ARTIFICIAL INTELLIGENCE - DECEMBER 2014
(SEMESTER 7)

TOTAL MARKS: 80
TOTAL TIME: 3 HOURS

(1) Question 1 is compulsory.
*(2) Attempt any **three** from the remaining questions.*
(3) Assume data if required.
(4) Figures to the right indicate full marks.

---

1 (a) Explain Heuristic function with example. (5 marks)

1 (b) Explain Robot workspace. (5 marks)

1 (c) Describe unsupervised learning with suitable example. (5 marks)

1 (d) List and define kinetic parameters. (5 marks)

---

2 (a) Describe the following sensors- (10 marks)
i) Sonar
ii) Infrared

2 (b) Explain A* algorithm with example. (10 marks)

---

3 (a) Obtain Inverse kinematic solution for 4-axis SCARA robot. (10 marks)

3 (b) Compare different uniformed search strategies. (10 marks)

---

4 (a) Describe Hill climbing algorithm. What are it's limitations. (10 marks)

4 (b) Explain various methods of knowledges representation with example. (10 marks)

---

5 (a) Define partial order planner. Explain STRIPS representation of planning problem. (10 marks)

5 (b) Give steps in designing the reactive behavioral system. (10 marks)

---

6 (a) What are PEAS descriptors? Give PEAS descriptors for (10 marks)
i) Part-picking Robot
ii) WUMPUS world.

6 (b) Explain supervised, unsupervised and reinforcement learning with example. (10 marks)

Write short notes on any four:

7 (a)PROLOG (5 marks)

7 (b) Belief network (5 marks)

7 (c) Forward and inverse kinematics (5 marks)

7 (d) Crypt Arithmetic (5 marks)

7 (e)GPS (5 marks)

7 (f) Uniform and Inform search. (5 marks)

1 (a) Explain Heuristic function with example. --- 5 Marks

ADD NOTE BOOKMARK

This question was repeated in May 15.

Answer:

For an 8-puzzle problem, admissible heuristic is the number of misplaced tiles.

7	2	4
5		6
8	3	1

Start State

1	2	3
4	5	6
7	8	

Goal State

h(n)=8 => all 8 tiles are misplaced from goal

The 8-puzzle was one of the earliest heuristic search problems. The objective of the puzzle us to side the tiles horizontally or vertically into empty space until configuration matched the goal configuration.

We can defince two heuristic functions for 8-puzzle problem

1. h_1=the number of misplaced tiles, all of the eight tiles are out of position in the diagram above, so that the start would have h_1=8. h_1 is admissible heuristic, because it is clear that any tile that is out of place must be moved at least once.
2. h_2=the sum of distances of the tiles from their goal positions. Because tiles cannot move along diagonals, the distance we will count is the sum of horizontal and vartical distances. This is something called the city of block distance or Manhattan distance. h_2 is also admissible because all any move can do is move over tile one step closer to the goal.

1 (c) Describe robot workspace. --- 5 Marks

ADD NOTE BOOKMARK

This question was repeated in Dec 14.

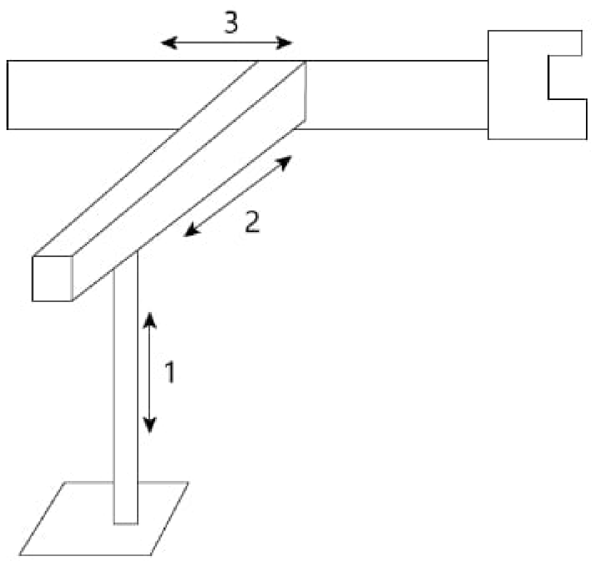
Answer:

Robot Workspace:

The workspace or configuration of a robot is defined as the locus of points in three dimensional space that can be reached by the wrist. Accordingly we get five configurations of the robot based on work envelope geometries or co-ordinate geometries which are as given as follows:

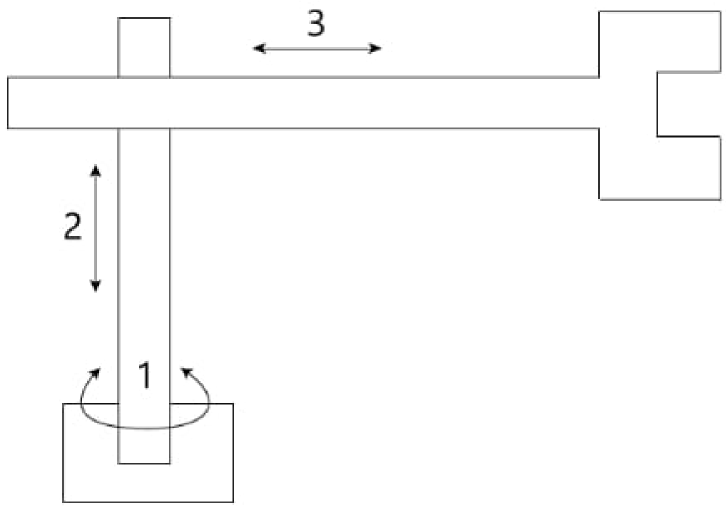
1. Cartesian robot
2. Cylindrical robot
3. Spherical robot
4. SCARA
5. Articulated robot

1. Cartesian robot:



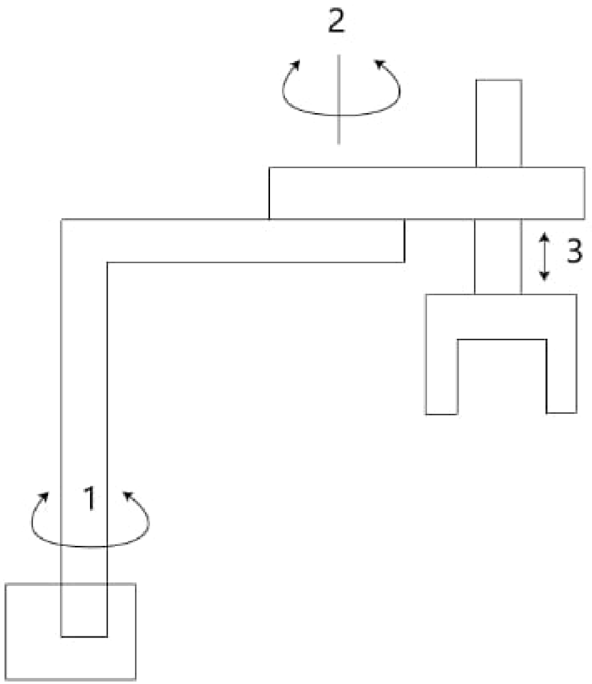
- In Cartesian robot, all three joints are prismatic joints.
- The workspace of Cartesian robot is rectangular box.

2. Cylindrical robot:



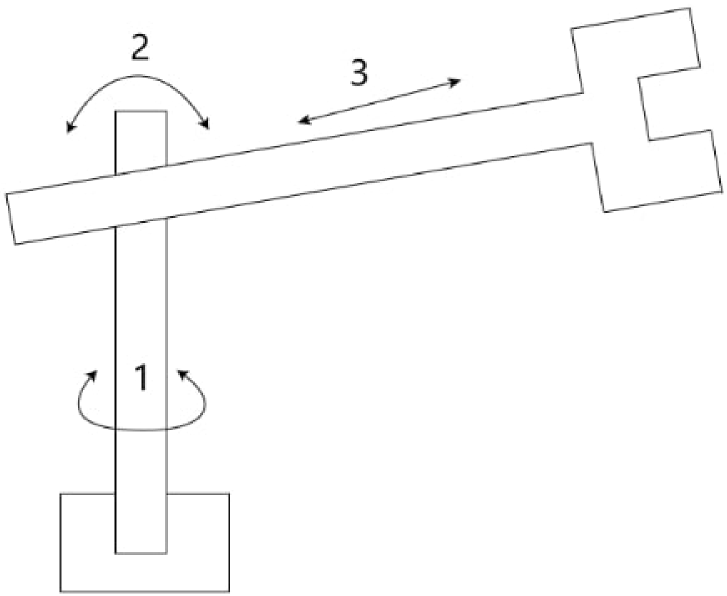
- In cylindrical robot, there is one rotary joint and two prismatic joint.
- Workspace of cylindrical robot is volume between 2 concentric cylinders.

3. Spherical robot:



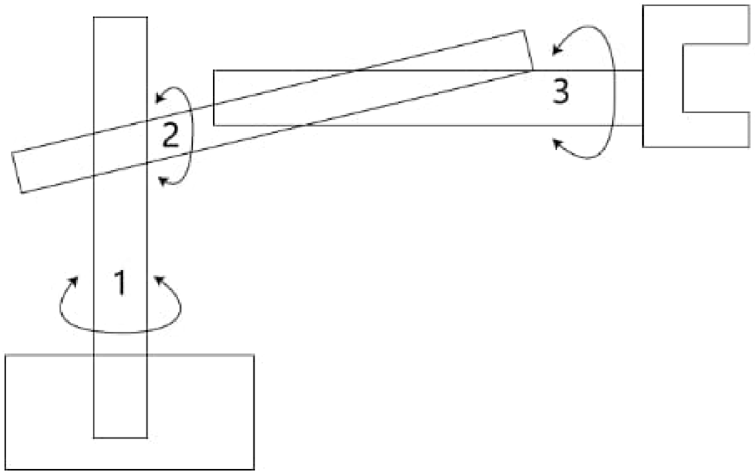
- In spherical robot, there are two rotary joints and one prismatic joint.
- Workspace of spherical robot is volume between 2 concentric cylinders.

4. SCARA robot:



- In SCARA robot, all axis are parallel, there are two rotary joints and one prismatic joint.
- Workspace of SCARA robot is complex.

5. Articulated robot:



- In articulated robot, all joints are rotary joints.
- Workspace of articulated robot is complex.

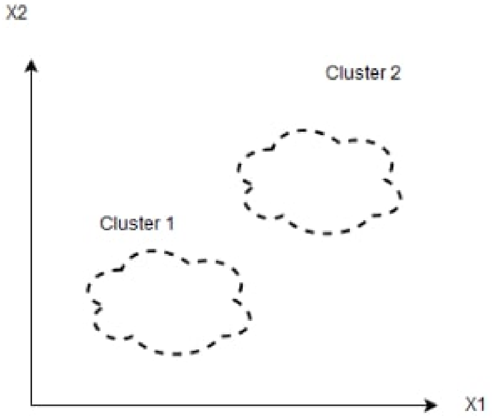
1 (c) Describe unsupervised learning with suitable example. ***--- 5 Marks***

ADD NOTE BOOKMARK

Answer:

Unsupervised Learning:

- In this method critic provides only input. The desired output is not provided.
- Agent forms clusters from the given input training samples
- The unknown sample is classified into one of the clusters depending on its closeness
- This method is mainly useful for classification type of problems
- Example: Clustering



1 (d) List and define kinetic parameters. --- 5 Marks

ADD NOTE BOOKMARK

Answer:

A robotic manipulator can be modeled as a chain of rigid links interconnected by either revolute or prismatic joints. The relative position and orientation of two successive links can be specified by two joint parameters.

θ_k - joint angle. It is the rotation about z^{k-1} needed to make axis x^{k-1} parallel with axiz x^k

d_k - joint distance. It is the translation along z^{k-1} needed to make axix x^{k-1} intersects with axis x^k.

The relative potition and orientation of the axis of two successive joints are specified by two link parameters.

α_k - Link twist angle. It is the rotation about x^k needed to make axis z^{k-1} parallel axis z^k.

a_k - Link lenght. It is the translation along x^k needed to mek axis zk-1 intersect with axis z^k

Kinematic Parameters:

Arm Parameter	Symbol	Revolute Joint (R)	Prismatic Joint (P)
Joint angle	θ	Variable	Fixed
Joint distance	d	Fixed	Variable
Link length	a	Fixed	Fixed
Link twist angle	α	Fixed	Fixed

2 (a) Describe the following sensors-
i) Sonar ***--- 10 Marks***
ii) Infrared

ADD NOTE BOOKMARK

Answer:

1. Sonar:

- Sonar is a popular sensor in robotics that employs acoustic pulses and their echoes to measure range to an object.
- Since the sound speed is usually known, the object range is proportional to the echo travel time. At ultrasonic frequencies the sonar energy 1s concentrated in a beam, providing directional information in addition to range.
- A well-known example of ultrasound sensing is submarine sonar, where knowledge of the speed of sound in water allows the submarine crew to determine the distance to an object through measurement of the time for a pulse to travel out and be reflected back.
- Mobile robots on land also measure distances with time of flight systems.
- Three main sonar sensing techniques exist:

1. Pulse Mode:

- In the pulse mode, distances are determined by the time of flight of a short sound pulse. Robots use the pulse mode at larger distances to determine their own position relative to their environment and at short distances to determine the range of target object.

2. Phase Mode:

- In the phase mode, a continuous sound wave is transmitted and changes in the phase of the reflected wave allow movements of a target relative to a robot to be measured accurately. The phase mode is used when a single dominant echoing target is present.

3. Frequency Modulation Mode:

- In the frequency modulation mode, a continuous sound wave is transmitted as in the phase mode. The signal is frequency modulated; typically using a linear saw tooth modulation.

2. Infrared:

- IR distance sensors are a low-cost, easy-to-use analog distance sensors. IR Sensors produce a constantly updated analog output signal depending upon the intensity of the reflected IR, which in turn can be used to calculate approximate range. These sensors are perfect for obstacle avoidance, line following, and even map building.
- Side-looking Infrared Emitters and IR Detectors are simple devices operating at 940nm and work well for generic IR systems including remote control and touch-less object sensing.
- IR Photo reflectors are generally used in line-following or encoder application, at short distances to detect white or black colours. These sensors are easy to use and low in cost.
- Infrared {IR) sensors are widely used as proximity sensors. They offer lower cost and faster response times than ultrasonic (US) sensors. However, because of their non-linear behaviour and their dependence on the reflectance of surrounding objects, measurements based on the intensity of the back-scattered IR light are very imprecise for ranging purposes. For this reason, environment maps made with this type of sensor are of poor quality, and IR sensors are almost exclusively used as proximity detectors in mobile robots.

2 (b) Explain A* algorithm with example. --- 10 Marks

ADD NOTE

BOOKMARK

This question was repeated in May 15.

Answer:

It is an uninformed search technique. It uses additional information beyond problem formulation or tree. Search is based on evaluation function f(n). Evaluation function is based on both heuristic function h(n) and g(n).

i.e. f(n)=g(n)+h(n)

$$\text{Root} \xrightarrow{g(n)} n \xrightarrow{h(n)} \text{goal}$$

For its implementation it uses two queue:

i. OPEN

ii. CLOSE

OPEN queue is priority queue which is arranged in ascending order of f(n).

Algorithm:

Step 1: Create a single member queue comprising of root node

Step 2: If first member of queue is goal, then goto step 5

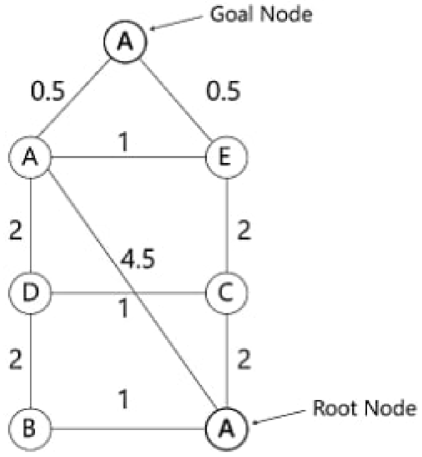
Step 3: If first member of queue is not goal, then remove t from queue and ADD to close queue. Consider it children, if any, add them into the queue, in ascending order of evaluation function f(n).

Step 4: If queue is not empty then go to step 2. If queue is empty then go to step 6

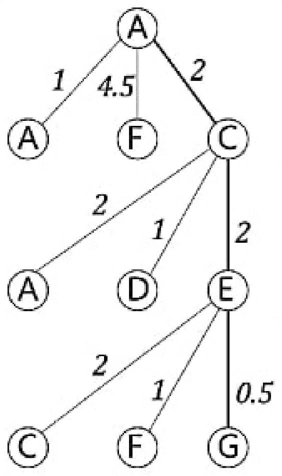
Step 5: print "SUCCESS" and stop

Step 6: print "FAILURE" and stop

Consider an example given below:



- Here we use straight line distance as heuristic h(n).
- We first convert graph into tree and then use informed search technique to solve the problem



Pass 1 of algo $\begin{cases} f(B) = g(B) + h(B) = 1 + 4.5 = 5.5 \\ f(F) = g(F) + h(F) = 4.5 + 0.5 = 5 \\ f(C) = g(C) + h(C) = 2 + 2.5 = 4.5 \end{cases}$

Pass 2 of algo $\begin{cases} f(A) = g(A) + h(A) = 4 + 4.5 = 8.5 \\ f(D) = g(D) + h(D) = 3 + 2.5 = 5.5 \\ f(E) = g(E) + h(E) = 4 + 0.5 = 4.5 \end{cases}$

Pass 3 of algo $\begin{cases} f(C) = g(C) + h(C) = 6 + 2.5 = 5.5 \\ f(F) = g(F) + h(F) = 5 + 0.5 = 5 \\ f(G) = g(G) + h(G) = 4.5 + 0 = 4.5 \end{cases}$

OPEN	CLOSE
$[A]$	
$4.5 \quad 5 \quad 5.5$	
$[C, \quad F, \quad B]$	$[A]$
$4.5 \quad 5 \quad 5.5 \quad 5.5 \quad 8.5$	
$[E, \quad F, \quad B, \quad D, \quad A]$	$[A, \quad C]$
$4.5 \quad 5.5 \quad 5.5 \quad 5.5 \quad 8.5 \quad 8.5$	
$[G, \quad F, \quad B, \quad D, \quad A, \quad C]$	$[A, \quad C, \quad E]$

First member of queue is goal "SUCCESS"

Advantages:

- It is a complete algorithm
- It is optimum because it considers evaluation function i.e. $f(n)=g(n)+h(n)$ where $h(n)$ is admissible heuristic

Disadvantages:

- It generates same node again and again
- Hence large memory is needed

4 (a) Obtain Inverse Kinematic solution for 4-axis SCARA robot. **--- 10 Marks**

ADD NOTE

BOOKMARK

This question was repeated in Dec 14.

Answer:

A robotic manipulator can be modeled as a chain of rigid links interconnected by either revolute or prismatic joints. The relative position and orientation of two successive links can be specified by two joint parameters.

- θ_k - joint angle. It is the rotation about z^{k-1} needed to make axis x^{k-1} parallel with axiz x^k
- d_k - joint distance. It is the translation along z^{k-1} needed to make axix x^{k-1} intersects with axis x^k.
- The relative potition and orientation of the axis of two successive joints are specified by two link parameters.
- α_k - Link twist angle. It is the rotation about x^k needed to make axis z^{k-1} parallel axis z^k.
- a_k - Link lenght. It is the translation along x^k needed to mek axis z^{k-1} intersect with axis z^k

Kinematic Parameters:

Arm Parameter	Symbol	Revolute Joint (R)	Prismatic Joint (P)
Joint angle	θ	Variable	Fixed
Joint distance	d	Fixed	Variable
Link length	a	Fixed	Fixed
Link twist angle	α	Fixed	Fixed

Given:

The final position of the robot $P=(P_x,P_y,P_z)$

To find

$\phi_1, \phi_2, \phi_3\ and\ \phi_4$ for the SCARA robot

The final T matrix is given below:

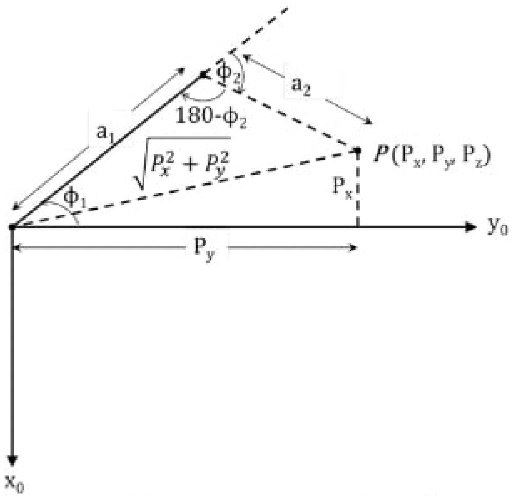
$$T_4^0 = \begin{bmatrix} C_{1-2-4} & S_{1-2-4} & 0 & a_1C_1 + a_2C_{1-2} \\ S_{1-2-4} & -C_{1-2-4} & 0 & a_1S_1 + a_2S_{1-2} \\ 0 & 0 & -1 & d_1 - q_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- To find $\varnothing_2$: if we squre and sum P_x and P_y, we can get an expression in $\varnothing_2$

$$P_x^2+P_y^2=(a_1C_1+a_2C_{1-2})^2+(a_1S_1+a_2S_{1-2})^2$$

$$P_x^2+P_y^2=a_1^2+a_2^2+2a_1a_2C_1(C_1C_2+S_1S_2)+2a_1a_2S_1(S_1C_2-S_2C_1)$$

$$P_x^2+P_y^2=a_1^2+a_2^2+2a_1a_2C_1^2C_2+2a_1a_2S_1^2C_2$$



Solution to ϕ_2 of Adept, as seen from above (along Z-axis)

$P_x^2+P_y^2=a_1^2+a_2^2+2a_1a_2C_2$

$$C_2 = \frac{P_x^2+P_y^2-a_1^2+a_2^2}{2a_1a_2}$$

$$S_2 = \sqrt{1-C_2^2}$$

2. This is really just the derivation of the Law of Cosines which we can also use to find Ø$_2$ (see figure above):

$a_1^2 + a_2^2 - 2a_1a_2cos(180 - \phi_2) = p_x^2 + p_y^2$ (Law of Cosines)

$$cos(180 - \phi_2) = \frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{-2a_1a_2}$$

$$-cos(\phi_2) = \frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{-2a_1a_2}$$

$$cos(\phi_2) = \frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{-2a_1a_2}$$

- To solve equations in two unknowns (C_1,S_1)

$$a_1C_1 + a_2C_1C_2 + a_2S_1S_2 = P_x$$
$$a_1S_1 + a_2S_1C_2 + a_2S_2C_1 = P_y$$
$$(a_1 + a_2C_2)C_1 + (a_2S_2)S_1 = P_x$$
$$(-a_2 + S_2)C_1 + (a_1 + a_2C_2)S_1 = P_y$$
$$S_1 = \frac{a_2S_2P_x + (a_1 + a_2C_2)P_y}{(a_2S_2)^2 + (a_1 + a_2C_2)^2}$$
$$\theta_2 = atan_2(a_2S_2P_x + (a_1 + a_2C_2)P_y, (a_1 + a_2C_2)P_x - a_2S_2P_x)$$

- To solve for Ø$_3$:

$$P_z = d_1 - \phi_3 - d_4;$$
$$\phi_3 = d_1 - d_4 - P_z$$

- 5. To solve for Ø$_4$: The final roll angle cannot be determined from the position vector [px,py,pz]

If we are given the orientation matrix, then we can use the ratio of N$_x$, N$_y$ to find Ø$_4$

$$\tan_{1-2-4} = \frac{S_{1-2-4}}{C_{1-2-4}} = \frac{N_y}{N_z}$$
$$\phi_1 - \phi_2 - \phi_4 = a\tan_2(Ny, Nz)$$
$$\phi_4 = -a\tan_2(N_y, N_z) + \phi_1 - \phi_2$$

3 (b) Compare different uniformed search strategies. --- *10 Marks*

ADD NOTE BOOKMARK

Answer:

Following are the different uniformed search strategies:

1. Breadth first search
2. Uniform cost search
3. Depth first search
4. Depth limited search
5. Iterative deepening search
6. Bidirectional search

1. Breadh first search:

- In Breadth first search, we search the graph from the root node in order of the distance from the root.
- Rather than digging deep down into the graph, progressing further and further from the root
- BFS chechs each node nearest the root before descending to the next level.
- The implementation of BFS uses FIFO (first in first out) principle.

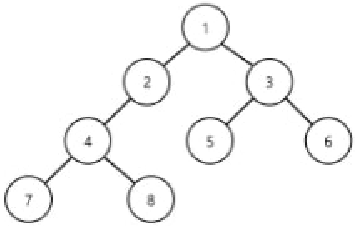


Figure: Search order for a BFS

2. Uniform cost search:

- Uniform cost search can be applied to find the least cost path through a graph by maintaining an ordered list of nodes in order of descending cost,
- This allows us to evaluate the least cost path first.

3. Depth first search:

- The Depth first search (DFS) algorithm is a technique for searching a graph that begins at the root node, and exhaustively searches each branch to its greatest depth before backtracking to previously unexplored branches.
- Nodes found but yet to be reviewed are stored in a LIFO (last in first out) principle.

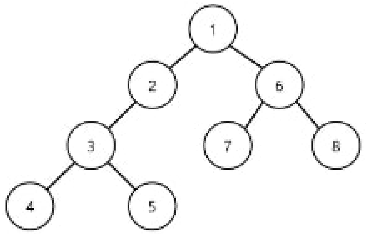


Figure: Searh order of the DFS

4. Depth limited search:

- Depth limited search is a modification of depth first search wherein the algorithm will not be descend below a certain provided depth.
- Any nodes below that depth are omitted from the search.

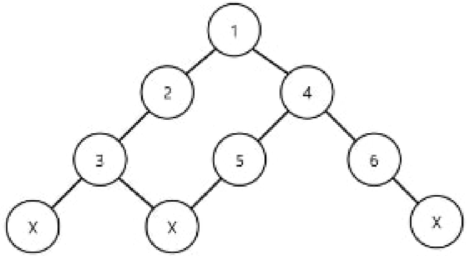


Figure: Search order for the DLS

5. Iterative deepening search:

- Iterative deepening search is a derivative of DLS and combines the features of DFS with that of BFS.
- IDS operates by performing DLS searches with increased depths until goal is found.
- The depth begins at one, and increases until the goal is found, or no further nodes can be enumerated

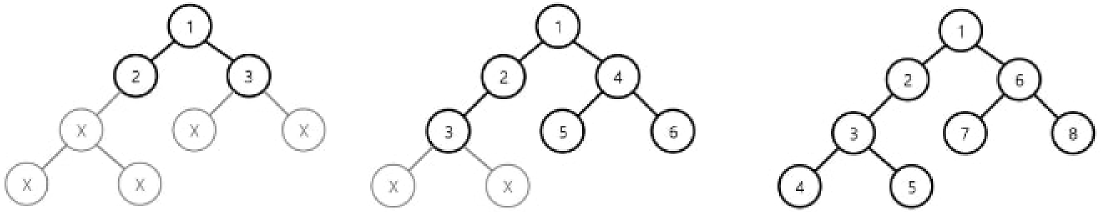


Figure: Search order for the IDS

6. Bidirectional Search:

- The Bidirectional search algorithm is a derivative of BFS that operates by performing two BFS searches simultaneously, one beginning from the root node and the other from the goal node.
- When the two searches meet in the middle, a path can be reconstructed from the root to the goal.

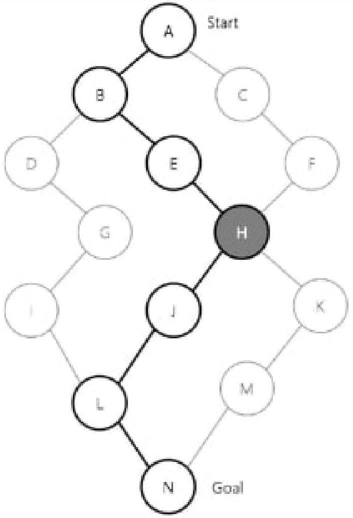


Figure: Search order for BS

Parameters	Breadth First Search	Uniform Cost Search	Depth First Search	Depth Limited Search	Iterative Deepening Search	Bidirectional Search
Completeness	YES	YES	NO	NO	YES	YES
Optimality	YES	YES	NO	YES if 1>=d	YES	YES
Space Complexity	$O(b^{d+1})$	$O(b^{1+[c^*/E]})$	O(bm)	O(bl)	O(bd)	$O(b^{d/2})$
Time Complexity	$O(b^{d+1})$	$O(b^{1+[c^*/E]})$	$O(b^m)$	$O(b^l)$	$O(b^d)$	$O(b^{d/2})$

4 (a) Describe Hill climbing algorithm. What are it's limitations. --- *10 Marks*

ADD NOTE BOOKMARK

This question was repeated in May 16.

Answer:

- Hill climbing is an optimization technique for solving computationally hard problems. The idea is to start with a sub-optimal solution to a problem and then repeatedly improve the solution (walk up the hill) until some condition is maximized (the top of the hill is reached)
- It is best used in problems with the property that the state description itself contains all the information needed for a solution. The algorithm is memory efficient since it does not maintain a search tree. It looks only at the current state and immediate future states.
- Hill climbing attempts to iteratively improve the current state by means of an evaluation function. It is simply a loop that continually moves in the direction of increasing value- that is, uphill. It terminates when it reaches a "peak" where no neighbour has a higher value.
- Consider all the possible states laid out on the surface of a landscape. The height of any point on the landscape corresponds to the evaluation function of the state at that point.
- It contrast with other iterative improvement algorithms, hill-climbing always attempts to make changes that imrove the current state. In other words, hill-climbing can only advance if there is a higher point in the adjacent landscape.

Algorithm

Step 1: function Hill_Climbing (problem) returns a solution state

Step 2: inputs: problem //a problem

Step 3: local variables: current // a node

Step 4: next // a node

Step 5: current <- Make-Node (initial state) // make random

Step 6: loop do // initial state

Step 7: next<- a highest valued successor of current

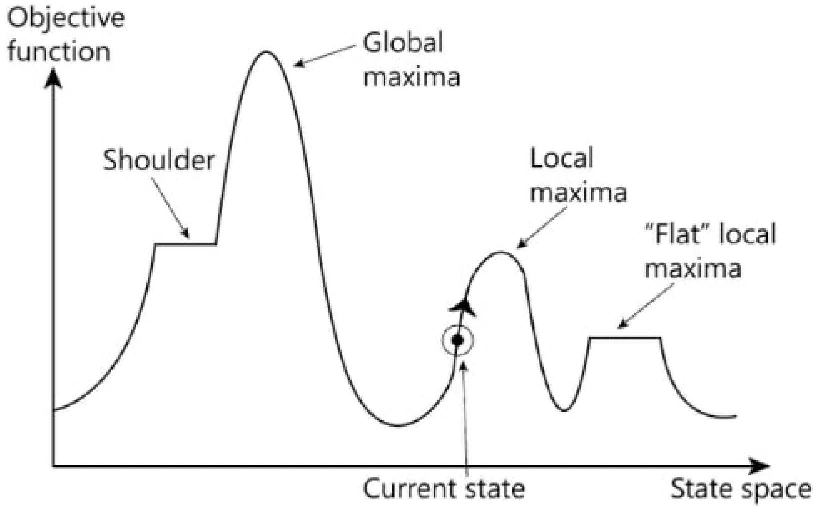
Step 8: if value [next] < value [current] then return current

Step 9: current <-next

Step 10: end

Limitations:

Consider following state-space landscape



1. Local Maxima:

A problem with hill climbing is that it will find only local maxima. Unless the heuristic is convex, it may not reach a global maximim. Other local search algorithms try to overcome this problem such as stochastic hill climbing, random walks and simulated appealing. This problem of hill climbing can be solved by using random hill climbing search technique.

2. Ridges:

A ridge is a curve in the search space that leads to a maximum but the orientation of the ridge compared to the available moves that are used to climb is such that each move will lead to a smaller point. In other words, each point on a ridge looks to the algorithm like a local maximum, eve, through the point is part of a curve leading to a better optimum

3. Plateau:

Another problem with hill climbing is that of a plateau, which occurs when we get to a "flat" part of the search space i.e. we have a path where the heuristics are all very close together. This kind of flatness can cause the algorithm to cease progress wander aimlessly.

4 (b) Explain various methods of knowledges representation with example. ***--- 10 Marks***

ADD NOTE BOOKMARK

This question was repeated in May 15.

Answer:

1. Predicate logic:

- Formal logic is a language with its own syntax, which defines how to make sentences, and corresponding semantics, which describe the meaning of the sentences.
- Sentences can be constructed using proposition symbols (P, Q, R) and Boolean connectives, such as conjunction (and), disjunction (or), implication (P implies Q).
- Modus Ponens: where given a rule, A implies B, if A is true, we can infer that B is also true.
- Predicate logic introduce concept of quantifiers which allow us to refer to sets of objects. Using objects, attributes, and relations, we can represent almost any type of knowledge.
- Two quantifiers:
    - Universal (all objects of this type have this attribute)
    - Existential (there exists some object that has the specified attribute)
- Example: "Minnesota is cold in the winter" can be represented in three single paramters:

    Place (Minnesota), Temperature (cold) and session (winter)

    Or it can be represented a single relation:

    Cold(Minnesota,winter),winter(Minnesota,cold)

2. Resolution:

- Resolution is an algorithm for proving facts true or false by virtue of contradiction. If we want to prove a theorem X is true, we have to show that the negation of X is not true.
- Example:

Suppose that we know the following two facts

A) not feathers (Tweety) or bird (Tweety)

B) feathers (Tweety)

C) no bird(Tweety)

D) bird (Tweety)

3. Unification:

- Unification is a technique for taking two sentences in predicate logic and finding a substitution that makes them look the same.
    - A variable can be replaced by a constant
    - A variable can be replaced by another variable
    - A variable can be replaced with a predicate, as long as the predicate does not contain that variable
- Given the following set of predicates:
    - Hates(X,Y)
    - Hates(George, broccoli)
    - Hates(Alex, spinach)

5 (a) Define partial order planner. Explain STRIPS representation of planning problem. ***--- 10 Marks***

ADD NOTE

BOOKMARK

Answer:

Partial Order Planner:

- A partial order plan is an approach to automated planning with partially ordered set of actions.
- A partial order planner is a planner that constructs a partial order plan, in which the current ordering of plan steps is only done to the minimal extent needed to measure proper performance. It is a planning algorithm that can place two action into a plan without specifying which comes first.

STRIPS Representation:

- The STanford Research Institutes Problem Solver (STRIPS) system is considered one of the first planning algorithms and representations.
- STRIPS representations have been the most common logic-based representations for discrete planning problems
- In STRIPS planning representation the clauses that are not subject to the direct effect of an action are assumed persistent
- The STRIPS planning problem can be described with < O, I,G? where
  - O is a set of STRIPS operators
  - I is a set of positive literals that defnie the initial state
  - G is a set of positive literals that define the goals of the problem
- Sample Problem:

  Initial state: you are at home and don't have milk, bananas and a drill

  Goal state: you are at home and you have milk, bananas, and a drill

- STRIPS Representation:

  Initial state:

  At(Home)

  Sells(SM,Bananas)

  Sells(HWS, Drill)

  Sells(SM, Milk)

  Goal state:

  At(Home)

  Have(Bananas)

  Have(Drill)

  Have(Milk)

  Operators:

  Go(here,there)

  Precond: At(here)

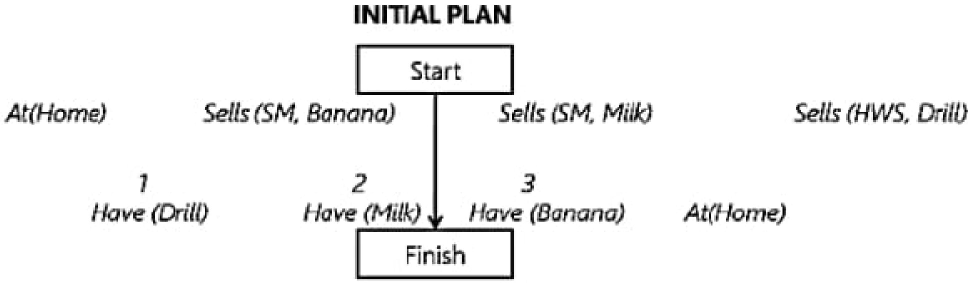
  Effect: At(there) AND NOT At(here)

  Buy(x,store)

  Precond: At(store) and Sells(store,x)

Precond: At(store) and Sells(store,x)

Effect: Have(x)



5 (b) Give steps in designing the reactive behavioral system. ***--- 10 Marks***

ADD NOTE BOOKMARK

Answer:

- Behaviour-based robotics or behavioural robotics is an approach in robotics that focuses on robots that are able to exhibit complex-appearing behaviour despite little internal variable state to model its immediate environment, mostly gradually correcting its actions via sensor-motor links.
- Most behaviour-based systems are also reactive, which means they need no programming of internal representations of what a chair looks like, or what kind of surface the robot is moving oin. Instead all the information is gleaned from the input of the robot's sensors. The robot uses that information to gradually correct its actions according to the changes in immediate environment.
- Behaviour-based robots (BBR) usually show more biological-appearing actions than their computing-intensive counterparts, which are very deliberate in their actions. A BBR often makes mistakes, repeats actions, and appears confused, but can also show the anthropomorphic quality of tenacity. Comparisons between BBRs and insects are frequent because of these actions. BBRs are sometimes considered examples of Weak artificial intelligence,
- ***Steps*** in ***designing reactive Behavioural System***:
  1. Describe the task,
  2. Describe the robot,
  3. Describe the environment,
  4. Describe how the robot should act in response to its environment,
  5. Refine each behaviour
  6. Test each behaviour independently,
     Test with other behaviours,
  7. and repeat the process as needed.
- *Describe the Task. Robot. and Environment:*

  The first 3 steps serve to remind the designer to specify the ecological niche of the robot. For example, in case of robotic car, the designer should describe its design and structure, whether it'll be run inside the house or on the road, what will it be expected to do (fetch a newspaper from the door etc.)

- *Describe how the robot should act:*

  The fourth step begins the iterative process of identifying and refining the set of behaviours for the task. Defining the ecological niche defines constraints and opportunities but doesn't necessarily introduce major insights into the situatedness of the robot. How it acts and reacts to the range of variability in its. ecological niche is taken care of in this step.
  For example, a robotic car should follow a straight line and stay in the middle. Its camera should point in a particular direction in the respective situation etc.

- *Refine and test each behaviour*

  Steps 5-7 are less abstract. Once the candidate set of behaviours are proposed, the designer starts working on designing each individual behaviour, specifying its motor and perceptual schemas.
  The designer usually programs each schema independently, he integrates them into a behaviour and tests the behaviour thoroughly in isolation before integrating all behaviours. This style of testing is consistent with good software engineering principles and emphasizes the practical advantages of the reactive paradigm.

- The five ***characteristics*** of Reactive Behavioural System:
  1. Robots are situated agents operating in an ecological niche. Here situated agent means that the robot is an integral part of the world. A robot has its own goals and intentions. When a robot acts, it changes the world, and receives immediate feedback about the world through sensing. What the robot senses affects its goals and how it attempts to meet them, generating a new cycle of actions.

2. Behaviours serve as the basic building blocks for robotic actions, and the overall behaviour of the robot is emergent. Behaviours are independent, computational entities and operate concurrently. The overall behaviour is emergent: there is no explicit "controller" module which determines what will be done, or functions which call other functions. There may be a coordinated control program in the schema of behaviour, but there is no external controller of all behaviours for a task.
3. Only local, behaviour-specific sensing is permitted. The use of explicit abstract representational knowledge in perceptual processing, even though it is behaviour-specific, is avoided. Any sensing which does require representation is expressed in ego-centric (robot-centric) coordinates.
4. These systems inherently follow good software design principles. The modularity of these behaviours supports the decomposition of a task into component behaviours. The behaviours are tested independently, and behaviours may be assembled from primitive behaviours.
5. Animal models of behaviour are often cited as a basis for these systems or a particular behaviour. Unlike in the early days of AI robotics, where there was a conscious effort to not mimic biological intelligence, it is very acceptable under the reactive paradigm to use animals as a motivation for a collection of behaviours.

6 (a) What are PEAS descriptors? Give PEAS descriptors for
i) Part-picking Robot ***--- 10 Marks***
ii) WUMPUS world.

ADD NOTE BOOKMARK

Answer:

- A Rational Agent should select an action that is expected to maximize its performance measure, provided by percept sequence and whatever built-in knowledge it has.
- To design a rational agent, we first need to think about task environment which are essentially problems to which the rational agents are "solutions".
- To design a task environment we need four main factors which can completely describe the environment.
- These four factors are called by the acronym PEAS: Performance measure, Environment, Actuators and Sensors.

PERFORMANCE MEASURE:

- Performance measure is how we measure the system's achievements.
- For example, consider the case of an automated driver. Its performance can be measured in terms of desirable qualitie like minimizing fuel consumption, minimizing cost and time, maximizing safety and passenger comfort and maximizing profits. Many-a-times, these goals conflict, so there will be optimum trade-offs involved.

ENVIRONMENT:

- Environment involves who the agent is interacting with.
- For example, an taxi driver deals with a variety of roads, ranging from rural lanes and urban valleys to 12 lane freeways. The roads contain other traffic, stray animals, pedestrians, police cars and potholes. The taxi driver must also interact with potential and actual customers. Thus, there are a various factos that will come into the picture. Obviously, the more restricted the environment, the easier the design problem.

ACTUATORS:

- Actuator involves knowing what decides the output of a system.
- Tha actuators available to an automated taxi will be more or less the same as those available to a human driver: Control over stering and braking. Also, it will need output to display screen or voice synthesizer to talk back to the passengers, and perhaps some way to communicate with other vehicles politely or otherwise.

SENSORS:

- Sensors involves knowing what provides input to the system.
- For example, to achieve its goals in the driving environment, the taxi will need to know ehre it is, what else is on the road, and how fast it is going. Its basic sensors should therefore include one or more controllable TV cameras, the speedometer, and the odometer. To control the vehicle especially on curves, it should have an accelerometer. Thus in, this way , we need to take into account the sensors.

PEAS DECRIPTORS

AGENT TYPE	PERFORMANCE MEASURE	ENVIRONMENT	ACTUATORS	SENSORS
PART-PICKING ROBOT	Percentage of parts in correct bins	Conveyor belts with parts; bins	Jointed arm and hand	Camera, joint angle sensors
WUMPUS WORLD	gold = +1000 death= -1000 -1 per step -10 for using the arrow	Squares adjacent to wumpus are smelly. Squares adjacent to pit are breezy. Glitter if the gold is in the same square. Shooting kills wumpus if you are facing it. Shooting uses up the only arrow. Grabbing picks up gold if in the same square. Releasing drops the gold in the same square.	Left turn Right turn Forward Grab Release Shoot	Breeze Glitter Smell

6 (b) Explain supervised, unsupervised and reinforcement learning with example. --- 10 Marks

ADD NOTE

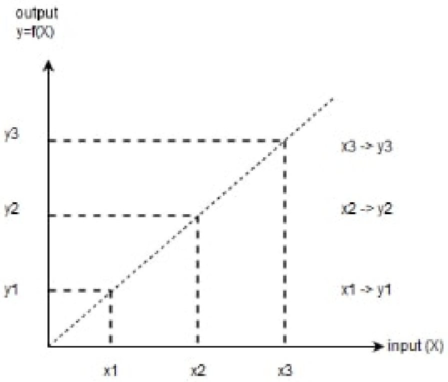
BOOKMARK

This question was repeated in May 15.

Answer:

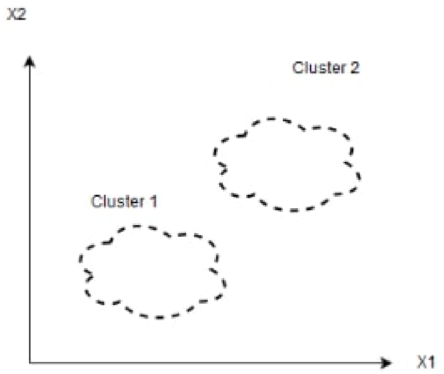
1. Supervised Learning:

- In this learning, critic provides input as well as desired output as an ordered pair
- During the training phase agent learns the relation between input (X) and output y=f(X)
- Once the relation f(X) is learned then testing phase starts.
- Example: Perceptron learning rule.



2. Unsupervised Learning:

- In this method critic provides only input. The desired output is not provided.
- Agent forms clusters from the given input training samples
- The unknown sample is classified into one of the clusters depending on its closeness
- This method is mainly useful for classification type of problems
- Example: Clustering



3. Reinforcement based learning:

- This method of learning is based on bonus/reward or penalty
- If agent action is along the performance measure then critic gives reward or bonus. Agent then learns that the action is to be repeated in future
- If critci gives penalty then agent learns that the action was not proper and should be avoided in future.
- Example: Training a dog

7 (a) PROLOG --- 5 Marks

ADD NOTE BOOKMARK

Answer:

- PROLOG is the short form for 'Programming in Logic'.
- It is used as a programming language for artificial intelligence problem.
- It is a declarative language and different compared to procedural languages like Java,C,C++,etc.
- Instead of giving all statements in program, only rules are provided, which is used to answer any query.
- Program logic is expressed in terms of relations and computation is initiated by running a query over these relations.
- Relations and queries are constructed using prolog's single datatype.
- Datatypes of PROLOG are atom, numbers, variables or compound terms.
- Given a query, PROLOG engine attempts to find resolution refutation of negated query.
- PROLOG is not useful for solving mathematical problems.
- The execution of a PROLOG program can happen in two modes:

    1. INTERPRETATION and
    2. COMPILATION

- **INTERPRETATION:**

    Instead of constructing the list of all possible answers for each subgoal before continuing to the next, PROLOG interpreters generate one answer and a promise to generate the rest when current answers have been fully explored. This promise is called choice point.

- **COMPILATION:**

    At any point in time, every variable in program is either bound or unbound to some value. When a path search fails, PROLOG will back up to a previous choice point and then it may have to unbind some variables. This is done by keeping track of all variables that have been bound in a stack called trail.

7 (b) Belief network --- 5 Marks

ADD NOTE BOOKMARK

This question was repeated in May 15.

Answer:

- A belief network is a directed acyclic graph (DAG) which encodes the casual relationships between particular variables, represented in the DAG as nodes. Nodes are connected by causal links – represented by arrows – which point from parent nodes (causes) to child nodes (effects).
- As an example, a BN diagnosing diseases may have a causal link from the node "cold" to "sneezing". This encodes the causal relationship "sneezing is caused by a cold". A node can have any number of parents; the nodes "hay-fever" and "allergies" may also point to the child node "sneezing".
- Belief networks have been found to be useful in many applications related to reasoning and decision making for some time. Besides more efficient representation and computation, an additional benefit of using belief networks over a joint probability distribution is that information is represented in a more understandable and logical manner, making construction and interpretation much simpler.
- The conditional or posterior probability is used to find the probability distribution of a variable over its state, given some evidence about the state of other variables in the system.
- The syntax used for conditional probability is $P_{(query|evidence)}$;

  $P_{((A=a|B=b))}$ is "the probability of variable A being in state a given the state of B is b".

- The probability distribution of A given the evidence B=b given by $P_{((A|B=b))}$, and the array of conditional probability for all states of A given each state of B is $P_{((A|B))}$ and is known as conditional probability distribution or CPD for the variable A

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$P(A \cap B) = P(A|B)P(B)$$

- Conditional probability can be defined in terms of standard unconditional probabilities using the product rule, as shown in equation above.

7 (c) Forward and inverse kinematics --- 5 Marks

ADD NOTE BOOKMARK

Answer:

Kinematics studies the motion of bodies without consideration of the forces or moments that cause the motion. Robot kinematics refers the analytical study of the motion of a robot manipulator. Robot kinematics can be divided into:

A. Forward kinematics
B. Inverse kinematics.

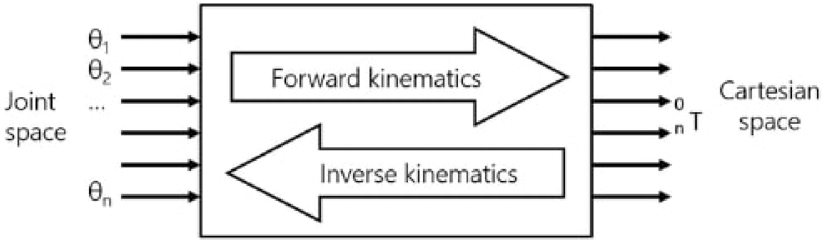
A. Forward kinematics:

- Forward kinematics problem is straightforward and there is no complexity deriving the equations. Hence, there is always a forward kinematics solution of a manipulator.
- A manipulator is composed of serial links which are affixed to each other revolute or prismatic joints from the base frame through the end-effector. Calculating the position and orientation of the end-effector in terms of the joint variables is called as forward kinematics.
- In order to have forward kinematics for a robot mechanism in a systematic manner, one should use a suitable kinematics model.

B. Inverse kinematics

- Inverse kinematics is a much more difficult problem than forward kinematics.
- The solution of the inverse kinematics problem is computationally expansive and generally takes a very long time in the real time control of manipulators.
- Singularities and nonlinearities that make the problem more difficult to solve.
- Two main solution techniques for the inverse kinematics problem are analytical and numerical methods. In the first type, the joint variables are solved analytically according to given configuration data. In the second type of solution the joint variables are obtained based on the numerical techniques.

The relationship between forward and inverse kinematics is illustrated in Figure



The schematic representation of forward and inverse kinematics.

7 (d) Crypt Arithmetic --- *5 Marks*

ADD NOTE BOOKMARK

Answer:

CRYPT ARITHMETIC:

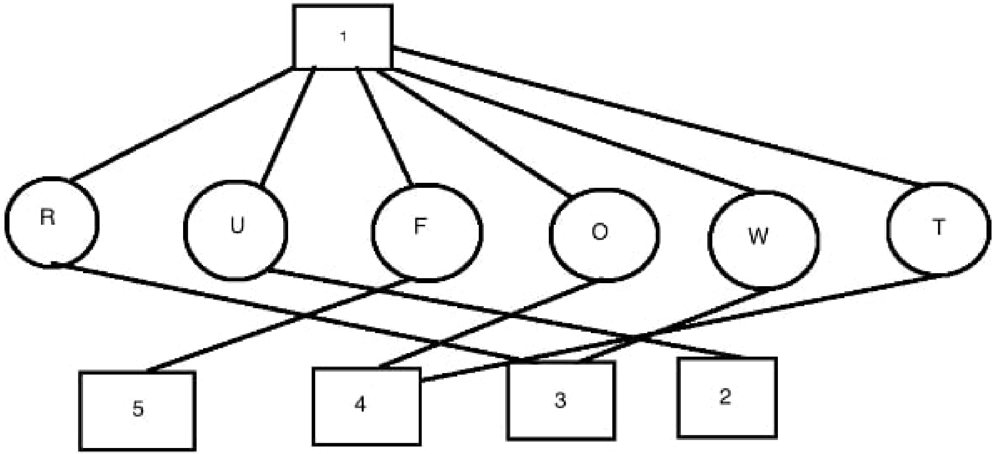
- This technique is used cryptography for encrypting some data.
- Its a type of substitution cipher.
- Alphabets are replaced by numbers, which are assigned to each alphabet in a unique way such that some arithmetic constraint gets satisfied. Consider following arithmetic problem:

			T	W	O
		+	T	W	O
		X3	X2	X1	CARRY BITS
		F	O	U	R

- Here, Arithmetic operation is Addition operation.
- We consider auxiliary variables for carry x1 , x2 , x3 .

Total number of constraints:

1. Unique number of constraints- (T,W,O,F,U,R)
2. O + O = R + 10.x1
3. W + W + x1 = U+ 10.x2
4. T + T + x2 = O + 10.x3
5. x3



7 (e) GPS --- *5 Marks*

ADD NOTE BOOKMARK

Answer:

GPS:

- The Global Positioning System (GPS) is a satellite-based navigation system made up of a network of satellites placed into orbit by the U.S. Department of Defense. It provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites.
- GPS is becoming more common on robots. GPS systems work by receiving signals from satellites orbiting the Earth. The receiver triarngulates itself relative to four GPS satellites computing its position in terms of latitude, longitude, altitude, and change in time.
- GPS isn't a proprioceptive sensor per se since the robot must receive signals from the satellites, external to the robot. However, they are not exteroceptive sensors either, since the robot isn't computing its position relative to its environment.
- Differential GPS involves a second ground receiver with known location, providing millimeter accuracy under ideal conditions.
- However, GPS and DGPS are not complete solutions to the dead reckoning problem in mobile robots for at least two reasons:

i. GPS does not work indoors in most buildings, especially offices or factories with large amounts of steel reinforced concrete.
ii. GPS does not work underwater.

7 (f) Uniform and Inform search. --- *5 Marks*

ADD NOTE BOOKMARK

Answer:

Uninform Search:

- In uninformed or blind search, there is no order in which the solution paths are considered. It uses no domain specific information to search a solution in the search space.
- In a route identifying problem, given various selections of cities to go to next, uniformed search strategies have no way ro choose any specific choices.
- Following are the different uniformed search strategies:
    1. Bread first search
    2. Uniform cost search
    3. Depth first search
    4. Depth liited search
    5. Iterative deepening

Informed Search:

- The informed or heuristic is used for algorithms which find solutions among all possible ones.
- It is a rule of thumb technique that leads to a solution but provides no gurantee of success.
- *Example:* In the route finding problem with a map, if a choice is in the direction of the goal city, prefer it.
    - The informed search have some domain-specific information - guess.
    - We can use this information to speed-up search
    - E.g. The number of tiles that are out of place in an 8-puzzle position.