Vidyavardhini's

College of Engineering & Technology

Vasai Road (W)

Department of Computer Engineering

Journal

Roll no.	58	Name	Keval Shah
Semester	VII	Class	BE
Course No.	CSL704	Academic Year	2020-21
Course Name	Robotics		

Index

Assignment 1 – Case Study of Industrial Robot

Articulated Robot IRB52 ABB Robotics Machine

ABB Machine's NEW model articulated robot IRB52

Features:

The IRB 52 is a compact painting robot designed specifically for painting small and medium sized parts in a wide range of industries.

It provides you with an affordable, professional and high-quality painting solution

With its small size and large working envelope, it is flexible and versatile, while its high speed and accuracy offers short cycle times.

It includes ABB's unique integrated paint system, IPS.

- Reliable – A combination of proven technology and well tested innovations
- Fast – High speed in all axes delivering improved cycle times
- Accurate – Outstanding position repeat-ability resulting in high process and product quality
- Flexible – versatile mounting options enabling space saving layouts
- Versatile – wide reach, long stroke and ability to handle large payloads
- Compact – smaller spray booth size and reduced ventilation needs resulting in increased energy efficiency

-

Fig: Articulated ROBOT IRB52 ABB Machine

Applications of Articulated Robot IRB52:

1. The compact design of the IRB 52 means smaller spray-booth sizes, reduced ventilation needs and system energy savings. Consistent reliable performance, repetitive, accuracy and high precision.
2. With its small size and impressive reach (1.2 or 1.45 meters) the IRB 52 is flexible and versatile, while its high speed and accuracy offer short cycle times and high-quality painting
3. The IRC5P is the newest generation paint robot control system specifically designed for the paint shop.

Technical Information

Robot Version	Payload on wrist(kg)	Reach vertical arm
IRB 52/1.2	7	0.475
IRB 52/1.45	7	0.7
Number of axes	6	
Protection	IP66 (wrist IP54)	
Controller	IRC5P Paint	

Technical Specification

Supply voltage	200-600 VAC, 3-phase, 50-60 Hz
Power consumption	Stand by <300 W, Production<800 W
Electrical safety	According to international standards
Robot footprint	484 x 648 mm
Robot controller	1450 x 725 x 710 mm
Robot unit weight	250 kg
Robot controller weight	180 kg
Height IRB 52/1.2	1069 mm
Height IRB 52/1.45	1294 mm
Robot unit	+5 °C (41 °F) to +40° C (104 °F)

Assignment 2 : Coordinate Transformation

In the general case, a homogeneous transformation matrix will represent both a rotation and a translation of the mobile frame with respect to the fixed frame. A sequence of individual rotations and translations can be represented as a product of fundamental homogeneous transformation matrices. However, because matrix mutiplication is not a commutative operation, the order in which the rotations and translations are to be performed is important. Furthermore, the mobile coordinate frame can be rotated or translated about the unit vectors of the fixed frame or about its own unit vectors. The effects of these different operations on the composite transformation matrix are summarized in the following algorithm:

Composite Homogeneous Transformations

1.Initialize the transformation matrix to T = I , which corresponds to the orthonormal coordinate frames F and M being coincident.

2. Represent rotations and translations using separate homogeneous transformation matrices.

3.Represent composite rotations as separate fundamental homogeneous rotation matrices.

4.If the mobile coordinate frame M is to be rotated about or translated along a unit vector of the fixed coordinate frame F , premultiply the homogeneous transformation matrix T by the appropriate fundamental homogeneous rotation or translation matrix.

5. If the mobile coordinate frame M is to be rotated about or translated along one of its own unit vectors, postmultiply the homogeneous transformation matrix T by the appropriate fundamental homogeneous rotation or translation matrix.

6. If there are more fundamental rotations or translations to be performed, go to step 4; otherwise, stop. The resulting composite homogeneous transformation matrix T maps mobile M coordinates into fixed F coordinates.

Thus composite homogeneous transformation matrices are built up in steps starting with the identity matrix, which corresponds to coincident mobile and fixed frames. Rotations and translations of frame M along the unit vectors of frame F are represented by premultiplication (multiplication on the left) by the appropriate fundamental homogeneous transformation matrix. Similarly, rotations and translations of frame M along one of its own unit vectors are represented by postmultiplication (multiplication on the right) by the appropriate fundamental homogeneous transformation matrix.

Solution:

$$[m^1]^F = T[m^1]^M$$

$$= Rot(\pi, 3)Tran(3i^2)[m^1]^M$$

$$= \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} -1 & -3 & 0 & 1 \end{bmatrix}^T$$

Code:

Python Implementation

```
rot =[[0,0,1,0],[0,1,0,0],[-1,0,0,0],[0,0,0,1]]

Trans=[[1,0,0,0],[0,1,0,3],[0,0,1,0],[0,0,0,1]]

p=[[2],[3],[4],[1]]

r=[[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]]

result=[[0],[0],[0],[0]]

for i in range(len(rot)):

    for j in range(len(Trans[0])):

        for k in range(len(Trans)):

            r[i][j] += rot[i][k] * Trans[k][j]




for i in range(len(r)):

    for j in range(len(p[0])):

        for k in range(len(p)):

            result[i][j] += r[i][k] * p[k][j]

for x in result:

   print(x)
```

Output:

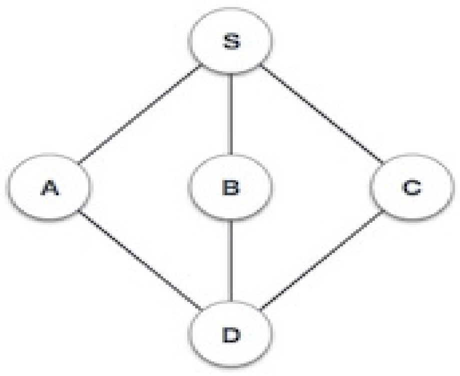
```
[4]

[6]

[-1]

[2]
```

Assignment : 3 Breadth First Search and Depth First Search

Breadth First Search (BFS) algorithm traverses a graph in a breadth ward motion and uses a queue to remember to get the next vertex to start a search, when a dead end occurs in any iteration.

Depth First Search (DFS) algorithm traverses a graph in a depth ward motion and uses a stack to remember to get the next vertex to start a search, when a dead end occurs in any iteration.

Objective: Example of BFS and DFS



BFS and DFS algorithm:

BFS

1. Check the starting node and add its neighbors to the queue.
2. Mark the starting node as explored.
3. Get the first node from the queue / remove it from the queue
4. Check if node has already been visited.
5. If not, go through the neighbors of the node.
6. Add the neighbor nodes to the queue.
7. Mark the node as explored.
8. Loop through steps 3 to 7 until the queues empty.

DFS

1. Check the starting node and push it in a stack.
2. Mark the starting node as explored.
3. If no adjacent vertex is found, pop up a vertex from the stack.
4. Check if node has already been visited.
5. If not, go through the neighbors of the node.
6. Add the neighbor nodes to the stack.
7. Mark the node as explored.
8. Loop through steps 3 to 7 until the stack empty.

Code:

Python Implementation

BFS Code:

```
graph = {
  'A' : ['B','C'],
  'B' : ['D', 'E'],
  'C' : ['F'],
  'D' : [],
  'E' : ['F'],
  'F' : []
}
visited = [] # List to keep track of visited nodes.
queue = []     #Initialize a queue

def bfs(visited, graph, node):
  visited.append(node)
  queue.append(node)
  while queue:
    s = queue.pop(0)
    print (s, end = " ")

    for neighbour in graph[s]:
      if neighbour not in visited:
        visited.append(neighbour)
        queue.append(neighbour)

bfs(visited, graph, 'A')
```

Output:

```
A B C D E F
```

<u>DFS Code:</u>

```
graph = {
    'A' : ['B','C'],
    'B' : ['D', 'E'],
    'C' : ['F'],
    'D' : [],
    'E' : ['F'],
    'F' : []
}

visited = set() # Set to keep track of visited nodes.

def dfs(visited, graph, node):
    if node not in visited:
        print (node, end =" “)
        visited.add(node)
        for neighbour in graph[node]:
            dfs(visited, graph, neighbour)
dfs(visited, graph, 'A')
```

Output:

```
A B D E F C
```

Assignment 4: Case study on Expert System

Decription:

A process in the development of mechanized production in which the control and monitoring fun ctions previously performed by humans are transferred to instruments and automatic devices. Au tomation of production is the basis of the development of modern industry and a general trend in technical progress. Its goal is to improve the efficiency of labor and the quality of manufactured products and to create conditions for the optimum utilization of all production resources. Partial, integrated, and total automation of production are distinguished.

Expert System on Production Management :

Supervisory control and data acquisition (SCADA) are a system of software and hardware elements that allows industrial organizations to:

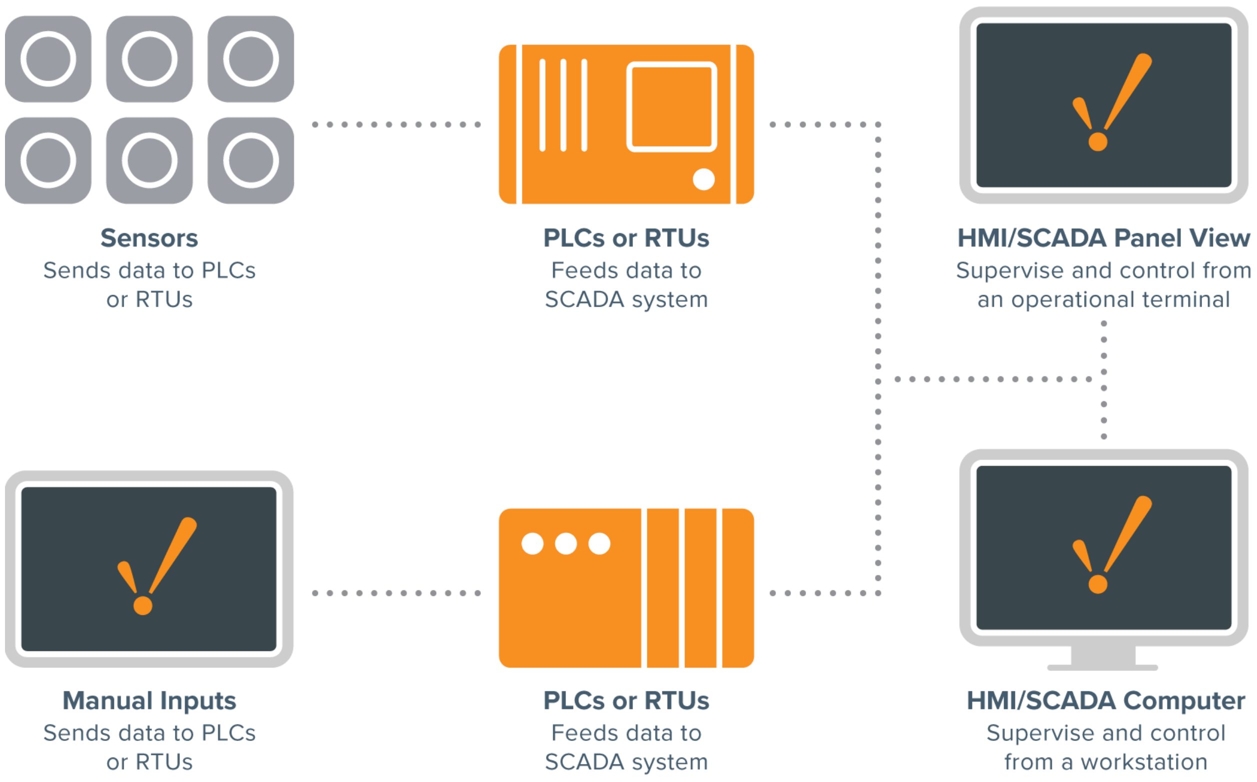
- Control industrial processes locally or at remote locations
- Monitor, gather, and process real-time data
- Directly interact with devices such as sensors, valves, pumps, motors, and more through human-machine interface (HMI) software
- Record events into a log file

SCADA systems are crucial for industrial organizations since they help to maintain efficiency, process data for smarter decisions, and communicate system issues to help mitigate downtime.

The basic SCADA architecture begins with programmable logic controllers (PLCs) or remote terminal units (RTUs). PLCs and RTUs are microcomputers that communicate with an array of objects such as factory machines, HMIs, sensors, and end devices, and then route the information from those objects to computers with SCADA software. The SCADA software processes, distributes, and displays the data, helping operators and other employees analyze the data and make important decisions.

For example, the SCADA system quickly notifies an operator that a batch of product is showing a high incidence of errors. The operator pauses the operation and views the SCADA system data via an HMI to determine the cause of the issue. The operator reviews the data and discovers that Machine 4 was malfunctioning. The SCADA system's ability to notify the operator of an issue helps him to resolve it and prevent further loss of product.

A Basic SCADA Diagram


Sensors
Sends data to PLCs or RTUs
PLCs or RTUs
Feeds data to SCADA system
HMI/SCADA Panel View
Supervise and control from an operational terminal
Manual Inputs
Sends data to PLCs or RTUs
PLCs or RTUs
Feeds data to SCADA system
HMI/SCADA Computer
Supervise and control from a workstation

Who Uses SCADA?

SCADA systems are used by industrial organizations and companies in the public and private sectors to control and maintain efficiency, distribute data for smarter decisions, and communicate system issues to help mitigate downtime. SCADA systems work well in many different types of enterprises because they can range from simple configurations to large, complex installations. SCADA systems are the backbone of many modern industries, including:

- Energy
- Food and beverage
- Manufacturing
- Oil and gas
- Power
- Recycling
- Transportation
- Water and waste water
- And many more

Famous Production Management System(SCADA) Expert Systems:

Softpro:Softpro is a software platform for idea BOX series, with the advantages of variety of programming languages, modular programming structures and integration programming ways.

B-Scada ;B-Scada provides software and hardware solutions for the monitoring and analysis of real time data in the SCADA, IoT and Smart City domains

Advance HMI:The Advanced MI software is an open software used to create HMI (Human Machine Interface) applications that communicate with your PLC or device.

ASTRA HMI/SCADA:ASTRA is Windows based Human Machine Interface Software for Supervisory Control and Data Acquisition (SCADA). ASTRA is a full-fledged, feature packed and user-friendly SCADA package that helps industrial users to address their manufacturing process automation needs ASTRA is Windows based Human Machine Interface Software for Supervisory Control and Data Acquisition (SCADA). ASTRA is a full-fledged, feature packed and user-friendly SCADA package that helps industrial users to address their manufacturing process automation needs

Atvise: atvise portal and the connected M1 controllers offer a lightweight concept, allowing a gradual extension according to the requirements.

Claroty: The Claroty Platform is an integrated set of cyber security products that provides extreme visibility, unmatched cyber threat detection, secure remote access, and risk assessments for industrial control networks (ICS/OT)

Datarecon: With DataRecon distributed architectures can be implemented Intranet / Internet and use the telephone network to communicate with remote devices or sending text messages.