल्स और लाइब्रेरी प्रदान करता है।
-

2. सॉफ्टवेयर पर्यावरण:

- **Python:** Python 3.8 या बाद का संस्करण; IDE: PyCharm या VS Code।
- **C:** GCC या MinGW कंपाइलर; IDE: Code::Blocks या VS Code।
- **C++:** GCC/G++ कंपाइलर; IDE: Code::Blocks, VS Code, या CLion।
- **MATLAB:** MATLAB R2022b या बाद का संस्करण।

- **ROS:** ROS 2 Humble Hawksbill; Ubuntu 22.04।

सिस्टम आवश्यकताएँ:

- **ऑपरेटिंग सिस्टम:** Windows 10, Linux (Ubuntu), या macOS।
 - **प्रोसेसर:** Intel Core i5 या समकक्ष।
 - **RAM:** न्यूनतम 8 GB।
 - **स्टोरेज:** कम से कम 50 GB खाली स्थान।
-

3. प्रयोग को करने की विधियाँ

3.1 सेटअप और प्रारंभिककरण:

1. Python:

- आधिकारिक वेबसाइट से Python इंस्टॉल करें।
- IDE (PyCharm या VS Code) सेट करें।
- टर्मिनल में `python --version` का उपयोग करके इंस्टॉलेशन सत्यापित करें।

2. C और C++:

- GCC या MinGW कंपाइलर इंस्टॉल करें।
- IDE (जैसे, Code::Blocks) सेट करें और कंपाइलर को लिंक करें।
- टर्मिनल में `gcc --version` का उपयोग करके इंस्टॉलेशन सत्यापित करें।

3. MATLAB:

- MATLAB इंस्टॉल करें और लाइसेंस सक्रिय करें।
- MATLAB लॉन्च करें और सुनिश्चित करें कि सभी टूलबॉक्स उपलब्ध हैं।

4. ROS:

- Ubuntu पर ROS 2 Humble इंस्टॉल करें।
- `source /opt/ros/humble/setup.bash` का उपयोग करके ROS पर्यावरण स्रोत करें।
- `ros2 --version` का उपयोग करके इंस्टॉलेशन सत्यापित करें।

3.2 बेसिक प्रोग्राम लिखना और निष्पादित करना:

1. Python:

- एक "Hello, World!" प्रोग्राम लिखें और निष्पादित करें।

```
print("Hello, World!")
```

2. **C:**

```
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

3. **C++:**

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello, World!" << endl;
    return 0;
}
```

4. **MATLAB:**

```
disp('Hello, World!');
```

5. **ROS (Python):**

```
import rclpy
from rclpy.node import Node

class HelloWorldNode(Node):
    def __init__(self):
        super().__init__('hello_world_node')
        self.get_logger().info('Hello, ROS 2 World!')

def main(args=None):
    rclpy.init(args=args)
    node = HelloWorldNode()
    rclpy.spin(node)
    rclpy.shutdown()

if __name__ == '__main__':
    main()
```

3.3 रन और डीबगिंग:

- संबंधित पर्यावरण में प्रोग्राम संकलित करें और निष्पादित करें।

- निर्मित डीबगिंग टूल्स का उपयोग करके त्रुटियों को हल करें।
-

4. अपेक्षित परिणाम:

- Python, C, C++, MATLAB और ROS प्रोग्राम्स की बुनियादी संरचना और सिंटैक्स को समझें।
 - प्रत्येक पर्यावरण में प्रोग्राम्स को निष्पादित और डीबग करें।
 - रोबोटिक्स में उपयोग किए जाने वाले प्रोग्रामिंग टूल्स का बुनियादी ज्ञान विकसित करें।
-

5. सामान्य त्रुटियाँ और समाधान:

1. Python:

- त्रुटि: "Module not found."
- समाधान: आवश्यक मॉड्यूल को `pip install module_name` का उपयोग करके इंस्टॉल करें।

2. C/C++:

- त्रुटि: "gcc: command not found."
- समाधान: सुनिश्चित करें कि कंपाइलर इंस्टॉल है और PATH में जोड़ा गया है।

3. MATLAB:

- त्रुटि: "Function not recognized."
- समाधान: सुनिश्चित करें कि आवश्यक टूलबॉक्स इंस्टॉल हैं।

4. ROS:

- त्रुटि: "ROS environment not sourced."
 - समाधान: `source /opt/ros/humble/setup.bash` का उपयोग करके पर्यावरण स्रोत करें।
-

6. विचारशील प्रश्न:

1. रोबोटिक्स में उपयोग के लिए Python और C++ के बीच क्या अंतर हैं?
 2. MATLAB रोबोटिक्स सिमुलेशन के लिए व्यापक रूप से क्यों उपयोग किया जाता है?
 3. ROS रोबोटिक अनुप्रयोग बनाने की प्रक्रिया को कैसे सरल करता है?
 4. विभिन्न पर्यावरणों में प्रोग्राम्स को डीबग करने में क्या चुनौतियाँ हैं?
 5. आप वास्तविक समय रोबोटिक अनुप्रयोगों के लिए कौन सी भाषा चुनेंगे और क्यों?
-

7. रोबोटिक्स में प्रोग्रामिंग के अनुप्रयोग:

- **Python:** डेटा प्रोसेसिंग, AI एकीकरण, और स्क्रिप्टिंग।
- **C:** एंबेडेड सिस्टम और फर्मवेयर।
- **C++:** प्रदर्शन-आधारित एल्गोरिदम और नियंत्रण प्रणाली।
- **MATLAB:** सिमुलेशन, प्रोटोटाइपिंग, और संख्यात्मक विश्लेषण।
- **ROS:** मॉड्यूलर रोबोटिक सिस्टम्स का विकास।

यह प्रयोग Python, C, C++, MATLAB और ROS में प्रोग्रामिंग के परिचय पर आधारित है।

Experiment No. 2

Title: Creation of Arrays in MATLAB and Python

1. Introduction and Overview

1.1 Objective

This experiment aims to introduce students to the creation and manipulation of arrays using MATLAB and Python. Students will explore how to initialize, access, and perform basic operations on arrays. This hands-on session will help students understand the fundamental differences and similarities between array handling in MATLAB and Python.

1.2 Overview of MATLAB and Python Arrays

- **MATLAB:** MATLAB's primary data structure is the matrix, and arrays are treated as special cases of matrices. MATLAB is widely used for mathematical modeling, simulations, and scientific computations.
- **Python:** Python provides powerful libraries such as NumPy for handling arrays. Arrays in Python are versatile and are frequently used for numerical computations and data manipulation.

2. Software Environment

2.1 Software Used

- **MATLAB R2023a or later**
- **Python 3.10 or later with NumPy library installed**

2.2 System Requirements

- **Operating System:** Windows 10 or later, macOS, or Linux
 - **Processor:** Intel Core i5 or equivalent
 - **RAM:** Minimum 8 GB
 - **Python Libraries:** NumPy, Matplotlib (optional for visualizations)
-

3. Methodologies to Perform the Experiment

3.1 Array Creation in MATLAB

1. Initialization of Arrays:

- Create arrays using square brackets []:

```
A = [1, 2, 3, 4]; % Row vector
```

```
B = [1; 2; 3; 4]; % Column vector
```

```
C = [1, 2; 3, 4]; % 2x2 Matrix
```

- Use built-in functions:

```
D = zeros(3, 3); % 3x3 matrix of zeros
```

```
E = ones(2, 4); % 2x4 matrix of ones
```

```
F = linspace(0, 10, 5); % Linearly spaced vector
```

2. Accessing Array Elements:

- Indexing starts at 1 in MATLAB:

```
element = A(2); % Access 2nd element of array A
```

```
row = C(1, :); % Access the first row of matrix C
```

3. Operations on Arrays:

- Element-wise addition, subtraction, and multiplication:

```
G = A + B'; % Add row and column vectors
```

```
H = C .* 2; % Element-wise multiplication
```

3.2 Array Creation in Python

1. Initialization of Arrays:

Using NumPy:

```
import numpy as np
```

```
A = np.array([1, 2, 3, 4]) # 1D array
```

```
B = np.array([[1, 2], [3, 4]]) # 2D array
```

```
C = np.zeros((3, 3)) # 3x3 matrix of zeros
```

```
D = np.ones((2, 4)) # 2x4 matrix of ones
```

```
E = np.linspace(0, 10, 5) # Linearly spaced vector
```

2. Accessing Array Elements:

- Indexing starts at 0 in Python:

```
element = A[1] # Access 2nd element of array A
```

```
row = B[0, :] # Access the first row of matrix B
```

3. Operations on Arrays:

- Element-wise addition, subtraction, and multiplication:

```
F = A + np.array([10, 20, 30, 40]) # Add arrays
```

```
G = B * 2 # Element-wise multiplication
```

4. Expected Outcomes

1. Students will be able to create and manipulate arrays in both MATLAB and Python.
2. Understanding the syntax and differences in handling arrays between MATLAB and Python.
3. Familiarity with indexing, initialization, and basic array operations in both environments.

5. Common Errors and Troubleshooting

1. MATLAB:

- **Error:** "Index exceeds matrix dimensions."
 - **Solution:** Ensure the index value is within the array size.
- **Error:** "Matrix dimensions must agree."
 - **Solution:** Use element-wise operators (e.g., `.*`, `./`) for array operations.

2. Python:

- **Error:** "IndexError: index out of bounds."
 - **Solution:** Ensure the index starts from 0 and is within the array range.
- **Error:** "ValueError: operands could not be broadcast together."
 - **Solution:** Check the shapes of arrays before performing operations.

6. Thought-Provoking Questions

1. How do the indexing conventions in MATLAB and Python differ?
2. What are the advantages of using NumPy for array manipulations in Python?

3. How can you create a 3D array in MATLAB and Python?
4. Discuss the use of linspace in MATLAB and its equivalent in Python.
5. What are the benefits of using built-in functions for initializing arrays?

7. Applications of Array Operations

1. Data analysis and visualization in scientific research.
2. Signal processing in engineering.
3. Simulation of mathematical models.
4. Machine learning and artificial intelligence.

This concludes the experiment on the creation of arrays in MATLAB and Python.

प्रयोग संख्या 2

शीर्षक: MATLAB और Python में Arrays बनाना

1. परिचय और अवलोकन

1.1 उद्देश्य:

यह प्रयोग छात्रों को MATLAB और Python का उपयोग करके arrays बनाने और उनमें हेरफेर करने की प्रक्रिया से परिचित कराता है। छात्र यह सीखेंगे कि arrays को कैसे प्रारंभ करें, उन तक कैसे पहुँचें, और उन पर बुनियादी संचालन कैसे करें। यह व्यावहारिक सत्र MATLAB और Python में array संचालन की समानताओं और भिन्नताओं को समझने में मदद करेगा।

1.2 MATLAB और Python Arrays का अवलोकन:

- **MATLAB:** MATLAB का प्राथमिक डेटा स्ट्रक्चर मैट्रिक्स है, और arrays को मैट्रिक्स के विशेष मामलों के रूप में माना जाता है। MATLAB गणितीय मॉडलिंग, सिमुलेशन और वैज्ञानिक गणनाओं के लिए व्यापक रूप से उपयोग किया जाता है।
- **Python:** Python में NumPy जैसी शक्तिशाली लाइब्रेरीज़ arrays को संभालने के लिए उपयोग की जाती हैं। Python में arrays बहुमुखी हैं और अक्सर संख्यात्मक गणना और डेटा प्रबंधन के लिए उपयोग किए जाते हैं।

2. सॉफ्टवेयर वातावरण

2.1 उपयोग किया गया सॉफ्टवेयर:

- MATLAB R2023a या बाद का संस्करण
- Python 3.10 या बाद का संस्करण, जिसमें NumPy लाइब्रेरी स्थापित हो

2.2 सिस्टम आवश्यकताएँ:

- ऑपरेटिंग सिस्टम: Windows 10 या बाद का, macOS, या Linux
 - प्रोसेसर: Intel Core i5 या समकक्ष
 - RAM: न्यूनतम 8 GB
 - Python लाइब्रेरीज़: NumPy, Matplotlib (वैकल्पिक, विज़ुअलाइज़ेशन के लिए)
-

3. प्रयोग की कार्यप्रणाली

3.1 MATLAB में Arrays बनाना

1. Arrays प्रारंभ करना:

वर्ग कोष्ठकों का उपयोग करके Arrays बनाना:

```
A = [1, 2, 3, 4]; % पंक्ति वेक्टर
```

```
B = [1; 2; 3; 4]; % स्तंभ वेक्टर
```

```
C = [1, 2; 3, 4]; % 2x2 मैट्रिक्स
```

इन-बिल्ट फ़ंक्शन का उपयोग करना:

```
D = zeros(3, 3); % 3x3 शून्य मैट्रिक्स
```

```
E = ones(2, 4); % 2x4 मैट्रिक्स
```

```
F = linspace(0, 10, 5); % रेखीय रूप से वितरित वेक्टर
```

2. Array तत्वों तक पहुँच:

- MATLAB में इंडेक्सिंग 1 से शुरू होती है:

```
element = A(2); % Array A के दूसरे तत्व तक पहुँच
```

```
row = C(1, :); % मैट्रिक्स C की पहली पंक्ति तक पहुँच
```

3. Arrays पर संचालन:

- तत्व-स्तरीय जोड़, घटाव, और गुणा:

```
G = A + B'; % पंक्ति और स्तंभ वेक्टर का जोड़
```

```
H = C .* 2; % तत्व-स्तरीय गुणा
```

3.2 Python में Arrays बनाना

1. Arrays प्रारंभ करना:

NumPy का उपयोग करना:

```
import numpy as np
```

```
A = np.array([1, 2, 3, 4]) # 1D array
```

```
B = np.array([[1, 2], [3, 4]]) # 2D array
```

```
C = np.zeros((3, 3)) # 3x3 शून्य मैट्रिक्स
```

```
D = np.ones((2, 4)) # 2x4 मैट्रिक्स
```

```
E = np.linspace(0, 10, 5) # रेखीय रूप से वितरित वेक्टर
```

2. Array तत्वों तक पहुँच:

- Python में इंडेक्सिंग 0 से शुरू होती है:

```
element = A[1] # Array A के दूसरे तत्व तक पहुँच
```

```
row = B[0, :] # मैट्रिक्स B की पहली पंक्ति तक पहुँच
```

3. Arrays पर संचालन:

- तत्व-स्तरीय जोड़, घटाव, और गुणा:

```
F = A + np.array([10, 20, 30, 40]) # Arrays का जोड़
```

```
G = B * 2 # तत्व-स्तरीय गुणा
```

4. अपेक्षित परिणाम

1. छात्र MATLAB और Python में arrays बनाने और उनमें हेरफेर करने में सक्षम होंगे।
2. MATLAB और Python में arrays संभालने की प्रक्रिया में अंतर और समानताओं को समझना।
3. दोनों वातावरणों में इंडेक्सिंग, प्रारंभिककरण और बुनियादी संचालन में परिचितता।

5. सामान्य त्रुटियाँ और समाधान

1. MATLAB:

- त्रुटि: "Index exceeds matrix dimensions."

समाधान: सुनिश्चित करें कि इंडेक्स मान array के आकार के भीतर है।

- त्रुटि: "Matrix dimensions must agree."

समाधान: तत्व-स्तरीय संचालन के लिए ऑपरेटर्स (e.g., `.*`, `./`) का उपयोग करें।

2. Python:

- त्रुटि: "IndexError: index out of bounds."
समाधान: सुनिश्चित करें कि इंडेक्स 0 से शुरू होता है और array रेंज के भीतर है।
- त्रुटि: "ValueError: operands could not be broadcast together."
समाधान: संचालन से पहले arrays के आकार की जाँच करें।

6. विचारोत्तेजक प्रश्न

1. MATLAB और Python में इंडेक्सिंग के नियम कैसे अलग हैं?
2. Python में arrays के लिए NumPy का उपयोग करने के क्या लाभ हैं?
3. MATLAB और Python में 3D array कैसे बनाया जा सकता है?
4. MATLAB में linspace और Python में इसके समकक्ष पर चर्चा करें।
5. Arrays को प्रारंभ करने के लिए इन-बिल्ट फ़ंक्शंस का उपयोग करने के क्या लाभ हैं?

7. Arrays संचालन के अनुप्रयोग

1. वैज्ञानिक अनुसंधान में डेटा विश्लेषण और विज़ुअलाइज़ेशन।
2. इंजीनियरिंग में सिग्नल प्रोसेसिंग।
3. गणितीय मॉडलों का सिमुलेशन।
4. मशीन लर्निंग और आर्टिफिशियल इंटेलिजेंस।

इस प्रकार MATLAB और Python में arrays बनाने के प्रयोग का समापन होता है।

Experiment 3

Title: Mathematical Operations with Arrays

1. Introduction and Overview

1.1 Objective

This experiment aims to familiarize students with performing various mathematical operations on arrays using MATLAB and Python. Students will learn how to perform element-wise and matrix-based arithmetic operations, apply mathematical functions, and manipulate arrays for computational purposes.

1.2 Overview of Arrays

Arrays are fundamental data structures used for storing and performing operations on a collection of numerical data. In MATLAB and Python, arrays are versatile and can be manipulated easily to perform mathematical operations such as addition, subtraction, multiplication, division, and complex mathematical computations.

2. Software Environment

MATLAB Environment:

- **Software Used:** MATLAB R2023a (or equivalent version)
- **System Requirements:**
 - Operating System: Windows/Mac/Linux
 - Processor: Intel Core i3 or higher
 - RAM: Minimum 4 GB

Python Environment:

- **Software Used:** Python 3.9 (or newer)
 - **Libraries Required:** NumPy
 - **System Requirements:**
 - Operating System: Windows/Mac/Linux
 - Python IDE: Jupyter Notebook, VS Code, or PyCharm
-

3. Methodologies to Perform the Experiment

3.1 MATLAB Implementation:

1. Creating Arrays:

% Define arrays

A = [1, 2, 3; 4, 5, 6; 7, 8, 9];

B = [9, 8, 7; 6, 5, 4; 3, 2, 1];

2. Element-wise Operations:

% Addition

C = A + B;

% Subtraction

D = A - B;

% Multiplication

E = A .* B; % Element-wise

% Division

F = A ./ B; % Element-wise

3. Matrix Operations:

% Matrix multiplication

G = A * B; % Ensure dimensions match

% Transpose

H = A';

4. Mathematical Functions:

% Applying functions

sqrt_A = sqrt(A); % Square root of elements

exp_A = exp(A); % Exponential of elements

sin_A = sin(A); % Sine of elements

5. Displaying Results:

disp('Result of Addition:'), disp(C);

disp('Result of Element-wise Multiplication:'), disp(E);

3.2 Python Implementation:

1. Creating Arrays:

import numpy as np

Define arrays

A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

B = np.array([[9, 8, 7], [6, 5, 4], [3, 2, 1]])

2. Element-wise Operations:

Addition

C = A + B

Subtraction

D = A - B

Multiplication

E = A * B

Division

F = A / B

3. Matrix Operations:

Matrix multiplication

```
G = np.dot(A, B)
```

Transpose

```
H = A.T
```

4. Mathematical Functions:

Applying functions

```
sqrt_A = np.sqrt(A) # Square root of elements
```

```
exp_A = np.exp(A) # Exponential of elements
```

```
sin_A = np.sin(A) # Sine of elements
```

5. Displaying Results:

```
print("Result of Addition:", C)
```

```
print("Result of Element-wise Multiplication:", E)
```

4. Expected Outcomes

1. Ability to perform element-wise arithmetic operations on arrays.
2. Understanding of matrix multiplication and transposition.
3. Application of mathematical functions to arrays.
4. Proficiency in displaying and interpreting results in MATLAB and Python.

5. Common Errors and Troubleshooting

1. **Error:** Mismatched array dimensions for matrix multiplication.
 - **Solution:** Ensure the number of columns in the first array equals the number of rows in the second array.
2. **Error:** Division by zero in element-wise operations.
 - **Solution:** Avoid zero values in the denominator array.
3. **Error:** Undefined function or variable in MATLAB.
 - **Solution:** Verify that the function or variable is defined in the workspace.

6. Thought-Provoking Questions

1. What is the difference between element-wise and matrix-based operations?

2. How can you handle arrays of different sizes for mathematical operations?
3. What are some real-world applications of matrix multiplication in robotics?
4. Why is the transpose operation important in matrix computations?
5. How can mathematical functions like sine and exponential be used in robotic control algorithms?

7. Applications of Array Operations

1. **Robotic Control:** Arrays are used to compute kinematics and dynamics of robotic systems.
2. **Data Analysis:** Manipulating large datasets for statistical computations.
3. **Image Processing:** Arrays represent pixel intensities for image transformations.
4. **Machine Learning:** Arrays are used for training models and handling datasets.

This concludes **Experiment 3: Mathematical Operations with Arrays**.

प्रयोग 3

शीर्षक: एरेज़ के साथ गणितीय संचालन

1. परिचय और अवलोकन

1.1 उद्देश्य

यह प्रयोग छात्रों को MATLAB और Python का उपयोग करके एरेज़ पर विभिन्न गणितीय संचालन करने से परिचित कराने का लक्ष्य रखता है। छात्र तत्व-स्तरीय और मैट्रिक्स-आधारित अंकगणितीय संचालन, गणितीय फ़ंक्शनों को लागू करना और कम्प्यूटेशनल उद्देश्यों के लिए एरेज़ को संशोधित करना सीखेंगे।

1.2 एरेज़ का अवलोकन

एरेज़ मौलिक डेटा संरचनाएँ हैं जिनका उपयोग संख्यात्मक डेटा के संग्रह को संग्रहीत करने और उन पर संचालन करने के लिए किया जाता है। MATLAB और Python में, एरेज़ बहुमुखी होते हैं और इन्हें जोड़, घटाव, गुणा, विभाजन और जटिल गणितीय गणनाएँ करने के लिए आसानी से संशोधित किया जा सकता है।

2. सॉफ्टवेयर पर्यावरण

MATLAB पर्यावरण:

- उपयोग किया गया सॉफ्टवेयर: MATLAB R2023a (या समकक्ष संस्करण)
- सिस्टम आवश्यकताएँ:
 - o ऑपरेटिंग सिस्टम: विंडोज/मैक/लिनक्स

- o प्रोसेसर: इंटेल कोर i3 या उच्चतर
- o रैम: न्यूनतम 4 GB

Python पर्यावरण:

- उपयोग किया गया सॉफ्टवेयर: Python 3.9 (या नया संस्करण)
 - आवश्यक लाइब्रेरीज़: NumPy
 - सिस्टम आवश्यकताएँ:
 - o ऑपरेटिंग सिस्टम: विंडोज/मैक/लिनक्स
 - o Python IDE: Jupyter Notebook, VS Code, या PyCharm
-

3. प्रयोग करने की विधियाँ

3.1 MATLAB कार्यान्वयन:

1. एरेज़ बनाना:

% एरेज़ को परिभाषित करें

```
A = [1, 2, 3; 4, 5, 6; 7, 8, 9];
```

```
B = [9, 8, 7; 6, 5, 4; 3, 2, 1];
```

2. तत्व-स्तरीय संचालन:

% जोड़

```
C = A + B;
```

% घटाव

```
D = A - B;
```

% गुणा

```
E = A .* B; % तत्व-स्तरीय
```

% विभाजन

```
F = A ./ B; % तत्व-स्तरीय
```

3. मैट्रिक्स संचालन:

% मैट्रिक्स गुणा

```
G = A * B; % सुनिश्चित करें कि आयाम मेल खाते हैं
```

% ट्रांसपोज़

```
H = A';
```

4. गणितीय फ़ंक्शन:

% फ़ंक्शन लागू करना

```
sqrt_A = sqrt(A); % तत्वों का वर्गमूल
```

```
exp_A = exp(A); % तत्वों का घातांक
```

```
sin_A = sin(A); % तत्वों का साइन
```

5. परिणाम दिखाना:

```
disp('जोड़ का परिणाम:'), disp(C);
```

```
disp('तत्व-स्तरीय गुणा का परिणाम:'), disp(E);
```

3.2 Python कार्यान्वयन:

1. एरेज़ बनाना:

```
import numpy as np
```

```
# एरेज़ को परिभाषित करें
```

```
A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
```

```
B = np.array([[9, 8, 7], [6, 5, 4], [3, 2, 1]])
```

2. तत्व-स्तरीय संचालन:

```
# जोड़
```

```
C = A + B
```

```
# घटाव
```

```
D = A - B
```

```
# गुणा
```

```
E = A * B
```

```
# विभाजन
```

```
F = A / B
```

3. मैट्रिक्स संचालन:

```
# मैट्रिक्स गुणा
```

```
G = np.dot(A, B)
```

```
# ट्रांसपोज़
```

```
H = A.T
```

4. गणितीय फ़ंक्शन:

फ़ंक्शन लागू करना

```
sqrt_A = np.sqrt(A) # तत्वों का वर्गमूल
```

```
exp_A = np.exp(A) # तत्वों का घातांक
```

```
sin_A = np.sin(A) # तत्वों का साइन
```

5. परिणाम दिखाना:

```
print("जोड़ का परिणाम:", C)
```

```
print("तत्व-स्तरीय गुणा का परिणाम:", E)
```

4. अपेक्षित परिणाम

1. तत्व-स्तरीय अंकगणितीय संचालन करने की क्षमता।
2. मैट्रिक्स गुणा और ट्रांसपोज़ समझ।
3. एरेज़ पर गणितीय फ़ंक्शन लागू करना।
4. MATLAB और Python में परिणाम प्रदर्शित करने और उनकी व्याख्या करने में प्रवीणता।

5. सामान्य त्रुटियाँ और समाधान

1. त्रुटि: मैट्रिक्स गुणा के लिए एरेज़ के आयाम मेल नहीं खाते।
o समाधान: सुनिश्चित करें कि पहले एरे का कॉलम और दूसरे एरे की पंक्ति की संख्या समान हो।
2. त्रुटि: तत्व-स्तरीय संचालन में शून्य द्वारा विभाजन।
o समाधान: हर स्थिति में भाजक एरे में शून्य मानों से बचें।
3. त्रुटि: MATLAB में अपरिभाषित फ़ंक्शन या चर।
o समाधान: सुनिश्चित करें कि फ़ंक्शन या चर कार्यक्षेत्र में परिभाषित हैं।

6. विचारोत्तेजक प्रश्न

1. तत्व-स्तरीय और मैट्रिक्स-आधारित संचालन में क्या अंतर है?
2. विभिन्न आकारों के एरेज़ के साथ गणितीय संचालन कैसे किया जा सकता है?
3. रोबोटिक्स में मैट्रिक्स गुणा के कुछ वास्तविक जीवन अनुप्रयोग क्या हैं?
4. मैट्रिक्स गणनाओं में ट्रांसपोज़ संचालन क्यों महत्वपूर्ण है?
5. रोबोटिक नियंत्रण एल्गोरिदम में साइन और घातांक जैसे गणितीय फ़ंक्शन कैसे उपयोगी हो सकते हैं?

7. एरेज़ संचालन के अनुप्रयोग

- रोबोटिक नियंत्रण:** एरेज़ का उपयोग रोबोटिक सिस्टम की काइनेमेटिक्स और डायनेमिक्स की गणना के लिए किया जाता है।
 - डेटा विश्लेषण:** सांख्यिकीय गणनाओं के लिए बड़े डेटा सेट का हेरफेर।
 - इमेज प्रोसेसिंग:** छवि परिवर्तन के लिए पिक्सल की तीव्रता को एरेज़ के रूप में दर्शाना।
 - मशीन लर्निंग:** मॉडल प्रशिक्षण और डेटा सेट प्रबंधन के लिए एरेज़ का उपयोग।
-

यह प्रयोग 3: एरेज़ के साथ गणितीय संचालन समाप्त करता है।

Experiment No. 4

Title: Creation of 2D Plots

1. Introduction and Overview

1.1 Objective

This experiment introduces students to creating 2D plots using MATLAB and Python. Students will learn how to visualize data through line plots, scatter plots, bar charts, and other common 2D plotting techniques. The session provides hands-on practice in customizing plots to enhance readability and presentation.

1.2 Overview of 2D Plotting

Visualization is a critical aspect of data analysis. 2D plots allow engineers, scientists, and designers to represent data relationships graphically, making patterns and trends easier to understand. MATLAB and Python offer robust libraries—such as matplotlib in Python—to generate high-quality 2D plots.

2. Software Environment

MATLAB Environment:

- Software Used:** MATLAB R2023a (or later)
- System Requirements:**
 - Operating System: Windows, macOS, or Linux
 - Processor: Intel Core i5 or equivalent
 - RAM: Minimum 4 GB

Python Environment:

- **Software Used:** Python 3.8 or later, with matplotlib library
 - **System Requirements:**
 - Operating System: Windows, macOS, or Linux
 - Processor: Intel Core i5 or equivalent
 - RAM: Minimum 4 GB
-

3. Methodologies to Perform the Experiment

3.1 Plotting in MATLAB

1. Setting Up:

- Open MATLAB.
- Create a new script file for code implementation.

2. Creating a Line Plot:

```
x = 0:0.1:10; % Generate data for x-axis
y = sin(x); % Compute sine of x
plot(x, y, 'r', 'LineWidth', 2); % Create a red line plot with thicker lines
title('Sine Wave');
xlabel('x-axis');
ylabel('y-axis');
grid on; % Turn on the grid
```

3. Scatter Plot:

```
x = rand(1, 20); % Random x values
y = rand(1, 20); % Random y values
scatter(x, y, 'filled'); % Create a scatter plot with filled markers
title('Scatter Plot');
xlabel('Random X Values');
ylabel('Random Y Values');
```

4. Bar Chart:

```
categories = {'A', 'B', 'C', 'D'}; % Define categories
values = [3, 7, 8, 5]; % Define values
bar(values);
```

```
set(gca, 'xticklabel', categories); % Label the x-axis  
title('Bar Chart');  
ylabel('Values');
```

3.2 Plotting in Python (Using Matplotlib)

1. Setting Up:

- Install matplotlib library if not already installed:
- `pip install matplotlib`
- Open your preferred Python IDE (e.g., Jupyter Notebook, VS Code).

2. Creating a Line Plot:

```
import numpy as np  
import matplotlib.pyplot as plt  
  
x = np.linspace(0, 10, 100) # Generate data for x-axis  
y = np.sin(x) # Compute sine of x  
  
plt.plot(x, y, color='red', linewidth=2) # Create a red line plot  
plt.title('Sine Wave')  
plt.xlabel('x-axis')  
plt.ylabel('y-axis')  
plt.grid(True) # Turn on the grid  
plt.show()
```

3. Scatter Plot:

```
x = np.random.rand(20) # Random x values  
y = np.random.rand(20) # Random y values  
  
plt.scatter(x, y, color='blue', s=50) # Create a scatter plot with blue markers  
plt.title('Scatter Plot')  
plt.xlabel('Random X Values')  
plt.ylabel('Random Y Values')  
plt.show()
```

4. Bar Chart:

```
categories = ['A', 'B', 'C', 'D'] # Define categories  
values = [3, 7, 8, 5] # Define values  
plt.bar(categories, values, color='green')
```

```
plt.title('Bar Chart')  
plt.ylabel('Values')  
plt.show()
```

4. Expected Outcomes

1. Ability to create various 2D plots in MATLAB and Python.
 2. Understanding how to customize plots using titles, labels, legends, and grids.
 3. Familiarity with different plotting techniques like line plots, scatter plots, and bar charts.
-

5. Common Errors and Troubleshooting

1. **Error:** Data size mismatch between x and y.
 - **Solution:** Ensure x and y arrays have the same length.
 2. **Error:** Undefined function or variable in MATLAB.
 - **Solution:** Check for typos in variable or function names.
 3. **Error:** Matplotlib figure does not display.
 - **Solution:** Use plt.show() to display the figure.
-

6. Thought-Provoking Questions

1. How can you enhance the readability of a plot for presentation purposes?
 2. What are the key differences in plotting capabilities between MATLAB and Python?
 3. How would you plot multiple lines in the same graph?
 4. Why is it important to label axes and provide titles for plots?
-

7. Applications of 2D Plotting

1. **Data Analysis:** Visualizing trends and patterns in datasets.
 2. **Engineering Design:** Representing simulation results graphically.
 3. **Scientific Research:** Displaying experimental data for analysis.
 4. **Education:** Demonstrating concepts through visual aids.
-

This concludes the experiment on creating 2D plots in MATLAB and Python.

प्रयोग संख्या 4

शीर्षक: 2D प्लॉट्स का निर्माण

1. परिचय और अवलोकन

1.1 उद्देश्य

इस प्रयोग का उद्देश्य MATLAB और Python का उपयोग करके 2D प्लॉट्स बनाना सीखाना है। विद्यार्थी लाइन प्लॉट्स, स्कैटर प्लॉट्स, बार चार्ट्स और अन्य सामान्य 2D प्लॉटिंग तकनीकों के माध्यम से डेटा का विज़ुअलाइज़ेशन करना सीखेंगे। सत्र में प्लॉट्स को अनुकूलित करने की प्रक्रिया को शामिल किया गया है ताकि पढ़ने और प्रस्तुति में आसानी हो।

1.2 2D प्लॉटिंग का अवलोकन

डेटा विश्लेषण में विज़ुअलाइज़ेशन एक महत्वपूर्ण पहलू है। 2D प्लॉट्स इंजीनियरों, वैज्ञानिकों और डिजाइनरों को डेटा के संबंधों को ग्राफिकल रूप से प्रदर्शित करने में सक्षम बनाते हैं, जिससे पैटर्न और रुझानों को समझना आसान हो जाता है। MATLAB और Python में उच्च गुणवत्ता वाले 2D प्लॉट्स बनाने के लिए मजबूत लाइब्रेरीज़ उपलब्ध हैं, जैसे Python में matplotlib।

2. सॉफ्टवेयर पर्यावरण

MATLAB पर्यावरण:

- उपयोग किया गया सॉफ्टवेयर: MATLAB R2023a (या बाद के संस्करण)
- सिस्टम आवश्यकताएं:
 - ऑपरेटिंग सिस्टम: Windows, macOS, या Linux
 - प्रोसेसर: Intel Core i5 या समकक्ष
 - रैम: न्यूनतम 4 GB

Python पर्यावरण:

- उपयोग किया गया सॉफ्टवेयर: Python 3.8 या बाद के संस्करण, matplotlib लाइब्रेरी
 - सिस्टम आवश्यकताएं:
 - ऑपरेटिंग सिस्टम: Windows, macOS, या Linux
 - प्रोसेसर: Intel Core i5 या समकक्ष
 - रैम: न्यूनतम 4 GB
-

3. प्रयोग की विधियां

3.1 MATLAB में प्लॉटिंग

1. सेटअप:

- MATLAB खोलें।
- कोड लागू करने के लिए एक नई स्क्रिप्ट फ़ाइल बनाएं।

2. लाइन प्लॉट बनाना:

```
x = 0:0.1:10; % x-अक्ष के लिए डेटा उत्पन्न करें  
y = sin(x); % x का साइन निकालें  
plot(x, y, 'r', 'LineWidth', 2); % मोटी लाल लाइन प्लॉट बनाएं  
title('Sine Wave');  
xlabel('x-अक्ष');  
ylabel('y-अक्ष');  
grid on; % ग्रिड चालू करें
```

3. स्कैटर प्लॉट:

```
x = rand(1, 20); % रैंडम x मान  
y = rand(1, 20); % रैंडम y मान  
scatter(x, y, 'filled'); % फिल्ड मार्कर्स के साथ स्कैटर प्लॉट बनाएं  
title('Scatter Plot');  
xlabel('Random X Values');  
ylabel('Random Y Values');
```

4. बार चार्ट:

```
categories = {'A', 'B', 'C', 'D'}; % श्रेणियां परिभाषित करें  
values = [3, 7, 8, 5]; % मान परिभाषित करें  
bar(values);  
set(gca, 'xticklabel', categories); % x-अक्ष को लेबल करें  
title('Bar Chart');  
ylabel('Values');
```

3.2 Python में प्लॉटिंग (matplotlib का उपयोग करते हुए)

1. सेटअप:

- यदि matplotlib स्थापित नहीं है, तो इसे स्थापित करें:
- `pip install matplotlib`
- अपनी पसंदीदा Python IDE खोलें (जैसे Jupyter Notebook, VS Code)।

2. लाइन प्लॉट बनाना:

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 10, 100) # x-अक्ष के लिए डेटा उत्पन्न करें

y = np.sin(x) # x का साइन निकालें

plt.plot(x, y, color='red', linewidth=2) # लाल रंग की लाइन प्लॉट बनाएं

plt.title('Sine Wave')

plt.xlabel('x-अक्ष')

plt.ylabel('y-अक्ष')

plt.grid(True) # ग्रिड चालू करें

plt.show()
```

3. स्कैटर प्लॉट:

```
x = np.random.rand(20) # रैंडम x मान

y = np.random.rand(20) # रैंडम y मान

plt.scatter(x, y, color='blue', s=50) # नीले मार्कर्स के साथ स्कैटर प्लॉट बनाएं

plt.title('Scatter Plot')

plt.xlabel('Random X Values')

plt.ylabel('Random Y Values')

plt.show()
```

4. बार चार्ट:

```
categories = ['A', 'B', 'C', 'D'] # श्रेणियां परिभाषित करें

values = [3, 7, 8, 5] # मान परिभाषित करें

plt.bar(categories, values, color='green')

plt.title('Bar Chart')
```

```
plt.ylabel('Values')
```

```
plt.show()
```

4. अपेक्षित परिणाम

1. MATLAB और Python में विभिन्न 2D प्लॉट्स बनाने की क्षमता।
2. शीर्षक, लेबल, लेजेंड्स और ग्रिड का उपयोग करके प्लॉट्स को अनुकूलित करने की समझ।
3. लाइन प्लॉट्स, स्कैटर प्लॉट्स, और बार चार्ट्स जैसी विभिन्न प्लॉटिंग तकनीकों का ज्ञान।

5. सामान्य त्रुटियां और समाधान

1. त्रुटि: x और y डेटा का आकार मेल नहीं खाता।
 - समाधान: सुनिश्चित करें कि x और y एरेस की लंबाई समान हो।
2. त्रुटि: MATLAB में "Undefined function or variable"।
 - समाधान: वेरिएबल या फंक्शन नामों में टाइपो जांचें।
3. त्रुटि: Matplotlib फिगर प्रदर्शित नहीं हो रही।
 - समाधान: plt.show() का उपयोग करें।

6. विचारोत्तेजक प्रश्न

1. प्लॉट को प्रस्तुति उद्देश्यों के लिए पठनीय बनाने के लिए आप क्या कर सकते हैं?
2. MATLAB और Python में प्लॉटिंग क्षमताओं में मुख्य अंतर क्या हैं?
3. एक ही ग्राफ में एक से अधिक लाइन्स को कैसे प्लॉट करेंगे?
4. प्लॉट्स के लिए एक्सिस को लेबल करना और शीर्षक प्रदान करना क्यों महत्वपूर्ण है?

7. 2D प्लॉटिंग के अनुप्रयोग

1. डेटा विश्लेषण: डेटा सेट्स में रुझानों और पैटर्न को विजुअलाइज़ करना।
 2. इंजीनियरिंग डिजाइन: सिमुलेशन परिणामों को ग्राफिकल रूप से प्रदर्शित करना।
 3. वैज्ञानिक अनुसंधान: विश्लेषण के लिए प्रायोगिक डेटा प्रदर्शित करना।
 4. शिक्षा: विजुअल साधनों के माध्यम से अवधारणाओं का प्रदर्शन।
-

यह MATLAB और Python में 2D प्लॉट्स बनाने के प्रयोग का समापन है।

Experiment No. 5

Title: Creation of 3D Plots

1. Introduction and Overview

1.1 Objective

This experiment introduces students to the creation and visualization of 3D plots using MATLAB and Python. By completing this exercise, students will gain the ability to generate 3D surfaces, wireframes, and scatter plots, which are essential for visualizing multi-dimensional data in engineering and scientific applications.

1.2 Overview of 3D Plotting

3D plotting is a critical skill that enables the visualization of relationships between three variables, represented in a three-dimensional Cartesian coordinate system. Both MATLAB and Python provide built-in libraries and functions to create 3D plots efficiently.

2. Software Environment

- **MATLAB**
 - Version: R2022b or later (recommended)
 - Key Toolboxes: MATLAB Graphics System
- **Python**
 - Version: 3.8 or later
 - Required Libraries: matplotlib, numpy
 - IDEs: Jupyter Notebook, PyCharm, or VS Code

3. Methodologies to Perform the Experiment

3.1 Steps in MATLAB

1. **Setup and Initialization:**
 - Open MATLAB and ensure necessary toolboxes are installed.
2. **Creating a 3D Surface Plot:**
 - Define x and y data using the meshgrid function.

```
[X, Y] = meshgrid(-5:0.5:5, -5:0.5:5);
```

```
Z = X.^2 + Y.^2; % Define Z as a function of X and Y
```

```
surf(X, Y, Z); % Create a surface plot

title('3D Surface Plot');

xlabel('X-axis');

ylabel('Y-axis');

zlabel('Z-axis');
```

3. Creating a Wireframe Plot:

- Use the mesh function.

```
mesh(X, Y, Z);

title('3D Wireframe Plot');

xlabel('X-axis'); ylabel('Y-axis'); zlabel('Z-axis');
```

4. Creating a 3D Scatter Plot:

- Generate random data and visualize.

```
x = rand(1, 50) * 10 - 5;

y = rand(1, 50) * 10 - 5;

z = x.^2 + y.^2;

scatter3(x, y, z, 'filled');

title('3D Scatter Plot');

xlabel('X-axis'); ylabel('Y-axis'); zlabel('Z-axis');
```

5. Saving the Figure:

- Use the saveas function.

```
saveas(gcf, '3DPlot.png');
```

3.2 Steps in Python

1. Setup and Initialization:

- Install necessary libraries using pip:

```
pip install matplotlib numpy
```

2. Creating a 3D Surface Plot:

```
import numpy as np

import matplotlib.pyplot as plt

from mpl_toolkits.mplot3d import Axes3D

# Define data
```

```

x = np.linspace(-5, 5, 50)
y = np.linspace(-5, 5, 50)
X, Y = np.meshgrid(x, y)
Z = X**2 + Y**2

# Create plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(X, Y, Z, cmap='viridis')
ax.set_title('3D Surface Plot')
ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis')
ax.set_zlabel('Z-axis')
plt.show()

```

3. Creating a Wireframe Plot:

```

ax.plot_wireframe(X, Y, Z, color='blue')
plt.show()

```

4. Creating a 3D Scatter Plot:

```

x = np.random.uniform(-5, 5, 50)
y = np.random.uniform(-5, 5, 50)
z = x**2 + y**2
ax.scatter(x, y, z, color='red')
ax.set_title('3D Scatter Plot')
plt.show()

```

5. Saving the Figure:

```

plt.savefig('3DPlot.png')

```

4. Expected Outcomes

1. Students will learn to generate 3D surface, wireframe, and scatter plots in MATLAB and Python.
2. Ability to visualize and interpret multi-dimensional data effectively.
3. Proficiency in customizing 3D plots, including axes labels, titles, and color maps.

5. Common Errors and Troubleshooting

1. **Error:** Missing libraries in Python.
 - **Solution:** Install the required libraries using pip (e.g., pip install matplotlib numpy).
 2. **Error:** meshgrid dimensions mismatch.
 - **Solution:** Ensure x and y have compatible dimensions for grid creation.
 3. **Error:** Plot not displaying.
 - **Solution:** Use the plt.show() command at the end of your Python script.
-

6. Thought-Provoking Questions

1. How can color maps enhance the interpretability of 3D surface plots?
 2. What are the differences between a surface plot and a wireframe plot?
 3. How can you modify plot aesthetics to make 3D visualizations more appealing?
-

7. Applications of 3D Plotting

1. **Engineering:** Stress-strain visualization in materials.
 2. **Data Science:** Representing loss functions in machine learning.
 3. **Physics:** Simulating 3D trajectories of particles.
 4. **Finance:** Visualizing risk analysis or option pricing.
-

This concludes the experiment on creating 3D plots using MATLAB and Python.

प्रयोग संख्या 5

शीर्षक: 3D प्लॉट्स का निर्माण

1. परिचय और अवलोकन

1.1 उद्देश्य

यह प्रयोग छात्रों को MATLAB और Python का उपयोग करके 3D प्लॉट्स बनाने और विज़ुअलाइज़ करने का परिचय देता है। इस अभ्यास को पूरा करने के बाद, छात्र 3D सतह, वायरफ्रेम, और स्कैटर प्लॉट्स बनाने में सक्षम होंगे, जो इंजीनियरिंग और वैज्ञानिक अनुप्रयोगों में बहु-आयामी डेटा को विज़ुअलाइज़ करने के लिए आवश्यक हैं।

1.2 3D प्लॉटिंग का अवलोकन

3D प्लॉटिंग एक महत्वपूर्ण कौशल है जो तीन चर के बीच संबंधों को त्रि-आयामी कार्टेशियन समन्वय प्रणाली में विज़ुअलाइज़ करने की अनुमति देता है। MATLAB और Python दोनों में 3D प्लॉट्स कुशलतापूर्वक बनाने के लिए अंतर्निर्मित लाइब्रेरी और फ़ंक्शन उपलब्ध हैं।

2. सॉफ़्टवेयर पर्यावरण

- **MATLAB**
 - संस्करण: R2022b या बाद का (सुझावित)
 - प्रमुख टूलबॉक्स: MATLAB ग्राफिक्स सिस्टम
 - **Python**
 - संस्करण: 3.8 या बाद का
 - आवश्यक लाइब्रेरी: matplotlib, numpy
 - आईडीई: Jupyter Notebook, PyCharm, या VS Code
-

3. प्रयोग करने की विधियां

3.1 MATLAB में चरण

1. **सेटअप और प्रारंभिककरण:**
 - MATLAB खोलें और सुनिश्चित करें कि आवश्यक टूलबॉक्स इंस्टॉल हैं।
2. **3D सतह प्लॉट बनाना:**

```
[X, Y] = meshgrid(-5:0.5:5, -5:0.5:5);  
Z = X.^2 + Y.^2; % Z को X और Y का फ़ंक्शन परिभाषित करें  
surf(X, Y, Z); % सतह प्लॉट बनाएं  
title('3D Surface Plot');  
xlabel('X-axis');  
ylabel('Y-axis');  
zlabel('Z-axis');
```
3. **वायरफ्रेम प्लॉट बनाना:**

```
mesh(X, Y, Z);  
title('3D Wireframe Plot');
```

```
xlabel('X-axis'); ylabel('Y-axis'); zlabel('Z-axis');
```

4. 3D स्कैटर प्लॉट बनाना:

```
x = rand(1, 50) * 10 - 5;  
y = rand(1, 50) * 10 - 5;  
z = x.^2 + y.^2;  
scatter3(x, y, z, 'filled');  
title('3D Scatter Plot');  
xlabel('X-axis'); ylabel('Y-axis'); zlabel('Z-axis');
```

5. फिगर सहेजना:

```
saveas(gcf, '3DPlot.png');
```

3.2 Python में चरण

1. सेटअप और प्रारंभिककरण:

आवश्यक लाइब्रेरी इंस्टॉल करें:

```
pip install matplotlib numpy
```

2. 3D सतह प्लॉट बनाना:

```
import numpy as np  
import matplotlib.pyplot as plt  
from mpl_toolkits.mplot3d import Axes3D  
x = np.linspace(-5, 5, 50)  
y = np.linspace(-5, 5, 50)  
X, Y = np.meshgrid(x, y)  
Z = X**2 + Y**2  
fig = plt.figure()  
ax = fig.add_subplot(111, projection='3d')  
ax.plot_surface(X, Y, Z, cmap='viridis')  
ax.set_title('3D Surface Plot')  
ax.set_xlabel('X-axis')  
ax.set_ylabel('Y-axis')  
ax.set_zlabel('Z-axis')
```

```
plt.show()
```

3. वायरफ्रेम प्लॉट बनाना:

```
ax.plot_wireframe(X, Y, Z, color='blue')  
plt.show()
```

4. 3D स्कैटर प्लॉट बनाना:

```
x = np.random.uniform(-5, 5, 50)  
y = np.random.uniform(-5, 5, 50)  
z = x**2 + y**2  
ax.scatter(x, y, z, color='red')  
ax.set_title('3D Scatter Plot')  
plt.show()
```

5. फिगर सहेजना:

```
plt.savefig('3DPlot.png')
```

4. अपेक्षित परिणाम

1. छात्र MATLAB और Python में 3D सतह, वायरफ्रेम, और स्कैटर प्लॉट्स बनाना सीखेंगे।
2. बहु-आयामी डेटा को प्रभावी ढंग से विज़ुअलाइज़ और व्याख्या करने की क्षमता।
3. 3D प्लॉट्स को अनुकूलित करने में दक्षता, जैसे कि एक्सिस लेबल, शीर्षक, और कलर मैप्स।

5. सामान्य त्रुटियां और समाधान

1. त्रुटि: Python में गायब लाइब्रेरी।
 - समाधान: आवश्यक लाइब्रेरी इंस्टॉल करें (e.g., `pip install matplotlib numpy`)।
2. त्रुटि: meshgrid का आयाम असंगत।
 - समाधान: सुनिश्चित करें कि x और y के आयाम ग्रिड निर्माण के लिए संगत हैं।
3. त्रुटि: प्लॉट प्रदर्शित नहीं हो रहा।
 - समाधान: Python स्क्रिप्ट के अंत में `plt.show()` कमांड का उपयोग करें।

6. सोचने के लिए प्रश्न

1. कलर मैप्स 3D सतह प्लॉट्स की व्याख्या करने की क्षमता को कैसे बढ़ाते हैं?
 2. सतह प्लॉट और वायरफ्रेम प्लॉट में क्या अंतर है?
 3. आप 3D विज़ुअलाइज़ेशन को अधिक आकर्षक बनाने के लिए प्लॉट एस्थेटिक्स को कैसे संशोधित कर सकते हैं?
-

7. 3D प्लॉटिंग के अनुप्रयोग

1. **इंजीनियरिंग:** सामग्रियों में तनाव-तनाव का विज़ुअलाइज़ेशन।
 2. **डेटा विज्ञान:** मशीन लर्निंग में लॉस फ़ंक्शन का प्रतिनिधित्व।
 3. **भौतिकी:** कणों के 3D प्रक्षेप पथ का सिमुलेशन।
 4. **वित्त:** जोखिम विश्लेषण या विकल्प मूल्य निर्धारण का विज़ुअलाइज़ेशन।
-

यह प्रयोग MATLAB और Python का उपयोग करके 3D प्लॉट्स बनाने पर समाप्त होता है।

Experiment No. 6

Title: Curve Fitting with Polynomials

Objective:

To understand and implement polynomial curve fitting techniques using MATLAB and Python, enabling interpolation and approximation of data.

Pre-requisites:

1. Familiarity with the concepts of polynomials.
 2. Basic understanding of MATLAB and Python programming.
 3. Knowledge of plotting and interpreting data.
-

Theory:

Curve fitting is the process of constructing a curve that best fits a given set of data points. Polynomial curve fitting involves fitting data to a polynomial function of degree n , typically represented as:

$$y = p_0 + p_1x^1 + p_2x^2 + \dots + p_nx^n$$

This equation represents a polynomial of degree n , where:

- Y : is the dependent variable.
- X : is the independent variable.
- $p_0, p_1, p_2, \dots, p_n$ are the coefficients of the polynomial.

Applications of polynomial curve fitting include trend analysis, interpolation, and approximation in various engineering and scientific fields.

Tools Required:

1. MATLAB software.
 2. Python (preferably with libraries such as numpy, matplotlib, and scipy).
-

Procedure:

Part A: Polynomial Curve Fitting in MATLAB

1. Generate or Input Data:

- Create a set of known data points (e.g., x and y).

```
x = [1, 2, 3, 4, 5];
```

```
y = [2.1, 4.1, 6.2, 8.3, 10.4];
```

2. Fit a Polynomial:

- Use the `polyfit` function to fit a polynomial of degree n to the data.

```
n = 2; % Degree of polynomial
```

```
p = polyfit(x, y, n);
```

3. Evaluate the Polynomial:

- Use the `polyval` function to evaluate the polynomial at specific points.

```
x_fit = linspace(min(x), max(x), 100);
```

```
y_fit = polyval(p, x_fit);
```

4. Plot the Results:

- Plot the original data and the fitted curve.

```
plot(x, y, 'o', 'MarkerSize', 8, 'DisplayName', 'Original Data');
```

```
hold on;
```

```
plot(x_fit, y_fit, 'r-', 'LineWidth', 1.5, 'DisplayName', 'Fitted Curve');  
legend;  
xlabel('X-axis'); ylabel('Y-axis');  
title('Polynomial Curve Fitting');  
grid on;  
hold off;
```

Part B: Polynomial Curve Fitting in Python

1. Import Required Libraries:

```
import numpy as np  
import matplotlib.pyplot as plt
```

2. Generate or Input Data:

```
x = np.array([1, 2, 3, 4, 5])  
y = np.array([2.1, 4.1, 6.2, 8.3, 10.4])
```

3. Fit a Polynomial:

- Use numpy.polyfit to fit a polynomial of degree n to the data.

```
n = 2 # Degree of polynomial  
p = np.polyfit(x, y, n)
```

4. Evaluate the Polynomial:

- Use numpy.polyval to evaluate the polynomial at specific points.

```
x_fit = np.linspace(min(x), max(x), 100)  
y_fit = np.polyval(p, x_fit)
```

5. Plot the Results:

- Plot the original data and the fitted curve.

```
plt.scatter(x, y, color='blue', label='Original Data', zorder=5)  
plt.plot(x_fit, y_fit, 'r-', linewidth=1.5, label='Fitted Curve')  
plt.xlabel('X-axis')  
plt.ylabel('Y-axis')  
plt.title('Polynomial Curve Fitting')  
plt.legend()  
plt.grid(True)  
plt.show()
```

Observations:

1. Compare the fitted polynomial curve to the original data points.
2. Experiment with different degrees of polynomials (e.g., $n = 1$, $n = 2$, $n = 3$) and observe the effect on the fit.
3. Record the coefficients of the polynomial and interpret their significance.

Results:

- The polynomial curve that best fits the data was successfully created in MATLAB and Python.
- The degree of the polynomial significantly affects the quality of the fit.

Conclusion:

Polynomial curve fitting is a powerful tool for approximating and interpolating data. MATLAB and Python provide robust functions for performing curve fitting and visualizing the results. This experiment reinforces the importance of selecting an appropriate polynomial degree to avoid overfitting or underfitting.

Viva Questions:

1. What is the role of the degree of a polynomial in curve fitting?
2. How does MATLAB's polyfit function work?
3. What is the difference between interpolation and approximation?
4. How can overfitting be prevented in polynomial curve fitting?
5. Can polynomial curve fitting be used for non-linear data? Justify your answer.

प्रयोग संख्या 6

शीर्षक: बहुपदों के साथ वक्र फिटिंग

उद्देश्य:

MATLAB और Python का उपयोग करके बहुपदीय वक्र फिटिंग तकनीकों को समझना और लागू करना, जो डेटा के मध्यस्थता और सन्निकटन को सक्षम बनाता है।

पूर्व-आवश्यकताएँ:

1. बहुपदों की अवधारणाओं से परिचितता।
2. MATLAB और Python प्रोग्रामिंग की बुनियादी समझ।
3. डेटा का प्लॉटिंग और व्याख्या करने का ज्ञान।

सिद्धांत:

वक्र फिटिंग वह प्रक्रिया है जिसके माध्यम से डेटा बिंदुओं के एक सेट को सर्वोत्तम तरीके से फिट करने वाली वक्र बनाई जाती है। बहुपदीय वक्र फिटिंग में डेटा को डिग्री n के बहुपद फंक्शन में फिट करना शामिल होता है, जिसे आम तौर पर इस प्रकार दर्शाया जाता है:

$$y = p_0 + p_1x^1 + p_2x^2 + \dots + p_nx^n$$

यह समीकरण डिग्री n के बहुपद का प्रतिनिधित्व करता है, जहां:

- y : आश्रित चर है।
- x : स्वतंत्र चर है।
- p_0, p_1, p_2, p_n : बहुपद के गुणांक हैं।

बहुपदीय वक्र फिटिंग के अनुप्रयोगों में ट्रेंड विश्लेषण, मध्यस्थता, और विभिन्न इंजीनियरिंग व वैज्ञानिक क्षेत्रों में सन्निकटन शामिल हैं।

आवश्यक उपकरण:

1. MATLAB सॉफ्टवेयर।
2. Python (विशेष रूप से numpy, matplotlib, और scipy जैसी लाइब्रेरी के साथ)।

प्रक्रिया:

भाग A: MATLAB में बहुपदीय वक्र फिटिंग

1. डेटा उत्पन्न करना या इनपुट देना:

% ज्ञात डेटा बिंदु बनाएं

$x = [1, 2, 3, 4, 5];$

$y = [2.1, 4.1, 6.2, 8.3, 10.4];$

2. बहुपद फिट करना:

3. % डेटा को फिट करने के लिए polyfit फंक्शन का उपयोग करें

```
n = 2; % बहुपद की डिग्री
```

```
p = polyfit(x, y, n);
```

4. बहुपद का मूल्यांकन करना:

```
% निर्दिष्ट बिंदुओं पर बहुपद का मूल्यांकन करें
```

```
x_fit = linspace(min(x), max(x), 100);
```

```
y_fit = polyval(p, x_fit);
```

5. परिणामों को प्लॉट करना:

```
% मूल डेटा और फिट की गई वक्र को प्लॉट करें
```

```
plot(x, y, 'o', 'MarkerSize', 8, 'DisplayName', 'मूल डेटा');
```

```
hold on;
```

```
plot(x_fit, y_fit, 'r-', 'LineWidth', 1.5, 'DisplayName', 'फिट की गई वक्र');
```

```
legend;
```

```
xlabel('X-अक्ष'); ylabel('Y-अक्ष');
```

```
title('बहुपदीय वक्र फिटिंग');
```

```
grid on;
```

```
hold off;
```

भाग B: Python में बहुपदीय वक्र फिटिंग

1. आवश्यक लाइब्रेरी आयात करें:

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

2. डेटा उत्पन्न करना या इनपुट देना:

```
x = np.array([1, 2, 3, 4, 5])
```

```
y = np.array([2.1, 4.1, 6.2, 8.3, 10.4])
```

3. बहुपद फिट करना:

```
n = 2 # बहुपद की डिग्री
```

```
p = np.polyfit(x, y, n)
```

4. बहुपद का मूल्यांकन करना:

```
x_fit = np.linspace(min(x), max(x), 100)
```

```
y_fit = np.polyval(p, x_fit)
```

5. परिणामों को प्लॉट करना:

```
plt.scatter(x, y, color='blue', label='मूल डेटा', zorder=5)

plt.plot(x_fit, y_fit, 'r-', linewidth=1.5, label='फिट की गई वक्र')

plt.xlabel('X-अक्ष')

plt.ylabel('Y-अक्ष')

plt.title('बहुपदीय वक्र फिटिंग')

plt.legend()

plt.grid(True)

plt.show()
```

अवलोकन:

1. मूल डेटा बिंदुओं के साथ फिट की गई बहुपदीय वक्र की तुलना करें।
 2. विभिन्न डिग्री के बहुपदों (जैसे, $n=1, n=2, n=3$) के साथ प्रयोग करें और फिट पर प्रभाव का अवलोकन करें।
 3. बहुपद के गुणांक रिकॉर्ड करें और उनके महत्व की व्याख्या करें।
-

परिणाम:

- MATLAB और Python में डेटा को सर्वोत्तम रूप से फिट करने वाली बहुपदीय वक्र सफलतापूर्वक बनाई गई।
 - बहुपद की डिग्री फिट की गुणवत्ता को महत्वपूर्ण रूप से प्रभावित करती है।
-

निष्कर्ष:

बहुपदीय वक्र फिटिंग डेटा को मध्यस्थता और सन्निकटन के लिए एक शक्तिशाली उपकरण है। MATLAB और Python वक्र फिटिंग को निष्पादित करने और परिणामों को विज़ुअलाइज़ करने के लिए मजबूत फ़ंक्शन प्रदान करते हैं। यह प्रयोग यह पुष्टि करता है कि अत्यधिक फिटिंग या अपर्याप्त फिटिंग से बचने के लिए उपयुक्त बहुपद डिग्री का चयन महत्वपूर्ण है।

विवा प्रश्न:

1. वक्र फिटिंग में बहुपद की डिग्री की भूमिका क्या है?

2. MATLAB के polyfit फ़ंक्शन का कार्य कैसे होता है?
3. मध्यस्थता और सन्निकटन में क्या अंतर है?
4. बहुपदीय वक्र फिटिंग में अत्यधिक फिटिंग को कैसे रोका जा सकता है?
5. क्या बहुपदीय वक्र फिटिंग गैर-रेखीय डेटा के लिए उपयोग की जा सकती है? अपना उत्तर सही ठहराएं।

Experiment No. 7

Title: Programming Applications in Numerical Analysis

Objective:

To understand and implement numerical analysis techniques using MATLAB and Python, focusing on solving mathematical problems such as root finding, integration, differentiation, and solving linear systems.

Pre-requisites:

1. Basic understanding of numerical methods.
 2. Familiarity with MATLAB and Python programming.
 3. Knowledge of mathematical operations and concepts like roots of equations, integration, and differentiation.
-

Theory:

Numerical analysis involves algorithms for approximating solutions to mathematical problems that cannot be solved analytically. Some common applications include:

1. **Root Finding:** Determining the roots of equations (e.g., using bisection, Newton-Raphson, or secant methods).
 2. **Integration:** Approximating definite integrals (e.g., using trapezoidal or Simpson's rule).
 3. **Differentiation:** Calculating derivatives numerically.
 4. **Solving Linear Systems:** Finding solutions to systems of linear equations (e.g., Gaussian elimination, LU decomposition).
-

Tools Required:

1. MATLAB software.

2. Python (with libraries like numpy, scipy, and matplotlib).
-

Procedure:

Part A: Numerical Analysis in MATLAB

1. Root Finding Using Newton-Raphson Method:

- Define the function and its derivative.

```
f = @(x) x^3 - 4*x - 9;  
df = @(x) 3*x^2 - 4;  
x0 = 2; % Initial guess  
tol = 1e-6; % Tolerance  
max_iter = 100;  
for i = 1:max_iter  
    x1 = x0 - f(x0)/df(x0);  
    if abs(x1 - x0) < tol  
        fprintf('Root found: %.6f\n', x1);  
        break;  
    end  
    x0 = x1;  
end
```

2. Numerical Integration Using Trapezoidal Rule:

```
f = @(x) x^2;  
a = 0; b = 1; % Limits of integration  
n = 100; % Number of intervals  
h = (b - a) / n;  
x = a:h:b;  
y = f(x);  
integral = h * (sum(y) - (y(1) + y(end)) / 2);  
fprintf('Approximate Integral: %.6f\n', integral);
```

3. Solving Linear Systems Using MATLAB:

```
A = [2, -1, 1; 1, 3, 2; 1, -1, 2];  
b = [3; 12; 7];  
x = A\b; % Solve Ax = b
```

```
disp('Solution:');
```

```
disp(x);
```

Part B: Numerical Analysis in Python

1. Root Finding Using Newton-Raphson Method:

```
import numpy as np

def f(x):
    return x**3 - 4*x - 9

def df(x):
    return 3*x**2 - 4

x0 = 2 # Initial guess
tol = 1e-6
max_iter = 100

for i in range(max_iter):
    x1 = x0 - f(x0)/df(x0)
    if abs(x1 - x0) < tol:
        print(f"Root found: {x1:.6f}")
        break
    x0 = x1
```

2. Numerical Integration Using Trapezoidal Rule:

```
import numpy as np

def f(x):
    return x**2

a, b = 0, 1 # Integration limits
n = 100 # Number of intervals
x = np.linspace(a, b, n+1)
y = f(x)
h = (b - a) / n

integral = h * (np.sum(y) - (y[0] + y[-1]) / 2)

print(f"Approximate Integral: {integral:.6f}")
```

3. Solving Linear Systems in Python:

```
import numpy as np
```

```
A = np.array([[2, -1, 1], [1, 3, 2], [1, -1, 2]])  
b = np.array([3, 12, 7])  
x = np.linalg.solve(A, b)  
print("Solution:")  
print(x)
```

Observations:

1. Compare the numerical results with analytical solutions where applicable.
 2. Analyze the effect of varying parameters (e.g., initial guess, number of intervals) on the accuracy of the solution.
 3. Record the time taken for different numerical methods.
-

Results:

- Successfully implemented root finding, integration, differentiation, and solving linear systems using MATLAB and Python.
 - Observed the effect of parameters such as tolerance and step size on the accuracy of solutions.
-

Conclusion:

This experiment demonstrates the application of programming in numerical analysis. MATLAB and Python provide efficient tools for solving complex mathematical problems numerically, which is critical for engineering and scientific computations.

Viva Questions:

1. What is the difference between analytical and numerical solutions?
2. How does the Newton-Raphson method find the root of an equation?
3. Why is numerical integration used instead of analytical integration?
4. What are the advantages of using MATLAB or Python for numerical computations?
5. How does the step size affect the accuracy of numerical integration?

प्रयोग क्रमांक 7

शीर्षक: संख्यात्मक विश्लेषण में प्रोग्रामिंग अनुप्रयोग

उद्देश्य:

MATLAB और Python का उपयोग करके संख्यात्मक विश्लेषण तकनीकों को समझना और लागू करना, विशेष रूप से जड़ों को खोजने, समाकलन (इंटीग्रेशन), अवकलन (डिफरेंशिएशन), और रेखीय प्रणालियों (लिनियर सिस्टम्स) को हल करने जैसे गणितीय समस्याओं को सुलझाने पर ध्यान केंद्रित करना।

पूर्व-आवश्यकताएँ:

- संख्यात्मक विधियों की बुनियादी समझ।
- MATLAB और Python प्रोग्रामिंग का ज्ञान।
- समीकरणों की जड़ें, समाकलन और अवकलन जैसे गणितीय अवधारणाओं की जानकारी।

सिद्धांत:

संख्यात्मक विश्लेषण उन एल्गोरिदम को शामिल करता है जो ऐसी गणितीय समस्याओं के लिए समाधान का अनुमान लगाते हैं जिन्हें विश्लेषणात्मक (एनालिटिकल) रूप से हल नहीं किया जा सकता। इसके सामान्य अनुप्रयोगों में शामिल हैं:

- जड़ें खोजना (Root Finding):** समीकरणों की जड़ें निर्धारित करना (जैसे, बाइसेक्शन, न्यूटन-रैफसन या सेकेंट विधियों का उपयोग)।
- समाकलन (Integration):** निश्चित समाकलनों का अनुमान लगाना (जैसे, ट्रैपेज़ॉइडल या सिम्पसन नियम का उपयोग)।
- अवकलन (Differentiation):** संख्यात्मक रूप से अवकलज की गणना।
- रेखीय प्रणालियों का हल (Solving Linear Systems):** रेखीय समीकरणों की प्रणालियों का समाधान ढूंढना (जैसे, गॉसियन उन्मूलन, LU विघटन)।

आवश्यक उपकरण:

- MATLAB सॉफ्टवेयर।
- Python (जैसे लाइब्रेरी: numpy, scipy, matplotlib)।

प्रक्रिया:

भाग A: MATLAB में संख्यात्मक विश्लेषण

- न्यूटन-रैफसन विधि का उपयोग करके जड़ें खोजना:

$$f = @(x) x^3 - 4*x - 9;$$

```

df = @(x) 3*x^2 - 4;
x0 = 2; % प्रारंभिक अनुमान
tol = 1e-6; % सहनशीलता
max_iter = 100;
for i = 1:max_iter
    x1 = x0 - f(x0)/df(x0);
    if abs(x1 - x0) < tol
        fprintf('जड़ मिली: %.6f\n', x1);
        break;
    end
    x0 = x1;
end

```

2. ट्रैपेज़ॉइडल नियम का उपयोग करके समाकलन:

```

f = @(x) x^2;
a = 0; b = 1; % समाकलन की सीमाएँ
n = 100; % खंडों की संख्या
h = (b - a) / n;
x = a:h:b;
y = f(x);
integral = h * (sum(y) - (y(1) + y(end)) / 2);
fprintf('अनुमानित समाकलन: %.6f\n', integral);

```

3. MATLAB में रेखीय प्रणालियों का हल:

```

A = [2, -1, 1; 1, 3, 2; 1, -1, 2];
b = [3; 12; 7];
x = A\b; % हल Ax = b
disp('हल:');
disp(x);

```

भाग B: Python में संख्यात्मक विश्लेषण

1. न्यूटन-रैफसन विधि का उपयोग करके जड़ें खोजना:

```

import numpy as np

def f(x):
    return x**3 - 4*x - 9

def df(x):
    return 3*x**2 - 4

x0 = 2 # प्रारंभिक अनुमान
tol = 1e-6
max_iter = 100

for i in range(max_iter):
    x1 = x0 - f(x0)/df(x0)
    if abs(x1 - x0) < tol:
        print(f"जड़ मिली: {x1:.6f}")
        break
    x0 = x1

```

2. ट्रैपेज़ॉइडल नियम का उपयोग करके समाकलन:

```

import numpy as np

def f(x):
    return x**2

a, b = 0, 1 # समाकलन की सीमाएँ
n = 100 # खंडों की संख्या
x = np.linspace(a, b, n+1)
y = f(x)
h = (b - a) / n
integral = h * (np.sum(y) - (y[0] + y[-1])) / 2
print(f"अनुमानित समाकलन: {integral:.6f}")

```

3. Python में रेखीय प्रणालियों का हल:

```

import numpy as np

A = np.array([[2, -1, 1], [1, 3, 2], [1, -1, 2]])
b = np.array([3, 12, 7])
x = np.linalg.solve(A, b)

```

```
print("हल:")
```

```
print(x)
```

पर्यवेक्षण:

1. संख्यात्मक परिणामों की तुलना जहाँ संभव हो, विश्लेषणात्मक समाधानों से करें।
2. मापदंडों (जैसे, प्रारंभिक अनुमान, खंडों की संख्या) में परिवर्तन का सटीकता पर प्रभाव विश्लेषण करें।
3. विभिन्न संख्यात्मक विधियों के लिए समय का मापन करें।

परिणाम:

- MATLAB और Python का उपयोग करके सफलतापूर्वक जड़ें खोजी, समाकलन किया, अवकलन किया, और रेखीय प्रणालियाँ हल कीं।
- सहनशीलता और कदम के आकार जैसे मापदंडों का समाधान की सटीकता पर प्रभाव देखा।

निष्कर्ष:

इस प्रयोग ने संख्यात्मक विश्लेषण में प्रोग्रामिंग के अनुप्रयोग को प्रदर्शित किया। MATLAB और Python जटिल गणितीय समस्याओं को संख्यात्मक रूप से हल करने के लिए कुशल उपकरण प्रदान करते हैं, जो इंजीनियरिंग और वैज्ञानिक गणनाओं के लिए महत्वपूर्ण हैं।

वाइवा प्रश्न:

1. विश्लेषणात्मक और संख्यात्मक समाधान में क्या अंतर है?
2. न्यूटन-रैफसन विधि समीकरण की जड़ें कैसे खोजती है?
3. विश्लेषणात्मक समाकलन के बजाय संख्यात्मक समाकलन का उपयोग क्यों किया जाता है?
4. MATLAB या Python का उपयोग संख्यात्मक गणनाओं के लिए कैसे लाभदायक है?
5. कदम के आकार का संख्यात्मक समाकलन की सटीकता पर क्या प्रभाव पड़ता है?

Experiment No. 8:

Title: Robot Programming and Path Planning

Objective:

To understand and implement basic robot programming concepts and path planning algorithms, enabling the design of efficient robotic motion and trajectory planning in simulated or real environments.

Pre-requisites:

1. Basic knowledge of robotics and kinematics.

2. Familiarity with programming in MATLAB and Python.
3. Understanding of coordinate systems and transformation matrices.
4. Knowledge of simulation tools (e.g., Robot Operating System (ROS), Robot Analyzer, or MATLAB Robotics Toolbox).

Theory:

Robot programming involves instructing a robot to perform specific tasks by creating a sequence of actions or movements. Path planning refers to generating an optimal path for a robot to move from a start position to a target while avoiding obstacles. Key algorithms used in path planning include:

1. **Graph-based Algorithms:**
 - Breadth-First Search (BFS)
 - Depth-First Search (DFS)
 - A* Algorithm
2. **Sampling-based Algorithms:**
 - Rapidly-exploring Random Trees (RRT)
 - Probabilistic Roadmap (PRM)
3. **Optimization-based Algorithms:**
 - Dijkstra's Algorithm
 - Gradient Descent

Applications of robot programming and path planning include industrial automation, autonomous vehicles, and service robots.

Tools Required:

1. MATLAB or Python with robotics libraries (e.g., numpy, matplotlib, scipy, pymap2d for Python).
2. Simulation tools (e.g., ROS, RobotStudio, MATLAB Simulink).

Procedure:

Part A: Robot Programming in MATLAB

1. **Define the Robot Model:**
 - Create a robot model using the Robotics Toolbox or a predefined robot configuration.

```
L(1) = Link([0, 0.5, 0.1, pi/2, 0]); % Define links
```

```
L(2) = Link([0, 0.3, 0, 0, 0]);
```

```
robot = SerialLink(L, 'name', '2DOF Robot');
```

2. Define Target Points:

- Specify start and target positions in the workspace.

```
start_pos = [0, 0.5];
```

```
target_pos = [0.4, 0.8];
```

3. Generate a Trajectory:

- Use trajectory generation functions to create a smooth path.

```
t = linspace(0, 1, 50);
```

```
x_traj = linspace(start_pos(1), target_pos(1), 50);
```

```
y_traj = linspace(start_pos(2), target_pos(2), 50);
```

```
trajectory = [x_traj; y_traj]';
```

4. Visualize the Path:

- Plot the robot's trajectory in the workspace.

```
figure;
```

```
plot(trajectory(:, 1), trajectory(:, 2), 'r-', 'LineWidth', 1.5);
```

```
hold on;
```

```
robot.plot([0, 0]);
```

```
title('Robot Path Planning');
```

```
xlabel('X-axis'); ylabel('Y-axis');
```

```
grid on;
```

```
hold off;
```

Part B: Path Planning in Python

1. Import Required Libraries:

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
from scipy.spatial import KDTree
```

2. Define the Workspace:

- Create a 2D workspace with obstacles.

```
workspace = np.zeros((10, 10))
```

```
workspace[3:7, 4:6] = 1 # Adding obstacles
```



```
f_score[neighbor] = tentative_g_score + heuristic(neighbor, goal)
if neighbor not in open_list:
    open_list.append(neighbor)
return []
```

```
start = (0, 0)
goal = (9, 9)
path = astar(workspace, start, goal)
```

4. Visualize the Path:

```
plt.imshow(workspace, cmap='Greys', origin='lower')
path_x, path_y = zip(*path)
plt.plot(path_y, path_x, 'r-', linewidth=2)
plt.scatter(start[1], start[0], color='green', label='Start')
plt.scatter(goal[1], goal[0], color='blue', label='Goal')
plt.legend()
plt.title('Path Planning with A*')
plt.show()
```

Observations:

1. Compare the planned path to the actual obstacles in the workspace.
2. Experiment with different start and target points to analyze the performance of the algorithm.
3. Record the execution time and efficiency of the path planning algorithm.

Results:

- The robot was successfully programmed to move along the computed path.
- The path planning algorithm generated an optimal and collision-free path.

Conclusion:

Robot programming and path planning are crucial components in robotics. Using MATLAB and Python, students can effectively simulate and implement these techniques, providing insights into

real-world robotic systems. The choice of the path planning algorithm depends on the complexity and constraints of the workspace.

Viva Questions:

1. What is the difference between global and local path planning?
2. Explain the working of the A* algorithm.
3. How can obstacles be represented in a 2D workspace?
4. What are the limitations of graph-based path planning algorithms?
5. How do sampling-based algorithms differ from optimization-based algorithms?

प्रयोग क्र. 8

शीर्षक: रोबोट प्रोग्रामिंग और पथ नियोजन

उद्देश्य:

मूलभूत रोबोट प्रोग्रामिंग अवधारणाओं और पथ नियोजन एल्गोरिदम को समझना और लागू करना, जिससे सिमुलेटेड या वास्तविक वातावरण में कुशल रोबोटिक गति और प्रक्षेप पथ नियोजन को डिज़ाइन किया जा सके।

पूर्व-आवश्यकताएँ:

1. रोबोटिक्स और किनेमैटिक्स का बुनियादी ज्ञान।
 2. MATLAB और Python प्रोग्रामिंग की जानकारी।
 3. निर्देशांक प्रणालियों और ट्रांसफॉर्मेशन मैट्रिक्स की समझ।
 4. सिमुलेशन टूल्स (जैसे ROS, Robot Analyzer, या MATLAB Robotics Toolbox) का ज्ञान।
-

सिद्धांत:

रोबोट प्रोग्रामिंग में रोबोट को विशिष्ट कार्य करने के लिए निर्देश देना शामिल है, जिसमें एक्शन या मूवमेंट की क्रमिक श्रृंखला बनानी होती है। पथ नियोजन का अर्थ है कि रोबोट के लिए एक प्रारंभिक स्थिति से लक्ष्य तक पहुँचने के लिए एक इष्टतम पथ तैयार करना, जो बाधाओं से बचते हुए किया जाए। पथ नियोजन में उपयोग होने वाले मुख्य एल्गोरिदम हैं:

1. **ग्राफ आधारित एल्गोरिदम:**
 - ब्रेड्थ-फर्स्ट सर्च (BFS)
 - डेप्थ-फर्स्ट सर्च (DFS)

- A* एल्गोरिदम
- 2. सैंपलिंग आधारित एल्गोरिदम:
 - रैपिडली-एक्सप्लोरिंग रैंडम ट्री (RRT)
 - प्रोबैबिलिस्टिक रोडमैप (PRM)
- 3. ऑप्टिमाइजेशन आधारित एल्गोरिदम:
 - डायक्स्ट्रा एल्गोरिदम
 - ग्रेडिएंट डीसेंट

रोबोट प्रोग्रामिंग और पथ नियोजन के अनुप्रयोगों में औद्योगिक स्वचालन, स्वायत्त वाहन, और सेवा रोबोट शामिल हैं।

आवश्यक उपकरण:

1. MATLAB या Python रोबोटिक्स लाइब्रेरी (जैसे numpy, matplotlib, scipy, pymap2d)।
2. सिमुलेशन टूल्स (जैसे ROS, RobotStudio, MATLAB Simulink)।

प्रक्रिया:

भाग A: MATLAB में रोबोट प्रोग्रामिंग

1. रोबोट मॉडल परिभाषित करें:

```
L(1) = Link([0, 0.5, 0.1, pi/2, 0]);  
L(2) = Link([0, 0.3, 0, 0, 0]);  
robot = SerialLink(L, 'name', '2DOF Robot');
```

2. लक्ष्य बिंदु परिभाषित करें:

```
start_pos = [0, 0.5];  
target_pos = [0.4, 0.8];
```

3. प्रक्षेप पथ उत्पन्न करें:

```
t = linspace(0, 1, 50);  
x_traj = linspace(start_pos(1), target_pos(1), 50);  
y_traj = linspace(start_pos(2), target_pos(2), 50);  
trajectory = [x_traj; y_traj]';
```

4. पथ का चित्रण करें:

```
figure;  
plot(trajecory(:, 1), trajecory(:, 2), 'r-', 'LineWidth', 1.5);  
hold on;  
robot.plot([0, 0]);  
title('Robot Path Planning');  
xlabel('X-axis'); ylabel('Y-axis');  
grid on;  
hold off;
```

भाग B: Python में पथ नियोजन

1. आवश्यक लाइब्रेरी आयात करें:

```
import numpy as np  
import matplotlib.pyplot as plt  
from scipy.spatial import KDTree
```

2. कार्यक्षेत्र परिभाषित करें:

```
workspace = np.zeros((10, 10))  
workspace[3:7, 4:6] = 1 # बाधाएँ जोड़ना पथ नियोजन एल्गोरिदम लागू करें (A):*  
  
def heuristic(a, b):  
    return np.linalg.norm(np.array(a) - np.array(b))  
  
def astar(workspace, start, goal):  
    neighbors = [(0, 1), (0, -1), (1, 0), (-1, 0)]  
    open_list = [start]  
    came_from = {}  
    g_score = {start: 0}  
    f_score = {start: heuristic(start, goal)}  
  
    while open_list:  
        current = min(open_list, key=lambda x: f_score.get(x, float('inf')))  
        if current == goal:  
            path = []
```

```

while current in came_from:
    path.append(current)
    current = came_from[current]
path.append(start)
return path[::-1]
open_list.remove(current)
for dx, dy in neighbors:
    neighbor = (current[0] + dx, current[1] + dy)
    if 0 <= neighbor[0] < workspace.shape[0] and 0 <= neighbor[1] <
workspace.shape[1]:
        if workspace[neighbor] == 1:
            continue
        tentative_g_score = g_score[current] + 1
        if tentative_g_score < g_score.get(neighbor, float('inf')):
            came_from[neighbor] = current
            g_score[neighbor] = tentative_g_score
            f_score[neighbor] = tentative_g_score + heuristic(neighbor, goal)
            if neighbor not in open_list:
                open_list.append(neighbor)
return []
start = (0, 0)
goal = (9, 9)
path = astar(workspace, start, goal)

```

3. पथ का चित्रण करें:

```

plt.imshow(workspace, cmap='Greys', origin='lower')
path_x, path_y = zip(*path)
plt.plot(path_y, path_x, 'r-', linewidth=2)
plt.scatter(start[1], start[0], color='green', label='Start')
plt.scatter(goal[1], goal[0], color='blue', label='Goal')
plt.legend()
plt.title('Path Planning with A*')

```

plt.show()

अवलोकन:

1. कार्यक्षेत्र में बाधाओं के अनुसार नियोजित पथ का मूल्यांकन करें।
2. विभिन्न प्रारंभिक और लक्ष्य बिंदुओं के साथ प्रयोग करें।
3. पथ नियोजन एल्गोरिदम की दक्षता और निष्पादन समय दर्ज करें।

परिणाम:

- रोबोट को सफलतापूर्वक गणना किए गए पथ के साथ प्रोग्राम किया गया।
- पथ नियोजन एल्गोरिदम ने एक इष्टतम और बाधा-मुक्त पथ उत्पन्न किया।

निष्कर्ष:

रोबोट प्रोग्रामिंग और पथ नियोजन रोबोटिक्स के महत्वपूर्ण घटक हैं। MATLAB और Python का उपयोग करके, छात्र इन तकनीकों का प्रभावी ढंग से सिमुलेशन और कार्यान्वयन कर सकते हैं, जो वास्तविक रोबोटिक प्रणालियों में अंतर्दृष्टि प्रदान करते हैं। पथ नियोजन एल्गोरिदम का चयन कार्यक्षेत्र की जटिलता और बाधाओं पर निर्भर करता है।

वायवा प्रश्न:

1. ग्लोबल और लोकल पथ नियोजन में क्या अंतर है?
2. A* एल्गोरिदम कैसे काम करता है?
3. 2D कार्यक्षेत्र में बाधाओं को कैसे दर्शाया जा सकता है?
4. ग्राफ-आधारित पथ नियोजन एल्गोरिदम की सीमाएँ क्या हैं?
5. सैंपलिंग-आधारित एल्गोरिदम और ऑप्टिमाइजेशन-आधारित एल्गोरिदम में क्या अंतर है?

Experiment No. 9

Title: Controlling of Robotics Manipulator using Arduino/Robot Programming

Objective:

1. To understand the basics of robotics manipulation and control systems.
2. To program and control a robotic manipulator using Arduino and robot programming.

3. To implement basic movements and joint control using Arduino to control robotic arms and manipulators.
 4. To use Robot Analyzer software for simulating and testing different robotic manipulator movements.
-

Apparatus/Equipment Required:

1. **Arduino UNO** (or any Arduino-compatible board)
 2. **Servo motors** (for controlling the joints of the robotic manipulator)
 3. **Robotic Manipulator Kit** (with at least 2-3 degrees of freedom)
 4. **Power supply** (for Arduino and servos)
 5. **Jumper wires** (for connections)
 6. **Breadboard** (for circuit setup)
 7. **Robot Analyzer software** (for simulating robot movements)
 8. **PC with Arduino IDE** (for programming)
 9. **USB Cable** (for connecting Arduino to PC)
-

Pre-Lab Theory:

Robotic manipulators are mechanical devices used to perform various tasks like lifting, sorting, and assembly. They typically consist of multiple joints (e.g., servo motors) that move in a controlled manner. In this experiment, you will control the movement of a robotic manipulator using an Arduino.

Key Concepts:

- **Degrees of Freedom (DOF):** The number of independent movements a robotic arm can make. Common robotic manipulators have 2 or 3 DOF.
 - **Servo Motors:** Used for precise control of joint angles in robotic manipulators.
 - **Inverse Kinematics:** The mathematical process to determine the necessary joint angles to position the end-effector (e.g., gripper) at a specific point in space.
 - **Robot Analyzer Software:** A simulation tool to model and visualize robotic arm movements and configurations.
-

Procedure:

Step 1: Hardware Setup

1. Connect the servo motors to the robotic manipulator kit, ensuring that each motor controls a specific joint of the manipulator.

2. Connect each servo motor to an appropriate PWM pin on the Arduino board.
3. Ensure the Arduino board is connected to your PC via the USB cable.
4. Use a breadboard and jumper wires to complete the connections for powering the Arduino and servo motors.

Step 2: Robot Programming (Arduino Code)

1. Open the **Arduino IDE** on your PC.
2. Write or load the code to control the robotic manipulator. A basic example to control a servo motor:

```
#include <Servo.h>

Servo servo1; // Create a servo object for joint 1
Servo servo2; // Create a servo object for joint 2

void setup() {
    servo1.attach(9); // Connect servo1 to pin 9 on Arduino
    servo2.attach(10); // Connect servo2 to pin 10 on Arduino }

void loop() {
    // Move servo1 from 0 to 90 degrees
    for (int pos = 0; pos <= 90; pos++) {
        servo1.write(pos);
        delay(15); }

    // Move servo2 from 0 to 90 degrees
    for (int pos = 0; pos <= 90; pos++) {
        servo2.write(pos);
        delay(15); }

    // Reset servo positions
    for (int pos = 90; pos >= 0; pos--) {
        servo1.write(pos);
        delay(15); }

    for (int pos = 90; pos >= 0; pos--) {
        servo2.write(pos);
        delay(15);
    } }
```

3. Upload the code to the Arduino board using the Arduino IDE.

4. Monitor the manipulator's response to check if the movements align with the programmed angles.

Step 3: Simulation using Robot Analyzer Software

1. Open **Robot Analyzer** on your PC.
2. Load a model of a simple robotic manipulator (2-DOF or 3-DOF).
3. Input the joint angles that you programmed into the Arduino (e.g., 0-90 degrees for joint 1 and 0-90 degrees for joint 2).
4. Observe the manipulator's movement in the simulation environment to verify the accuracy of your code.

Step 4: Experimentation and Adjustment

1. Test different movements and joint combinations, and analyze the behavior of the manipulator.
2. Try to perform a simple task using the manipulator (e.g., moving an object or reaching a target position).
3. Adjust the code or hardware connections if needed for smoother operation.

Observations:

- Note any irregularities in movement or position of the robotic manipulator.
- Record the time it takes for the manipulator to reach its target positions.
- Analyze how accurately the manipulator follows the programmed commands.

Post-Lab Questions:

1. How does the number of degrees of freedom (DOF) of a manipulator affect its movement?
2. Explain the role of inverse kinematics in controlling a robotic manipulator.
3. How would you modify the Arduino code to control more joints (e.g., add more servos)?
4. What challenges did you face in programming the manipulator, and how did you overcome them?
5. Compare the real-world movement of the manipulator with the simulation in Robot Analyzer. Were there any discrepancies?

Key Takeaways:

- The experiment demonstrated how to control the movement of a robotic manipulator using Arduino.

- You learned how to program joint control for precise movements and simulate robotic movements using software.
- Hands-on practice with Arduino and servo motors helped develop practical skills in robotics programming.

References:

1. **Robot Analyzer Software Documentation**
2. **Arduino Servo Motor Library Documentation**
3. **Basic Robotics and Inverse Kinematics Theory**

प्रयोग क्रमांक 9

शीर्षक: Arduino/Robot प्रोग्रामिंग का उपयोग करके रोबोटिक मैनिपुलेटर का नियंत्रण

उद्देश्य:

1. रोबोटिक मैनिपुलेशन और नियंत्रण प्रणाली की मूल बातें समझना।
2. Arduino और रोबोट प्रोग्रामिंग का उपयोग करके एक रोबोटिक मैनिपुलेटर को प्रोग्राम और नियंत्रित करना।
3. Arduino का उपयोग करके रोबोटिक आर्म्स और मैनिपुलेटर्स के लिए बुनियादी मूवमेंट्स और जॉइंट नियंत्रण को लागू करना।
4. Robot Analyzer सॉफ्टवेयर का उपयोग करके विभिन्न रोबोटिक मैनिपुलेटर मूवमेंट्स का परीक्षण और सिमुलेशन करना।

अपकरण/उपकरण आवश्यक:

1. Arduino UNO (या कोई Arduino-संगत बोर्ड)
2. सर्वो मोटर (मैनिपुलेटर के जोड़ों को नियंत्रित करने के लिए)
3. रोबोटिक मैनिपुलेटर किट (कम से कम 2-3 डिग्री ऑफ फ्रीडम के साथ)
4. पावर सप्लाई (Arduino और सर्वो के लिए)
5. जंपर तार (कनेक्शन के लिए)
6. ब्रेडबोर्ड (सर्किट सेटअप के लिए)
7. Robot Analyzer सॉफ्टवेयर (रोबोट मूवमेंट्स का सिमुलेशन करने के लिए)
8. पीसी (Arduino IDE के साथ)

9. यूएसबी केबल (Arduino को पीसी से कनेक्ट करने के लिए)

प्री-लैब थ्योरी:

रोबोटिक मैनिपुलेटर एक यांत्रिक उपकरण है, जिसका उपयोग उठाने, छांटने और असेंबली जैसे विभिन्न कार्यों को करने के लिए किया जाता है। यह आमतौर पर कई जोड़ों (जैसे, सर्वो मोटर्स) से बना होता है, जो नियंत्रित तरीके से चलते हैं। इस प्रयोग में, Arduino का उपयोग करके एक रोबोटिक मैनिपुलेटर की मूवमेंट को नियंत्रित किया जाएगा।

मुख्य अवधारणाएँ:

- **डिग्री ऑफ फ्रीडम (DOF):** यह एक रोबोटिक आर्म द्वारा की जा सकने वाली स्वतंत्र मूवमेंट्स की संख्या है। सामान्य रोबोटिक मैनिपुलेटर में 2 या 3 DOF होते हैं।
 - **सर्वो मोटर्स:** रोबोटिक मैनिपुलेटर्स में जोड़ों के कोणों को सटीक रूप से नियंत्रित करने के लिए उपयोग किया जाता है।
 - **इन्वर्स काइनेमैटिक्स:** वह गणितीय प्रक्रिया, जिसमें एंड-इफेक्टर (जैसे, गिपर) को किसी विशिष्ट बिंदु पर स्थान देने के लिए आवश्यक जोड़ों के कोण निर्धारित किए जाते हैं।
 - **Robot Analyzer सॉफ्टवेयर:** यह रोबोटिक आर्म मूवमेंट्स और कॉन्फिगरेशन को मॉडल और विज़ुअलाइज़ करने का एक सिमुलेशन टूल है।
-

प्रक्रिया:

चरण 1: हार्डवेयर सेटअप

1. सर्वो मोटर्स को रोबोटिक मैनिपुलेटर किट से कनेक्ट करें, यह सुनिश्चित करते हुए कि प्रत्येक मोटर मैनिपुलेटर के एक विशिष्ट जॉइंट को नियंत्रित करती है।
2. प्रत्येक सर्वो मोटर को Arduino बोर्ड के उपयुक्त PWM पिन से जोड़ें।
3. सुनिश्चित करें कि Arduino बोर्ड को USB केबल के माध्यम से पीसी से जोड़ा गया है।
4. ब्रेडबोर्ड और जंपर तारों का उपयोग करके Arduino और सर्वो मोटर्स को पावर देने के लिए कनेक्शन पूरा करें।

चरण 2: रोबोट प्रोग्रामिंग (Arduino कोड)

1. अपने पीसी पर Arduino IDE खोलें।
2. रोबोटिक मैनिपुलेटर को नियंत्रित करने के लिए कोड लिखें या लोड करें। एक बुनियादी उदाहरण:

```
#include <Servo.h>
```

```
Servo servo1; // जॉइंट 1 के लिए एक सर्वो ऑब्जेक्ट बनाएं
```

```
Servo servo2; // जॉइंट 2 के लिए एक सर्वो ऑब्जेक्ट बनाएं
```

```
void setup() {
```

```
    servo1.attach(9); // सर्वो1 को Arduino के पिन 9 से जोड़ें
```

```
    servo2.attach(10); // सर्वो2 को Arduino के पिन 10 से जोड़ें
```

```
}
```

```
void loop() {
```

```
    // सर्वो1 को 0 से 90 डिग्री तक ले जाएं
```

```
    for (int pos = 0; pos <= 90; pos++) {
```

```
        servo1.write(pos);
```

```
        delay(15);
```

```
    }
```

```
    // सर्वो2 को 0 से 90 डिग्री तक ले जाएं
```

```
    for (int pos = 0; pos <= 90; pos++) {
```

```
        servo2.write(pos);
```

```
        delay(15);
```

```
    }
```

```
    // सर्वो पोजीशन रीसेट करें
```

```
    for (int pos = 90; pos >= 0; pos--) {
```

```
        servo1.write(pos);
```

```
        delay(15);
```

```
    }
```

```
    for (int pos = 90; pos >= 0; pos--) {
```

```
        servo2.write(pos);
```

```
        delay(15);
```

```
    }
```

```
}
```

3. Arduino IDE का उपयोग करके कोड को Arduino बोर्ड में अपलोड करें।
4. सुनिश्चित करें कि मैनिपुलेटर का मूवमेंट प्रोग्राम किए गए कोणों के अनुसार है।

चरण 3: Robot Analyzer सॉफ्टवेयर का उपयोग करके सिमुलेशन

1. अपने पीसी पर Robot Analyzer खोलें।
2. एक साधारण रोबोटिक मैनिपुलेटर मॉडल (2-DOF या 3-DOF) लोड करें।
3. Arduino में प्रोग्राम किए गए जॉइंट कोण (जैसे, जॉइंट 1 के लिए 0-90 डिग्री और जॉइंट 2 के लिए 0-90 डिग्री) दर्ज करें।
4. सिमुलेशन वातावरण में मैनिपुलेटर की मूवमेंट देखें और अपने कोड की सटीकता की पुष्टि करें।

चरण 4: प्रयोग और समायोजन

1. विभिन्न मूवमेंट्स और जॉइंट संयोजनों का परीक्षण करें और मैनिपुलेटर के व्यवहार का विश्लेषण करें।
2. मैनिपुलेटर का उपयोग करके एक साधारण कार्य (जैसे, किसी वस्तु को स्थानांतरित करना) करें।
3. सुचारु संचालन के लिए कोड या हार्डवेयर कनेक्शन को समायोजित करें।

प्रेक्षण:

- मैनिपुलेटर की मूवमेंट या पोजिशन में किसी भी अनियमितता को नोट करें।
- मैनिपुलेटर को उसके लक्ष्य स्थान तक पहुंचने में लगने वाले समय को रिकॉर्ड करें।
- विश्लेषण करें कि मैनिपुलेटर कितनी सटीकता से प्रोग्राम किए गए कमांड का पालन करता है।

पोस्ट-लैब प्रश्न:

1. मैनिपुलेटर की मूवमेंट पर डिग्री ऑफ फ्रीडम (DOF) की संख्या का क्या प्रभाव पड़ता है?
2. रोबोटिक मैनिपुलेटर को नियंत्रित करने में इन्वर्स काइनेमैटिक्स की भूमिका क्या है?
3. अधिक जोड़ों (जैसे, और अधिक सर्वो जोड़ना) को नियंत्रित करने के लिए आप Arduino कोड को कैसे संशोधित करेंगे?
4. मैनिपुलेटर को प्रोग्राम करने में आपने किन चुनौतियों का सामना किया और उन्हें कैसे हल किया?
5. वास्तविक मैनिपुलेटर की मूवमेंट और Robot Analyzer सिमुलेशन की तुलना करें। क्या कोई अंतर पाया गया?

मुख्य निष्कर्ष:

- इस प्रयोग में Arduino का उपयोग करके रोबोटिक मैनिपुलेटर की मूवमेंट को नियंत्रित करना सीखा।

- प्रोग्रामिंग और सिमुलेशन से रोबोटिक मूवमेंट्स को समझने और नियंत्रित करने की व्यावहारिक कौशल विकसित हुई।
- Servo मोटर्स और सिमुलेशन टूल्स का उपयोग करके सटीक मूवमेंट को लागू करना सीखा।

Experiment No-10

Title: Do it yourself (DIY) experiments (Students should take the real-world issue and they have to think, decide and do things independently)

प्रयोग क्रमांक 10:

शीर्षक: इसे स्वयं करें (DIY) प्रयोग (छात्रों को वास्तविक दुनिया के मुद्दे को लेना चाहिए और उन्हें स्वतंत्र रूप से सोचना, निर्णय लेना और कार्य करना होगा)

Maulana Azad National Institute of Technology Bhopal

Department of Mechanical Engineering



Lab Manual

Automation and Robotics Lab-II

M. Tech (Automation and Robotics)

Automation and Robotics Lab-II

List of Experiments

1. Geometrical Modeling Software — Basics
2. Starting with Geometrical Modeling Software — Various Draw Commands
3. 2D Solid Models Creation Using Geometrical Modeling Software (CREO)
4. 3D Solid Models Creation Using Geometrical Modeling Software (CREO)
5. Assembling of CAD Parts in CREO
6. Process Simulation for CNC Turning-Based Subtractive Manufacturing Processes Using Open-Source Software
7. Process Simulation for CNC Milling-Based Subtractive Manufacturing Processes Using Open-Source Software
8. Process Simulation for Solid-Based Additive Manufacturing Processes Using Open-Source Software
9. Process Simulation for Powder-Based Additive Manufacturing Processes Using Open-Source Software
10. Robot Simulation Using Open-Source Software

स्वचालन और रोबोटिक्स लैब II

प्रयोगों की सूची

1. ज्यामितीय मॉडलिंग सॉफ्टवेयर (Geometrical Modeling Software) — Basic Concepts
2. ज्यामितीय मॉडलिंग सॉफ्टवेयर के साथ शुरुआत — अलग-अलग Draw Commands
3. CREO सॉफ्टवेयर का उपयोग करके 2D Solid Models बनाना
4. CREO सॉफ्टवेयर का उपयोग करके 3D Solid Models बनाना
5. CREO में CAD Parts को Assemble करना
6. Open-Source सॉफ्टवेयर का उपयोग करके CNC Turning-Based Subtractive Manufacturing Process का Simulation
7. Open-Source सॉफ्टवेयर का उपयोग करके CNC Milling-Based Subtractive Manufacturing Process का Simulation
8. Open-Source सॉफ्टवेयर का उपयोग करके Solid-Based Additive Manufacturing Process का Simulation
9. Open-Source सॉफ्टवेयर का उपयोग करके Powder-Based Additive Manufacturing Process का Simulation
10. Open-Source सॉफ्टवेयर का उपयोग करके Robot का Simulation

Experiment No. 1

Title: Introduction to Geometrical Modelling Software — Basics

1. Introduction and Overview

1.1 Geometrical Modelling Software Overview

Geometrical modelling software, such as SolidWorks, Creo, or AutoCAD, enables designers and engineers to create, modify, analyze, and optimize designs digitally. These tools facilitate precision and efficiency in conceptualizing mechanical components, assemblies, and other structures. They are foundational for product design, engineering simulations, and manufacturing processes.

Key functionalities of geometrical modelling software include:

1. Sketch creation using basic 2D commands.
2. Transitioning 2D sketches to 3D models.
3. Defining constraints and dimensions for accuracy.
4. Tools for visualization, assembly, and simulation.

1.2 Importance in Modern Engineering

The integration of CAD tools in engineering design is essential for:

1. Enhancing productivity by reducing design time.
 2. Allowing precise visualization of models before production.
 3. Supporting analysis and validation of designs.
 4. Enabling easy modification and iterative improvements.
-

2. Methodologies to Perform the Experiment

2.1 Setup and Initialization

1. **Software Launch:** Open the selected software (e.g., SolidWorks or Creo). Ensure the system is properly configured with the necessary plugins.
2. **Interface Familiarization:**
 - Identify key areas such as the toolbar, workspace, and property manager.
 - Learn navigation commands such as zoom, pan, and rotate.
3. **Set Units:** Configure the default unit system (e.g., millimetres, inches) based on the requirements.

2.2 Exploring Basic Tools

1. **Drawing Tools:**
 - Line, rectangle, circle, and arc commands.

- Techniques for selecting and combining basic shapes.
- 2. **Editing Tools:**
 - Trim, extend, offset, and fillet tools.
 - Methods to modify sketches to meet desired specifications.
- 3. **Constraints and Dimensions:**
 - Apply geometric constraints such as parallelism, perpendicular, and tangency.
 - Use dimensioning tools to define precise measurements.
- 4. **Saving and Exporting:**
 - Save the project in both native and neutral file formats (e.g., .sldprt, .stp).

2.3 Practice Session

1. Create basic shapes, such as a rectangle with specified dimensions.
2. Modify the shape by adding an arc, fillets, and chamfers.
3. Save the completed sketch for later use.

3. Expected Outcomes

1. Understanding the basic interface of geometrical modelling software.
2. Ability to use fundamental drawing and editing tools.
3. Familiarity with constraints and dimensioning.
4. Capability to save and export 2D sketches for further use.

4. Possibilities of Deviations from Expected Outcomes

1. **Unfamiliarity with Tools:** Initial difficulty in identifying and using certain commands.
2. **Dimensioning Errors:** Incorrect application of constraints may lead to unsatisfactory sketches.
3. **Interface Customization:** Variations in software settings or shortcuts might differ from standard configurations.

5. Reasons for Observed Deviations

1. **Lack of Practice:** Limited hands-on experience with the software.
 2. **System Configuration Issues:** Software settings not optimized for user requirements.
 3. **Human Errors:** Mistakes in following step-by-step instructions.
-

6. Thought-Provoking Questions

1. What are the advantages of using constraints in a 2D sketch?
 2. How does the choice of the unit system affect the accuracy of a design?
 3. Why is it important to save projects in both native and neutral formats?
 4. How would you modify your sketch if a design specification changes?
 5. What are the key differences between 2D and 3D modelling tools?
-

प्रयोग संख्या 1

शीर्षक: ज्यामितीय मॉडलिंग सॉफ्टवेयर का परिचय – मूल बातें

1. परिचय और अवलोकन

1.1 ज्यामितीय मॉडलिंग सॉफ्टवेयर का अवलोकन

ज्यामितीय मॉडलिंग सॉफ्टवेयर, जैसे SolidWorks, Creo, या AutoCAD, डिज़ाइनरों और इंजीनियरों को डिज़ाइन को डिजिटल रूप में बनाने, संशोधित करने, विश्लेषण करने और अनुकूलित करने में सक्षम बनाता है। ये उपकरण यांत्रिक घटकों, असेंबली और अन्य संरचनाओं के संकल्पना में सटीकता और दक्षता प्रदान करते हैं। ये उत्पाद डिज़ाइन, इंजीनियरिंग सिमुलेशन और निर्माण प्रक्रियाओं के लिए मौलिक होते हैं। ज्यामितीय मॉडलिंग सॉफ्टवेयर की प्रमुख कार्यक्षमताओं में शामिल हैं:

1. बुनियादी 2D आदेशों का उपयोग करके स्केच बनाना।
2. 2D स्केच को 3D मॉडल में बदलना।
3. सटीकता के लिए प्रतिबंध और आयाम निर्धारित करना।
4. दृश्यात्मकता, असेंबली और सिमुलेशन के लिए उपकरण।

1.2 आधुनिक इंजीनियरिंग में महत्व

इंजीनियरिंग डिज़ाइन में CAD उपकरणों का एकीकरण आवश्यक है:

1. डिज़ाइन समय को कम करके उत्पादकता बढ़ाना।
2. उत्पादन से पहले मॉडल का सटीक दृश्यावलोकन प्रदान करना।
3. डिज़ाइन के विश्लेषण और सत्यापन में सहायता करना।
4. आसान संशोधन और पुनरावृत्त सुधार की सुविधा प्रदान करना।

2. प्रयोग करने की विधियाँ

2.1 सेटअप और आरंभिकरण

1. सॉफ्टवेयर लॉन्च: चयनित सॉफ्टवेयर (जैसे, SolidWorks या Creo) खोलें। सुनिश्चित करें कि सिस्टम आवश्यक प्लगइन्स के साथ ठीक से कॉन्फ़िगर किया गया है।
2. इंटरफ़ेस से परिचित होना:
 - o टूलबार, कार्यक्षेत्र और प्रॉपर्टी मैनेजर जैसे प्रमुख क्षेत्रों की पहचान करें।
 - o ज़ूम, पैन और घुमाने जैसे नेविगेशन आदेश सीखें।
3. इकाइयाँ सेट करें: आवश्यकताओं के आधार पर डिफ़ॉल्ट इकाई प्रणाली (जैसे, मिलीमीटर, इंच) को कॉन्फ़िगर करें।

2.2 बुनियादी उपकरणों का अन्वेषण

1. ड्राइंग उपकरण:
 - o रेखा, आयत, वृत्त, और आर्क आदेश।
 - o बुनियादी रूपों को चुनने और संयोजित करने की तकनीक।
2. संपादन उपकरण:
 - o ट्रिम, विस्तार, ऑफसेट, और फिलेट उपकरण।
 - o स्केच को वांछित विनिर्देशों के अनुसार संशोधित करने की विधियाँ।
3. प्रतिबंध और आयाम:
 - o समांतरता, लंबवतता, और टांगेंसी जैसे ज्यामितीय प्रतिबंध लागू करें।
 - o सटीक मापों को परिभाषित करने के लिए आयाम निर्धारण उपकरणों का उपयोग करें।
4. सहेजना और निर्यात करना:
 - o परियोजना को दोनों स्थानीय और न्यूट्रल फ़ाइल प्रारूपों में सहेजें (जैसे, .sldprt, .stp)।

2.3 अभ्यास सत्र

1. बुनियादी आकार बनाएं, जैसे कि निर्दिष्ट आयामों के साथ एक आयत।
 2. आकार को आर्क, फिलेट और चाम्फर जोड़कर संशोधित करें।
 3. पूरा हुआ स्केच बाद में उपयोग के लिए सहेजें।
-

3. अपेक्षित परिणाम

1. ज्यामितीय मॉडलिंग सॉफ्टवेयर के बुनियादी इंटरफ़ेस की समझ।
 2. बुनियादी ड्राइंग और संपादन उपकरणों का उपयोग करने की क्षमता।
 3. प्रतिबंधों और आयामों के साथ परिचित होना।
 4. 2D स्केच को सहेजने और निर्यात करने की क्षमता।
-

4. अपेक्षित परिणामों से विचलन की संभावनाएँ

1. उपकरणों से अपरिचितता: कुछ आदेशों का पहचानने और उपयोग करने में प्रारंभिक कठिनाई।
 2. आयाम निर्धारण त्रुटियाँ: प्रतिबंधों का गलत उपयोग असंतोषजनक स्केच उत्पन्न कर सकता है।
 3. इंटरफ़ेस अनुकूलन: सॉफ्टवेयर सेटिंग्स या शॉर्टकट्स में भिन्नताएँ मानक कॉन्फ़िगरेशन से भिन्न हो सकती हैं।
-

5. देखे गए विचलनों के कारण

1. अभ्यास की कमी: सॉफ्टवेयर के साथ सीमित अनुभव।
 2. सिस्टम कॉन्फिगरेशन समस्याएँ: सॉफ्टवेयर सेटिंग्स उपयोगकर्ता की आवश्यकताओं के लिए अनुकूलित नहीं हैं।
 3. मानव त्रुटियाँ: कदम-दर-कदम निर्देशों का पालन करते समय गलतियाँ।
-

6. सोचने के लिए प्रश्न

1. 2D स्केच में प्रतिबंधों का उपयोग करने के क्या लाभ हैं?
 2. इकाई प्रणाली का चयन डिज़ाइन की सटीकता को कैसे प्रभावित करता है?
 3. परियोजनाओं को स्थानीय और न्यूट्रल प्रारूपों में सहेजने का महत्व क्या है?
 4. यदि डिज़ाइन विनिर्देश बदलते हैं तो आप अपने स्केच को कैसे संशोधित करेंगे?
 5. 2D और 3D मॉडलिंग उपकरणों के बीच मुख्य अंतर क्या हैं?
-

Experiment No: 2

Title: Introduction to Geometrical Modelling Software - Understanding and Using Various Draw Commands in CREO Software

1. Introduction

1.1 Overview of Geometrical Modelling in CREO

CREO is a powerful parametric 3D CAD software widely used in product design and engineering for creating precise models of mechanical components, assemblies, and detailed drawings. The software provides an intuitive interface and a robust set of tools for creating both 2D sketches and 3D models.

1.2 Importance of Drawing Commands

The drawing commands in CREO form the foundation for creating accurate and complex geometrical shapes. Mastering these commands allows engineers and designers to:

- Sketch and modify 2D profiles quickly.
- Define constraints and dimensions for accurate modeling.
- Build the base for 3D models by extruding or revolving the sketches.

1.3 Objective of the Experiment To familiarize students with the various draw commands available in CREO and develop the skills required to create precise 2D sketches as the basis for 3D solid models.

2. Methodologies to Perform the Experiment

2.1 Software Setup and Initialization

1. **Launch CREO Software:** Open the application and create a new project file.
2. **Set Up Workspace:** Choose the appropriate units (e.g., mm or inches) for the design and select a plane (e.g., Front, Top, Right) to start the sketch.

2.2 Key Draw Commands in CREO Below are the primary draw commands with detailed instructions on their usage:

1. **Line Command:**
 - Select the Line tool from the Sketch tab.
 - Click to specify the starting point and drag the mouse to create a line segment.
 - Define the line's length and angle using the dimensioning tool.
2. **Rectangle Command:**
 - Choose the Rectangle tool.
 - Click and drag to create a rectangular shape.

- Use constraints (e.g., parallel or perpendicular) to ensure the desired alignment.
- 3. **Circle Command:**
 - Select the Circle tool and click to define the center point.
 - Drag outward to define the radius or input a specific value.
- 4. **Arc Command:**
 - Choose the Arc tool.
 - Define the start point, end point, and a point along the arc's curve to create the desired shape.
- 5. **Ellipse Command:**
 - Select the Ellipse tool.
 - Define the center and the major and minor axes.
- 6. **Spline Command:**
 - Click on the Spline tool to create freeform curves.
 - Place control points along the desired path, adjusting curvature as needed.
- 7. **Trim/Extend Command:**
 - Use the Trim tool to remove unnecessary segments of overlapping lines or arcs.
 - Use the Extend tool to lengthen a line to intersect with another entity.
- 8. **Fillet/Chamfer Command:**
 - Select Fillet to create rounded edges between two intersecting lines.
 - Use Chamfer to create a beveled edge.
- 9. **Mirror Command:**
 - Use the Mirror tool to replicate geometry across a defined axis of symmetry.
- 10. **Dimensioning Tool:**
 - Apply dimensions to control the size and position of the sketch entities. Use parametric relations to link dimensions dynamically.

2.3 Constraints in CREO

1. **Geometric Constraints:** Automatically applied constraints such as horizontal, vertical, and perpendicular alignments.
2. **Dimensional Constraints:** Define distances, angles, and sizes.
3. **Relations:** Use mathematical expressions to control relationships between dimensions.

2.4 Saving and Exporting

1. Save the sketch file as a .prt file for further 3D operations.

2. Export the file as a 2D drawing in formats such as .dxf or .dwg if needed.
-

3. Expected Outcomes

1. Familiarity with the CREO interface and tools.
 2. Ability to use various draw commands to create precise 2D sketches.
 3. Understanding of constraints and dimensions to control geometry effectively.
-

4. Common Errors and Troubleshooting

1. **Overlapping Geometry:** Ensure no redundant lines or arcs are present in the sketch.
 2. **Unconstrained Sketch:** Check for under-defined elements by enabling the Sketch Analysis tool.
 3. **Incorrect Units:** Verify the unit settings to match the design requirements.
-

5. Thought-Provoking Questions

1. How do geometric constraints help in maintaining the integrity of a sketch?
 2. Why is it important to fully constrain a sketch before converting it into a 3D model?
 3. What are the advantages of using parametric relations between dimensions?
 4. How can the Trim tool be used to optimize sketching efficiency?
 5. What challenges might arise while working with freeform curves such as splines?
-

6. Additional Notes and Tips

- Practice creating sketches on different planes to understand spatial orientation.
 - Experiment with the Undo and Redo commands to refine sketches efficiently.
 - Use keyboard shortcuts for frequently used tools to speed up the workflow.
-

7. Conclusion

This experiment introduces the foundational tools and techniques for sketching in CREO. Mastery of these commands is essential for progressing to advanced topics like 3D modeling, assembly creation, and simulation.

प्रयोग संख्या 2

शीर्षक: ज्यामितीय मॉडलिंग सॉफ्टवेयर का परिचय – CREO सॉफ्टवेयर में विभिन्न ड्रा आदेशों को समझना और उनका उपयोग करना

1. परिचय

1.1 CREO में ज्यामितीय मॉडलिंग का अवलोकन

CREO एक शक्तिशाली पैरामीट्रिक 3D CAD सॉफ्टवेयर है, जिसे उत्पाद डिज़ाइन और इंजीनियरिंग में यांत्रिक घटकों, असेंबली और विस्तृत ड्राइंग के सटीक मॉडल बनाने के लिए व्यापक रूप से उपयोग किया जाता है। यह सॉफ्टवेयर एक सहज इंटरफ़ेस और 2D स्केच और 3D मॉडल बनाने के लिए एक मजबूत उपकरण सेट प्रदान करता है।

1.2 ड्राइंग आदेशों का महत्व

CREO में ड्राइंग आदेश सटीक और जटिल ज्यामितीय आकार बनाने की नींव बनाते हैं। इन आदेशों में महारत हासिल करने से इंजीनियरों और डिज़ाइनरों को सक्षम बनाता है:

- त्वरित 2D प्रोफाइल बनाना और संशोधित करना।
- सटीक मॉडलिंग के लिए प्रतिबंधों और आयामों को परिभाषित करना।
- स्केचों को निष्कर्षित या घुमाकर 3D मॉडल के लिए आधार बनाना।

1.3 प्रयोग का उद्देश्य

CREO में उपलब्ध विभिन्न ड्रा आदेशों से छात्रों को परिचित करना और 3D ठोस मॉडलों के आधार के रूप में सटीक 2D स्केच बनाने के लिए आवश्यक कौशल विकसित करना।

2. प्रयोग करने की विधियाँ

2.1 सॉफ्टवेयर सेटअप और प्रारंभिककरण

1. CREO सॉफ्टवेयर लॉन्च करें: एप्लिकेशन खोलें और एक नया परियोजना फ़ाइल बनाएं।
2. कार्यस्थान सेट करें: डिज़ाइन के लिए उपयुक्त इकाइयाँ (जैसे, मिमी या इंच) चुनें और स्केच शुरू करने के लिए एक विमान (जैसे, फ्रंट, टॉप, राइट) का चयन करें।

2.2 CREO में प्रमुख ड्रा आदेश

नीचे मुख्य ड्रा आदेश दिए गए हैं, जिनके उपयोग के लिए विस्तृत निर्देश हैं:

3. लाइन आदेश:

- o Sketch टैब से लाइन उपकरण का चयन करें।
- o शुरुआत बिंदु निर्दिष्ट करने के लिए क्लिक करें और माउस को खींचकर एक रेखा खींचें।
- o आयाम उपकरण का उपयोग करके रेखा की लंबाई और कोण परिभाषित करें।

4. आयत आदेश:

- o आयत उपकरण का चयन करें।
- o क्लिक करें और खींचकर एक आयताकार आकार बनाएं।
- o वांछित संरेखण सुनिश्चित करने के लिए प्रतिबंधों (जैसे, समांतर या लंबवत) का उपयोग करें।

5. **वृत्त आदेश:**
 - o वृत्त उपकरण का चयन करें और केंद्र बिंदु परिभाषित करने के लिए क्लिक करें।
 - o त्रिज्या परिभाषित करने के लिए बाहर की ओर खींचें या एक विशिष्ट मान इनपुट करें।
6. **आर्क आदेश:**
 - o आर्क उपकरण का चयन करें।
 - o वांछित आकार बनाने के लिए शुरुआत बिंदु, समाप्ति बिंदु और आर्क की वक्र के साथ एक बिंदु परिभाषित करें।
7. **दीर्घवृत्त आदेश:**
 - o दीर्घवृत्त उपकरण का चयन करें।
 - o केंद्र और प्रमुख और गौण अक्षों को परिभाषित करें।
8. **स्प्लाइन आदेश:**
 - o स्प्लाइन उपकरण पर क्लिक करें ताकि आप फ्रीफॉर्म वक्र बना सकें।
 - o वांछित मार्ग के साथ नियंत्रण बिंदु रखें, और वक्रता को आवश्यकतानुसार समायोजित करें।
9. **ट्रिम/एक्सटेंड आदेश:**
 - o अधिकारी रेखाओं या आर्कों के अव्यावहारिक खंडों को हटाने के लिए ट्रिम उपकरण का उपयोग करें।
 - o एक रेखा को दूसरी इकाई से मिलाने के लिए एक्सटेंड उपकरण का उपयोग करें।
10. **फिलेट/चम्फर आदेश:**
 - o दो परस्पर रेखाओं के बीच गोल कोने बनाने के लिए फिलेट का चयन करें।
 - o चम्फर का उपयोग करके एक बेवल एज बनाएं।
11. **मिरर आदेश:**
 - o संरचना को परिभाषित सीमा रेखा पर परावर्तित करने के लिए मिरर उपकरण का उपयोग करें।
12. **आयाम उपकरण:**
 - o स्केच की इकाई आकार और स्थिति को नियंत्रित करने के लिए आयाम लागू करें। आयामों को गतिशील रूप से जोड़ने के लिए पैरामीट्रिक संबंधों का उपयोग करें।

2.3 CREO में प्रतिबंध

13. **ज्यामितीय प्रतिबंध:** स्वचालित रूप से लागू प्रतिबंध जैसे क्षैतिज, लंबवत, और लंबवत संरेखण।
14. **आयामात्मक प्रतिबंध:** दूरी, कोण और आकारों को परिभाषित करें।
15. **संबंध:** आयामों के बीच संबंधों को नियंत्रित करने के लिए गणितीय अभिव्यक्तियों का उपयोग करें।

2.4 सहेजना और निर्यात करना

16. स्केच फ़ाइल को .prt फ़ाइल के रूप में सहेजें ताकि 3D संचालन जारी रख सकें।
 17. आवश्यकता के अनुसार 2D ड्राइंग के रूप में फ़ाइल को .dxf या .dwg जैसे प्रारूपों में निर्यात करें।
-

3. अपेक्षित परिणाम

1. CREO इंटरफ़ेस और उपकरणों से परिचित होना।
 2. विभिन्न ड्रॉ आदेशों का उपयोग करके सटीक 2D स्केच बनाने की क्षमता।
 3. ज्यामिती को प्रभावी रूप से नियंत्रित करने के लिए प्रतिबंधों और आयामों की समझ।
-

4. सामान्य त्रुटियाँ और समाधान

1. **संपन्न ज्यामिति:** सुनिश्चित करें कि स्केच में कोई अतिरिक्त रेखाएँ या आर्क नहीं हैं।
 2. **अप्रतिबंधित स्केच:** Sketch Analysis उपकरण का उपयोग करके अपरिभाषित तत्वों की जांच करें।
 3. **गलत इकाइयाँ:** डिज़ाइन आवश्यकताओं से मेल खाने के लिए इकाई सेटिंग्स की पुष्टि करें।
-

5. सोचने के लिए प्रश्न

1. ज्यामितीय प्रतिबंध कैसे स्केच की अखंडता बनाए रखने में मदद करते हैं?
 2. क्यों जरूरी है कि एक स्केच को 3D मॉडल में परिवर्तित करने से पहले पूरी तरह से प्रतिबंधित किया जाए?
 3. आयामों के बीच पैरामीट्रिक संबंधों का उपयोग करने के लाभ क्या हैं?
 4. ट्रिम उपकरण का उपयोग करके स्केचिंग दक्षता को कैसे अनुकूलित किया जा सकता है?
 5. फ्रीफॉर्म वक्र (जैसे स्प्लाइनों) के साथ काम करते समय कौन सी चुनौतियाँ सामने आ सकती हैं?
-

6. अतिरिक्त नोट्स और सुझाव

- विभिन्न विमानों पर स्केच बनाने का अभ्यास करें ताकि आप स्थानिक अभिविन्यास को समझ सकें।
 - स्केच को प्रभावी रूप से परिष्कृत करने के लिए Undo और Redo आदेशों का अभ्यास करें।
 - बारंबार उपयोग किए जाने वाले उपकरणों के लिए कीबोर्ड शॉर्टकट्स का उपयोग करें ताकि कार्यप्रवाह गति बढ़ सके।
-

7. निष्कर्ष

यह प्रयोग CREO में स्केचिंग के लिए बुनियादी उपकरणों और तकनीकों से परिचित कराता है। इन आदेशों में महारत हासिल करना 3D मॉडलिंग, असेंबली निर्माण और सिमुलेशन जैसे उन्नत विषयों के लिए आवश्यक है।

Experiment No. 3

Title: 2D Solid Models Creation Using Geometrical Modelling Software - CREO

1. Introduction and Physical Information about the Platform

1.1 Overview of 2D Solid Modelling

2D solid modelling refers to the creation of two-dimensional representations of solid objects using computer-aided design (CAD) software. These models serve as a foundation for creating 3D designs and can include profiles, sketches, and detailed views essential for engineering and manufacturing purposes. In this experiment, we use CREO software to explore the creation of 2D solid models.

1.2 CREO Software

CREO is a powerful CAD tool that supports 2D and 3D modelling. It provides a variety of tools for creating accurate, precise, and highly detailed 2D sketches and models. Features such as constraints, dimensions, and parametric modelling make it an ideal choice for designing complex parts and assemblies.

2. Objectives

1. To familiarize students with 2D sketching and modelling tools in CREO.
 2. To understand the creation of 2D profiles that serve as a base for 3D models.
 3. To apply dimensional constraints and geometric relations in 2D modelling.
 4. To simulate the transition from 2D sketches to 3D designs.
-

3. Tools and Commands Used

1. **Sketch Tools:** Line, Circle, Rectangle, Arc, Spline, Fillet.
 2. **Constraint Tools:** Coincident, Horizontal, Vertical, Tangent, Symmetric.
 3. **Dimensioning:** Linear, Angular, Radial.
 4. **Editing Tools:** Trim, Extend, Offset, Mirror, Pattern.
 5. **Feature Tools:** Extrude (to verify the transition to 3D).
-

4. Procedure

4.1 Setup and Initialization

1. Launch CREO Parametric and create a new part file.
2. Select the appropriate measurement unit (e.g., mm).

3. Open the sketch mode by selecting a plane (Front, Top, or Right plane).

4.2 Creation of 2D Models

1. **Start with Basic Shapes:** Use the Line, Circle, and Rectangle tools to create the basic outline of the part.
2. **Apply Constraints:** Ensure that all lines and curves are constrained properly. For example:
 - Horizontal or vertical constraints for lines.
 - Tangency between arcs and lines.
 - Symmetry for mirroring elements.
3. **Dimension the Sketch:** Use the dimensioning tools to add precise measurements to the sketch.
4. **Modify and Refine:** Use tools like Trim, Extend, and Fillet to finalize the shape.
5. **Mirror and Pattern:** Create repetitive elements efficiently using the mirror and pattern tools.

4.3 Transition to Solid Model

1. Save the sketch.
 2. Exit the sketch mode and use the Extrude tool to verify the transition from 2D to 3D.
 3. Adjust extrusion settings to visualize the solid model.
-

5. Expected Outcomes

1. Students will create accurate 2D profiles using CREO.
 2. They will apply appropriate constraints and dimensions to ensure design intent.
 3. The final 2D sketches will be ready for conversion into 3D models.
 4. Students will understand how 2D models serve as a foundation for complex designs.
-

6. Possibilities of Deviations from Ideal Outputs

1. **Incorrect Constraints:** Over-constrained or under-constrained sketches can lead to errors.
 2. **Dimensional Inconsistencies:** Failure to apply precise dimensions can affect model accuracy.
 3. **Geometric Errors:** Misalignment or overlapping entities in the sketch can result in errors during extrusion.
-

7. Reasons for Observed Deviations

1. Lack of familiarity with sketching tools and commands.
2. Errors in applying constraints or dimensions.

3. Insufficient planning before creating the sketch.
-

8. Thought-Provoking Questions

1. What are the key differences between 2D sketches and 3D models in CREO?
 2. How do geometric constraints influence the accuracy of 2D models?
 3. What steps can you take to troubleshoot an over-constrained sketch?
 4. How does the selection of a sketch plane affect the final model?
 5. What additional tools or features in CREO can enhance 2D sketching?
-

9. Conclusion

The experiment demonstrated the creation of 2D solid models using CREO software. Students learned to utilize various sketch tools, apply constraints, and dimension their designs accurately. This hands-on exercise laid the groundwork for transitioning to 3D modelling, highlighting the importance of a robust 2D foundation in design engineering.

10. References

1. CREO Parametric User Guide.
2. PTC Learning Portal (<https://www.ptc.com/>).
3. Relevant lecture notes and design manuals.

प्रयोग संख्या: 3

शीर्षक: ज्यामितीय मॉडलिंग सॉफ्टवेयर का उपयोग करके 2D ठोस मॉडल निर्माण - CREO

1. परिचय और प्लेटफॉर्म की भौतिक जानकारी

1.1 2D ठोस मॉडलिंग का अवलोकन

2D ठोस मॉडलिंग का अर्थ है ठोस वस्तुओं के दो-आयामी प्रतिनिधित्वों का निर्माण, जिसे कंप्यूटर-एडेड डिज़ाइन (CAD) सॉफ्टवेयर का उपयोग करके किया जाता है। ये मॉडल 3D डिज़ाइनों के निर्माण के लिए आधार के रूप में कार्य करते हैं और इसमें इंजीनियरिंग और निर्माण उद्देश्यों के लिए आवश्यक प्रोफाइल, स्केच और विस्तृत दृश्य शामिल हो सकते हैं। इस प्रयोग में, हम CREO सॉफ्टवेयर का उपयोग करके 2D ठोस मॉडल बनाने का अन्वेषण करेंगे।

1.2 CREO सॉफ्टवेयर

CREO एक शक्तिशाली CAD उपकरण है जो 2D और 3D मॉडलिंग का समर्थन करता है। यह सटीक, विस्तृत और उच्च गुणवत्ता वाले 2D स्केच और मॉडल बनाने के लिए विभिन्न उपकरण प्रदान करता है। कन्स्ट्रेंट्स, डाइमेंशन्स और पैरामीट्रिक मॉडलिंग जैसी विशेषताएँ इसे जटिल भागों और असेंबलियों को डिज़ाइन करने के लिए आदर्श बनाती हैं।

2. उद्देश्य

- छात्रों को CREO में 2D स्केचिंग और मॉडलिंग उपकरणों से परिचित कराना।
 - 3D मॉडल्स के लिए आधार के रूप में 2D प्रोफाइल्स का निर्माण समझना।
 - 2D मॉडलिंग में डाइमेंशनल कन्स्ट्रेंट्स और ज्यामितीय रिलेशनन्स का उपयोग करना।
 - 2D स्केच से 3D डिज़ाइनों में संक्रमण का अनुकरण करना।
-

3. उपकरण और आदेश

- स्केच उपकरण:** रेखा, वृत्त, आयत, आर्क, स्प्लाइन, फिलेट।
 - कन्स्ट्रेंट उपकरण:** समान, क्षैतिज, ऊर्ध्वाधर, स्पर्श, सममित।
 - माप:** रेखिक, कोणीय, रेडियल।
 - संपादन उपकरण:** ट्रिम, एक्सटेंड, ऑफसेट, मिरर, पैटर्न।
 - फीचर उपकरण:** एक्सट्रूड (3D में संक्रमण की पुष्टि के लिए)।
-

4. प्रक्रिया

4.1 सेटअप और प्रारंभिककरण

1. CREO पैरामीट्रिक लॉन्च करें और एक नया पार्ट फ़ाइल बनाएं।
2. उपयुक्त माप यूनिट (जैसे मिमी) चुनें।
3. स्केच मोड को खोलने के लिए एक विमान (सामना, शीर्ष, या दायां विमान) चुनें।

4.2 2D मॉडल का निर्माण

1. बुनियादी आकारों से शुरुआत करें: रेखा, वृत्त, और आयत उपकरणों का उपयोग करके भाग का बुनियादी रूपरेखा बनाएं।
2. कन्स्ट्रेंट्स लागू करें: सुनिश्चित करें कि सभी रेखाएं और वक्र ठीक से कन्स्ट्रेंट की गई हैं।
उदाहरण के लिए:
 - रेखाओं के लिए क्षैतिज या ऊर्ध्वाधर कन्स्ट्रेंट्स।
 - आर्क और रेखाओं के बीच स्पर्श।
 - मिररिंग तत्वों के लिए सममिति।
3. स्केच का माप करें: स्केच में सटीक माप जोड़ने के लिए माप उपकरणों का उपयोग करें।
4. संशोधित और परिष्कृत करें: ट्रिम, एक्सटेंड, और फिलेट जैसे उपकरणों का उपयोग करके रूपरेखा को अंतिम रूप दें।
5. मिरर और पैटर्न: मिरर और पैटर्न उपकरणों का उपयोग करके पुनरावृत्त तत्वों को कुशलता से बनाएं।

4.3 ठोस मॉडल में संक्रमण

1. स्केच को सेव करें।
2. स्केच मोड से बाहर निकलें और 2D से 3D में संक्रमण की पुष्टि करने के लिए एक्सट्रूड उपकरण का उपयोग करें।
3. ठोस मॉडल को दृश्य रूप में देखने के लिए एक्सट्रूड सेटिंग्स को समायोजित करें।

5. अपेक्षित परिणाम

1. छात्र CREO का उपयोग करके सटीक 2D प्रोफाइल बनाएंगे।
 2. वे डिज़ाइन इरादे को सुनिश्चित करने के लिए उपयुक्त कन्स्ट्रेंट्स और माप लागू करेंगे।
 3. अंतिम 2D स्केच 3D मॉडल में रूपांतरित करने के लिए तैयार होंगे।
 4. छात्र यह समझेंगे कि कैसे 2D मॉडल जटिल डिज़ाइनों के लिए आधार के रूप में कार्य करते हैं।
-

6. आदर्श परिणाम से विचलन की संभावनाएँ

1. गलत कन्स्ट्रेंट्स: अत्यधिक या अपर्याप्त कन्स्ट्रेंट्स वाली स्केच से त्रुटियाँ हो सकती हैं।
 2. माप में असंगतताएँ: सटीक माप लागू न करने से मॉडल की सटीकता प्रभावित हो सकती है।
 3. ज्यामितीय त्रुटियाँ: स्केच में मिसअलाइनमेंट या ओवरलैपिंग तत्वों से एक्सट्रूजन के दौरान त्रुटियाँ हो सकती हैं।
-

7. देखी गई विचलनों के कारण

1. स्केचिंग उपकरणों और आदेशों से अपरिचितता।
 2. कन्स्ट्रेंट्स या माप लागू करने में त्रुटियाँ।
 3. स्केच बनाने से पहले अपर्याप्त योजना।
-

8. विचारणीय प्रश्न

1. CREO में 2D स्केच और 3D मॉडल्स के बीच प्रमुख अंतर क्या हैं?
 2. ज्यामितीय कन्स्ट्रेंट्स 2D मॉडल्स की सटीकता को कैसे प्रभावित करते हैं?
 3. अत्यधिक कन्स्ट्रेंट वाली स्केच को ठीक करने के लिए आप कौन से कदम उठा सकते हैं?
 4. एक स्केच विमान का चयन अंतिम मॉडल को कैसे प्रभावित करता है?
 5. CREO में 2D स्केचिंग को बेहतर बनाने के लिए कौन से अतिरिक्त उपकरण या विशेषताएँ हैं?
-

9. निष्कर्ष

इस प्रयोग में CREO सॉफ्टवेयर का उपयोग करके 2D ठोस मॉडल बनाने का प्रदर्शन किया गया। छात्रों ने विभिन्न स्केच उपकरणों का उपयोग करना, कन्स्ट्रेंट्स लागू करना और माप को सटीक रूप से परिभाषित करना सीखा। यह अभ्यास 3D मॉडलिंग में संक्रमण के लिए आधार तैयार करता है, जो डिज़ाइन इंजीनियरिंग में मजबूत 2D नींव के महत्व को उजागर करता है।

10. संदर्भ

1. CREO पैरामीट्रिक उपयोगकर्ता गाइड।
2. PTC लर्निंग पोर्टल (<https://www.ptc.com/>)।
3. संबंधित व्याख्यान नोट्स और डिज़ाइन मैनुअल्स।
- 4.

Experiment No. 4

Title: 3D Solid Models Creation Using Geometrical Modelling Software - CREO

1. Introduction and Overview

1.1 Objective

This experiment aims to introduce students to 3D solid modeling concepts using CREO software. The students will learn to create basic and advanced 3D models, exploring the use of features such as extrude, revolve, sweep, and loft. The session will provide hands-on practice to develop proficiency in 3D modeling techniques.

1.2 Overview of CREO

CREO is a powerful 3D CAD (Computer-Aided Design) software widely used for product design and engineering. It enables users to design, simulate, and validate 3D models, making it an essential tool for modern engineering and product development. CREO provides a user-friendly interface, robust modeling features, and parametric capabilities, which allow designers to create precise and complex solid models efficiently.

2. Software Environment

- **Software Used:** CREO Parametric
 - **System Requirements:**
 - Operating System: Windows 10 or later
 - Processor: Intel Core i5 or equivalent
 - RAM: Minimum 8 GB
 - Graphics Card: NVIDIA or AMD GPU with OpenGL support
-

3. Methodologies to Perform the Experiment

3.1 Setup and Initialization

1. **Launch CREO Software:** Open the CREO application on your system.
2. **Create a New Part File:** Go to the "File" menu, select "New," and choose "Part" from the options. Assign a name to your file and click "OK."
3. **Set Units:** Ensure the correct unit system (e.g., mm, cm, inches) is selected under the "Setup" tab.

3.2 Creating Basic 3D Models

1. **Sketch Creation:**
 - Open the sketch environment by selecting a plane (e.g., Top, Front, or Right plane).
 - Use the following sketch tools to create a 2D profile:

- Line
- Circle
- Rectangle
- Arc
- Spline

- Constrain the sketch by defining dimensions and geometric relationships (e.g., parallel, perpendicular).

2. **Extrude Feature:**

- Select the sketch and use the "Extrude" command to convert the 2D profile into a 3D solid.
- Specify the extrusion depth and direction.

3. **Revolve Feature:**

- Create a sketch with an axis of revolution.
- Use the "Revolve" command to generate a 3D solid by revolving the sketch around the axis.

4. **Sweep Feature:**

- Create a profile and a path (e.g., spline or curve).
- Use the "Sweep" command to create a solid by sweeping the profile along the path.

5. **Loft Feature:**

- Create multiple sketches on parallel or inclined planes.
- Use the "Loft" command to generate a solid by connecting the sketches smoothly.

3.3 Editing and Modifying Models

1. Use tools such as "Fillet," "Chamfer," and "Shell" to refine the model's edges and surfaces.
2. Apply Boolean operations (e.g., union, subtract, intersect) to combine or modify solid bodies.

3.4 Saving the File

- Save your model using the "Save" option in the "File" menu. Ensure the file is stored in a dedicated project folder for future use.

4. Expected Outcomes

1. Ability to create 3D models with basic and advanced features such as extrude, revolve, sweep, and loft.
2. Understanding of the parametric approach in CREO for efficient modeling.

3. Development of skills to refine and modify solid models using fillets, chamfers, and other tools.
 4. Familiarity with saving, exporting, and organizing 3D CAD files.
-

5. Common Errors and Troubleshooting

1. **Error:** Sketch is not fully constrained.
Solution: Ensure all dimensions and constraints are applied to the sketch.
 2. **Error:** Feature fails to regenerate.
Solution: Check for errors in the sketch or input parameters (e.g., invalid extrusion depth).
 3. **Error:** Model edges are not smooth.
Solution: Use fillets or chamfers to smooth sharp edges.
-

6. Thought-Provoking Questions

1. What are the advantages of using parametric modeling in 3D CAD software?
 2. How does the "Sweep" feature differ from the "Loft" feature in CREO?
 3. What are some real-world applications of 3D modeling in engineering and product design?
 4. How can Boolean operations be used to create complex 3D geometries?
 5. Why is it important to save 3D models in organized project folders?
-

7. Applications of 3D Modeling

1. **Product Design:** Designing mechanical parts, consumer products, and industrial components.
 2. **Simulation:** Preparing models for analysis in simulation software.
 3. **Prototyping:** Creating 3D prints or CNC-machined prototypes.
 4. **Manufacturing:** Generating manufacturing drawings and tool paths.
-

This concludes the experiment on creating 3D solid models using CREO software.

प्रयोग क्रमांक 4

शीर्षक: ज्यामिति मॉडलिंग सॉफ्टवेयर - CREO का उपयोग करके 3D ठोस मॉडल बनाना

1. परिचय और अवलोकन

1.1 उद्देश्य

इस प्रयोग का उद्देश्य छात्रों को CREO सॉफ्टवेयर का उपयोग करके 3D ठोस मॉडलिंग की अवधारणाओं से परिचित कराना है। छात्र इस सत्र के दौरान एक्सट्रूड, रिवोल्व, स्वीप, और लॉट जैसे फीचर्स का उपयोग करके बुनियादी और उन्नत 3D मॉडल बनाना सीखेंगे। यह सत्र 3D मॉडलिंग तकनीकों में प्रवीणता विकसित करने के लिए हाथों से अभ्यास प्रदान करेगा।

1.2 CREO का अवलोकन

CREO एक शक्तिशाली 3D CAD (कंप्यूटर-आधारित डिज़ाइन) सॉफ्टवेयर है, जिसका उपयोग उत्पाद डिज़ाइन और इंजीनियरिंग में व्यापक रूप से किया जाता है। यह उपयोगकर्ताओं को 3D मॉडल डिज़ाइन, सिमुलेट और सत्यापित करने की सुविधा प्रदान करता है, जिससे यह आधुनिक इंजीनियरिंग और उत्पाद विकास के लिए एक आवश्यक उपकरण बन जाता है। CREO उपयोगकर्ता के अनुकूल इंटरफेस, मजबूत मॉडलिंग फीचर्स और पैमानेनुपातिक क्षमताएँ प्रदान करता है, जो डिज़ाइनरों को सटीक और जटिल ठोस मॉडल बनाने में कुशलता प्रदान करते हैं।

4. प्रयोग को निष्पादित करने की विधियाँ

3.1 सेटअप और आरंभ

1. CREO सॉफ्टवेयर लॉन्च करें: अपने सिस्टम पर CREO एप्लिकेशन खोलें।
2. नया भाग फ़ाइल बनाएं: "फ़ाइल" मेनू पर जाएं, "नया" चुनें, और विकल्पों से "भाग" चुनें। अपनी फ़ाइल को नाम दें और "ठीक है" पर क्लिक करें।
3. इकाइयाँ सेट करें: सुनिश्चित करें कि "सेटअप" टैब के तहत सही यूनिट सिस्टम (जैसे, मिमी, सेंटीमीटर, इंच) चयनित है।

3.2 बुनियादी 3D मॉडल बनाना

1. स्केच निर्माण:

- o एक तल (जैसे, टॉप, फ्रंट, या राइट प्लेन) का चयन करके स्केच वातावरण खोलें।
- o निम्नलिखित स्केच उपकरणों का उपयोग करके एक 2D प्रोफाइल बनाएं:

- रेखा
- वृत्त
- आयत
- आर्क

- स्प्लाइन
 - o स्केच को आयामों और भौतिक रिश्तों (जैसे, समानांतर, लंबवत) को परिभाषित करके प्रतिबद्ध करें।
- 2. **एक्सट्रूड फीचर:**
 - o स्केच का चयन करें और "एक्सट्रूड" कमांड का उपयोग करके 2D प्रोफाइल को 3D ठोस में बदलें।
 - o एक्सट्रूज़न गहराई और दिशा निर्दिष्ट करें।
- 3. **रिवोल्व फीचर:**
 - o एक रिवोल्यूशन अक्ष के साथ स्केच बनाएं।
 - o "रिवोल्व" कमांड का उपयोग करके स्केच को अक्ष के चारों ओर घुमा कर 3D ठोस बनाएं।
- 4. **स्वीप फीचर:**
 - o एक प्रोफाइल और एक मार्ग (जैसे, स्प्लाइन या वक्र) बनाएं।
 - o "स्वीप" कमांड का उपयोग करके प्रोफाइल को मार्ग के साथ घुमा कर एक ठोस बनाएं।
- 5. **लॉट फीचर:**
 - o समानांतर या तिरछे तल पर कई स्केच बनाएं।
 - o "लॉट" कमांड का उपयोग करके स्केचों को आसानी से जोड़ते हुए एक ठोस बनाएं।
- 3.3 मॉडल्स को संपादित और संशोधित करना**
- 6. मॉडल के किनारों और सतहों को परिष्कृत करने के लिए "फिलेट", "चेम्फर" और "शेल" जैसे उपकरणों का उपयोग करें।
- 7. ठोस निकायों को जोड़ने या संशोधित करने के लिए बूलियन ऑपरेशन्स (जैसे, यूनियन, घटाना, इंटरसेक्ट) लागू करें।
- 3.4 फ़ाइल को सहेजना**
 - अपनी मॉडल को "फ़ाइल" मेनू में "सहेजें" विकल्प का उपयोग करके सहेजें। सुनिश्चित करें कि फ़ाइल भविष्य के उपयोग के लिए समर्पित परियोजना फ़ोल्डर में संग्रहित है।

4. अपेक्षित परिणाम

1. एक्सट्रूड, रिवोल्व, स्वीप, और लॉट जैसे बुनियादी और उन्नत फीचर्स के साथ 3D मॉडल बनाने की क्षमता।
 2. CREO में पैमानेनुपातिक दृष्टिकोण की समझ जो कुशल मॉडलिंग की सुविधा प्रदान करता है।
 3. फिलेट, चेम्फर, और अन्य उपकरणों का उपयोग करके ठोस मॉडल को परिष्कृत और संशोधित करने की क्षमता।
 4. 3D CAD फ़ाइलों को सहेजने, निर्यात करने और व्यवस्थित करने में परिचितता।
-

5. सामान्य त्रुटियाँ और समस्या निवारण

1. त्रुटि: स्केच पूरी तरह से प्रतिबद्ध नहीं है।
समाधान: सुनिश्चित करें कि सभी आयाम और प्रतिबद्धताएँ स्केच पर लागू की गई हैं।
2. त्रुटि: फीचर पुनः उत्पन्न करने में विफल है।
समाधान: स्केच या इनपुट पैरामीटरों (जैसे, अमान्य एक्सट्रूज़न गहराई) में त्रुटियों के लिए जांच करें।
3. त्रुटि: मॉडल के किनारे चिकने नहीं हैं।
समाधान: तीव्र किनारों को चिकना करने के लिए फिलेट या चेम्फर का उपयोग करें।

6. सोचने के लिए प्रश्न

1. 3D CAD सॉफ्टवेयर में पैमानेनुपातिक मॉडलिंग का उपयोग करने के क्या लाभ हैं?
2. CREO में "स्वीप" फीचर "लॉट" फीचर से कैसे भिन्न है?
3. इंजीनियरिंग और उत्पाद डिज़ाइन में 3D मॉडलिंग के कुछ वास्तविक-विश्व अनुप्रयोग क्या हैं?
4. जटिल 3D ज्यामितियाँ बनाने के लिए बूलियन ऑपरेशन्स का उपयोग कैसे किया जा सकता है?
5. 3D मॉडल को व्यवस्थित परियोजना फ़ोल्डरों में सहेजने का महत्व क्या है?

7. 3D मॉडलिंग के अनुप्रयोग

1. उत्पाद डिज़ाइन: यांत्रिक भागों, उपभोक्ता उत्पादों और औद्योगिक घटकों का डिज़ाइन।
2. सिमुलेशन: सिमुलेशन सॉफ्टवेयर में विश्लेषण के लिए मॉडल तैयार करना।
3. प्रोटोटाइपिंग: 3D प्रिंट्स या CNC-निर्मित प्रोटोटाइप बनाना।
4. निर्माण: निर्माण ड्राइंग और उपकरण पथ उत्पन्न करना।

यह CREO सॉफ्टवेयर का उपयोग करके 3D ठोस मॉडल बनाने पर आधारित प्रयोग समाप्त होता है।

Experiment No. 5

Title: Assembly of CAD Parts in CREO

1. Introduction and Objectives

1.1 Overview of Assembly Modeling

Assembly modeling is the process of bringing together multiple individual parts designed in CAD software to create a complete product. This involves defining the positional and relational constraints (such as alignments, mating, and offsets) between parts to ensure they interact and fit together as intended. In the context of CREO, assembly modeling allows users to visualize and validate the functionality of the product before physical manufacturing.

1.2 Objective

The objective of this experiment is to learn and practice assembling individual CAD parts into a cohesive assembly using CREO. The exercise will also involve applying various constraints and performing a simulation to verify the fit and function of the assembly.

2. Methodologies to Perform the Experiment

2.1 Setup and Initialization

1. **Launch CREO Software:** Open CREO Parametric and create a new assembly file by selecting New > Assembly. Assign a suitable name to the assembly.
2. **Load Individual Parts:** Ensure the required part files (.prt) are available. Use the Assemble command to import each part into the assembly environment.

2.2 Performing the Assembly

1. **Base Component Placement:**
 - Choose the main part to serve as the base of the assembly.
 - Use the Default Constraint to fix the base part at the assembly origin.
2. **Adding Components:**
 - Add additional parts by selecting Assemble and placing them relative to the base part.
 - Apply constraints such as Mate, Align, Distance, or Angle to position the parts correctly.
3. **Apply Constraints:**
 - **Mate Constraint:** Ensures two surfaces are in contact.
 - **Align Constraint:** Ensures two surfaces or edges are parallel.
 - **Offset Constraint:** Defines a specific distance between two surfaces or edges.

- **Angle Constraint:** Maintains an angular relationship between two parts.
- 4. **Regenerate Assembly:** After applying constraints, click on the Regenerate button to update the assembly structure and resolve constraints.

2.3 Validation and Simulation

1. **Interference Check:**
 - Use the Analysis > Interference tool to ensure that no two parts overlap.
2. **Dynamic Simulation:**
 - Test the movement of parts using the Drag Components option.
 - Verify whether the parts function as intended, such as rotational or sliding motion.

2.4 Saving the Assembly

- Save the completed assembly file as .asm for future use.
 - Generate a 2D assembly drawing, if required, by selecting New > Drawing and specifying the views (top, front, isometric, etc.).
-

3. Expected Outcomes

1. A complete assembly model of all individual CAD parts.
 2. Proper application of constraints ensuring accurate positioning and alignment of parts.
 3. Verification of part functionality through dynamic simulation.
 4. Elimination of interference or overlap issues in the assembly.
-

4. Possibilities of Deviations from Real Scenarios

1. **Tolerance Issues:** Idealized constraints may not consider real-world manufacturing tolerances.
 2. **Constraint Conflicts:** Errors due to over-constraining parts in the assembly.
 3. **Motion Validation Limitations:** Simulated motions may not account for physical imperfections such as friction.
-

5. Reasons for Observed Deviations

1. **Incorrect Constraint Application:** Misalignment or unintentional gaps between parts due to wrong constraints.
2. **Inaccurate Dimensions:** Discrepancies in part dimensions leading to fit issues.
3. **Simplified Simulation:** CREO simulations may not account for material properties or forces applied during operation.

6. Thought-Provoking Questions

1. How would you modify your constraints if a design revision alters the dimensions of a critical part?
2. What methods can you use to identify and resolve interference issues in complex assemblies?
3. How can you ensure the assembly remains adaptable for future design iterations?
4. In what ways can assembly modeling be integrated with other engineering analyses, such as stress testing or thermal simulations?
5. How does the assembly process in CREO compare to physical assembly in terms of time and accuracy?

7. Practical Tips

1. Organize part files systematically to avoid confusion during the assembly process.
2. Use named constraints to maintain clarity and avoid over-constraining the model.
3. Save and frequently regenerate the assembly to ensure constraints are correctly applied.
4. Leverage CREO's built-in help and tutorials for advanced assembly features like flexible components or exploded views.

This concludes the experiment on creating Assembly models using CREO software

प्रयोग संख्या 5

शीर्षक: CREO में CAD पार्ट्स का असेंबली

1. परिचय और उद्देश्य

1.1 असेंबली मॉडलिंग का अवलोकन

असेंबली मॉडलिंग वह प्रक्रिया है जिसमें CAD सॉफ्टवेयर में डिज़ाइन किए गए विभिन्न व्यक्तिगत भागों को एक पूर्ण उत्पाद बनाने के लिए एकत्रित किया जाता है। इसमें भागों के बीच स्थिति और संबंधित प्रतिबंध (जैसे संरेखण, जोड़ना और ऑफ़सेट) को परिभाषित करना शामिल है, ताकि यह सुनिश्चित किया जा सके कि वे इंटरैक्ट करें और इच्छानुसार एक साथ फिट हो जाएं। CREO के संदर्भ में, असेंबली मॉडलिंग उपयोगकर्ताओं को उत्पाद की कार्यक्षमता को भौतिक निर्माण से पहले विज़ुअलाइज़ और सत्यापित करने की अनुमति देती है।

1.2 उद्देश्य

इस प्रयोग का उद्देश्य CREO का उपयोग करके व्यक्तिगत CAD भागों को एक सुसंगत असेंबली में इकट्ठा करने का अभ्यास करना है। इस अभ्यास में विभिन्न प्रतिबंधों को लागू करना और असेंबली की फिट और कार्यक्षमता को सत्यापित करने के लिए एक सिमुलेशन करना भी शामिल होगा।

2. प्रयोग को निष्पादित करने की विधियाँ

2.1 सेटअप और प्रारंभिक चरण

1. **CREO सॉफ्टवेयर लॉन्च करें:** CREO Parametric खोलें और New > Assembly का चयन करके एक नई असेंबली फ़ाइल बनाएं। असेंबली को उपयुक्त नाम दें।
2. **व्यक्तिगत पार्ट लोड करें:** यह सुनिश्चित करें कि आवश्यक पार्ट फ़ाइलें (.prt) उपलब्ध हैं। प्रत्येक पार्ट को असेंबली वातावरण में आयात करने के लिए Assemble कमांड का उपयोग करें

2.2 असेंबली निष्पादन

○ आधार घटक का स्थान निर्धारण:

1. मुख्य पार्ट का चयन करें जो असेंबली का आधार बनेगा।
2. Default Constraint का उपयोग करके आधार पार्ट को असेंबली की उत्पत्ति पर फिक्स करें।

○ घटक जोड़ना:

1. अतिरिक्त भाग जोड़ने के लिए Assemble का चयन करें और उन्हें आधार भाग के सापेक्ष रखें।
2. पार्ट्स को सही स्थान पर रखने के लिए Mate, Align, Distance या Angle जैसे प्रतिबंध लागू करें।

- **प्रतिबंध लागू करना:**

1. **Mate Constraint:** यह सुनिश्चित करता है कि दो सतहें संपर्क में हैं।
 2. **Align Constraint:** यह सुनिश्चित करता है कि दो सतहें या किनारे समानांतर हैं।
 3. **Offset Constraint:** यह दो सतहों या किनारों के बीच एक विशिष्ट दूरी को परिभाषित करता है।
 4. **Angle Constraint:** यह दो भागों के बीच कोणीय संबंध बनाए रखता है।
- **असेंबली को फिर से जनरेट करें:** प्रतिबंध लागू करने के बाद, असेंबली संरचना को अपडेट करने और प्रतिबंधों को हल करने के लिए Regenerate बटन पर क्लिक करें।

2.3 सत्यापन और सिमुलेशन

- **इंटरफेरेंस चेक:**

5. यह सुनिश्चित करने के लिए Analysis > Interference टूल का उपयोग करें कि कोई दो पार्ट एक-दूसरे से ओवरलैप न हों।

- **डायनेमिक सिमुलेशन:**

6. Drag Components विकल्प का उपयोग करके पार्ट्स की गति का परीक्षण करें।
7. यह सत्यापित करें कि पार्ट्स इच्छानुसार कार्य करते हैं, जैसे कि घुमाव या स्लाइडिंग गति।

2.4 असेंबली को सहेजना

- पूर्ण असेंबली फ़ाइल को .asm के रूप में भविष्य के उपयोग के लिए सहेजें।
- यदि आवश्यक हो, तो New > Drawing का चयन करके 2D असेंबली ड्राइंग जनरेट करें और दृश्य (ऊपर, सामने, आइसोमेट्रिक, आदि) निर्दिष्ट करें।

3. अपेक्षित परिणाम

1. सभी व्यक्तिगत CAD पार्ट्स का एक पूर्ण असेंबली मॉडल।
2. पार्ट्स की सही स्थिति और संरेखण सुनिश्चित करने के लिए प्रतिबंधों का सही ढंग से लागू करना।
3. डायनेमिक सिमुलेशन के माध्यम से पार्ट्स की कार्यक्षमता का सत्यापन।
4. असेंबली में इंटरफेरेंस या ओवरलैप समस्याओं का समाधान।

4. वास्तविक परिदृश्यों से विचलन की संभावनाएँ

1. **सहिष्णुता समस्याएँ:** आदर्श प्रतिबंध वास्तविक दुनिया के निर्माण सहिष्णुता को ध्यान में नहीं रखते।
 2. **प्रतिबंध संघर्ष:** असेंबली में भागों को अत्यधिक प्रतिबंधित करने के कारण त्रुटियाँ हो सकती हैं।
 3. **गति सत्यापन सीमाएँ:** सिमुलेटेड गति में भौतिक दोष जैसे घर्षण को ध्यान में नहीं रखा जा सकता।
-

5. देखी गई विचलन के कारण

1. **गलत प्रतिबंध लागू करना:** गलत प्रतिबंधों के कारण भागों के बीच संरेखण या अनचाही गैप हो सकती है।
 2. **असंतुलित आयाम:** पार्ट के आयामों में विसंगतियाँ फिट समस्याएँ उत्पन्न कर सकती हैं।
 3. **सरलीकृत सिमुलेशन:** CREO सिमुलेशन सामग्री गुण या संचालन के दौरान लागू बलों को ध्यान में नहीं रखते।
-

6. विचारणीय प्रश्न

1. यदि डिज़ाइन संशोधन से एक महत्वपूर्ण भाग का आयाम बदलता है, तो आप अपने प्रतिबंधों को कैसे संशोधित करेंगे?
 2. जटिल असेंबली में इंटरफेरेंस समस्याओं को पहचानने और हल करने के लिए आप कौन सी विधियाँ अपना सकते हैं?
 3. आप यह कैसे सुनिश्चित कर सकते हैं कि असेंबली भविष्य के डिज़ाइन पुनरावृत्तियों के लिए अनुकूलनीय बनी रहे?
 4. असेंबली मॉडलिंग को अन्य इंजीनियरिंग विश्लेषणों जैसे तनाव परीक्षण या थर्मल सिमुलेशन के साथ कैसे एकीकृत किया जा सकता है?
 5. CREO में असेंबली प्रक्रिया भौतिक असेंबली से समय और सटीकता के संदर्भ में कैसे तुलना की जाती है?
-

7. व्यावहारिक सुझाव

1. असेंबली प्रक्रिया के दौरान भ्रम से बचने के लिए पार्ट फ़ाइलों को व्यवस्थित रूप से व्यवस्थित करें।
2. मॉडल को अत्यधिक प्रतिबंधित करने से बचने के लिए नामित प्रतिबंधों का उपयोग करें।
3. यह सुनिश्चित करने के लिए असेंबली को सहेजें और नियमित रूप से फिर से जनरेट करें कि प्रतिबंध सही ढंग से लागू हो रहे हैं।

4. CREO की अंतर्निहित सहायता और ट्यूटोरियल का लाभ उठाएं ताकि लचीले घटकों या विस्फोटित दृश्यों जैसे उन्नत असेंबली सुविधाओं को समझा जा सके।

यह CREO सॉफ्टवेयर का उपयोग करके असेंबली मॉडल बनाने पर प्रयोग को समाप्त करता है।

Maulana Azad National Institute of Technology Bhopal

Department of Mechanical Engineering



Lab Manual

Automation and Robotics Lab-I

M. Tech (Automation and Robotics)

Automation and Robotics Lab-I

List of Experiments

1. Verify the Position & Orientation of the end effector of the 3-DOF (RPR) manipulator using Robo-Analyzer
2. Obtain the joint angles of 3-DOF (RRR) manipulator by using Robo-Analyzer
3. Development of Trajectory for 2-DOF manipulator using Robo-Analyzer software
4. Write a program for 3-wheeled mobile robot to move forward, stop, turn left and turn right using Arduino.
5. Control of an autonomous line following robot using IR sensors
6. Control of an autonomous obstacle avoidance robot using ultrasonic sensor
7. Write the program using Arduino for Human Following Robot
8. Development of control algorithm for 4-DOF robotic Manipulator
9. Calculate the Gait angles and dynamic balance margin of 16 DOF Humanoid Robot
10. Do it yourself (DIY) experiments (Students should take the real-world issue and they have to think, decide and do things independently)

स्वचालन और रोबोटिक्स लैब

प्रयोगों की सूची

1. रोबो-विश्लेषक का उपयोग करके 3-डीओएफ (आरपीआर) मैनिपुलेटर के अंतिम प्रभावक की स्थिति और अभिविन्यास को सत्यापित करें
2. रोबो-विश्लेषक का उपयोग करके 3-डीओएफ (आरआरआर) मैनिपुलेटर के संयुक्त कोण प्राप्त करें
3. रोबो-विश्लेषक सॉफ्टवेयर का उपयोग करके 2-डीओएफ मैनिपुलेटर के लिए प्रक्षेपवक्र का विकास
4. Arduino का उपयोग करके 3-पहियों वाले मोबाइल रोबोट को आगे बढ़ने, रुकने, बाएं मुड़ने और दाएं मुड़ने के लिए एक प्रोग्राम लिखें।
5. आईआर सेंसर का उपयोग करके रोबोट के बाद एक स्वायत्त लाइन का नियंत्रण
6. अल्ट्रासोनिक सेंसर का उपयोग करके एक स्वायत्त बाधा निवारण रोबोट का नियंत्रण
7. ह्यूमन फॉलोइंग रोबोट के लिए Arduino का उपयोग करके प्रोग्राम लिखें
8. 4-डीओएफ रोबोटिक मैनिपुलेटर के लिए नियंत्रण एल्गोरिदम का विकास
9. 16 डीओएफ ह्यूमनॉइड रोबोट के चाल कोण और गतिशील संतुलन मार्जिन की गणना करें
10. इसे स्वयं करें (DIY) प्रयोग (छात्रों को वास्तविक दुनिया के मुद्दे को लेना चाहिए और उन्हें स्वतंत्र रूप से सोचना, निर्णय लेना और कार्य करना होगा)

Experiment No-1

Statement: Verify the Position & Orientation of the end effector of the 3-DOF (RPR) manipulator using Robo-Analyzer

1. Introduction and Physical Information about the Platform

1.1 Overview of the 3-DOF RPR Manipulator

The RPR (Revolute-Prismatic-Revolute) manipulator is a three-degree-of-freedom robotic system. It comprises:

- **Revolute Joint (R):** Allows rotational motion about a fixed axis.
- **Prismatic Joint (P):** Allows linear motion along a fixed direction.
- **Revolute Joint (R):** Another rotational joint allowing motion about a fixed axis.

This manipulator is often used in scenarios requiring constrained motion, such as pick-and-place tasks or trajectory tracing in confined workspaces.

1.2 Robo-Analyzer Software

Robo-Analyzer is a robotic simulation and analysis tool designed to model and simulate robotic manipulators. It provides a virtual environment to test and verify the kinematic parameters, position, and orientation of the end effector.

2. Methodologies to Perform the Experiment

2.1 Setup and Initialization

1. **Launch Robo-Analyzer:** Open the Robo-Analyzer software and create a new project.
2. **Define Manipulator Parameters:** Input the Denavit-Hartenberg (D-H) parameters for the RPR manipulator:
 - Link lengths, joint angles, and displacement ranges.
 - Specify base and tool configurations if necessary.
3. **Visualize the Manipulator:** Verify that the virtual manipulator corresponds to the desired 3-DOF RPR configuration.

2.2 Performing the Experiment

1. **Set Joint Configurations:**
 - Adjust the revolute and prismatic joints to specified values for different test cases.
2. **Compute Forward Kinematics:**
 - Use Robo-Analyzer's kinematic analysis module to calculate the position and orientation of the end effector.
3. **Record Data:**
 - Note the x, y, z coordinates and the orientation (roll, pitch, yaw angles) of the end effector for each configuration.
4. **Verify with Theoretical Calculations:**
 - Compare the simulation results with manually computed forward kinematics results based on D-H parameters.

2.3 Visualization and Validation

- Observe the trajectory and workspace of the manipulator in the 3D simulation.
 - Ensure the motion adheres to the constraints of the RPR structure.
-

3. Expected Outcomes

1. Accurate determination of the position (x, y, z) of the end effector for various joint configurations.
 2. Correct orientation of the end effector represented by roll, pitch, and yaw angles.
 3. Visual confirmation of the manipulator's motion and workspace.
 4. Validation of forward kinematics equations using Robo-Analyzer outputs.
-

4. Possibilities of Deviations from Real Scenarios

1. **Numerical Errors:** Rounding-off errors in simulation calculations.
 2. **Idealized Assumptions:** Robo-Analyzer assumes perfect conditions without physical imperfections such as friction or backlash.
 3. **Simplified D-H Parameters:** Simplified modeling may not account for structural deformations or assembly inaccuracies.
-

5. Reasons for Observed Deviations

1. **Friction and Backlash:** These mechanical imperfections are not included in simulation models.
 2. **Manufacturing Tolerances:** Real-world components may differ slightly from the modeled dimensions.
 3. **Actuator Inaccuracies:** Limitations in actuator resolution and repeatability are not simulated.
 4. **Environmental Factors:** Effects like thermal expansion, which are present in real scenarios, are ignored in simulation.
-

6. Thought-Provoking Questions

1. What are the critical differences between the theoretical and simulated kinematics of the RPR manipulator?
2. How would you modify the D-H parameters if a structural modification, such as an extended link length, is introduced?
3. What factors could limit the accuracy of position and orientation measurements in real-world applications?
4. How would you adapt the manipulator's design to minimize deviations observed in simulations versus real scenarios?
5. What additional analyses can be performed using Robo-Analyzer to enhance understanding of manipulator dynamics?

प्रयोग 1:

कथन: रोबो-विश्लेषक का उपयोग करके 3-डीओएफ (आरपीआर) मैनिपुलेटर के अंतिम प्रभावक की स्थिति और अभिविन्यास को सत्यापित करें

1. परिचय और प्लेटफॉर्म के बारे में भौतिक जानकारी

1.1 3-डीओएफ आरपीआर मैनिपुलेटर का अवलोकन

आरपीआर (रिवॉल्यूट-प्रिज़मैटिक-रिवॉल्यूट) मैनिपुलेटर एक तीन-डिग्री-ऑफ-फ्रीडम रोबोटिक सिस्टम है। इसमें शामिल हैं:

- रिवॉल्यूट जॉइंट (R):** एक निश्चित धुरी के चारों ओर घूर्णी गति की अनुमति देता है।
- प्रिज़मैटिक जॉइंट (P):** एक निश्चित दिशा में रैखिक गति की अनुमति देता है।
- रिवॉल्यूट जॉइंट (R):** एक अन्य घूर्णी जॉइंट जो एक निश्चित धुरी के चारों ओर गति की अनुमति देता है।

यह मैनिपुलेटर उन परिदृश्यों में उपयोग किया जाता है जहां प्रतिबंधित गति की आवश्यकता होती है, जैसे कि पिक-एंड-प्लेस कार्य या सीमित कार्यक्षेत्रों में ट्राजेक्टरी ट्रेसिंग।

1.2 रोबो-एनालाइज़र सॉफ़्टवेयर

रोबो-एनालाइज़र एक रोबोटिक सिमुलेशन और विश्लेषण उपकरण है जिसे रोबोटिक मैनिपुलेटर्स को मॉडल और सिमुलेट करने के लिए डिज़ाइन किया गया है। यह अंतिम प्रभावक के सिनेमेटिक पैरामीटर, स्थिति और ओरिएंटेशन का परीक्षण और सत्यापन करने के लिए एक वर्चुअल वातावरण प्रदान करता है।

2. प्रयोग करने की विधियाँ

2.1 सेटअप और प्रारंभिककरण

- रोबो-एनालाइज़र लॉन्च करें:** रोबो-एनालाइज़र सॉफ़्टवेयर खोलें और एक नई परियोजना बनाएं।
- मैनिपुलेटर पैरामीटर परिभाषित करें:** आरपीआर मैनिपुलेटर के लिए डेनाविट-हार्टनबर्ग (डी-एच) पैरामीटर इनपुट करें:
 - लिंक की लंबाई, जॉइंट कोण, और विस्थापन श्रेणियाँ।
 - यदि आवश्यक हो तो बेस और टूल कॉन्फ़िगरेशन निर्दिष्ट करें।
- मैनिपुलेटर को विज़ुअलाइज़ करें:** सत्यापित करें कि वर्चुअल मैनिपुलेटर वांछित 3-डीओएफ आरपीआर कॉन्फ़िगरेशन के अनुरूप है।

2.2 प्रयोग करना

- जॉइंट कॉन्फ़िगरेशन सेट करें:**
 - विभिन्न परीक्षण मामलों के लिए रिवॉल्यूट और प्रिज़मैटिक जॉइंट्स को निर्दिष्ट मानों पर समायोजित करें।
- फॉरवर्ड काइनामेटिक्स की गणना करें:**
 - रोबो-एनालाइज़र के काइनामेटिक विश्लेषण मॉड्यूल का उपयोग करके एंड इफेक्टर की स्थिति और ओरिएंटेशन की गणना करें।
- डेटा रिकॉर्ड करें:**

- प्रत्येक कॉन्फिगरेशन के लिए एंड इफेक्टर के x, y, z निर्देशांक और ओरिएंटेशन (रोल, पिच, यॉ एंगल्स) को नोट करें।

4. सैद्धांतिक गणनाओं के साथ सत्यापन करें:

- डी-एच पैरामीटर आधारित मैनुअल फॉरवर्ड काइनामेटिक्स परिणामों के साथ सिमुलेशन परिणामों की तुलना करें।

2.3 विजुअलाइज़ेशन और सत्यापन

- 3डी सिमुलेशन में मैनिपुलेटर के प्रक्षेपवक्र और कार्यक्षेत्र का अवलोकन करें।
- सुनिश्चित करें कि गति आरपीआर संरचना की बाधाओं का पालन करती है।

3. अपेक्षित परिणाम

1. विभिन्न जॉइंट कॉन्फिगरेशनों के लिए एंड इफेक्टर की स्थिति (x, y, z) का सटीक निर्धारण।
2. रोल, पिच और यॉ कोणों द्वारा दर्शाए गए एंड इफेक्टर की सही ओरिएंटेशन।
3. मैनिपुलेटर की गति और कार्यक्षेत्र का दृश्य पुष्टि।
4. रोबो-एनालाइज़र आउटपुट का उपयोग करके फॉरवर्ड काइनामेटिक्स समीकरणों का सत्यापन।

4. वास्तविक परिदृश्यों से संभावित विचलन

1. **सांख्यिकीय त्रुटियाँ:** सिमुलेशन गणनाओं में राउंडिंग-ऑफ त्रुटियाँ।
2. **आदर्श धारणाएँ:** रोबो-एनालाइज़र आदर्श परिस्थितियों को मानता है जिसमें भौतिक अपूर्णताएँ, जैसे कि घर्षण या बैकलैश, शामिल नहीं हैं।
3. **सरलित डी-एच पैरामीटर:** सरलीकृत मॉडलिंग संरचनात्मक विकृति या असेंबली अशुद्धियों को ध्यान में नहीं रखती।

5. विचलन के कारण

1. **घर्षण और बैकलैश:** ये यांत्रिक अपूर्णताएँ सिमुलेशन मॉडलों में शामिल नहीं होती हैं।
2. **निर्माण सहनशीलताएँ:** वास्तविक दुनिया के घटक मॉडल किए गए आयामों से थोड़े भिन्न हो सकते हैं।
3. **अधिनियमक अशुद्धियाँ:** अधिनियमक के रिज़ॉल्यूशन और पुनरावृत्ति में सीमाएँ सिमुलेशन में शामिल नहीं होती हैं।
4. **पर्यावरणीय कारक:** तापीय विस्तार जैसे प्रभाव, जो वास्तविक परिदृश्यों में उपस्थित होते हैं, सिमुलेशन में अनदेखे होते हैं।

6. विचारोत्तेजक प्रश्न

1. आरपीआर मैनिपुलेटर की सैद्धांतिक और सिमुलेटेड काइनामेटिक्स के बीच महत्वपूर्ण अंतर क्या हैं?
2. यदि संरचनात्मक संशोधन, जैसे लिंक की लंबाई बढ़ाना, किया जाए तो आप डी-एच पैरामीटर को कैसे संशोधित करेंगे?
3. वास्तविक विश्व अनुप्रयोगों में स्थिति और ओरिएंटेशन मापन की सटीकता को कौन से कारक सीमित कर सकते हैं?
4. आप सिमुलेशन बनाम वास्तविक परिदृश्यों में देखे गए विचलनों को कम करने के लिए मैनिपुलेटर के डिज़ाइन को कैसे अनुकूलित करेंगे?
5. मैनिपुलेटर की गतिशीलता की समझ को बढ़ाने के लिए आप रोबो-एनालाइज़र का उपयोग करके कौन से अतिरिक्त विश्लेषण कर सकते हैं?

Experiment No-2

Statement: Obtain the joint angles of 3-DOF (RRR) manipulator by using Robo-Analyzer

1. Introduction and Physical Information of the Platform

1.1 Overview of the 3-DOF RRR Manipulator

The RRR (Revolute-Revolute-Revolute) manipulator is a robotic system with three degrees of freedom, characterized by:

- **Revolute Joint (R):** Allows rotational motion around a fixed axis.
- **Revolute Joint (R):** Facilitates additional rotational motion.
- **Revolute Joint (R):** Provides a third degree of rotational freedom.

These manipulators are widely used in industrial applications, including arc welding, material handling, and precision assembly tasks.

1.2 Robo-Analyzer Software

Robo-Analyzer is a simulation tool designed for the kinematic and dynamic analysis of robotic manipulators. It provides a virtual platform to visualize and analyze the joint angles, link lengths, and other parameters of a robotic arm.

2. Methodology to Perform the Experiment

2.1 Setup and Initialization

1. **Launch Robo-Analyzer:** Open the software and create a new project.
2. **Define the Manipulator Parameters:**
 - Input Denavit-Hartenberg (DH) parameters for the RRR manipulator, including link lengths and joint angles.
 - Specify base and tool configurations, if applicable.
3. **Visualize the Manipulator:** Verify the virtual manipulator conforms to the 3-DOF RRR configuration.

2.2 Conducting the Experiment

1. **Set the Joint Configurations:**
 - Adjust the revolute joints to specific angles for different test cases.
2. **Perform Forward Kinematics Analysis:**
 - Use the kinematic analysis module in Robo-Analyzer to calculate the end-effector's position and orientation.
3. **Compute Inverse Kinematics:**
 - Calculate the joint angles required to achieve a given end-effector position.
4. **Record the Data:**
 - Document the joint angles (in degrees) and the corresponding end-effector coordinates (x, y, z).

2.3 Visualization and Verification

- Observe the manipulator's trajectory and workspace in the 3D simulation.
 - Cross-verify the computed joint angles with theoretical values derived manually using DH parameters.
-

3. Expected Outcomes

1. Accurate determination of joint angles for various end-effector positions.
 2. Visualization of the manipulator's motion and workspace.
 3. Validation of forward and inverse kinematic computations using Robo-Analyzer.
-

4. Possibilities of Deviations in Real Scenarios

1. **Numerical Errors:** Rounding-off errors during software calculations.
 2. **Ideal Assumptions:** Robo-Analyzer assumes ideal conditions without considering mechanical imperfections like friction or backlash.
 3. **Simplified DH Parameters:** Real-world manipulators may exhibit slight deviations due to construction tolerances or wear.
-

5. Reasons for Deviations

1. **Friction and Backlash:** Mechanical imperfections affecting joint motion.
 2. **Construction Tolerances:** Variability in component dimensions.
 3. **Actuator Limitations:** Finite resolution and repeatability of actuators.
 4. **Environmental Factors:** Effects such as thermal expansion in real-world scenarios.
-

6. Thought-Provoking Questions

1. What factors contribute to discrepancies between simulated and real-world kinematic analyses?
2. How would you modify DH parameters if the manipulator's link lengths were altered?
3. What challenges might arise in achieving precise joint angles in physical manipulators?
4. How can the design of a manipulator be optimized to minimize deviations in joint angles?
5. What additional analyses can be performed using Robo-Analyzer to enhance the understanding of manipulator dynamics?

प्रयोग 2:

कथन: रोबो-विश्लेषक का उपयोग करके 3-डीओएफ (आरआरआर) मैनिपुलेटर के संयुक्त कोण प्राप्त करें

1. परिचय और प्लेटफॉर्म की भौतिक जानकारी

1.1 3-डीओएफ आरआरआर मैनिपुलेटर का अवलोकन

आरआरआर (रिवॉल्यूट-रिवॉल्यूट-रिवॉल्यूट) मैनिपुलेटर तीन डिग्री-ऑफ-फ्रीडम वाला रोबोटिक सिस्टम है, जिसमें शामिल हैं:

- **रिवॉल्यूट जॉइंट (R):** एक निश्चित धुरी के चारों ओर घूर्णी गति की अनुमति देता है।
- **रिवॉल्यूट जॉइंट (R):** अतिरिक्त घूर्णी गति की सुविधा प्रदान करता है।
- **रिवॉल्यूट जॉइंट (R):** घूर्णी स्वतंत्रता की तीसरी डिग्री प्रदान करता है।

ये मैनिपुलेटर औद्योगिक अनुप्रयोगों में व्यापक रूप से उपयोग किए जाते हैं, जैसे आर्क वेल्डिंग, सामग्री संचालन, और सटीक असेंबली कार्य।

1.2 रोबो-एनालाइज़र सॉफ़्टवेयर

रोबो-एनालाइज़र एक सिमुलेशन टूल है जिसे रोबोटिक मैनिपुलेटर्स के काइनेमेटिक और डायनेमिक विश्लेषण के लिए डिज़ाइन किया गया है। यह संयुक्त कोण, लिंक लंबाई, और अन्य मापदंडों का विश्लेषण और कल्पना करने के लिए एक वर्चुअल प्लेटफॉर्म प्रदान करता है।

2. प्रयोग करने की विधियाँ

2.1 सेटअप और प्रारंभिककरण

1. **रोबो-एनालाइज़र लॉन्च करें:** सॉफ़्टवेयर खोलें और एक नई परियोजना बनाएं।
2. **मैनिपुलेटर पैरामीटर परिभाषित करें:**
 - आरआरआर मैनिपुलेटर के लिए डेनाविट-हार्टनबर्ग (डीएच) पैरामीटर इनपुट करें।
 - बेस और टूल कॉन्फ़िगरेशन निर्दिष्ट करें, यदि आवश्यक हो।
3. **मैनिपुलेटर को विजुअलाइज़ करें:** सत्यापित करें कि वर्चुअल मैनिपुलेटर 3-डीओएफ आरआरआर कॉन्फ़िगरेशन के अनुरूप है।

2.2 प्रयोग करना

1. **जॉइंट कॉन्फ़िगरेशन सेट करें:**
 - विभिन्न परीक्षण मामलों के लिए रिवॉल्यूट जॉइंट्स को निर्दिष्ट कोणों पर समायोजित करें।
2. **फॉरवर्ड काइनेमेटिक्स विश्लेषण करें:**
 - एंड-इफेक्टर की स्थिति और ओरिएंटेशन की गणना के लिए रोबो-एनालाइज़र के काइनेमेटिक विश्लेषण मॉड्यूल का उपयोग करें।
3. **इनवर्स काइनेमेटिक्स की गणना करें:**
 - एंड-इफेक्टर की दी गई स्थिति को प्राप्त करने के लिए आवश्यक संयुक्त कोणों की गणना करें।

4. डेटा रिकॉर्ड करें:

- संयुक्त कोण (डिग्री में) और संबंधित एंड-इफेक्टर निर्देशांक (x, y, z) को दस्तावेज़ करें।

2.3 विज़ुअलाइज़ेशन और सत्यापन

- 3डी सिमुलेशन में मैनिपुलेटर के प्रक्षेपवक्र और कार्यक्षेत्र का अवलोकन करें।
- डीएच पैरामीटर का उपयोग करके मैनुअल रूप से प्राप्त सैद्धांतिक मूल्यों के साथ गणना की गई संयुक्त कोणों को सत्यापित करें।

3. अपेक्षित परिणाम

1. विभिन्न एंड-इफेक्टर स्थितियों के लिए संयुक्त कोणों का सटीक निर्धारण।
2. मैनिपुलेटर की गति और कार्यक्षेत्र का दृश्य अवलोकन।
3. रोबो-एनालाइज़र का उपयोग करके फॉरवर्ड और इनवर्स काइनेमेटिक गणनाओं का सत्यापन।

4. वास्तविक परिदृश्यों से संभावित विचलन

1. **सांख्यिकीय त्रुटियाँ:** सॉफ़्टवेयर गणनाओं में राउंडिंग-ऑफ़ त्रुटियाँ।
2. **आदर्श धारणाएँ:** रोबो-एनालाइज़र आदर्श परिस्थितियों को मानता है और यांत्रिक अपूर्णताओं को ध्यान में नहीं रखता।
3. **सरलित डीएच पैरामीटर:** वास्तविक मैनिपुलेटर में निर्माण सहनशीलताओं के कारण मामूली विचलन हो सकते हैं।

5. विचलन के कारण

1. **घर्षण और बैकलैश:** संयुक्त गति को प्रभावित करने वाली यांत्रिक अपूर्णताएँ।
2. **निर्माण सहनशीलताएँ:** घटकों के आयामों में विविधता।
3. **एक्चुएटर सीमाएँ:** एक्चुएटर की सीमित रिज़ॉल्यूशन और पुनरावृत्ति।
4. **पर्यावरणीय कारक:** तापीय विस्तार जैसे वास्तविक परिदृश्यों में उपस्थित प्रभाव।

6. विचारोत्तेजक प्रश्न

1. सिमुलेटेड और वास्तविक काइनेमेटिक विश्लेषणों के बीच अंतर में कौन से कारक योगदान करते हैं?
2. यदि मैनिपुलेटर की लिंक लंबाई को बदला जाए तो आप डीएच पैरामीटर को कैसे संशोधित करेंगे?
3. वास्तविक मैनिपुलेटर में सटीक संयुक्त कोण प्राप्त करने में कौन सी चुनौतियाँ हो सकती हैं?
4. संयुक्त कोणों में विचलन को कम करने के लिए आप मैनिपुलेटर के डिज़ाइन को कैसे अनुकूलित करेंगे?
5. मैनिपुलेटर गतिशीलता की समझ बढ़ाने के लिए आप रोबो-एनालाइज़र का उपयोग करके कौन से अतिरिक्त विश्लेषण कर सकते हैं?

Experiment No-3

Statement: Development of Trajectory for 2-DOF manipulator using Robo-Analyzer software

1. Introduction and Physical Information of the Platform

1.1 Overview of the 3-DOF RRR Manipulator

The RRR (Revolute-Revolute-Revolute) manipulator is a robotic system with three degrees of freedom, characterized by:

- **Revolute Joint (R):** Allows rotational motion around a fixed axis.
- **Revolute Joint (R):** Facilitates additional rotational motion.
- **Revolute Joint (R):** Provides a third degree of rotational freedom.

These manipulators are widely used in industrial applications, including arc welding, material handling, and precision assembly tasks.

1.2 Robo-Analyzer Software

Robo-Analyzer is a simulation tool designed for the kinematic and dynamic analysis of robotic manipulators. It provides a virtual platform to visualize and analyze the joint angles, link lengths, and other parameters of a robotic arm.

2. Methodology to Perform the Experiment

2.1 Setup and Initialization

1. **Launch Robo-Analyzer:** Open the software and create a new project.
2. **Define the Manipulator Parameters:**
 - Input Denavit-Hartenberg (DH) parameters for the RRR manipulator, including link lengths and joint angles.
 - Specify base and tool configurations, if applicable.
3. **Visualize the Manipulator:** Verify the virtual manipulator conforms to the 3-DOF RRR configuration.

2.2 Conducting the Experiment

1. **Set the Joint Configurations:**
 - Adjust the revolute joints to specific angles for different test cases.
2. **Perform Forward Kinematics Analysis:**
 - Use the kinematic analysis module in Robo-Analyzer to calculate the end-effector's position and orientation.
3. **Compute Inverse Kinematics:**
 - Calculate the joint angles required to achieve a given end-effector position.
4. **Record the Data:**
 - Document the joint angles (in degrees) and the corresponding end-effector coordinates (x, y, z).

2.3 Visualization and Verification

- Observe the manipulator's trajectory and workspace in the 3D simulation.
 - Cross-verify the computed joint angles with theoretical values derived manually using DH parameters.
-

3. Expected Outcomes

1. Accurate determination of joint angles for various end-effector positions.
 2. Visualization of the manipulator's motion and workspace.
 3. Validation of forward and inverse kinematic computations using Robo-Analyzer.
-

4. Possibilities of Deviations in Real Scenarios

1. **Numerical Errors:** Rounding-off errors during software calculations.
 2. **Ideal Assumptions:** Robo-Analyzer assumes ideal conditions without considering mechanical imperfections like friction or backlash.
 3. **Simplified DH Parameters:** Real-world manipulators may exhibit slight deviations due to construction tolerances or wear.
-

5. Reasons for Deviations

1. **Friction and Backlash:** Mechanical imperfections affecting joint motion.
 2. **Construction Tolerances:** Variability in component dimensions.
 3. **Actuator Limitations:** Finite resolution and repeatability of actuators.
 4. **Environmental Factors:** Effects such as thermal expansion in real-world scenarios.
-

6. Thought-Provoking Questions

1. What factors contribute to discrepancies between simulated and real-world kinematic analyses?
2. How would you modify DH parameters if the manipulator's link lengths were altered?
3. What challenges might arise in achieving precise joint angles in physical manipulators?
4. How can the design of a manipulator be optimized to minimize deviations in joint angles?
5. What additional analyses can be performed using Robo-Analyzer to enhance the understanding of manipulator dynamics?

प्रयोग 3:

कथन: रोबो-विश्लेषक सॉफ्टवेयर का उपयोग करके 2-डीओएफ मैनिपुलेटर के लिए प्रक्षेपवक्र का विकास

1. परिचय और प्लेटफॉर्म की भौतिक जानकारी

1.1 3-डीओएफ आरआरआर मैनिपुलेटर का अवलोकन

आरआरआर (रिवॉल्यूट-रिवॉल्यूट-रिवॉल्यूट) मैनिपुलेटर तीन डिग्री-ऑफ-फ्रीडम वाला रोबोटिक सिस्टम है, जिसमें शामिल हैं:

- **रिवॉल्यूट जॉइंट (R):** एक निश्चित धुरी के चारों ओर घूर्णी गति की अनुमति देता है।
- **रिवॉल्यूट जॉइंट (R):** अतिरिक्त घूर्णी गति की सुविधा प्रदान करता है।
- **रिवॉल्यूट जॉइंट (R):** घूर्णी स्वतंत्रता की तीसरी डिग्री प्रदान करता है।

ये मैनिपुलेटर औद्योगिक अनुप्रयोगों में व्यापक रूप से उपयोग किए जाते हैं, जैसे आर्क वेल्डिंग, सामग्री संचालन, और सटीक असेंबली कार्य।

1.2 रोबो-एनालाइज़र सॉफ्टवेयर

रोबो-एनालाइज़र एक सिमुलेशन टूल है जिसे रोबोटिक मैनिपुलेटर्स के काइनेमेटिक और डायनेमिक विश्लेषण के लिए डिज़ाइन किया गया है। यह संयुक्त कोण, लिंक लंबाई, और अन्य मापदंडों का विश्लेषण और कल्पना करने के लिए एक वर्चुअल प्लेटफॉर्म प्रदान करता है।

2. प्रयोग करने की विधियाँ

2.1 सेटअप और प्रारंभिककरण

1. **रोबो-एनालाइज़र लॉन्च करें:** सॉफ्टवेयर खोलें और एक नई परियोजना बनाएं।
2. **मैनिपुलेटर पैरामीटर परिभाषित करें:**
 - आरआरआर मैनिपुलेटर के लिए डेनाविट-हार्टनबर्ग (डीएच) पैरामीटर इनपुट करें।
 - बेस और टूल कॉन्फिगरेशन निर्दिष्ट करें, यदि आवश्यक हो।
3. **मैनिपुलेटर को विजुअलाइज़ करें:** सत्यापित करें कि वर्चुअल मैनिपुलेटर 3-डीओएफ आरआरआर कॉन्फिगरेशन के अनुरूप है।

2.2 प्रयोग करना

1. **जॉइंट कॉन्फिगरेशन सेट करें:**
 - विभिन्न परीक्षण मामलों के लिए रिवॉल्यूट जॉइंट्स को निर्दिष्ट कोणों पर समायोजित करें।
2. **फॉरवर्ड काइनेमेटिक्स विश्लेषण करें:**
 - एंड-इफेक्टर की स्थिति और ओरिएंटेशन की गणना के लिए रोबो-एनालाइज़र के काइनेमेटिक विश्लेषण मॉड्यूल का उपयोग करें।
3. **इनवर्स काइनेमेटिक्स की गणना करें:**
 - एंड-इफेक्टर की दी गई स्थिति को प्राप्त करने के लिए आवश्यक संयुक्त कोणों की गणना करें।

4. डेटा रिकॉर्ड करें:

- संयुक्त कोण (डिग्री में) और संबंधित एंड-इफेक्टर निर्देशांक (x, y, z) को दस्तावेज़ करें।

2.3 विज़ुअलाइज़ेशन और सत्यापन

- 3डी सिमुलेशन में मैनिपुलेटर के प्रक्षेपवक्र और कार्यक्षेत्र का अवलोकन करें।
- डीएच पैरामीटर का उपयोग करके मैनुअल रूप से प्राप्त सैद्धांतिक मूल्यों के साथ गणना की गई संयुक्त कोणों को सत्यापित करें।

3. अपेक्षित परिणाम

1. विभिन्न एंड-इफेक्टर स्थितियों के लिए संयुक्त कोणों का सटीक निर्धारण।
2. मैनिपुलेटर की गति और कार्यक्षेत्र का दृश्य अवलोकन।
3. रोबो-एनालाइज़र का उपयोग करके फॉरवर्ड और इनवर्स काइनेमेटिक गणनाओं का सत्यापन।

4. वास्तविक परिदृश्यों से संभावित विचलन

1. **सांख्यिकीय त्रुटियाँ:** सॉफ़्टवेयर गणनाओं में राउंडिंग-ऑफ़ त्रुटियाँ।
2. **आदर्श धारणाएँ:** रोबो-एनालाइज़र आदर्श परिस्थितियों को मानता है और यांत्रिक अपूर्णताओं को ध्यान में नहीं रखता।
3. **सरलित डीएच पैरामीटर:** वास्तविक मैनिपुलेटर में निर्माण सहनशीलताओं के कारण मामूली विचलन हो सकते हैं।

5. विचलन के कारण

1. **घर्षण और बैकलैश:** संयुक्त गति को प्रभावित करने वाली यांत्रिक अपूर्णताएँ।
2. **निर्माण सहनशीलताएँ:** घटकों के आयामों में विविधता।
3. **एक्चुएटर सीमाएँ:** एक्चुएटर की सीमित रिज़ॉल्यूशन और पुनरावृत्ति।
4. **पर्यावरणीय कारक:** तापीय विस्तार जैसे वास्तविक परिदृश्यों में उपस्थित प्रभाव।

6. विचारोत्तेजक प्रश्न

1. सिमुलेटेड और वास्तविक काइनेमेटिक विश्लेषणों के बीच अंतर में कौन से कारक योगदान करते हैं?
2. यदि मैनिपुलेटर की लिंक लंबाई को बदला जाए तो आप डीएच पैरामीटर को कैसे संशोधित करेंगे?
3. वास्तविक मैनिपुलेटर में सटीक संयुक्त कोण प्राप्त करने में कौन सी चुनौतियाँ हो सकती हैं?
4. संयुक्त कोणों में विचलन को कम करने के लिए आप मैनिपुलेटर के डिज़ाइन को कैसे अनुकूलित करेंगे?
5. मैनिपुलेटर गतिशीलता की समझ बढ़ाने के लिए आप रोबो-एनालाइज़र का उपयोग करके कौन से अतिरिक्त विश्लेषण कर सकते हैं?

Experiment No-4

Statement: Write a program for 3-wheeled mobile robot to move forward, stop, turn left and turn right using Arduino.

Objective

To develop and execute an Arduino-based program for controlling the movements of a 3-wheeled mobile robot, demonstrating fundamental robot motion control techniques.

Methodology

1. Hardware Setup

1. Assemble a 3-wheeled mobile robot with the following components:
 - 3 DC motors for wheel actuation.
 - Motor driver module (L298N or similar).
 - Arduino microcontroller board.
 - Power source for motors and Arduino.
2. Connect the motors to the motor driver and Arduino as per the wiring diagram.

2. Programming Steps

1. Open the Arduino IDE and create a new sketch.
2. Write code to control the robot's motion:
 - **Forward Motion:** Set all motors to move in the forward direction.
 - **Stop:** Turn off all motors.
 - **Left Turn:** Set left motor to stop and right motor to move forward.
 - **Right Turn:** Set right motor to stop and left motor to move forward.
3. Upload the code to the Arduino board.

3. Testing and Execution

1. Place the robot on a flat surface.
 2. Execute the program and observe its movements:
 - Move forward.
 - Stop.
 - Turn left.
 - Turn right.
 3. Adjust motor speed using PWM if necessary for smooth motion.
-

Expected Outcomes

1. The robot moves forward, stops, and turns as per the programmed commands.

2. Demonstration of basic motion control using Arduino.
 3. Understanding the integration of hardware and software for robotic applications.
-

Thought-Provoking Questions

1. How would you modify the program to include additional movements like reverse or diagonal motion?
2. What factors affect the accuracy and smoothness of the robot's motion?
3. How can sensors be integrated to make the robot autonomous?
4. What are the limitations of using DC motors for robotic motion?
5. How can the program be optimized for energy efficiency?

प्रयोग 4:

कथन: Arduino का उपयोग करके 3-पहियों वाले मोबाइल रोबोट को आगे बढ़ने, रुकने, बाएं मुड़ने और दाएं मुड़ने के लिए एक प्रोग्राम लिखें।

उद्देश्य

Arduino पर आधारित प्रोग्राम विकसित करना और निष्पादित करना ताकि 3-पहिए वाले मोबाइल रोबोट की गति को नियंत्रित किया जा सके और रोबोट गति नियंत्रण तकनीकों को प्रदर्शित किया जा सके।

कार्यप्रणाली

1. हार्डवेयर सेटअप

- 3-पहिए वाले मोबाइल रोबोट को निम्नलिखित घटकों के साथ असेंबल करें:
 - पहियों के संचालन के लिए 3 डीसी मोटर्स।
 - मोटर ड्राइवर मॉड्यूल (L298N या समान)।
 - Arduino माइक्रोकंट्रोलर बोर्ड।
 - मोटर्स और Arduino के लिए पावर स्रोत।
- मोटर्स को मोटर ड्राइवर और Arduino से वायरिंग आरेख के अनुसार कनेक्ट करें।

2. प्रोग्रामिंग चरण

- Arduino IDE खोलें और एक नई स्केच बनाएं।
- रोबोट की गति को नियंत्रित करने के लिए कोड लिखें:
 - आगे की गति:** सभी मोटर्स को आगे की दिशा में चलने के लिए सेट करें।
 - रुकें:** सभी मोटर्स को बंद करें।
 - बाएं मुड़ें:** बाएं मोटर को बंद करें और दाएं मोटर को आगे की ओर चलाएं।
 - दाएं मुड़ें:** दाएं मोटर को बंद करें और बाएं मोटर को आगे की ओर चलाएं।
- कोड को Arduino बोर्ड में अपलोड करें।

3. परीक्षण और निष्पादन

- रोबोट को एक सपाट सतह पर रखें।
- प्रोग्राम को निष्पादित करें और उसकी गतिविधियों का निरीक्षण करें:
 - आगे बढ़ें।
 - रुकें।
 - बाएं मुड़ें।
 - दाएं मुड़ें।

3. चिकनी गति के लिए आवश्यकतानुसार PWM का उपयोग करके मोटर की गति को समायोजित करें।
-

अपेक्षित परिणाम

1. रोबोट आगे बढ़ता है, रुकता है, और प्रोग्राम किए गए कमांड्स के अनुसार मुड़ता है।
 2. Arduino का उपयोग करके बुनियादी गति नियंत्रण का प्रदर्शन।
 3. रोबोटिक अनुप्रयोगों के लिए हार्डवेयर और सॉफ्टवेयर के एकीकरण की समझ।
-

सोचने योग्य प्रश्न

1. प्रोग्राम में अतिरिक्त गतिविधियों जैसे रिवर्स या तिरछी गति को शामिल करने के लिए कैसे संशोधित किया जा सकता है?
2. PWM नियंत्रण मोटर की गति और दिशा को कैसे प्रभावित करता है?
3. तीन पहिए वाले रोबोट में संतुलन और स्थिरता सुनिश्चित करने के लिए कौन से डिज़ाइन विचार महत्वपूर्ण हैं?
4. यदि रोबोट असमान सतह पर है, तो गति नियंत्रण को कैसे अनुकूलित किया जा सकता है?
5. सेंसर का उपयोग रोबोट के स्वायत्त नेविगेशन को कैसे सक्षम कर सकता है?

Experiment No-5

Statement: Control of an autonomous line following robot using IR sensors

Objective:

To design and implement an autonomous line-following robot using IR sensors to detect and follow a predefined path. The robot should be able to make real-time decisions based on sensor inputs to stay on the track.

Apparatus/Components Required:

1. Microcontroller (Arduino Uno or similar)
 2. IR sensors (Infrared sensors)
 3. Motor driver module (L298N or similar)
 4. DC motors
 5. Chassis (robot body)
 6. Wheels
 7. Power supply (battery pack)
 8. Connecting wires
 9. Breadboard
 10. Resistors (as needed)
 11. Jumper wires
 12. Black tape or black line on a white surface for the track
-

Theory:

A line-following robot is an autonomous robot that follows a specific path, typically a black line on a white surface or a white line on a black surface. It uses IR sensors to detect the line and sends signals to the microcontroller, which then controls the motors to keep the robot on track. IR sensors work by emitting infrared light and detecting the reflected light. A black surface absorbs the IR light, while a white surface reflects it, allowing the sensor to distinguish between the two.

Working Principle:

The robot uses two or more IR sensors to detect the line. The sensors provide real-time feedback to the microcontroller, which adjusts the speed and direction of the motors accordingly. The basic control logic is as follows:

- **Both sensors on the black line:** Move forward.
 - **Left sensor on the black line, right sensor off the line:** Turn left.
 - **Right sensor on the black line, left sensor off the line:** Turn right.
 - **Both sensors off the line:** Stop or adjust to find the line.
-

Circuit Diagram:

(Include a circuit diagram showing the connections between the microcontroller, IR sensors, motor driver, and motors.)

Algorithm:

1. Initialize the microcontroller and configure the input/output pins.
 2. Read data from the IR sensors.
 3. If both sensors detect the line, move forward.
 4. If the left sensor detects the line and the right sensor does not, turn left.
 5. If the right sensor detects the line and the left sensor does not, turn right.
 6. If neither sensor detects the line, stop and adjust to find the line.
 7. Repeat the process continuously to follow the line.
-

Procedure:

1. Assemble the robot by connecting the motors to the chassis.
 2. Connect the IR sensors to the microcontroller and place them in front of the robot.
 3. Connect the motor driver module to the microcontroller and the motors.
 4. Connect the power supply to the circuit.
 5. Upload the code to the microcontroller using the Arduino IDE.
 6. Place the robot on the track and power it on.
 7. Observe the robot's behavior and adjust the code or sensors if necessary.
-

Observation Table:

Sensor Reading (Left)	Sensor Reading (Right)	Motor Action
Black	Black	Move forward
Black	White	Turn left
White	Black	Turn right
White	White	Stop or adjust

Precautions:

1. Ensure all connections are secure.
2. Use the correct voltage power supply to avoid damaging components.
3. Place the robot on a flat, clean surface.
4. Adjust sensor sensitivity if required.
5. Avoid placing the robot in direct sunlight as it may affect sensor readings.

Result:

The autonomous line-following robot was successfully controlled using IR sensors to follow a predefined path.

Conclusion:

This experiment demonstrated the use of IR sensors in controlling an autonomous line-following robot. The robot was able to detect and follow a black line on a white surface using real-time sensor inputs and motor control.

Viva Questions:

1. What is a line-following robot?
 2. How do IR sensors work?
 3. Why do we use a motor driver module?
 4. What is the role of the microcontroller in this experiment?
 5. How can you improve the accuracy of the line-following robot?
-

प्रयोग 5:

कथन: आईआर सेंसर का उपयोग करके रोबोट के बाद एक स्वायत्त लाइन का नियंत्रण

उद्देश्य:

आईआर सेंसर का उपयोग करके एक स्वायत्त रेखा-अनुसरण करने वाले रोबोट को डिजाइन और कार्यान्वित करना। रोबोट को वास्तविक समय में सेंसर इनपुट के आधार पर निर्णय लेने और ट्रैक पर बने रहने में सक्षम होना चाहिए।

आवश्यक उपकरण/घटक:

1. माइक्रोकंट्रोलर (Arduino Uno या समान)
 2. आईआर सेंसर (इन्फ्रारेड सेंसर)
 3. मोटर ड्राइवर मॉड्यूल (L298N या समान)
 4. डीसी मोटर
 5. चैसिस (रोबोट बॉडी)
 6. पहिए
 7. पावर सप्लाई (बैटरी पैक)
 8. कनेक्टिंग तार
 9. ब्रेडबोर्ड
 10. रेजिस्टर्स (आवश्यकतानुसार)
 11. जम्पर तार
 12. काले टेप या सफेद सतह पर काली रेखा (ट्रैक के लिए)
-

सिद्धांत:

एक रेखा-अनुसरण करने वाला रोबोट एक स्वायत्त रोबोट है जो एक विशेष पथ का अनुसरण करता है, आमतौर पर एक सफेद सतह पर काली रेखा या काली सतह पर सफेद रेखा। यह लाइन का पता लगाने के लिए आईआर सेंसर का उपयोग करता है और माइक्रोकंट्रोलर को सिग्नल भेजता है, जो रोबोट को ट्रैक पर बनाए रखने के लिए मोटर्स को नियंत्रित करता है। आईआर सेंसर इन्फ्रारेड लाइट उत्सर्जित करके और परावर्तित लाइट का पता लगाकर काम करते हैं। काली सतह आईआर लाइट को अवशोषित करती है, जबकि सफेद सतह इसे परावर्तित करती है, जिससे सेंसर दोनों के बीच अंतर कर सकता है।

कार्य सिद्धांत:

रोबोट रेखा का पता लगाने के लिए दो या अधिक आईआर सेंसर का उपयोग करता है। सेंसर माइक्रोकंट्रोलर को वास्तविक समय की प्रतिक्रिया प्रदान करते हैं, जो तदनुसार मोटर्स की गति और दिशा को समायोजित करता है। नियंत्रण तर्क निम्नलिखित है:

- दोनों सेंसर काली रेखा पर हैं: आगे बढ़ें।
- बायां सेंसर काली रेखा पर है, दायां सेंसर रेखा से बाहर है: बाएं मुड़ें।

- दायां सेंसर काली रेखा पर है, बायां सेंसर रेखा से बाहर है: दाएं मुड़ें।
- दोनों सेंसर रेखा से बाहर हैं: रोकें या रेखा खोजने के लिए समायोजित करें।

सर्किट आरेख:

(माइक्रोकंट्रोलर, आईआर सेंसर, मोटर ड्राइवर और मोटर्स के बीच कनेक्शन को दर्शाने वाला सर्किट आरेख शामिल करें।)

एल्गोरिदम:

1. माइक्रोकंट्रोलर को प्रारंभ करें और इनपुट/आउटपुट पिन कॉन्फिगर करें।
2. आईआर सेंसर से डेटा पढ़ें।
3. यदि दोनों सेंसर लाइन का पता लगाते हैं, तो आगे बढ़ें।
4. यदि बायां सेंसर लाइन का पता लगाता है और दायां सेंसर नहीं करता है, तो बाएं मुड़ें।
5. यदि दायां सेंसर लाइन का पता लगाता है और बायां सेंसर नहीं करता है, तो दाएं मुड़ें।
6. यदि कोई भी सेंसर लाइन का पता नहीं लगाता है, तो रुकें और लाइन खोजने के लिए समायोजित करें।
7. लाइन का अनुसरण करने के लिए इस प्रक्रिया को लगातार दोहराएं।

प्रक्रिया:

1. रोबोट को चेसिस से जोड़कर असेंबल करें।
 2. आईआर सेंसर को माइक्रोकंट्रोलर से कनेक्ट करें और उन्हें रोबोट के सामने रखें।
 3. मोटर ड्राइवर मॉड्यूल को माइक्रोकंट्रोलर और मोटर्स से कनेक्ट करें।
 4. सर्किट को पावर सप्लाई से कनेक्ट करें।
 5. Arduino IDE का उपयोग करके कोड को माइक्रोकंट्रोलर में अपलोड करें।
 6. रोबोट को ट्रैक पर रखें और इसे चालू करें।
 7. रोबोट के व्यवहार का निरीक्षण करें और यदि आवश्यक हो तो कोड या सेंसर समायोजित करें।
-

पर्यवेक्षण तालिका:

सेंसर रीडिंग (बायां)	सेंसर रीडिंग (दायां)	मोटर क्रिया
काला	काला	आगे बढ़ें
काला	सफेद	बाएं मुड़ें
सफेद	काला	दाएं मुड़ें
सफेद	सफेद	रोकें या समायोजित करें

सावधानियां:

1. सुनिश्चित करें कि सभी कनेक्शन सुरक्षित हैं।

2. घटकों को नुकसान से बचाने के लिए सही वोल्टेज पावर सप्लाई का उपयोग करें।
 3. रोबोट को सपाट, साफ सतह पर रखें।
 4. यदि आवश्यक हो तो सेंसर की संवेदनशीलता समायोजित करें।
 5. सेंसर रीडिंग को प्रभावित करने से बचने के लिए रोबोट को सीधी धूप में न रखें।
-

परिणाम:

स्वायत्त रेखा-अनुसरण करने वाले रोबोट को आईआर सेंसर का उपयोग करके सफलतापूर्वक नियंत्रित किया गया और उसने पूर्वनिर्धारित पथ का अनुसरण किया।

निष्कर्ष:

इस प्रयोग ने आईआर सेंसर का उपयोग करके एक स्वायत्त रेखा-अनुसरण करने वाले रोबोट के नियंत्रण का प्रदर्शन किया। रोबोट वास्तविक समय के सेंसर इनपुट और मोटर नियंत्रण का उपयोग करके सफेद सतह पर काली रेखा का पता लगाने और उसका अनुसरण करने में सक्षम था।

वाइवा प्रश्न:

1. रेखा-अनुसरण करने वाला रोबोट क्या है?
 2. आईआर सेंसर कैसे काम करते हैं?
 3. हम मोटर ड्राइवर मॉड्यूल का उपयोग क्यों करते हैं?
 4. इस प्रयोग में माइक्रोकंट्रोलर की भूमिका क्या है?
 5. रेखा-अनुसरण करने वाले रोबोट की सटीकता को आप कैसे सुधार सकते हैं?
-

Experiment No-6

Statement: Control of an autonomous obstacle avoidance robot using ultrasonic sensor

Objective:

To design and implement an autonomous obstacle avoidance robot using an ultrasonic sensor. The robot should be capable of adjusting its direction to avoid obstacles in its path.

Required Components/Materials:

1. Microcontroller (Arduino Uno or equivalent)
 2. Ultrasonic Sensor (HC-SR04)
 3. Motor Driver Module (L298N or equivalent)
 4. DC Motors
 5. Chassis (Robot Body)
 6. Wheels
 7. Power Supply (Battery Pack)
 8. Connecting Wires
 9. Breadboard
 10. Jumper Wires
-

Theory:

An obstacle avoidance robot uses an ultrasonic sensor to detect obstacles in its path and changes its direction to avoid collisions. The ultrasonic sensor emits sound waves and calculates the time taken for these waves to hit an object and bounce back. This calculation helps in determining the distance between the sensor and the object.

Working Principle:

1. The ultrasonic sensor detects the presence of any obstacle in front.
 2. If the obstacle is at a distance greater than a set threshold (e.g., 20 cm), the robot continues to move forward.
 3. If the obstacle is within the threshold, the robot changes its direction (either left or right).
 4. The process repeats continuously to ensure safe movement.
-

Circuit Diagram:

(Include a circuit diagram showing the connections between the microcontroller, ultrasonic sensor, motor driver, and motors.)

Algorithm:

1. Initialize the microcontroller and configure pins.
 2. Measure the distance using the ultrasonic sensor.
 3. If the distance is greater than 20 cm, move the robot forward.
 4. If the distance is 20 cm or less, turn the robot left or right.
 5. Repeat the process continuously.
-

Flowchart:

(Include a flowchart representing the obstacle avoidance algorithm using the ultrasonic sensor.)

Procedure:

1. Connect the ultrasonic sensor and motors to the microcontroller.
 2. Connect the motor driver module to the microcontroller and motors.
 3. Power the circuit using a suitable power supply.
 4. Upload the code to the microcontroller using Arduino IDE.
 5. Place the robot on a flat surface and switch it on.
 6. Observe the robot's behavior and adjust the code if necessary.
-

Observation Table:

Distance (cm)	Motor Action
More than 20	Move Forward
20 or Less	Turn Left/Right

Precautions:

1. Ensure all connections are secure and correct.
 2. Use a proper voltage power supply to avoid damage to components.
 3. Make sure the ultrasonic sensor is properly aligned.
 4. Test the robot on a flat and obstacle-rich surface.
-

Result:

The obstacle avoidance robot was successfully controlled using an ultrasonic sensor. The robot adjusted its direction to avoid obstacles.

Conclusion:

In this experiment, the control of an autonomous obstacle avoidance robot using an ultrasonic sensor was demonstrated. The robot was able to calculate the distance in real-time and adjust its direction to avoid obstacles in its path.

Viva Questions:

1. How does an ultrasonic sensor work?
2. What is the difference between an ultrasonic sensor and an IR sensor?
3. Why is a motor driver module used?
4. Where can obstacle avoidance robots be applied?
5. How can you improve the accuracy of the robot?

प्रयोग 6:

कथन: अल्ट्रासोनिक सेंसर का उपयोग करके एक स्वायत्त बाधा निवारण रोबोट का नियंत्रण

उद्देश्य:

अल्ट्रासोनिक सेंसर का उपयोग करके एक स्वायत्त बाधा-परिहार रोबोट को डिजाइन और कार्यान्वित करना। यह रोबोट अपनी दिशा समायोजित करके सामने आने वाली बाधाओं से बचने में सक्षम होना चाहिए।

आवश्यक उपकरण/सामग्री:

- माइक्रोकंट्रोलर (Arduino Uno या समकक्ष)
 - अल्ट्रासोनिक सेंसर (HC-SR04)
 - मोटर ड्राइवर मॉड्यूल (L298N या समकक्ष)
 - डीसी मोटर
 - चेसिस (रोबोट बॉडी)
 - पहिए
 - पावर सप्लाई (बैटरी पैक)
 - कनेक्टिंग तार
 - ब्रेडबोर्ड
 - जम्पर तार
-

सिद्धांत:

एक बाधा-परिहार करने वाला रोबोट अल्ट्रासोनिक सेंसर का उपयोग करके अपनी राह में आने वाली बाधाओं का पता लगाता है और टकराव से बचने के लिए अपनी दिशा बदलता है। अल्ट्रासोनिक सेंसर ध्वनि तरंगें भेजता है और उस समय की गणना करता है जो इन तरंगों को किसी वस्तु से टकराकर वापस लौटने में लगता है। यह गणना सेंसर और वस्तु के बीच की दूरी निर्धारित करने में मदद करती है।

कार्य सिद्धांत:

- अल्ट्रासोनिक सेंसर सामने की ओर किसी भी बाधा की उपस्थिति का पता लगाता है।
 - यदि बाधा एक निर्धारित सीमा (जैसे 20 सेमी) से अधिक दूरी पर है, तो रोबोट आगे बढ़ता रहता है।
 - यदि बाधा निर्धारित सीमा के भीतर है, तो रोबोट अपनी दिशा (बाएं या दाएं) बदलता है।
 - यह प्रक्रिया तब तक जारी रहती है जब तक रोबोट बाधाओं से बचकर आगे बढ़ता रहता है।
-

सर्किट आरेख:

(माइक्रोकंट्रोलर, अल्ट्रासोनिक सेंसर, मोटर ड्राइवर और मोटरों के बीच कनेक्शन को दर्शाने वाला सर्किट आरेख शामिल करें।)

एल्गोरिदम:

- माइक्रोकंट्रोलर को प्रारंभ करें और पिन कॉन्फिगर करें।
- अल्ट्रासोनिक सेंसर से दूरी की गणना करें।
- यदि दूरी 20 सेमी से अधिक है, तो रोबोट आगे बढ़ता है।
- यदि दूरी 20 सेमी या उससे कम है, तो रोबोट बाएं या दाएं मुड़ता है।
- इस प्रक्रिया को लगातार दोहराएं।

फ्लोचार्ट:

(अल्ट्रासोनिक सेंसर आधारित बाधा-परिहार एल्गोरिदम को दर्शाने वाला फ्लोचार्ट शामिल करें।)

प्रक्रिया:

- अल्ट्रासोनिक सेंसर और मोटरों को माइक्रोकंट्रोलर से कनेक्ट करें।
- मोटर ड्राइवर मॉड्यूल को माइक्रोकंट्रोलर और मोटरों से कनेक्ट करें।
- सर्किट को पावर सप्लाई से कनेक्ट करें।
- Arduino IDE का उपयोग करके कोड को माइक्रोकंट्रोलर में अपलोड करें।
- रोबोट को एक सपाट सतह पर रखें और इसे चालू करें।
- रोबोट के व्यवहार का निरीक्षण करें और यदि आवश्यक हो तो कोड को समायोजित करें।

पर्यवेक्षण तालिका:

दूरी (सेमी)	मोटर क्रिया
20 से अधिक	आगे बढ़ें
20 या कम	बाएं/दाएं मुड़ें

सावधानियां:

- सभी कनेक्शन सही और सुरक्षित तरीके से करें।
- घटकों को नुकसान से बचाने के लिए उचित वोल्टेज पावर सप्लाई का उपयोग करें।
- अल्ट्रासोनिक सेंसर को सही दिशा में इंगित करें।
- रोबोट को सपाट और बाधा युक्त सतह पर परीक्षण करें।

परिणाम:

अल्ट्रासोनिक सेंसर का उपयोग करके बाधा-परिहार रोबोट को सफलतापूर्वक नियंत्रित किया गया। रोबोट ने बाधाओं से बचने के लिए अपनी दिशा समायोजित की।

निष्कर्ष:

इस प्रयोग में अल्ट्रासोनिक सेंसर का उपयोग करके एक स्वायत्त बाधा-परिहार रोबोट के नियंत्रण का प्रदर्शन किया गया। रोबोट ने वास्तविक समय में दूरी की गणना की और सामने आने वाली बाधाओं से बचने के लिए अपनी दिशा बदलने में सक्षम था।

वाइवा प्रश्न:

1. अल्ट्रासोनिक सेंसर कैसे काम करता है?
 2. अल्ट्रासोनिक सेंसर और आईआर सेंसर में क्या अंतर है?
 3. मोटर ड्राइवर मॉड्यूल का उपयोग क्यों किया जाता है?
 4. बाधा-परिहार रोबोट का उपयोग कहाँ किया जा सकता है?
 5. आप रोबोट की सटीकता कैसे सुधार सकते हैं?
-

Experiment No-7

Statement: Write the program using Arduino for Human Following Robot

Objective:

To develop a human-following robot using Arduino that can detect and follow a person autonomously.

Required Components/Materials:

1. Arduino Uno
 2. Ultrasonic Sensors (2 or more)
 3. Motor Driver Module (L298N or equivalent)
 4. DC Motors
 5. Chassis (Robot Body)
 6. Wheels
 7. Power Supply (Battery Pack)
 8. Breadboard
 9. Connecting Wires
 10. Jumper Wires
-

Theory:

A human-following robot uses ultrasonic sensors to detect the position of a person and follows them by adjusting its direction and speed. The sensors calculate the distance to the person, and the microcontroller processes this data to control the motors accordingly.

Working Principle:

1. The robot uses two ultrasonic sensors to measure distances from the left and right sides.
 2. Based on the difference in the distance values, the robot adjusts its direction to follow the person.
 3. If the person moves left, the robot turns left; if the person moves right, the robot turns right.
 4. The robot continuously measures distances to maintain a safe following distance.
-

Circuit Diagram:

(Include a circuit diagram showing the connections between the Arduino, ultrasonic sensors, motor driver, and motors.)

Algorithm:

1. Initialize the Arduino and configure the ultrasonic sensors and motor pins.
 2. Measure the distance from both the left and right ultrasonic sensors.
 3. Compare the distances:
 - If the left sensor detects a shorter distance, turn left.
 - If the right sensor detects a shorter distance, turn right.
 - If both distances are similar, move forward.
 4. Continuously adjust direction based on sensor readings.
-

Flowchart:

(Include a flowchart representing the human-following robot algorithm.)

Procedure:

1. Connect the ultrasonic sensors and motors to the Arduino.
 2. Connect the motor driver module to the motors and the Arduino.
 3. Power the circuit using a suitable power supply.
 4. Upload the code to the Arduino using Arduino IDE.
 5. Place the robot on a flat surface and switch it on.
 6. Observe the robot's behavior and adjust the code if necessary.
-

Observation Table:

Left Sensor Distance (cm)	Right Sensor Distance (cm)	Motor Action
Less than Right Distance	Greater than Left Distance	Turn Left
Greater than Right Distance	Less than Left Distance	Turn Right
Equal Distances	Equal Distances	Move Forward

Precautions:

1. Ensure all connections are secure and correct.
 2. Use a proper voltage power supply to avoid damage to components.
 3. Make sure the ultrasonic sensors are properly aligned.
 4. Test the robot on a flat and obstacle-free surface.
-

Result:

The human-following robot was successfully controlled using Arduino. The robot adjusted its direction and maintained a safe distance while following a person.

Conclusion:

In this experiment, a human-following robot was developed using Arduino and ultrasonic sensors. The robot was able to detect and follow a person by adjusting its direction based on sensor readings.

Viva Questions:

1. How does an ultrasonic sensor work?
2. What are the key components used in a human-following robot?
3. Why is a motor driver module necessary?
4. What are the real-life applications of a human-following robot?
5. How can you improve the accuracy of the robot's movement?

प्रयोग 7:

कथन: ह्यूमन फॉलोइंग रोबोट के लिए Arduino का उपयोग करके प्रोग्राम लिखें

उद्देश्य:

Arduino का उपयोग करके एक मानव-अनुसरण करने वाले रोबोट को विकसित करना, जो स्वायत्त रूप से किसी व्यक्ति का पता लगा सके और उसका अनुसरण कर सके।

आवश्यक उपकरण/सामग्री:

1. Arduino Uno
 2. अल्ट्रासोनिक सेंसर (2 या अधिक)
 3. मोटर ड्राइवर मॉड्यूल (L298N या समकक्ष)
 4. डीसी मोटर
 5. चैसिस (रोबोट बॉडी)
 6. पहिए
 7. पावर सप्लाई (बैटरी पैक)
 8. ब्रेडबोर्ड
 9. कनेक्टिंग वायर
 10. जम्पर वायर
-

सिद्धांत:

एक मानव-अनुसरण करने वाला रोबोट अल्ट्रासोनिक सेंसर का उपयोग करता है ताकि किसी व्यक्ति की स्थिति का पता लगाया जा सके और उसकी दिशा और गति को समायोजित करके उसका अनुसरण किया जा सके। सेंसर व्यक्ति की दूरी की गणना करते हैं और माइक्रोकंट्रोलर इस डेटा को प्रोसेस करके मोटर्स को नियंत्रित करता है।

कार्य सिद्धांत:

1. रोबोट बाएँ और दाएँ दोनों तरफ से दूरी मापने के लिए दो अल्ट्रासोनिक सेंसर का उपयोग करता है।
 2. दूरी के मानों में अंतर के आधार पर, रोबोट अपनी दिशा को समायोजित करता है ताकि वह व्यक्ति का अनुसरण कर सके।
 3. यदि व्यक्ति बाईं ओर जाता है, तो रोबोट बाईं ओर मुड़ता है।
 4. यदि व्यक्ति दाईं ओर जाता है, तो रोबोट दाईं ओर मुड़ता है।
 5. रोबोट लगातार दूरी की माप करता रहता है ताकि सुरक्षित दूरी बनाए रखी जा सके।
-

सर्किट आरेख:

(Arduino, अल्ट्रासोनिक सेंसर, मोटर ड्राइवर, और मोटरों के बीच कनेक्शन को दर्शाने वाला सर्किट आरेख शामिल करें।)

एल्गोरिदम:

1. Arduino को प्रारंभ करें और अल्ट्रासोनिक सेंसर और मोटर पिन को कॉन्फ़िगर करें।
2. बाएँ और दाएँ दोनों अल्ट्रासोनिक सेंसर से दूरी मापें।
3. दूरी की तुलना करें:
 - यदि बाएँ सेंसर कम दूरी का पता लगाता है, तो बाएँ मुड़ें।
 - यदि दाएँ सेंसर कम दूरी का पता लगाता है, तो दाएँ मुड़ें।
 - यदि दोनों दूरी समान हैं, तो आगे बढ़ें।
4. सेंसर रीडिंग के आधार पर दिशा को लगातार समायोजित करें।

फ्लोचार्ट:

(मानव-अनुसरण करने वाले रोबोट के एल्गोरिदम को दर्शाने वाला फ्लोचार्ट शामिल करें।)

प्रक्रिया:

1. अल्ट्रासोनिक सेंसर और मोटरों को Arduino से कनेक्ट करें।
2. मोटर ड्राइवर मॉड्यूल को Arduino और मोटरों से कनेक्ट करें।
3. सर्किट को पावर सप्लाई से कनेक्ट करें।
4. Arduino IDE का उपयोग करके कोड को Arduino में अपलोड करें।
5. रोबोट को एक सपाट सतह पर रखें और इसे चालू करें।
6. रोबोट के व्यवहार का निरीक्षण करें और यदि आवश्यक हो तो कोड को समायोजित करें।

पर्यवेक्षण तालिका:

बाएँ सेंसर की दूरी (सेमी)	दाएँ सेंसर की दूरी (सेमी)	मोटर की क्रिया
दाएँ दूरी से कम	बाएँ दूरी से अधिक	बाएँ मुड़ें
बाएँ दूरी से अधिक	दाएँ दूरी से कम	दाएँ मुड़ें
समान दूरी	समान दूरी	आगे बढ़ें

सावधानियाँ:

1. सभी कनेक्शन को सही और सुरक्षित तरीके से करें।
2. घटकों को नुकसान से बचाने के लिए उचित वोल्टेज पावर सप्लाई का उपयोग करें।
3. अल्ट्रासोनिक सेंसर को सही दिशा में इंगित करें।

4. रोबोट को सपाट और बाधा रहित सतह पर परीक्षण करें।
-

परिणाम:

मानव-अनुसरण करने वाले रोबोट को सफलतापूर्वक Arduino का उपयोग करके नियंत्रित किया गया। रोबोट ने व्यक्ति का अनुसरण करते हुए दिशा को समायोजित किया और सुरक्षित दूरी बनाए रखी।

निष्कर्ष:

इस प्रयोग में, Arduino और अल्ट्रासोनिक सेंसर का उपयोग करके एक मानव-अनुसरण करने वाले रोबोट का विकास किया गया। रोबोट व्यक्ति की दूरी को मापकर और सेंसर रीडिंग के आधार पर दिशा को समायोजित करके व्यक्ति का अनुसरण करने में सक्षम था।

वाइवा प्रश्न:

1. अल्ट्रासोनिक सेंसर कैसे काम करता है?
2. मानव-अनुसरण करने वाले रोबोट में उपयोग किए जाने वाले मुख्य घटक कौन से हैं?
3. मोटर ड्राइवर मॉड्यूल की आवश्यकता क्यों होती है?
4. मानव-अनुसरण करने वाले रोबोट का वास्तविक जीवन में कहाँ उपयोग किया जा सकता है?
5. आप रोबोट की सटीकता को कैसे सुधार सकते हैं?

Experiment No-8

Statement: Development of control algorithm for 4-DOF robotic Manipulator

Objective:

To develop a control algorithm for a 4-degree-of-freedom (4-DOF) robotic manipulator using appropriate sensors, actuators, and microcontroller.

Apparatus/Components Required:

1. Arduino Uno or any microcontroller
 2. 4-DOF robotic manipulator kit
 3. Servo motors (4 units)
 4. Power supply
 5. Sensors (optional based on application)
 6. Breadboard
 7. Connecting wires
 8. Jumper wires
 9. Laptop with Arduino IDE
-

Theory:

A 4-DOF robotic manipulator is a robotic arm with four independent joints, allowing it to move and rotate in different directions. The degrees of freedom determine how versatile the robotic arm is in performing complex tasks. The control algorithm involves managing the movements of each joint to achieve the desired position and orientation of the end effector (the tool attached to the end of the robotic arm).

The control algorithm calculates the required joint angles based on the desired end-effector position using kinematic equations. The Arduino or microcontroller sends signals to the servo motors, which control each joint's rotation.

Working Principle:

1. The robotic manipulator has four joints, each driven by a servo motor.
 2. The control algorithm calculates the necessary joint angles to move the manipulator to a specific position.
 3. The microcontroller sends signals to the servo motors to achieve the desired movement.
 4. Sensors can be used to provide feedback for precise control.
-

Circuit Diagram:

Include a diagram showing the connection between the Arduino, servo motors, power supply, and other components.

Algorithm:

- 1. Start the Arduino and initialize all servo motor pins.
- 2. Define the initial positions of all joints.
- 3. Receive the desired end-effector position from the user.
- 4. Calculate the joint angles using kinematic equations.
- 5. Move each joint to the calculated angle.
- 6. Continuously monitor the end-effector position and adjust the joints as necessary.
- 7. End.

Flowchart:

Include a flowchart depicting the steps of the control algorithm from receiving input to controlling the robotic manipulator's movements.

Procedure:

- 1. Connect the servo motors to the Arduino as per the circuit diagram.
- 2. Connect the power supply to the Arduino and motors.
- 3. Upload the sample code to the Arduino using the Arduino IDE.
- 4. Observe the movement of the robotic manipulator.
- 5. Adjust the code to achieve the desired movements of the manipulator.

Observation Table:

Joint	Initial Angle (degrees)	Final Angle (degrees)	Movement Description
Joint 1	90	45	Rotated left
Joint 2	90	60	Rotated upward
Joint 3	90	75	Rotated forward
Joint 4	90	90	No change

Precautions:

- 1. Ensure all connections are secure and correct.
- 2. Use a stable power supply to avoid damage to the motors.
- 3. Avoid sudden movements that can damage the robotic manipulator.
- 4. Ensure the Arduino is properly programmed before starting the experiment.

Result:

The control algorithm for the 4-DOF robotic manipulator was successfully developed and tested. The manipulator was able to move to the desired positions based on the control inputs.

Conclusion:

In this experiment, a control algorithm was developed for a 4-DOF robotic manipulator using Arduino. The manipulator successfully performed the desired movements by calculating and applying the appropriate joint angles.

Viva Questions:

1. What is a 4-DOF robotic manipulator?
2. Why is a control algorithm important for a robotic manipulator?
3. How are servo motors controlled using Arduino?
4. What are the practical applications of robotic manipulators?
5. How can the accuracy of a robotic manipulator be improved?

प्रयोग 8:

कथन: 4-डीओएफ रोबोटिक मैनिपुलेटर के लिए नियंत्रण एल्गोरिदम का विकास

उद्देश्य:

4-डिग्री-ऑफ-फ्रीडम (4-DOF) रोबोटिक मैनिपुलेटर के लिए एक नियंत्रण एल्गोरिदम विकसित करना, जिसमें उपयुक्त सेंसर, एक्ज्यूटर और माइक्रोकंट्रोलर का उपयोग किया जाए।

आवश्यक उपकरण/सामग्री:

1. Arduino Uno या कोई अन्य माइक्रोकंट्रोलर
 2. 4-DOF रोबोटिक मैनिपुलेटर किट
 3. सर्वो मोटर्स (4 यूनिट)
 4. पावर सप्लाय
 5. सेंसर (आवश्यकता के आधार पर)
 6. ब्रेडबोर्ड
 7. कनेक्टिंग वायर
 8. जम्पर वायर
 9. लैपटॉप जिसमें Arduino IDE हो
-

सिद्धांत:

4-DOF रोबोटिक मैनिपुलेटर एक रोबोटिक आर्म है जिसमें चार स्वतंत्र जोड़ होते हैं, जो इसे विभिन्न दिशाओं में घूमने और हिलने की अनुमति देते हैं। स्वतंत्रता की डिग्री निर्धारित करती है कि रोबोटिक आर्म कितने जटिल कार्य कर सकता है। नियंत्रण एल्गोरिदम का उपयोग प्रत्येक जोड़ की गति को प्रबंधित करने के लिए किया जाता है ताकि अंतिम प्रभावक (end effector) की वांछित स्थिति और उन्मुखीकरण प्राप्त किया जा सके।

यह नियंत्रण एल्गोरिदम वांछित अंतिम प्रभावक की स्थिति के आधार पर आवश्यक जोड़ कोणों की गणना करता है। Arduino या माइक्रोकंट्रोलर सर्वो मोटर्स को सिग्नल भेजता है, जो प्रत्येक जोड़ की घूर्णन को नियंत्रित करता है।

कार्य सिद्धांत:

1. रोबोटिक मैनिपुलेटर में चार जोड़ होते हैं, जिनमें से प्रत्येक को एक सर्वो मोटर द्वारा संचालित किया जाता है।
 2. नियंत्रण एल्गोरिदम आवश्यक जोड़ कोणों की गणना करता है ताकि मैनिपुलेटर एक विशिष्ट स्थिति में जा सके।
 3. माइक्रोकंट्रोलर सर्वो मोटर्स को सिग्नल भेजता है ताकि वांछित गति प्राप्त हो सके।
 4. सटीक नियंत्रण के लिए सेंसर का उपयोग किया जा सकता है।
-

सर्किट आरेख:

Arduino, सर्वो मोटर्स, पावर सप्लाई और अन्य घटकों के बीच कनेक्शन को दर्शाने वाला एक आरेख शामिल करें।

एल्गोरिदम:

1. Arduino प्रारंभ करें और सभी सर्वो मोटर पिन को इनिशियलाइज करें।
2. सभी जोड़ की प्रारंभिक स्थिति को परिभाषित करें।
3. उपयोगकर्ता से वांछित अंतिम प्रभावक की स्थिति प्राप्त करें।
4. काइनेमेटिक समीकरणों का उपयोग करके जोड़ कोणों की गणना करें।
5. प्रत्येक जोड़ को गणना किए गए कोण पर ले जाएं।
6. अंतिम प्रभावक की स्थिति की निरंतर निगरानी करें और आवश्यकतानुसार जोड़ों को समायोजित करें।
7. समाप्त करें।

फ्लोचार्ट:

नियंत्रण एल्गोरिदम के चरणों को इनपुट प्राप्त करने से लेकर रोबोटिक मैनिपुलेटर की गति को नियंत्रित करने तक दिखाने वाला फ्लोचार्ट शामिल करें।

प्रक्रिया:

1. सर्किट आरेख के अनुसार सर्वो मोटर्स को Arduino से कनेक्ट करें।
2. पावर सप्लाई को Arduino और मोटर्स से कनेक्ट करें।
3. Arduino IDE का उपयोग करके कोड को Arduino में अपलोड करें।
4. रोबोटिक मैनिपुलेटर की गति का अवलोकन करें।
5. मैनिपुलेटर की वांछित गति प्राप्त करने के लिए कोड को समायोजित करें।

पर्यवेक्षण तालिका:

जोड़	प्रारंभिक कोण (डिग्री)	अंतिम कोण (डिग्री)	गति का वर्णन
जोड़ 1	90	45	बाएँ घूर्णन
जोड़ 2	90	60	ऊपर की ओर घूर्णन
जोड़ 3	90	75	आगे की ओर घूर्णन
जोड़ 4	90	90	कोई परिवर्तन नहीं

सावधानियाँ:

1. सभी कनेक्शन को सही और सुरक्षित रूप से करें।

2. मोटर्स को नुकसान से बचाने के लिए एक स्थिर पावर सप्लाय का उपयोग करें।
 3. अचानक गति से बचें जो रोबोटिक मैनिपुलेटर को नुकसान पहुंचा सकती है।
 4. Arduino को सही ढंग से प्रोग्राम करें और फिर प्रयोग शुरू करें।
-

परिणाम:

4-DOF रोबोटिक मैनिपुलेटर के लिए नियंत्रण एल्गोरिदम को सफलतापूर्वक विकसित और परीक्षण किया गया। मैनिपुलेटर ने नियंत्रण इनपुट के आधार पर वांछित स्थिति प्राप्त की।

निष्कर्ष:

इस प्रयोग में, Arduino का उपयोग करके 4-DOF रोबोटिक मैनिपुलेटर के लिए एक नियंत्रण एल्गोरिदम विकसित किया गया। मैनिपुलेटर ने आवश्यक जोड़ कोणों की गणना करके और उन्हें लागू करके वांछित गति प्राप्त की।

वाइवा प्रश्न:

1. 4-DOF रोबोटिक मैनिपुलेटर क्या है?
2. रोबोटिक मैनिपुलेटर के लिए नियंत्रण एल्गोरिदम क्यों महत्वपूर्ण है?
3. Arduino का उपयोग करके सर्वो मोटर्स को कैसे नियंत्रित किया जाता है?
4. रोबोटिक मैनिपुलेटर के व्यावहारिक अनुप्रयोग क्या हैं?
5. रोबोटिक मैनिपुलेटर की सटीकता को कैसे सुधार सकते हैं?

Experiment No-9

Statement: Calculate the Gait angles and dynamic balance margin of 16 DOF Humanoid Robot

Objective:

To calculate the gait angles and dynamic balance margin of a 16 Degree of Freedom (DOF) humanoid robot. This involves determining the angles between various body segments during walking and assessing the robot's ability to maintain balance while moving.

Prerequisites:

- Basic understanding of humanoid robot kinematics.
- Familiarity with gait analysis concepts.
- Understanding of dynamic balance and stability criteria.

Equipment Required:

- 16 DOF humanoid robot (or simulation software for humanoid robots).
 - MATLAB or Python (for computations and simulations).
 - Motion capture system or joint angle measurement tool (optional for real robot).
 - Force sensors or simulators to assess dynamic stability (optional for real robot).
-

Theory:

1. Gait Angles Calculation:

- The humanoid robot has 16 DOF, with each degree corresponding to a joint. For example, the joints may correspond to the hip, knee, ankle, elbow, wrist, etc., in a typical humanoid configuration.
- The gait cycle of the humanoid robot includes phases such as stance phase and swing phase, where each joint angle varies.
- Gait angles are the angles between adjacent body segments (e.g., thigh and shin, upper arm and forearm, etc.) calculated at different points in the gait cycle.

2. Dynamic Balance Margin:

- Dynamic balance refers to the robot's ability to maintain stability while in motion, accounting for forces, accelerations, and momentum.
 - The dynamic balance margin is the robot's ability to prevent a fall while walking, determined by the position of the center of mass (COM) and the support polygon (foot contact points).
 - The distance between the COM and the edge of the support polygon (e.g., foot contact points) is used to measure balance margin. A larger margin indicates better stability.
-

Procedure:

Step 1: Kinematic Model Setup

1. Robot Configuration:

- Define the humanoid robot's joint configurations (positions, orientations).
- Input the number of degrees of freedom (16 DOF) corresponding to the robot's joints, including arms, legs, and trunk.
- Represent the robot in terms of its link lengths and joint types (revolute or prismatic).

2. Joint Angle Definitions:

- Define joint angles for each degree of freedom (e.g., hip, knee, ankle for legs, shoulder, elbow for arms).
- Initialize the joint angles at a neutral position (standing posture).

Step 2: Simulation of Walking Cycle

1. Gait Phase Segmentation:

- Break down the walking cycle into key phases: initial contact, mid-stance, terminal stance, and swing phase.
- Define the duration for each phase based on the robot's walking speed and cadence.

2. Joint Angle Variation:

- For each phase of the gait cycle, determine how each joint angle changes. For example:
 - Hip flexion/extension
 - Knee flexion/extension
 - Ankle dorsiflexion/plantarflexion
- Use kinematic equations or forward kinematics to calculate the position and angle of each joint for each phase of the gait.

Step 3: Gait Angle Calculation

1. Forward Kinematics:

- Use forward kinematics to calculate the joint angles at different phases of the gait cycle. For a humanoid robot, this typically involves calculating the position and orientation of each link (thigh, shin, foot, etc.).
- Calculate angles at key joints (e.g., hip, knee, ankle) for each phase of the gait.

2. Angle Data Extraction:

- Use inverse kinematics (if needed) to extract joint angles for each limb at any given time.
- Visualize these angles over time as the robot walks.

Step 4: Dynamic Balance Margin Calculation

1. Center of Mass (COM) Calculation:

- For each gait phase, calculate the position of the robot's COM. The COM can be calculated as the weighted average of the positions of all the links in the humanoid robot.
- Apply the formula for COM:

$$COM = \frac{\sum_i m_i \cdot x_i}{\sum_i m_i}$$

where m_i is the mass of the i -th link and x_i is its position.

2. Support Polygon:

- Identify the contact points of the robot's feet with the ground at any given time.
- Define the support polygon, which is the convex hull of these contact points.

3. Balance Margin Calculation:

- Calculate the distance from the COM to the edge of the support polygon. A typical method is to calculate the distance from the COM to the center of the support polygon and determine if the COM is inside the polygon.
- If the COM falls outside the polygon, the robot is at risk of falling.

4. Dynamic Balance Evaluation:

- Evaluate the robot's dynamic balance by checking if the COM remains within the support polygon during the gait cycle.
- If the COM crosses the edge of the polygon during the swing phase, this indicates a potential loss of balance.

Step 5: Data Analysis

1. Gait Angles Plotting:

- Plot the calculated gait angles over time for key joints (hip, knee, ankle) in each phase of the walking cycle.
- Analyze the symmetry, range, and patterns of motion of the joints.

2. Balance Margin Evaluation:

- Plot the dynamic balance margin as a function of time or gait phase.
- Evaluate the robot's ability to maintain stability by observing if the margin remains positive and if the COM stays within the support polygon.

Results:

• Gait Angles:

- Present the calculated joint angles (hip, knee, ankle) during different phases of the gait cycle (stance, swing).

• Dynamic Balance Margin:

- Provide a graph showing the dynamic balance margin at each point in the gait cycle.
- Indicate any moments where the robot is close to losing balance.

Discussion:

- Discuss the results of the gait angle calculations, including the range of motion of each joint.
- Analyze the dynamic balance margin and the robot's ability to maintain stability during the walking cycle.
- Suggest potential improvements for the humanoid robot's balance and gait based on the analysis.

Conclusion:

This experiment allows for the calculation of gait angles and dynamic balance margin for a humanoid robot, offering insight into the robot's walking efficiency and stability. By analyzing joint angles and the balance margin, you can assess the robot's performance and identify areas for improvement.

References:

1. Spong, M. W., & Vidyasagar, M. (2008). *Robot Dynamics and Control*. Wiley.
2. Khatib, O. (1987). *A unified approach for motion and force control of robot manipulators: The operational space formulation*. IEEE Journal of Robotics and Automation.
3. Raibert, M. H., & Blankespoor, K. (2008). *BigDog, the rough-terrain robot: An overview*. Journal of Field Robotics.

प्रयोग 9:

कथन: 16 डीओएफ ह्यूमनॉइड रोबोट के चाल कोण और गतिशील संतुलन मार्जिन की गणना करें

उद्देश्य:

16 डिग्री ऑफ फ्रीडम (DOF) ह्यूमनॉइड रोबोट के गेट एंगल और डायनेमिक बैलेंस मार्जिन की गणना करना। इसमें चलने के दौरान विभिन्न शरीर के खंडों के बीच कोणों की गणना करना और चलते समय रोबोट की संतुलन बनाए रखने की क्षमता का आकलन करना शामिल है।

पूर्व आवश्यकताएँ:

- ह्यूमनॉइड रोबोट के काइनेमेटिक्स की बुनियादी समझ।
- गेट विश्लेषण की अवधारणाओं से परिचित होना।
- डायनेमिक बैलेंस और स्थिरता मानदंडों की समझ।

आवश्यक उपकरण:

- 16 DOF ह्यूमनॉइड रोबोट (या ह्यूमनॉइड रोबोट के लिए सिमुलेशन सॉफ्टवेयर)।
- MATLAB या Python (गणनाओं और सिमुलेशन के लिए)।
- मोशन कैप्चर सिस्टम या जॉइंट एंगल मापने का उपकरण (वास्तविक रोबोट के लिए वैकल्पिक)।
- डायनेमिक स्थिरता का मूल्यांकन करने के लिए बल सेंसर या सिमुलेटर (वास्तविक रोबोट के लिए वैकल्पिक)।

सिद्धांत:

1. गेट एंगल्स की गणना:

- ह्यूमनॉइड रोबोट में 16 DOF होते हैं, जिसमें प्रत्येक डिग्री एक जॉइंट से संबंधित होती है। उदाहरण के लिए, जॉइंट्स हिप, घुटना, टखना, कोहनी, कलाई आदि में हो सकते हैं।
- ह्यूमनॉइड रोबोट का गेट साइकल स्टैंस फेज़ और स्विंग फेज़ जैसे चरणों में विभाजित होता है, जहाँ प्रत्येक जॉइंट एंगल बदलता है।
- गेट एंगल्स उन संयुक्तों के बीच के कोण होते हैं (जैसे, जांघ और शिन, ऊपरी भुजा और अग्रभुजा, आदि) जो गेट साइकल के विभिन्न बिंदुओं पर गणना किए जाते हैं।

2. डायनेमिक बैलेंस मार्जिन:

- डायनेमिक बैलेंस रोबोट की स्थिरता बनाए रखने की क्षमता को संदर्भित करता है, जबकि यह गति में है, बलों, त्वरणों और संवेगों का ख्याल रखते हुए।
- डायनेमिक बैलेंस मार्जिन रोबोट की गिरावट को रोकने की क्षमता है, जो केंद्रक (COM) और सपोर्ट पॉलीगन (पैरों के संपर्क बिंदु) की स्थिति से निर्धारित होती है।

- COM और सपोर्ट पॉलीगन के किनारे के बीच की दूरी का उपयोग बैलेंस मार्जिन को मापने के लिए किया जाता है। एक बड़ा मार्जिन बेहतर स्थिरता को दर्शाता है।
-

कार्यप्रणाली:

चरण 1: काइनेमेटिक मॉडल सेटअप

1. रोबोट का विन्यास:

- ह्यूमेनॉइड रोबोट के जॉइंट विन्यास (स्थितियाँ, अभिविन्यास) को परिभाषित करें।
- रोबोट के जॉइंट्स के लिए डिग्री ऑफ फ्रीडम (16 DOF) की संख्या दर्ज करें, जिसमें पैर, हाथ और धड़ शामिल हैं।
- रोबोट को इसके लिंक लंबाई और जॉइंट प्रकारों (रिवोल्यूट या प्रिज़मैटिक) के संदर्भ में प्रस्तुत करें।

2. जॉइंट एंगल्स की परिभाषा:

- प्रत्येक डिग्री ऑफ फ्रीडम के लिए जॉइंट एंगल्स परिभाषित करें (जैसे पैर के लिए हिप, घुटना, टखना, हाथ के लिए कंधा, कोहनी)।
- जॉइंट एंगल्स को एक न्यूट्रल स्थिति (खड़ा मुद्रा) में प्रारंभिक रूप से परिभाषित करें।

चरण 2: वॉकिंग सायकल का सिमुलेशन

1. गेट फेज़ का विभाजन:

- गेट साइकल को मुख्य चरणों में विभाजित करें: प्रारंभिक संपर्क, मध्य-स्थिति, अंतिम स्थिति, और स्विंग फेज़।
- प्रत्येक चरण के लिए अवधि को रोबोट की वॉकिंग गति और कदेंस के आधार पर परिभाषित करें।

2. जॉइंट एंगल्स का परिवर्तन:

- प्रत्येक गेट फेज़ के लिए, यह निर्धारित करें कि प्रत्येक जॉइंट एंगल कैसे बदलता है। उदाहरण के लिए:
 - हिप फ्लेक्सन/एक्सटेंशन
 - घुटना फ्लेक्सन/एक्सटेंशन
 - टखना डॉर्सीफ्लेक्सन/प्लांटारफ्लेक्सन
- काइनेमेटिक समीकरणों या फॉरवर्ड काइनेमेटिक्स का उपयोग करके प्रत्येक जॉइंट की स्थिति और कोण की गणना करें।

चरण 3: गेट एंगल्स की गणना

1. फॉरवर्ड काइनेमेटिक्स:

- फॉरवर्ड काइनेमेटिक्स का उपयोग करके गेट सायकल के विभिन्न चरणों में जॉइंट एंगल्स की गणना करें। ह्यूमेनॉइड रोबोट के लिए, इसमें प्रत्येक लिंक (जांघ, शिन, पैर आदि) की स्थिति और अभिविन्यास की गणना करना शामिल होता है।
- गेट के प्रत्येक चरण के लिए प्रमुख जोड़ों (हिप, घुटना, टखना) के एंगल्स की गणना करें।

2. एंगल डेटा का निष्कर्षण:

- यदि आवश्यक हो तो, इनवर्स काइनेमेटिक्स का उपयोग करके किसी भी समय पर प्रत्येक अंग के जॉइंट एंगल्स को निष्कर्षित करें।
- जैसे-जैसे रोबोट चलता है, इन एंगल्स को समय के साथ दिखाने के लिए इनकी दृश्यता करें।

चरण 4: डायनेमिक बैलेंस मार्जिन की गणना

1. केंद्रक (COM) की गणना:

- प्रत्येक गेट फेज़ के लिए, रोबोट के COM की स्थिति की गणना करें। COM को ह्यूमेनॉइड रोबोट के सभी लिंक के स्थानों का भारित औसत के रूप में गणना किया जा सकता है।
- COM के लिए सूत्र लागू करें:

$$COM = \frac{\sum_i m_i \cdot x_i}{\sum_i m_i}$$

जहाँ m_i , i -th लिंक का द्रव्यमान है और x_i उसका स्थान है।

2. सपोर्ट पॉलीगन:

- प्रत्येक समय पर, रोबोट के पैर के संपर्क बिंदुओं की पहचान करें।
- इन संपर्क बिंदुओं का कॉन्वेक्स हल परिभाषित करें, जिसे सपोर्ट पॉलीगन कहते हैं।

3. बैलेंस मार्जिन की गणना:

- COM से सपोर्ट पॉलीगन के किनारे तक की दूरी की गणना करें। एक सामान्य विधि COM से सपोर्ट पॉलीगन के केंद्र तक की दूरी की गणना करना है और यह निर्धारित करना है कि COM पॉलीगन के अंदर है या नहीं।
- यदि COM पॉलीगन के बाहर चला जाता है, तो रोबोट गिरने की स्थिति में है।

4. डायनेमिक बैलेंस का मूल्यांकन:

- गेट साइकल के दौरान COM को सपोर्ट पॉलीगन के भीतर बनाए रखने की जांच करके रोबोट के डायनेमिक बैलेंस का मूल्यांकन करें।
- यदि COM स्विंग फेज़ के दौरान पॉलीगन के किनारे को पार करता है, तो यह बैलेंस खोने का संकेत हो सकता है।

चरण 5: डेटा विश्लेषण

1. गेट एंगल्स का प्लॉटिंग:

- गेट साइकल के प्रत्येक चरण में प्रमुख जोड़ों (हिप, घुटना, टखना) के लिए गणना किए गए जॉइंट एंगल्स को समय के साथ प्लॉट करें।
- जोड़ों की गति के पैटर्न, सीमा और समरूपता का विश्लेषण करें।

2. बैलेंस मार्जिन का मूल्यांकन:

- गेट साइकल के प्रत्येक बिंदु पर डायनेमिक बैलेंस मार्जिन को समय या फेज़ के अनुसार प्लॉट करें।
- बैलेंस मार्जिन की निगरानी करें और जांचें कि COM सपोर्ट पॉलीगन के भीतर रहता है या नहीं।

परिणाम:

- गेट एंगल्स:

- गेट सायकल के विभिन्न चरणों के दौरान गणना किए गए जॉइंट एंगल्स (हिप, घुटना, टखना) को प्रस्तुत करें।

- **डायनेमिक बैलेंस मार्जिन:**

- गेट सायकल के प्रत्येक बिंदु पर डायनेमिक बैलेंस मार्जिन दिखाने वाला ग्राफ प्रदान करें।
- ऐसे क्षणों का संकेत दें जहाँ रोबोट संतुलन खोने के करीब है।

चर्चा:

- गेट एंगल्स की गणना के परिणामों पर चर्चा करें, जिसमें प्रत्येक जॉइंट की गति सीमा।
- डायनेमिक बैलेंस मार्जिन का विश्लेषण करें और यह मूल्यांकन करें कि रोबोट गेट सायकल के दौरान संतुलन बनाए रखने में सक्षम है या नहीं।
- विश्लेषण के आधार पर ह्यूमेनॉइड रोबोट के बैलेंस और गेट में सुधार के लिए संभावित उपायों का सुझाव दें।

निष्कर्ष:

यह प्रयोग ह्यूमेनॉइड रोबोट के गेट एंगल्स और डायनेमिक बैलेंस मार्जिन की गणना करने में सक्षम बनाता है, जिससे रोबोट की चलने की दक्षता और स्थिरता के बारे में जानकारी मिलती है। जॉइंट एंगल्स और बैलेंस मार्जिन का विश्लेषण करके आप रोबोट के प्रदर्शन का आकलन कर सकते हैं और सुधार के क्षेत्रों की पहचान कर सकते हैं।

संदर्भ:

1. स्पॉन्ग, M. W., & विद्यासागर, M. (2008). *रोबोट डायनेमिक्स एंड कंट्रोल*. विली।
2. खतीब, O. (1987). *रोबोट मैनिपुलेटर के गति और बल नियंत्रण के लिए एक एकीकृत दृष्टिकोण: ऑपरेशनल स्पेस फॉर्मूलेशन*. IEEE जर्नल ऑफ रोबोटिक्स एंड ऑटोमेशन।
3. राइबर्ट, M. H., & ब्लैकस्पूर, K. (2008). *BigDog, the rough-terrain robot: An overview*. Journal of Field Robotics.

Experiment No-10

Statement: Do it yourself (DIY) experiments (Students should take the real-world issue and they have to think, decide and do things independently)

प्रयोग 10:

कथन: इसे स्वयं करें (DIY) प्रयोग (छात्रों को वास्तविक दुनिया के मुद्दे को लेना चाहिए और उन्हें स्वतंत्र रूप से सोचना, निर्णय लेना और कार्य करना होगा)

2. उत्पाद विकास प्रयोगशाला

कार्यक्रम का नाम	M Tech in Industrial Design	Semester: II	Year: I
पाठ्यक्रम का नाम	उत्पाद विकास प्रयोगशाला		
Course Code	ID 525		
Core / Elective / Other	मूल /ऐच्छिक/अन्य		
आवश्यक शर्तें:			
1.	वैचारिक डिजाइन चरण, समस्या समाधान पद्धति की समझ।		
2.	सीएडी प्लेटफॉर्म पर काम करने का ज्ञान।		
Course Outcomes:			
CO 1	मौजूदा उत्पादों में नवीनता की अवधारणा और कल्पना करने में सक्षम बनाना ।		
CO 2	संपूर्ण डिजाइन विशिष्टताओं के साथ नए भाग/उत्पाद या वेरिएंट को विकसित करने में सक्षम करें।		
CO 3	सीएडी अनुप्रयोगों का उपयोग करके आशाजनक उत्पाद की जांच और विकास के लिए औद्योगिक डिजाइन ज्ञान और कौशल लागू करें।		
Description of Contents in brief:			
Unit 1.	उत्पाद सौंदर्यशास्त्र का अध्ययन और वैकल्पिक, डिजाइन सोच का विकास - एक चयनित उत्पाद का मामला।		
Unit 2.	उपलब्ध वेरिएंट ढूँढ़ें, भागों, घटकों और उप-असेंबली को चित्रों के साथ सूचीबद्ध करें।		
Unit 3.	फ़ंक्शन को समझाएं और प्रत्येक भाग, घटक और उप-असेंबली के लिए सामग्री की पहचान करें।		
Unit 4.	विभिन्न कंप्यूटर सहायता प्राप्त डिजाइन सॉफ्टवेयर पैकेजों के अनुप्रयोग का उपयोग करके उत्पाद का विश्लेषण और डिजाइन करें।		
Unit 5.	उत्पाद एर्गोनॉमिक्स, रिवर्स इंजीनियरिंग के बारे में अध्ययन करें और वैचारिक रूप से व्यवहार्य समाधान प्रदान करें।		
List of Text Books:			
1.	Engineering Design, George E. Dieter and Linda C. Schmidt, Mc Graw hill.		
2.	Product Design and Development, Karl T. Ulrich and Steven D. Eppinger, Mc Graw Hill.		
3.	Product Design for Engineers, Devdas Shetty, Cengage Learning.		
List of Reference Books:			
1.	Engineering Design Methods, Nigel Cross, Wiley.		
2.	Materials Selection in Mechanical Design, Michael F. Ashby		
3.	Product Design, Techniques in Reverse Engineering and New Product Development, Kevin Otto, Kristin wood, Pearson Education		
URLs:			
1.	https://www.digimat.in/nptel/courses/video/112104230/L01.html		

2.	https://nptel.ac.in/courses/110/105/110105087/	
Lecture Plan:		
प्रयोग- संख्या	विषय	
प्रयोग - 1:	परिचय - डिजाइन और विकास के लिए भाग/उत्पाद की पहचान पर विचार-मंथन (व्यवहार्यता, सामाजिक, पर्यावरणीय और आर्थिक विचार)।	
प्रयोग - 2:	CAD अनुप्रयोगों का उपयोग करके उत्पाद का वर्णन करें।	
प्रयोग - 3:	उपलब्ध वेरिएंट ढूँढ़ें. उनके पक्ष-विपक्ष पर चर्चा।	
प्रयोग - 4:	भागों, घटकों और उप-असेंबली को चित्रों के साथ सूचीबद्ध करें।	
प्रयोग - 5:	भागों, घटकों और उप-असेंबली के लिए सामग्री की पहचान करें।	
प्रयोग - 6:	प्रत्येक भाग, घटक और उप-असेंबली के कार्य को समझाएं।	
प्रयोग - 7:	उत्पाद को अलग करने और दोबारा जोड़ने के चरण (बाधित पहुंच का पता लगाना, आवश्यक समय आदि) - रिवर्स इंजीनियरिंग।	
प्रयोग - 8:	एर्गोनॉमिक्स पर चर्चा करें (आवश्यक ऑपरेटरों की संख्या, आराम, कंपन, होल्डिंग आवश्यकताएं आदि सहित)।	
प्रयोग - 9:	इस उत्पाद से जुड़ी सामान्य समस्याओं की पहचान करें।	
प्रयोग - 10:	उन समस्याओं के समाधान सुझाएं और मौजूदा उत्पाद के लिए वैचारिक रूप से प्रस्तावित परिवर्तन/संशोधन प्रदान करें।	

Evaluation Criteria:

Sl. No.	Name of Examination	Marks Allotted	Remarks
1	Mini Test		
2	Mid Semester Test		
3	Assignment if any		
4	Tutorial if any		
5	Quiz if any		
6	Seminar, Viva voce if any		
7	End Semester Examination		
8	Experiments if any (for practical courses)		
9	Any other		

प्रयोग संख्या 1

उद्देश्य - विचार-मंथन (व्यवहार्यता, सामाजिक, आर्थिक, नियामक और पर्यावरणीय विश्लेषण) के माध्यम से रचना और विकास के लिए उत्पाद की पहचान करना।

अपने रुचि के भाग/घटक/उत्पाद की पहचान के लिए निम्नलिखित गतिविधियाँ करें:

चरण 01 - अनुरोधित/ नहीं-अनुरोधित की गई आवश्यकता की पहचान करें आवश्यक जानकारी की पहचान करना और उसे ढूँढना या विकसित करना। यह आवश्यकता अक्सर अधिक विशिष्ट और वर्तमान जानकारी की माँग करती है।

महत्वपूर्ण जानकारी के स्रोत:

- आर एंड डी विभागों और कंपनियों द्वारा प्रकाशित तकनीकी विवरण।
- व्यापार पत्रिकाएँ, एकस्व, सर्वाधिकार, सूचीपत्र।
- सामग्री और उपकरणों के विक्रेताओं और आपूर्तिकर्ताओं द्वारा प्रकाशित पुस्तिकाएँ।
- तकनीकी कोड और मानक, सरकारी नियम।
- अंतराजाल खोज के माध्यम से जानकारी।
- आन्तरिक विशेषज्ञों के साथ चर्चा।
- बाहरी सलाहकार।
- ग्राहकों (एकल या समूह) के साक्षात्कार।
- उत्पाद का उपयोग करते समय निरीक्षण।

महत्वपूर्ण प्रश्न:

- मुझे क्या जानने की आवश्यकता है?
- यह जानकारी कहाँ मिलेगी और कैसे प्राप्त की जाएगी?
- जानकारी कितनी विश्वसनीय और सटीक है?
- जानकारी को मेरी विशिष्ट आवश्यकता के लिए कैसे व्याख्या किया जाए?
- कब पर्याप्त जानकारी हो जाती है?
- इस जानकारी से क्या निर्णय निकलते हैं?
- इस उत्पाद का उपयोग कब और क्यों किया जाता है?
- मौजूदा उत्पाद के बारे में आपको क्या पसंद या नापसंद है?
- इस उत्पाद को खरीदते समय आप किन मुद्दों पर विचार करते हैं?

चरण 02 - विचार निर्माण (रचनात्मकता भागफल) यह चरण रचनात्मकता से भरा है जिसमें उत्पाद विचार, अवधारणा विकास और उत्पाद छवि को मान लिया जाता है। इसके बाद, विचार जांच की जाती है जिसमें कारण और प्रभाव विश्लेषण शामिल होता है।

चरण 03 - रचना का विश्लेषण:

- व्यवहार्यता विश्लेषण।
- समाज पर प्रभाव।
- अर्थशास्त्र।
- पर्यावरणीय विश्लेषण।
- नियामक विश्लेषण।

प्रयोग संख्या 2

उद्देश्य - कैड / चित्रात्मकता का उपयोग करके निर्मित भाग/उत्पाद के कार्य सिद्धांत और कार्य को समझाएं।

चरण:

1. **विवरण और दस्तावेजीकरण** - क्या यह एकल वस्तु या एकत्रा है?
2. **आकार (हस्तलिखित रेखा-चित्र)** - बिना प्रारूपण उपकरण के चित्रकला ।
3. **आकार (तकनीकी चित्रकला)** - उचित माप सुनिश्चित करने के लिए उपकरणों का उपयोग।
 - मल्टीव्यू प्रक्षेपण।
 - एक्सोनोमेट्रिक प्रक्षेपण।
 - परिप्रेक्ष्य प्रक्षेपण।
4. **सहिष्णुता** - अनुमेय भिन्नता को परिभाषित करें।
5. **दृश्य सौंदर्यशास्त्र** - आकार, रूप, बनावट और रंग दस्तावेज़ करें।

प्रयोग संख्या 3

उद्देश्य - उपलब्ध संस्करणों को ढूँढ़ें और उनके लाभ और हानि पर चर्चा करें।

चरण 01:

- बाजार में उपलब्ध उत्पादों की पहचान करें।
- विशेषताओं और तकनीकों का अध्ययन करें।
- लाभ और हानि दस्तावेज़ करें।
- उत्पादों को श्रेणियों में विभाजित करें।
- दोषों की पहचान करें और रचना को बेहतर बनाएं।

चरण 02 - रचना का चयन:

- मूल रचना ।
- अनुकूलित रचना ।
- पुनः रचना ।
- संस्करण रचना ।
- चयन रचना ।
- औद्योगिक रचना ।

चरण 03 - संकेतावली और मानकों के साथ रचना:

- दक्षता और सुरक्षा को बढ़ावा दें।
- विनिमेयता और संगतता सुनिश्चित करें।

प्रयोग संख्या 4

उद्देश्य - चित्रों के साथ भागों, घटकों और उप-संयोजन को सूचीबद्ध करें।

क्र.सं.	विवरण	चित्र	टिप्पणी
1			
2			
3			
4			

(आवश्यकतानुसार पंक्तियाँ जोड़ें। चित्रों को लेबल करें।)

प्रयोग संख्या 5

उद्देश्य - भागों, घटकों और उप-संयोजन के लिए सामग्री चयन।

सर्वोत्तम सामग्री का चयन करने में केवल आवश्यक गुणों के साथ सामग्री चुनना ही शामिल नहीं है, बल्कि इसमें उन विधियों का उपयोग भी शामिल है जिनसे अंतिम भाग बनाया जाएगा। सामग्री चयन एक निर्णय लेने की प्रक्रिया है जिसमें निम्नलिखित शामिल हैं:

- सामग्री आवश्यकताओं का विश्लेषण। सेवा और पर्यावरण की स्थितियों का निर्धारण।
- इन आवश्यकताओं को सामग्री गुणों में बदलना।
- उम्मीदवार सामग्रियों की जाँच करें।
- प्रदर्शन, लागत, उत्पादन और उपलब्धता के आधार पर सामग्री का चयन करें।

क्र.सं. भाग/ संयोजन विवरण चयनित सामग्री टिप्पणी

1

2

(आवश्यकतानुसार पंक्तियाँ जोड़ें।)

प्रयोग संख्या 6

उद्देश्य - प्रत्येक भाग, घटक और उप- संयोजन के कार्य को समझाएं।

क्र.सं.	भाग/ संयोजन विवरण	भूमिका	टिप्पणी
1			
2			

(आवश्यकतानुसार पंक्तियाँ जोड़ें।)

प्रयोग संख्या 7

उद्देश्य - उत्पाद के विघटन और पुनः संयोजन के चरण (रुकावट, आवश्यक समय आदि) - उत्क्रम अभियांत्रिकी ।

विघटन के लिए रचना (DFD) प्रक्रिया को सरल बना सकता है और इस प्रकार संसाधन, ऊर्जा, समय और लागत को बचा सकता है। उत्पाद विच्छेदन या रिवर्स इंजीनियरिंग में शामिल हैं:

- उत्पाद का उपयोग करते समय निरीक्षण।
- वस्तु को अलग करके देखना कि यह कैसे काम करता है, उत्पाद विच्छेदन उत्क्रम अभियांत्रिकी कहलाता है।
- किसी उपकरण, वस्तु या प्रणाली के तकनीकी सिद्धांतों की खोज इसके संरचना, कार्य और संचालन के विश्लेषण के माध्यम से।

संयोजन के लिए रचना (DFA) के दिशा-निर्देश:

- कुल भागों की संख्या कम करें।
- संयोजन सतहों को न्यूनतम करें।
- उप- संयोजन का उपयोग करें।
- रचना और संयोजन में त्रुटि-सबूत बनाएं।
- अलग फास्टरों से बचें।
- संयोजन में हैंडलिंग को न्यूनतम करें।
- संयोजनदिशा को न्यूनतम करें।
- भागों और उपकरणों के लिए अवरोध-मुक्त पहुंच प्रदान करें।
- संयोजन में अनुपालन को अधिकतम करें।
- मानकीकरण का अधिकतम उपयोग करें।

क्र.सं. भाग/ संयोजन विवरण डीएफडी या डीएफए के लिए अनुक्रम/चरण टिप्पणी

1

2

(आवश्यकतानुसार पंक्तियाँ जोड़ें।)

प्रयोग संख्या 8

उद्देश्य - श्रमदक्षता शास्त्र पर चर्चा करें (जैसे ऑपरेटर्स की संख्या, आराम, कंपन, पकड़ने की आवश्यकताएं आदि)।

सुविधापूर्ण रचना, जिसे मानव कारक रचना भी कहा जाता है, लोगों, उनके द्वारा उपयोग किए जाने वाले उत्पादों और प्रणालियों, और जिस पर्यावरण में वे काम करते हैं, का अध्ययन है। यह मुख्य रूप से सुरक्षा, आराम, सौंदर्यशास्त्र, संवेदनाओं और उपयोग में आसानी जैसे पहलुओं को सम्मिलित करता है।

इस रचना के साथ जुड़े ऐसे श्रमदक्षता शास्त्र के गुणों और विशेषताओं की सूची बनाएं।

क्र.सं.	विशेषता/भाग/ संयोजन विवरण	एर्गोनॉमिक रचना विशेषता	टिप्पणी
1			
2			

(आवश्यकतानुसार पंक्तियाँ जोड़ें।)

प्रयोग संख्या 9

उद्देश्य - इस उत्पाद से संबंधित सामान्य समस्याओं की पहचान करें।

क्र.सं. समस्याओं का विवरण

(आवश्यकतानुसार पंक्तियाँ जोड़ें।)

प्रयोग संख्या 10

उद्देश्य - इन समस्याओं के समाधान सुझाएँ और मौजूदा उत्पाद के लिए प्रस्तावित परिवर्तन/संशोधन प्रदान करें।

विवरण और विस्तृत चित्र प्रदान करें।

3. नैनो-स्नेहक प्रयोगशाला

प्रयोगशाला मैनुअल



नैनो स्नेहक अनुसंधान प्रयोगशाला

यांत्रिकी अभियांत्रिकी विभाग
मौलाना आज़ाद राष्ट्रीय प्रौद्योगिकी संस्थान, भोपाल
म.प्र. (भारत) 462003

नैनो स्नेहक अनुसंधान प्रयोगशाला

प्रयोगों की सूची

क्र.सं	प्रयोग
1.	अंकीय श्यानतामापी की सहायता से दिए गए द्रव की श्यानता का निर्धारण।
2.	पेन्स्की मार्टेंस उपकरण का उपयोग करके चिकनाई वाले तेल के फ़्लैश बिंदु का मापन।
3.	अंकीय तप्त वायु भट्ठी के कार्य सिद्धांत, संचालन और प्रभावी निष्फलन और गर्म करने के उद्देश्यों के लिए उपयोग का अध्ययन।
4.	नैनोकण आकृति विज्ञान और स्नेहन घटना में उनकी भूमिका का अध्ययन।
5.	चार बॉल टेस्टर का उपयोग करके स्नेहन तेल की घर्षण-रोधी और घर्षण संबंधी विशेषताओं का प्रायोगिक अध्ययन।
6.	वायु जेट क्षरण परीक्षक का उपयोग करके विभिन्न सामग्रियों पर ठोस कण क्षरण घटना का प्रायोगिक अध्ययन।

प्रयोग संख्या: 1

ध्येय: डिजिटल विस्कोमीटर (**अंकीय श्यानतामापी**) के कार्य सिद्धांत, संचालन और उपयोग का अध्ययन करना तथा विस्कोसिटी (श्यानता) की अवधारणा को एक प्रमुख द्रव गुण के रूप में समझना।

उद्देश्य: दिए गए तरल की श्यानता को अंकीय श्यानतामापी की सहायता से मापना ।

आवश्यक उपकरण: अंकीय श्यानतामापी, सेंसर तत्व (धुरा), फ्लास्क, स्थिर तापमान स्नान, प्रिंटर आदि।

परिचय: श्यानता द्रवों का एक मौलिक गुण है जो द्रव के प्रवाह के प्रति आंतरिक प्रतिरोध को दर्शाती है। यह द्रव की सन्निकट परतों के बीच घर्षण का प्रतिनिधित्व करती है जब वे एक दूसरे के सापेक्ष गति करते हैं। SI प्रणाली में श्यानता की इकाई पास्कल-सेकंड ($\text{Pa}\cdot\text{s}$) है, लेकिन प्रयोगशालाओं में सबसे अधिक उपयोग की जाने वाली इकाई सेंटीपोइज़ (cP) है, जहाँ $1 \text{ cP} = 0.001 \text{ Pa}\cdot\text{s}$ । उच्च श्यानता वाले द्रव (जैसे तेल, शहद) धीरे-धीरे प्रवाहित होते हैं, जबकि निम्न श्यानता वाले द्रव (जैसे पानी, अल्कोहल) आसानी से प्रवाहित होते हैं।

अंकीय श्यानतामापी एक इलेक्ट्रॉनिक उपकरण है जो उच्च सटीकता के साथ द्रवों की श्यानता को मापने के लिए उपयोग किया जाता है। यह द्रव में डूबे हुए धुरा को घुमाकर काम करता है और धुरा के द्वारा सामना किए गए प्रतिरोध को मापता है। यह प्रतिरोध द्रव की श्यानता के अनुपात में होता है। अंकीय श्यानतामापी वास्तविक समय में डिजिटल स्क्रीन पर श्यानता को प्रदर्शित करता है, सामान्यतः सेंटीपोइज़ (cP) या मिलीपास्कल-सेकंड ($\text{mPa}\cdot\text{s}$) जैसी इकाइयों में। अंकीय श्यानतामापियों में स्वचालित गति नियंत्रण, तापमान समायोजन, और डेटा संग्रहण जैसी सुविधाएँ होती हैं, जो सटीकता और उपयोग में आसानी को बढ़ाती हैं। इन्हें औषधि, खाद्य उत्पादन, कॉस्मेटिक्स, और सामग्री अनुसंधान जैसी उद्योगों में गुणवत्ता नियंत्रण और प्रक्रिया अनुकूलन के लिए व्यापक रूप से उपयोग किया जाता है।

कार्य सिद्धांत: अंकीय श्यानतामापी प्रसिद्ध घूर्ण ऐंठन (rotational torque) के सिद्धांत का उपयोग करता है। यह श्यानता को मापता है द्वारा उस टॉर्क का पता लगाकर जो एक धुरा को स्थिर गति से घुमाने के लिए आवश्यक होता है, जब वह नमूने के द्रव में डूबा होता है। यह टॉर्क डूबे हुए धुरा पर विस्कस खींचने के लिए समानुपातिक होता है और इस प्रकार द्रव की श्यानता से संबंधित होता है। श्यानता को कई रेंज में मापना संभव है क्योंकि, किसी दिए गए स्प्रिंग विक्षेप के लिए, वास्तविक श्यानता विशिष्ट गति के समानुपात में होती है और यह धुरा के आकार और आकार से संबंधित होती है। दिए गए श्यानता वाले एक सामग्री के लिए, प्रतिरोध तब अधिक होगा जब धुरा का आकार और/या घूर्णन गति बढ़ती है। न्यूनतम

श्यानता रेंज को सबसे बड़े धुरा का उपयोग करके और उच्चतम गति पर प्राप्त किया जाता है; अधिकतम रेंज को सबसे छोटे धुरा का उपयोग करके और न्यूनतम गति पर प्राप्त किया जाता है। एक ही धुरा का उपयोग करके विभिन्न गति पर किए गए माप परीक्षण सामग्री की रिओलॉजिकल विशेषताओं का पता लगाने और उनका मूल्यांकन करने के लिए उपयोग किए जाते हैं। धुरा का निरंतर घूर्णन लंबे समय तक बिना रुकावट के माप लेने की अनुमति देता है, जिससे समय पर निर्भर द्रव गुणों का विश्लेषण संभव होता है। धुरा के द्रव के प्रति जोश की दर निरंतर होती है, इसलिए उपकरण न्यूटोनियन और गैर-न्यूटोनियन द्रवों को मापने के लिए उपयुक्त है। डूबे हुए धुरा को कई अलग-अलग गति पर घुमाकर, गैर-न्यूटोनियन द्रवों के शीयर निर्भर व्यवहार का पता लगाया और विश्लेषण किया जा सकता है।



प्रक्रिया:

1. अंकीय श्यानतामापी को स्टैंड पर स्थापित करें और स्पिरिट लेवल की सहायता से इसे क्षैतिज बनाएं।
2. परीक्षण ल्यूब्रिकेंट में केंद्रीय धुरा डालें जब तक कि स्नेहक का स्तर क्रीज तक न पहुँच जाए।
3. श्यानता मापन के लिए अंकीय श्यानतामापी का मोटर स्विच "चालू" करें, जो श्यानतामापी ड्राइव मोटर को सक्रिय करता है। प्रदर्शित रीडिंग के स्थिर होने के लिए समय दें। स्थिरीकरण के लिए आवश्यक समय अंकीय श्यानतामापी की गति और नमूने के द्रव की विशेषताओं पर निर्भर करेगा।
4. जब आप रिकॉर्डर के साथ अंकीय श्यानतामापी का उपयोग कर रहे हों, तो रिकॉर्डर को "रन" मोड में सेट करें और श्यानतामापी की रीडिंग लें/रिकॉर्ड करें। ध्यान दें कि स्ट्रिप चार्ट रिकॉर्डर में उपयोग किया जाने वाला पेपर 0-100 स्केल पर होता है। चार्ट पर रीडिंग का उपयोग अंकीय श्यानतामापी के डिस्प्ले रीडिंग की तरह ही किया जाता है।
5. जब आप श्यानता मापन कर रहे हों, तो रीडिंग को नोट करें और इसे उस अंकीय श्यानतामापी मॉडल/धुरा/गति संयोजन के लिए उपयुक्त गुणांक से गुणा करें जो उपयोग किया जा रहा है। गुणांक को बुकफील्ड गुणांक खोजक से प्राप्त किया जा सकता है। अधिकतम सटीकता के लिए 10.0 से नीचे के डिस्प्ले रीडिंग से बचना चाहिए।
6. धुरा बदलते समय, नमूने बदलते समय आदि, अंकीय श्यानतामापी का मोटर स्विच "बंद" करें। सफाई से पहले धुरा को हटा दें।

आर.पी.एम	एस.ए.ई-40 स्नेहक तेल			आर.वी.- -5	एस.ए.ई-90 स्नेहक तेल			आर.वी.- -5
	आर.वी.- 2	आर.वी.- -3	आर.वी.- -4		आर.वी.- -2	आर.वी.- -3	आर.वी.- -4	
	श्यानता = डायल रीडिंग * 10 म-पा-से (सेंटिपॉइज़)				श्यानता = डायल रीडिंग * 10 म-पा-से (सेंटिपॉइज़)			

सावधानियाँ:

1. प्रयोगशाला स्टैंड क्लैम्प असेंबली की स्थिति महत्वपूर्ण है।
2. अंकीय श्यानतामापी के आउटपुट केबल को विद्युत स्रोत से न जोड़ें।
3. श्यानता मापन धुरा के आकार और आकृति, साधारण पात्र के आकार, धुरा की गति, नमूने के तापमान आदि पर निर्भर करता है, इसलिए उपयुक्त संयोजन का चयन करना आवश्यक है।
4. किसी भी स्थिति में धुरा को असामान्य रूप से नहीं घुमाना चाहिए।
5. अंकीय श्यानतामापी की रीडिंग कभी भी 100 से अधिक नहीं होनी चाहिए।

प्रयोग संख्या: 2

उद्देश्य: पेंसकी मार्टेंस यंत्र का उपयोग करके स्नेहन तेल का फ्लैश बिंदु मापना।

आवश्यक उपकरण: पेंसकी मार्टेंस बंद प्रकार का फ्लैश बिंदु यंत्र

सिद्धांत:

फ्लैश बिंदु किसी स्नेहक या द्रव की दहनशीलता के तापमान का प्रतिनिधित्व करता है। इसे उस न्यूनतम तापमान के रूप में परिभाषित किया जाता है जिस पर स्नेहक वाष्पित होकर ऑक्सीजन के साथ एक ज्वलनशील मिश्रण बनाता है। जब इस वाष्प पर एक छोटी पायलट ज्वाला डाली जाती है, तो क्षणिक रूप से एक चमक उत्पन्न होती है और प्रज्वलन स्रोत की अनुपस्थिति में यह जलना बंद कर देती है। फ्लैश बिंदु उस तापमान को दर्शाता है जहाँ पदार्थ में आग नहीं लगती क्योंकि वाष्प की मात्रा पर्याप्त नहीं होती।

परिधि: IP-34-ASTM-D-93- 58T/ 1ST 1448/1209(P:21) में ईंधन तेल, स्नेहक, और चिपचिपे पदार्थों के फ्लैश बिंदु के निर्धारण की प्रक्रिया सम्मिलित है।

विधि का सारांश:

नमूने को एक परीक्षण कप में धीमी और सतत दर पर गर्म किया जाता है और इसे लगातार हिलाया जाता है। नियमित अंतराल पर परीक्षण ज्वाला को कप में डाला जाता है और हिलाने की प्रक्रिया को रोक दिया जाता है। इस प्रक्रिया में फ्लैश बिंदु उस सबसे कम तापमान को माना जाता है जिस पर परीक्षण ज्वाला डालने पर नमूने के ऊपर का वाष्प क्षणिक रूप से प्रज्वलित हो जाता है।

नमूना तैयारी: चिपचिपे पदार्थ के नमूने को तब तक गर्म किया जा सकता है जब तक वे तरल रूप में आ जाएं। जिन नमूनों में घुला हुआ या मुक्त जल होता है, उन्हें कैल्शियम क्लोराइड या सूखी शोषक रुई के ढीले प्लग से सुखाया जा सकता है।

प्रयोग/अध्ययन – 3

उद्देश्य: अंकीय तप्त वायु भट्टी का अध्ययन करना और इसके कार्य सिद्धांत, संचालन, और प्रयोगशाला में प्रभावी निष्फलन और गर्म करने के उद्देश्यों के लिए उपयोग को समझना।

उपकरण: अंकीय तप्त वायु भट्टी, कांच के बर्तन, चिमटे या फोर्सेप्स एल्यूमीनियम पन्नी, ताप-प्रतिरोधी दस्ताने, निष्फलन संकेतक, टाइमर, सुरक्षात्मक गियर।

परिचय:

यह उष्मा उपचार की विधि को सबसे पहले फ्रांसीसी वैज्ञानिक लुई पाश्चर द्वारा 19वीं सदी के अंत में पेश किया गया था। प्रारंभिक प्रकार के भट्टी में, शुष्क गर्मी का उपयोग एक संक्षिप्त समय के लिए हानिकारक सूक्ष्मजीवों को मदिरा से मारने के लिए किया गया था, बिना इसके स्वाद को बदले। वर्षों के दौरान, इलेक्ट्रिकल तकनीक में प्रगति के कारण अंकीय तप्त वायु भट्टी का निर्माण हुआ, जो सटीक तापमान नियंत्रण, नियंत्रणीय (प्रोग्रामेबल) सुविधाएँ और समान गर्मी वितरण प्रदान करते हैं, जिससे ये वैज्ञानिक और औद्योगिक सेटिंग्स में अनिवार्य हो जाते हैं।

एक तप्त वायु भट्टी एक आवश्यक प्रयोगशाला उपकरण है जो प्रयोगशाला की वस्तुओं और नमूनों को निष्फल करने के लिए शुष्क उष्मा (गर्म हवा) का उपयोग करता है। इस प्रकार के निष्फलन को शुष्क उष्मा निष्फलन भी कहा जाता है। एक तप्त वायु भट्टी आमतौर पर उन नमूनों के लिए उपयोग किया जाता है जो गर्मी प्रतिरोधी होते हैं और जो उच्च तापमान के संपर्क में आने पर पिघलते नहीं, आकार नहीं बदलते या आग नहीं लगाते। यह सामान्यतः सूक्ष्मजीवों और जीवाणु बीजाणु को अत्यधिक उच्च तापमान पर कई घंटों के दौरान मारता है और वस्तुओं को निष्फल करता है। प्रभावी निष्फलन केवल तभी प्राप्त किया जा सकता है जब उचित तापमान और धारण समय का चयन किया जाए, जो लक्षित सूक्ष्मजीव के प्रकार और निष्फल किए जा रहे सामग्री के प्रकार पर निर्भर करता है। वस्तुओं को निष्फल करने के लिए सबसे सामान्य तापमान और समय 170°C पर 30 मिनट, 160°C पर 60 मिनट और 150°C पर 150 मिनट है।

सिद्धांत और निर्माण:

तप्त वायु भट्टी शुष्क हवा निष्फलन प्रक्रिया के सिद्धांत पर काम करता है, जो संवहन, और विकिरण के माध्यम से होता है। गर्म करने वाला तत्व कक्ष के भीतर हवा को गर्म करता है और उसे पंखों की सहायता से समान रूप से वितरित करता है, ताकि नमूनों की सतहें गर्म और शुष्क हवा के संपर्क में आएँ। यह

संपर्क वस्तुओं की बाहरी सतह को संवहन विधि के माध्यम से गर्म करता है, जो वस्तु के केंद्र तक पहुँचता है। इसी प्रकार, सूक्ष्मजीवों में, गर्मी के प्रभाव से उनके भीतर का पानी वाष्पित हो जाता है, जिससे कोशिकीय घटकों में ऑक्सीडेटिव क्षति, प्रोटीन का विकृतिकरण, और इलेक्ट्रोलाइट के उच्च स्तर का विषाक्त प्रभाव उत्पन्न होता है, जो सूक्ष्मजीवों के नष्ट होने का कारण बनता है।



चित्र: तप्त वायु भट्ठी (बाहरी भाग)



चित्र: तप्त वायु भट्ठी (आंतरिक भाग)

यांत्रिक भाग:

कोट/कैबिनेट: बाहरी ढाल एल्यूमीनियम या स्टेनलेस स्टील से बनी होती है, जो यांत्रिक झटकों और ऑक्सीडेशन का प्रतिरोध करती है। यह बाहरी वातावरण से आंतरिक वातावरण को भी इंसुलेट करती है और गर्मी के नुकसान को रोकती है।

तन्तु कांच: बाहरी कैबिनेट और आंतरिक कक्ष के बीच की जगह मोटे कांच वूल इंसुलेशन से भरी होती है। दो प्रकार के तन्तु कांच होते हैं, अर्थात् भूरा तन्तु कांच और पीला तन्तु कांच। पीला तन्तु कांच भूरे की तुलना में कम खतरनाक होता है। भूरा तन्तु कांच श्वसन तंत्र में सूजन उत्पन्न करता है, जबकि पीला तन्तु कांच त्वचा में संवेदनशीलता का कारण बनता है। इसलिए इसके साथ काम करते समय हाथों में दस्ताने पहनना उचित होता है। यह उपकरण के अंदर की गर्मी को बाहर जाने से रोकने का कार्य भी करता है।

कक्ष: आयताकार आकार का कक्ष एल्यूमीनियम या स्टेनलेस स्टील से बना होता है, जिसमें शेल्स को वांछित स्तर पर रखने के लिए पसलियों के लिए जगह होती है।

शेल्स (जाली): ये प्लेटों को रखने के लिए बनाए गए वस्तु होते हैं और एल्यूमीनियम से बने होते हैं। वस्तुओं की संख्या और आकार के साथ-साथ भट्टी की क्षमता के आधार पर, इनकी संख्या भिन्न हो सकती है। जब इन्हें पसलियों पर रखा जाता है, तो हवा का संचलन कुछ क्षेत्रों को उठाकर सुगम बनाया जाता है। कुछ शेल्स में वायुप्रवेश के लिए खुली जगह भी हो सकती है।

मोटर चालित पंखे/ब्लोअर: पंखा मोटर द्वारा संचालित होता है और कक्ष के अंदर गर्म हवा को समान रूप से वितरित करने के लिए उपयोग किया जाता है।

दरवाजा: एक तरफ एकल दरवाजा भारी काजों पर लगाया गया है। ऑपरेशन के दौरान गर्मी के नुकसान को कम करने के लिए दरवाजे के किनारे पर एस्बेस्टस गास्केट लगाया जाता है।

इलेक्ट्रिक भाग:

विद्युत आपूर्ति: विद्युत आपूर्ति 220V-50Hz ट्रांसफार्मर और रेक्टिफायर के उपयोग से की जाती है।

तापक: एक संवाहक के माध्यम से विद्युत धारा के प्रवाह के साथ, तापमान में वृद्धि के साथ गर्मी उत्पन्न होती है। तापक तत्व की तीन मुख्य विशेषताएँ हैं: उच्च प्रतिरोध, विद्युत इंसुलेशन, और उच्च तापीय चालकता। तप्त वायु भट्ठी में उपयोग किए जाने वाले विभिन्न प्रकार के तापक हैं: एक तरफ गोलाकार प्रकार का तापक, एक तरफ यू प्रकार का तापक, एक तरफ तरंग प्रकार का तापक, एक तरफ चौकोर प्रकार का तापक, तीन तरफ का तापक, और चार तरफ का तापक। तापक 50 से 300 डिग्री सेल्सियस के तापमान पर काम करता है।

तापस्थापी: यह एक ताप संवेदक है जो सीधे तापक से जुड़ा होता है और अत्यधिक गर्मी का सामना कर सकता है, जिसमें उच्च नकारात्मक तापमान गुणांक होता है। यह उपयोगकर्ताओं को तप्त वायु भट्ठी में वांछित तापमान प्राप्त करने में सहायता करता है और तापमान के ओवरशूट को रोकता है।

तापमान संकेतक: भट्ठी के आंतरिक तापमान को निर्धारित करने के लिए या तो थर्मामीटर या तापयुग्म का उपयोग किया जा सकता है।

टाइमर: टाइमर के दो प्रकार हो सकते हैं: इलेक्ट्रिकल या मैकेनिकल, जो निष्फलन के लिए 5-60 मिनट तक कार्य कर सकते हैं।

फ्यूज: फ्यूज उच्च धारा के कारण विद्युत क्षति को रोकने का कार्य करता है, जैसे शॉर्ट सर्किट या उच्च लोड के दौरान।

नियंत्रण पैनल: यह क्षेत्र उपयोगकर्ता को विभिन्न प्राचल सेटिंग्स जैसे तापमान, समय आदि को नियंत्रित करने की अनुमति देता है, और इसमें एक संकेतक विद्युत लाइट (आमतौर पर हरा), संकेतक तापक लैंप (आमतौर पर लाल), और स्विच नॉब होता है।

संचालन प्रक्रिया:

तप्त वायु भट्ठी संचालित करने के कदम:

1. भट्ठी को सॉकेट में लगाया जाता है और चालू किया जाता है।
2. वस्तुओं को ट्रे या शेल्स पर रखने से पहले भट्ठी को 30 मिनट के लिए पहले से गरम करना किया जाता है।

3. तापमान गेज को वांछित समय पर सेट किया जाता है, जो निष्फलन के लिए सामग्री के मात्रा पर निर्भर करता है।
4. वस्तुओं को शेल्स पर रखा जाता है (ट्रे पर वस्तुओं के बीच उचित स्थान बनाए रखना चाहिए ताकि गर्मी का कुशलता से संचलन हो सके)।
5. दरवाजे को दिए गए स्कू को कसकर बंद किया जाता है, जिसके बाद तापमान बढ़ना शुरू होता है।
6. तप्त वायु भट्ठी के निष्फलन धारण समय और तापमान की सही जानकारी होना बेहतर है। यह सुनिश्चित करने के लिए थर्मामीटर की जांच की जाती है कि क्या वांछित तापमान एक निश्चित धारण समय के बाद प्राप्त किया गया है।
7. एक बार तापमान धारण अवधि प्राप्त होने के बाद, उपकरण को बंद कर दिया जाता है, दरवाजा खोलने से पहले ठंडा होने की अनुमति दी जाती है, और फिर नमूनों को भट्ठी दस्ताना या चिमटे का उपयोग करके निकाला जाता है।
8. नमूनों को निकालने के बाद दरवाजा बंद कर दिया जाता है।

अवलोकन:

पैरामीटर	अवलोकन विवरण
सामग्री	
प्रारंभिक स्थिति	
प्रक्रिया	
निष्फलन के बाद की स्थिति	
निष्फलन की प्रभावशीलता	
निष्कर्ष	

तप्त वायु भट्ठी के अनुप्रयोग:

1. **निष्फलन:** इसका उपयोग प्रयोगशाला उपकरणों जैसे कि ग्लासवेयर (पतली गर्दन की बोतल, नलिका, पेट्री प्लेट्स और परखनली), कल्चर मीडिया, धातु के सामान (फोर्सेप्स, स्पैचुला, स्कैल्पल, कैंची), गैर-वाष्पशील यौगिक (जिंक और स्टार्च पाउडर, सल्फोनामाइड) और अन्य तेल युक्त सामग्रियों के निष्फलन के लिए किया जाता है।

2. **खाद्य परीक्षण:** इसे खाद्य पदार्थों, औषधीय उत्पादों और अन्य उपभोक्ता सामग्रियों के तापमान स्थिरता सुनिश्चित करने के लिए परीक्षण में उपयोग किया जा सकता है।
3. **अनुसंधान में उपयोग:** यह जीवविज्ञान, रसायन विज्ञान और सामग्री विज्ञान के क्षेत्र में अनुसंधान सेटिंग्स में प्रयोग किया जा सकता है।
4. **उष्मा उपचार और सुखाने:** यह धातुओं, मिश्र धातुओं, मिट्टी और अन्य सामग्रियों के उष्मा उपचार और सुखाने के लिए भी इस्तेमाल किया जाता है।

तप्त वायु भट्टी के लाभ:

1. **पानी की आवश्यकता नहीं:** यह निष्फलन के लिए पानी की आवश्यकता नहीं होती, जैसे कि ऑटोक्लेव में होती है।
2. **आर्थिक और सरल संचालन:** यह संचालन में आर्थिक है और इसे चलाना आसान है।
3. **उच्च तापमान पर कार्य:** यह ऑटोक्लेव की तुलना में उच्च तापमान पर तेज़ी से कार्य कर सकता है।
4. **छोटा आकार:** ओवन का आकार छोटा होता है, जिससे इसे कम स्थान की आवश्यकता होती है और इसकी स्थापना सरल होती है।
5. **धातुओं का सुरक्षा:** शुष्क उष्मा धातुओं या अन्य तेज़ वस्तुओं को न तो जंग लगने देती है और न ही उन्हें नष्ट करती है।
6. **सुविधाजनक आकार:** इसका आकार छोटा होने के कारण काम करने में सहूलियत होती है।
7. **कम दबाव:** इसमें कम दबाव बनता है, जिससे संचालन के दौरान सुरक्षा सुनिश्चित होती है।
8. **गैर-विषैला:** इसमें कोई हानिकारक रासायनिक अवशेष नहीं छोड़े जाते हैं।
9. **गहरी निष्फलन:** सूखी गर्मी मोटी वस्तुओं के अंदर गहराई तक प्रवेश कर सकती है, जिससे गहरे निष्फलन का प्रभाव प्राप्त होता है।

निष्कर्ष:

1. अंकीय तप्त वायु भट्टी शुष्क उष्मा का उपयोग करके वस्तुओं को प्रभावी रूप से निष्फल और सुखाता है। प्रयोग ने यह दर्शाया कि यह निरंतर और समान तापमान प्रदान करता है, जिससे कांच के बर्तन, धातु उपकरण और नमी-संवेदनशील सामग्रियों का पूर्ण निष्फलन सुनिश्चित होता है।
2. यह प्रयोगशाला सेटिंग्स में निष्फलन और ताप उपचार के लिए एक विश्वसनीय और आवश्यक उपकरण है।

सावधानियाँ:

1. केवल वे सामग्री जिनका शुष्क उष्मा के निष्फलन के साथ संगतता है, का उपयोग किया जाना चाहिए। ज्वलनशील वस्तुओं का तप्त वायु भट्ठी में निष्फलन सख्त मना है।
2. वस्तुओं को कागज या समाचार पत्र में लपेटकर कार्डबोर्ड या धातु के कंटेनर में रखा जाना चाहिए। परखनली , पतली गर्दन की बोतल , और नालिका को प्लग करने के लिए कपास की ऊन का उपयोग किया जा सकता है।
3. व्यक्ति को दरवाजा खोलने से पहले प्रतीक्षा करनी चाहिए ताकि ओवन 40°C तक ठंडा हो जाए। यह काँच के सामान के टूटने से बचाता है।
4. भट्ठी से वस्तुओं को निकालने के लिए थर्मल दस्ताने या चिमटे का उपयोग करना चाहिए।
5. भट्ठी को कभी भी अधिभारित नहीं किया जाना चाहिए।
6. वस्तुओं को शेल्फ पर उचित दूरी पर रखा जाना चाहिए ताकि गर्म हवा का बिना बाधा के संचलन हो सके।

प्रयोग संख्या: 4

उद्देश्य: नैनोकणों की आकृति विज्ञान का अध्ययन करना और इसकी चिकनाई उत्पादों में प्रयोज्यता को समझना।

आवश्यक उपकरण: नैनोकण, गर्म हवा ओवन, अल्ट्रासोनिक बाथ, मैग्नेटिक स्टिरर, ओलिक एसिड।

परिचय: आधुनिक युग में ऊर्जा उत्पादन, निर्माण और परिवहन महत्वपूर्ण औद्योगिक गतिविधियाँ हैं, जो ऊर्जा रूपांतरण और स्थानांतरण से सीधे संबंधित हैं। घर्षण और पहनने के कारण ऊर्जा की हानि मशीनों की दक्षता बढ़ाने में एक बड़ा चुनौती है। सभी गतिविधियों में कई संपर्क सतहों की प्रचुरता होती है, जो अंततः घर्षण और पहनने के रूप में अपनी कार्यशील ऊर्जा खो देती हैं। यह कुल विश्व उपयोग योग्य ऊर्जा का 23% है। इसमें से 20% ऊर्जा केवल घर्षण के कारण खो जाती है, और शेष 3% पहनने और Tear तथा उसके बाद की हानियों में खर्च होती है। ट्राइबो सतहों में चिकनाई के उपयोग के कारण, घर्षण और पहनने में काफी कमी आती है। नैनोकण, बहुत छोटी मात्रा में, चिकनाई के प्रदर्शन को बढ़ाने की उनकी क्षमता के कारण महत्वपूर्ण साबित हुए हैं।

नैनोकण

परिभाषा: नैनोकण अत्यधिक छोटे कण होते हैं, जिनका आकार आमतौर पर 1 से 100 नैनोमीटर के बीच होता है। उनके अद्वितीय गुण, जैसे उच्च सतह क्षेत्र, बढ़ी हुई प्रतिक्रियाशीलता, और बेहतर यांत्रिक शक्ति, उन्हें चिकनाई उत्पादों में एडिटिव के रूप में एक मजबूत उम्मीदवार बनाते हैं। नैनोकणों का उपयोग डीजल और बायोडीजल में ईंधन दक्षता, इंजन के प्रदर्शन, उत्सर्जन, और दहन को सुधारने के लिए किया जा सकता है।

आकृतियाँ और उनके उपयोग

नैनोकणों की आकृतियाँ विभिन्न उद्देश्यों की पूर्ति में मदद करती हैं। उदाहरण के लिए, लंबे कार्बन नैनोट्यूब एक विद्युत जंक्शन को जोड़ने के लिए उपयोग किए जाते हैं। अमोर्फ कण आमतौर पर गोल आकार या नैनो c₆₀ को अपनाते हैं, जबकि एनीसोट्रोपिक माइक्रोक्रीस्टलाइन व्हिस्कर उनके विशेष क्रिस्टल आकार के अनुसार होते हैं। छोटे नैनोकण अक्सर क्लस्टर बनाते हैं, जो विभिन्न आकारों में हो सकते हैं, जैसे कि रॉड, फाइबर, और कप आदि। बारीक कणों का अध्ययन माइक्रोमेरिटिक्स कहलाता है। नैनोकणों के

आकार और आकृति को नियंत्रित करना उनके विभिन्न उभरते प्रौद्योगिकियों में उपयोग के लिए उनके गुणों का लाभ उठाने के लिए महत्वपूर्ण है।

नैनोकणों की विशेषताएँ

1. **प्लैटनस (Flatness):** यह बताता है कि नैनोकण का आकार एक सपाट सतह के कितना करीब है। इसे कण की ऊँचाई की तुलना उसकी पार्श्विक मापों से आंका जाता है। एक सपाट नैनोकण की ऊँचाई उसकी चौड़ाई और लंबाई के मुकाबले कम होगी।
 2. **गोलाई (Sphericity):** गोलाई कण के आकार की एक आदर्श गोले के साथ समानता को मापती है। इसे निम्नलिखित सूत्र से परिभाषित किया जाता है:
$$\text{गोलाई} = (\text{कण के समान वॉल्यूम का गोले का सतह क्षेत्र} / \text{कण का सतह क्षेत्र})$$

1 का मान आदर्श गोले को दर्शाता है, जबकि 1 से कम के मान गोलाई में बढ़ते विचलन को दर्शाते हैं।
- आस्पेक्ट अनुपात (Aspect Ratio):** यह नैनोकण की सबसे लंबी और सबसे छोटी माप का अनुपात है। यह आमतौर पर खींचे हुए आकारों का वर्णन करने के लिए उपयोग किया जाता है। उदाहरण के लिए, रॉड-नुमा नैनोकणों में, उच्च आस्पेक्ट अनुपात चौड़ाई के मुकाबले महत्वपूर्ण खींचाव को दर्शाता है।

नैनोकणों का वर्गीकरण

- **1D नैनो सामग्री:** ये एक आयामी होती हैं, जैसे कि पतली फिल्में या सतह कोटिंग्स। इनका उपयोग इलेक्ट्रॉनिक्स, रसायन विज्ञान, और अभियांत्रिकी में होता है।
- **2D नैनो सामग्री:** ये दो आयामी होती हैं और इनमें नैनोस्ट्रक्चर्ड फिल्में शामिल होती हैं, जो किसी सब्सट्रेट पर मजबूती से जुड़ी होती हैं, या नैनोपोर फिल्टर, जो छोटे कणों को अलग करने और फिल्टर करने के लिए उपयोग किए जाते हैं।
- **3D नैनो सामग्री:** जो सभी तीन आयामों में नैनोस्केल पर होती हैं, इन्हें 3D नैनो सामग्री माना जाता है। इनमें पतली फिल्में शामिल हैं जो एटॉमिक-स्केल पोरसिटी उत्पन्न करने की परिस्थितियों में जमा होती हैं, कोलॉइड्स, और विभिन्न आकृतियों के फ्री नैनोकण शामिल हैं।

नैनोकणों के भौतिक गुण

नैनोकण तीन परतों से बने होते हैं: सतही परत, शेल परत, और कोर। सतही परत में विभिन्न अणु जैसे धातु आयन, सर्फैक्टेंट्स, और पॉलिमर शामिल होते हैं। नैनोकण एकल सामग्री या कई सामग्रियों का संयोजन हो सकते हैं। ये नैनोकण उनकी रासायनिक और विद्युतचुम्बकीय गुणों के अनुसार निलंबन, कोलॉइड, या वितरित एरोसोल के रूप में अस्तित्व में हो सकते हैं।

नैनोकणों के गुण उनके आकार पर निर्भर करते हैं। उदाहरण के लिए, 50 नैनोमीटर से छोटे तांबे के नैनोकण सुपर हार्ड सामग्री होते हैं और वे बड़े ताम्बे की मॉलियाबिलिटी या डक्टिलिटी के गुण नहीं दिखाते। अन्य गुण जो आकार पर निर्भर करते हैं उनमें सुपरपैरामैग्नेटिज़्म, क्वांटम कॉन्फाइनमेंट, और कुछ धातु कणों में सतही प्लाज्मन रिसोनेंस शामिल हैं।

नैनोकणों की वर्गीकरण

नैनोकणों को कई मानदंडों के आधार पर वर्गीकृत किया जा सकता है, जैसे कि उनके आकार, आकृति, संघटन, और संश्लेषण की विधि। यहां कुछ सामान्य वर्गीकरण दिए गए हैं:

1. आकार के आधार पर:

- **अल्ट्राफाइन कण:** आमतौर पर 1-100 नैनोमीटर के आकार के।
- **नैनोकण:** 1-100 नैनोमीटर की श्रेणी में कणों को सामान्यतः संदर्भित किया जाता है।

2. आकृति के आधार पर:

- **गोल नैनोकण:** समान आकार, जो अक्सर दवा वितरण में उपयोग होते हैं।
- **नैनोरोड्स:** रॉड के आकार के नैनोकण जो अद्वितीय ऑप्टिकल गुण रख सकते हैं।
- **नैनोडिस्क:** सपाट, डिस्क-आकार के नैनोकण।
- **नैनोट्यूब:** सिलेंडर संरचनाएँ, जैसे कि कार्बन नैनोट्यूब।
- **नैनोप्लेट्स:** पतली, सपाट संरचनाएँ जो बड़े सामग्रियों की तुलना में विभिन्न गुण प्रदर्शित कर सकती हैं।

3. संघटन के आधार पर:

- **धात्विक नैनोकण:** धातुओं से बने (जैसे, सोना, चांदी, प्लेटिनम)।
- **ऑक्साइड नैनोकण:** धातु ऑक्साइड से बने (जैसे, टाइटेनियम डाइऑक्साइड, जिंक ऑक्साइड)।

- **पॉलीमर नैनोकण:** पॉलिमर से बने, जो अक्सर दवा वितरण प्रणालियों में उपयोग होते हैं।
 - **सिरेमिक नैनोकण:** विभिन्न अनुप्रयोगों के लिए अकार्बनिक यौगिक।
4. **संश्लेषण विधि के आधार पर:**
- **टॉप-डाउन विधियाँ:** बड़ी सामग्रियों को छोटे कणों में तोड़ना (जैसे, मिलिंग, लिथोग्राफी)।
 - **बॉटम-अप विधियाँ:** छोटे यौगिकों से नैनोकणों का निर्माण (जैसे, सोल-गेल, रासायनिक वाष्प निक्षेपण)।
5. **कार्यात्मकता के आधार पर:**
- **थेराप्यूटिक नैनोकण:** दवा वितरण या उपचार के लिए डिज़ाइन किए गए (जैसे, लिपोसोम, डेंड्रिमर)।
 - **डायग्नोस्टिक नैनोकण:** इमेजिंग या सेंसिंग अनुप्रयोगों में उपयोग किए जाते हैं (जैसे, क्वांटम डॉट)।
 - **कैटालिटिक नैनोकण:** रासायनिक प्रतिक्रियाओं में उत्प्रेरक के रूप में कार्य करते हैं (जैसे, ईंधन कोशिकाओं में प्लेटिनम नैनोकण)।
6. **सतही गुणों के आधार पर:**
- **फंक्शनलाइज्ड नैनोकण:** विशिष्ट अणुओं के साथ लेपित, जैविक प्रणालियों के साथ इंटरैक्शन को बढ़ाने के लिए।
 - **बैर नैनोकण:** बिना किसी सतही संशोधनों के, आमतौर पर मूल गुणों को दिखाते हैं।

नैनोकणों की संश्लेषण तकनीकें

नैनोकणों के संश्लेषण के लिए कई विधियाँ हैं, जिनमें से प्रत्येक के अपने फायदे और अनुप्रयोग होते हैं। यहां कुछ सामान्य संश्लेषण विधियाँ दी गई हैं:

1. भौतिक विधियाँ:

- **मिलिंग:** बड़े सामग्रियों को नैनोस्केल कणों में प्राप्त करने के लिए यांत्रिक पीसना।
- **लेजर एब्लेशन:** उच्च-ऊर्जा लेज़रों का उपयोग करके लक्षित सामग्रियों को वाष्पित करना, जो फिर नैनोकणों में संघनित होते हैं।
- **फिजिकल वेपर डिपोजिशन (PVD):** एक प्रक्रिया जिसमें सामग्री को वैक्यूम में वाष्पित किया जाता है और सबस्ट्रेट पर नैनोकणों के रूप में जमा किया जाता है।

2. रासायनिक विधियाँ:

- **सोल-गेल प्रक्रिया:** एक समाधान (सोल) को ठोस (जेल) चरण में परिवर्तित करना, जो नैनोकणों के निर्माण को सुगम बनाता है।
- **स्खलन:** दो या अधिक समाधानों को मिलाकर एक ठोस उत्पन्न करना जो नैनोकणों के रूप में निकासी होती है।
- **हाइड्रोथर्मल संश्लेषण:** उच्च दबाव और तापमान वाले पानी का उपयोग करके सामग्रियों को घुलाकर नैनोकणों की वृद्धि को बढ़ावा देना।

3. जैविक विधियाँ:

- **बायोमिमेटिक संश्लेषण:** जैविक जीवों (जैसे बैक्टीरिया, फफूंद, या पौधे) का उपयोग करके प्राकृतिक चयापचय प्रक्रियाओं के माध्यम से नैनोकणों का उत्पादन करना।
- **एंजाइमेटिक प्रक्रियाएँ:** एंजाइम धातु आयनों के कमी में मदद कर सकते हैं, जो नैनोकणों के निर्माण की ओर ले जाता है।

4. इलेक्ट्रोकेमिकल विधियाँ:

- **इलेक्ट्रोकेमिकल डिपोजिशन:** एक समाधान पर वोल्टेज लागू करना जिसमें धातु आयन होते हैं, जिससे वे कमी कर सकते हैं और एक इलेक्ट्रोड पर नैनोकणों का निर्माण कर सकते हैं।

5. सेल्फ-एसेम्बली:

- अणुओं का स्वाभाविक रूप से संरचित व्यवस्थाओं में संगठित होना, जो नैनोकणों के निर्माण की ओर ले जाता है।

इन विधियों को विशिष्ट सामग्रियों और इच्छित गुणों के लिए अनुकूलित किया जा सकता है, जैसे आकार, आकृति, और कार्यात्मकता। विधि का चयन अक्सर नैनोकणों के लक्षित अनुप्रयोग पर निर्भर करता है।

नैनो चिकनाई (Nanolubricants)

परिभाषा: नैनो चिकनाई एक समरूप मिश्रण है जिसमें आधार तेल, नैनोकण, और प्रसारक या सर्फैक्टेंट्स शामिल होते हैं। नैनो चिकनाई के निर्माण में दो चरणों की प्रक्रिया होती है:

1. **सतह संशोधन:** सबसे पहले, नैनोकणों (जैसे Al_2O_3) की सतह को ओलिक एसिड से कोट किया जाता है, जो मैग्नेटिक स्टिरर का उपयोग करके किया जाता है और फिर गर्म हवा के ओवन में सुखाया जाता है।
2. **वितरण:** सूखे नैनोकणों के पाउडर को आधार चिकनाई तेल में मिलाया जाता है, जिसके लिए मैग्नेटिक स्टिरर और अल्ट्रासोनिक पानी बाथ का उपयोग किया जाता है।

चिकनाई तंत्र

चिकनाई तंत्र का स्रोत चिकनाई के अणुओं, सामग्री की सतहों, और वातावरण के बीच के भौतिक और रासायनिक इंटरैक्शन से होता है। नैनोकणों के ट्राइबोलॉजिकल सतहों के भीतर चार तंत्र विकसित होते हैं:

1. **गोलाकार नैनोकण:** ये नैनोकण ट्राइबोलॉजिकल सतहों के बीच छोटे गेंदों की तरह कार्य करते हैं, जिससे स्लाइडिंग क्रिया को रोलिंग क्रिया में बदलते हैं और घर्षण गुणांक में कमी आती है।
2. **सुरक्षात्मक परत:** कभी-कभी नैनोकण एक सुरक्षा परत बनाते हैं जो संपर्क में आते हैं।
3. **मेंडिंग प्रभाव:** नैनोकण असपरिटीज़ को भरते हैं, और घर्षण दरारें एक चिकनी सतह में परिणत होती हैं।
4. **पॉलिशिंग प्रभाव:** नैनोकणों की एब्रेसिव क्रिया से सतही खुरदरापन कम होता है।

ओलिक एसिड और अन्य बाइंडर्स तथा सर्फैक्टेंट्स इन नैनोकणों को जोड़ने और उनके जीवनकाल को बढ़ाने में मदद करते हैं।

नैनोकणों के अनुप्रयोग

नैनोकणों के अद्वितीय गुणों के कारण उनके कई क्षेत्रों में व्यापक अनुप्रयोग हैं। यहाँ कुछ प्रमुख अनुप्रयोग दिए गए हैं:

1. **चिकित्सा:**
 - **दवा वितरण:** नैनोकणों को लक्षित कोशिकाओं तक दवाएं पहुँचाने के लिए तैयार किया जा सकता है, जिससे प्रभावशीलता बढ़ती है और दुष्प्रभाव कम होते हैं।
 - **इमेजिंग:** MRI और CT स्कैन जैसी इमेजिंग तकनीकों में कंट्रास्ट एजेंट के रूप में उपयोग होते हैं।

- **थेराप्यूटिक्स:** कुछ नैनोकणों को लक्षित चिकित्सा या फोटोथर्मल थेरपी के माध्यम से बीमारियों का इलाज करने के लिए डिज़ाइन किया गया है।
- 2. **इलेक्ट्रॉनिक्स:**
 - **ट्रांजिस्टर और सेंसर:** नैनोकण इलेक्ट्रॉनिक उपकरणों के प्रदर्शन को बढ़ाने में मदद करते हैं।
 - **सौर कोशिकाएँ:** नैनोसंरचित सामग्री सौर ऊर्जा रूपांतरण की दक्षता को बढ़ाती है।
- 3. **पर्यावरणीय अनुप्रयोग:**
 - **जल उपचार:** नैनोकण जल से भारी धातुओं और रोगाणुओं को हटाने में मदद कर सकते हैं।
 - **प्रदूषण नियंत्रण:** प्रदूषकों के विघटन या कणों को पकड़ने में सहायक होते हैं।
- 4. **कॉस्मेटिक्स:**
 - **सन्सक्रीन:** जिंक ऑक्साइड और टाइटेनियम डाइऑक्साइड के नैनोकण UV-blocking गुणों के लिए उपयोग होते हैं।
 - **एंटी-एजिंग उत्पाद:** ये सक्रिय सामग्रियों के वितरण को सुधारने में मदद करते हैं।
- 5. **खाद्य उद्योग:**
 - **पैकेजिंग:** नैनोकण पैकेजिंग सामग्रियों के बैरियर गुणों को बढ़ाते हैं।
 - **खाद्य सुरक्षा:** रोगाणुओं या खराबी का पता लगाने के लिए सेंसर में उपयोग होते हैं।
- 6. **कपड़े:**
 - **दाग और पानी प्रतिरोध:** नैनोकण कपड़ों में दाग प्रतिरोध या एंटीमाइक्रोबियल प्रभाव डाल सकते हैं।
- 7. **निर्माण:**
 - **कंक्रीट की मजबूती:** नैनोकण कंक्रीट के यांत्रिक गुणों को बढ़ाते हैं।
- 8. **कृषि:**
 - **कीटनाशक और उर्वरक:** नैनोकण कृषि रसायनों की दक्षता और वितरण को सुधारते हैं।
- 9. **ट्रिबोलॉजिकल:**
 - **चिकनाई और ईंधन:** नैनोकण चिकनाई में प्रदर्शन बढ़ाने वाले एडिटिव के रूप में उपयोग हो सकते हैं।

चुनौतियाँ

हालांकि नैनोकणों में कई अनुप्रयोगों के लिए संभावनाएँ हैं, वे कुछ चुनौतियों का सामना भी करते हैं:

1. **विषाक्तता:** कुछ नैनोकण मानव स्वास्थ्य और पर्यावरण के लिए विषैला हो सकते हैं।
2. **पर्यावरणीय प्रभाव:** उनकी छोटी आकार उन्हें पारिस्थितिक तंत्र में आसानी से प्रवेश करने की अनुमति देती है।
3. **निर्माण चुनौतियाँ:** नैनोकणों का निरंतर और सुरक्षित उत्पादन तकनीकी रूप से जटिल और महंगा हो सकता है।
4. **नियामक मुद्दे:** नैनोटेक्नोलॉजी का विकास नियामक ढाँचे से तेज़ी से आगे बढ़ रहा है।
5. **सीमित समझ:** नैनोकणों के आणविक स्तर पर व्यवहार और इंटरैक्शन पूरी तरह से समझ में नहीं आए हैं।
6. **पर्यावरणीय स्थिरता:** कुछ नैनोकण पर्यावरण में लंबे समय तक रह सकते हैं।
7. **जनता की धारणा:** नैनो सामग्री की सुरक्षा को लेकर अक्सर सार्वजनिक चिंताएँ होती हैं।
8. **लागत:** जबकि कुछ अनुप्रयोग आर्थिक रूप से व्यवहार्य हो सकते हैं, नैनोटेक्नोलॉजी अनुसंधान और विकास में प्रारंभिक निवेश उच्च हो सकता है।

इन कारकों को नैनोकण प्रौद्योगिकियों के विकास और अनुप्रयोग में ध्यान में रखा जाना चाहिए।

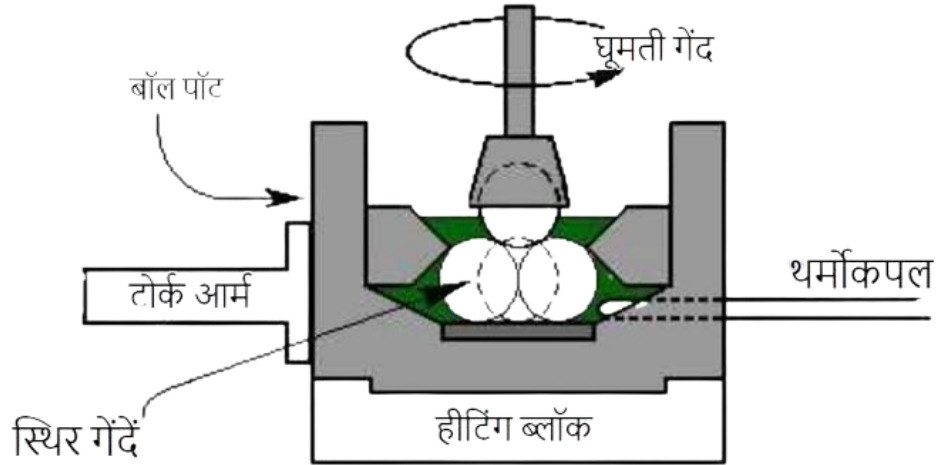
प्रयोग संख्या: 5

उद्देश्य: चार बॉल टेस्टर का उपयोग करके स्नेहन तेल की घर्षण प्रतिरोध और घर्षण सम्बन्धी विशेषताओं का प्रायोगिक अध्ययन।

आवश्यक उपकरण: डेटा अधिग्रहण प्रणाली के साथ चार बॉल परीक्षण यंत्र

सिद्धांत:

चार बॉल मशीन एक उपकरण है जिसमें उच्च दबाव के तहत स्लाइडिंग गति उत्पन्न की जाती है, जो आमतौर पर स्टील या बॉल के आकार के अन्य परीक्षण नमूनों के बीच होती है। यह संपर्क की स्थिति उन स्थितियों को काफी हद तक अनुकरण करती है, जो गियर्स और धातु काटने के ऑपरेशन में होती हैं। इस मशीन का प्राथमिक उद्देश्य स्नेहन द्रवों जैसे गियर ऑयल की एंटी-सीज़िंग गुणों का मूल्यांकन करना है। नमूना धारक को आवश्यकता अनुसार अन्य आकार के परीक्षण नमूनों को समायोजित करने के लिए अनुकूलित किया जा सकता है। यह मशीन निम्न लोड पर और लंबे समय तक चलने वाले परीक्षण के रूप में घिसाव परीक्षण के लिए भी इस्तेमाल की जा सकती है। परीक्षण परिणाम विभिन्न भारों पर घिसाव का संकेत देते हैं, जिस भार पर स्क्रैफिंग या सीज़िंग होती है, और उस दबाव को जहां एक स्नेहक सतह बिना सीज़िंग के संचालन कर सकती है। धात्विक संपर्क से पहले प्राप्त स्कार इंप्रेशन स्नेहक की घिसाव-रोधी विशेषताओं को दर्शाते हैं। स्कार का आकार स्नेहक की घिसाव-रोधी क्षमता को दर्शाता है; बड़े व्यास का संकेत घिसाव-रोधी गुणों की खराबी का होता है, जबकि छोटा व्यास बेहतर घिसाव-रोधी गुणों को दर्शाता है।



चार गेंद परीक्षण उपकरण का योजनाबद्ध आरेख

प्रक्रिया: घर्षण और घिसाव गुणों के मूल्यांकन के लिए यांत्रिक परीक्षण निम्नलिखित दो तरीकों से किए जा सकते हैं:

1. अल्प अवधि का वेल्ड लोड परीक्षण
2. दीर्घ अवधि का एंटी-वेअर परीक्षण

अल्प अवधि का परीक्षण 10 सेकंड के लिए विशेष परिचालन स्थितियों में किया जाता है जब तक कि गेंदें वेल्ड न हो जाएं या मोटर ट्रिप न हो जाए। दीर्घ अवधि का परीक्षण एक घंटे की अवधि तक विशेष परिचालन स्थितियों में किया जाता है।

प्रक्रियात्मक कदम:

1. चार नई परीक्षण गेंदों, बॉल पॉट और कोलेट को हेक्सेन से साफ करें ताकि मौजूद अशुद्धियाँ हटाई जा सकें। एक स्वच्छ गेंद को बेस प्लेट पर लगे बॉल एक्सट्रैक्टर का उपयोग करके कोलेट में डालें और इसे स्पिंडल में लगाएं।

2. बेस प्लेट पर दिए गए स्लॉट में बॉल पॉट असेंबली को रखें। लॉक नट को हटा दें, बॉल रेस को लगभग 5 मिमी उठाएं, बॉल पॉट की गुहा में तीन गेंदें डालें और बॉल रेस को नीचे करके गेंदों की स्थिति में रखें। टॉर्क रिंच को आवश्यक टॉर्क मान पर सेट करें और लॉक करें। बॉल पॉट पर लॉक नट लगाएं और इसे हाथ से तब तक घुमाएं जब तक कि यह गेंद की नोक को छू न ले और फिर टॉर्क रिंच से कसें।
3. बॉल पॉट असेंबली को स्नेहक या परीक्षण तेल से गेंद की नोक से 3 मिमी ऊपर तक भरें। लोडिंग आर्म को उठाएं और लॉक करें। बॉल पॉट असेंबली को एंटी-फ्रिक्शन डिस्क पर रखें और सुनिश्चित करें कि बॉल पॉट और सेंसर के बीच संपर्क बना हुआ है और हीटर चालू है।
4. प्रयोग के दौरान डेटा अधिग्रहण के लिए इलेक्ट्रॉनिक कंट्रोलर की तैयारी सुनिश्चित करें। परीक्षण पैरामीटर चुनें और डेटा को सुरक्षित रखने के लिए एक फ़ाइल नाम दर्ज करें। कंट्रोलर पर स्टार्ट स्विच दबाएं ताकि परीक्षण निर्धारित समय के लिए चल सके जब तापमान पैनेल पर दिखने लगे।
5. परीक्षण समाप्त होते ही मोटर स्वचालित रूप से रुक जाती है और डेटा अधिग्रहण बंद हो जाता है। पैनेल बोर्ड को बंद करें। फ्रंट गार्ड को खोलें और बॉल पॉट असेंबली और कोलेट को हटा दें। मृत भारों को हटा दें, आर्म को ऊपर उठाएं और लॉक करें। कोलेट से बॉल को कोलेट एक्सट्रेक्टर का उपयोग करके और बॉल पॉट असेंबली से टॉर्क रिंच का उपयोग करके अलग करें। गेंदों से परीक्षण स्नेहक को पोंछ लें और घिसाव निशान माप के लिए उन्हें तैयार करें।

प्रयोग संख्या: 6

उद्देश्य: वायु जेट क्षरण परीक्षक का उपयोग करके विभिन्न सामग्रियों पर ठोस कण क्षरण घटना का प्रायोदगक अध्ययन

आवश्यक उपकरण: एयर जेट इरोशन परीक्षक, डीह्यूमिडिफाइड एयर, परीक्षण नमूने, अपघर्षी कण जैसे एल्युमिना, सिलिका रेत आदि।

सिद्धांत: एयर जेट इरोशन परीक्षक दोहराए जाने वाले प्रभाव कटाव विधि का उपयोग करता है, जिसमें एक छोटी नोजल गैस की एक धारा में अपघर्षी कण भेजती है, जो परीक्षण नमूने की सतह से टकराते हैं। यह उपकरण नियंत्रित परिस्थितियों में नमूने की सतह से सामग्री को क्षरण (इरोड) करने में सक्षम है। अध्ययन कणों की प्रभावी गति, कण सांद्रता, नमूने की स्थिति और दिशा को प्रभावित करने वाली धारा के सापेक्ष बदलकर किया जा सकता है।

उपकरण सेटअप: एयर जेट इरोशन परीक्षक ASTM G76 परीक्षण विधि के अनुरूप है। यह उपकरण विभिन्न सामग्रियों और कोटिंग्स पर ठोस कणों द्वारा कटाव को मापने और इरोसिविटी के आधार पर सामग्री को रैंक करने के लिए उपयुक्त है। उपकरण में कण गति और डिस्चार्ज, साथ ही नमूने की स्थिति और दिशा को नियंत्रित और समायोजित करने की सुविधाएं होती हैं। इसे तीन कक्षों में विभाजित किया जा सकता है: पहला कक्ष इलेक्ट्रिकल मोटर यूनिट, एरोडेंट डिस्चार्ज यूनिट, फीडर यूनिट, हॉपर, फ़िल्टर रेगुलेटर यूनिट, डिजिटल डिस्प्ले यूनिट, पाइपलाइन आदि को समायोजित करता है। दूसरा कक्ष मिश्रण कक्ष, नोजल धारक, नमूना एडॉप्टर आदि को समायोजित करता है। फीडर यूनिट से निकले कण मिश्रण कक्ष में गिरते हैं और हवा के साथ मिश्रित होकर नोजल से उच्च गति पर निकलते हैं। तीसरा कक्ष टेपर फ़नल होता है, जहाँ से कण परीक्षण के बाद बाहर निकलते हैं। नोजल का आंतरिक व्यास 1.5 मिमी होता है और लंबाई 50 मिमी होती है।

प्रक्रिया:

- सुनिश्चित करें कि कंप्रेसर में उपयुक्त दबाव (8 बार) है और फ़िल्टर रेगुलेशन यूनिट को खोलें।
- कंट्रोलर बॉक्स पर मुख्य वायु प्रवाह चालू करें। वायु जेट की गति सेट करने के लिए कंट्रोलर बॉक्स पर प्रेशर रेगुलेटर वाल्व घुमाएँ।
- सही आकार के नमूने का चयन करें, इसे सटीक तुला से 0.001 मि.ग्रा. सटीकता से तौलें और होल्डर में रखें। नोजल होल्डर में सही आकार की नोजल को कसें।
- ऑपरेशन के दौरान धूल को अलग करने के लिए सामने का दरवाजा बंद करें।

- इलेक्ट्रॉनिक कंट्रोलर को चालू करें और आवश्यक एरोडेंट डिस्चार्ज दर और टाइमर सेट करें। स्टार्ट स्विच दबाएं, एरोडेंट डिस्चार्ज होना शुरू हो जाएगा। मुख्य वायु आपूर्ति का लीवर चालू करें जिससे वायु मिश्रण कक्ष में प्रवाहित होकर एरोडेंट की गति बढ़ाए।
- परीक्षण की अवधि समाप्त होने पर, दरवाजा खोलें, नमूने को साफ करें और होल्डर से निकालें। परीक्षण के बाद नमूने को तौलें, जो द्रव्यमान ह्रास और घिसाव के रुझानों का संकेत देता है।

पर्यवेक्षण:

क्रमांक	परीक्षण अवधि (मिनट)	वायु दबाव (बार)	गति (मीटर/सेकंड)	आवृत्ति (हर्ट्ज)	डिस्चार्ज दर (ग्राम/मिनट)
1					
2					
3					
4					

परिणाम और चर्चा:

सावधानियाँ:

- कार्य क्षेत्र में प्रकाश और उचित वेंटिलेशन की जाँच करें।
- उपयुक्त विद्युत आपूर्ति 415V, 3-फेज, 50Hz की जाँच करें और सुनिश्चित करें कि सभी वायरिंग और केबल सही स्थिति में हैं, उचित ग्राउंडिंग की गई है और सही रूप से जुड़े हुए हैं।
- समय पर उपयोग के लिए उपकरण और सहायक उपकरणों की जाँच करें।
- परीक्षण से पहले सुरक्षा दस्ताने, जूते और कान प्लग की उपलब्धता सुनिश्चित करें।
- स्विच और लीवर के सुचारू संचालन की जाँच करें।
- इलेक्ट्रॉनिक कंट्रोलर के साथ परीक्षक की उचित कनेक्टिविटी सुनिश्चित करें और पैनल पर कुछ रीडिंग प्रदर्शित हों।
- परीक्षण के लिए पर्याप्त मात्रा में परीक्षण कण और परीक्षण नमूने की उपलब्धता सुनिश्चित करें।
- कंप्रेसर की सुरक्षित कार्यक्षमता की जाँच करें।

4. नैनो-कंपोजिट प्रयोगशाला

नियमावली

नैनो कम्पोजिट प्रयोगशाला

यान्त्रिक अभियांत्रिकी विभाग

मैनिट, भोपाल

उपकरणों की सूची:

1. क्षैतिज बॉल मिल (टम्बलर बॉल मिल)
2. आर्गन सिलेंडर के साथ नियंत्रित वायुमंडल प्रोग्रामयोग्य ट्यूब फर्नेस
3. ऊर्ध्वाधर पिघलने वाली भट्ठी
4. अल्ट्रासोनिक असिस्टेड स्टिर कास्टिंग सेट-अप
5. कम्प्यूटरीकृत यूनिवर्सल टेस्टिंग मशीन (यूटीएम)
6. डिजिटल रॉकवेल सह ब्रिनेल कठोरता परीक्षण मशीन
7. मैनुअल कठोरता परीक्षण मशीन
8. मफल फर्नेस
9. इलेक्ट्रॉनिक तुला
10. अपघर्षक काटने की मशीन
11. संपीडन मोल्लिंग मशीन
12. स्वचालित एक्सट्रूजन लाइन
13. डिजिटल टैकोमीटर
14. हॉट प्लेट के साथ डिजिटल चुंबकीय स्टिरर
15. बेल्ट ग्राइंडर
16. माइक्रोवेव ओवन
17. इन्फ्रा-रेड गन
18. अल्ट्रासोनिक क्लीनर
19. ड्रिल के सेट के साथ हैंड ड्रिल
20. वैक्यूम ओवन
21. एग्लोमरेटर
22. हीट डिफ्लेक्टर तापमान परीक्षक (एचडीटी)

1. क्षैतिज बॉल मिल (टम्बलर बॉल मिल)

विशेष विवरण:

व्यास: 320 मिमी

चौड़ाई: 120 मिमी

गति: 100 आरपीएम (लगभग) (गति नियामक के साथ)

बॉल मिल में नॉन रिटर्न वाल्व के माध्यम से निष्क्रिय वातावरण के लिए आर्गन गैस भरने का प्रावधान।

बॉल मिल की सामग्री: स्टेनलेस स्टील

बॉल्स की सामग्री: स्टेनलेस स्टील।

दो या तीन अलग-अलग व्यास की गेंदें प्रदान की जानी चाहिए।

घिसी-पिटी गेंदों के स्थान पर अतिरिक्त गेंदें उपलब्ध करायी जानी चाहिए।

आर्गन गैस भरने के लिए होज़ पाइप उपलब्ध कराए जाने चाहिए।

2. नियंत्रित वायुमंडल प्रोग्रामयोग्य ट्यूब भट्टी

विशेष विवरण

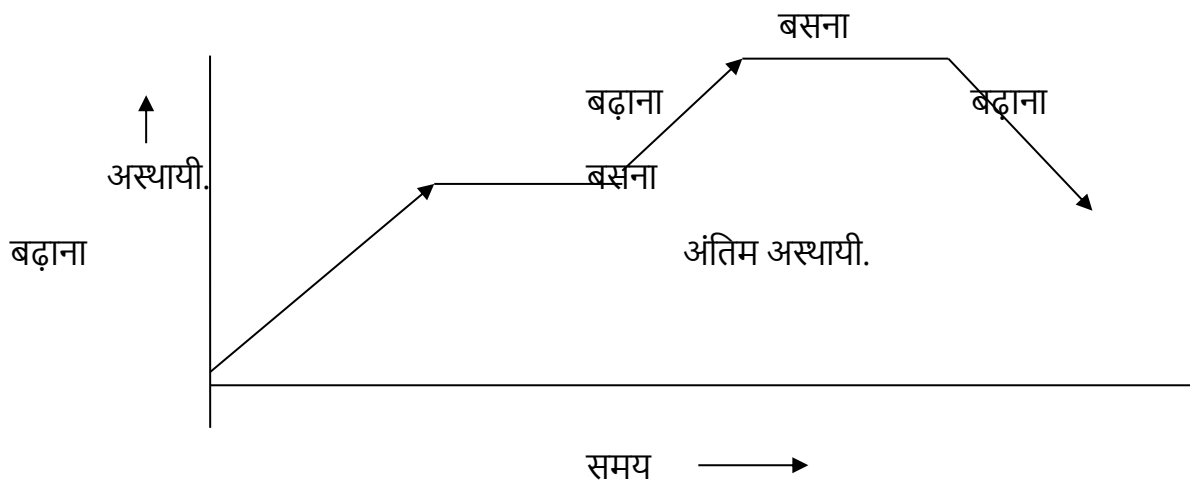
1. अधिकतम तापमान: 1500°सी
2. ताप क्षेत्र का आयाम: व्यास = 100 मिमी, लंबाई = 180 मिमी
3. भट्टी की लंबाई = 500 मिमी (लगभग)
4. वैक्यूम पंप द्वारा हवा को निकालने और अक्रिय गैस (आर्गन) को शुद्ध करने की व्यवस्था।
5. भट्टी को अक्रिय गैस को शुद्ध किए बिना, गर्म करते समय वैक्यूम पंप को लगातार 'चालू' करते हुए, वैक्यूम पर चलाने का प्रावधान।
6. वैक्यूम पंप में प्रवेश करने से पहले गैस या हवा को ठंडा करने की व्यवस्था।
7. उपयुक्त वैक्यूम पंप (10-3 टोर वैक्यूम)।
8. निर्वात को मापने का प्रावधान.
9. हीटिंग और कूलिंग की दर को नियंत्रित करने के लिए भट्टी को प्रोग्राम करने योग्य होना चाहिए।
10. अधिकतम पर सेट करने की व्यवस्था. वांछित तापमान.
11. एक आर्गन सिलेंडर (अक्रिय गैस आपूर्ति के लिए)।
12. आर्गन सिलेंडर और होसेस के लिए एक आर्गन नियामक।
13. भट्टी के बाहरी आवरण का तापमान न्यूनतम (60 से अधिक नहीं) होना चाहिए°सी)।
14. 3 साल की गारंटी.

पीआईडी नियंत्रक की प्रोग्रामिंग की प्रक्रिया

सिंट्रिंग भट्टी

1. प्रवेश करना। सेट प्वाइंट आ जाएगा.
2. फिर एंटर करें. प्रोग्राम नंबर पूछा जाएगा. इनपुट प्रोग्राम नंबर (1 से 4 के बीच)।
3. दोनों बटन दबाएँ `` और ऊपर ↑ दर्ज करें।
4. फिर प्रोग्राम जारी रहेगा और स्क्रीन पर समाप्ति आ जाएगी।
5. प्रोग्राम का चयन करें.
6. (यदि Loc कोड है तो मशीन ULoc पूछेगी, ULOC कोड यानि मशीन कोड डालें। फिर Enter करें।)
7. फिर स्क्रीन पर रैम्प (यानि तापमान परिवर्तन की दर) आ जाएगा। यह हमारी आवश्यकता के लिए वृद्धि अर्थात ऊपर ↑ या नीचे ↓ यानी कमी का प्रबंधन करेगा। अधिकतम रैंप 250 (बेहतर 200) होना चाहिए।

8. तब हम अंतिम तापमान प्राप्त करते हैं या बदल सकते हैं। एंटर दबाये।
9. तब हमें रहने का समय मिलता है। एंटर दबाये।
10. डेल टाइम इनपुट करें. एंटर दबाये।
11. फिर स्क्रीन पर फिर से रैंप दिखाएं। इनपुट रैंप (अधिकतम 250)। एंटर दबाये।
12. फिर से अंतिम तापमान आ रहा है।
13. फिर से हमें रहने का समय मिलता है। (यहां समाप्त करने के लिए नीचे तीर दबाएं)
14. **(नोट: समाप्त करने के लिए रैंप 0 लगाएं, फिर रुकें समय आएगा, फिर नीचे तीर दबाएं, अंत आ जाएगा)**
15. फिर स्क्रीन शो चक्र. एंटर दबाये। चक्र दर्ज करें 1. समय (घंटा) यानी आधार समय दिखाएँ। (एंटर दबाये)।
16. `111 E_01 एंटर दबाएँ।
17. यूएलओसी 0 दिखा रहा है।
18. रन बटन पर क्लिक करें या दबाएँ।
19. प्रोग्राम नंबर चुनें. फिर लगभग 10 सेकंड के लिए फिर से रन दबाएं ताकि हमारा वांछित प्रोग्राम लाइट शो तक चलता रहे।



3. ऊर्ध्वाधर पिघलने वाली भट्टी

ऊर्ध्वाधर पिघलने वाली भट्टी: एचएस कोड: 8417

कंपनी: मेट्रेक्स

भीतरी मफल आकार: (डब्ल्यू x एच x डी) 200 X 200 X 250 मिमी

अधिकतम. तापमान 1400 डिग्री सेल्सियस तक; कार्य तापमान: 1350 डिग्री सेल्सियस

अस्थायी. सटीकता: ± 1 डिग्री सेल्सियस

हीटिंग तत्व: सिलिकॉन कार्बाइड, अंत कनेक्शन के साथ, आसानी से प्रतिलिपि बनाने योग्य

आंतरिक कक्ष: सिरेमिक बोर्ड, जिसके दोनों तरफ हीटिंग तत्व रखे गए हैं।

इन्सुलेशन: ज़िरकोनिया ग्रेड का सिरेमिक फाइबर

निर्माण: न्यूनतम त्वचा तापमान के लिए बाहरी तरफ वायु इन्सुलेशन के साथ ट्रिपल दीवार, बाहरी शरीर हल्के स्टील से बना है जो विधिवत पाउडर लेपित है।

10 नंबर का कूसिबल. 2 लीटर की क्षमता के साथ पिघल को निकालने के लिए निचले छेद वाले कक्ष में लंबवत रखा जाता है, जिसे लीवर और स्प्रिंग सिस्टम से जुड़े स्टेनलेस स्टील रॉड से प्लग किया जाता है।

5 किलोग्राम तक एल्युमीनियम को पिघलाया जा सकता है।

गैस प्रवाह मीटर के माध्यम से कक्ष में आर्गन गैस प्रवाह की व्यवस्था।

ढक्कन: स्टिरर, थर्मोकपल आदि डालने की अनुमति देने के लिए शीर्ष ढक्कन दो भागों में है।

एससीआर इकाई के साथ पावर नियंत्रण (चरण कोण थाइरिस्टर)

अस्थायी. नियंत्रण: माइक्रोप्रोसेसर आधारित पीआईडी प्रोग्रामयोग्य नियंत्रक सेट और वास्तविक का दोहरा प्रदर्शन।

अस्थायी. कोमल स्पर्श कुंजी के साथ. थर्मोकपल ब्रेक प्रोटेक्शन और कोल्ड जंक्शन मुआवजे के साथ। सेंसर: पीटी/पीटी-आरएच 13%।

सुरक्षा: ओवर करंट और शॉर्ट सर्किट से सुरक्षा

थर्मोकपल ब्रेक अलार्म, उच्च तापमान। खतरे की घंटी

बिजली आपूर्ति: 220 वोल्ट एसी सिंगल फेज आपूर्ति।

2 लीटर क्षमता के 1 सिलिकॉन कार्बाइड कूसिबल के साथ

4. अल्ट्रासोनिक असिस्टेड स्टिर कास्टिंग सेट-अप

अल्ट्रासोनिक जांच:

उच्च शक्ति वाले अल्ट्रासोनिक का उपयोग पिघली हुई धातु की क्रिस्टल प्रक्रिया के उपचार के लिए किया जाता है, जो धातु के दाने, समान मिश्र धातु संरचना को उल्लेखनीय रूप से परिष्कृत कर सकता है, बुलबुले की गति को तेज कर सकता है। धातु सामग्री की ताकत, कठोरता, लचीलापन, क्रूरता आदि में उल्लेखनीय सुधार होता है।

प्रमुख अनुप्रयोग:

1. उच्च शक्ति एल्यूमीनियम, मैग्नीशियम मिश्र धातु की फाउंड्री
2. एल्यूमीनियम और मैग्नीशियम मिश्र धातु बार, शीट धातु का उत्पादन करें
3. विभिन्न मिश्र धातु सामग्री और मोटर रोटार क्रिस्टलीकरण को नष्ट करना
4. फाउंड्री विभिन्न धातु कंपोजिट और उच्च शक्ति एल्यूमीनियम पिस्टन

उत्पाद पैरामीटर:

1. 550°C-850°C स्थिर कार्य
2. बुद्धिमान जनरेटर
3. अल्ट्रासोनिक आवृत्ति: 20KHz
4. पावर: 2000 डब्ल्यू

उच्च शक्ति अल्ट्रासोनिक धातु पिघल उपचार उपकरण मुख्य भाग:

जनक

यह 50-60Hz बिजली को उच्च-शक्ति की उच्च-आवृत्ति बिजली आपूर्ति में स्थानांतरित कर सकता है, और फिर ट्रांसड्यूसर को प्रदान कर सकता है। जिसमें रेक्टिफायर सर्किट, ऑसिलेटिंग सर्किट, एम्प्लीफायर सर्किट, फीडबैक सर्किट, ट्रैक सर्किट, प्रोटेक्शन सर्किट, मैचिंग सर्किट, डिस्प्ले इंस्ट्रूमेंट्स आदि शामिल हैं।

ट्रांसड्यूसर

यह उच्च आवृत्ति बिजली ऊर्जा को मशीन कंपन ऊर्जा में स्थानांतरित कर सकता है।

बूस्टर

इसका उपयोग ट्रांसड्यूसर और हॉर्न को जोड़ने और ठीक करने के लिए किया जाता है। जब ट्रांसड्यूसर का आयाम बढ़ाया जाएगा, तो यह हॉर्न तक पहुंच जाएगा।

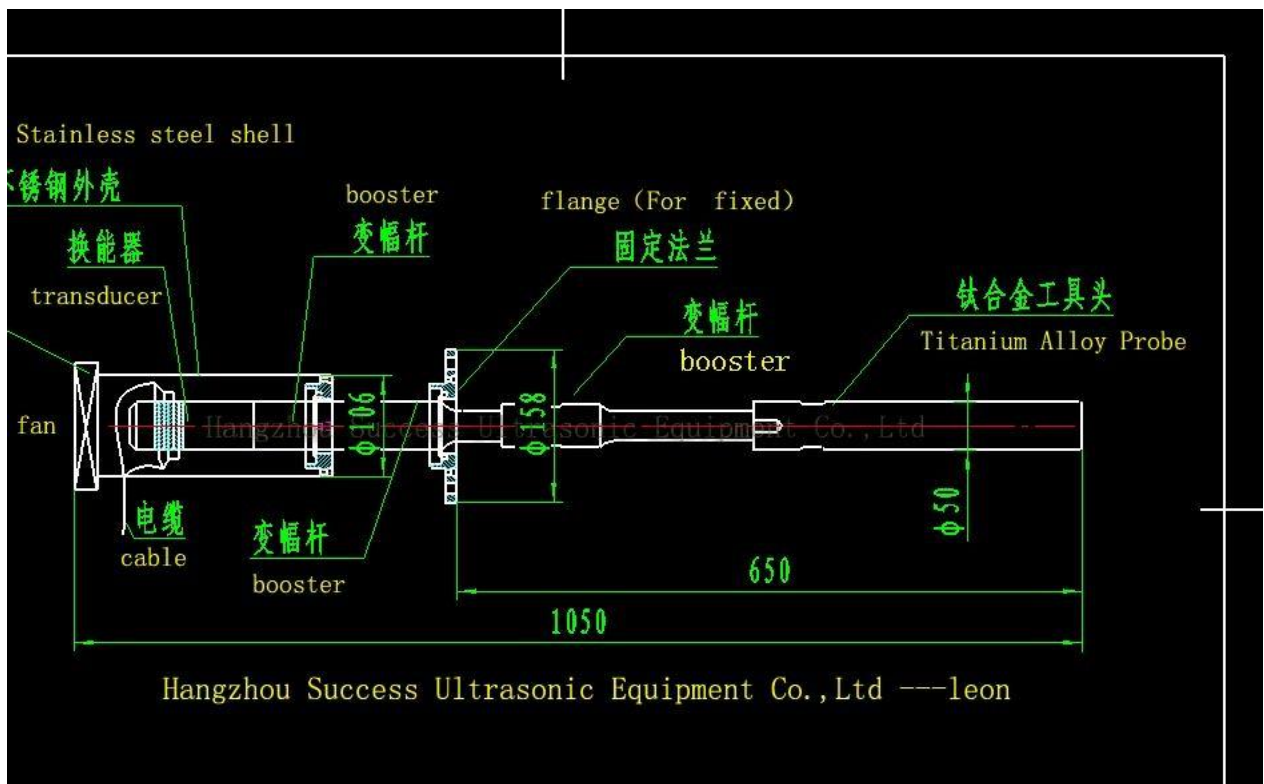
हॉर्न (सोनोट्रोड)

यह मशीन की ऊर्जा और दबाव को कार्यशील पदार्थ पर भेजता था।

पेंच

इसका उपयोग ट्रांसड्यूसर, बूस्टर और हॉर्न को जोड़ने के लिए किया जाता है।

धातु के पिघलने के बाद, आपको केवल जांच (सोनोट्रोड) को पिघली हुई धातु में डुबाना होगा, जो धातु को पिघलाने के लिए अल्ट्रासोनिक लॉन्च करेगा। फ़्लैंग का उपयोग स्थिरीकरण के लिए किया जाता है। चूँकि अल्ट्रासोनिक कंपन प्रणाली में कोई गतिमान भाग और बल घटक नहीं होते हैं, इसलिए निश्चित के लिए कोई विशेष आवश्यकता नहीं होती है। किसी विशेष कंटेनर में धातु को पिघलाएं, जैसे क्रूसिबल, पिघलने वाली भट्ठी, भट्ठी क्रिस्टलीकरण, फिर अल्ट्रासोनिक लॉन्च करने के लिए जांच को पिघली हुई धातु में डुबो दें। हम पिघली हुई धातु की मात्रा, अल्ट्रासोनिक जनरेटर शक्ति और अल्ट्रासोनिक की तीव्रता को नियंत्रित करने के लिए अल्ट्रासाउंड के समय को नियंत्रित कर सकते हैं। धातु सामग्री की विशेषताएं और आवश्यकताएं अलग-अलग होती हैं, प्रभाव और प्रभाव अल्ट्रासोनिक द्वारा भिन्न होता है। धातु की विशिष्ट प्रदर्शन आवश्यकताओं के लिए, केवल परीक्षण और त्रुटि के माध्यम से अल्ट्रासोनिक तीव्रता और वास्तविक परिणाम का सर्वोत्तम संतुलन ढूँढ़ें, ताकि सर्वोत्तम उपचार प्रक्रिया तैयार की जा सके।



चित्र: अल्ट्रासोनिक जांच 2.5 किलोवाट का चित्रण



चित्र: अल्ट्रासोनिक जांच की नियंत्रक इकाई

स्टिरर के साथ नियंत्रित वातावरण वर्टिकल मेल्टिंग फर्नेस:

विद्युत ताप प्रकार प्रोग्रामयोग्य ऊर्ध्वाधर पिघलने वाली भट्टी:

1. हीटिंग चेंबर का आकार: व्यास: 150 मिमी, लंबाई: 180 मिमी (लगभग)
2. भट्टी की ऊंचाई: 380 मिमी (लगभग)
3. अधिकतम तापमान: 1200°C
4. भट्टी को गर्म करने और ठंडा करने की दर तथा गर्म करने के समय को नियंत्रित करने के लिए प्रोग्राम करने योग्य होना चाहिए।
5. अधिकतम पर सेट करने की व्यवस्था. वांछित तापमान.
6. चेंबर को दोनों सिरों पर आसानी से अलग किए जा सकने वाले सिरेमिक वूल बोर्ड रिफ्रेक्टरी सपोर्ट/कवर के साथ बंद करने में सक्षम होना चाहिए।
7. शीर्ष वियोज्य समर्थन में ऊपर से समर्थित जांच (3 किग्रा, लगभग) को स्वीकार करने के लिए एक सह-अक्षीय 27 मिमी छेद होना चाहिए।
8. गियर मोटर का उपयोग करके लिफ्टिंग अटैचमेंट के माध्यम से अल्ट्रासोनिक प्रोब को ऊपर और नीचे करने का प्रावधान, डीसी ड्राइव का उपयोग करके एक बटन दबाकर या मैन्युअल रूप से हैंडव्हील को घुमाकर जांच को सुचारू रूप से ऊपर और नीचे करने में सक्षम बनाना। इसके अलावा जांच को स्थिति में लाने (भट्टी के

केंद्रीय अक्ष के साथ) और भट्टी से बाहर निकालने के बाद इसे भट्टी के एक तरफ रेडियल रूप से ले जाने का प्रावधान होना चाहिए (रेडियल ड्रिलिंग मशीन में ड्रिल चक की गति के समान)।

9. भट्टी के नीचे रखे ड्राई में नैनो कणों के मिश्रण के तुरंत बाद, नीचे एक मार्ग (प्लग के साथ पाइप) के माध्यम से पिघली हुई धातु को डालने की व्यवस्था।

10. धातु को अलग कूसिबल में पिघलाने और पिघलने के बाद कूसिबल को आसानी से बाहर निकालने का प्रावधान होना चाहिए, जब तली डालने की प्रणाली का उपयोग नहीं किया जाना हो।

11. कूसिबल में धातु के ऊपर आर्गन गैस की आपूर्ति के लिए 6 मिमी व्यास की स्टेनलेस स्टील ट्यूब डालने के लिए शीर्ष अलग करने योग्य दुर्दम्य समर्थन में एक तरफ 6 मिमी व्यास का छेद होना चाहिए।

12. 6.0 मिमी व्यास के गैस प्रवाह पाइप के साथ कूसिबल में धातु के ऊपर अक्रिय गैस (आर्गन) की निरंतर आपूर्ति का प्रावधान और ऐसी सामग्री से बना होना चाहिए जो 1200 °C तापमान का सामना कर सके। यह गैस प्रवाह सर्किट गैस प्रवाह मीटर से सुसज्जित होना चाहिए।

13. भट्टी में धातु की स्थिति देखने के लिए शीर्ष पर पर्याप्त आकार का एक नेत्र शीशा।

14. आर्गन आपूर्ति के लिए एक आर्गन सिलेंडर।

15. आर्गन सिलेंडर के लिए एक आर्गन रेगुलेटर।

16. आर्गन आपूर्ति के लिए नली पाइप।

17. भट्टी के बाहरी आवरण का तापमान न्यूनतम (60 से अधिक नहीं) होना चाहिए (सी)।

18. भट्टी के निचले सिरे पर रबर गैसकेट को पानी से ठंडा करने की व्यवस्था।

19. 1 साल की गारंटी.

यांत्रिक स्टिरर:

1. गति: 400 आरपीएम (परिवर्तनीय, परिवर्तनीय गति डीसी मोटर का उपयोग करके)
2. लंबाई: 300 मिमी (भट्टी में रखे कूसिबल की गहराई के अनुसार लगभग)
3. भट्टी में रखे कूसिबल में स्टिरर को ऊपर-नीचे करने की व्यवस्था तथा स्टिरर को भट्टी से बाहर निकालने के बाद स्थिति में (भट्टी के केंद्रीय अक्ष के साथ) घुमाने तथा भट्टी के एक तरफ रेडियल घुमाने की व्यवस्था (जैसा कि) रेडियल ड्रिलिंग मशीन में ड्रिल चक की गति)।

अल्ट्रासोनिक जांच के लिए शीतलन व्यवस्था:

गोलाकार ट्यूब के माध्यम से ठंडा पानी प्रसारित करके अल्ट्रासोनिक जांच (व्यास: 25.4 मिमी, लंबाई: 254 मिमी) के पानी को ठंडा करने की व्यवस्था। ठंडे पानी की आपूर्ति के लिए चिलर प्रदाता होना चाहिए।

5. कम्प्यूटरीकृत यूनिवर्सल परीक्षण मशीन (यूटीएम)

विशेष विवरण

क्षमता: 1000 के. एन. (100 टन)

कंपनी: फाइन उपकरण

स्ट्रोक: 1200 मिमी (लगभग)

तनाव की गति: 50-120 मिमी/मिनट (लगभग)

पिस्टन स्ट्रोक: 200-300 मिमी

संकल्प पिस्टन आंदोलन: 0.1 - 0.2 मिमी

बल माप के लिए न्यूनतम गणना: अधिकतम 0.3% भार

यूटीएम को विभिन्न रेंजों जैसे 0-100 केएन, 0-250 केएन, 0-500 केएन 0-1000 केएन आदि में संचालित करने का प्रावधान)

कंट्रोल पैनल (एनालॉग + कम्प्यूटरीकृत) द्वारा कंप्यूटर के साथ और कंप्यूटर के बिना भी काम करने का प्रावधान।

सिस्टम को एसटीएम और आईएसओ मानकों को पूरा करना होगा।

सभी मानक सहायक उपकरण इस प्रकार हैं:

क्र.सं.	सामान	आयाम (लगभग)
1.	संपीड़न प्लेट व्यास के लिए जोड़ी। (मिमी)	160
2	गोल नमूना व्यास के लिए. (मिमी)	10 – 30
3	गोल नमूना व्यास के लिए. (मिमी)	30 - 50
4	गोल नमूना व्यास के लिए. (मिमी)	50 – 70
5	प्लैट नमूना मोटाई (मिमी) के लिए	0 – 20
6	प्लैट नमूना मोटाई (मिमी) के लिए	20 - 44
7	प्लैट नमूना मोटाई (मिमी) के लिए	44 – 65
8	अधिकतम. प्लैट नमूने की चौड़ाई (मिमी)	70
9	चौड़ाई का समायोज्य रोलर समर्थन (मिमी)	160
		50

	व्यास (मिमी) मैक्स के साथ. समायोज्य निकासी (मिमी)	800
10	त्रिज्या के पंच शीर्ष (मिमी)	16
11	त्रिज्या के पंच शीर्ष (मिमी)	20

अतिरिक्त सहायक उपकरण:

1	इलेक्ट्रॉनिक एक्सटेन्सोमीटर
2	10 मिमी व्यास तक तार रस्सी के लिए तार रस्सी तन्यता परीक्षण लगाव।
3	कंधे वाले और पिरोए गए नमूने के लिए तन्यता परीक्षण लगाव, नंगे लगाव
4	व्यास के लिए स्प्रिंग्स के साथ 8 विभाजित झाड़ियों का 1 सेट। 6, 8, 10, 12, 14, 16, 18 और 20 मिमी कंधे वाला नमूना (आकार मिमी में)
5	एम 6, एम 8, एम 12, एम 14, एम 18 और एम 20 थ्रेडेड नमूने के लिए 6 थ्रेडेड झाड़ियों का 1 सेट
6	50 मिमी चौड़ाई और 3 मिमी मोटाई तक फ्लैट बेल्ट की तन्यता परीक्षण मशीन के लिए अनुलग्नक।
7	व्यास के कतरनी परीक्षण के लिए झाड़ियों के 5 सेट के साथ छोटा डबल कतरनी परीक्षण। 5, 8, 12, 16 एवं 20 मिमी.
8	व्यास के कतरनी परीक्षण के लिए झाड़ियों के 5 सेट के साथ बड़ा डबल कतरनी परीक्षण। 25, 30, 35 और 40 मिमी.
9	फाउंडेशन बोल्ट का सेट (8 का 1 सेट)
10	180 डिग्री बेंड टेस्ट अटैचमेंट
11	पेसर लोड करें
12	140, 175, 240, 280 और 320 मिमी तक बार के लिए बीएस 4449 - 1997 के अनुसार बेंड री-बेंड टेस्ट अटैचमेंट
13	कंधे और थ्रेडेड नमूने के लिए तन्यता परीक्षण अनुलग्नक
14	एच 8 स्प्लिट झाड़ियाँ दीया के झरने के साथ। 6, 8, 10, 12, 14, 16, 18, 20 मिमी
15	M6, M8, M12, M14, M18, M20 के लिए 6 थ्रेडेड झाड़ियों का एक सेट।

6. डिजिटल रॉकवेल सह ब्रिनेल कठोरता परीक्षण मशीन

विशेष विवरण:

परीक्षण भार (किलोग्राम): 60, 100, 150 (रॉकवेल), 187.5, 250 (ब्रिनेल)

प्रारंभिक भार (किलोग्राम): 10

अधिकतम. परीक्षण ऊंचाई (मिमी): 300 (लगभग)

मेक: फाइन इंस्ट्रूमेंट्स कंपनी लिमिटेड, पुणे

सामान:

1. रॉकवेल डायमंड इंडेंटर: 2 नं.
2. लगभग 45 मिमी तक के व्यास के लिए गोल जॉब के लिए 'वी' ग्रूव के साथ परीक्षण तालिका: 1 नं.
3. कठोर स्टील बॉल इंडेंटर Φ 1/16" : न्यूनतम 5 संख्या।
4. कठोर स्टील बॉल इंडेंटर Φ 2.5 मिमी: न्यूनतम। 5 नं.
5. कठोर स्टील बॉल इंडेंटर Φ 5 मिमी: न्यूनतम। 5 नं.
6. कठोर स्टील बॉल इंडेंटर Φ 10 मिमी: न्यूनतम। 5 नं.
7. टेस्ट ब्लॉक एचआरसी: 1 नं.
8. टेस्ट ब्लॉक एचआरबी: 1 नं.
9. टेस्ट ब्लॉक एचबी 2.5 मिमी / 187.5 किलोग्राम: 1 नं.
10. टेस्ट ब्लॉक 250 किलोग्राम: 1 नं.
11. एलन स्पैनर: 3 नं.
12. क्लैम्पिंग डिवाइस: 1 नं.
13. ब्रिनेल माइक्रोस्कोप : 1 नं.
14. रबर बेलो: 1 नं.
15. अनुदेश पुस्तिका

कठोरता परीक्षक में निम्नलिखित विशेषताएं हैं:

- बड़ा चित्रमय प्रदर्शन
- रॉकवेल कठोरता परीक्षण (जैसे ए, बी, सी, डी, ई, एफ, जी, एचके आदि) के विभिन्न पैमानों पर कठोरता मापने का प्रावधान।
- माइनर लोड को ग्राफिक रूप से एलसीडी स्क्रीन पर प्रदर्शित किया जाना चाहिए
- 1 से 99 सेकंड तक के बड़े लोड और छोटे लोड के लिए दोहरी समय सेटिंग।

- परीक्षण परिणाम समय टिकट के लिए बैटरी बैकअप रियल टाइम क्लॉक
- परिणाम बड़े फ्रॉन्ट आकार में प्रदर्शित होने चाहिए
- औसत विकल्प के लिए, पिछले 9 परिणाम तक स्क्रीन पर प्रदर्शित होने चाहिए। परीक्षण पूरा होने के बाद औसत और मानक विचलन मान प्रदर्शित किए जाते हैं।
- स्टोरेज में स्वचालित परिणाम डेटा प्रविष्टि के साथ सिस्टम में लगभग 100 या अधिक परिणामों को संग्रहीत करने का प्रावधान।
- सिस्टम में परिणाम भंडारण को सक्षम या अक्षम करने का प्रावधान। बैचवार रिपोर्ट और सांख्यिकीय रिपोर्ट स्क्रीन पर प्रदर्शित की जानी चाहिए और प्रिंटर पर मुद्रित की जानी चाहिए
- एक समय में उपयोगकर्ता को ऊपर/नीचे कुंजी के साथ स्कॉल करके एलसीडी डिस्प्ले पर लगभग 7 रीडिंग देखने में सक्षम होना चाहिए
- स्वचालित रूप से चयनित वजन का पता लगाना
- उपयोगकर्ता को चयनित वजन के अनुसार उचित पैमाने का चयन करने में सक्षम होना चाहिए
- मोटर चालित स्वचालित लोडिंग या मैन्युअल लोडिंग का चयन किया जाना चाहिए
- रॉकवेल, रॉकवेल सतही और ब्रिनेल कठोरता परीक्षण संभव होना चाहिए
- रॉकवेल से ब्रिनेल (एचबी), विकर्स (एचवी), नूप (एचके), टेन्साइल स्ट्रेंथ (केएसआई), माइक्रोफिशियल (डब्ल्यूएमएन) में परिणामों का रूपांतरण स्वचालित होना चाहिए
- प्रिंटर इंटरफ़ेस के लिए यूएसबी पोर्ट/सीरियल पोर्ट
- प्रिंट परिणाम में अधिमानतः बैच नंबर, सीरियल नंबर, दिनांक, समय, कठोरता मूल्य और कठोरता स्केल शामिल होना चाहिए
- RS232C पोर्ट प्रोग्रामयोग्य होना चाहिए.
- कठोरता परीक्षक पर किए गए परीक्षण आईएस 1586 - 2000, बीएसईएन - आईएसओ - 6508 - 2, रॉकवेल परीक्षण के लिए एसटीएम - ई - 18 और ब्रिनेल परीक्षण के लिए आईएस 2281 - 1983, बीएस 240, एसटीएम - ई - 10 और आईएस 1586 - 2000 के अनुरूप होने चाहिए। , रॉकवेल सतही परीक्षण के लिए एसटीएम - ई - 18
- गहराई माप .001 मिमी तक उच्च सटीकता डायल गेज के साथ होना चाहिए
- स्वचालित वजन पहचान का प्रावधान
- स्वचालित पैमाने चयन का प्रावधान
- त्रुटि संकेत के लिए प्रावधान
- विंडोज़-8 तक संगत ग्राफ़िकल उपयोगकर्ता विंडोज़ सॉफ़्टवेयर प्रदान किया गया

- सॉफ्टवेयर के माध्यम से कठोरता डेटा के सांख्यिकीय विश्लेषण का प्रावधान
- ग्राफिकल यूजर इंटरफेस (जीयूआई) के माध्यम से डेटा भंडारण का प्रावधान
- स्वचालित और मैनुअल दोनों लोडिंग का प्रावधान
- एनएबीएल परीक्षण ब्लॉक प्रदान किया जाना चाहिए
- एक साल की वारंटी

7. मैनुअल कठोरता परीक्षण मशीन

विशेष विवरण:

मॉडल: एसई ट्विन

कंपनी: समर्थ

रॉकवेल कठोरता परीक्षक (संयुक्त मॉडल)

रॉकवेल और सतही कठोरता माप के लिए उपयुक्त।

परीक्षण भार (Kgf): 60,100, 150 (रॉकवेल); 15, 30, 45 (रॉकवेल सतही)

प्रारंभिक भार (किग्रा): 3 (रॉकवेल सतही) और 10 (रॉकवेल)

अधिकतम. परीक्षण की ऊँचाई (मिमी): 295, गले की गहराई (मिमी) 150

अधिकतम. आधार के नीचे एलिवेटिंग स्कू की गहराई: 310 मिमी

8. मफल भट्टी

विशेष विवरण:

कंपनी: मेट्रक्स आईएसओ 9001: 2008 प्रमाणित /सीई मार्क।

नमूना; एमएफ - 12 एम

सिरेमिक फाइबर ऊन इन्सुलेशन के साथ हल्के वजन (ईट इन्सुलेशन के बजाय)। बाहरी आवरण दोहरी दीवार वाली मोटी पीसीआरसी शीट से बना है, जो पाउडर कोटिंग के साथ तैयार किया गया है।

कंथल "ए-1" तार से बना तत्व।

दरवाजे पर देखने वाली टोपी उपलब्ध कराई गई है।

सॉफ्ट टच कीपैड और प्रक्रिया और सेट तापमान के दोहरे प्रदर्शन के साथ माइक्रोप्रोसेसर आधारित पीआईडी नियंत्रक प्रदान किया गया।

220/230 वोल्ट एसी सिंगल फेज आपूर्ति।

अधिकतम. तापमान 1200°C तक

कार्य तापमान रेंज 1150 डिग्री सेल्सियस तक

अस्थायी. सटीकता: 2 डिग्री सेल्सियस

मफल आकार: 300 X 300 X 300 मिमी

9. इलेक्ट्रॉनिक तुला

विशेष विवरण:

कंपनी: वेन्सर प्रा. लिमिटेड

क्षमता: 30 किग्रा

सटीकता: ± 0.01 ग्राम

10. अपघर्षक काटने की मशीन

विशेष विवरण:

कंपनी: वैशेषिक

गति: 1500 आरपीएम

ग्राइंडिंग व्हील का आकार: 300 X 2 X 32 मिमी

अधिकतम. नमूना आकार: 30 X 30 मिमी

आयाम: 24" X 26" X 30"

11. संपीडन मोल्डिंग मशीन

विशेष विवरण:

250 किलोग्राम/सेमी² तक की यांत्रिक सुई गेज रेंज के साथ दबाव विनियमन प्रणाली के साथ हाइड्रोलिक पावर पैक सिस्टम के माध्यम से संचालन

मशीन का आकार: 4 फीट X 4 फीट X 4 फीट (LXBXH)

क्षमता: 0-30 टन

प्लेटन का आकार: 350 मिमी X 350 मिमी

वर्किंग प्लेटन का आकार: 300 मिमी X 300 मिमी

नमूना मोटाई सहनशीलता 0.05 मिमी

स्ट्रोक: डे लाइट ओपनिंग 200 मिमी

तापमान नियंत्रक: दोनों प्लेटों के तापमान और शीतलन नियंत्रण के लिए टच स्क्रीन एचएमआई होनी चाहिए।

तापमान नियंत्रक: अधिकतम. कार्य सीमा परिवेश 280.0°C और कूलिंग टाइमर अधिकतम। रेंज 999 मिनट.

शीतलन: टच स्क्रीन एचएमआई नियंत्रक द्वारा नियंत्रित सोलनॉइड वाल्व के माध्यम से।

सभी इलेक्ट्रिकल और इलेक्ट्रॉनिक घटक प्रतिष्ठित कंपनी से प्राप्त किए गए हैं।

कनेक्टिंग पावर सप्लाई: 5 किलोवाट

एचएमआई टच स्क्रीन डिस्प्ले पर डिजिटल दबाव संकेत प्रदान किया जाना चाहिए

कंपन से बचने के लिए अध्ययन और भारी कोण संरचना।

जलने की चोट से बचने के लिए सुरक्षा कवर।

फर्श मॉडल

12. स्वचालित एक्सट्रूजन लाइन

विशेष विवरण

विभिन्न आकारों के टेप, ट्यूब, रॉड और तार के लिए स्वचालित एक्सट्रूजन लाइन

कंपनी : साई एक्स्ट्रुमेक

प्रसंस्करण सामग्री : पीई जैसी अधिकांश थर्मोप्लास्टिक सामग्रियों के लिए उपयुक्त होना चाहिए, पीवीसी, पीयू आदि।

निर्माण का प्रकार : क्षैतिज

केंद्र की ऊंचाई : लगभग 1000 मिमी

अधिकतम. तापमान : 300 डिग्री सेल्सियस या अधिक

इसके घटकों का विस्तृत विवरण:

ए ए/सी मोटर के साथ 25 मिमी एक्सट्रूडर

- एल/डी अनुपात लगभग 25:1
- पानी के माध्यम से अनुभूति क्षेत्र में शीतलता
- वायु के माध्यम से बैरल क्षेत्र में शीतलन
- लगभग 2 किलोवाट के सिरेमिक हीटर के माध्यम से बैरल को गर्म करना शक्ति

भाग:

बैरल: उच्च कठोरता के साथ घिसाव और संक्षारण प्रतिरोधी सामग्री से बना होना चाहिए।
50 बार से अधिक दबाव सहन करने में सक्षम होना चाहिए और मिरर फ़िनिश होना चाहिए।

पेंच: उच्च गुणवत्ता के साथ घिसाव और संक्षारण प्रतिरोधी सामग्री से बना होना चाहिए दर्पण पॉलिश के साथ कठोरता।

स्कू का डिज़ाइन ऐसा होना चाहिए जिससे अधिकतम आउटपुट मिले सजातीय मिश्रण के साथ

आउटपुट: लगभग 10 किलोग्राम/घंटा (समायोज्य)

रफ़्तार: लगभग 100 आरपीएम (परिवर्तनीय)

संक्षिप्तीकरण अनुपात: लगभग 3:1

गियरबॉक्स: पेचदार प्रकार के गियर;

थ्रस्ट हाउसिंग को अधिकतम बल सहन करने के लिए डिज़ाइन किया जाना चाहिए और इसकी अनुमति नहीं देनी चाहिए रोलर बेयरिंग का उपयोग करके गियर को नुकसान पहुंचाने के लिए जोर का दबाव।

जोर राशन: लगभग 1:15 बजे

हूपर: संक्षारण प्रतिरोधी सामग्री से बना होना चाहिए, अधिमानतः स्टेनलेस स्टील शीट से
लगभग 8 किलोग्राम क्षमता के लिए आई मार्क प्रणाली प्रदान की जानी चाहिए

दबाना: घिसाव और संक्षारण प्रतिरोधी सामग्री से बना होना चाहिए

बी तीन रोलर कैलेंडरिंग इकाई

- टेप खींचने के लिए उपयुक्त (2 मिमी X 20 मिमी)
- 03 नग रोलर (Ø50 मिमी X 150 मिमी)
- रोलर के लिए कूलिंग पंखा उपलब्ध कराया जाना चाहिए
- लगभग 0.25 एचपी की ए/सी गियर वाली मोटर

सी कैलेंडरिंग इकाई के लिए पैनल

- तीन रोलर कैलेंडरिंग यूनिट के 0.25 एचपी गियर मोटर के लिए उपयुक्त।

डी एक्सट्रूजन लाइन के लिए नियंत्रण कक्ष (पीएलसी)

- 4 पीआईडी तापमान नियंत्रक
- 3 एचपी एसी थाइरिस्टर ड्राइव मोटर
- ओवरलोड, ओवर वोल्टेज, अंडर वोल्टेज, सिंगल फेजिंग और फ्रील्ड विफलता से सुरक्षा।

इ टेप हेड:

- विभिन्न आकारों (छोटे, मध्यम और बड़े) के टेप डाई सेट।
- अधिकतम 5 मिमी X 30 मिमी तक विभिन्न चौड़ाई और मोटाई के टेप बनाने के लिए उपयुक्त।
- घिसाव और संक्षारण प्रतिरोधी सामग्री से बना होना चाहिए
- एकल बाईपास प्रणाली

एफ ट्यूब हेड

- विभिन्न आकारों (छोटे, मध्यम और बड़े) के ट्यूब डाई सेट।
- विभिन्न आकार की ट्यूबों के लिए उपयुक्त
- घिसाव और संक्षारण प्रतिरोधी सामग्री से बना होना चाहिए

जी तार का सिर

- विभिन्न आकारों (छोटे, मध्यम और बड़े) के वायर डाई सेट।
- गोलाकार और चौकोर क्रॉस सेक्शन के विभिन्न आकार के तारों के लिए उपयुक्त
- घिसाव और संक्षारण प्रतिरोधी सामग्री से बना होना चाहिए
- भोजन के प्रयोजन के लिए सतह पर सेरेशन (दांत) बनाने का प्रावधान

एच रॉड हेड

- विभिन्न आकारों (छोटे, मध्यम और बड़े) के आयताकार रॉड के लिए डाई सेट।
- वर्गाकार या आयताकार क्रॉस सेक्शन की विभिन्न आकार की छड़ों के लिए उपयुक्त
- घिसाव और संक्षारण प्रतिरोधी सामग्री से बना होना चाहिए
- भोजन के प्रयोजन के लिए सतह पर सेरेशन (दांत) बनाने का प्रावधान

मैं कंपाउंडिंग यूनिट

1. पैलेटाइजिंग (सिर) मरो: 1 नं.
2. पैलेटाइजिंग कटर: 1 नं.
3. पानी की टंकी: 1 नं.

न्यूनतम एक वर्ष की वारंटी प्रदान की जानी चाहिए।

13. डिजिटल टैकोमीटर

विशेष विवरण

माप श्रेणी:

फोटो टैकोमीटर: 5 से 99,999 आरपीएम
 संपर्क टैकोमीटर: 0.5 से 19,999 आरपीएम
 सतही गति: 0.05 से 1,999.9 मीटर/मिनट।
 0.2 से 6,560 फीट/मिनट।

संकल्प:

आरपीएम: 0.1 आरपीएम (<1,000 आरपीएम)
 1 आरपीएम (1,000 आरपीएम)
 मी/मिनट: 0.01 मी/मिनट (<100 मी/मिनट)
 0.1 मी/मिनट (100 मी/मिनट)
 फीट/मिनट: 0.1 फीट/मिनट (<1,000 फीट/मिनट)
 1 फीट/मिनट (1,000 फीट/मिनट)
 डिस्प्ले: 5 अंक, 10 मिमी (0.4") एलसीडी
 सटीकता: $\pm (0.05\% + 1 \text{ अंक})$
 समय का आधार: क्वार्ट्ज क्रिस्टल
 सर्किट: माइक्रो कंप्यूटर एलएसआई सर्किट का विशेष एक-चिप
 ऑपरेटिंग तापमान: 0 - 50°C (32 - 122°F)
 परिचालन आर्द्रता: 80% आरएच से कम
 मेमोरी: अंतिम/अधिकतम/न्यूनतम मान
 बैटरी: 4 X 1.5V AA (UM-3) बैटरी

बिजली की खपत:

फोटो प्रकार: लगभग. डीसी 153 एमए
 संपर्क प्रकार: लगभग. डीसी 10 एमए
 आकार: 195 X 61 X 38.5 मिमी (7.6 X 2.4 X 1.5 इंच)
 वजन: बैटरी सहित 280 ग्राम (0.61 पाउंड)*

सहायक उपकरण शामिल:

कैरी केस - 1 पीसी
 प्रतिबिंबित टेप के निशान - 1 पीसी
 आरपीएम एडॉप्टर (सीओएनई) - 1 पीसी
 आरपीएम एडॉप्टर (फ़नल) - 1 पीसी
 भूतल गति परीक्षण पहिया - 1 पीसी
 ऑपरेशन मैनुअल - 1 पीसी

14. हॉट प्लेट के साथ डिजिटल मैग्नेटिक स्टिरर

मॉडल: MS-H280-प्रो, मेक: SCIOLOGEX,

एलईडी डिजिटल हॉट प्लेट चुंबकीय स्टिरर

एलईडी डिजिटल हॉट प्लेट चुंबकीय स्टिरर, सिरेमिक/स्टेनलेस स्टील हॉट प्लेट,

गति सीमा: 100-1500 आरपीएम,

अस्थायी. 280°C तक,

गति और तापमान का एलईडी डिस्प्ले,

बाहरी पीटी 1000 सेंसर,

हिलाने की क्षमता: 3 लीटर

15. बेल्ट ग्राइंडर

मॉडल: SLE BG-21, मेक: स्पेक्ट्रो, एंडलेस बेल्ट: 100X 915 मिमी

मोटर 1HP, 220 V, पानी के नल के साथ संलग्न मॉडल

16. माइक्रोवेव ओवन

कंपनी: आईएफबी

क्षमता: 30 लीटर

17. इन्फ्रा-रेड गन

उच्च तापमान के लिए इन्फ्रा रेड गनइरेचर रेंज: 0 से 650 0C

लेजर बीम पॉइंटर के साथ

18. अल्ट्रासोनिक क्लीनर/स्नान

आवृत्ति: 40 KHz

पावर: 300 डब्ल्यू

तापमान: 0-80 डिग्री सेल्सियस

टाइमर: 1-99 मिनट

19. ड्रिल के सेट के साथ हैंड ड्रिल

कंपनी: बोश,

पावर: 550 डब्ल्यू

ड्रिल आकार सीमा

20. वैक्यूम ओवन

कंपनी: **लैब तकनीक** प्रतिरूप संख्या। तेल पंप के साथ VOR-654 (बीवीएस एंटरप्राइज, भोपाल द्वारा आपूर्ति)

तकनीकी निर्देश:

वैक्यूम ओवन (गोल गुहा)

बॉडी का प्रकार: अंदर और बाहर की बॉडी स्टेनलेस स्टील से बनी है, बाहरी बॉडी पाउडर से लेपित है। ऑपरेशन के दौरान ओवन की बाहरी बॉडी गर्म नहीं होनी चाहिए और ठीक से इंसुलेटेड होनी चाहिए।

गुहा के चारों ओर समान तापन के लिए परिधि के चारों ओर अच्छी गुणवत्ता वाले हीटर।

पीआईडी तापमान. माइक्रोप्रोसेसर आधारित पीआईडी डिजिटल तापमान और समय नियंत्रक द्वारा नियंत्रित। समय और तापमान माप की आसान सेटिंग।

दोहरे डिस्प्ले के साथ तापमान डिस्प्ले सूचक सह नियंत्रक।

तापमान सटीकता: $\pm 1^{\circ}\text{C}$

तापमान माप के लिए K प्रकार थर्मोकपल

ओवन को बिना किसी वैक्यूम हानि के उच्च तापमान पर लगातार काम करने में सक्षम होना चाहिए। डायल गेज पर -760 मिमी एचजी, (10-2 एमबार) तक वैक्यूम प्राप्य।

वैक्यूम संकेतक: सेंसर के साथ डिजिटल प्रकार का वैक्यूम संकेतक और एक बॉर्डन प्रकार डायल गेज 0-30" एचजी के साथ भी जुड़ा हुआ है।

वैक्यूम पंप: डबल स्टेज रोटरी प्रकार का तेल भरा वैक्यूम पंप

क्षमता 75 लीटर/मिनट; मोटर 0.5 एचपी.

नियमित वाल्व और गेज के साथ पूरा करें। जोड़ रहित गैस्केट और टाइट नॉब सील द्वारा सीलिंग। तापमान सीमा: $\text{RT}+5^{\circ}\text{C}$ से 300°C ; शूट के ऊपर या शूट के नीचे कोई तापमान नहीं होना चाहिए।

आंतरिक आकार: 3000X375 मिमी (व्यास x गहराई), दो एसएस ट्रे

एकल चरण बिजली आपूर्ति 220V, 50Hz

वारंटी: किसी भी विनिर्माण दोष के खिलाफ एक वर्ष की व्यापक वारंटी।

अतिरिक्त: वैक्यूम पंप तेल।

21. एग्लोमेरेटर

पॉलिमर और नैनो पार्टिकल के लिए मिक्सिंग सेट-अप में एग्लोमेरेटर, ग्राइंडिंग मैकेनिज्म आदि शामिल हैं।

कंपनी: DEW (टेक्निकल सिस्टम्स, भोपाल द्वारा आपूर्ति)

विशेष विवरण:

ड्रम का आकार: 20" X 24" ; सामग्री: एमएस और एसएस निर्मित।

ब्लेड: डार्क स्टील, ब्लेड की संख्या: 06,

मोटर: 20 एचपी क्रॉम्पटन

22. हीट डिफ्लेक्टर तापमान परीक्षक (एचडीटी)

कंपनी: ग्लोब एंटरप्राइजेज

परीक्षण उपकरण के आयाम: 3 से 4.2 मिमी चौड़ा 9.8 से 15 मिमी मोटा 120 मिमी लंबा

सहायक किनारे के बीच की दूरी 100 ± 2 मिमी

सहायक और लोडिंग किनारे की त्रिज्या: 3.0 ± 0.2 मिमी

तापमान 3000C तक, न्यूनतम गणना 1°C)

विक्षेपण माप: 0-10 मिमी (न्यूनतम गणना 0.01 मिमी)

नमूने पर बल: 25 ग्राम के चरणों में 200 से 1800 ग्राम

नैनो कम्पोजिट प्रयोगशाला में काम करने के लिए सुरक्षा सावधानियाँ

नैनो कम्पोजिट लैब में काम करते समय निम्नलिखित सुरक्षा सावधानियों का पालन किया जाना चाहिए।

1. नैनो कम्पोजिट लैब में काम करते समय सभी छात्रों को जूते पहनने चाहिए।
2. नैनो कम्पोजिट लैब में काम करते समय एप्रन और दस्ताने अवश्य पहनने चाहिए।
3. छात्रों को प्रयोगशाला में बेकार नहीं बैठना चाहिए या खड़े नहीं रहना चाहिए।
4. जिन छात्रों को किसी विशेष मशीन या उपकरण के संचालन के संबंध में उचित ज्ञान नहीं है, उन्हें उपकरण या मशीन का संचालन नहीं करना चाहिए।
5. किसी भी मशीन पर काम करने से पहले छात्र को उचित संचालन निर्देश और सुरक्षा सावधानियां सीखनी चाहिए।
6. प्रयोगशाला बंद करते समय छात्र को कोई भी उपकरण या भट्टी चालू हालत में नहीं छोड़नी चाहिए।

प्रयोगों की सूची

1. एएल-एसआईसी नैनो कंपोजिट के गुणों पर सिंटरिंग तापमान के प्रभाव का अध्ययन करना।
2. एएल-एसआईसी कम्पोजिट पाउडर की सूक्ष्म संरचना पर बॉल मिलिंग समय और बॉल से पाउडर वजन अनुपात के प्रभाव का अध्ययन करना।
3. ताँबा मैट्रिक्स कम्पोजिट पाउडर के माइक्रोस्ट्रक्चर पर बॉल मिलिंग मापदंडों के प्रभाव का अध्ययन करना।
4. उच्च कार्बन स्टील नमूनों के गुणों पर ताप उपचार तापमान और समय के प्रभाव का अध्ययन करना।
5. अल-मैट्रिक्स कंपोजिट नमूनों के गुणों पर हीट ट्रीटमेंट तापमान और समय के प्रभाव का अध्ययन करना।
6. विभिन्न सामग्रियों के नमूनों की रॉकवेल कठोरता (बी-स्केल) निर्धारित करने के लिए।
7. विभिन्न सामग्रियों के नमूनों की रॉकवेल कठोरता (सी-स्केल) निर्धारित करने के लिए।
8. एएल-एसआईसी समग्र नमूनों की ब्रिनेल कठोरता निर्धारित करने के लिए।
9. उच्च कार्बन स्टील के नमूनों के पतले नमूनों की सतही कठोरता का निर्धारण करना।
10. विभिन्न नमूनों के गुणों पर शॉट पीनिंग के प्रभाव का अध्ययन करना।
11. माइल्ड स्टील नमूने पर तन्यता परीक्षण करना।
12. एएल -मैट्रिक्स कम्पोजिट नमूने पर तन्यता परीक्षण करने के लिए
13. एक्सटेन्सोमीटर का उपयोग करके एल्यूमीनियम मिश्र धातु नमूने पर तन्यता परीक्षण करना।
14. उच्च कार्बन स्टील नमूने पर संपीड़न परीक्षण करने के लिए।
15. अल-मैट्रिक्स कम्पोजिट नमूने पर संपीड़न परीक्षण करने के लिए।
16. माइल्ड स्टील नमूने पर कतरनी परीक्षण करना।
17. माइल्ड स्टील फ्लैट नमूने पर बेंड टेस्ट करने के लिए।
18. कमरे या उच्च तापमान पर धातु और मिश्रित पाउडर का संघनन।

अनुसंधान एवं परामर्श परियोजनाओं की सूची

क्र.सं.	परियोजना का शीर्षक	अन्वेषक का नाम	मात्रा	प्रायोजक एजेंसी	अवधि	स्थिति
1.	765 केवी ट्रांसमिशन लाइन के टॉवर भागों के यांत्रिक गुणों का मूल्यांकन	डॉ. राजेश पुरोहित डॉ. आरएस राणा डॉ. रमन नटेरिया	40,450/-	पावर ग्रिड कॉर्पोरेशन इंडिया लिमिटेड शुजालपुर	01 अक्टूबर से 31 दिसंबर 2014	पुरा होना।
2.	400 केवी डीसी ट्रांसमिशन लाइन के प्रयुक्त और ताजा मूस कंडक्टरों का यांत्रिक संपत्ति विश्लेषण।	डॉ. राजेश पुरोहित एवं डॉ. रमन नटेरिया	11,450/-	पावर ग्रिड कॉर्पोरेशन इंडिया लिमिटेड, जबलपुर	25 फरवरी- 05 अप्रैल 16	पुरा होना।
3.	अल्ट्रासोनिक सहायता प्राप्त स्टिर कास्टिंग द्वारा अल-नैनो TiC कंपोजिट का निर्माण	डॉ. राजेश पुरोहित एवं डॉ. आरएस राणा	12190/-	एसएटीआई, विदिशा	23 जून - 31 जुलाई 2016	पुरा होना।
4.	अल-बी4सी कंपोजिट का निर्माण और मिलिंग मापदंडों का अनुकूलन	डॉ. आरएस राणा एवं डॉ. राजेश पुरोहित	14950/-	एसएटीआई, विदिशा	23 जून - 31 जुलाई 2016	पुरा होना।
5.	हड्डी/संयुक्त कृत्रिम अनुप्रयोगों के लिए विटामिन के साथ प्रबलित पॉलिमर का संश्लेषण और लक्षण वर्णन	डॉ. राजेश पुरोहित एवं डॉ. राहुल श्रीवास्तव	11500/-	एपीएस यूनिवर्सिटी, रीवा	7 जून 2017- 6 जुलाई 2017	पुरा होना।
6.	मेटल मैट्रिक्स नैनो कंपोजिट: संश्लेषण और लक्षण वर्णन	डॉ. राजेश पुरोहित एवं	11500/-	बीयूआईटी, भोपाल	26 जून 2017 25 जून 2018	पुरा होना।

		डॉ. आरएस राणा				
7.	नैनो झरझरा सामग्री का निर्माण और लक्षण वर्णन	डॉ. राजेश पुरोहित एवं डॉ. आरएस राणा	11500/-	बीयूआईटी, भोपाल	26 जून 2017 25 जून 2018	पुरा होना।
8.	पॉलिमर मैट्रिक्स नैनो कंपोजिट का विकास और विश्लेषण	डॉ. आरएस राणा एवं डॉ. राजेश पुरोहित	11500/-	बीयूआईटी, भोपाल	26 जून 2017 25 जून 2018	पुरा होना।
9.	नैनो कण प्रबलित पॉलिमर मैट्रिक्स कंपोजिट का निर्माण और लक्षण वर्णन	डॉ. आरएस राणा एवं डॉ. राजेश पुरोहित	11500/-	बीयूआईटी, भोपाल	26 जून 2017 25 जून 2018	पुरा होना।
10.	पॉलिमर मैट्रिक्स नैनो SiO ₂ कंपोजिट का डिजाइन और निर्माण	डॉ. राजेश पुरोहित, डॉ. आरएस राणा	11800/-	सिस्टेक, भोपाल	1 जून से 30 जून 2019	पुरा होना।
11.	मेटल मैट्रिक्स कंपोजिट का निर्माण और परीक्षण	डॉ. राजेश पुरोहित डॉ. आरएस राणा डॉ. रमन नटेरिया	29500/-	एसएसबीटी कॉलेज ऑफ इंजीनियरिंग एंड टेक्नोलॉजी, जलगांव (एमएच)	10 दिसंबर 2020 से 31 जुलाई 2021	पुरा होना।

12.	एफपॉलिमर मैट्रिक्स बांस फाइबर समग्र नमूनों का एब्रिकेशन	डॉ. राजेश पुरोहित एवं डॉ. आरएस राणा	11800/-	सीएसआईटी दुर्ग, छत्तीसगढ़	अगस्त 2021 से दिसंबर 2021	पुरा होना।
13.	डिज़ाइन एवं एफपॉलिमर मैट्रिक्स नैनो कम्पोजिट नमूनों का एब्रिकेशन	डॉ. आरएस राणा एवं डॉ. राजेश पुरोहित	11800/-	बीयूआईटी, भोपाल	नवंबर 2021 से अप्रैल 2022	पुरा होना।
14.	विकास एवं परीक्षणपॉलिमर मैट्रिक्स नैनो कम्पोजिट नमूने	डॉ. राजेश पुरोहित एवं डॉ. आरएस राणा	11800/-	बीयूआईटी, भोपाल	नवंबर 2021 से अप्रैल 2022	पुरा होना।
	परामर्श की कुल राशि		201440/-			

प्रयोगशाला प्रभारी: डॉ. राजेश पुरोहित और डॉ. आरएस राणा