BHADRAK ENGINEERING SCHOOL & TECHNOLOGY
(BEST), ASURALI, BHADRAK

Artificial Intelligence & Robotics Engineering (Th- 04)

(As per the 2020-21 syllabus of the SCTE&VT, Bhubaneswar, Odisha)

Sixth Semester

Electronics & Tele-comm. Engg.

Prepared By: *Er B.R.Nayak*

CHAPTER-1 Artificial Inteligence

1.1-Defination of AI,Scope of AI-games,theorom proveing, natural language processing, vision and speach processing, robotics,expert systems.

Defination of AI

- **A**rtificial Intelligence (AI) is a branch of *Science* which deals with helping machines find solutions to complex problems in a more human-like fashion.
- This generally involves borrowing characteristics from human intelligence, and applying them as algorithms in a computer friendly way.
- A more or less flexible or efficient approach can be taken depending on the requirements established, which influences how artificial the intelligent behavior appears
- Artificial intelligence can be viewed from a variety of perspectives.
    - ✓ From the perspective of **intelligence** artificial intelligence is making machines "intelligent" -- acting as we would expect people to act.
        - The inability to distinguish computer responses from human responses is called the Turing test.
        - Intelligence requires knowledge
        - Expert problem solving - restricting domain to allow including significant relevant knowledge

Scope of AI

- **Game Playing**

  You can buy machines that can play master level chess for a few hundred dollars. There is some AI in them, but they play well against people mainly through brute force computation--looking at hundreds of thousands of positions. To beat a world champion by brute force and known reliable heuristics requires being able to look at 200 million positions per second.

Speech Recognition

In the 1990s, computer speech recognition reached a practical level for limited purposes. Thus United Airlines has replaced its keyboard tree for flight information by a system using speech recognition of flight numbers and city names. It is quite convenient. On the other hand, while it is possible to instruct some computers using speech, most users have gone back to the keyboard and the mouse as still more convenient.

- **Understanding Natural Language**

  Just getting a sequence of words into a co mputer is not enough. Parsing sentences is not enough either. The computer has to be provided with an understanding of the domain the text is about, and this is presently possible only for very limited domains.

- **Computer Vision**

  The world is composed of three-dimensional objects, but the inputs to the human eye and computers' TV cameras are two dimensional. Some useful programs can work solely in two dimensions, but full computer vision requires partial three-dimensional information that is not just a set of two-dimensional views. At present there are only limited ways of representing three-dimensional information directly, and they are not as good as what humans evidently use.

Theoram proving technique

First-order theorem proving

In the late 1960s agencies funding research in automated deduction began to emphasize the need for practical applications. One of the first fruitful areas was that of program verification whereby first-order theorem provers were applied to the problem of verifying the correctness of computer programs in languages such as Pascal, Ada, etc. Notable among early program verification systems was the Stanford Pascal Verifier developed by David Luckham at Stanford University. This was based on the Stanford Resolution Prover also developed at Stanford using John Alan Robinson's resolution principle. This was the first automated deduction system to demonstrate an ability to solve mathematical problems that were announced in the Notices of the American Mathematical Society before solutions were formally published.

First-order theorem proving is one of the most mature subfields of automated theorem proving. The logic is expressive enough to allow the specification of arbitrary problems, often in a reasonably natural and intuitive way. On the other hand, it is still semi-decidable, and a number of sound and complete calculi have been developed, enabling *fully* automated systems.More expressive logics, such as Higher-order logics, allow the convenient expression of a wider range of problems than first order logic, but theorem proving for these logics is less well developed.

Natural language processing

Natural Language Processing (NLP) makes it possible for computers to understand the human language.

Probably, the most popular examples of NLP in action are virtual assistants, like Google Assist, Siri, and Alexa. NLP understands and translates the human language, like *"Hey Siri, where is the nearest gas station?"* into numbers, making it easy for machines to understand.

Another well-known application of NLP are chatbots, which can help you solve issues while performing

natural language generation – in other words, holding a conversation in plain English!
There are many other everyday apps you use, where you've probably encountered NLP without even noticing. Text recommendations when writing an email, offering to translate a Facebook post written in a different language, or filtering unwanted promotional emails into your spam folder.

Vision and Speech processing

In traditional speech recognition frameworks, many practical complexities need to be dealt with in the case of traditional speech recognition systems. First of all, natural language has various components like accent, semantics, context, and words from foreign languages. Further, the traditional algorithms used to perform speech recognition have limited capabilities and can identify a limited number of words only. These algorithms are not capable of adapting as languages change over time. Finally, the accuracy rate of traditional algorithms is poor, making the speech recognition system unreliable.
With the advent of AI and machine learning (ML) models, the capability of algorithms improved exponentially. ML models can process a much larger dataset with more accuracy as compared to traditional models. Further, the ML models can improve their accuracy and adapt to changes in language on their own, thanks to their self-learning abilities. **Speech to text using AI** has become a rather commonplace service with the increasing application of these models.

Robotics

Robotics is an interdisciplinary field that integrates computer science and engineering.[1] Robotics involves design, construction, operation, and use of robots. The goal of robotics is to design machines that can help and assist humans. Robotics integrates fields of mechanical engineering, electrical engineering, information engineering, mechatronics, electronics, bioengineering, computer engineering, control engineering, software engineering, among others.

Expert Systems

A ``knowledge engineer'' interviews experts in a certain domain and tries to embody their knowledge in a computer program for carrying out some task. How well this works depends on whether the intellectual mechanisms required for the task are within the present state of AI. When this turned out not to be so, there were many disappointing results. One of the first expert systems was MYCIN in 1974, which diagnosed bacterial infections of the blood and suggested treatments. It did better thanmedical students or practicing doctors, provided its limitations were observed. Namely, its ontology included bacteria, symptoms, and treatments and did not include patients, doctors, hospitals, death, recovery, and events occurring in it.

1.2-AI techniques search knowledge abstraction.Problem solving- statespace search production systems.

1. Machine Learning

It is one of the applications of AI where machines are not explicitly programmed to perform certain tasks; rather, they learn and improve from experience automatically. Deep Learning is a subset of machine learning based on artificial neural networks for predictive analysis. There are various machine learning algorithms, such as Unsupervised Learning, Supervised Learning, and Reinforcement Learning. In Unsupervised Learning, the algorithm does not use classified information to act on it without any guidance. In Supervised Learning, it deduces a function from the training data, which consists of a set of an input object and the desired output. Reinforcement learning is used by machines to take suitable actions to increase the reward to find the best possibility which should be taken in to account.

2. NLP (Natural Language Processing)

It is the interactions between computers and human language where the computers are programmed to process natural languages. Machine Learning is a reliable technology for Natural Language Processing to obtain meaning from human languages. In NLP, the audio of a human talk is captured by the machine. Then the audio to text conversation occurs, and then the text is processed where the data is converted into audio. Then the machine uses the audio to respond to humans. Applications of Natural Language Processing can be found in IVR (Interactive Voice Response) applications used in call centres, language translation applications like Google Translate and word processors such as Microsoft Word to check the accuracy of grammar in text. However, the nature of human languages makes the Natural Language Processing difficult because of the rules which are involved in the passing of information using natural language, and they are not easy for the

computers to understand. So NLP uses algorithms to recognize and abstract the rules of the natural languages where the unstructured data from the human languages can be converted to a format that is understood by the computer.

3. Automation and Robotics

The purpose of Automation is to get the monotonous and repetitive tasks done by machines which also improve productivity and in receiving cost-effective and more efficient results. Many organizations use machine learning, neural networks, and graphs in automation. Such automation can prevent fraud issues while financial transactions online by using CAPTCHA technology. Robotic process automation is programmed to perform high volume repetitive tasks which can adapt to the change in different circumstances.

4. Machine Vision

Machines can capture visual information and then analyze it. Here cameras are used to capture the visual information, the analogue to digital conversion is used to convert the image to digital data, and digital signal processing is employed to process the data. Then the resulting data is fed to a computer. In machine vision, two vital aspects are sensitivity, which is the ability of the machine to perceive impulses that are weak and resolution, the range to which the machine can distinguish the objects. The usage of machine vision can be found in signature identification, pattern recognition, and medical image analysis, etc.

Search Knowledge

Knowledge Representation in AI describes the representation of knowledge. Basically, it is a study of how the beliefs, intentions, and judgments of an intelligent agent can be expressed suitably for automated reasoning. One of the primary purposes of Knowledge Representation includes modeling intelligent behavior for an agent.

Knowledge Representation and Reasoning (KR, KRR) represents information from the real world for a computer to understand and then utilize this knowledge to solve complex real-life problems like communicating with human beings in natural language. Knowledge representation in AI is not just about storing data in a database, it allows a machine to learn from that knowledge and behave intelligently like a human being.

The different kinds of knowledge that need to be represented in AI include:

Objects
Events
Performance
Facts
Meta-Knowledge
Knowledge-base

Abstraction of AI

In software engineering and computer science, abstraction is a technique for arranging complexity of computer systems. It works by establishing a level of complexity on which a person interacts with the system, suppressing the more complex details below the current level. The programmer works with an idealized interface (usually well defined) and can add additional levels of functionality that would otherwise be too complex to handle. For an example, a programmer writing code that involves numerical operations may not be interested in the way numbers are represented in the underlying hardware .

1.3-Problem solving,State space search,production systems,Search space control-depth-first,breadth first search,heuristicsearch.Hill climbing,dest-first search,branch and bound.

Problem Solving Techniques

Artificial Intelligence is beneficial for solving complex problems due to its efficient methods of solving. Following are some of the standard problem-solving techniques used in AI:

Heuristics

The heuristic method helps comprehend a problem and devises a solution based purely on experiments and trial and error methods. However, these heuristics do not often provide the best optimal solution to a specific problem. Instead, these undoubtedly offer efficient solutions to attain immediate goals. Therefore, the developers utilize these when classic methods do not provide an efficient solution for the problem. Since heuristics only provide time-efficient solutions and compromise the accuracy, these are combined with optimization algorithms to improve efficiency.

Searching Algorithms

Searching is one of the primary methods of solving any problem in AI. Rational agents or problem-solving agents use these searching algorithms to find optimal solutions. These problem-solving agents are often goal-based and utilize atomic representation. Moreover, these searching algorithms possess completeness, optimality, time complexity, and space complexity properties based on the quality of the solution provided by them.

Types of Searching Algorithms

There are following two main types of searching algorithms:
Informed Search
Uninformed Search

State space search

State space search is a process used in the field of computer science, including artificial intelligence (AI), in which successive configurations or *states* of an instance are considered, with the intention of finding a *goal state* with a desired property.

Problems are often modelled as a state space, a set of *states* that a problem can be in. The set of states forms a graph where two states are connected if there is an *operation* that can be performed to transform the first state into the second.

State space search often differs from traditional computer science search methods because the state space is *implicit*: the typical state space graph is much too large to generate and store in memory. Instead, nodes are generated as they are explored, and typically discarded thereafter. A solution to a combinatorial search instance may consist of the goal state itself, or of a path from some *initial state* to the goal state.

Production System of AI

Production system is a computer programme typically used to provide some form of AI, whuch consists primary link of a set of rules about behavier but it also include the mechanism necessary to follow those rules as the system responds to states of the word.

Componet of production system

The measure components of production system in AI are1-Global database

Search space control :-

Depth First Search (DFS) Algorithm

Depth first search (DFS) algorithm starts with the initial node of the graph G, and then goes to deeper and deeper until we find the goal node or the node which has no children. The algorithm, then backtracks from the dead end towards the most recent node that is yet to be completely unexplored.

The data structure which is being used in DFS is stack. The process is similar to BFS algorithm. In DFS, the edges that leads to an unvisited node are called discovery edges while the edges that leads to an already visited node are called block edges.

Algorithm

Step 1: SET STATUS = 1 (ready state) for each node in G
Step 2: Push the starting node A on the stack and set its STATUS = 2 (waiting state)
Step 3: Repeat Steps 4 and 5 until STACK is empty
Step 4: Pop the top node N. Process it and set its STATUS = 3 (processed state)
Step 5: Push on the stack all the neighbours of N that are in the ready state (whose STATUS = 1) and set their
STATUS = 2 (waiting state)
[END OF LOOP]
Step 6: EXIT

Breadth First Search (BFS) Algorithm

Breadth first search is a graph traversal algorithm that starts traversing the graph from root node and explores all the neighbouring nodes. Then, it selects the nearest node and explore all the unexplored nodes. The algorithm follows the same process for each of the nearest node until it finds the goal.

The algorithm of breadth first search is given below. The algorithm starts with examining the node A and all of its neighbours. In the next step, the neighbours of the nearest node of A are explored and process continues in the further steps. The algorithm explores all neighbours of all the nodes and ensures that each node is visited exactly once and no node is visited twice.

Algorithm

Step 1: SET STATUS = 1 (ready state)
for each node in G
Step 2: Enqueue the starting node A

and set its STATUS = 2
(waiting state)
Step 3: Repeat Steps 4 and 5 until
QUEUE is empty
Step 4: Dequeue a node N. Process it
and set its STATUS = 3
(processed state).
Step 5: Enqueue all the neighbours of
N that are in the ready state
(whose STATUS = 1) and set
their STATUS = 2
(waiting state)
[END OF LOOP]
Step 6: EXIT

Heuristic search

Heuristic search refers to a search strategy that attempts to optimize a problem by iteratively improving the solution based on a given heuristic function or a cost measure. A heuristic search method does not always guarantee to find an optimal or the best solution, but may instead find a good or acceptable solution within a reasonable amount of time and memory space. Several commonly used heuristic search methods include hill climbing methods, the best-first search

Hill climbing

Hill Climbing is a heuristic search used for mathematical optimization problems in the field of Artificial Intelligence.

Given a large set of inputs and a good heuristic function, it tries to find a sufficiently good solution to the problem. This solution may not be the global optimal maximum.

Features of Hill Climbing

1. Variant of generate and test algorithm : It is a variant of generate and test algorithm. The generate and test algorithm is as follows :

i. Generate possible solutions.

ii. Test to see if this is the expected solution.

iii. If the solution has been found quit else go to step 1.

Hence we call Hill climbing as a variant of generate and test algorithm as it takes the feedback from the test procedure. Then this feedback is utilized by the generator in deciding the next move in search space.

2. Uses the Greedy approach : At any point in state space, the search moves in that direction only which optimizes the cost of function with the hope of finding the optimal solution at the end.

Best first search

Best first search is a traversal technique that decides which node is to be visited next by checking which node is the most promising one and then check it. For this it uses an evaluation function to decide the traversal.

This best first search technique of tree traversal comes under the category of heuristic search or informed search technique.

The cost of nodes is stored in a priority queue. This makes implementation of best-first search is same as that of breadth First search. We will use the priorityqueue just like we use a queue for BFS.

Algorithm for implementing Best First Search

Step 1 : Create a priorityQueue pqueue.

Step 2 : insert ‘start’ in pqueue : pqueue.insert(start)

Step 3 : delete all elements of pqueue one by one.

Step 3.1 : if, the element is goal . Exit.

Step 3.2 : else, traverse neighbours and mark the node examined.

Step 4 : End.

Branch and bound

Branch and bound is an algorithm design paradigm which is generally used for solving combinatorial optimization problems. These problems are typically exponential in terms of time complexity and may require exploring all possible permutations in worst case. The Branch and Bound Algorithm technique

solves these problems relatively quickly.

1.4 Knowledge representation-Predicate logic,unification,Modus pones, resolution

Knowledge Representation

Knowledge Representation in AI describes the representation of knowledge. Basically, it is a study of how the beliefs, intentions, and judgments of an intelligent agent can be expressed suitably for automated reasoning. One of the primary purposes of Knowledge Representation includes modeling intelligent behavior for an agent.

Knowledge Representation and Reasoning (KR, KRR) represents information from the real world for a computer to understand and then utilize this knowledge to solve complex real-life problems like communicating with human beings in natural language. Knowledge representation in AI is not just about storing data in a database, it allows a machine to learn from that knowledge and behave intelligently like a human being.

The different kinds of knowledge that need to be represented in AI include:

- Objects
- Events
- Performance
- Facts
- Meta-Knowledge
- Knowledge-base

Predicate Logic

Predicate Logic deals with predicates, which are propositions, consist of variables.

Predicate Logic - Definition

A predicate is an expression of one or more variables determined on some specific domain. A predicate with variables can be made a proposition by either authorizing a value to the variable or by quantifying the variable.

The following are some examples of predicates.

Consider E(x, y) denote "x = y"

Consider X(a, b, c) denote "a + b + c = 0"

Consider M(x, y) denote "x is married to y."

Quantifier:

The variable of predicates is quantified by quantifiers. There are two types of quantifier in predicate logic - Existential Quantifier and Universal Quantifier.

Existential Quantifier:

If p(x) is a proposition over the universe U. Then it is denoted as ∃x p(x) and read as "There exists at least one value in the universe of variable x such that p(x) is true. The quantifier ∃ is called the existential quantifier.

There are several ways to write a proposition, with an existential quantifier, i.e.,

(∃x∈A)p(x) or ∃x∈A such that p (x) or (∃x)p(x) or p(x) is true for some x ∈A.

Universal Quantifier:

If p(x) is a proposition over the universe U. Then it is denoted as ∀x,p(x) and read as "For every x∈U,p(x) is true." The quantifier ∀ is called the Universal Quantifier.

There are several ways to write a proposition, with a universal quantifier.

∀x∈A,p(x) or p(x), ∀x ∈A Or ∀x,p(x) or p(x) is true for all x ∈A.

Negation of Quantified Propositions:

When we negate a quantified proposition, i.e., when a universally quantified proposition is negated, we obtain an existentially quantified proposition,and when an existentially quantified proposition is negated, we obtain a universally quantified proposition.

The two rules for negation of quantified proposition are as follows. These are also called DeMorgan's Law.

Example: Negate each of the following propositions:

1.∀x p(x)A ∃ y q(y)
Sol: ~.∀x p(x)A ∃ y q(y))
≅~∀ x p(x)∨~∃yq (y) (∴~(pAq)=~p∨~q)
≅ ∃ x ~p(x)∨∀y~q(y)
2. (∃x∈U) (x+6=25)
Sol: ~(∃ x∈U) (x+6=25)
≅∀ x∈U~ (x+6)=25
≅(∀ x∈U) (x+6)≠25
3. ~(∃ x p(x)∨∀ y q(y)
Sol: ~(∃ x p(x)∨∀ y q(y))
≅~∃ x p(x)A~∀ y q(y) (∴~(p∨q)= ~pA~q)
≅ ∀ x ~ p(x)A∃y~q(y))

Propositions with Multiple Quantifiers:

The proposition having more than one variable can be quantified with multiple quantifiers. The multiple universal quantifiers can be arranged in any order without altering the meaning of the resulting proposition. Also, the multiple existential quantifiers can be arranged in any order without altering the meaning of the proposition.

The proposition which contains both universal and existential quantifiers, the order of those quantifiers can't be exchanged without altering the meaning of the proposition, e.g., the proposition ∃x ∀ y p(x,y) means "There exists some x such that p (x, y) is true for every y."

Example: Write the negation for each of the following. Determine whether the resulting statement is true or false. Assume U = R.

1.∀ x ∃ m(x2<m)

Sol: Negation of ∀ x ∃ m(x2<m) is ∃ x ∀ m (x2≥m). The meaning of ∃ x ∀ m (x2≥m) is that there exists for some x such that x2≥m, for every m. The statement is true as there is some greater x such that x2≥m, for every m.

2. ∃ m∀ x(x2<m)

Sol: Negation of ∃ m ∀ x (x2<m) is ∀ m∃x (x2≥m). The meaning of ∀ m∃x (x2≥m) is that for every m, there exists for some x such that x2≥m. The statement is true as for every m, there exists for some greater x such that x2≥m.

Unification

- Unification is a process of making two different logical atomic expressions identical by finding a substitution. Unification depends on the substitution process.
- It takes two literals as input and makes them identical using substitution.
- Let Ψ_1 and Ψ_2 be two atomic sentences and σ be a unifier such that, $\boldsymbol{\Psi_1}\boxtimes = \boldsymbol{\Psi_2}\boxtimes$, then it can be expressed as UNIFY(Ψ_1, Ψ_2).
- Example: Find the MGU for Unify{King(x), King(John)}

Conditions for Unification:

Following are some basic conditions for unification:

- Predicate symbol must be same, atoms or expression with different predicate symbol can never be unified.
- Number of Arguments in both expressions must be identical.
- Unification will fail if there are two similar variables present in the same expression.

Modus ponens

In propositional logic, modus ponens (/ˈmoʊdəs ˈpoʊnɛnz/; MP), also known as modus ponendo ponens (Latin for "mode that by affirming affirms")[1] or implication elimination or affirming the antecedent,[2] is a deductive argument form and rule of inference.[3] It can be summarized as "P implies Q. P is true. Therefore Q must also be true."

Modus ponens is closely related to another valid form of argument, modus tollens. Both have apparently similar but invalid forms such as affirming the consequent, denying the antecedent, and evidence of absence. Constructive dilemma is the disjunctive version of modus ponens. Hypothetical syllogism is closely related to modus ponens and sometimes thought of as "double modus ponens."

Resolution

Resolution is a theorem proving technique that proceeds by building refutation proofs, i.e., proofs by contradictions. It was invented by a Mathematician John Alan Robinson in the year 1965.
Resolution is used, if there are various statements are given, and we need to prove a conclusion of those statements. Unification is a key concept in proofs by resolutions. Resolution is a single inference rule which can efficiently operate on the **conjunctive normal form or clausal form**.

Steps for Resolution:

Conversion of facts into first-order logic.

- Convert FOL statements into CNF
- Negate the statement which needs to prove (proof by contradiction)
- Draw resolution graph (unification).
- To better understand all the above steps, we will take an example in which we will apply resolution.

1.5-Structured knowledge representation:Semantic nets:slots Exceptions and defults frames,conceptual dependencey,scripts.

Structured Knowledge Representation

- Representation and Mapping - Mapping is the process that maps facts to representations and vice versa. - Forward mapping : facts to representation - Backward mapping : representation to facts Facts Internal Representation English Representation English GenerationEnglish Understanding
- Properties of Good Knowledge Representation 1. Represential Adequacy : Ability to represent all knowledge needed in the domain 2. Inferential Adequacy : Ability to manipulate knowledge to derive new structures inferred from old 3. Inferential Efficiency : Ability to perform inference in the most efficient directions 4. Acquisitional Efficiency : Ability to acquire new information easily.

A semantic network

A semantic network, or frame network is a knowledge base that represents semantic relations between concepts in a network. This is often used as a form of knowledge representation. It is a directed or undirected graph consisting of vertices, which represent concepts, and edges, which represent semantic relations between concepts,[1] mapping or connecting semantic fields. A semantic network may be instantiated as, for example, a graph database or a concept map.

Slots

A frame has associated with it a set of own slots, and each own slot of a frame has associated with it a set of entities called slot values. Formally, a slot is a binary relation, and each value V of an own slot S of a frame F represents the assertion that the relation S holds for the entity represented by F and the entity represented by V (i.e., (S F V)). For example, the assertion that Fred's favorite foods are potato chips and ice cream could be represented by the own slot Favorite-Food of the frame Fred having as values the frame Potato-Chips and the string ``ice cream".

Exceptions and defult frames

Frames

Frames are an artificial intelligence data structure used to divide knowledge into substructures by representing "stereotyped situations". They were proposed by Marvin Minsky in his 1974 article "A Framework for Representing Knowledge". Frames are the primary data structure used in artificial intelligence frame language; they are stored as ontologies of sets.
Frames are also an extensive part of knowledge representation and reasoning schemes.

Frame Structure

The frame contains information on how to use the frame, what to expect next, and what to do when these expectations are not met. Some information in the frame is generally unchanged while other information, stored in "terminals", usually change. Terminals can be considered as variables. Top level frames carry information, that is always true about the problem in

hand, however, terminals do not have to be true. Their value might change with the new information encountered. Different frames may share the same terminals.
Each piece of information about a particular frame is held in a slot. The information can contain:
Facts or Data
Values (called facets)
Procedures (also called procedural attachments)
IF-NEEDED : deferred evaluation
IF-ADDED : updates linked information
Default Values
For Data
For Procedures
Other Frames or Subframes

Conceptual dependency

Conceptual dependency theory is a model of natural language understanding used in artificial intelligence systems.
Roger Schank at Stanford University introduced the model in 1969, in the early days of artificial intelligence. This model was extensively used by Schank's students at Yale University such as Robert Wilensky, Wendy Lehnert, and Janet Kolodner.
Schank developed the model to represent knowledge for natural language input into computers. Partly influenced by the work of Sydney Lamb, his goal was to make the meaning independent of the words used in the input, i.e. two sentences identical in meaning, would have a single representation. The system was also intended to draw logical inferences
The model uses the following basic representational tokens: real world objects, each with some attributes. real world actions, each with attributes
times locations
A set of conceptual transitions then act on this representation, e.g. an ATRANS is used to represent a transfer such as "give" or "take" while a PTRANS is used to act on locations such as "move" or "go". An MTRANS represents mental acts such as "tell", etc.
A sentence such as "John gave a book to Mary" is then represented as the action of an ATRANS on two real world objects John and Mary.

Script

A script is a structured representation describing a stereotyped sequence of events in a particular context. Scripts are used in natural language understanding systems to organize a knowledge base in terms of the situations that the system should understand. Scripts use a frame-like structure to represent the commonly occurring experience like going to the movies eating in a restaurant, shopping in a supermarket, or visiting an ophthalmologist.
Thus, a script is a structure that prescribes a set of circumstances that could be expected to follow on from one another.

1.6-Concept of learning,learning automation,Genetic AlgorithmConcept of learning

Learning is one of the fundamental building blocks of artificial intelligence (AI) solutions.
From a conceptual standpoint, learning is a process that improves the knowledge of an AI program by making observations about its environment. From a technical/mathematical standpoint, AI learning processes focused on processing a collection of input-output pairs for a specific function and predicts the outputs for new inputs. Most of the artificial intelligence(AI) basic literature identifies two main groups of learning models: supervised and unsupervised. However, that classification is an oversimplification of real world AI learning models and techniques.
To understand the different types of AI learning models, we can use two of the main elements of human learning processes: knowledge and feedback. From the knowledge perspective, learning models can be classified based on the representation of input and output data points. In terms of the feedback, AI learning models can be classified based on the interactions with the outside environment and user.

Learning Automation

Artificial intelligence (AI) is sometimes confused with automation, and the terms are often used interchangeably. When talking to vendors, it's critical for digital marketers to decipher exactly what's being offered so you know it will truly address your needs. Today, let's demystify these terms and discover what happens when the two are combined.

Robotic Process Automation (RPA) or marketing automation software is great for simple activities and repetitive tasks that follow instructions or workflows set by individuals. It is best suited for highly repetitive and predictable tasks. Automated tools require manual configuration and human supervision to effectively execute campaigns. The trick with robotic process automation or marketing automation is for humans to anticipate every permutation so the machine is programmed to behave the right way every time. This is why constant vigilance is required. If the environment changes, marketers must manually step in and make the necessary adjustments.
AI refers to how computer systems can use huge amounts of data to imitate human intelligence and reasoning, allowing the system to learn, predict and recommend what to do next.
Capabilities enabled by AI include natural language processing (NLP), computer vision, and facial recognition.
Machine learning and deep learning (in the form of neural networks) power the most sophisticated AI available today.
An AI capable of understanding marketing KPIs can use various algorithms that act in concert to find signal in the noise of data and find paths to solutions that no human would be capable of. Most AI that uses machine learning today works in an assistive fashion, providing next best action recommendations to humans who then decide whether to trust them or not and then manually make adjustments.

Genetic algorithms

Genetic algorithms (GA) work by simulating the logic of Darwinian selection, where only the best are selected for replication. Over many generations, natural populations evolve according to the principles of natural selection and stated by Charles Darwin in The Origin of Species. Only the most suited elements in a population are likely to survive and
generate offspring, thus transmitting their biological heredity to new generations. Genetic algorithms are able to address complicated problems with many variables and a large number of possible outcomes by simulating the evolutionary process of “survival of the fittest” to reach a defined goal. They operate by generating many random answers to a problem, eliminating the worst and cross-pollinating better answers. Repeating this
elimination and regeneration process gradually improves the quality of the answers to an optimal or near-optimal condition.
In computing terms, a genetic algorithm implements the model of computation by having arrays of bits or characters (binary string) to represent the chromosomes. Each string represents a potential solution. The genetic algorithm then manipulates the most promising chromosomes searching for improved solutions. A genetic algorithm operates through a cycle of three stages:

- Build and maintain a population of solutions to a problem
- Choose the better solutions for recombination with each other
- Use their offspring to replace poorer solutions.

If the GA has been correctly implemented, the population will evolve over successive generations so that the fitness of the best and the average individual in each generation increases towards the global optimum.
The basic Genetic algorithm is :
Initialize a random population of individuals Compute fitness of each individual
WHILE NOT finished DO
BEGIN /* produce new generation */ FOR population_size DO
BEGIN /* reproductive cycle */

Select two individuals from old generation, recombine the two individuals to give two offspring
Make a mutation for selected individuals by altering a random bit in a string
Create a new generation (new populations) END
IF population has converged THEN finished := TRUE
END

Learning by induction

Inductive learning also called Concept Learning is a way in which AI systems try to use a generalized rule to carry out observations.....The goal of inductive learning is to learn the function for new data when the output and the examples of the function are input into the AI system.

Example

Induction starts with the specifics and then draws the general conclusion based on the specific facts. Examples of Induction: I have seen four students at this school leave trash on the floor. The students in this school are disrespectful.

Neural nets

Neural nets are a means of doing machine learning, in which a computer learns to perform some task by analyzing training examples Most of today's neural nets are organized into layers of nodes, and they're "feed-forward," meaning that data moves through them in only one direction.

A neural network (NN), in the case of artificial neurons called artificial neural network (ANN) or simulated neural network (SNN), is an interconnected group of natural or artificial neurons that uses a mathematical or computational model for information processing based on a connectionistic approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network.

In more practical terms neural networks are non-linear statistical data modeling or decision making tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. An artificial neural network involves a network of simple processing elements (artificial neurons) which can exhibit complex global behavior, determined by the connections between the processing elements and element parameters. Artificial neurons were first proposed in 1943 by Warren McCulloch, a neurophysiologist, and Walter Pitts, a logician, who first collaborated at the University of Chicago.[17]

One classical type of artificial neural network is the recurrent Hopfield network.

The concept of a neural network appears to have first been proposed by Alan Turing in his 1948 paper Intelligent Machinery in which he called them "B-type unorganised machines".[18]

The utility of artificial neural network models lies in the fact that they can be used to infer a function from observations and also to use it. Unsupervised neural networks can also be used to learn representations of the input that capture the salient characteristics of the input distribution, e.g., see the Boltzmann machine (1983), and more recently, deep learning algorithms, which can implicitly learn the distribution function of the observed data. Learning in neural networks is particularly useful in applications where the complexity of the data or task makes the design of such functions by hand impractical.

Applications

Neural networks can be used in different fields. The tasks to which artificial neural networks are applied tend to fall within the following broad categories:

Function approximation, or regression analysis, including time series prediction and modeling.

Classification, including pattern and sequence recognition, novelty detection and sequential decision making.

Data processing, including filtering, clustering, blind signal separation and compression.

Application areas of ANNs include nonlinear system identification[19] and control (vehicle control, process control), game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems, face identification, object recognition), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications, data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering. For example, it is possible to create a semantic profile of user's interests emerging from pictures trained for object recognitio

Short question with answer.

Q 1- Write deffination of AI.
Artificial intelligence (AI) is the ability of a computer or a robot controlled by a computer to do tasks that are usually done by humans because they require human intelligence and discernment.

Q 2-Write different scope of AI.

- AI in Science and Research. AI is making lots of progress in the scientific sector. ...
- AI in Cyber Security. Cybersecurity is another field that's benefitting from AI. ...
- AI in Data Analysis. Data analysis can benefit largely from AI and ML. ...
- AI in Transport. ...
- AI in Home. ...
- AI in Healthcare.

Q 3- Define robotics.
Robotics, design, construction, and use of machines (robots) to perform tasks done traditionally by human beings ..... Robots are widely used in such industries as automobile manufacture to perform simple repetitive tasks, and in industries where work must be performed in environments hazardous to humans.

Q 4-Write different technique of AI.

Different technique of AI are-
1. Machine Learning
2. NLP (Natural Language Processing)
3. Automation and Robotics
4. Machine Vision

Q 5-Define DFS.
Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking.

Q 6- Define BFS.
Breadth-first search (BFS) is an algorithm for traversing or searching tree or graph data structures. It starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a 'search key'[1]), and explores all of the neighbor nodes at the present depth prior to moving on to the nodes at the next depth level.

Q 7-Define heurostic search

A Heuristic is a technique to solve a problem faster than classic methods, or to find an approximate solution when classic methods cannot. This is a kind of a shortcut as we often trade one of optimality, completeness, accuracy, or precision for speed.

Q 8-Define Best first search

A search is the most commonly known form of best-first search. It uses heuristic function h(n), and cost to reach the node n from the start state g(n). It has combined features of UCS and greedy best-first search, by which it solve the problem efficiently.

Q 9-Define Hill climbing techineque

Hill Climbing Algorithm in Artificial Intelligence. Hill climbing algorithm is a local search algorithm which continuously moves in the direction of increasing elevation/value to find the peak of the mountain or best solution to the problem ..... Hill Climbing is mostly used when a good heuristic is available.

Q 10-define predicate logic.

In logic and philosophy, predicate logic is a system of mathematical logic. It uses predicates to express the state of certain things, which are "incomplete propositions" with a placeholder for objects or subjects that must be inserted in order to obtain a valid proposition.

Q 11-Define semintic nets.
A semantic network is used when one has knowledge that is best understood as a set of concepts that are related to one another. Most semantic networks are cognitively based. They also consist of arcs and nodes which can be organized into a taxonomic hierarchy.

Q 12- Define slots.
A slot is a narrow opening in a machine or container, for example, a hole that you put coins in to make a machine work. He dropped a coin into the slot and dialed. Synonyms: opening, hole, groove, vent More Synonyms of slot. 2. transitive verb/intransitive verb.

Q 13-Define defult frames.
Vision draws a line between the first natural join (columns with the same name, data type and length) shared by the tables .....You cannot display these columns as fields on the form, because they are duplicates of the same columns in the Master table.

Q14-Define genetic algorithem.
A genetic algorithm (GA) is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution.

Q 15-Define neural nets
A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature.

Long questions
1- Explain scope of AI in details.
2- Explain about theorom proving techineque.
3- Explain DFS and BFS brifly and differianciate between them.
4- Explain The hill climbing techineque and best first search techeneque.
5-Expalin knowledge representation in AI.
6-Explain about predicate logic with algorithem.

CHAPTER-2

Introduction to Robotics

2.1-Types and components of a robot,classification of robots,close loop and open loop control systems.

Types and componets of a robot:

1. Manipulator:

Just like the human arm, the robot consists of what is called a manipulator having several joints and links.

2. Endeffector:

The base of the manipulator is fixed to base support and at its other free end, the Endeffector is attached. The Endeffector is expected to perform tasks normally performed by the palm and finger arrangements of the human arm.

3. The Locomotion Device:

In the case of Human Beings the power for the movement of the arm, the palm and fingers is provided by muscles. For the robot the power for the movement (locomotion) is provided by the motors. The motors used for providing locomotion in robots are of three types depending on the source of energy: Electric, Hydraulic or Pneumatic.

4. The Controller:

The digital computer (both the hardware and the software) acts as a controller to the robot. The controller functions in a manner analogous to the human brain. With the help of this controller, the robot is able to carry out the assigned tasks. The controller directs and controls the movement of the Manipulator and the Endeffector. In other words, the controller controls the robot.

5. The Sensors:

Without the data supplied by the sense organs, the brain would be incapable of intelligence. In other words the controller (the computer) of the robot cannot do any meaningful task, if the robot is not with a component analogous to the sense organs of the human body. Thus, the fifth and the most important component of the robot is the set of sensors. Sensors are nothing but measuring instruments which measures quantities such as position, velocity, force, torque, proximity, temperature, etc.

Classification of robot:

The robot is a computer programmable machine that can carry out complex actions automatically. It can be guided and controlled either by an external control device or by the controller embedded within the machine. The classification of robots can be done on various criteria such as their power source, size of the robot, type of drive system used etc.

Classification of robots based on the power source

On the basis of the power source, the robots can be classified into 5 major divisions namely electrical, hydraulic, pneumatic, nuclear, and green.

Electrical power source

Robots operating with the electrical power source can further be subdivided as AC or DC systems. Direct current systems usually provide greater torque but they often require more maintenance for the motors. The use of motors generates dust and spark that can create hazards to the process.

DC systems are common for the hobby robotics world as those systems are usually mobile, battery-

powered robots.
AC powered robots are common in industries and these often use Servo motors. Stepper motors are also used for these systems.

Hydraulic Power Source

Hydraulic power generates a large amount of force and it is used for heavy loads in robotics. The system uses some other form of energy for generating hydraulic pressure. The robot uses this hydraulic force for performing its tasks.
However, due to the improvements in servo motors, the hydraulic powered robots are losing ground.
Hydraulic robots have some drawbacks such as-as a hydraulic leak, fire hazard, increased noise, increased maintenance and the cost of oil.

Pneumatic Power Source

Pneumatic robots are powered by compressed air or compressed inert gases. These are used for high speed and high load carrying capabilities.
These systems are very fast and the industries use them as a ready supply of cheap pneumatic pressure.
However, the biggest problem with these robots is the difficulty in maintaining their position. This is due to the fact that gas is compressible, and stopping it mid stroke leads to drifting.
The only way to hold its position is to use hardstop and constant pressure.
Pneumatic robots also suffer from the issue of noise and leaks.

Nuclear Power Source

Nuclear-powered robots used their own nuclear reactor that is smaller than the nuclear reactors of nuclear power plants or submarines.
Nuclear powered robots are used by space agencies such as NASA for deep space exploration.
Nuclear powered robots run for years and decades without the need for human interaction which makes them perfect fit for the space missions.
However, if these robots are used on earth, there will be the need for proper disposal of nuclear material after the fuel is completely spent.

Green Power Source

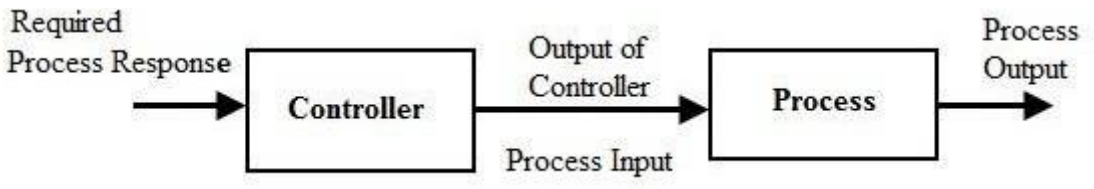
Green Power source refers to a wide variety of power sources that have the commonality of easy replacement without any negative ecological impact.
The potential green power sources for powering the robots include solar power, wind power, organic sources, natural heat sources etc.

Closed loop and open loop control system

The behavior of the system can be determined with the help of a differential equation is known as the control system. So it controls different devices as well as systems with the help of control loops. Control systems are classified into two types like open loop and closed loop.

Open Loop Control System

In this kind of control system, the output doesn't change the action of the control system otherwise; the working of the system which depends on time is also called the open-loop control system. It doesn't have any feedback. It is very simple, needs low maintenance, quick operation, and cost-effective. The accuracy of this system is low and less dependable. The example of the open-loop type is shown below.
The main advantages of the open-loop control system are easy, needs less protection; operation of this system is fast & inexpensive and the disadvantages are, it is reliable and has less accuracy.



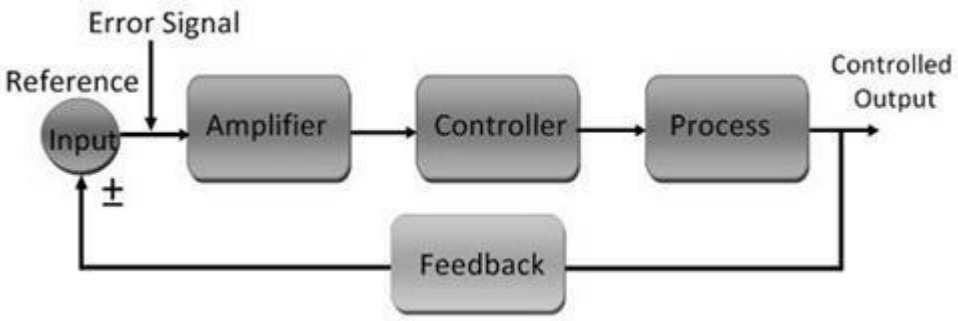
Open Loop Control System

Example

The clothes dryer is one of the examples of the open-loop control system. In this, the control action can be done physically through the operator. Based on the clothing's wetness, the operator will fix the timer to 30 minutes. So after that, the timer will discontinue even after the machine clothes are wet.
The dryer in the machine will stop functioning even if the preferred output is not attained. This displays that the control system doesn't feedback. In this system, the controller of the system is the timer.

Closed-Loop Control System

The closed-loop control system can be defined as the output of the system that depends on the input of the system. This control system has one or more feedback loops among its input & output. This system provides the required output by evaluating its input. This kind of system produces the error signal and it is the main disparity between the output and input of the system.



Closed-Loop Control System

The main advantages of the closed-loop control system are accurate, expensive, reliable, and requires high maintenance.

Example

The best example of the closed-loop control system is AC or air conditioner. The AC controls the temperature by evaluating it with the nearby temperature. The evaluation of temperature can be done through the thermostat. Once the air conditioner gives the error signal is the main difference between the room and the surrounding temperature. So the thermostat will control the compressor.
These systems are accurate, expensive, reliable, and requires high maintenance.

2.2-Kinematics systems;Deffination of mechanisms and manipulators,social issues safety

Robot kinematics studies the relationship between the dimensions and connectivity of kinematic chains and the position, velocity and acceleration of each of the links in the robotic system, in order to plan and control movement and to compute actuator forces and torques. The relationship between mass and inertia properties, motion, and the associated forces and torques is studied as part of robot dynamics.

Forward kinematics

Forward kinematics specifies the joint parameters and computes the configuration of the chain. For serial manipulators this is achieved by direct substitution of the joint parameters into the forward kinematics equations for the serial chain. For parallel manipulators substitution of the joint parameters into the kinematics equations requires solution of the a set of polynomial constraints to determine the set of possible end-effector locations.

Inverse kinematics

Robot Jacobian
The time derivative of the kinematics equations yields the Jacobian of the robot, which relates the joint rates to the linear and angular velocity of the end-effector. The principle of virtual work shows that the Jacobian also provides a relationship between joint torques and the resultant force and torque applied by the end-effector. Singular configurations of the robot are identified by studying its Jacobian.
Velocity kinematics

The robot Jacobian results in a set of linear equations that relate the joint rates to the six-vector formed from the angular and linear velocity of the end-effector, known as a twist. Specifying the joint rates yields the end-effector twist directly.
The inverse velocity problem seeks the joint rates that provide a specified end-effector twist. This is solved by inverting the Jacobian matrix. It can happen that the robot is in a configuration where the Jacobian does not have an inverse. These are termed singular configurations of the robot.

Defination of mechanism and menipulator

Mechanism

An assembly of moving parts performing a complete functional motion, often being part of a large machine; linkage.
The agency or means by which an effect is produced or a purpose is accomplished.
Machinery or mechanical appliances in general.
The structure or arrangement of parts of a machine or similar device, or of anything analogous.
The mechanical part of something; any mechanical device:the mechanism of a clock.

Manipulator

In robotics, a manipulator is a device used to manipulate materials without direct physical contact by the operator. The applications were originally for dealing with radioactive or biohazardous materials, using robotic arms, or they were used in inaccessible places. In more recent developments they have been used in diverse range of applications including welding automation,[1] robotic surgery and in space. It is an arm-like mechanism that consists of a series of segments, usually sliding or jointed called cross-slides, which grasp and move objects with a number of degrees of freedom.

Social Issue and safety

Legislation

Human societies are regulated by bodies of legislation. While remaining within the academic realm, AI and ML developments have stayed fairly oblivious to legal concerns, but the moment these technologies start occupying the social space at large, their impact on people is likely to hit a few legal walls.

Interpretability and Explainability

Biological brains have not necessarily evolved the means to explain themselves. Arguably, this has only happened in species with social behaviour (although it could also be argued that social behaviour can only happen in species whose brains are capable of explaining themselves through some form of communication). In the human species, natural language performs that communicative or explanatory function.

Privacy and Anonymity

Technological advances and the widespread adoption of networked computing and telecommunication systems are flooding our societies (and mostly governments and technology providers) with data. The physical society bonds are being swiftly amplified by our use of virtual social networks.

2.3-Robort anatomy-Link-joints and joint national scheme-degrees of freedom(DOF)

Robort anatomy-Link-joints and joint national scheme

The anatomy of robot is also known as structure of robot. The basic components or sections in anatomy of robots are as follows.
The RIA (Robotics Industries Association) has officially given the definition for Industrial Robots *An .Industrial* According *Robot* is are programmable, to RIA, multifunctional " manipulator designed to move materials, parts, tools, or special devices through variable programmed motions for the performance of a

The *Anatomy* of Industrial Robots deals with the assembling of outer components of a robot such as wrist, arm, and body. Before jumping into Robot Configurations, here are some of the key facts about robot anatomy.

- ***End Effectors*:** A hand of a robot is considered as end effectors. ***The*** grippers and tools are the two significant types of end effectors. The grippers are used to pick and place an object, while the tools are used to carry out operations like spray painting, spot welding, etc. on a work piece.

- ***Robot Joints*:** The joints in an industrial robot are helpful to perform sliding and rotating movements of a component.

- ***Manipulator*:** The manipulators in a robot are developed by the integration of links and joints. In the body and arm, it is applied for moving the tools in the work volume. It is also used in the wrist to adjust the tools.

- ***Kinematics*:** It concerns with the assembling of robot links and joints. It is also used to illustrate the robot motions.

Degree of freedoms(DOF)

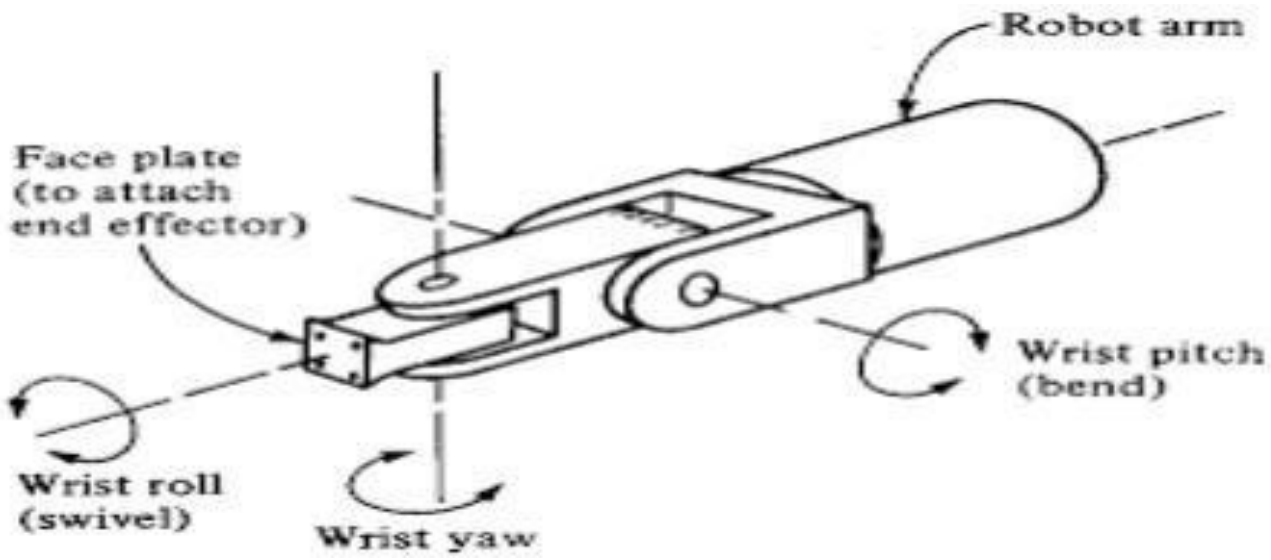
In physics, the degrees of freedom (DOF) of a mechanical system is the number of independent parameters that define its configuration or state. It is important in the analysis of systems of bodies in mechanical engineering, structural engineering, aerospace engineering, robotics, and other fields.

The position of a single railcar (engine) moving along a track has one degree of freedom because the position of the car is defined by the distance along the track. A train of rigid cars connected by hinges to an engine still has only one degree of freedom because the positions of the cars behind the engine are constrained by the shape of the track.

Six degrees of freedom (6 DOF)

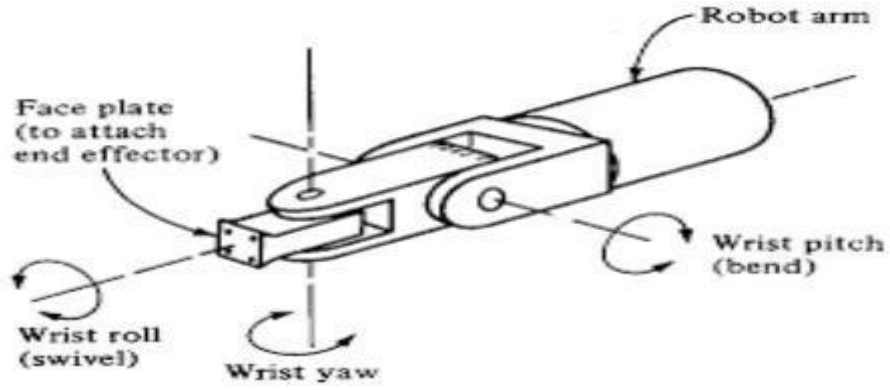
- Moving up and down (elevating/heaving);
- Moving left and right (strafing/swaying);
- Moving forward and backward (walking/surging);
- Swivels left and right (yawing);
- Tilts forward and backward (pitching);
- Pivots side to side (rolling).

Wrist configuration



Roll- This is also called wrist swivel, this involves rotation of the wrist mechanism about the arm axis.

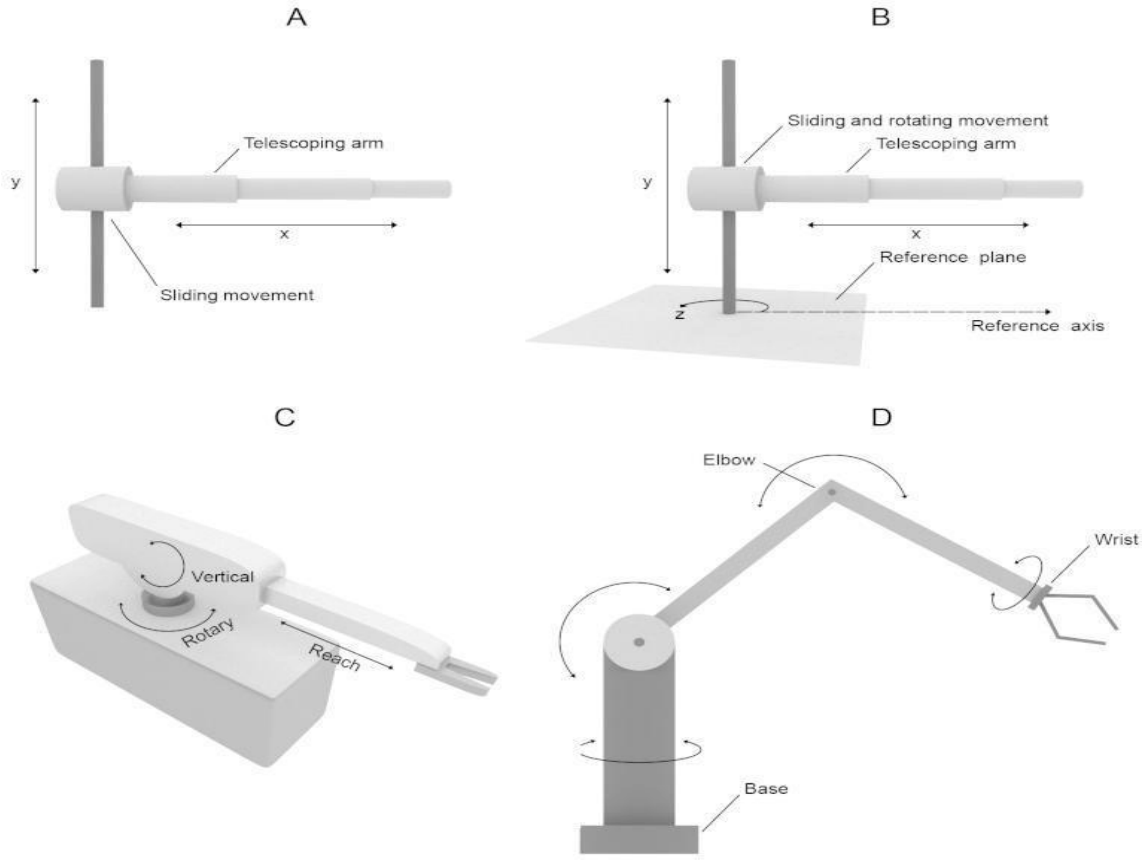
Roll- This is also called wrist swivel, this involves rotation of the wrist mechanism about the arm axis.
Pitch- It involves up & down rotation of the wrist. This is also called as wrist bend.
Yaw- It involves right or left rotation of the wrist.
Notation TRL:
f Consists of a sliding armbody,which(Lcan joint)rotateaboutboth actuat vertical axis (T joint) and horizontal axis (R joint)



Notation TLO:
*f*Consists of a vertical column, relative to which an arm assembly is moved up or down *f*the arm can be moved in or out relative to the column
Notation LOO:
f Consistsiding joints,of twothreeofwhich areslorthogonal Other names include rectilinear robot and x-y-z robot.
Robots are devices that are programmed to move parts, or to do work with a tool. Robotics is a multidisciplinary engineering field dedicated to the development of autonomous devices, including manipulators and mobile vehicles.
Roboticists develop man-made mechanical devices that can move by themselves, whose motion must be modelled, planned, sensed, actuated and controlled, and whose motion
behaviour can be influenced by nt”“programming”iftheys in moving in safe interaction with an unstructured environment, while autonomously achieving their specified tasks.
This definition implies that a device c movable mechanism, influenced by sensing, planning, actuation and control components. It does not imply that a minimum number of these components must be implemented in software, or be changeable by the “consumer” who uses can have been hard-wired into the device by the manufacturer.

Robot Arm Configurations

A robotic arm is a type of mechanical arm, usually programmable, with similar functions to a human arm. The links of such a manipulator are connected by joints allowing either rotational motion or translation displacement. The joints in these machines can be given names like “shoulder”, “elbow”, and “wrist”. A robot arm can be categorized according to its geometry. Identify and briefly discuss the four types of robot arm configurations illustrated in the figure.



a. Cartesian Configuration

The arm movement of a robot using the Cartesian configuration can be described by three intersecting perpendicular straight lines, referred to as the X, Y, and Z axes.One advantage of robots with a Cartesian configuration is that their totally linear movement allows for simpler controls. They also have a high degree of mechanical rigidity, accuracy, and repeatability. Typical applications for Cartesian robots include the following: assembly, machining operations, adhesive application, surface finishing, inspection, waterjet cutting, welding, nuclear material handling, robotic x-ray and neutron radiography,automated CNC lathe loading and operation, remotely operated decontamination, advanced munitions handling.

b. Cylindrical Configuration

A cylindrical configuration consists of two orthogonal slides, placed at a 90° angle, mounted on a rotary axis. A cylindrical configuration generally results in a larger work envelope than a Cartesian configuration. These robots are ideally suited for pick-and-place operations. Typical applications for cylindrical configurations include the following: machine loading and unloading, investment casting, conveyor pallet transfers, foundry and forging applications, general material handling and special payload handling and manipulation, meat packing, coating applications, assembly,injection molding and ,die casting.

c. Spherical (Polar) Configuration

The spherical configuration, sometimes referred to as the polar configuration, resembles the action of the turret on a military tank. The spherical configuration generally provides a larger work envelope than the Cartesian or cylindrical configurations. The design is simple and provides good weight lifting capabilities. Typical applications of spherical configurations include the following: die casting, injection molding, forging, machine tool loading, heat treating, glass handling, parts cleaning, dip coating, press loading, material transfer, stacking and unstacking.

d. Revolute (Articulated) Configuration

The revolute configuration, or jointed-arm, is the most common. These robots are often referred to as anthropomorphic because their movements closely resemble those of the human body. It also offers a more flexible reach than the other configurations, making it ideally suited to welding and spray painting operations. Typical applications of revolute configurations include the following: automatic assembly, parts and material handling, multiple-point light machining operations, in-process inspection, palletizing, machine loading and unloading, machine vision, material cutting, material removal, thermal coating, paint and adhesive application, welding and, die casting.

The End effector

In robotics, an end effector is a device or tool that's connected to the end of a robot arm where the hand would be. The end effector is the part of the robot that interacts with the environment. The structure of an end effector and the nature of the programming and hardware that drives it depend on the task the robot will be performing.

In manufacturing, a robot arm can accommodate only certain tasks without changes to its end effector's ancillary hardware and/or programming. If a robot needs to pick something up, a type of robot hand called a gripper is the most functional end effector. If a robot needs to be able to tighten screws, however, then the robot must be fitted with an end effector that can spin.

End effectors used in manufacturing include:

- anti-collision sensors
- brushes
- cameras
- cutting tools
- drills
- grippers
- magnets
- sanders
- screw drivers

- spray guns
- vacuum cups

Sensor and Vision

Vision sensors use images captured by a camera to determine presence, orientation, and accuracy of parts. These sensors differ from image inspection “systems” in that the camera, light, and controller are contained in a single unit, which makes the unit’s construction and operation simple. There are differences between these sensors and other general-purpose sensors. For example, multi-point inspections can be done with a single sensor. In addition, thanks to the wide field of view, detection is possible even when the target position is not consistent.

CHAPTER-3

Coordinate frames , mapping and transforms

3.1 Coordinate frames –Mapping

In a 3-D space, a c0ordinate frame is set of three orthogonal right handed axesX,Y, Z called principal axes. Such a frame shown in figure with the origin of the principal axes at o along with the unit vector { x, y ,z } along t5he axes. This frame is labelled as {x y z} or by a number as {1} using a numbering scheme.

Other frames in the space are similarly labelled.

A point P in a 3-d space can be defend with respect to this coordinate frame by vector OP (a directed line from origin O to point P pointing towards p).In vector notation

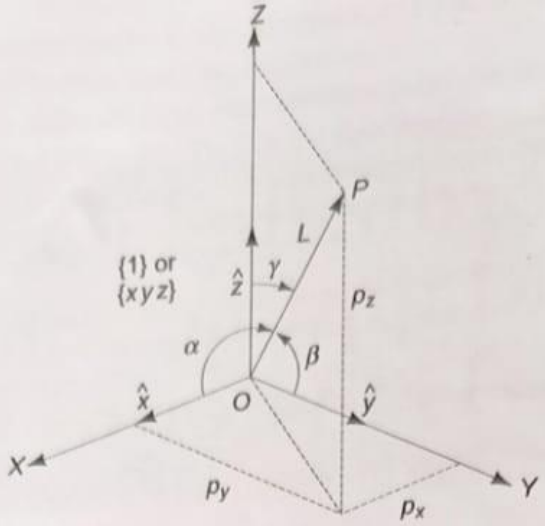
$$\vec{P} = \overrightarrow{OP} = p_x\hat{x} + p_y\hat{y} + p_z\hat{z}$$

Second frame is rotated with respect to the first, the origin of both the frame is same. In robotics, this is referred as changing the orientation

The second frame is moved away from the first the axes of both frame remain parallel respectively. This is the translation of the origin of the second frame from the first frame in space.

Second frame is rotating with respect to the first and moved away from it that is the second frame is translated and it is orientation is also changed.

This situation are modelled and it is important to note that mapping changes the description of the point and not the point itself.



Position and orientation of a point P in a coordinate frame

where p_x, p_y, p_z are the components of the vector $\overrightarrow{OP}$ along the three coordinate axes or the projections of the vector $\overrightarrow{OP}$ on the axes X, Y, Z, respectively. A *frame-space* notation is introduced as ${}^1\boldsymbol{P}$ to refer to the point P (or vector $\overrightarrow{OP}$) with respect to frame {1} with its components in the frame as ${}^1p_x$, ${}^1p_y$, and ${}^1p_z$, that is,

$$ {}^1\boldsymbol{P} = {}^1p_x\hat{\boldsymbol{x}} + {}^1p_y\hat{\boldsymbol{y}} + {}^1p_z\hat{\boldsymbol{z}} \tag{2.2} $$

In vector-matrix notation, this equation can be written in terms of the vector components only as:

$$ {}^1\boldsymbol{P} = \begin{bmatrix} {}^1p_x \\ {}^1p_y \\ {}^1p_z \end{bmatrix} = \begin{bmatrix} {}^1p_x & {}^1p_y & {}^1p_z \end{bmatrix}^T \tag{2.3} $$

Observe that the leading superscript refers to the coordinate frame number (frame {1} in this case) and $[\boldsymbol{A}]^T$ indicates the transpose of matrix $\boldsymbol{A}$. In addition, the direction of the position vector $\overrightarrow{OP}$ can be expressed by the direction cosines:

$$ \cos\alpha = \frac{{}^1p_x}{L}, \cos\beta = \frac{{}^1p_y}{L}, \cos\gamma = \frac{{}^1p_z}{L} $$

with

$$ L = |\vec{P}| = |\overrightarrow{OP}| = \sqrt{\left({}^1p_x\right)^2 + \left({}^1p_y\right)^2 + \left({}^1p_z\right)^2} \tag{2.4} $$

where α, β, and γ are, respectively, the right handed angles measured from the coordinate axes to the vector $\overrightarrow{OP}$, which has a length L.

Mapping

Mappings refer to changing the description of a point (or vector) in space from one frame to another frame. The second frame has three possibilities in relation to the first frame:

3.1 Mapping between rotated frame

(a) Second frame is rotated with respect to the first; the origin of both the frames is same. In robotics, this is referred as changing the orientation.
(b) Second frame is moved away from the first, the axes of both frames remain parallel, respectively. This is a translation of the origin of the second frame from the first frame in space.
(c) Second frame is rotated with respect to the first and moved away from it, that is, the second frame is translated and its orientation is also changed.

These situations are modelled in the following sections. It is important to note that mapping changes the description of the point and not the point itself.

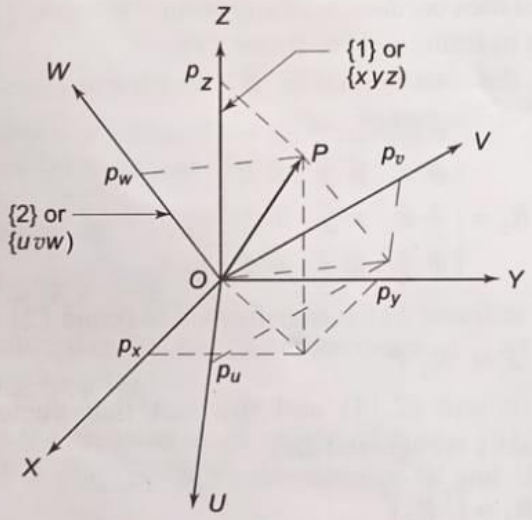
Mapping between Rotated Frames

Consider two frames, frame {1} with axes X, Y, Z, and frame {2} with axes U, V, W with a common origin, as shown in Fig. 2.2. A point P in space can be described by the two frames and can be expressed as vectors ${}^1\boldsymbol{P}$ and ${}^2\boldsymbol{P}$,

$$ {}^1\boldsymbol{P} = {}^1p_x\hat{\boldsymbol{x}} + {}^1p_y\hat{\boldsymbol{y}} + {}^1p_z\hat{\boldsymbol{z}} \tag{2.5} $$

$$ {}^2\boldsymbol{P} = {}^2p_u\hat{\boldsymbol{u}} + {}^2p_v\hat{\boldsymbol{v}} + {}^2p_w\hat{\boldsymbol{w}} \tag{2.6} $$

where ${}^2p_u$, ${}^2p_v$, ${}^2p_w$ are projections of point P on frame {2} or $\{u\ v\ w\}$ (the U, V, W coordinates). Because the point P is same, its two descriptions given by Eqs. (2.5) and (2.6) are related.



Representation of a point P in two frames {x y z} and {u v w} rotated with respect to each other

Now, let the problem be posed as, "The description of point P in frame {2} is known and its description in frame {1} is to be found (or vice-versa)." This is accomplished by projecting the vector ${}^2\boldsymbol{P}$ on to the coordinates of frame {1}. Projections of ${}^2\boldsymbol{P}$ on frame {1} are obtained by taking the dot product of ${}^2\boldsymbol{P}$ with the unit vectors of frame {1}. Thus, substituting for ${}^2\boldsymbol{P}$ from Eq. (2.6) gives

$$
\begin{aligned}
{}^1p_x &= \hat{x}\cdot {}^2\boldsymbol{P} = \hat{x}\cdot {}^2p_u\hat{u} + \hat{x}\cdot {}^2p_v\hat{v} + \hat{x}\cdot {}^2p_w\hat{w} \\
{}^1p_y &= \hat{y}\cdot {}^2\boldsymbol{P} = \hat{y}\cdot {}^2p_u\hat{u} + \hat{y}\cdot {}^2p_v\hat{v} + \hat{y}\cdot {}^2p_w\hat{w} \\
{}^1p_z &= \hat{z}\cdot {}^2\boldsymbol{P} = \hat{z}\cdot {}^2p_u\hat{u} + \hat{z}\cdot {}^2p_v\hat{v} + \hat{z}\cdot {}^2p_w\hat{w}
\end{aligned}
\tag{2.7}
$$

This can be written in matrix form as

$$
\begin{bmatrix} {}^1p_x \\ {}^1p_y \\ {}^1p_z \end{bmatrix} = \begin{bmatrix} \hat{x}\cdot\hat{u} & \hat{x}\cdot\hat{v} & \hat{x}\cdot\hat{w} \\ \hat{y}\cdot\hat{u} & \hat{y}\cdot\hat{v} & \hat{y}\cdot\hat{w} \\ \hat{z}\cdot\hat{u} & \hat{z}\cdot\hat{v} & \hat{z}\cdot\hat{w} \end{bmatrix} \begin{bmatrix} {}^2p_x \\ {}^2p_y \\ {}^2p_z \end{bmatrix} \tag{2.8}
$$

In compressed vector-matrix notation Eq. (2.8) is written as

$$
{}^1\boldsymbol{P} = {}^1\boldsymbol{R}_2\ {}^2\boldsymbol{P} \tag{2.9}
$$

where

$$
{}^1\boldsymbol{R}_2 = \begin{bmatrix} \hat{x}\cdot\hat{u} & \hat{x}\cdot\hat{v} & \hat{x}\cdot\hat{w} \\ \hat{y}\cdot\hat{u} & \hat{y}\cdot\hat{v} & \hat{y}\cdot\hat{w} \\ \hat{z}\cdot\hat{u} & \hat{z}\cdot\hat{v} & \hat{z}\cdot\hat{w} \end{bmatrix} \tag{2.10}
$$

Because frames {1} and {2} have the same origin, they can only be rotated with respect to each other, therefore, $\boldsymbol{R}$ is called a *rotation matrix* or *rotational transformation matrix*. It contains only the dot products of unit vectors of the two frames and is independent of the point P. Thus, rotation matrix ${}^1\boldsymbol{R}_2$ can be used for transformation of the coordinates of any point P in frame {2} (which has been rotated with respect to frame {1}) to frame {1}.

On similar lines, the rotation matrix ${}^2\boldsymbol{R}_1$, which expresses frame {1} as seen from frame {2}, is established as

$$
{}^2\boldsymbol{R}_1 = \begin{bmatrix} \hat{u}\cdot\hat{x} & \hat{u}\cdot\hat{y} & \hat{u}\cdot\hat{z} \\ \hat{v}\cdot\hat{x} & \hat{v}\cdot\hat{y} & \hat{v}\cdot\hat{z} \\ \hat{w}\cdot\hat{y} & \hat{w}\cdot\hat{y} & \hat{w}\cdot\hat{z} \end{bmatrix} \tag{2.11}
$$

Hence, a point P in frame {1} is transformed to frame {2} using

$$
{}^2\boldsymbol{P} = {}^2\boldsymbol{R}_1\ {}^1\boldsymbol{P} \tag{2.12}
$$

From Eqs. (2.10) and (2.11) and the fact that vector dot product is commutative, it is easily recognized that

$$
{}^2\boldsymbol{R}_1 = \left[{}^1\boldsymbol{R}_2\right]^T \tag{2.13}
$$

From Eqs. (2.9), (2.12), and (2.13), ${}^2\boldsymbol{P}$ is expressed as

$$
{}^2\boldsymbol{P} = \left[{}^1\boldsymbol{R}_2\right]^{-1}\ {}^1\boldsymbol{P} = {}^2\boldsymbol{R}_1\ {}^1\boldsymbol{P} = \left[{}^1\boldsymbol{R}_2\right]^T\ {}^1\boldsymbol{P} \tag{2.14}
$$

Therefore, it is concluded that

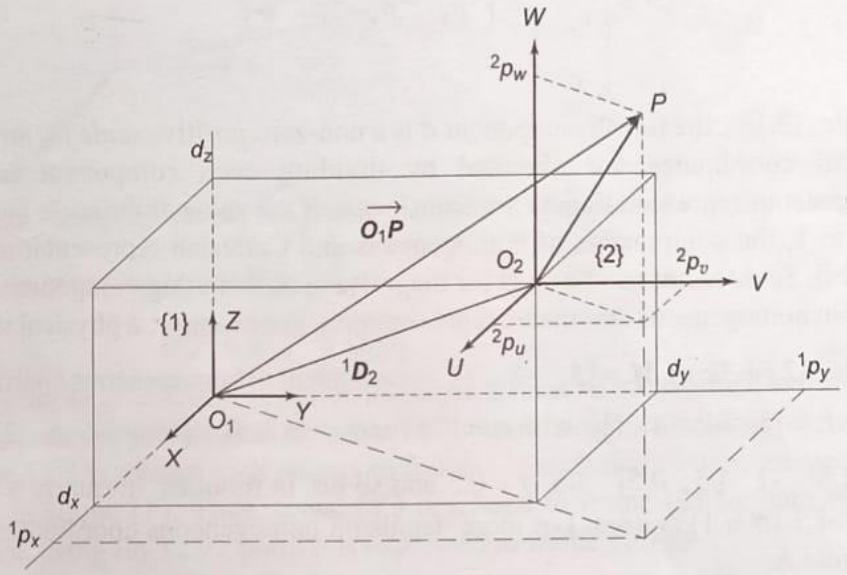
$$
{}^2\boldsymbol{R}_1 = \left[{}^1\boldsymbol{R}_2\right]^{-1} = \left[{}^1\boldsymbol{R}_2\right]^T
$$

or, in general, for any rotational transformation matrix $\boldsymbol{R}$

3.1 Mapping between translated frame

Mapping between Translated Frames

Consider two frames, frame {1} and frame {2}, with origins O_1 and O_2 such that the axes of frame {1} are parallel to axes of frame {2}, as shown in Fig. 2.3. A point P in space can be expressed as vectors $\overrightarrow{O_1P}$ and $\overrightarrow{O_2P}$ with respect to the frames {1} and {2}, respectively.



Translation of frames: frame {2} is translated with respect to frame {1} by distance $^1D_2$

The two vectors are related as

$$\overrightarrow{O_1P} = \overrightarrow{O_2P} + \overrightarrow{O_1O_2} \tag{2.16}$$

or in the notation introduced earlier Eq. (2.16) becomes

$$^1\boldsymbol{P} = {}^2\boldsymbol{P} + {}^1\boldsymbol{D}_2 \tag{2.17}$$

where $^1\boldsymbol{D}_2 = \overrightarrow{O_1O_2}$ is the translation of origin of frame {2} with respect to frame {1}. Because $^2\boldsymbol{P} = [^2p_u\ {}^2p_v\ {}^2p_w]^T$, substituting $^2\boldsymbol{P}$ and $^1\boldsymbol{D}_2$ in Eq. (2.17) gives

$$^1\boldsymbol{P} = (^2p_u + d_x)\hat{x} + (^2p_v + d_y)\hat{y} + (^2p_w + d_z)\hat{z} \tag{2.18}$$

As $^1\boldsymbol{P} = {}^1p_x\hat{x} + {}^1p_y\hat{y} + {}^1p_z\hat{z}$, this gives

$$^1p_x = {}^2p_u + d_x;\ {}^1p_y = {}^2p_v + d_y;\ {}^1p_z = {}^2p_z + d_z$$

which is verified from Fig. 2.3.

Translation is qualitatively different from rotation in one important respect. In rotation, the origin of two coordinate frames is same. This invariance of the origin

characteristic allows the representation of rotations in 3-D space as a 3×3 rotation matrix $\boldsymbol{R}$. However, in translation, the origins of translated frame and original frame are not coincident and translation is represented by a 3×1 vector, ${}^1\boldsymbol{D}_2$.

A powerful representation of translation is in a 4-D space of *homogeneous coordinates*. In these coordinates, point P in space with respect to frame $\{1\}$ is denoted as [refer to Fig. 2.1 and Eq. (2.3)]:

$$ {}^1\boldsymbol{P} = \begin{bmatrix} {}^1p_x \\ {}^1p_y \\ {}^1p_z \\ \sigma \end{bmatrix} = \begin{bmatrix} {}^1p_x & {}^1p_y & {}^1p_z & \sigma \end{bmatrix}^T \tag{2.19} $$

In Eq. (2.19), the fourth component σ is a non-zero positive *scale factor*. The physical coordinates are obtained by dividing each component in the homogeneous representation by the scale factor. If the value of the scale factor σ is set to 1, the components of homogeneous and Cartesian representation are identical. Scale factor can be used for magnifying or shrinking components of a vector in homogeneous coordinate representation. For example, a physical vector $\overrightarrow{\boldsymbol{M}} = 5\hat{\boldsymbol{i}} - 2\hat{\boldsymbol{j}} + 3\hat{\boldsymbol{k}}$ or $\boldsymbol{M} = [5 \quad -2 \quad 3]^T$ is equivalent to homogeneous coordinate vector $\boldsymbol{L} = [5 \quad -2 \quad 3 \quad 1]^T$ with $\sigma = 1$ or for $\sigma = 2$ it is $\boldsymbol{L} = [10 \quad -4 \quad 6 \quad 2]^T$ or $\boldsymbol{L} = [2.5 \quad -1 \quad 1.5 \quad 0.5]^T$ for $\sigma = 0.5$ and so on. In robotics, normally a scale factor of 1 ($\sigma = 1$) is used. For more details on homogeneous coordinates, see Appendix **A**.

Using the homogeneous coordinates, Eq. (2.17) is written in the vector-matrix form as:

$$ {}^1\boldsymbol{P} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^2p_u \\ {}^2p_v \\ {}^2p_w \\ 1 \end{bmatrix} $$

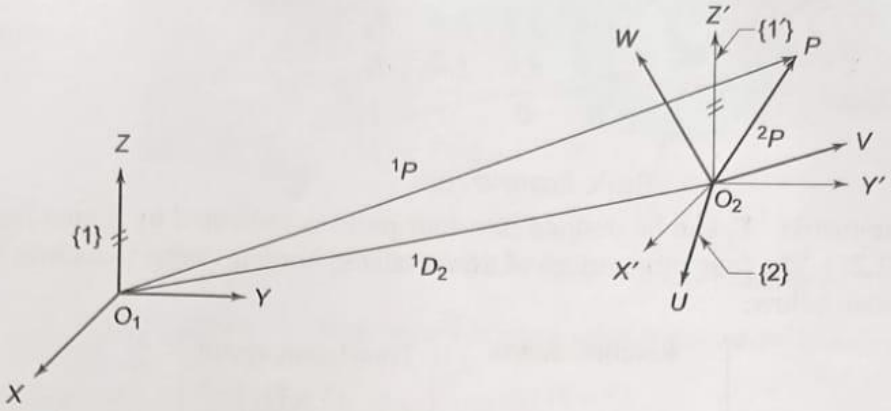
or

$$ {}^1\boldsymbol{P} = {}^1\boldsymbol{T}_2 \, {}^2\boldsymbol{P} \tag{2.20} $$

Here, ${}^1\boldsymbol{T}_2$ is a 4×4 homogeneous *transformation matrix* for translation of origin by ${}^1\boldsymbol{D}_2 = \overrightarrow{\boldsymbol{O}_1\boldsymbol{O}_2} = [d_x \quad d_y \quad d_z \quad 1]^T$. It is easily seen that Eq. (2.20) is same as Eq. (2.18). The 4×4 transformation matrix in Eq. (2.20) is called the *basic homogeneous translation matrix*.

3.1 Mapping between rotated and translated frame

point P described with respect to frame {2} as ${}^2\boldsymbol{P}$, it is required to refer it to frame {1}, that is, to find ${}^1\boldsymbol{P}$.



Mapping between two frames–translated and rotated with respect to each other

In terms of vectors in Fig. 2.4,

$$\overrightarrow{O_1P} = \overrightarrow{O_2P} + \overrightarrow{O_1O_2} \tag{2.21}$$

Vector $\overrightarrow{O_2P}$ in frame {2} is ${}^2\boldsymbol{P}$; therefore, it must be transformed to frame {1}. First, consider an intermediate frame {1′} with its origin coincident with O_2. The frame {1′} is rotated with respect to frame {2} such that its axes are parallel to axes of frame {1}. Thus, frame {1′} is related to frame {2} by pure rotation. Hence, using Eq. (2.9), point P is expressed in frame {1′} as

$${}^{1'}\boldsymbol{P} = {}^{1'}\boldsymbol{R}_2\,{}^2\boldsymbol{P} \tag{2.22}$$

Because frame {1′} is aligned with frame {1}, ${}^{1'}\boldsymbol{R}_2 = {}^1\boldsymbol{R}_2$. Hence

$$\overrightarrow{O_2P} = {}^{1'}\boldsymbol{P} = {}^1\boldsymbol{R}_2\,{}^2\boldsymbol{P} \tag{2.23}$$

Substituting this in Eq. (2.21) and converting to vector-matrix notation,

$${}^1\boldsymbol{P} = {}^1\boldsymbol{R}_2\,{}^2\boldsymbol{P} + {}^1\boldsymbol{D}_2 \tag{2.24}$$

The vector $\overrightarrow{O_1O_2}$ or ${}^1\boldsymbol{D}_2$ has components $(d_x\ d_y\ d_z)$ in frame {1} as

$$\overrightarrow{O_1O_2} = {}^1\boldsymbol{D}_2 = [d_x \quad d_y \quad d_z]^T \tag{2.25}$$

Using the homogeneous coordinates, from Eqs. (2.10) and (2.20), the two terms on the right-hand side of Eq. (2.24) can be combined into a single 4×4 matrix, which is then written as

$${}^1\boldsymbol{P} = {}^1\boldsymbol{T}_2\,{}^2\boldsymbol{P} \tag{2.26}$$

Here, ${}^1\boldsymbol{P}$ and ${}^2\boldsymbol{P}$ are 4×1 vectors as in Eq. (2.19) with a scale factor of 1 and $\boldsymbol{T}$ is 4×4 matrix referred to as the *homogeneous transformation matrix* (or *homogeneous transform*). It describes both the position and orientation of frame {2} with respect to frame {1} or any frame with respect to any other frame. The components of ${}^1\boldsymbol{T}_2$ matrix are as under

$$
{}^1T_2 = \left[\begin{array}{ccc:c}
\hat{x}.\hat{u} & \hat{x}.\hat{v} & \hat{x}.\hat{w} & d_x \\
\hat{y}.\hat{u} & \hat{y}.\hat{v} & \hat{y}.\hat{w} & d_y \\
\hat{z}.\hat{u} & \hat{z}.\hat{v} & \hat{z}.\hat{w} & d_z \\
\hdashline
0 & 0 & 0 & 1
\end{array}\right] \qquad (2.27)
$$

${}^1R_2$ — rotation matrix; ${}^1D_2$ — translation vector; Scale factor σ

The matrix ${}^1T_2$ can be divided into four parts as indicated by dotted lines in Eq. (2.27). The four submatrices of a generalized homogeneous transform T are as shown below:

$$
T = \left[\begin{array}{c:c}
\begin{array}{c}\text{Rotation matrix} \\ (3\times 3)\end{array} & \begin{array}{c}\text{Translation vector} \\ (3\times 1)\end{array} \\
\hdashline
\begin{array}{c}\text{Perspective} \\ \text{transformation matrix} \\ (1\times 3)\end{array} & \begin{array}{c}\text{Scale factor} \\ (1\times 1)\end{array}
\end{array}\right] \qquad (2.28)
$$

Perspective transformation matrix is useful in vision systems and is set to zero vector wherever no perspective views are involved. The scale factor σ has non-zero positive ($\sigma > 0$) values and is called *global scaling* parameter. $\sigma > 1$ is useful for reducing and $0 < \sigma < 1$ is useful for enlarging. For robotic study presented here $\sigma = 1$ is used. For describing the position and orientation of frame {2} with respect to frame {1}, T takes the form

$$
{}^1T_2 = \left[\begin{array}{c:c}
{}^1R_2 & {}^1D_2 \\
\hdashline
0\ \ 0\ \ 0 & 1
\end{array}\right] \qquad (2.29)
$$

In the reverse problem when ${}^1P$ is known and ${}^2P$ is to be found, Eq. (2.26), takes the form

$$
{}^2P = {}^2T_1\ {}^1P \qquad (2.30)
$$

where ${}^2T_1 = \left[{}^1T_2\right]^{-1}$.

3.2 Fundamental rotation

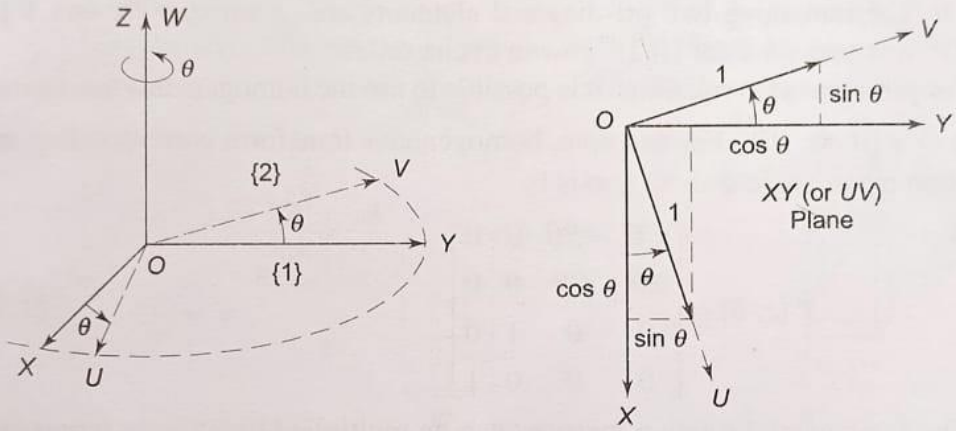
Principal rotation

FUNDAMENTAL ROTATION MATRICES

In the previous section, the background to describe the orientation of frame {2} with respect to frame {1} has been developed. These are now applied to rotation matrices in different situations. A frame {2} may be rotated about one or more of the principal axes, an arbitrary axis, or by some fixed angles relative to frame {1}. Each of these situations is discussed in this section.

Principal Axes Rotation

To determine the orientation of frame {2}, which is rotated about one of the three principal axes of frame {1}, consider, for example, the rotation of frame {2} with respect to frame {1} by angle θ about the z-axis of frame {1}, as shown in a 3-D view in Fig. 2.11 (a) and on xy-plane in Fig. 2.11 (b). The corresponding rotation matrix ${}^1\boldsymbol{R}_2$, known as the *fundamental rotation matrix*, is denoted by the symbol $\boldsymbol{R}_z(\theta)$ or $\boldsymbol{R}(z, \theta)$ or $\boldsymbol{R}_{z,\theta}$.



(a) Rotation of frame {1} by θ

(b) xy-plane showing the rotation

Fundamental rotation by an angle θ about z-axis

From Eq. (2.10), $\boldsymbol{R}_z(\theta)$ is computed from the dot product of unit vectors along the principal axes. The dot product of two unit vectors is the cosine of the angle between them, for example, $\hat{\boldsymbol{x}} \cdot \hat{\boldsymbol{u}} = \cos\theta$. Thus,

$$\boldsymbol{R}_z(\theta) = \begin{bmatrix} \cos\theta & \cos(90°+\theta) & \cos 90° \\ \cos(90°-\theta) & \cos\theta & \cos 90° \\ \cos 90° & \cos 90° & \cos 0° \end{bmatrix}$$

or
$$\boldsymbol{R}_z(\theta) = \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.54}$$

where $S\theta = \sin\theta$ and $C\theta = \cos\theta$.

Equation (2.54) is the fundamental rotation matrix for a rotation of angle θ about z-axis of the frame. Similarly, fundamental rotation matrices for rotation about x-axis and y-axis can be obtained and these are:

$$\boldsymbol{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & -S\theta \\ 0 & S\theta & C\theta \end{bmatrix} \tag{2.55}$$

and
$$\boldsymbol{R}_y(\theta) = \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} \tag{2.56}$$

The rotation matrices $\boldsymbol{R}_x$, $\boldsymbol{R}_y$, and $\boldsymbol{R}_z$ exhibit a pattern and using this pattern these matrices can be easily written. The rotation matrix for rotation about k^{th} principal axis $\boldsymbol{R}_k(\theta)$ can be obtained as follows: The elements of ith row and i^{th} column for $i = 1, 2$, or 3 for $k = x, y$, or z respectively, of 3×3 matrix $\boldsymbol{R}_k(\theta)$ are zero except the element (i, i), which is 1. The other two diagonal elements are $\cos\theta$. The remaining two off-diagonal elements are $\pm \sin\theta$, with $-\sin\theta$ for $(i+1)^{\text{th}}$ row and $\sin\theta$ for $(i+2)^{\text{th}}$ row in cyclic order.

For principal axes rotations, it is possible to use the homogeneous transform $\boldsymbol{T}$ with $\boldsymbol{D} = [0 \;\; 0 \;\; 0]^T$. For example, homogeneous transform corresponding to a rotation by an angle θ about z-axis is

$$\boldsymbol{T}(z, \theta) = \left[\begin{array}{ccc:c} C\theta & -S\theta & 0 & 0 \\ S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \tag{2.57}$$

The fundamental rotation matrices can be multiplied together to represent a sequence of finite rotations. For example, the overall rotation matrix representing a rotation of angle θ_1 about x-axis followed by a rotation of angle θ_2 about y-axis can be obtained by multiplying Eq. (2.55) and Eq. (2.56). That is,

$$\boldsymbol{R} = R_y(\theta_2)\, R_x(\theta_1)$$

or
$$\boldsymbol{R} = \begin{bmatrix} C\theta_2 & 0 & S\theta_2 \\ 0 & 1 & 0 \\ -S\theta_2 & 0 & C\theta_2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta_1 & -S\theta_1 \\ 0 & S\theta_1 & C\theta_1 \end{bmatrix}$$

or
$$\boldsymbol{R} = \begin{bmatrix} C_2 & S_1S_2 & C_1S_2 \\ 0 & C_1 & -S_1 \\ -S_2 & S_1C_2 & C_1C_2 \end{bmatrix} \tag{2.58}$$

where $C_i = C\theta_i = \cos\ \theta_i$ and $S_i = S\theta_i = \sin\ \theta_i$.

It is important to note the sequence of multiplication of $\boldsymbol{R}$ matrices. A different sequence may not give the same result and obviously will not correspond to same orientation of the rotated frame. This is because the matrix product is not commutative. In view of this, it can be concluded that two rotations in general do not result in same orientation and the resultant rotation matrix depends on the order of rotations.

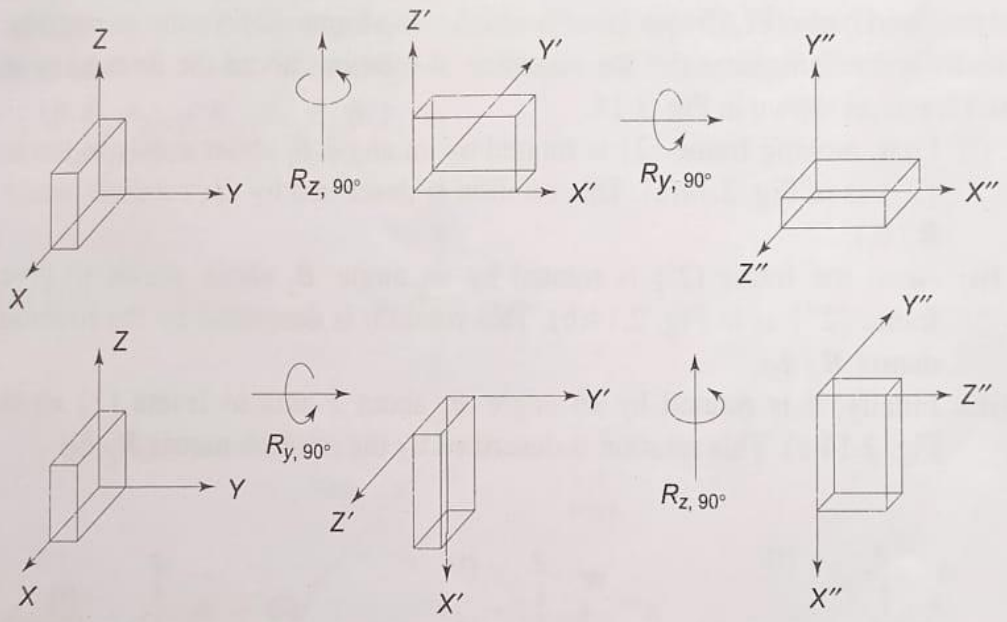
Another significant variable is how the rotations are performed. There are two alternatives:

(i) to perform successive rotations about the principal axes of the fixed frame.

(ii) to perform successive rotations about the current principal axes of a moving frame.

The successive rotations in either case, in general, do not produce identical results.

Figure 2.12 shows the effect of two successive rotations of 90° to an object about the principal axes of the fixed frame. It is observed that the final orientation of the object is different when same two rotations are made but the order of rotations is changed.



Effect of order of rotations of a cuboid about principal axes of a fixed frame

Similarly, the order of rotations about the principal axes of the moving frame also produces different final orientation of the object. This is illustrated in Fig.

The representation of orientation of rotated frames for different types of rotations is discussed next.

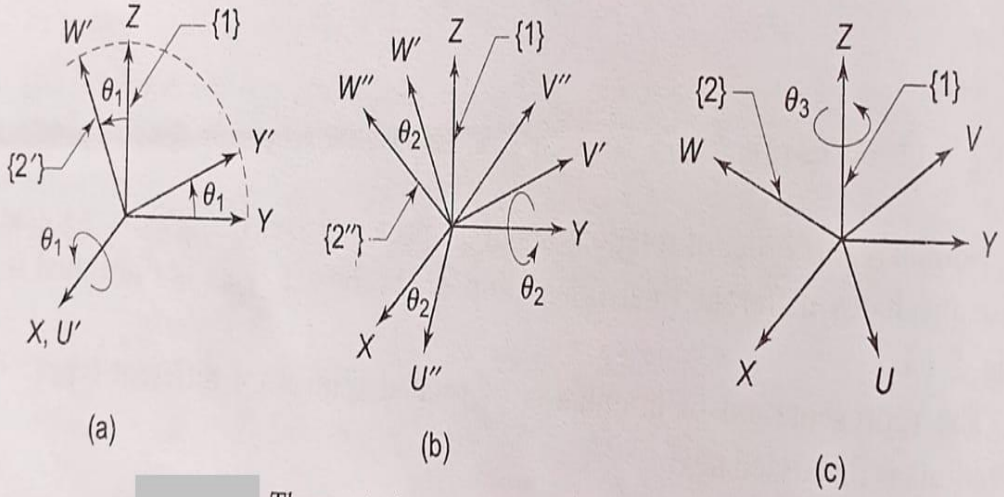
3.2 Fixed angel representation

Let the fixed frame {1}(frame $\{x\,y\,z\}$) and moving frame {2} (frame $\{u\,v\,w\}$) be initially coincident. Consider the sequence of rotations about the three axes of fixed frame as shown in Fig. 2.14.

(i) First, moving frame {2} is rotated by an angle θ_1 about x-axis to frame {2′} as in Fig. 2.14(a). This rotation is described by the rotation matrix $\boldsymbol{R}_x(\theta_1)$.

(ii) Next, the frame {2′} is rotated by an angle θ_2 about y-axis to give frame {2″} as in Fig. 2.14(b). This rotation is described by the rotation matrix $\boldsymbol{R}_y(\theta_2)$.

(iii) Finally, it is rotated by an angle θ_3 about z-axis to frame {2} as in Fig. 2.14(c). This rotation is described by the rotation matrix $\boldsymbol{R}_z(\theta_3)$.



Three rotations of θ_1, θ_2, and θ_3 about fixed axes

This convention for specifying orientation is known as *fixed angle representation* because each rotation is specified about an axis of fixed reference frame. The above three rotations are referred as *XYZ-fixed angle* rotations.

The final frame orientation is obtained by composition of rotations with respect to the fixed frame and the overall rotation matrix ${}^{1}\boldsymbol{R}_{2}$ is computed by pre-multiplication of the matrices of elementary rotations, that is,

$$\boldsymbol{R}_{xyz}(\theta_3\,\theta_2\,\theta_1) = {}^{1}\boldsymbol{R}_{2} = \boldsymbol{R}_{z}(\theta_3)\boldsymbol{R}_{y}(\theta_2)\boldsymbol{R}_{x}(\theta_1) \tag{2.59}$$

(rotation ordering right to left)

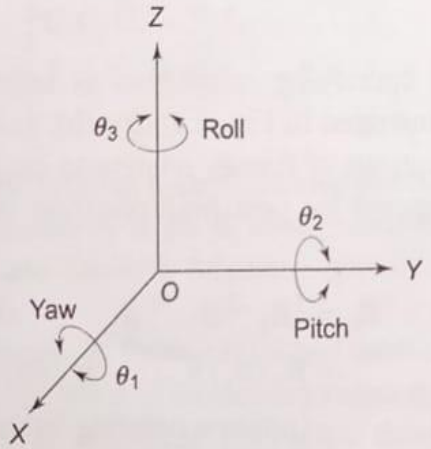
Substituting the results of Eqs. (2.54)–(2.56) in Eq. (2.59) for fixed angle rotations, the final rotation matrix is

$$\boldsymbol{R}_{xyz}(\theta_3\,\theta_2\,\theta_1) = \begin{bmatrix} C_3 & -S_3 & 0 \\ S_3 & C_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} C_2 & 0 & S_2 \\ 0 & 1 & 0 \\ -S_2 & 0 & C_2 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & C_1 & -S_1 \\ 0 & S_1 & C_1 \end{bmatrix}$$

or

$$\boldsymbol{R}_{xyz}(\theta_3\,\theta_2\,\theta_1) = \begin{bmatrix} C_2C_3 & S_1S_2C_3 - C_1S_3 & C_1S_2C_3 + S_1S_3 \\ C_2S_3 & S_1S_2S_3 + C_1C_3 & C_1S_2S_3 - S_1C_3 \\ -S_2 & S_1C_2 & C_1C_2 \end{bmatrix} \tag{2.60}$$

The final frame orientation for any set of rotations performed about the axes of the fixed frame (e.g. *ZYX*, *ZXZ* etc.) can be obtained by multiplying the rotation matrices in a consistent order as indicated in Eq. (2.59). In fixed angle representation, order of rotations *XYZ* or *ZYX* are equivalent, that is, $\boldsymbol{R}_{xyz}(\theta_1\,\theta_2\,\theta_3) = \boldsymbol{R}_{zyx}(\theta_1\,\theta_2\,\theta_3)$.



Representation of roll, pitch, and yaw (RPY) rotations

The three rotations about the three fixed principal axes in fixed angle rotation produce the motions, which are also known as *roll*, *pitch*, and *yaw* motions, as shown in Fig. 2.15. The *XYZ*-fixed angle transformation in Eq. (2.60) is equivalent to *roll-pitch-yaw* (RPY) transformation.

3.2 Euler angel representation

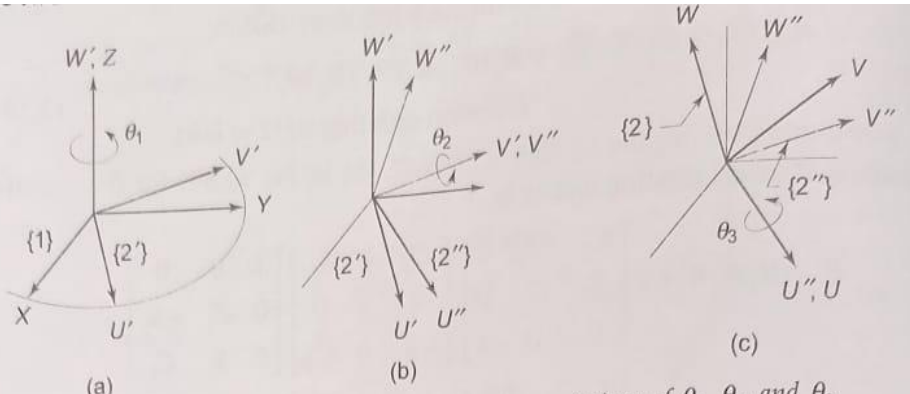


Fig. 2.16 *Euler angle representation for three rotations of θ_1, θ_2, and θ_3*

(i) To begin with, frame {2} is rotated by an angle θ_1 about its w-axis coincident with z-axis of frame {1}. The rotated frame is now {2′} and the rotation is described by the rotation matrix $\boldsymbol{R}_w(\theta_1)$.

(ii) Next, moving frame {2′} is rotated by an angle θ_2 about v'-axis, the rotated v-axis to frame {2″}. This rotation is described by the rotation matrix $\boldsymbol{R}_{v'}(\theta_2)$.

(iii) Finally, frame {2″} is rotated by an angle θ_3 about its u''-axis, the rotated u-axis to give frame {2}. This rotation is described by the rotation $\boldsymbol{R}_{u''}(\theta_3)$.

This convention for specifying orientation is called *WVU-Euler angle representation* and is illustrated in Fig. 2.16(*a*), (*b*), and (*c*). Viewing each of these rotations as descriptions of frames relative to each other, the equivalent rotation matrix is computed by post multiplication of the matrices of the elementary rotations as

$$\begin{aligned}\boldsymbol{R}_{wvu}(\theta_1\theta_2\theta_3) &= {}^1\boldsymbol{R}_2 = {}^1\boldsymbol{R}_{2'}\ {}^{2'}\boldsymbol{R}_{2''}\ {}^{2''}\boldsymbol{R}_2 \\ &= \boldsymbol{R}_w(\theta_1)\boldsymbol{R}_{v'}(\theta_2)\boldsymbol{R}_{u''}(\theta_3) \qquad (2.61)\\ &\quad \text{(rotation ordering left to right)}\end{aligned}$$

The rotations are performed about the current axes of the moving frame $\{uvw\}$. Using the results of Eqs. (2.54)–(2.56), the resulting frame orientation or the rotation matrix is

$$\boldsymbol{R}_{wvu}(\theta_1\theta_2\theta_3) = \begin{bmatrix} C_1 & -S_1 & 0 \\ S_1 & C_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} C_2 & 0 & S_2 \\ 0 & 1 & 0 \\ -S_2 & 0 & C_2 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & C_3 & -S_3 \\ 0 & S_3 & C_3 \end{bmatrix}$$

or

$$\boldsymbol{R}_{wvu}(\theta_1\theta_2\theta_3) = \begin{bmatrix} C_2C_3 & S_1S_2C_3 - C_1S_3 & C_1S_2C_3 + S_1S_3 \\ C_2S_3 & S_1S_2S_3 + C_1C_3 & C_1S_2S_3 - S_1C_3 \\ -S_2 & S_1C_2 & C_1C_2 \end{bmatrix} \qquad (2.62)$$

It is observed that this result is exactly same as that obtained for fixed angle representation, Eq. (2.60), but the rotations about the fixed axes were performed in opposite order. In general, three rotations performed about fixed axes give the same final orientation as obtained by the same three rotations performed in the opposite order about the moving axes. Hence,

$$\boldsymbol{R}_{xyz}(\theta_3\,\theta_2\,\theta_1) = \boldsymbol{R}_{wvu}(\theta_1\theta_2\theta_3) = \boldsymbol{R}_{zyx}(\theta_1\,\theta_2\,\theta_3) \tag{2.63}$$

Another most widely used Euler angle representation consists of the so called ZYZ representation for rotations about the axes of the current frame. The sequences of elementary rotations corresponding to this representation are:

(i) A rotation by angle θ_1 about the w-axis (or z-axis of the fixed frame), that is, $\boldsymbol{R}_w(\theta_1)$.

(ii) The second rotation by angle θ_2 about the rotated v-axis, that is $\boldsymbol{R}_{v'}(\theta_2)$. These two rotations are same as the previous case in Fig. 2.13.

(iii) Finally, a rotation of angle θ_3 about the rotated w-axis, that is $\boldsymbol{R}_{w''}(\theta_3)$.

The resulting rotation matrix is

$$\begin{aligned}\boldsymbol{R}_{wvw}(\theta_1\theta_2\theta_3) &= {}^1\boldsymbol{R}_2 = \boldsymbol{R}_w(\theta_1)\boldsymbol{R}_{v'}(\theta_2)\boldsymbol{R}_{w''}(\theta_3)\\ &= \begin{bmatrix} C_1 & -S_1 & 0\\ S_1 & C_1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix} C_2 & 0 & S_2\\ 0 & 1 & 0\\ -S_2 & 0 & C_2\end{bmatrix}\begin{bmatrix} C_3 & -S_3 & 0\\ S_3 & C_3 & 0\\ 0 & 0 & 1\end{bmatrix}\\ &= \begin{bmatrix} C_1C_2C_3 - S_1S_3 & -C_1C_2S_3 - S_1C_3 & C_1S_2\\ S_1C_2C_3 + C_1S_3 & -S_1C_2S_3 + C_1C_3 & S_1S_2\\ -S_2C_3 & S_2S_3 & C_2\end{bmatrix}\end{aligned} \tag{2.64}$$

The above Euler angle rotation matrix can also be obtained by rotations about the fixed frame as: a rotation by angle θ_3 about z-axis followed by a rotation by angle θ_2 about y-axis and finally a rotation angle θ_1 again about z-axis. The reader should verify this.

In all, twelve distinct sets of Euler angles and twelve sets of fixed angles are possible, with regard to sequence of elementary rotations. Other alternative Euler angle representations are also in vogue. For each of these, the rotation matrix can be found on similar lines.

Equevalent angel axis representation

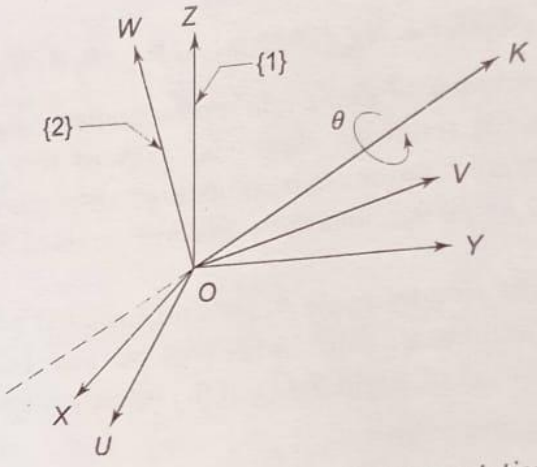


Fig. 2.17 *Equivalent angle axis representation*

These rotations are illustrated with the help of a vector $\boldsymbol{P}$, initially in the direction of k-axis, in Fig. 2.18.

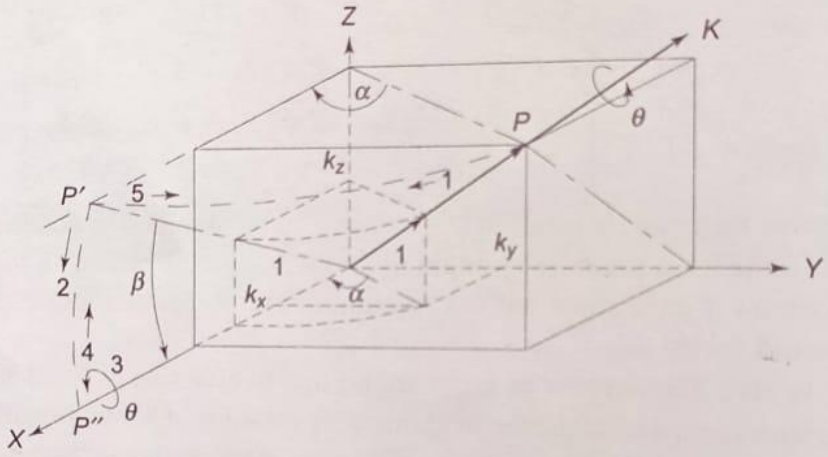


Fig. 2.18 *Rotations of frame about k-axis*

First, rotate the vector $\boldsymbol{P}$ (along with axis k and frame {2} of Fig. 2.17) by an angle $-\alpha$ about z-axis such that this rotation causes the axis k to lie in xz-plane of frame {1}. This rotation, marked as "1" in Fig. 2.18 and is written as

$$^{1}\boldsymbol{R}_{2} = \boldsymbol{R}_{z}(-\alpha) \tag{2.65}$$

Next, vector $\boldsymbol{P}$ (along with rotated axis k) is rotated about y-axis by an angle β so that axis k aligns with x-axis, rotation "2". At the end of this rotation,

$$^{1}\boldsymbol{R}_{2} = \boldsymbol{R}_{y}(\beta)\,\boldsymbol{R}_{z}(-\alpha) \tag{2.66}$$

Now a rotation "3" of angle θ about the rotated axis k, which is rotation about x-axis, is made. The resulting rotation matrix is then

$$^1R_2 = R_x(\theta)\, R_y(\beta)\, R_x(-\alpha) \tag{2.67}$$

Finally, the rotations "4" and "5" of $-\beta$ and α are made about y- and z-axes, respectively, in the opposite sense and reverse order so as to restore the k-axis to its original position leaving frame {2} in the rotated position. This gives

$$^1R_2 = R_k(\theta) = R_z(\alpha)\, R_y(-\beta)\, R_x(\theta)\, R_y(\beta)\, R_z(-\alpha) \tag{2.68}$$

Substituting the values for the fundamental rotation matrices from Eqs. (2.54)–(2.56) gives,

$$^1R_2 = \begin{bmatrix} C\alpha & -S\alpha & 0 \\ S\alpha & C\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\beta & 0 & -S\beta \\ 0 & 1 & 0 \\ S\beta & 0 & C\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & -S\theta \\ 0 & S\theta & C\theta \end{bmatrix} \begin{bmatrix} C\beta & 0 & S\beta \\ 0 & 1 & 0 \\ -S\beta & 0 & C\beta \end{bmatrix} \begin{bmatrix} C\alpha & S\alpha & 0 \\ -S\alpha & C\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.69}$$

The angles α and β can be eliminated from Eq. (2.69) using the geometry. From Fig. 2.18, following are easily observed for the unit vector $\hat{k} = \begin{bmatrix} k_x & k_y & k_z \end{bmatrix}^T$

$$\sin\alpha = \frac{k_y}{\sqrt{k_x^2 + k_y^2}}, \cos\alpha = \frac{k_x}{\sqrt{k_x^2 + k_y^2}}$$

$$\sin\beta = k_z, \cos\beta = \sqrt{k_x^2 + k_y^2} \tag{2.70}$$

Substituting these in Eq. (2.69) and simplifying gives

$$^1R_2 = R_k(\theta) = \begin{bmatrix} k_x^2 V\theta + C\theta & k_x k_y V\theta - k_z S\theta & k_x k_z V\theta + k_y S\theta \\ k_x k_y V\theta + k_z S\theta & k_y^2 V\theta + C\theta & k_y k_z V\theta - k_x S\theta \\ k_x k_z V\theta - k_y S\theta & k_y k_z V\theta + k_x S\theta & k_z^2 V\theta + C\theta \end{bmatrix} \tag{2.71}$$

where k_x, k_y, k_z are the projections of a unit vector $\hat{k}$ on frame $\{xyz\}$, and $V\theta = 1 - \cos\theta$.

This is an important rotation matrix and must be thoroughly understood. The principal axes fundamental rotations can be obtained from Eq. (2.71). For example, if k-axis is aligned with z-axis, $R_k(\theta)$ becomes $R_z(\theta)$ with $k_x = k_y = 0$ and $k_z = 1$. Substituting these k_x, k_y, k_z in Eq. (2.71) gives

$$R_k(\theta) = R_z(\theta) = \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.72}$$

which is same as Eq. (2.54) derived before.

Note that any combination of rotations about the principal axes of a coordinate frame is always equivalent to a single rotation by some angle θ about some

Short questions

Q1 Define coordinate frames

A *coordinate frame* in two-dimensional space is a set of two vectors having unit length and that make a right angle with one another. The vectors are called the *x*-vector and the *y*-vector. (It is often justa matter of convenience which is *x* and which is *y,* as long as you don't switch in the middle of a problem.)

Q 2 Define mapping

Mapping refer to changing the description of a point or vector in space from one frame to another frame. The 2nd frame has threeposibilities in relation to the 1st frame. It is important to note that mapping changes the description of the point not the point itself.

Q3 Define fundamental rotation

A rotation is a circular movement of an object around a center (or point) of rotation. The geometric plane along which the rotation occurs is called the *rotation plane*, and the imaginary line extending from the center and perpendicular to the rotation plane is called the *rotation axis* . A three-dimensional object can always be rotated about an infinite number ofrotation axes.

Q 4 Define Euler angel representation

The **Euler angles** are three angles introduced by Leonhard Euler to describe the orientation ofa rigid body with respect to a fixed coordinate system.[1]

They can also represent the orientation of a mobile frame of reference in physics or the orientation of a general basis in 3-dimensional linear algebra. Alternative forms were later introduced by Peter Guthrie Tait and George H. Bryan intended for use in aeronautics andengineering.

Q 5What is principial axe rotation

A **principal axis** of **rotation** (or **principal** direction) is an eigenvector of the mass moment of inertia tensor (introduced in the previous section) defined relative to some point (typicallythe center of mass). The corresponding eigenvalues are called the **principal** moments of inertia.

Long questions

1- The coordinates of point P with respect to a moving coordinate frame are given as $P=[0.5 \quad 0.8\ 1.3\ 1]^T$. What are the coordinates of P with respect to coordinate frame,if the moving frame is rotated by 90^0 about z axis of the fixed frame .

2- Determine the rotation matrix for a rotation of 45^0 of y axis, followed bya rotation 120^0 of z axis and final rotation of 90^0 about x axis.

CHAPTER -4

Robot kimenatics and dynamic

4.1 Kinametic modeling

With the definition of fixed and variable kinematic parameters for each link, kinematic models can be defined. This model is the analytical description of the spatial geometry of motion of the manipulator with respect to a fixed (inertial) reference frame, as a function of time. In particular, the relation between the joint-variables and the position and orientation of the end-effector is the kinematic model. It is required to control position and orientation of the end-effector, in 3-D space, so that it can follow a defined trajectory or manipulate objects in the workspace. The kinematic modeling problem is split into two problems as:

1. Given the set of joint-link parameters, the problem of finding the position and orientation of the end-effector with respect to a known (immobile or inertial) reference frame for an n-DOF manipulator is the first problem. This is referred to as *direct (or forward) kinematic model* or *direct kinematics*. This model gives the position and orientation of the end-effector as a function of the joint variables and other joint-link constant parameters.
2. For a given position and orientation of the end-effector (of the n-DOF manipulator), with respect to an immobile or inertial reference frame, it is required to find a set of joint variables that would bring the end-effector in the specified position and orientation.
   This is the second problem and is referred to as the *inverse kinematic model* or *inverse kinematics*.

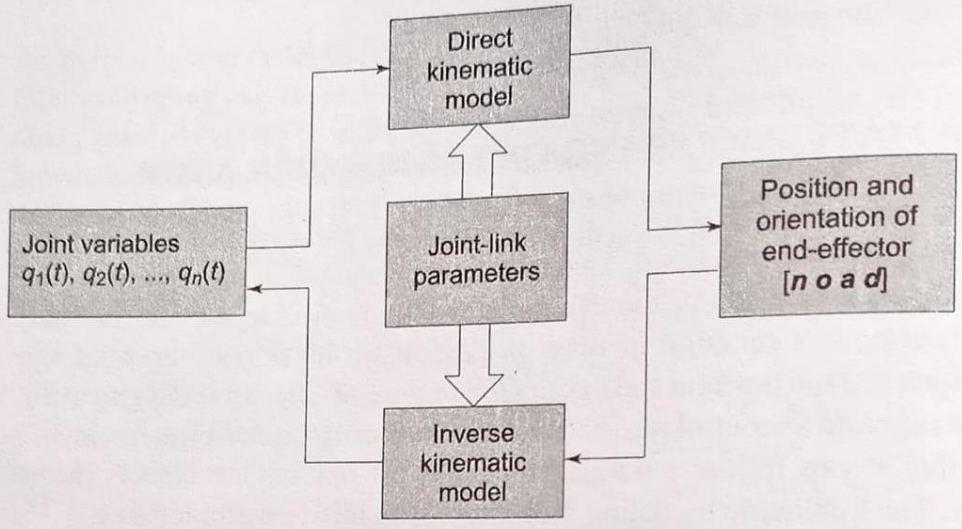


Fig. 3.7 *The direct and inverse kinematic models*

Mechanical structure and notation

The anatomy of the manipulator was discussed in Chapter 1. A manipulator consists of a chain of rigid bodies, called *links*, connected to each other by joints, which allow linear or revolute motion between connected links each of which exhibits just one degree of freedom (DOF). Joints with more than one DOF are not common. A joint with m degrees of freedom can be modeled as m joints with one degree of freedom each connected with $(m-1)$ links of zero length. Most industrial robotic manipulators are open serial kinematic chains, that is, each link is connected to two other links, at the most, without the formation of closed loops. In open chain robots, all joints are motorized (active). Some robots may have closed kinematic chains such as parallelogram linkages and require different considerations to model them.

The number of degrees of freedom a manipulator possesses is the number of independent parameters required to completely specify its position and orientation in space. Because each joint has only one degree of freedom, the degrees of freedom of a manipulator are equal to number of joints.

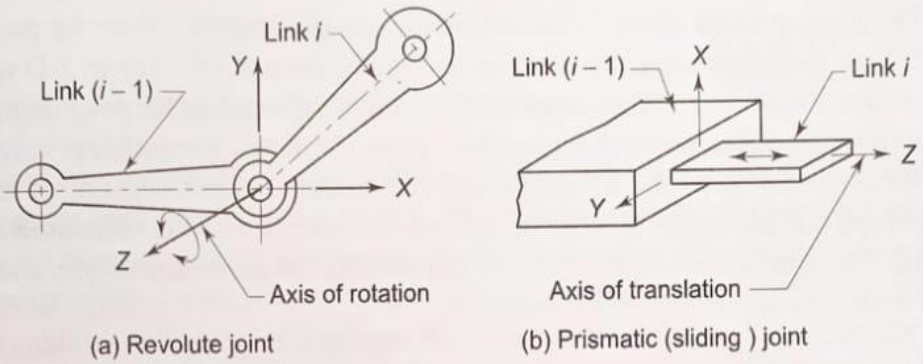


Fig. 3.1 *Two common types of joints and axis of motion (joint axis)*

Single DOF joints between links of a manipulator can be classified as revolute or prismatic. A *revolute joint*, denoted as R-joint, allows rotational motion between connected links. A *prismatic joint*, denoted as P-joint, also known as sliding or rectilinear joint, permits translational motion between the connected links. Each joint has a *joint axis* with respect to which, the motion of joint is described, as shown in Fig. 3.1. In the case of revolute joints, the axis of relative rotation is the joint axis. For the prismatic joint, the axis of relative translational motion is the joint axis. By convention, the z-axis of a coordinate frame is aligned with the joint axis.

The links of a manipulator are numbered outwardly starting from the immobile base as link 0, first moving body as link 1, to the last link out to the free end as link n. Link n is the "*tool*" or "*end-effector*". The joints are numbered, similarly, with joint 1 between link 0 and link 1 and so on, out to the joint n between link $(n-1)$ and link n. The numbering scheme for labelling links and joints is shown in Fig. 3.2 for a 3-DOF manipulator arm which is an open serial kinematic chain of

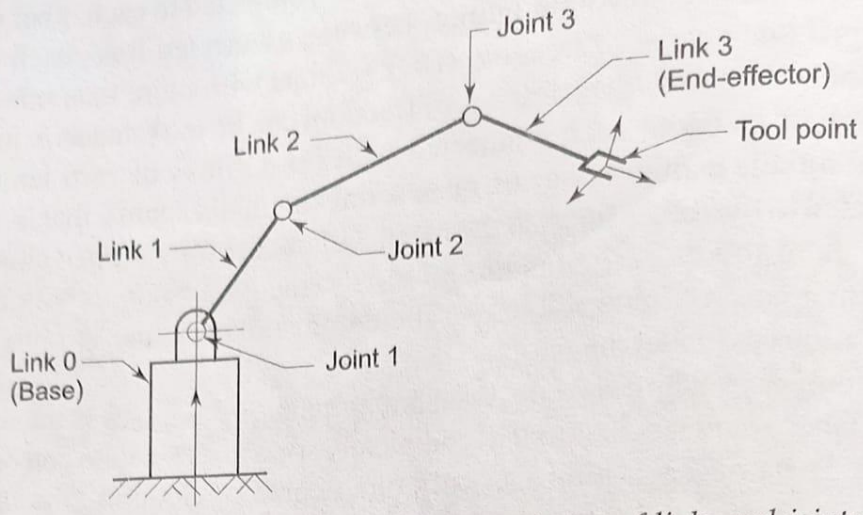


Fig. 3.2 *A 3-DOF manipulator arm—numbering of links and joints*

Description of an object in space requires six parameters—three for position and three for orientation. To position and orient the end-effector in 3-D space, therefore, a minimum of three degrees of freedom are required for positioning and three degrees of freedom for orientation. Typical robotic manipulators have five or six degrees of freedom. A manipulator is considered to be consisting of an *arm* with typically first three links from the base and a *wrist* with the remaining 2 or 3 links. The arm accomplishes the task of reaching the desired position, whereas the wrist helps to orient the end-effector.

4.1 Description of links and joints

The n-DOF robotic manipulator is modelled as a chain of rigid links interconnected by revolute and/or prismatic joints. To describe the position and orientation of a link in space, a coordinate frame is attached to each link, namely, frame $\{i\}$ to link i. The position and orientation of frame $\{i\}$, relative to previous frame $\{i-1\}$, can be described by a homogeneous transformation matrix as discussed in the previous chapter.

In this section, the parameters required to completely specify the position and orientation of links and joints of a manipulator are discussed. Every link of the manipulator is connected to two other links with joints at either end, with the exception of the base and the end-effector, the first and the last link (recall that immobile base is link 0), which have only one joint. Figure 3.3 shows link i of a manipulator with associated joint axes $(i-1)$ and i.

From a geometric viewpoint, the link defines the relative position and orientation of joint axes at its two ends. For the two axes $(i-1)$ and i, there exist a mutual perpendicular, which gives the shortest distance between the two axes. This shortest distance along the common normal is defined as the *link length* and

is denoted as a_i. The angle between the projection of axis $(i-1)$ and axis i, on a plane perpendicular to the common normal AB, is known as the *link twist* and is denoted by α_i. The link twist α_i is measured from axis $(i-1)$ to axis i in the right-hand sense about AB, as shown in Fig. 3.3.

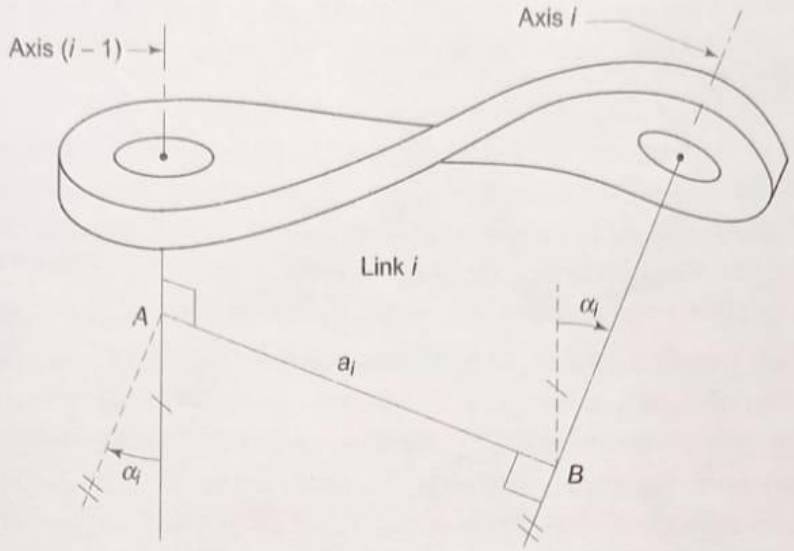


Fig. 3.3 *Description of link parameters a_i and α_i*

These two parameters, a_i and α_i are known as *link parameters* and are constant for a given link. For industrial robots, the links are usually straight, that is, the two joint axes are parallel, giving link length equal to physical link dimension and link twist equal to zero. Another common link geometry is straight link with link twist angle as multiple of $\pi/2$ radians. Sometimes, the link may have a bend such that the axis of joint $(i-1)$ and joint i intersect and in this case the link length of link i is zero although physical link dimension is not zero. Figure 3.4 shows a straight link with link twist of $\pi/2$ radians.

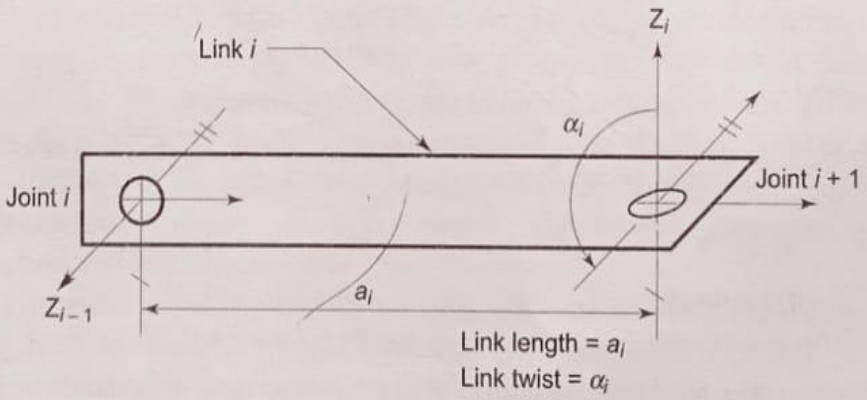


Fig. 3.4 *Link parameters for a straight link with a twist of 90°*

For two links connected by either a revolute or a prismatic joint, the relative position of these links is measured by the displacement at the joint, which is either *joint distance* or *joint angle*, depending on the type of joint. Joint distance (d_i) is

4.2-Translation and rotation Representation

Suppose a rotation by θ is performed, followed by a translation by x_t, y_t. This can be used to place the robot in any desired position and orientation. Note that translations and rotations do not commute! If the operations are applied successively, each $(x, y) \in \mathcal{A}$ is transformed to

$$\begin{pmatrix} x\cos\theta - y\sin\theta + x_t \\ x\sin\theta + y\cos\theta + y_t \end{pmatrix}. \tag{3.33}$$

The following matrix multiplication yields the same result for the first two vector components:

$$\begin{pmatrix} \cos\theta & -\sin\theta & x_t \\ \sin\theta & \cos\theta & y_t \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x\cos\theta - y\sin\theta + x_t \\ x\sin\theta + y\cos\theta + y_t \\ 1 \end{pmatrix}. \tag{3.34}$$

This implies that the 3×3 matrix,

$$T = \begin{pmatrix} \cos\theta & -\sin\theta & x_t \\ \sin\theta & \cos\theta & y_t \\ 0 & 0 & 1 \end{pmatrix}, \tag{3.35}$$

Represents a rotation followed by a translation. The matrix T will be referred to as a *homogeneous transformation matrix*. It is important to remember that T represents a rotation *followed by* a translation (not the other way around). Each primitive can be transformed using the inverse of T, resulting in a transformed solid model of the robot. The transformed robot is denoted by $\mathcal{A}(x_t, y_t, \theta)$, and in this case there are three degrees of freedom. The homogeneous transformation matrix is a convenient representation of the combined transformations; therefore, it is frequently used in robotics, mechanics, computer graphics, and elsewhere. It is called homogeneous because over $\mathbb{R}^3$ it is just a linear transformation without any translation

4.2-Coorinate and Transformation

In almost all fields of science and engineering, it is essential to identify and manipulate mathematical representations of physical, real-world quantities. Robotics is no exception. Intelligent robots build a "mental model" of themselves and the world as they perceive their environments, and they modify those models when interpreting the past and predicting the future. In an abstract sense, these models are simply a collection of numbers and labels, with no *explicit* meaning to the robot. It is the job of a robot engineer to correctly associate numbers with meaning, and correspondingly, meanings with numbers. (If this sounds like mumbo jumbo at the moment, some concrete examples will be offered soon enough!)

Vectors and coordinates

Vectors extend concepts that are familiar to us from working with real numbers RR to other spaces of interest. They also succinctly represent collections of real numbers that have a common meaning like position or direction, or readings from a signal taken at a given time. They make mathematical expressions more compact, which helps us wrap our heads around more difficult concepts.

2D coordinate frames

In the "layman's definition", an n-dimensional *vector* $\mathbf{x}$ is a tuple of real numbers $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$. For now, we will work in $\mathbb{R}^2$. We will use boldface notation only temporarily to help distinguish between vectors and real numbers. In the future, the boldface will typically be dropped.

A 2D position P is represented by a 2-element vector $\mathbf{p} = (p_x, p_y)$ that gives its coordinates relative to axis directions X and Y, offset from a position O where the axes cross, called the origin (Fig. 1). We will also represent vectors in column vector form:

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} \tag{1}$$

for use in matrix-vector products. Both parenthetical and column vector notations are equivalent and interchangeable.

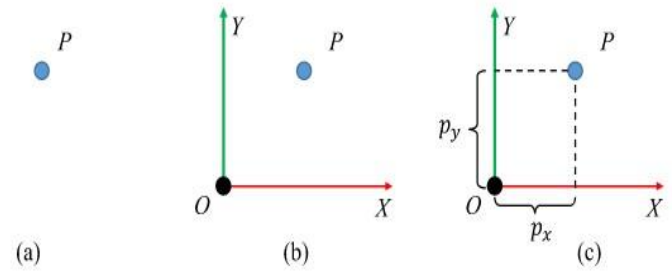


Figure 1. A point P in the plane (a) has no numerical representation until we define a reference coordinate frame (b), which has origin point O and orthogonal coordinate axes X and Y. Its coordinates $\mathbf{p} = (p_x, p_y)$ are respectively the extents of P along X and Y from the origin (c).

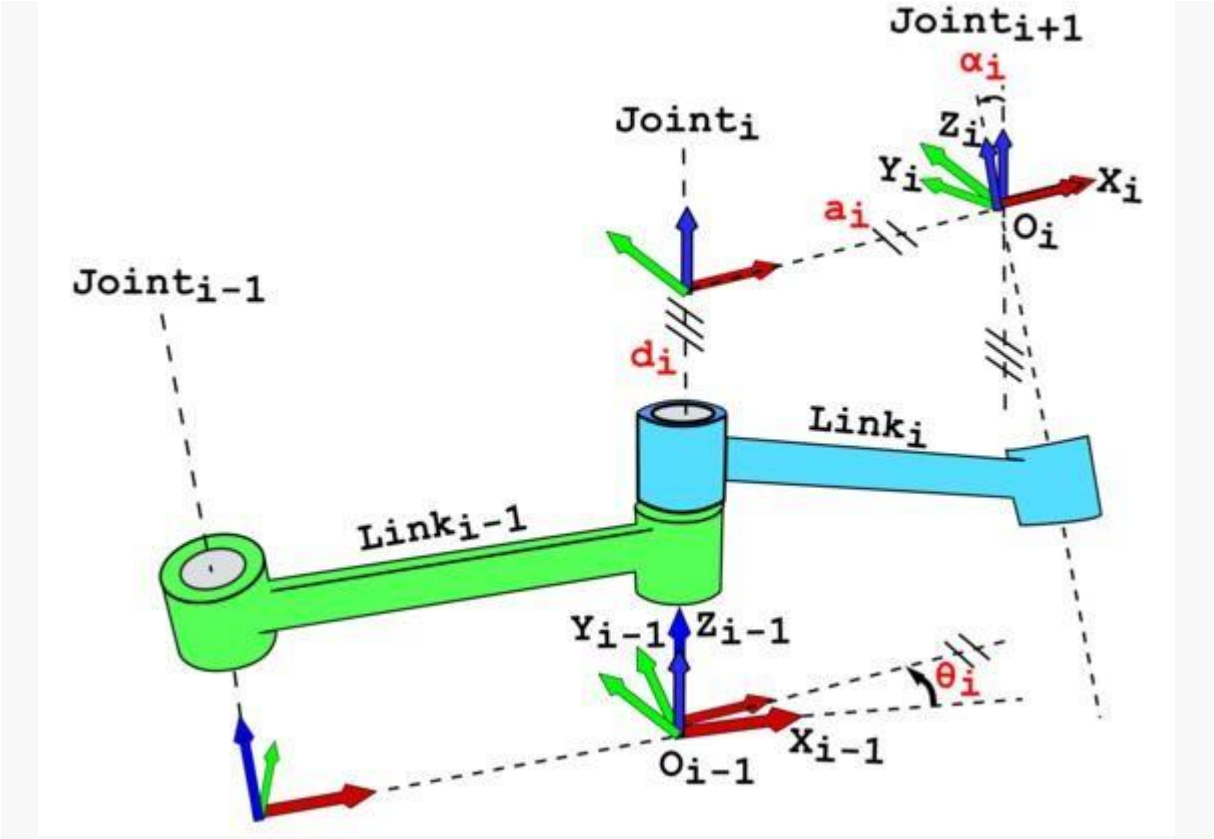
.

4.2 DH parameters

In mechanical engineering, the **Denavit–Hartenberg parameters** (also called **DH parameters**) are the four parameters associated with a particular convention for attaching reference frames tothe links of a spatial kinematic chain, or robot manipulator.

Jacques Denavit and Richard Hartenberg introduced this convention in 1955 in order tostandardize the coordinate frames for spatial linkages

Four parameters



The following four transformation parameters are known as D–H parameters:.[4]

- d: offset along previous z to the common normal
- θ: angle about previous z, from old x to new x
- r: length of the common normal (aka a, but if using this notation, do not confuse with α). Assuming a revolute joint, this is the radius about previous z.
- α: angle about common normal, from old z axis to new z axis

There is some choice in frame layout as to whether the previous x axis or the next x points along the common normal. The latter system allows branching chains more efficiently, as multiple frames can all point away from their common ancestor, but in the alternative layout the ancestor can only point toward one successor. Thus the commonly used notation places each down-chain x axis collinear with the common normal, yielding the transformation calculations shown below.

We can note constraints on the relationships between the axes:

- the x_n-axis is perpendicular to both the z_{n-1} and z_n axes
- the x_n-axis intersects both z_{n-1} and z_n axes
- the origin of joint n is at the intersection of x_n and z_n
- y_n completes a right-handed reference frame based on x_n and z_n

4.2 Jacobian singularity

The manipulator Jacobian $\boldsymbol{J}$, a function of the configuration $\boldsymbol{q}$, may become rank-deficient or singular at certain configuration in Cartesian space. In such cases, the inverse Jacobian does not exist and Eq. (5.66) is not valid. Those manipulator configurations at which $\boldsymbol{J}$ becomes noninvertible are termed as Jacobian singularities and the configuration is itself called singular.

At a singular configuration, the Jacobian matrix $\boldsymbol{J}$ is not full rank; hence, its column vectors are linearly dependent. This means that there exists at least one direction in which the end-effector cannot be moved irrespective of values chosen for joint velocities $\dot{q}_1$ to $\dot{q}_n$.

The study of manipulator singularities is of great significance for the following reasons:

(a) It is not possible to give an arbitrary motion to end-effector; that is, singularities represent configurations at which structural mobility of the manipulator is reduced.
(b) At a singularity, no solution may exist for the inverse Jacobian problem.
(c) In the neighborhood of a singularity, small velocities in the Cartesian space require very high velocities in the joint space. This causes problems when the manipulator is required to track a trajectory that passes close to the singularity.

At a singular configuration, the manipulator loses one or more degrees of freedom. The singular configurations are classified into two categories based on the location of end-effector in the workspace.

(i) *Boundary singularities* occur when the end-effector is on the boundary of the workspace, that is, the manipulator is either fully stretched out or fully retracted. For example, consider the case of a two-link, 2-DOF-planar arm fully stretched out, as shown in Fig. 5.10. In this configuration, two links are in a straight line and the end-effector can be moved only in a direction perpendicular to the two links because it cannot move out of the workspace. Thus, the manipulator loses one degree of freedom. A similar situation will occur with θ_2 equal to 180°. Boundary singularities can be avoided by ensuring that the manipulator is not driven to boundaries of the reachable workspace during its work cycle.

(ii) *Interior singularities* occur when the end-effector is located inside the reachable workspace of the manipulator. These are caused when two or more joint axes become collinear or at specific end-effector configurations. For example, many spherical wrists cause interior singularities due to the lining up of two or more joint axes. These singularities cause serious problems such as path planning and control. They can occur anywhere in the reachable workspace. For certain manipulator configurations, the interior singularity points form a volume within the workspace called *void*.

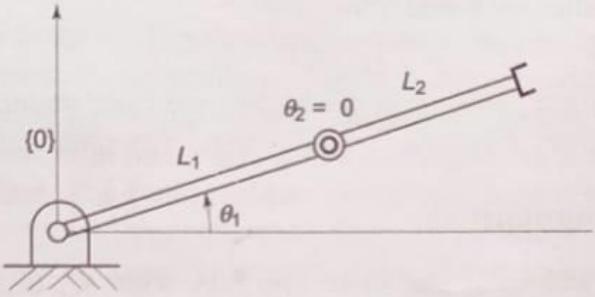


Fig. 5.10 *Two-link, 2-DOF planar manipulator fully stretched out*

In all the situations, it is essential that singularities are avoided. Therefore, one important criterion for a good design of manipulator configuration is to minimize the singularities.

Computation of Singularities

The computation of internal singularities can be carried out by analyzing the rank of the Jacobian matrix. The Jacobian matrix loses its rank and becomes illconditioned at values of joint variables $\boldsymbol{q}$ at which its determinant vanishes, that is,

$$|\boldsymbol{J}| = 0 \tag{5.67}$$

The solutions of Eq. (5.67) give the singular configurations of a manipulator. For manipulators that have 3-DOF arm and 3-DOF spherical wrists, it is possible to simplify the problem of singularity computation by dividing the problem into two separate problems:

- Computation of singularities resulting from motion of first three joints called *arm singularities*.
- Computation of singularities resulting from the motion of wrist joints called *wrist singularities*.

This is achieved by partitioning the Jacobian matrix into four 3×3 sub-matrices as

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \tag{5.68}$$

As the singularities are typical of a configuration and are not dependent on frames chosen for kinematic analysis, the origin of the end-effector frame can be chosen at the end-of-arm point. This will make

$$J_{12} = 0 \tag{5.69}$$

In such a situation computation of determinant is greatly simplified, as,

$$|J| = |J_{11}||J_{22}| \tag{5.70}$$

Hence, for a manipulator with a spherical wrist, the arm singularities are found from

$$|J_{11}| = 0 \tag{5.71}$$

and wrist singularities are found from

$$|J_{22}| = 0 \tag{5.72}$$

Note that this form of Jacobian cannot be used to relate joint velocities and end-effector velocities. It is useful only for singularity computations.

Wrist Singularities

Consider a spherical wrist shown in Fig. 5.11 with θ_4, θ_5, and θ_6 as joint variables.

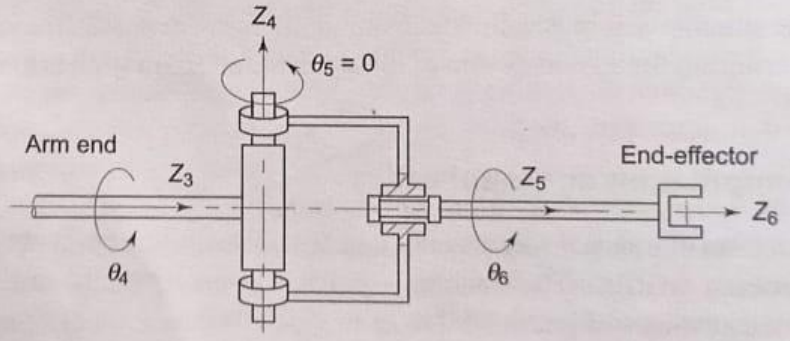


Fig. 5.11 *Spherical wrist at a singularity configuration*

It is known that a singularity occurs whenever two joint axes are aligned. The kinematic structure of the wrist reveals that only axis z_3 and axis z_5 can be aligned. For the configuration in Fig. 5.11 this occurs whenever

4.2 Jacobian static

The relactionship between the joint torque and the end point force vector is derivedusing the principle of virtual work.This is use to determine the joint torques necessary to exert a given end effecter force and movement.

$$\delta W = \tau^T \delta q - \mathcal{F}^T \delta p \tag{5.81}$$

where δp is the $n \times 1$ vector of infinitesimal end-effector displacements $(\delta x_e, \delta \phi_e)$ caused by the endpoint force $\mathcal{F}$. In the above, it is assumed that the joints of the mechanism are frictionless and the joint torques are the net torques that balance the endpoint force $\mathcal{F}$.

From Eq. (5.52), the Jacobian J relates infinitesimal joint displacement δq to infinitesimal end-effector displacement δp as:

$$\delta p = J(q)\, \delta q \tag{5.82}$$

Substituting Eq. (5.82) into Eq. (5.81) and rearranging gives

$$\delta W = (\tau - J(q)^T \mathcal{F})^T \delta q \tag{5.83}$$

According to the principle of virtual work the manipulator mechanism is in static equilibrium, if and only if, the net virtual work is zero for arbitrary virtual displacements, that is,

$$\delta W = 0 \tag{5.84}$$

Substituing it in Eq. (5.83), leads to the result

$$\tau = J(q)^T \mathcal{F} \tag{5.85}$$

Equation (5.85) states that the transpose of the Jacobian matrix transforms the end-effector torque to the corresponding joint torques.

4.3 Dynamic modelling

Two degree of freedom menupulator

The dynamic of a simple menupulator is worked out to illustrate the language euler formulation and to clarify the problem involve in the dynamic modelling.A two DOF manipulator with both rotary joints as shown in the figure.for the manipulator, coordinate frames {0} and {1} joint variable theta 1 and theta 2, link length L1 and L2and mass of link m1 and m2 respectively.The liner angular velocity are v1,v2,theta 1and theta2

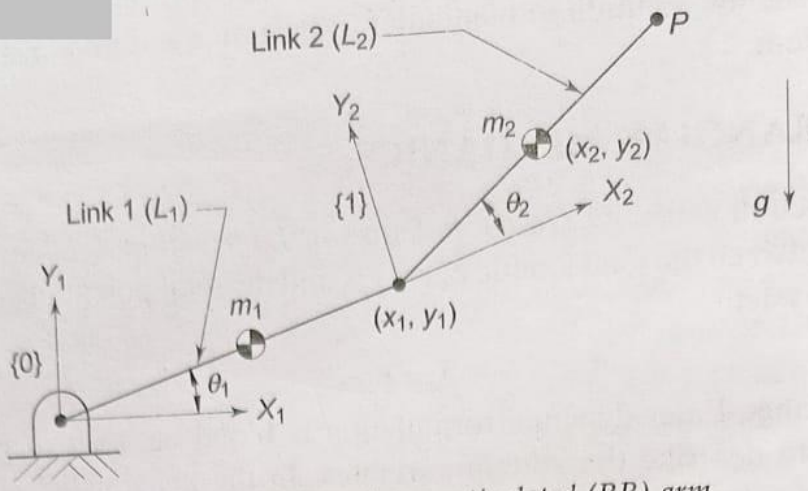


Fig. 6.1 *A 2-DOF planar articulated (RR) arm*

The Lagrangian requires kinetic and potential energies of the manipulator. The kinetic energy of a rigid body (a link), can be expressed as:

$$\mathcal{K} = \frac{1}{2} mv^2 + \frac{1}{2} I\omega^2 \tag{6.3}$$

where v is the linear velocity , ω is the angular velocity, m is the mass, and I is the moment of inertia of the rigid body at its center of mass.

Thus, the kinetic energy for the link 1 with the linear velocity $v_1 = \frac{1}{2} L_1\dot{\theta}_1$, angular velocity $\omega_1 = \dot{\theta}_1$, moment of inertia $I_1 = \frac{1}{12} m_1 L_1^2$, and mass m_1 is

$$\mathcal{K}_1 = \frac{1}{2} m_1 v_1^2 + \frac{1}{2} I_1 \omega_1^2 = \frac{1}{8} m_1 L_1^2 \dot{\theta}_1^2 + \frac{1}{24} m_1 L_1^2 \dot{\theta}_1^2 = \frac{1}{6} m_1 L_1^2 \dot{\theta}_1^2 \tag{6.4}$$

and its potential energy is

$$\mathcal{P}_1 = \frac{1}{2} m_1 g L_1 \sin\theta_1 \tag{6.5}$$

where g is the magnitude of acceleration due to gravity in the negative y-axis direction.

For the second link, link 2, the Cartesian position coordinates (x_2, y_2) of the center of mass of link are:

$$x_2 = L_1 \cos\theta_1 + \frac{1}{2} L_2 \cos(\theta_1 + \theta_2)$$
$$y_2 = L_1 \sin\theta_1 + \frac{1}{2} L_2 \sin(\theta_1 + \theta_2) \tag{6.6}$$

Differentiating Eq. (6.6) gives the components of velocity of link 2 as

$$\dot{x}_2 = -L_1 \sin\theta_1 \dot{\theta}_1 - \frac{1}{2} L_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)$$
$$\dot{y}_2 = L_1 \cos\theta_1 \dot{\theta}_1 + \frac{1}{2} L_2 \cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2) \tag{6.7}$$

From these components, the square of the magnitude of velocity of the end of link 2 is

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2$$

or
$$v_2^2 = L_1^2 S_1^2 \dot{\theta}_1^2 + \frac{1}{4} L_2^2 S_{12}^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + L_1 L_2 S_1 S_{12} (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2)$$
$$+ L_1^2 C_1^2 \dot{\theta}_1^2 + \frac{1}{4} L_2^2 C_{12}^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + L_1 L_2 C_1 C_{12} (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2)$$

Simplifying

$$v_2^2 = L_1^2 \dot{\theta}_1^2 + \frac{1}{4} L_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + L_1 L_2 C_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \tag{6.8}$$

where $C_i = \cos\theta_i$, $S_i = \sin\theta_i$, $C_{12} = \cos(\theta_1 + \theta_2)$ and $S_{12} = \sin(\theta_1 + \theta_2)$.

Thus, the kinetic energy of link 2 with $\omega_2 = \dot{\theta}_1 + \dot{\theta}_2$ and $I_2 = \frac{1}{12} m_2 L_2^2$ is

$$\mathcal{K}_2 = \frac{1}{2} m_2 v_2^2 + \frac{1}{2} I_2 \omega_2^2$$
$$= \frac{1}{2} m_2 \left[L_1^2 \dot{\theta}_1^2 + \frac{1}{4} L_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + L_1 L_2 C_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \right]$$
$$+ \frac{1}{24} m_2 L_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \tag{6.9}$$
$$= \frac{1}{2} m_2 L_1^2 \dot{\theta}_1^2 + \frac{1}{6} m_2 L_2^2 (\dot{\theta}_1^2 + \dot{\theta}_1^2 + 2\dot{\theta}_1 \dot{\theta}_2) + \frac{1}{2} m_2 L_1 L_2 C_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2)$$

The potential energy of link 2, from Eq. (6.6), is

$$\mathcal{P}_2 = m_2 g L_1 S_1 + \frac{1}{2} m_2 g L_2 S_{12} \tag{6.10}$$

4.3 Equetation of motion

t is common to see the equations of motion of manipulators or humanoids reminded in the preamble of research papers. Their origin can be tracked down in textbooks, where they are derived

from Newtonian or Lagrangian mechanics. In this note, we will see what are the equations of motion for manipulators and legged robots, and how to compute them in practice.

Definition

The equations of motion of a physical system describe its *motion* as a function of *time* and optional *control inputs*. In their general form, they are written:

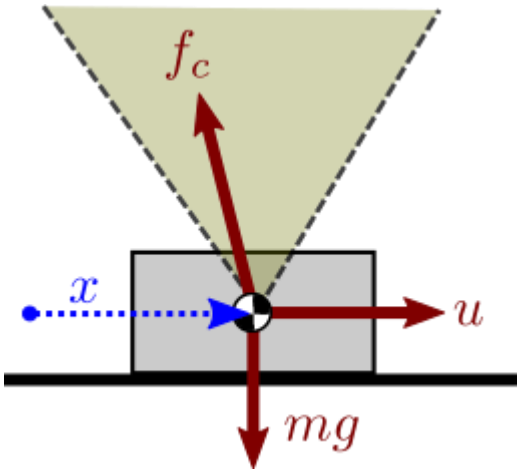
$$F(q(t),\dot{q}(t),\ddot{q}(t),u(t),t)=0,F(q(t),\dot{q}(t),\ddot{q}(t),u(t),t)=0,$$

where

- t is the time variable,
- q is the vector of generalized coordinates, for instance the vector of joint-angles for a manipulator,
- $\dot{q}$ is the first time-derivative (velocity) of q,
- $\ddot{q}$ is the second time-derivative (acceleration) of q,
- u is the vector of control inputs.

These equations provide a mapping between the *control space*, the commands that we send to actuators, and the *state space* of robot motions

Imagine a rigid block sliding on a table, which we can see as a 1-DOF (one degree of freedom) system with coordinate x. The operator applies a horizontal force u that pushes the block forward.



If the force is high enough to overcome friction, applying Newton's second law of motion (and Coulomb's model of sliding friction)

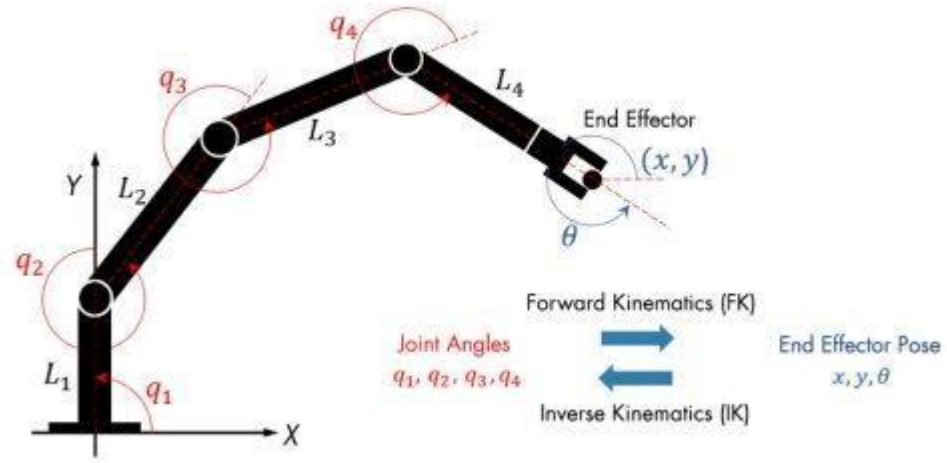
4.3 Euler lagrange formulation

The Euler lagrange formulation is a systematic procedure for obtaining the dynamic model of an n-DOF manupulator.The n-DOF open kynamatic chain serial link manipulator has n joint position or displacement variable q=[q1….qn]^T . The LE formulation establishes the relation between the joint positions,velocities,acceleration and the generalised turque applied to the manipulator.The generalised torque are the non- conservative torques contributed by joints,joint friction forces,and indused joint torque.The indused joint torque are thetorques at the joints due to contact or interaction of the end effector with the environment.

4.4 Inverse kinematics-Manupulation workspace

Kinematics is the study of motion without considering the cause of the motion, such as forces and torques. Inverse kinematics is the use of kinematic equations to determine the motion of a robot to reach a desired position. For example, to perform automated binpicking, a robotic arm used in a manufacturing line needs precise motion from an initial position to a desired position between bins and manufacturing machines. The grasping end of a robot arm is designated as the end-effector. The robot configuration is a list of joint positions that are within the position limits of the robot model and do not violate anyconstraints the robot has.

Given the desired robot's end-effector positions, inverse kinematics (IK) can determinean appropriate joint configuration for which the end-effectors move to the target pose.



Configuring the joint positions of a robot using forward or inverse kinematics.

Once the robot's joint angles are calculated using the inverse kinematics, a motion profile can be generated using the Jacobian matrix to move the end-effector from theinitial to the target pose. The Jacobian matrix helps define a relationship between therobot's joint parameters and the end-effector velocities.

In contrast to forward kinematics (FK), robots with multiple revolute joints generally have multiple solutions to inverse kinematics, and various methods have been proposed according to the purpose. In general, they are classified into two methods, one that is

analytically obtained (i.e., analytic solution) and the other that uses numericalcalculation.

4.3 Manupulation Workspace

The work space of manipulator is define is the volume of space in which the manipulator is able to lacate its end effectot.The workspace gets specified by the existence or non existence of solution to the inverse problem.The region that can be reached by the originof the end effector frame with at least one orientation is called the reachable workspace(RWS).If a point in workspace can be reached only one orientaion , the end effector is very poor and it is not possible to do my pratical work satisfactorily with just one fixed orirenatation. The space where the end effector can reach every point from all orientation is called dexterous wprkspace(DWS).It is obvious that the dexterous workspace is either smaller or same as the reachable workspace.

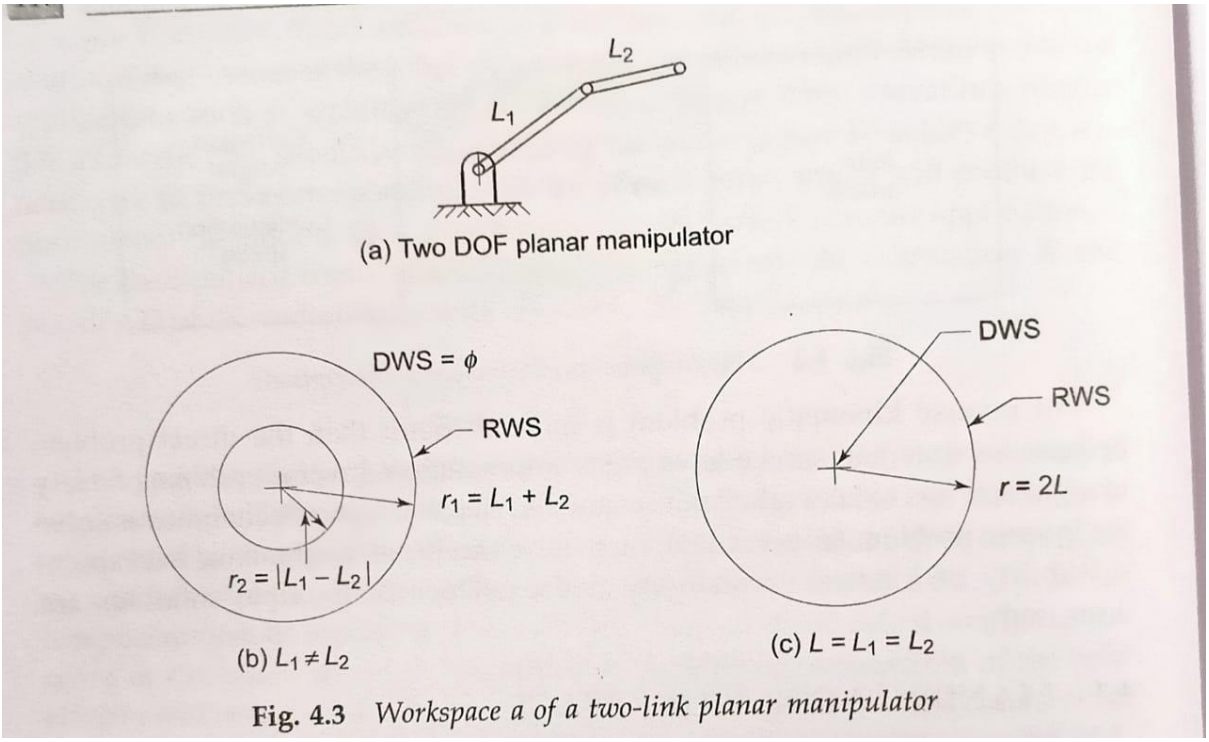
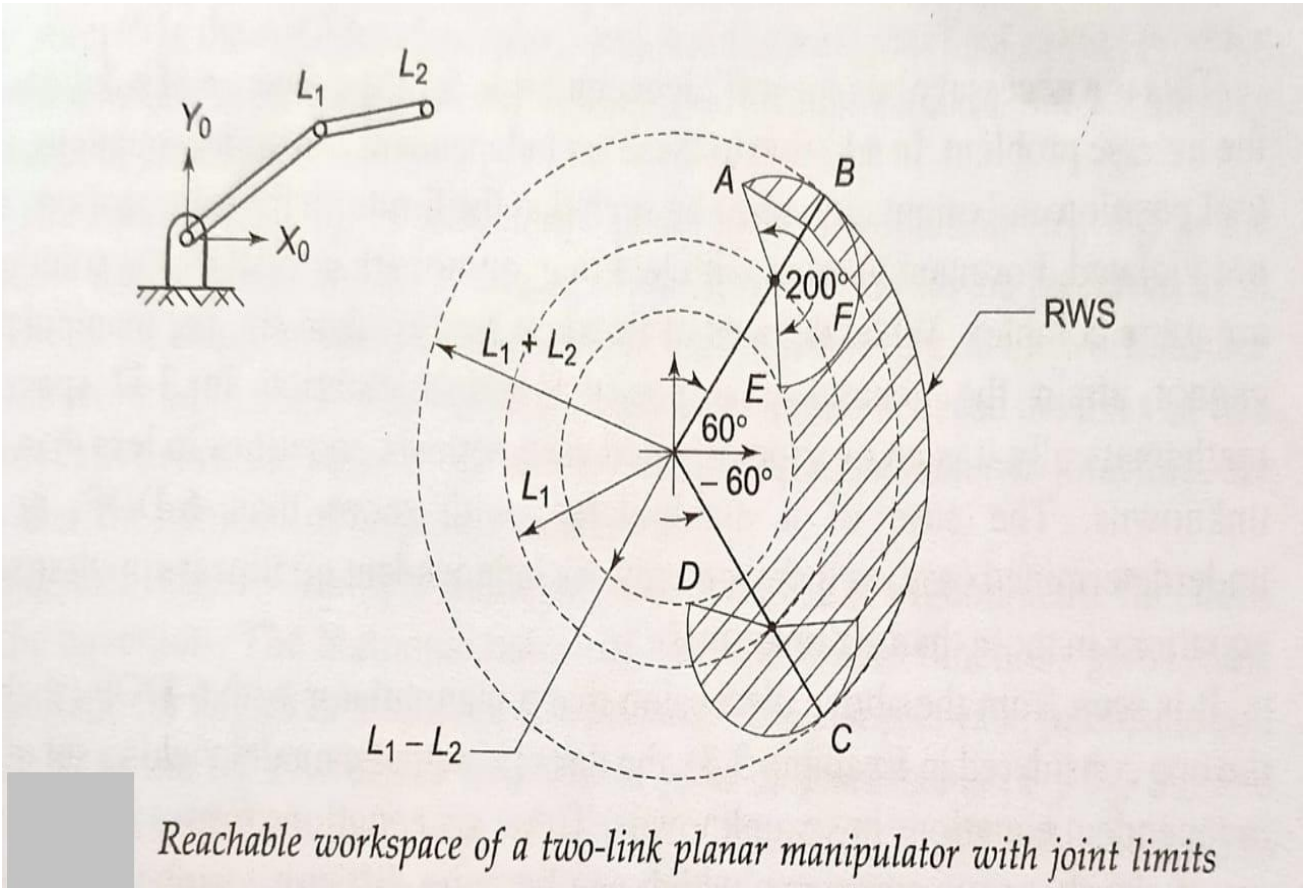


Fig. 4.3 *Workspace a of a two-link planar manipulator*

4.4 Solvadility of inverse kinematic model

Inverse kinematic is complex because the solluction is to be found for non linear simoltanious equation, involving transcendental functions. The number of simultaneous equation is also generally more than the number of unknown, making some of the equeations mutually dependent.This condition lead to the possibility of multiple solution.The exestance of solution , multiple solution and method of solution.



Reachable workspace of a two-link planar manipulator with joint limits

Short questions

Q1 Define kinamatic model

A **kinematic model** is a mathematical description of the robot: its functional dimensions and DoF. It describes the robot's workspace, its positional capabilities and constraints. Most often used for describing robot **kinematics**, which is also thecase in this method, is the modified Denavit–Hartenberg (MDH) notation.

Q 2 Define link and joints

A mechanical linkage is an assembly of bodies connected to manage forces and movement.The movement of a body, or **link**, is studied using geometry so the **link** is considered to be rigid. The connections between **links** are modeled as providing ideal movement, pure rotation or sliding for example, and are called **joints**.

Q 3 Define transalation and rotation

A **rotation** is the turning of a figure or object around a fixed point. And a **translation** is a scenario where every point in a figure is moved the exact same distance and in the sameexact direction, without being rotated, reflected, or resized.

Q 4 State DH parameter

In mechanical engineering, the Denavit–Hartenberg **parameters** (also called **DH parameters**) are the four **parameters** associated with a particular convention for attachingreference frames to the links of a spatial kinematic chain, or robot manipulator.

The **four parameters** a_i, α_i, d_i, and θ_i are generally given the names link length, link twist, link offset, and joint angle, respectively.

Q 5 Define Jacobian singularity

A **singularity** occurs when the joint velocity in joint space becomes infinite to maintain Cartesian. velocity. It shows us where the continuity in joint space breaks down as related toCartesian space. A. **singularity** occurs whenever the determinant of the **Jacobian** is 0

Q 6 What is dynamic modelling

Dynamic Modeling represents the temporal aspects of a system, capturing the control elements through which the behavior of objects can be understood over time..... It is necessary to **model** only those objects with a definite lifecycle or those which exhibitsignificant behavior.

Q 7 State Euler-lagrange formulation

In the calculus of variations and classical mechanics, the Euler-Lagrange equations[1] is a systemof second-order ordinary differential equations whose solutions are stationary points of the given action functional. The equations were discovered in the 1750s by Swiss mathematician Leonhard Euler and Italian mathematician Joseph-Louis Lagrange.

Because a differentiable functional is stationary at its local extrema, the Euler–Lagrange equation is useful for solving optimization problems in which, given some functional, one seeks the

function minimizing or maximizing it. This is analogous to Fermat's theorem in calculus, statingthat at any point where a differentiable function attains a local extremum its derivative is zero.

Q 8 What is inverse Kinamatics

In computer animation and robotics, **inverse kinematics** is the mathematical process of calculating the variable joint parameters needed to place the end of a kinematic chain, such as a robot manipulator or animation character's skeleton, in a given position and orientation relative tothe start of the chain. Given joint parameters, the position and orientation of the chain's end, e.g. the hand of the character or robot, can typically be calculated directly using multiple applications of trigonometric formulas, a process known as forward kinematics. However, the reverse operation is, in general, much more challenging.

Long questions

1-State and explain different kinematic model2-

Describe different types of link and joints

3-State and explain Jacobian singularity4-

State and explain DH parameters

5-Sketch the approximate reachable workspace and the dextrous workspace of the two-DOF planar manipulator.

Chapter-5

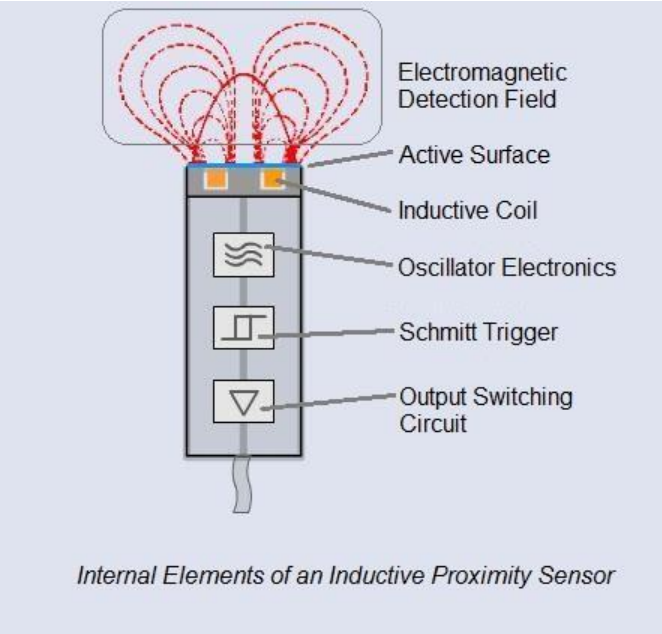
Sensors and vision system

5.1 CONTACT AND PROXIMITY SENSOR

Proximity sensors are sensors that detect movement/presence of objects without physical contact and relay that information captured into an electrical signal. It can also be defined as a proximity switch, a definition given by the Japanese Industrial Standards (JIS) to all contactless detecting sensors

Types of proximity sensor

Inductive Proximity Sensors



Internal Elements of an Inductive Proximity Sensor

Image Credits: Automation Insights

Inductive proximity sensors are contactless sensors used to only detectmetal objects. It's based on the law of induction, driving a coil with an oscillator once a metallic object approaches it.

It has two versions and comprises of 4 main components:

Versions:

Unshielded: Electromagnetic field generated by the coil is unrestricted, allowing for wider and greater sensing distances

Shielded: Electromagnetic field generated is concentrated in the front,where sides of the sensor coil are covered up

Components:

It comprises of 4 main components as seen in the picture; Coil, Oscillator, Schmitt Trigger, and output switching circuit

Common applications:

- Industrial usages
  - Production automation machines that count products, product transfers
- Security usages
  - Detection of metal objects, armory, land mines, etc.

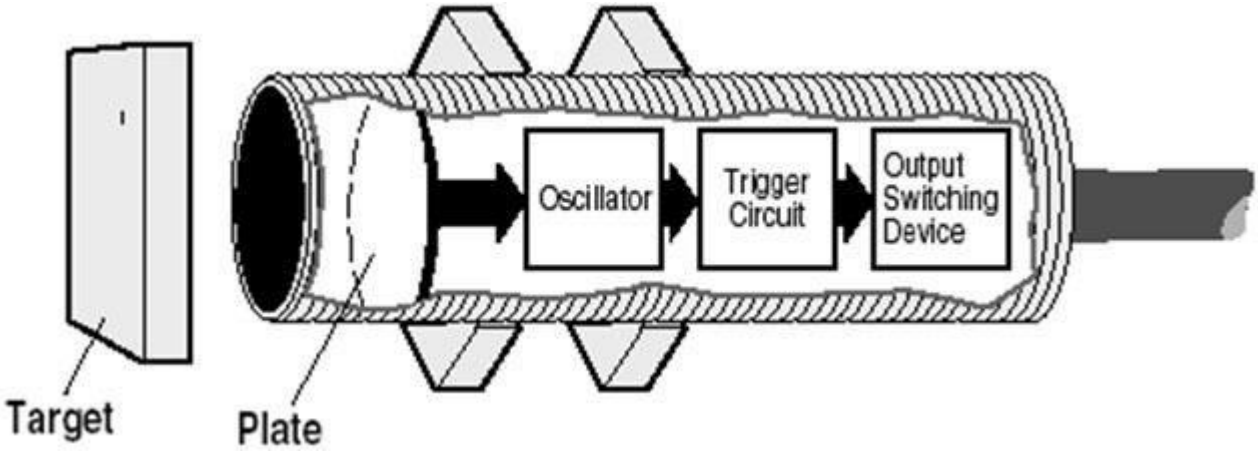
Advantages of inductive proximity sensors

- Contactless detection
- Environment adaptability; resistant to common conditions seen in industrialareas such as dust and dirt
- Capable and versatile in metal sensing
- Considerably cheap when it comes to price
- No moving parts, ensuring a longer service life

Disadvantages of inductive proximity sensors

- Lack in detection range, averaging a max range of up to 80mm
- Can only detect metal objects
- Performance can be affected by external conditions; extreme temperatures,cutting fluids or chemicals

Capacitive Proximity Sensors



Capacitive proximity sensors are contactless sensors that detect both metallic and non-metallic objects, including liquid, powders, and granular. Itoperates by detecting a change in capacitance.

Similarly to inductive sensors, it consists of an oscillator, Schmitt triggerand output switching circuit. The only difference is it comprises of 2 charging plates (1 internal, 1 external) for capacitation:

- Internal plate connected to the oscillator
- External plate (sensor electrodes) used as the sensing surface

Common applications:

- Industrial usages
  - Production automation machines that count products, product transfers
  - Filling processes, pipelines, inks, etc.
  - Fluid level, composition, and pressure
- Moisture control
- Non-invasive content detection
- Touch applications

Advantages of Capacitive proximity sensors

- Contactless detection
- A wide array of materials able to be detected
- Able to detect objects through non-metallic walls with its wide sensitivityband
- Well-suited to be used in an industrial environment
- Contains potentiometer that allows users to adjust sensor sensitivity, suchthat only wanted objects will be sensed
- No moving parts, ensuring a longer service life

Disadvantages of Capacitive proximity sensors

- Relative low range, though incremental increase from inductive sensors
- Higher price as compared to inductive sensors

Position sensors are devices that can detect the movement of an object or determine its relative position measured from an established reference point.These types of sensors can also be used to detect the presence of an objector its absence.

5.1 Position,velocity,force,tactile sensor

Position Sensors

The overall intent of a position sensor is to detect an object and relay its positionthrough the generation of a signal that that provides positional feedback. This feedback can then be used to control automated responses in a process, sound alarms, or trigger other activity as dictated by the specific application. Generally speaking, position sensors may be divided into three broad classes that

include linear position sensors, rotary position sensors, and angular position sensors. There are several specific technologies that can be employed to achievethis result, and the different types of position sensors reflect these underlying technologies.

The primary types of position sensors include the following:

- Potentiometric Position Sensors (resistance-based)
- Inductive Position Sensors
- Eddy Current-Based Position Sensors
- Capacitive Position Sensors

Potentiometric Position Sensors

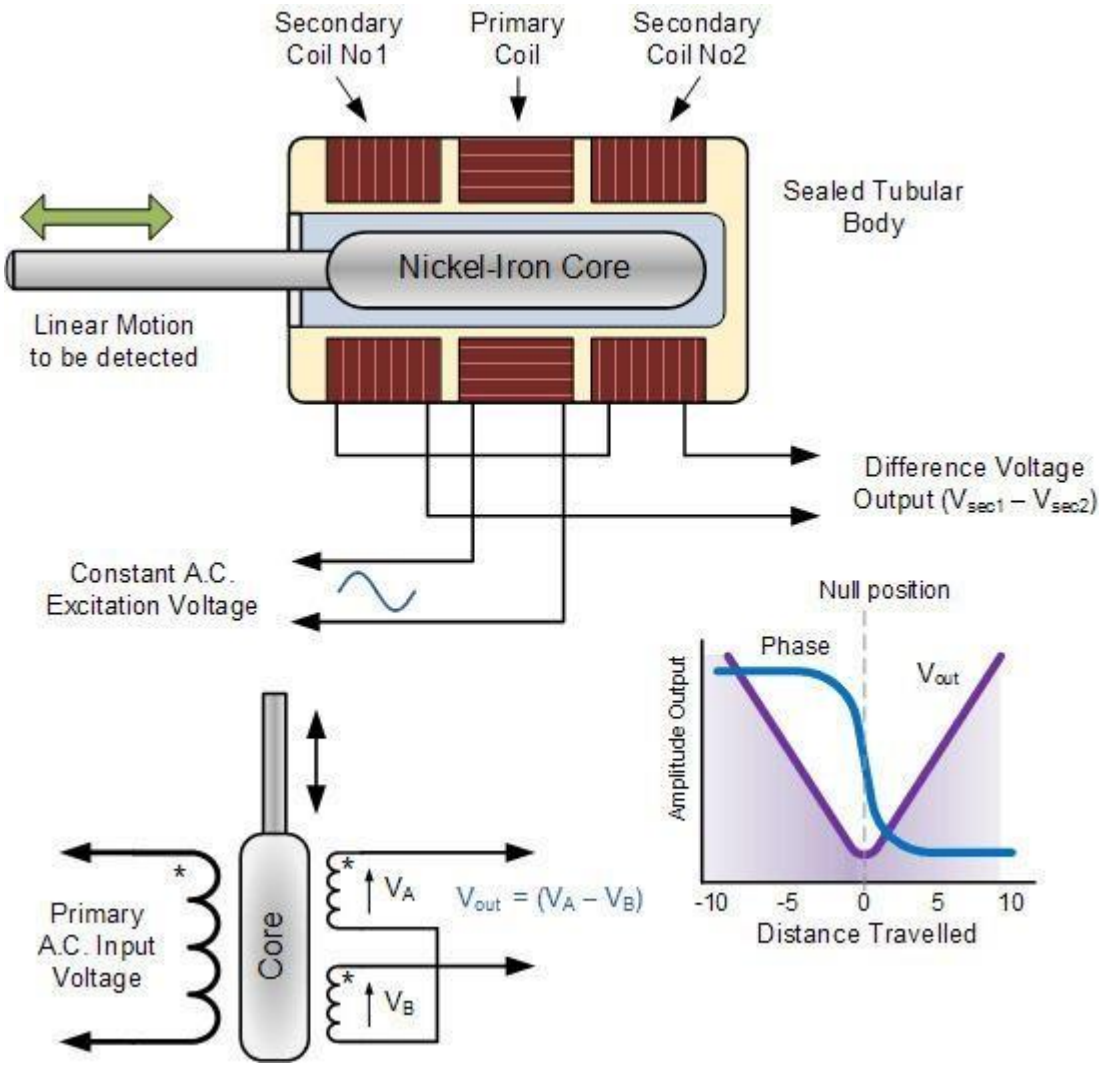
Potentiometric Position Sensors are resistance-based sensors that use a resistive track with a wiper that is attached to the object whose position is being monitored. Movement of the object causes the wiper to change its position along the resistance track and therefore alter the measured resistance value between the wiper position and the end of the track. In this manner, the measured resistance can be used as anindicator of the object's position. This is accomplished by using a voltage divider where a fixed voltage is applied across the ends of the resistance track, and the measured voltage from the wiper position to one end of the track yields a value that is proportional to the wiper position. This approach works for both linear displacements and rotary displacements.

Potentiometer styles used for potentiometric position sensors include wirewound, cermet, or plastic (polymer) film. These types of position sensors offer relatively lowcost, but also suffer from low accuracy and repeatability. In addition, the size limitation of the device by design constrains the range over which the positional change can be measured.

Inductive Position Sensors

Inductive position sensors detect the position of an object by changes in the characteristics of a magnetic field that is induced in the coils of the sensor. One type is called an LVDT, or Linear Variable Differential Transformer. In an LVDT position sensor, three separate coils are wound on a hollow tube. One of these is a primary coil, and the other two are secondary coils. They are wired electrically in series, but the phase relationship of the secondary coils is 180^{o} out of phase with respect to the primary coil. A ferromagnetic core or armature is placed inside of the hollow tube, and the armature is connected to the object whose position is being measured. An excitation voltage signal is applied to the primary coil which induces an EMF in the secondary coils of the LVDT. By measuring the voltage difference between the two secondary coils, the relative position of the armature (and thus the object to which it is attached) can be determined. When the armature is exactly centered in the tube, the EMFs cancel out resulting in no voltage output. But as the armature moves off the null position, the voltage and its polarity change. Therefore, the amplitude of the voltage along with its phase angle serves to provide information that reflects not onlythe amount of movement away from the center (null) position but also its direction.

Figure 1 below illustrates the operation of a Linear Variable Differential Transformer,showing the translation of the voltage measurement into an indication of position.



Operation of an LVDT Inductive Position Sensor

These types of position sensors provide good accuracy, resolution, have high sensitivity, and offer good linearity across the sensing range. They are also frictionless and can be sealed for use in conditions where there might be exposure tothe elements.

While LVDTs function to track linear movement, an equivalent device called an RVDT (for Rotary Voltage Differential Transformer) can provide tracking of the rotational position of an object. The RVDT functions identically to the LVDT andvaries only in the specifics of their construction.

Eddy Current-Based Position Sensors

Eddy currents are induced currents that occur in a conductive material that is in the presence of a changing magnetic field and are a result of Faraday's law of induction.

These currents flow in closed loops and in turn, result in the generation of asecondary magnetic field.

If a coil is energized by ay an alternating current to generate a primary magnetic field, the presence of a conductive material brought near the coil can be sensed due to the interaction of the secondary field generated by the Eddy currents, which impacts the impedance of the coil. So, the change in the coil impedance can be usedto establish the distance of an object from the coil.

Eddy current position sensors work with electrically conductive objects. Most Eddy current sensors function as proximity sensors, designed to establish that an object has approached the sensor location. They are limited as position sensors because they are omnidirectional, meaning that they can establish the relative distance of theobject from the sensor but not the direction of the object relative to the sensor.

Capacitive Position Sensors

Capacitive position sensors rely on detecting a change in capacitance value to establish the position of the object being measured. Capacitors consist of two platesseparated from each other with a dielectric material between the plates. There are two general methods that are used to detect the position of an object using a capacitive position sensor:

1. By altering the dielectric constant of the capacitor
2. By altering the overlapping area of the capacitor plates

In the first case, the object being measured is attached to the dielectric material, whose position relative to the capacitor plates changes as the object moves. As the dielectric material is shifted, the effective dielectric constant of the capacitor changesbeing the resultant of a partial area of dielectric material and the balance being the dielectric constant of air. This approach provides a linear variation in the capacitancevalue with respect to the object's relative position.

In the second case, instead of attaching the object to the dielectric material, it is connected to one of the capacitor plates. Therefore, as the object moves its position,the overlapping area of the capacitor plates changes, which again changes the capacitance value.

The principle of varying capacitance to measure an object's position can be appliedto motion in both linear and angular directions.

Velocity sensors in_robotics

VELOCITY SENSORS

Velocity Sensor: A velocity or speed sensor measures consecutive position measurements at known intervals and computes the time rate of change in the position values.

Velocity Sensors: 1) Tachometers 2) LSV 3) Piezoelectric Sensors 4) AccelerometerSensor

Tachometers: A most important device that is used to provide velocity feedback is the tachometer. It is also known as rpm gauge, and revolution counter. A tachometeris employed in a motor to calculate the rotational speed of a shaft. The output is displayed as RPM (revolution per minute) in an analog device.

The two common types of a tachometer are: 1) AC tachometer 2) DC tachometer

AC tachometer: It possesses primary and secondary stators with fixed windings, and a rotor with permanent magnet. If the rotor is stationary, a constant output voltage will be obtained. If the rotor is moving, proportional to the speed of a rotor isinduced. This type of tachometer cannot provide information of direction with only one output winding.

DC tachometer: DC tachometer is the most commonly used instrument in the robotics. It is a DC generator implemented to provide an output voltage that is proportional to the angular velocity of the armature. In this mechanism, the rotor androtational part will be attached directly. It has a stationary device called as commutator, which is connected with the split slip rings.

It is used for picking the induced output signal from the rotating coil.

Laser surface velocimeter: A laser surface velocimeter (LSV) is a non- contactoptical speed sensor measuring velocity and length on moving surfaces. Lasersurface velocimeters use the laser Doppler principle to evaluate the laser light scattered back from a moving object

LSV is based on Doppler effect (or Doppler shift) that is the change in frequency ofa wave (or other periodic event) for an observer moving relative to its source.

They are widely used for process and quality control in industrial production processes.
Piezoelectric Sensor: A piezoelectric sensor is a device that uses the piezoelectric effect, to measure changes in velocity, pressure, acceleration, temperature, strain, or force by converting them to an electrical charge. The prefixpiezo- is Greek for 'press'

Principle of operation: The way a piezoelectric material is cut produces three mainoperational modes: 1) Transverse 2) Longitudinal 3) Shear.

Two main groups of materials are used for piezoelectric sensors: piezoelectric ceramics and single crystal materials. Schematic of piezoelectricsensor

Piezoelectric sensors are versatile tools for the measurement of various processes.They are used for quality assurance, process control, and for research and development in many industries

ACCELEROMETER SENSOR An accelerometer is a device that measures proper acceleration(Time rate of change of velocity). They are typically used in one of threemodes: 1) As an inertial measurement of velocity and position; 2) As a sensor of inclination, tilt, or orientation in 2 or 3 dimensions, as referenced from the acceleration of gravity (1 g = 9.8m/s2)

Tactile sensor

A **tactile sensor** is a device that measures information arising from physical interaction with its environment. Tactile sensors are generally modeled after the biological sense of cutaneous touch which is capable of detecting stimuli resulting from mechanical stimulation, temperature, and pain (although pain sensing is not common in artificial tactile sensors). Tactile sensors areused in robotics, computer hardware and security systems. A common application of tactile sensors is in touchscreen devices on mobile phones and computing.

Tactile sensors may be of different types including piezoresistive, piezoelectric, capacitive and elastoresistive sensors.

Tactile sensors appear in everyday life such as elevator buttons and lamps which dim or brightenby touching the base. There are also innumerable applications for tactile sensors of which most people are never aware.

Sensors that measure very small changes must have very high sensitivities. Sensors need to be designed to have a small effect on what is measured; making the sensor smaller often improves this and may introduce other advantages. Tactile sensors can be used to test the performance ofall types of applications. For example, these sensors have been used in
the manufacturing of automobiles (brakes, clutches, door seals, gasket), battery lamination,bolted joints, fuel cells etc.

Tactile imaging, as a medical imaging modality, translating the sense of touch into a digital imageis based on the tactile sensors. Tactile imaging closely mimics manual palpation, since the probe of the device with a pressure sensor array mounted on its face acts similar to human fingers during clinical examination, deforming soft tissue by the probe and detecting resulting changes inthe pressure pattern.

Robots designed to interact with objects requiring handling involving precision, dexterity, or interaction with unusual objects, need sensory apparatus which is functionally equivalent to a human's tactile ability. Tactile sensors have been developed for use with robots. Tactile sensors can complement visual systems by providing added information when the robot begins to grip anobject. At this time vision is no longer sufficient, as the mechanical properties of the object cannot be determined by vision alone. Determining weight, texture, stiffness, center of
mass, coefficient of friction, and thermal conductivity require object interaction and some sort oftactile sensing.

Several classes of tactile sensors are used in robots in warfare and engineering Some methodsfor simultaneous localization and mapping are based on tactile sensors.

5.2 Introduction to cameras

1. Camera ♣ A camera is an optical instrument that records images that can be stored directly, transmitted to another location, or both. ♣ These images may be still photographs or moving images such as videos or movies. ♣ The term camera comes from the word camera obscura (Latin for "dark chamber"), an early mechanism for projecting images. ♣ The modern camera evolved from the camera obscura & functioning of the camera is very similar to the functioning of the human eye.
2. History ♣ The history of the camera can be traced much further back than the introduction of photography. ♣ Cameras evolved from the camera obscura, and continued to change through many generations of photographic technology, including Daguerre types, callow types, dry plates, film, and digital cameras.
3. History: Camera Obscura Photographic cameras were a development of the camera obscura, a device dating back to the ancient Chinese[1] and

ancient Greeks,[2][3] which uses a pinhole or lens to project an image of the scene outside upside- down onto a viewing surface.

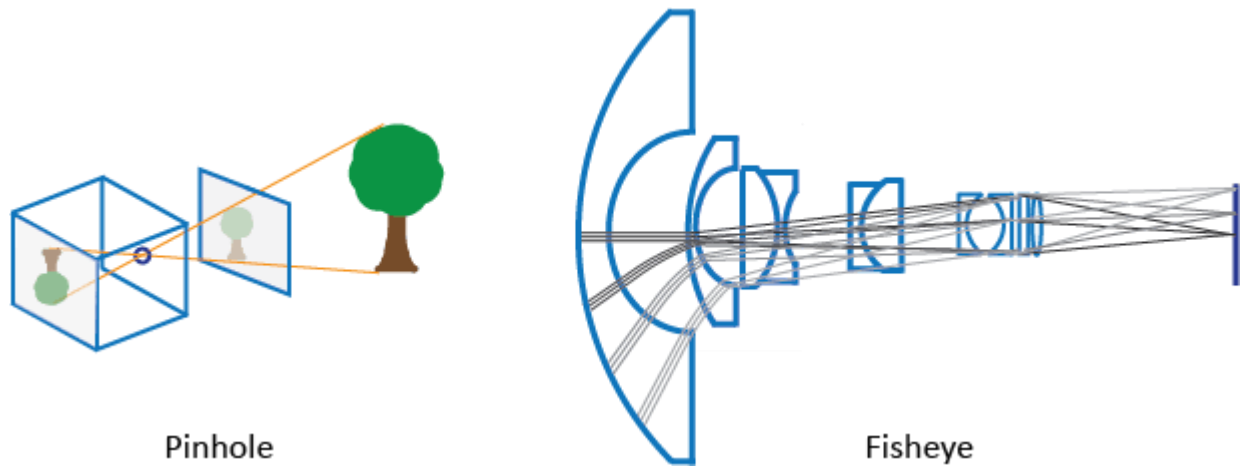
4. Further Developments ♣ The first partially successful photograph of a camera image was made in approximately 1816 by Nicéphore Niépce,[6][7] using a very small camera of his own making and a piece of paper coated with silver chloride, which darkened where it was exposed to light. ♣ After Niépce's death in 1833, his partner Louis Daguerre continued to experiment and by 1837 had created the first practical photographic process, which he named the daguerreotype and publicly unveiled in 1839.

5.2 Camera calibration

Geometric camera calibration, also referred to as *camera resectioning*, estimates the parameters of a lens and image sensor of an image or video camera. You can use these parameters to correct for lens distortion, measure the size of an object in worldunits, or determine the location of the camera in the scene. These tasks are used in applications such as machine vision to detect and measure objects. They are also used in robotics, for navigation systems, and 3-D scene reconstruction

Camera Models

The Computer Vision Toolbox™ contains calibration algorithms for the pinhole camera model and the fisheye camera model. You can use the fisheye model with cameras up to afield of view (FOV) of 195 degrees.



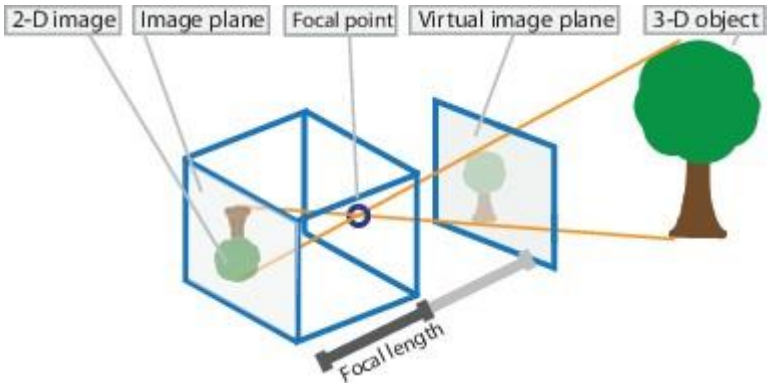
The pinhole calibration algorithm is based on the model proposed by Jean-Yves Bouguet .The model includes, the pinhole camera model and lens distortion .The pinhole camera model does not account for lens distortion because an ideal pinhole camera does nothave a lens. To accurately represent a real camera, the full camera model used by the algorithm includes the radial and tangential lens distortion.

Because of the extreme distortion a fisheye lens produces, the pinhole model cannot model a fisheye camera. For details on camera calibration using the fisheye model, see Fisheye Calibration Basics.

Pinhole Camera Model

A pinhole camera is a simple camera without a lens and with a single small aperture. Light rays pass through the aperture and project an inverted image on the opposite side of the

camera. Think of the virtual image plane as being in front of the camera and containing theupright image of the scene.



The pinhole camera parameters are represented in a 4-by-3 matrix called the *camera matrix*. This matrix maps the 3-D world scene into the image plane. The calibration algorithm calculates the camera matrix using the extrinsic and intrinsic parameters. The extrinsic parameters represent the location of the camera in the 3-D scene. The intrinsic parameters represent the optical center and focal length of the camera.

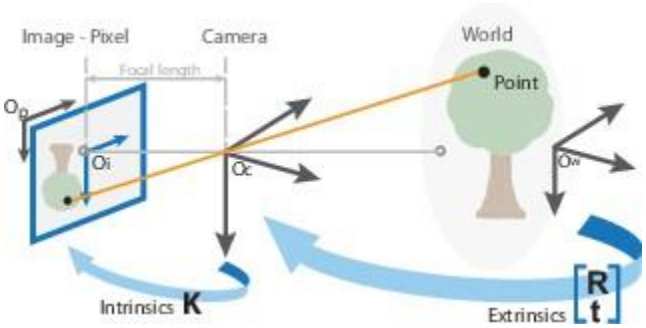
$$w\,[x\ y\ 1] = [X\ Y\ Z\ 1]\,P$$

Scale factor — Image points — World points

$$P = \begin{bmatrix} R \\ t \end{bmatrix} K$$

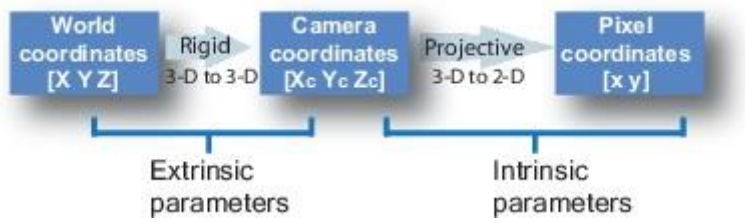
Camera matrix — Extrinsics: Rotation and translation — Intrinsic matrix

The world points are transformed to camera coordinates using the extrinsics parameters. The camera coordinates are mapped into the image plane using the intrinsics parameters.



Camera Calibration Parameters

The calibration algorithm calculates the camera matrix using the extrinsic and intrinsic parameters. The extrinsic parameters represent a rigid transformation from 3-D world coordinate system to the 3-D camera's coordinate system. The intrinsic parameters representa projective transformation from the 3-D camera's coordinates into the 2-D image coordinates.



Extrinsic Parameters

The extrinsic parameters consist of a rotation, *R*, and a translation, *t*. The origin of the camera's coordinate system is at its optical center and its *x*- and *y*-axis define the image plane.

5.2 Geometry of image formation

The two parts of the image formation process

The geometry of image formation which determines where in the image plane theprojection of a point in the scene will be located.

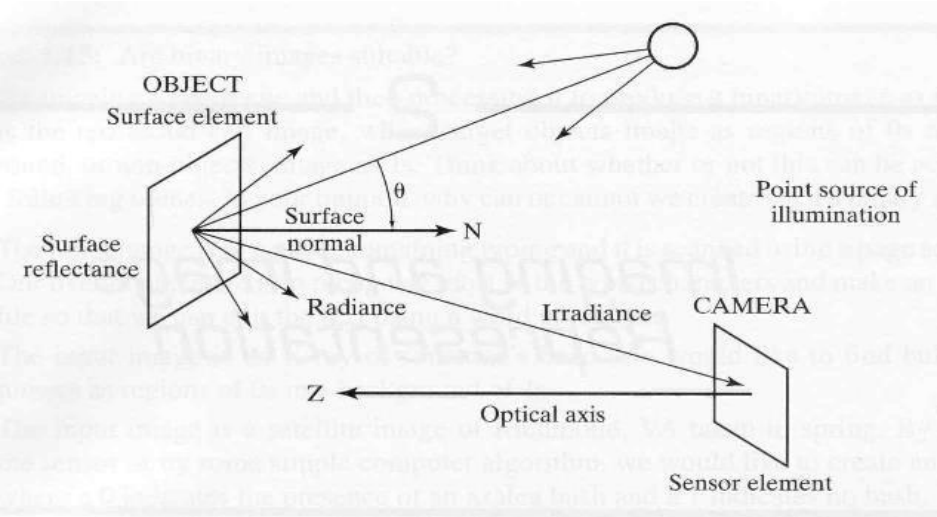
The physics of light which determines the brightness of a point in the imageplane asa function of illumination and surface properties.

A simple model

The scene is illuminated by a single source.

The scene reflects radiation towards the camera.

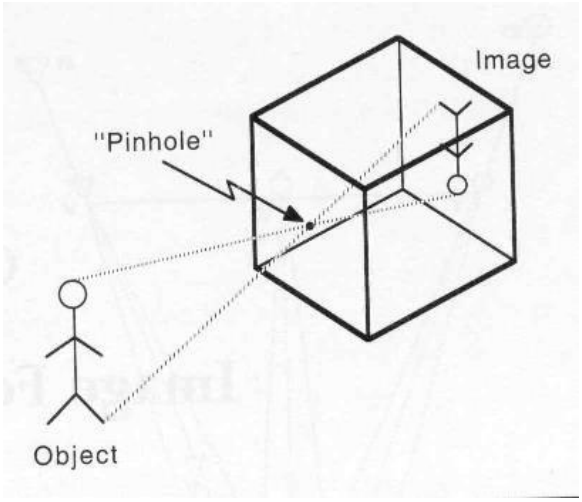
The camera senses it via chemicals on film.



• Camera Geometry

- The simplest device to form an image of a 3D scene on a 2D surface is the "pinhole"camera.

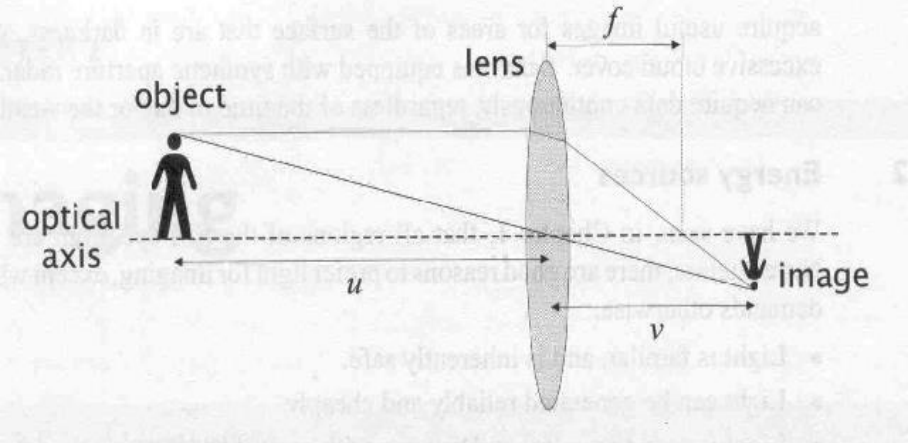
- Rays of light pass through a "pinhole" and form an inverted image of the object onthe image plane.



• Camera Optics

- In practice, the aperture must be larger to admit more light.

- Lens are placed in the aperture to focus the bundle of rays from each scene pointonto the corresponding point in the image plane.



5.2 **Euclidean/similarity/affine**

Euclidean geometry is a mathematical system attributed to Alexandrian Greek mathematician Euclid, which he described in his textbook on geometry: the *Elements*. Euclid's method consists in assuming a small set of intuitively appealing axioms, and deducing many other propositions (theorems) from these. Although many of Euclid's resultshad been stated by earlier mathematicians,[1] Euclid was the first to show how these propositions could fit into a comprehensive deductive and logical
system.[2] The *Elements* begins with **plane geometry**, still taught in secondary school (high school) as the first axiomatic system and the first examples of mathematical proofs. It goeson to the solid geometry of three dimensions. Much of the *Elements* states results of what are now called algebra and number theory, explained in geometrical language.[1]

For more than two thousand years, the adjective "Euclidean" was unnecessary because noother sort of geometry had been conceived. Euclid's axioms seemed so intuitively obvious (with the possible exception of the parallel postulate) that any theorem proved from them was deemed true in an absolute, often metaphysical, sense. Today, however, many
other self-consistent non-Euclidean geometries are known, the first ones having been discovered in the early 19th century. An implication of Albert Einstein's theory of generalrelativity is that physical space itself is not Euclidean, and Euclidean space is a good approximation for it only over short distances (relative to the strength of the gravitationalfield).

Euclidean geometry is an example of synthetic geometry, in that it proceeds logically from axioms describing basic properties of geometric objects such as points and lines, to propositions about those objects, all without the use of coordinates to specify those objects.This is in contrast to analytic geometry, which uses coordinates to translate geometric propositions into algebraic formulas.

Affine transformation

In Euclidean geometry, an **affine transformation**, or an **affinity** (from the Latin, *affinis*, "connected with"), is a geometric transformation that preserves lines and parallelism (but not necessarily distances and angles).

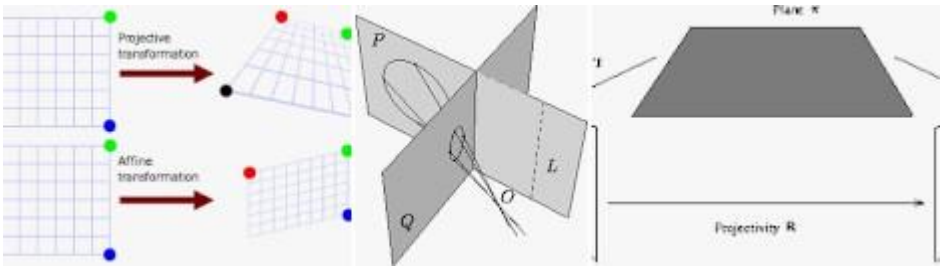
More generally, an *affine transformation* is an automorphism of an affine space (Euclidean spaces are specific affine spaces), that is, a function which maps an affine space onto itself while preserving both the dimension of any affine subspaces (meaning that it sends points topoints, lines to lines, planes to planes, and so on) and the ratios of the lengths of parallel linesegments. Consequently, sets of parallel affine subspaces remain parallel after an affine transformation. An affine transformation does not necessarily preserve angles between lines or distances between points, though it does preserve ratios of distances between points lying on a straight line.

If X is the point set of an affine space, then every affine transformation on X can be represented as the composition of a linear transformation on X and a translation of X. Unlikea purely linear transformation, an affine transformation need not preserve the origin of the affine space. Thus, every linear transformation is affine, but not every affine transformation islinear.

Examples of affine transformations include
translation, scaling, homothety, similarity, reflection, rotation, shear mapping, and compositions of them in any combination and sequence.

Viewing an affine space as the complement of a hyperplane at infinity of a projective space,the affine transformations are the projective transformations of that projective space that leave the hyperplane at infinity invariant, restricted to the complement of that hyperplane.

5.2 Projective Transformation



A transformation that maps lines to lines (but does not necessarily preserve parallelism) isa projective transformation. Any plane projective transformation can be expressed by an invertible 3×3 matrix in homogeneous coordinates; conversely, any invertible 3×3 matrix defines a projective transformation of the plane.

5.3 Visison application in Robotics

Inspection

Inspection tasks can be carried out by integrating machine vision and robots. Machine visionis used to make checks for visual factors such as surface finish, dimensions, potential errorsin labeling, and the presence of holes and other elements.

Machine vision can carry out these tasks faster and with fewer errors than humans can,meaning that production becomes faster and more profitable as a result.

Identification

Machine vision can be incorporated in robotics, giving them the skills of object detection toallow for identification and the classification of numerous objects simultaneously. Machine vision looks for the "variable" part of the object, the bit that is different and sets it apart, in order to successfully identify it.

This can help robots in warehouses to find the right item quickly, this speeds up production,and can also make retail processes more efficient.

Navigation

Machine vision is used to enhance and correct data coming in through other sources in orderto move robots safely and autonomously in a dynamic environment. Other measures of incoming data, such as accelerometers and encoders, can relay small errors that add up over time.

With the addition of vision, the robots can move more accuracy. This capability has implications for many industries, manufacture, mining and even autonomous vehicles.

Quality Control

Through the capabilities of inspection and identification, machine vision can be reliably usedin quality control applications.

The machine vision techniques of inspection and identification are combined in order to assess whether products meet various quality control checks. This has the impact of makingproduction more efficient and cost-effective.

Assembling

Research has shown that machine vision can be integrated with robotic systems to createpick and place capabilities.

Together, the system can accurately pick the correct assembly parts from the storage stationand put them in the right assembly spaces and on the appropriate parts where they need to be fixed. This gives the possibility of automated assembly lines with the use of robots with machine vision.

5.4 Kinds of sensors used in robotics

Types of Robot Sensors

- Light **sensors**. A Light **sensor** is **used** to detect light and create a voltage difference. ...
- Sound **Sensor**. ...
- Temperature **Sensor**. ...
- Contact **Sensor**. ...
- Proximity **Sensor**. ...
- Distance **Sensor**. ...
- Pressure **Sensors**. ...
- Tilt **Sensors**.

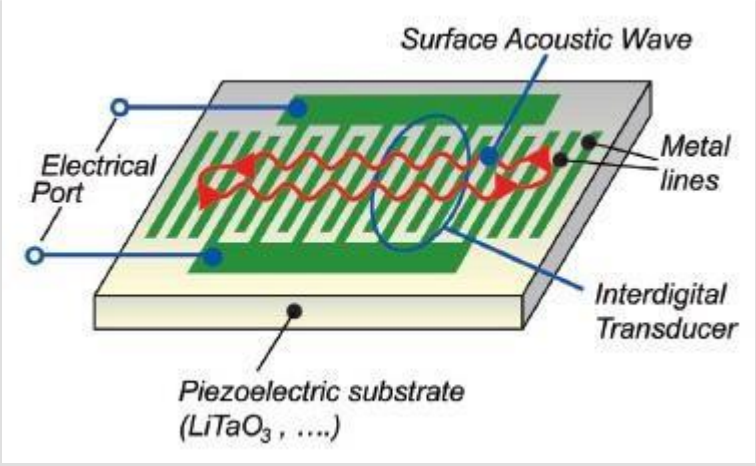
Acoustic sensor

Acoustic wave sensors

A sensor is used to measure (sense) an environment and converts this information into a digital or analogue data signal that can be interpreted by a computer or observer. This article will cover the area of acoustic wave sensors. An acoustic wave sensor is an electronic device that can measure sound levels. They are called acoustic wave sensors because their detection mechanism is a mechanical (or acoustic) wave. When an acoustic wave (input) travels through a certain material or along the surface of a material, it is influenced by the different material properties and obstacles it travels through. Any changes to the characteristics of this travelling path affect the velocity and/or amplitude of the wave. These

characteristics are translated into a digital signal (output) using transducers. These changes can be monitored by measuring the frequency or phase characteristics of the sensor. Then these changes can be translated to the corresponding physical differences being measured.

Piezoelectric



The principle of an acoustic wave sensor

Practically all acoustic wave devices and sensors use a piezoelectric material to generate the acoustic wave. Piezoelectricity essentially means electricity resulting from pressure. It refers to the production of electrical charges as a result of mechanical stress. The sensors normally use two inter digital transducer (IDT) that can convert the incoming signal into a mechanical wave signal trough a piezoelectric substrate. The transducers are interlocked electrodes in a comb-structure. What they do is turning an electrical signal into a mechanical wave and convert it into a electrical signal again. The distance of these electrodes determines the frequency of the wave and can be used to register torque or strain (the distance changes when deformed).The performance of these sensors can be changed by varying the length, width and position of the IDT. Piezoelectric acoustic wave sensors are relatively cheap, rugged, very sensitive, reliable, and can be used passively (without a power source) and wirelessly.

Applications

All acoustic wave sensors are sensitive to changes from many different physical parameters. These sensors are often used in the telecommunications industry. All acoustic wave devices manufactured for the telecommunications industry must be hermetically sealed to prevent any disturbances. The reason for this is that these disturbances will be sensed by the device and cause an unwanted change in output. These sensors can be used as pressure, mass, thickness, torque, shock,

acceleration, angular rate, viscosity, displacement, flow and force detectors under an applied stress that changes the dynamics of the object it travels through. Sensors can also detect mechanical failures or hick-ups such as grinding of components. Other applications are the monitoring of closed systems such as water pipe lines irregularities, turbulence, noise in hydraulic and pneumatic systems and pressure fluctuations. The sensors also have an acoustic electric sensitivity, allowing the detection of pH levels, ionic contaminants, and electric fields.

5.4 Optics sensors

A optic sensor is a sensor that uses optical fiber either as the sensing element ("intrinsic sensors"), or as a means of relaying signals from a remote sensor to the electronics that process the signals ("extrinsic sensors"). Fibers have many uses in remote sensing.
Depending on the application, fiber may be used because of its small size, or becauseno electrical power is needed at the remote location, or because many sensors can be multiplexed along the length of a fiber by using light wavelength shift for each sensor, orby sensing the time delay as light passes along the fiber through each sensor. Time delay can be determined using a device such as an optical time-domain reflectometer and wavelength shift can be calculated using an instrument implementing optical frequency domain reflectometry.

optic sensors are also immune to electromagnetic interference, and do not conduct electricity so they can be used in places where there is high voltage electricity or flammablematerial such as jet fuel.optic sensors can be designed to withstand high temperatures as well.

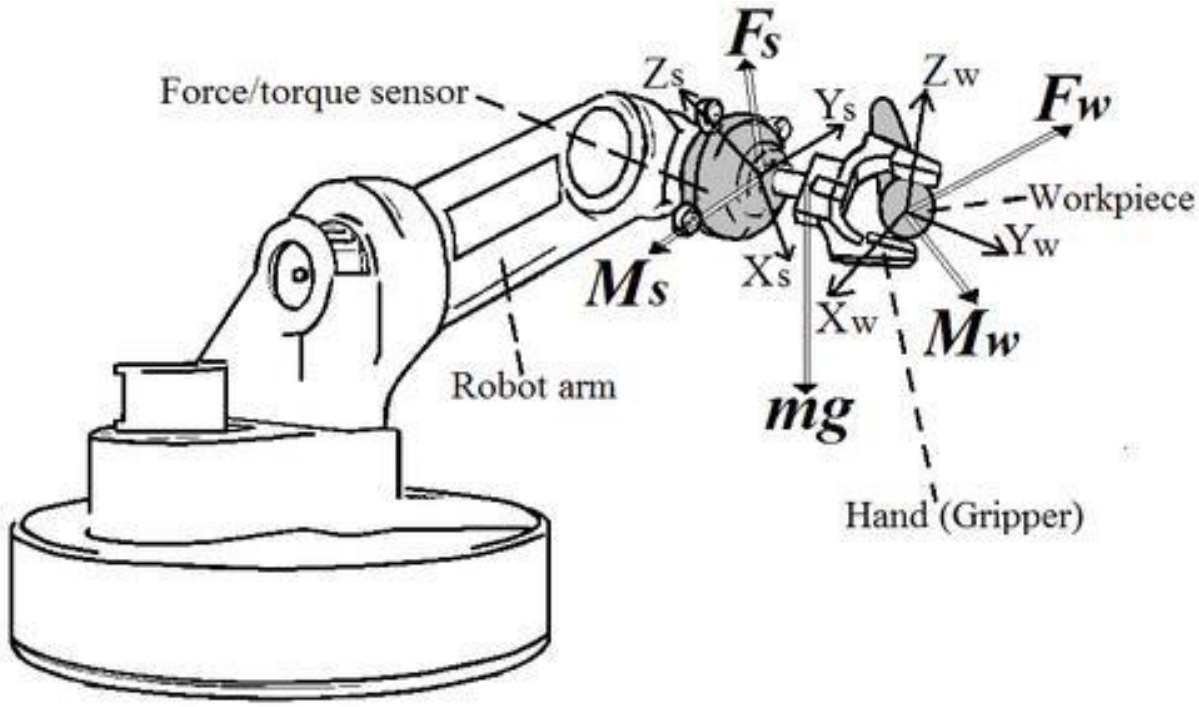
5.4 Pneumatic Sensor

Sensors are used to provide position feedback to control systems in automated machinery and equipment. Pneumatic cylinders use sensors to detect the linear position of the piston for applications where position feedback is crucial. The most common type of sensor used for pneumatic cylinders are magnetic proximity sensors, which detect the magnetic field of a magnet integrated in the cylinder piston. The sensor is mounted onto the pneumatic cylinder's body and will indicate "ON" or "OFF" based on proximity to the magnet. Depending on the application, various different magnetic proximity sensor technologies can be used to maximize performance, space, and reliability. Figure 1 shows examples of different pneumaticcylinder proximity sensors.

5.4 Force/Torque sensor

A force torque (FT) sensor is an electronic device that is designed to monitor, detect,record and regulate linear and rotational forces exerted upon it. In other words, the FT sensor in a robotic or mechanical system can be compared to the micro- receptors in skin that equip animals with the sense of "touch."

As a contact sensor, it is specifically designed to interact with physical objects in itsenvironment. In order to mitigate interference from sound waves and debris, this sensor is designed to operate in a variety of climates and external conditions. Depending on the model and intended function, a force torque sensor isable to send digital or analog signals, and measure static or dynamic forces.

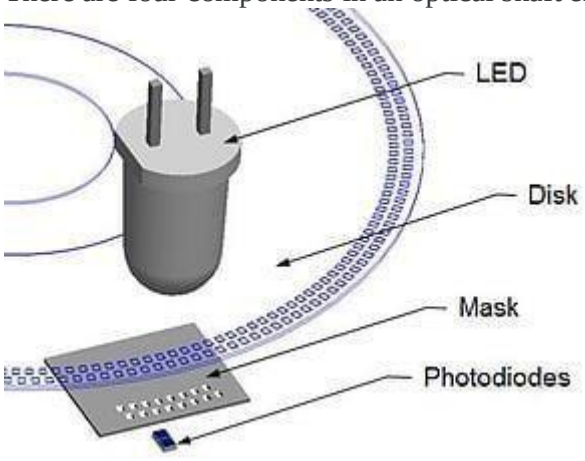


Measurement of external forces and moments by a six-axis force/torque sensor. The most popular type of force torque sensor is the six-axis sensor. This particular FT sensor is capable of measuring forces in every direction. A six-axis FT sensor generally utilizes strain gauge technology; when pressure is applied, the resistancewithin the gauge increases or decreases proportionally to the force it receives. Thisis how the sensor measures the movement of its external frames in relation to one another. Six-axis sensors can be found in robotic arms at the "joint."

5.5 Optical encoders

An optical encoder is a type of rotary encoder that uses a sensor to identify positionchange as light passes through a patterned encoder wheel or disk.

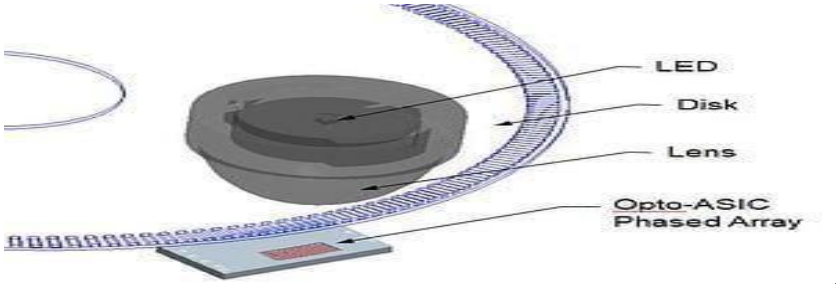
There are four components in an optical shaft encoder:



- A light source (an LED light)
- A sensor
- A moveable disk
- A fixed mask

The LED shines through one side of the optical shaft encoder. The encoder wheel or disk has a series of tracks on it, similar to the concentric grooves in an LP. The maskhas a corresponding track for every track on the disk of the optical encoder, and small perforations, called windows, are cut along the tracks in the mask. As the disk moves, different windows in the mask are covered or open, showing the movement and position of the optical shaft encoder. Each arc in the rotation indicates a differentposition and has a different pattern of open/closed windows. The sensor behind the mask identifies the optical encoders' current pattern.

Each sensor represents one single signal for the optical encoder. A track can containtwo sensors, which are offset to give two slightly different signals produced at the same time. These offset signals can be used by the optical encoder engine to determine more detailed motion information, like speed. A second track can be used to give an index pulse once per revolution, providing a method to orient the signals.



Phased-array optical encoders use multiple signal outputs to average together to create a single signal that is delivered by the engine. These multiple signals that are used by an optical shaft encoder are called the array. By using averages instead of asingle reading, phased array-optical encoders have much more stable signals so they can be used in less stable environments, such as mining or heavy manufacturing, where vibrations or shock could affect a traditional mask optical shaftencoder. They require less precision during installation than traditional mask optical encoders.

Short questions

Q1-Define contact and proximity sensor

A proximity sensor is a non-contact sensor that detects the presence of an object (often referredto as the "target") when the target enters the sensor's field .....Proximity sensors are used in phones, recycling plants, self-driving cars, anti-aircraft systems, and assembly lines

Q 2 Define Tactile sensor

Tactile sensors are data acquisition devices, or transducers, that are designed to sense adiversity of properties via direct physical contact (Nicholls and Lee, 1989). **Tactile sensor** designs are based around a range of different technologies some of which are directlyinspired by research on biological touch

Q 3 What is cameraq calibiration

Geometric **camera calibration**, also referred to as **camera** resectioning, estimates the parameters of a lens and image sensor of an image or video **camera**. You can use these

parameters to correct for lens distortion, measure the size of an object in world units, ordetermine the location of the **camera** in the scene.

Q 4 Define Euclidean

Wiktionary. **Euclidean**(Adjective) Adhering to the principles of traditional geometry, in which parallel lines are equidistant. Etymology: Named after **Euclid**, who established the principles ofplane geometry.

Q 5 What is Affine

An **affine** function is a function composed of a linear function + a constant and its graphis a straight line. The general equation for an **affine** function in 1D is: $y = Ax + c$.
An **affine** function demonstrates an **affine** transformation which is equivalent to a lineartransformation followed by a translation.

Q 6 State vision application in robotics

Machine Vision Applications

- Final inspection of sub-assemblies.
- Engine part inspection.
- Label inspection on products.
- Checking medical devices for defects.
- Final inspection cells.
- Robot guidance.
- Verifying datamatrix codes.
- Checking orientation of components. Q 7 State

different sensor used in robotics

List of Sensors

- Vision and Imaging **Sensors**.
- Temperature **Sensors**.
- Radiation **Sensors**.
- Proximity **Sensors**.
- Pressure **Sensors**.
- Position **Sensors**.
- Photoelectric **Sensors**.
- Particle **Sensors**.

Q 8 Define force/torque sensor

A **force torque** (FT) **sensor** is an electronic device that is designed to monitor, detect, record and regulate linear and rotational **forces** exerted upon it. In other words, the FT **sensor** in a robotic or mechanical system can be compared to the micro-receptors in skin that equip animalswith the sense of "touch

Long Questions

Q 1-State and explain different types of contact and proximity sensor2-State and explain about camera calibration

3- State and explain acoustic sensor in robotics.4-Explain about optical encoder.

5-Expalin about projective transformation of sensor of robotics.

CHAPTRER-6

ROBOT CONTROLAND ROBOT ACTUATION SYSTEM

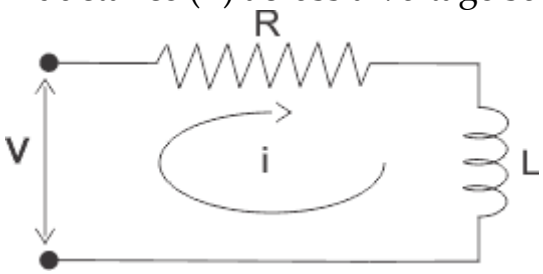
6.1-BASICS OF CONTROL-TRANSFER FUCTION

The transfer function of a control system is defined as the ratio of the Laplace transform of the output variable to Laplace transform of the input variable assuming all initial conditions to be zero.

$$G(s) = \frac{C(s)}{R(s)}$$

Concept of Transfer Function

The transfer function is generally expressed in Laplace Transform and it is nothing but the relation between input and output of a system. Let us consider a system consists of a series connected resistance (R) and inductance (L) across a voltage source (V).



In this circuit, the current 'i' is the response due to applied voltage (V) as cause. Hence the voltage and current of the circuit can be considered as input and output of the system respectively.

From the circuit, we get,

$$V = Ri + L\frac{\mathrm{d}i}{\mathrm{d}t}$$

Now applying Laplace Transform, we get,

$$V(s) = RI(s) + L\left[sI(s) - i(0^+)\right]$$

$$\left[\because Initially\ inductor\ behaves\ as\ open,\ hence, i(0^+) = 0\right]$$

$$\Rightarrow V(s) = I(s)\left[R + Ls\right]$$

$$\Rightarrow \frac{I(s)}{V(s)} = \frac{1}{R + Ls} = \frac{1/L}{s + R/L}$$

The transfer function of the system, G(s) = I(s)/V(s), the ratio of output to input.

6.1-Control law-P,PD,PID

Proportional Controllers

All controllers have a specific use case to which they are best suited. We cannot just insert any type of controller at any system and expect a good result – there are certain conditions that must be fulfilled. For a **proportional controller,** there are two conditions and these are written below:

1. The deviation should not be large; i.e. there should not be a large deviation between the input and output.
2. The deviation should not be sudden.

Now we are in a condition to discuss proportional controllers, as the name suggests in a proportional controller the output (also called the actuating signal) is directly proportional to the error signal. Now let us analyze the proportional controller mathematically. As we know in proportional controller output is directly proportional to the error signal, writing this mathematically we have,

$$A(t) \propto e(t)$$

Removing the sign of proportionality we have,

$$A(t) = K_p \times e(t)$$

Where K_p is proportional constant also known as controller gain.
It is recommended that K_p should be kept greater than unity. If the value of K_p is greater than unity (>1), then it will amplify the error signal and thus the amplified error signal can be detected easily.

Advantages of Proportional Controller

Now let us discuss some advantages of the proportional controller.

1. The proportional controller helps in reducing the steady-state error, thus makes the system more stable.
2. The slow response of the overdamped system can be made faster with the help of these controllers.

Disadvantages of Proportional Controller

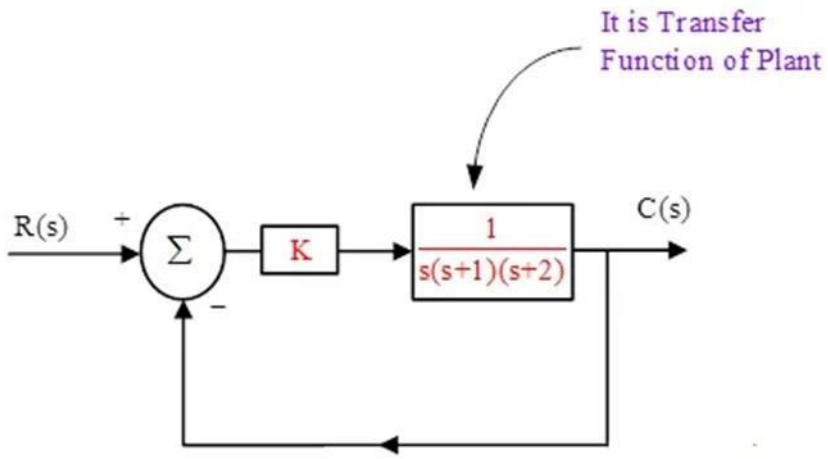
Now there are some serious disadvantages of these controllers and these are written as follows:

1. Due to the presence of these controllers, we get some offsets in the system.
2. Proportional controllers also increase the maximum overshoot of the system.

Now, we will explain the Proportional Controller (P-controller) with a unique example. With this example reader's knowledge about 'Stability' and 'Steady State Error' will also enhance.

Control System with Proportional Controller

'K' is called a proportional controller (also called error amplifier). Characteristics equation of this control system can be written as:



$$s^3+3s^2+2s+K=0$$

If the Routh-Hurwitz is applied in this characteristics equation, then the range of 'K' for the stability can be found as $0<K<6$. (It implies that for the values $K>6$ system will be unstable; for the value of $K=0$, the system will be marginally stable).

Proportional and Derivative Controller

As the name suggests it is a combination of proportional and a derivative controller the output (also called the actuating signal) is equals to the summation of proportional and derivative of the error signal. Now let us

analyze proportional and derivative controller mathematically. As we know in a proportional and derivative controller output is directly proportional to the summation of proportional of error and differentiation of the error signal, writing this mathematically we have,

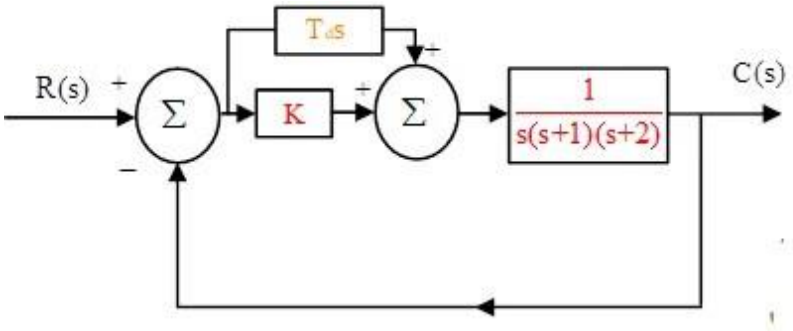
$$A(t) \propto \frac{de(t)}{dt} + A(t) \propto e(t)$$

Removing the sign of proportionality we have,

$$A(t) = K_d \frac{de(t)}{dt} + K_p e(t)$$

Where, K_d and K_p proportional constant and derivative constant respectively.
Advantages and disadvantages are combinations of advantages and disadvantages of proportional and derivative controllers.
Readers should note that adding 'zero' at the proper location in the open-loop transfer function improves stability, while the addition of pole in the open-loop transfer function may reduce the stability. The words "at proper location" in the above sentence are very important & it is called designing of the control system (i.e. both zero & pole should be added at proper points in the complex plane to get the desired result).

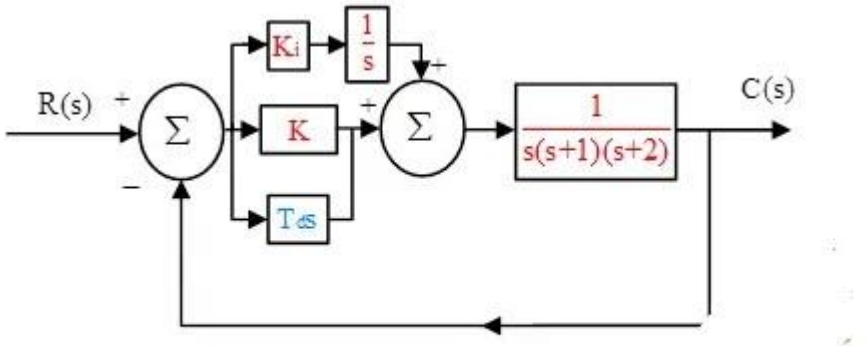
Inserting the PD controller is like the addition of zero in open-loop transfer function [G(s)H(s)]. Diagram of PD Controller is shown in Fig



Closed-loop control system with PD Controller

Proportional plus Integral plus Derivative Controller (PID Controller)

A PID controller is generally used in industrial control applications to regulate temperature, flow, pressure, speed, and other process variables.



Closed loop control system with PID Controller

The transfer function of the PID Controller can be found as:

$$Tds + K + \frac{Ki}{s} \text{ or } \frac{Tds^2+Ks+Ki}{s}$$

It can be observed that one pole at origin is fixed, remaining parameters T_d, K, and Ki decide the position of two zeros. In this case, we can keep two complex zeros or two real zeros as per the requirement, hence PID controller can provide better tuning. In the olden days, the PI controller was one of the best choice of control engineers, because designing (tuning of parameters) of the PID controller was a little difficult, but nowadays, due to the development of software designing of PID controllers have become an easy task.

Designing a PID Controller

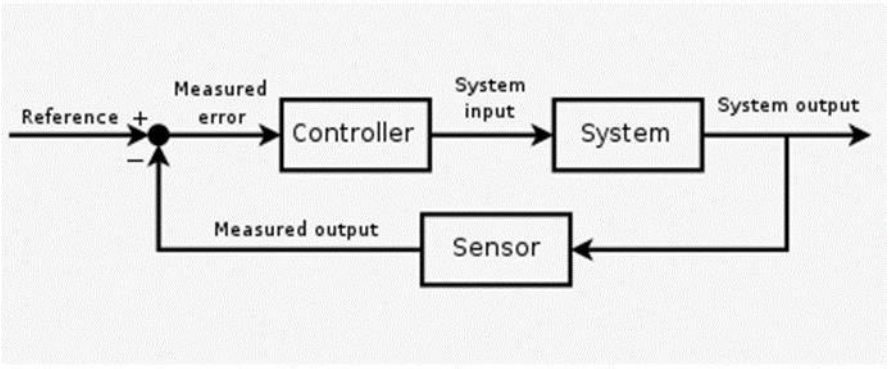
When you are designing a PID controller for a given system, general guidelines to obtain the desired response are as follows:

- Obtain the transient response of closed-loop transfer function and determine what needs to be improved.
- Insert the proportional controller, Design the value of 'K' through Routh-Hurwitz or suitable software.
- Add an integral part to reduce steady-state error.

- Add the derivative part to increase damping (damping should be between 0.6-0.9). The derivative part will reduce overshoots & transient time.
- Sisotool, available in MATLAB can also be used for proper tuning and to obtain a desired overall response.
- Please note, above steps of tuning of parameters (designing of a control system) are general guidelines. There are no fixed steps for designing controllers.

6.2-Non-Linear and Advance Controls

Nonlinear control theory is the area of control theory which deals with systems that are nonlinear, time-variant, or both. Control theory is an interdisciplinary branch of engineering and mathematics that is concerned with the behavior of dynamical systems with inputs, and how to modify the output by changes in the input using feedback, feedforward, or signal filtering. The system to be controlled is called the "plant". One way to make the output of a system follow a desired reference signal is to compare the output of the plant to the desired output, and provide feedback to the plant to modify the output to bring it closer to the desired output.



Control theory is divided into two branches. Linear control theory applies to systems made of devices which obey the superposition principle. They are governed by linear differential equations. A major subclass is systems which in addition have parameters which do not change with time, called *linear time invariant* (LTI) systems. These systems can be solved by powerful frequency domain mathematical techniques of great generality, such as the Laplace transform, Fourier transform, Z transform, Bode plot, root locus, and Nyquist stability criterion.

Properties of nonlinear systems

Some properties of nonlinear dynamic systems are

- They do not follow the principle of superposition (linearity and homogeneity).
- They may have multiple isolated equilibrium points.
- They may exhibit properties such as limit cycle, bifurcation, chaos.
- Finite escape time: Solutions of nonlinear systems may not exist for all times.

6.3-Actuators-Electric, hydraulic and Pneumatic

Actuators

An actuator is a device that uses a form of power to convert a control signal intomechanical motion. From electric door locks in automobiles, to ailerons on aircraft, actuators are all around us. Industrial plants use actuators to operate valves, dampers, fluid couplings, and other devices used in industrial process control. The industrial actuator can use air, hydraulic fluid, or electricity for motive power. These are referred to as pneumatic, electro-hydraulic, or electric actuators.

Robotics, considered to be one of the drivers of the so-called fourth industrialrevolution, is starting to find no limitations in regards to the variety of applications. Electrical actuators or gear motors are key when controlling positions and speeds demanded by several mechanisms and actuations for each robot. In this article, we go over some robotics innovations and the role actuators play in each of them.

Actuators, in conjunction with sensors and controllers, generate motion in each different part of a robot. Their use in a robotics project will depend on the stability we wish to obtain, the weight we need to move, and the repetition speed or precision with which the robot is to work, among other factors. Three types of actuators are mainly used in robotics, depending on theenergy they transform:

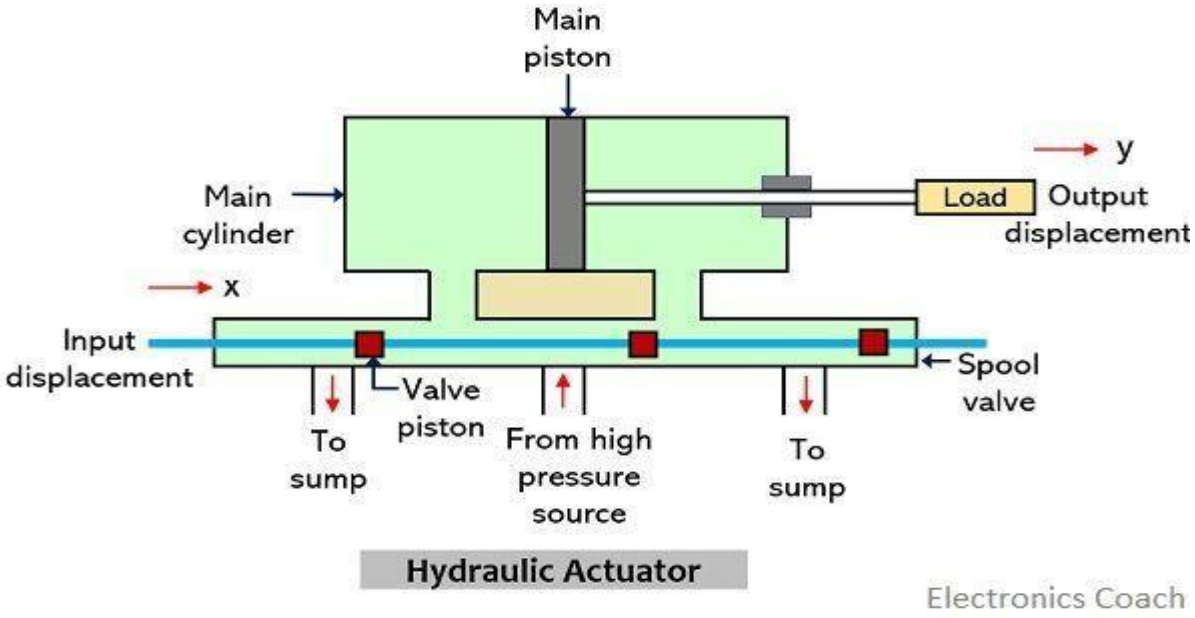
Hydraulic: they are used in large robots which require speed when executingrepetitive tasks, as well as great stability and mechanical strength for heavy loads. These actuators are classified as hydraulic cylinders, hydraulic motors and hydraulic valves.

Pneumatic: used in small-sized robots and actuator mechanisms that generallyrequire two states. Pneumatic actuators can be broken down into pneumatic cylinders and pneumatic motors.

Electrical: they are the most appropriate for robots that do not require great speed or power, but which do require accuracy and repetitiveness, as is the case of industrial robotics. Their use in this sector is particularly interesting dueto their simple installation, ease of control and reliability. Electrical actuators are classified as direct current motors or servomotors, alternate current motors, and step motors.

Working of Hydraulic Actuator

The figure below represents the schematic representation of the hydraulic actuator:



The major component of the unit is pilot valve also known as spool valve and main cylinder (or power cylinder).

It operates in a way that difference in pressure created at the two regions of the main cylinder leads to the occurrence of translational motion of the piston. Let us see the functioning of the hydraulic linear actuator.

As we have already mentioned that the main cylinder has two regions. These two regions are obtained by dividing the main cylinder with a main piston. Thus, there are two chambers of the main cylinder.

The rate with which the fluid flows inside the cylinder is controlled by the spool valve. The spool valve has 4 ports and each port is connected to a different part of the system.

Two separate ports are connected to the fluid supply and drain region respectively. While the other two ports are connected separately to the two chambers of the main cylinder.

Initially, the spool is present at the neutral position say x = 0 and at this position, there will be no flow of fluid inside the main cylinder. The assembly of the hydraulic actuator is such that the load will move according to the fluid flow.

Thus, when input displacement, x is 0 then the output displacement y will also be 0.

As soon as a certain input displacement is provided, then the spool moves towards the right. The movement of spool towards the right causes the fluid from the high-pressure source to move towards the left chamber of the main cylinder.

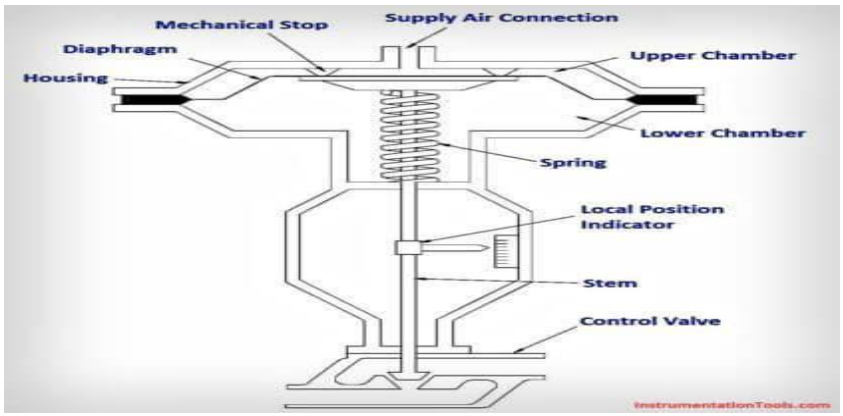
Thus, the pressure on the left chamber of the cylinder rises comparatively more than that present on the right chamber. This results in the generation of accelerating force that causes movement of the load.

In this way, the direction in which fluid flows corresponds to the direction in which the load moves. This acts as power amplification as discussed in operating principle because the force supplied to displace the valve is comparatively very small than the force generated that actually displaces the load.

Pneumatic Actuators :

A simplified diagram of a pneumatic actuator is shown in Figure 1. It operatesby a combination of force created by air and spring force. The actuator positions a control valve by transmitting its motion through the stem.

A rubber diaphragm separates the actuator housing into two air chambers. Theupper chamber receives supply air through an opening in the top of the housing.



Pneumatic Actuator: Air-to-Close/Spring-to-Open

The bottom chamber contains a spring that forces the diaphragm against mechanical stops in the upper chamber. Finally, a local indicator is connectedto the stem to indicate the position of the valve.

The position of the valve is controlled by varying supply air pressure in the upper chamber. This results in a varying force on the top of the diaphragm.

Initially, with no supply air, the spring forces the diaphragm upward against themechanical stops and holds the valve fully open.

As supply air pressure is increased from zero, its force on top of the diaphragmbegins to overcome the opposing force of the spring. This causes the diaphragm to move downward and the control valve to close. With increasing supply air pressure, the diaphragm will continue to move downward and compress the spring until the control valve is fully closed.

6.3-Transmission:Gears,Timing Belt and Bearing

Functions of a Gear Drive:

A gear drive has three main functions: to increase torque from the driving equipment (motor) to the driven equipment, to reduce the speed generated bythe motor, and/or to change the direction of the rotating shafts. The connection of this equipment to the gear box can be accomplished by the use of couplings, belts, chains, or through hollow shaft connections.

Speed and torque are inversely and proportionately related when power is held constant. Therefore, as speed decreases, torque increases at the sameratio.

The heart of a gear drive is obviously the gears within it. Gears operate in pairs, engaging one another to transmit power.

Spur Gear

Spur gears transmit power through shafts that are parallel. The teeth of the spur gears are parallel to the shaft axis. This causes the gears to produce radialreaction loads on the shaft, but not axial loads. Spur gears tend to be noisier than helical gears because they operate with a single line of contact between

teeth. While the teeth are rolling through mesh, they roll off of contact with one tooth and accelerate to contact with the next tooth. This is different thanhelical gears, which have more than one tooth in contact and transmit torque *more smoothly***.**

Helical Gear

Helical gears have teeth that are oriented at an angle to the shaft, unlike spur gears which are parallel. This causes more than one tooth to be in contact during operation and helical gears are capable of carrying more load than spurgears. Due to the load sharing between teeth, this arrangement also allows helical gears to operate smoother and quieter than spur gears. Helical gears produce a thrust load during operation which needs to be considered when they are used. Most enclosed gear drives use helical gears.

Double Helical Gear

Double helical gears are a variation of helical gears in which two helical facesare placed next to each other with a gap separating them. Each face has

identical, but opposite, helix angles. Employing a double helical set of gears eliminates thrust loads and offers the possibility of even greater tooth overlapand smoother operation. Like the helical gear, double helical gears are commonly used in enclosed gear drives.

Herringbone Gear

Herringbone gears are very similar to the double helical gear, but they do nothave a gap separating the two helical faces. Herringbone gears are typically smaller than the comparable double helical, and are ideally suited for high shock and vibration applications. Herringbone gearing is not used very often due to their manufacturing difficulties and high cost.

Bevel Gear

Bevel gears are most commonly used to transmit power between shafts that intersect at a 90 degree angle. They are used in applications where a right angle gear drive is required. Bevel gears are generally more costly and are notable to transmit as much torque, per size, as a parallel shaft arrangement.

Worm Gear

Worm gears transmit power through right angles on non-intersecting shafts. Worm gears produce thrust load and are good for high shock load applicationsbut offer very low efficiency in comparison to the other gears. Due to this low efficiency, they are often used in lower horsepower applications.

Hypoid Gear

Hypoid gears look very much like a spiral bevel gear but they operate on shaftswhich do not intersect, which is the case with a spiral bevel gear. In the hypoidarrangement because the pinion is set on a different plane than the gear, the shafts are supported by the bearings on either end of the shaft.

Bearing--

The bearing in its current form was developed towards the end of the 19thcentury. It was initially made by hand.

Nowadays, bearings are one of the most commonly used machine parts because their rolling motion make almost all movements easier and they helpreduce friction.

Bearings have two key functions:

They transfer motion, i.e. they support and guide components which turnrelative to one another

They transmit forces

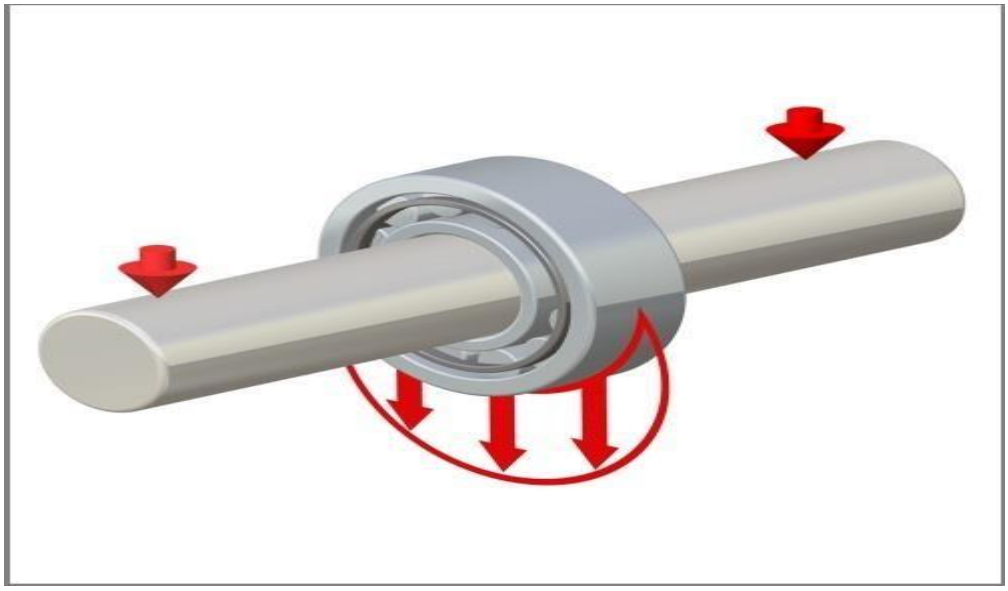
Rolling bearings and sleeve bearings

In a sleeve or plain bearing, the axle and the bearing move in opposite directions on a sliding surface. By contrast, the two components of a rolling bearing that move towards one another – the inner and outer rings – are separated by rolling elements. This design generates significantly less frictionthan a sleeve bearing.

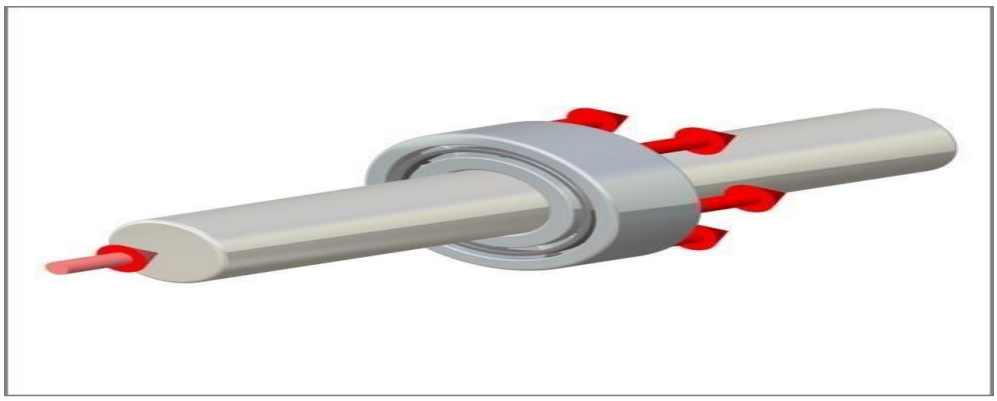
Radial bearings and axial bearings

Bearings can transmit loads in a radial direction or an axial direction (thrust)and in many cases there is a combination of both radial and axial loads to transmit.

Both designs are available as ball bearings or roller bearings. The choice ofbearing design depends upon the application in question.



Radial Bearing



Axial Bearing

Components

Bearings usually consist of the following components:

Two rings or discs with raceways

Rolling elements in the form of rollers or balls

A cage which keeps the rolling elements apart and guides them

Inner Ring / Outer Ring

The inner and outer ring are usually made from a special high-purity, chromealloy steel. This material has the necessary hardness and purity – both important factors for a high load rating and a long service life.

The raceways are hardened, ground and honed.

Special materials such as ceramic and plastics are also used. Although plastics cannot withstand extremely high temperatures, they are considerablylighter than steel. This makes them invaluable in sectors such as the automotive industry, where every gram matters.

Inner Ring

Outer Ring

Rolling Elements

Rolling elements are either balls, rollers, cones, spheres or needles. Theyare usually made from a special high-purity, chrome alloy steel. Special materials such as ceramic and plastics are also used.

The rolling elements roll on the specially formed raceways of the rings or discsand are kept apart and guided by the cage.

Cage

The cage is responsible for keeping the rolling elements apart and guiding them. The materials used include steel, brass and plastic. Solid metal cages can be produced using machining techniques, while pressed cages are made from sheet metal. Similarly, plastic cages can be machined from solid plastic orinjection moulded.

6.4-Parameters for selection of actuators

valves are used to control the flow of fluid in process systems. During routineoperations, the position of valves often needs to be opened or closed regularly. In large plants, manually adjusting these valves can be time- consuming and largely impractical.

In these cases, valve actuators are used in place of hand-operated wheels andlevers. Valve actuators are mechanical devices used to adjust valve positions. Instead of having operators physically locate and reposition valves, valve positions can be adjusted from a remotely located control room.

The actuator itself is a mechanism that produces a particular motion to controla valve.

Valve actuators can be used with control valves to throttle or regulate processflows, or they can be used in isolation valves to stop the flow of fluid at a givenlocation. These devices are commonly used in a variety of industrial plants, including:

Water/wastewater treatment facilitiesPower plants

Oil refineries

Food and beverageOil and

gas

Types of Valve Actuators

The working principle of actuator valves primarily depends on their drivingforce. The two most common types of valve actuators are pneumatic and electric.

Pneumatic Valve Actuators

Pneumatic valve actuators are the most common type of actuator used in process systems. These actuators use air (or other gas) pressure as a main power source. Air pressure is used to produce motion to control the position ofthe valve.

For double acting and spring return models, we offer two different styles of rack and pinion pneumatic actuators. Rack and pinion pneumatics are excellentfor a variety of industrial applications in fields such as chemical & plastic, food & beverage, and a whole breadth of general industrial sectors.

Our rack and pinion actuators are available in the PN Series or the SS Series with two different body material options (hard anodized extruded aluminumor stainless steel), plus feature ISO and NAMUR mounting to meet international standards.

Electric Valve Actuators (Motor Driven Actuators)

Electric actuators use electricity as their primary power source. Electricity is used to produce the motion which opens or closes the valve as required. Thesetypes of actuators fall into two general classifications: solenoid actuators or motor driven actuators.

In solenoid actuators, an electric current is applied to a wire coil (solenoid) which produces a magnetic field. This magnetic field pulls on a metal plungerwhich pushes the valve stem downward. When the solenoid is de-energized,the spring retracts the plunger upwards and reopens the valve.

In motor driven actuators, the valve stem is controlled by an electric motor. Aservo amplifier provides a DC signal to the motor which moves the valve stemin a linear motion to the desired position.

Actuator Selection Guide

Valve actuators are an essential element in the efficiency of process systemsand also play a significant role in process plant safety. It is therefore crucial that the right type of valve is selected for the given environment to ensure optimal valve operation.

The following factors must be taken into consideration when determining themost appropriate type of valve:

Power source

One of the first factors to consider is the most effective power source for theactuator. Power source availability, control access, valve size, frequency of operation, and required torque are all factors to consider when choosing between pneumatic or electric actuators. For pneumatic actuators, an air pressure generally between 40 and 120 psi would need to be provided. Most electric actuators would require available access to a 110 AC power supply, although different sized AC and DC motors are available.

Temperature range

The operating temperature range for pneumatic actuators is typically between -4° and 174° F (-20° to 80° C). With appropriately rated seals, bearings, and grease, this temperature range may increase to -40° to 250° F (-40° to 121° C).In low temperatures, the dew point in relation to the environmental temperature should be considered, as frozen condensate can affect the abilityof the air supply port to provide sufficient air pressure.

Electric motors are typically available in temperature ranges between -40° to 150°F (-40° to 65°C). Special care should be given to electric motors in outdoorapplications. The unit should be properly enclosed to prevent the

accumulation of condensation and rainwater which can negatively affect theactuator.

Hazardous locations

Pneumatic actuators are preferred in areas with hazardous, explosive, or flammable gases, vapor, or dust due to their lack of ignition source. Electric actuators, on the other hand, work with electricity, which is a classified as a source of ignition. Therefore, electric actuators need to be contained in NEMA,UL, or CSA-approved enclosures to prevent ignition of the external environment.

Short Question:-

Q1-Define Transfer function.

The transfer function of a control system is defined as the ratio of the Laplace transform of the output variable to Laplace transform of the input variable assuming all initial conditions to be zero

Q2-Define PD,PID

PI-PD controller is designed by approximating the PID feedback control to take advantages of excellent capabilities of a PID feedback control mechanism in predicting and controlling future error as well as eliminating the steady-state error.

PID stands for Proportional, Integral, Derivative. PID control provides a continuous variation of output within a control loop feedback mechanism to accurately control the process, removing oscillation and increasing process efficiency

Q3-define Non- linear control in robotics.

In nonlinear control field, a. common strategy is called model based control, which can be derived from the. mathematical model of the system. However, in case of robot manipulator, it is weakened. by inaccuracies present in the robot model, where the performance of the control algorithm.

Q4-Define Actuators and name different Actuators.

Similar to pneumatic actuators, they also create precise motion as the flow of electrical power is constant. The different types of electrical actuators include: ...
Electrohydraulic actuators: This type of actuator is also powered electrically but gives movement to a hydraulic accumulator

- Comb drive.
- Hydraulic piston.
- Electric motor.
- Relay.
- Thermal bimorph.

Q5-what is the function of pneumatic Actuators

A Pneumatic actuator mainly consists of a piston or a diaphragm which develops the motive power. It keeps the air in the upper portion of the cylinder, allowing air pressure to force the diaphragm or piston to move the valve stem or rotate the valve control element.

Q6 What are the 4 types of gears?

Different Types of Gears are

- Spur gears.
- Helical gears.
- Bevel gears.
- Worm gears.

Q7 What are the 3 types of bearings?

- Plain Bearings. The most basic type, plain bearings consist of a flat surface without any balls or rollers. ...
- Ball Bearings.
- Roller Bearings
- Fluid Bearings.
- Magnetic Bearings.

Long Question-

Q1-Explain detail about P and PID control.

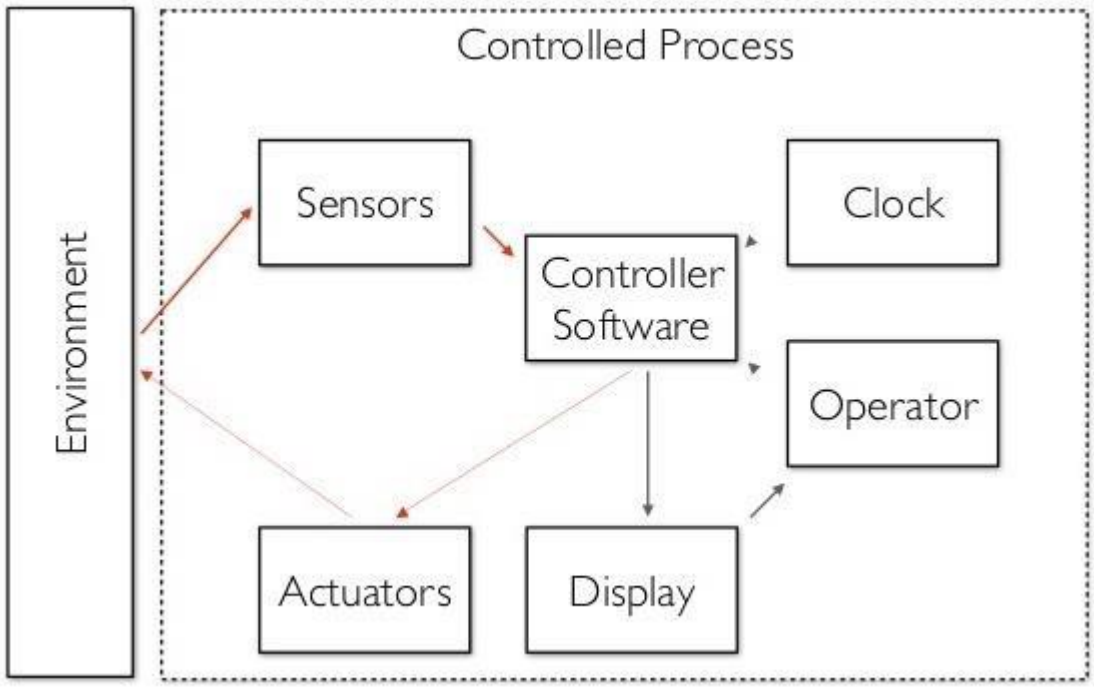
Q2-write the working principle of Hydraulic actuators. Q3-state the working principle of Pneumatic ActuatorsQ4-Explain different types of Gears Bearing

CHAPTER-7

CONTROL HARDWARE AND INTERFACE

7.1-Embedded system

An embedded system is a microprocessor- or microcontroller-based system of hardware and software designed to perform dedicated functions within a larger mechanical or electrical system.



An embedded system is a microprocessor-based computer hardware system with software that is designed to perform a dedicated function, either as an independent system or as a part of a large system. At the core is an integrated circuit designed to carry out computation for real-time operations.

Complexities range from a single microcontroller to a suite of processors with connected peripherals and networks; from no user interface to complex graphical user interfaces. The complexity of an embedded system varies significantly depending on the task for which it is designed.

Embedded system applications range from digital watches and microwaves to hybrid vehicles and avionics. As much as 98 percent of all microprocessors manufactured are used in embedded systems.

Working principle

Embedded systems are managed by microcontrollers or digital signal processors (DSP), application-specific integrated circuits (ASIC), field-programmable gate arrays (FPGA), GPU

technology, and gate arrays. These processing systems are integrated with components dedicated to handling electric and/or mechanical interfacing.

Embedded systems programming instructions, referred to as firmware, are stored in read-only memory or flash memory chips, running with limited computer hardware resources. Embedded systems connect with the outside world through peripherals, linking input and output devices.

Basic Structure of an Embedded System

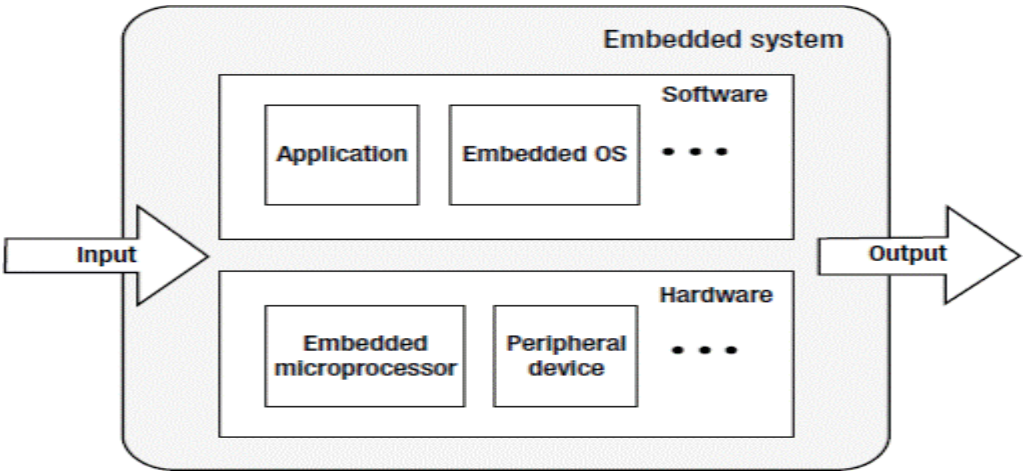
The basic structure of an embedded system includes the following components:

- Sensor: The sensor measures and converts the physical quantity to an electrical signal, which can then be read by an embedded systems engineer or any electronic instrument. A sensor stores the measured quantity to the memory.
- A-D Converter: An analog-to-digital converter converts the analog signal sent by the sensor into a digital signal.
  Processor & ASICs: Processors assess the data to measure the output and store it to the memory.
- D-A Converter: A digital-to-analog converter changes the digital data fed by the processor to analog data
- Actuator: An actuator compares the output given by the D-A Converter to the actual output stored and stores the approved output

Architecture of Embedded System

Figure shows a configuration diagram of a typical embedded system consisting of two main parts: embedded hardware and embedded software.The embedded hardware

Primarily includes the processor, memory, bus, peripheral devices, I/O ports, and various controllers. The embedded software usually contains theembedded operating system and various applications.

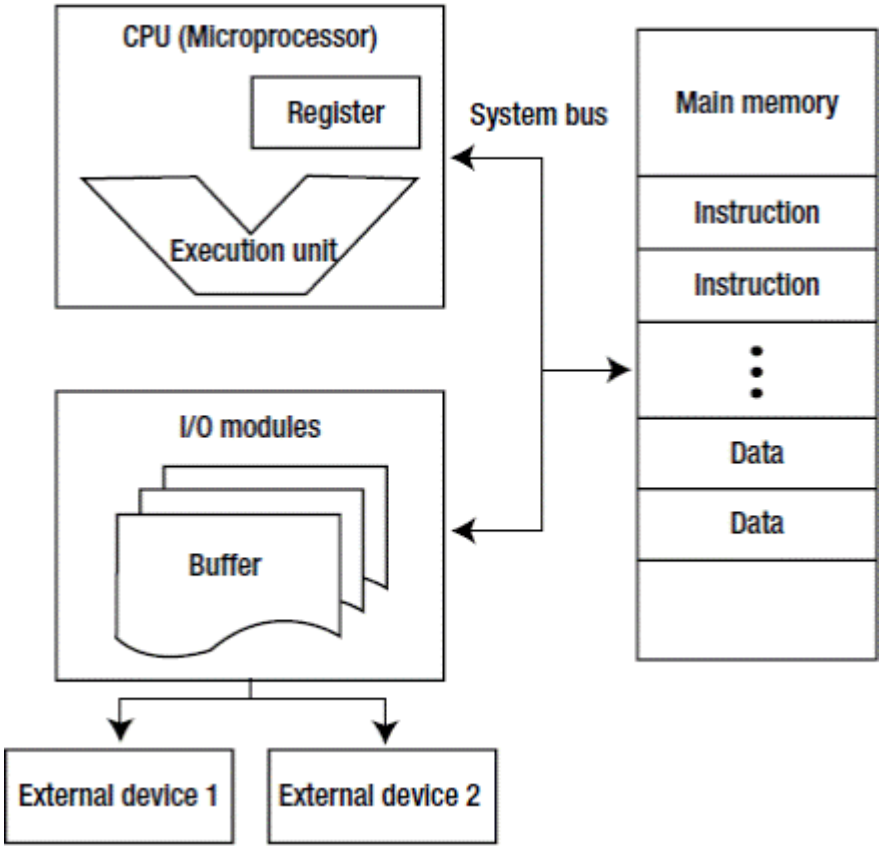


Basic architecture of an embedded system

Input and output are characteristics of any open system, and the embeddedsystem is no exception. In the embedded system, the hardware and software often collaborate to deal with various input signals from the outside and output the processing results through some form. The input signal may be an ergonomic device (such as a keyboard, mouse, or touch screen) or the output of a sensor circuit in another embedded system. The output may be in the form of sound, light, electricity, or another analog signal, or a record or file for a database.

Typical Hardware Architecture

The basic computer system components—microprocessor, memory, and input and output modules—are interconnected by a system bus in order forall the parts to communicate and execute a program (see Figure 1-3).



Computer architecture

In embedded systems, the microprocessor's role and function are usually the same as those of the CPU in a general-purpose computer: control computer operation, execute instructions, and process data. In many cases,the microprocessor in an embedded system is also called the CPU. Memoryis used to store instructions and data. I/O modules are responsible for the

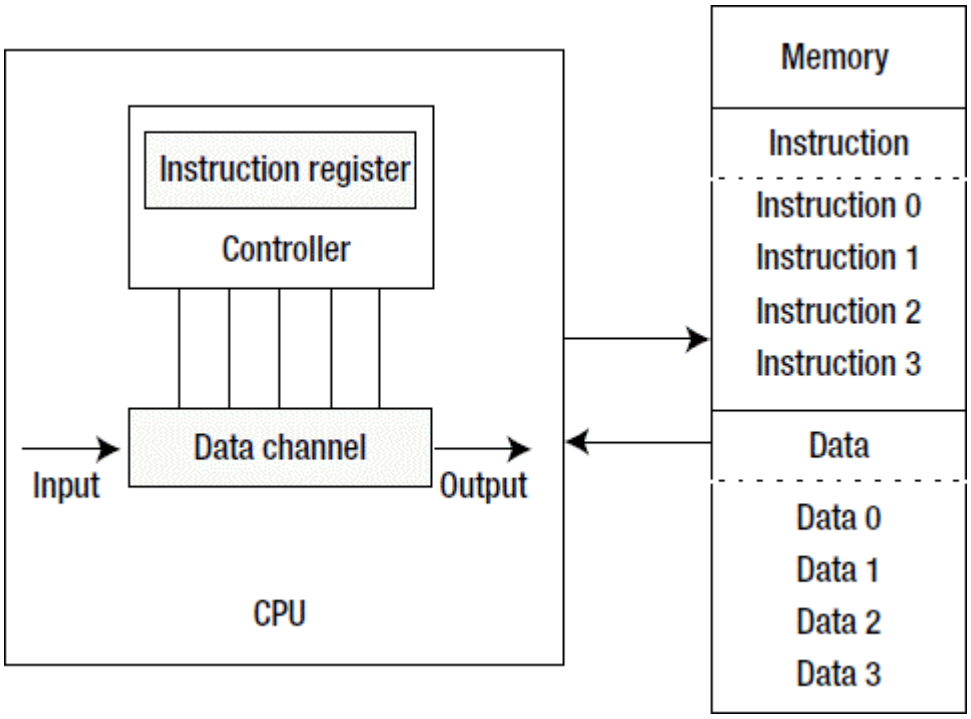
data exchange between the processor, memory, and external devices. External devices include secondary storage devices (such as flash and hard disk), communications equipment, and terminal equipment. The system busprovides data.

There are basically two types of architecture that apply to embedded systems:Von Neumann architecture and Harvard architecture.

Von Neumann Architecture

Von Neumann architecture (also known as Princeton architecture) was firstproposed by John von Neumann. The most important feature of this architecture is that the

software and data use the same memory: that is, “The program is data, andthe data is the program” (as shown in Figure 1-4).

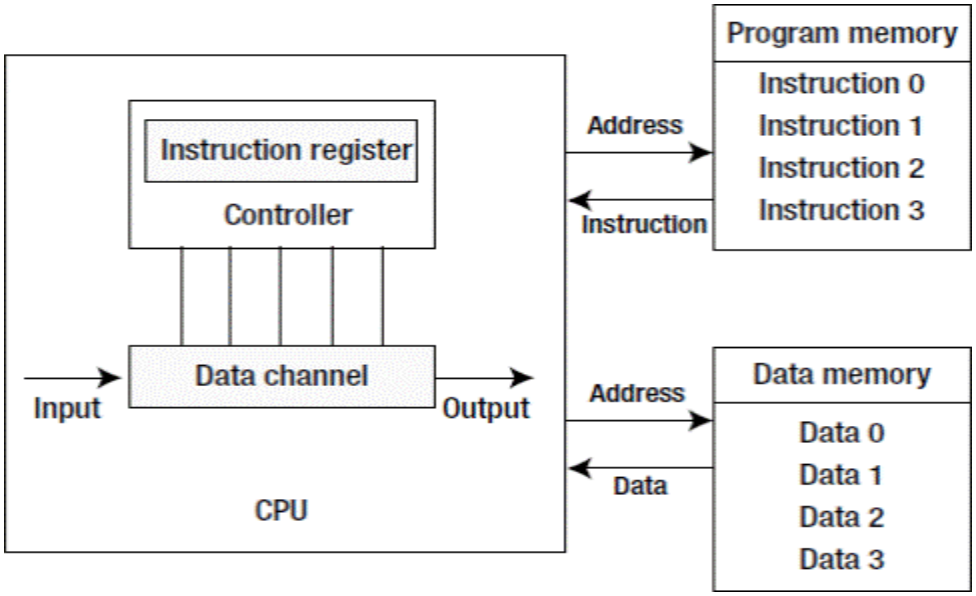


Von Neumann architecture

In the Von Neumann architecture, an instruction and data share the same bus.In this architecture, the transmission of information becomes the bottleneck ofcomputer performance and affects the speed of data processing; so, it is often called the *Von Neumann bottleneck*. In reality, cache and branch-prediction technology can effectively solve this issue.

Harvard Architecture

The Harvard architecture was first named after the Harvard Mark I computer. Compared with the Von Neumann architecture, a Harvard architecture processor has two outstanding features. First, instructions and data are storedin two separate memory modules; instructions and data do not coexist in the same module. Second, two independent buses are used as dedicated communication paths between the CPU and memory; there is no connection between the two buses. The Harvard architecture is shown in Figure 1-5.



B

Harvard architecture

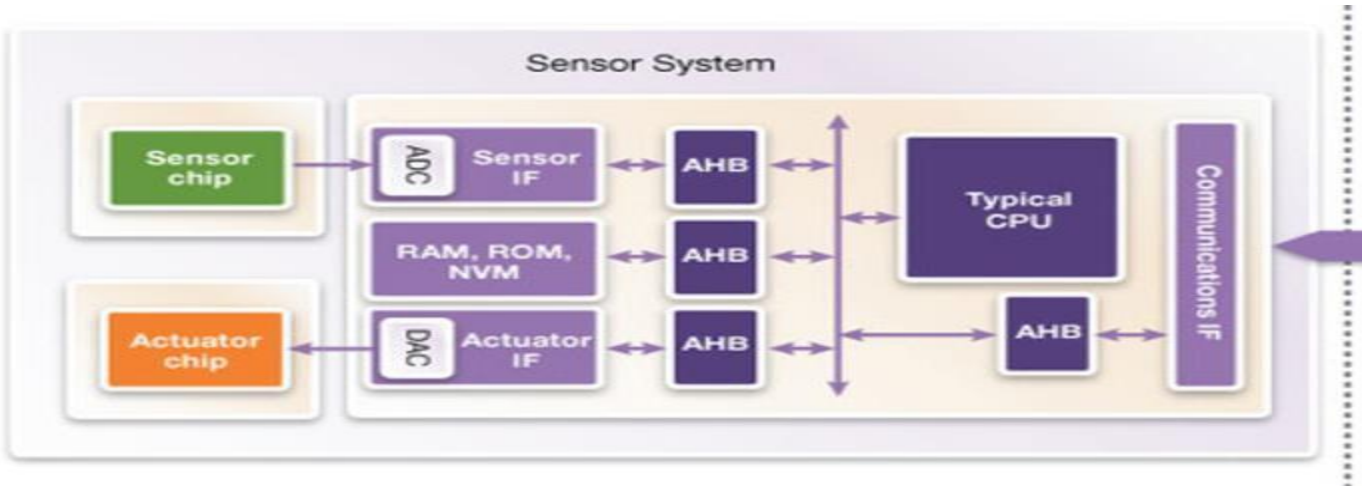
Because the Harvard architecture has separate program memory and data memory, it can provide greater data-memory bandwidth, making it the ideal choice for digital signal processing. Most systems designed for digital signal processing (DSP) adopt the Harvard architecture. The Von Neumann architecture features simple hardware design and flexible program and data storage and is usually the one chosen for general-purpose and most embeddedsystems.

To efficiently perform memory reads/writes, the processor is not directly connected to the main memory, but to the cache. Commonly, the only difference between the Harvard architecture and the Von Neumann architecture is single or dual L1 cache. In the Harvard architecture, the L1 cache is often divided into an instruction cache (I cache) and a data cache (Dcache), but the Von Neumann architecture has a single cache.

7.1-Integration with sensor.

An integrated sensor is the core technology of a sensor without the package. It allows for multiple sensor technologies to be combined or"integrated" into a single plug-and-play assembly

Within the sensor subsystem, the advantages of increased integration can differentiate a silicon vendor’s device. A typical “integrated” solution today involves incorporating the sensor interfaces into a microcontroller-like architecture. This architecture generally includes a typical CPU, connected through an on-chip bus to peripheral interfaces (ADC and DAC) as well as on-chip memories (ROM, RAM, Flash). The processor is connected to a standard bus (typically AMBA based) and all of the peripherals are connected to the bus. Transactions between the processor and peripherals take three to seven clocks or more due to buslatency and traffic on the bus. This is very inefficient in terms of performance and power consumption.



typical bus-based systems.

Closely coupled memories (I-CCM / D-CCM):

The ARC EM4 core provides an option to integrate tightly coupled memories for both instructions and data to reduce access times goingacross the AHB bus to ROM and RAM in a bus-based topology.

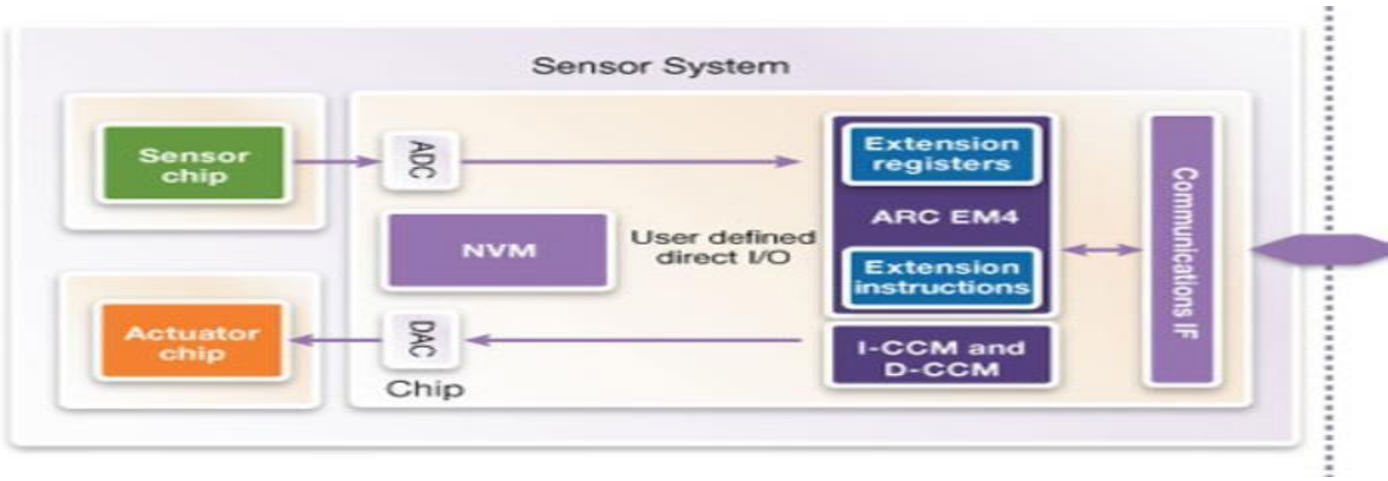
Tightly integrated I/O:

Starting from existing bus-based I/O peripheral, an ARC processor implementation can eliminate the interface to on-chip bus by replacing

load/store instructions to the I/O peripheral with register move instructions. The peripheral block registers are mapped using ARC'sauxiliary bus.
This implementation effectively pulls the I/O peripheral interface functionality into the CPU complex, eliminating the buses and bridges. Tightly coupling the I/O in this manner also enables single cycle accessto all peripheral functions – increasing performance while reducing power consumption and die area to use fewer gates.

Customizable hardware / custom instructions:

ARC processors support an implementation to add any combination of hardware extensions to the core: CPU extension registers, auxiliary extension registers, or memory mapped blocks. Designers can add 32- bit custom instructions as well. The hardware extensions are added withstandard code (either as Verilog or C) and are accessed and run using special (custom defined) instructions.



In addition to these ARC EM4 features, Synopsys' Sensor Subsystemprovides an additional level of hardware integration to accelerate standard DSP sensor application functions. These functions include complex math functions (square root), filtering (FIR, IIR, correlation), matrix/vector operations, and interpolation. Implementing the DSP functions in hardware reduces the memory footprint by replacing application software code with hardware accelerator instructions.

7.2-Programming for Robot Application

Here are the ten most popular programming languages in robotics at themoment. If your favorite language isn’t on the list, please tell everyone about it in the comments! Each language has different advantages for robotics. The way I have ordered them is roughly in order of importance from least to most valuable.

1. Pascal

Pascal was one of the first programming languages that I ever learned. However, that's not why I've included it here. It is the basis for several ofthe industrial robot languages (see number 8 below). As a result, it's stillquite a good starting point if you are going to be programming industrial robots.

Pascal is a basic language (literally based on the language BASIC) andwas designed to encourage good programming practices.

These days, Pascal is too outdated to be good for everyday use. However, it can be useful if you want to become familiar with other industrial robot languages without becoming tied to one particular robotbrand.

2. Scratch

A brand new entry to this list, Scratch is used by thousands of buddingroboticists around the world every year. This visual programming language is specially designed for new programmers — targeted at users aged 8 to 16 — and is often the language of choice in school technology classes and robotics clubs.

Scratch programming is achieved by dragging around blocks and connecting them together. "Under the hood" it is written in a combinationof Squeak (a dialect of Smalltalk), ActionScript and Javascript.

Although you're probably not going to write any industrial robot programsin Scratch, it's an incredibly good and popular language for complete beginners and is paving the way for many of our future robotics engineers.

3. Industrial Robot Languages

Almost every robot manufacturer has developed their own proprietary robot programming language, which has long been one of the problemsin industrial robotics. You can become familiar with several of them by

learning Pascal. However, you are still going to have to learn a newlanguage every time you start using a new robot brand.

ABB has its RAPID programming language. Kuka has KRL (Kuka RobotLanguage). Comau uses PDL2, Yaskawa uses INFORM and Kawasaki uses AS. Then, Fanuc robots use Karel, Stäubli robots use VAL3 and Universal Robots use URScript.

In recent years, more general-purpose programming options like ROS Industrial, manufacturer agnostic offline programming, and hand guidinghave started to provide more standardized options for programmers.

4. LISP and Prolog

Artificial Intelligence (AI) has really been gaining in popularity recently.This means that AI programming languages like LISP and Prolog are starting to make their way back into people's programming toolkits.

LISP is the world's second oldest programming language (FORTRAN is older, but only by one year). Parts of ROS (the Robot Operating System)are written in LISP, although you don't need to know it to use ROS.

Prolog is a logic programming language and allows programmers torepresent "knowledge" in a form that an AI algorithm can understand. Prolog was used as part of the programming in IBM's Watson AI.

It is also possible to program artificial intelligence using some of the other languages on this list and more that are not listed. However, LISPand Prolog remain at the core of some AI implementations and certainlydeserve their place on this list. It's also worth remembering that roboticsand AI are not the same thing.

5. Hardware Description Languages (HDLs)

Hardware Description Languages are basically a programming way ofdescribing electronics. These languages will be very familiar to electronic engineers who create the low-level electronics of robots.

HDLs are commonly used to program Field Programmable Gate Arrays(FPGAs). These devices allow you to develop electronic hardware without having to actually produce a silicon chip, which makes them a quicker and easier option for some development tasks.

If you don't create prototypes of robotic electronics in your job, you maynever use HDLs. Even so, it is important to know that they exist as they

are quite different from other programming languages. For one thing, alloperations are carried out in parallel, rather than sequentially as with processor-based languages.

6. MATLAB

MATLAB (and its open-source relatives like Octave) is very popular withsome robotic engineers for analyzing data and developing control systems. It is used extensively in research and data processing. It's alsoused extensively in some university courses.

For robotics, there is also a very popular Robotics Toolbox for MATLAB. I know people who have developed entire robotics systemsusing MATLAB alone. If you want to analyze data, produce advancedgraphs or implement control systems, you will probably want to learn MATLAB.

7. C#/.NET

C# is a proprietary programming language provided by Microsoft. Iinclude C#/.NET here for two reasons:

It is the primary language of the Microsoft Robotics Developer Studio. Ifyou are going to use this system, you're probably going to have to use C#.

It is used as the basis for some Virtual Reality engines, like Unity, whichare growing in popularity right now.

C# is not the easiest option to learn first as it is a complex language. I would usually recommend learning C/C++ first. However, it is certainlynecessary in some areas of robotics.

8. Java

If you come to robotics from a computer science background (and manypeople do) you will probably already have learned Java first.

As an electronics engineer, I have never understood Java. I've always preferred languages that allowed for lower-level programming and morecontrol, like C. This is a good example of how people from different disciplines within robotics have different programming preferences — just because I don't like it doesn't mean you shouldn't.

Like C# and MATLAB, Java is an interpretive language, which meansthat it is not compiled into machine code. Rather, the Java Virtual

Machine interprets the instructions at runtime, allowing you to use thesame code on many different machines.

Java is quite popular in some parts of robotics. It's apparently one of thecore languages of several modern AIs, including IBM's Watson and AlphaGo.

9. Python

Python is on a roll at the moment. According to statistics it is has grownrapidly to become one of the top languages.

One of the reasons for its popularity in robotics is probably that Python(and C++) are the two main programming languages found in ROS.

The prime focus of the language is ease-of-use. Many people agree thatit achieves this very well. Python dispenses with a lot of the usual thingswhich take up time in programming, such as defining and casting variable types. Like Java, it is an interpreted language.

There are also a huge number of free libraries for Python which means you don't have to "reinvent the wheel" when you need to implement some basic functionality. And since it allows simple bindings with C/C++code, the performance-heavy parts of the code can be implemented in these languages to avoid performance loss.

With more and more robotics-friendly electronics now supporting Python"out-of-the-box" (e.g. Raspberry Pi), we are likely to continue to see a lotmore Python in robotics.

10. C/C++

Finally, we reach the Number 1 programming language in robotics!

Many people agree that C and C++ are required languages in robotics.Why? Because a lot of hardware libraries used in robotics use one of these languages. These libraries allow interaction with low-level hardware, allow for real-time performance and are very mature programming languages. These days, you'll probably use C++ more than C, although C remains one of the most efficient programming languages available.

C/C++ are not as simple to use as, say, Python or MATLAB. It can takequite a lot longer to implement the same functionality using C and it willrequire many more lines of code. However, as robotics is very dependent on real-time performance, C and C++ are probably the

closest thing that we roboticists have to "a standard language". This is trueeven despite the growing popularity of Python.

However, if you are a technician, you are still more likely to have to use themanufacturer's language.

ROBOT PROGRAMMING

- Robot Programming Revisited • Robot Programming is the defining of desired motions so that the robot may perform them without human intervention. – identifying and specifying the robot configurations (i.e. the pose of the end-effector, Pe, with respect to the base-frame) 1.MANUAL METHOD 2.WALKTHROUGH METHOD 3.LEADTHROUGH METHOD 4.OFF-LINE PROGRAMMING
- Type of Robot Programming • Joint level programming – basic actions are positions (and possibly movements) of the individual joints of the robot arm: joint angles in the case of rotational joints and linear positions in the case of linear or prismatic joints. • Robot-level programming – the basic actions are positions and orientations (and perhaps trajectories) of Pe and the frame of reference attached to it. • High-level programming – Object-level programming – Task-level programming
- pick-up part-A by side-A1 and side-A3 move part-A to location-2 pick-uppart-B by side-B1 and side-B3 put part-B on-top-off part-A with side-A5 in-plane-with side-B6 and with side-A1 in-plane-with side-B1 and with side-A2 in-plane-with side-B2 Object Level Programming • basic actionsare operations to be performed on the parts, or relationships that mustbe established between parts
- paint-the car-body red assemble the gear-box Task Level Programming basic actions specified by the program are complete tasks or subtasks
- ROBOT PROGRAMMING -Typically performed using one of the following – On line teach pendant lead through programming – Off line robotprogramming languages task level programming

Robot Programming Methods • Offline: – write a program using a text-based robot programming language – does not need access to the robot until its finaltesting and implementation • On-line: – Use the robot to generate the program • Teaching/guiding the robot through a sequence of motions that canthem be executed repeatedly • Combination Programming: – Often

programming is a combination of on-line and off-line • on-line to teach locations in space • off-line to define the task or “sequence of operations"

Use of Teach Pendant • hand held device with switches used to control the robot motions • End points are recorded in controller memory • sequentiallyplayed back to execute robot actions • trajectory determined by robot controller • suited for point to point control applications

Lead Through Programming • lead the robot physically through the required sequence of motions • trajectory and endpoints are recorded, using a sampling routine which records points at 60-80 times a second • when played backresults in a smooth continuous motion • large memory requirements

Short question-

Q1-Define Embedded System with Example

An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task. For example, a fire alarm is an embedded system; it will sense only smoke .....A small scale embedded system may not have RTOS.

Q2-How many types of Architecture are apply in embedded system

There are basically two types of architecture that apply to embedded systems:1-Von Neumann architecture

2-Harvard architecture.

Q3- What are characteristics of Harvard architecture?

Harvard Architecture consists of Arithmetic Logic Unit, Data Memory, Input/output, Data Memory, Instruction Memory, and the Control Unit. Harvard Architecture, has separate memory for data and instructions. In that way, both instruction and data can be fetched at the same time, thus making it comfortable to the users.

Q4- How does Von Neumann architecture differ from Harvard?

Von Neumann architecture	Harvard architecture
➢ it is ancient computer architecture based on stored program computer concept.	It is modern computer architecture based on Harvard Mark I relay based model.

➢ Same physical memory address is used for instructions and data.	Separate physical memory address is used for instructions and data.
➢ There is common bus for data and instruction transfer.	Separate buses are used for transferring data and instruction.

Q5- What is robot and its application?

Robots are widely used in manufacturing, assembly, packing and packaging, mining, transport, earth and space exploration, surgery, weaponry, laboratoryresearch, safety, and the mass production of consumer and industrial goods.

Long Question-

Q1-Draw and explain internal Architecture of Embedded System.

Q2-Explain the Types of Embedded System.

GOOD LUCK