ADVANCED DYNAMIC 3D MODELING OF ROBOTICS

Dr. N. Balaji[1], Mr. G.Jerome Nithin Gladson[2],P.Sudharshan[3]

[1]Professor, Department of Mechanical Engineering Saveetha Engineering College Thandalam, Chennai-602105.

[2]Professor, Department of Mechanical Engineering Saveetha Engineering College Thandalam, Chennai-602105.

[3]UG Scholar, [1]Professor, Department of Mechanical Engineering Saveetha Engineering College Thandalam, Chennai-602105

Abstract— This project based to working on blue tooth another method to working on mobile through app we give the input task give from mobile. Mobile signal is taken from mobile to mobile input module through to mobile data controller signal from data controller to electronic control unit (ECU) command signal from ECU (Robotic Head) to legs or hands this legs are working through drive control of stepper motor hands are working through drive control of stepper motor also head turning also working with drive control of stepper motor. Robotic moving front or back foot path with motor or roller type also available switched mode power supply controls used motors are working with SMPS power supply only battery are available, this battery making charge through the charger control this charger input power supply alternating current 220 v AC charging facilities also available for options following spare parts are used: battery, mobile, mobile data controller modules, electronics control unit module, battery charger, stepper motor with motor mechanism, foot path or legs with roller, switched mode power supply module, body frames, if need means monitor control module

Keywords—Agriculture, IOT.

I INTRODUCTION

Introduction Robotics is concerned with the study of those machines that can replace human beings in the execution of a task, as regards both physical activity and decision making. The goal of the introductory chapter is to point out the problems related to the use of robots in industrial applications, as well as the perspectives offered by advanced robotics. A classification of the most common mechanical structures of robot manipulators and mobile robots is presented. Topics of modelling, planning and control are introduced which will be examined in the following chapters. The chapter ends with a list of references dealing with subjects both of specific interest and of related interest to those covered by this textbook.

ROBOTICS:

Robotics has profound cultural roots. Over the course of centuries, human beings have constantly attempted to seek substitutes that would be able to mimic their behaviour in the various instances of interaction with the surrounding environment. Several motivations have inspired this continuous search referring to philosophical, economic, social and scientific principles. Oneof human beings' greatest ambitions has been to give life to their artifacts. The legend of the Titan Prometheus, who molded humankind from clay, as well as that of the giant Talus, the bronze slave forged by Hephaestus, testify how Greek mythology was influenced by that ambition, which has been revisited in the tale of Frankenstein in modern times. Just as the giant Talus was entrusted with the task of protecting the island of Crete from invaders, in the Industrial Age a mechanical creature (automaton) has been entrusted with the task of substituting a human being in subordinate labor duties. This concept was introduced by the Czech playwright Karel Capek who wrote the play ˇ Rossum's Universal Robots (R.U.R.) in 1920. On that occasion he coined the term robot derived from the term Introduction robot a that means executive labour in Slav languages to denote the automaton built by Rossum who ends up by rising up against humankind in the science fiction tale. In the subsequent years, in view of the development of science fiction, the behaviour conceived for the robot has often been conditioned by feelings. This has contributed to rendering the robot more and more similar to its creator. It is worth noticing how Rossum's robots were represented as creatures made with organic material.

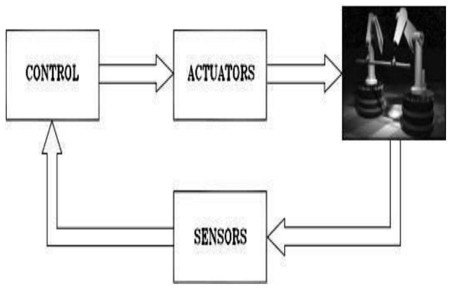


Fig.1

A. *ROBOT MECHANICAL STRUCTURE*

The key feature of a robot is its mechanical structure. Robots can be classified as those with a fixed base, robot manipulators, and those with a mobile base, mobile robots. In the following, the geometrical features of the two classesare presented.

B. *ROBOT MANIPULATORS*

The mechanical structure of a robot manipulator consists of a sequence of rigid bodies (links) interconnected by means of articulations (joints); a manipulator is characterized by an arm that ensures mobility, a wrist that confers dexterity, and an end-effector that performs the task required of the robot. The fundamental structure of a manipulator is the serial or open kinematic chain. From a topological viewpoint, a kinematic chain is termed open when there is only one sequence of links connecting the two ends of the chain. Alternatively, a manipulator contains a closed kinematic chain when a sequence of links forms a loop. A manipulator's mobility is ensured by the presence of joints. The articulation between two consecutive links can be realized by means of either a prismatic or a revolute joint. In an open kinematic chain, each prismatic or revolute joint provides the structure with a single degree offreedom (DOF). A prismatic joint creates a relative translational motion between the two links, whereas a revolute joint creates a relative rotational motion between the two links.

II. TYPICAL APPLICATIONS

- palletizing (placing objects on a pallet in an ordered way),
- warehouse loading and unloading,
- mill and machine tool tending,
- part sorting,
- packaging.

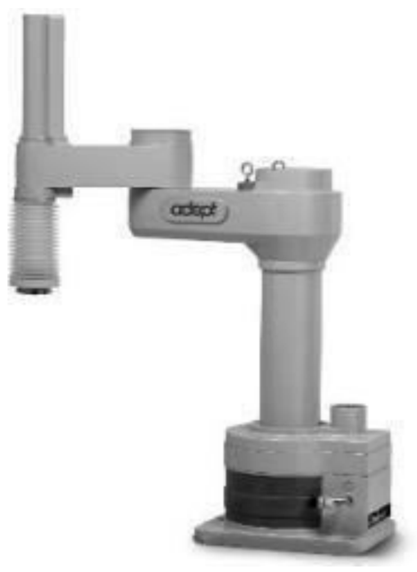
In these applications, besides robots, Automated Guided Vehicles (AGV) are utilized which ensure handling of parts and tools around the shop floorfrom one manufacturing cell to the next . As compared to the traditional fixed guide pathsfor vehicles (inductive guide wire, magnetic tape, or optical visible line), modern AGVs utilize high-tech systems with onboard microprocessors and sensors (laser, odometry, GPS) which allow their localization within the plant layout, and manage their work flow and functions, allowing their complete integration in the FMS.

The mobile robots employed in advanced applications can be considered as the natural evolution of the AGV systems, as far as enhanced autonomy is concerned. Manufacturing consists of transforming objects from raw material into finished products; during this process, the part either changes its own physical characteristics as a result of machining, or loses its identity as a result of an assemblyof more parts.

The robot's capability to manipulate both objects and tools make it suitable to be employed in manufacturing. Typical applications include:

- arc and spot welding,
- painting and coating,
- gluing and sealing,
- laser and water jet cutting,
- milling and drilling,
- casting and die spraying,
- deburring and grinding,
- screwing, wiring and fastening,
- assembly of mechanical and electrical groups,

Fig. 2



LITERATURE REVIEW

Autonomous vehicles are also employed for civil applications, i.e., for mass transit systems, thus contributing to the reduction of pollution levels. Such vehicles are part of the so-called Intelligent Transportation Systems (ITS) devoted to traffic management in urbanareas. Another feasible application where the adoption of mobile robots offers potential advantages is museum guided tours. Many countries are investing in establishing the new market of service robots which will co-habitat with human beings in everyday life. autonomous wheelchairs, mobility aid lifters, feeding aids and rehabilitation robots allowing tetraplegics to perform manual labor tasks are examples of such service devices. In perspective, other than an all-purpose robot waiter, assistance, and healthcare systems integrating robotic and telematic modules will be developed for home service management (domotics). Several robotic systems are employed for medical applications. Surgery assistance systems exploit a robot's high accuracy to position a tool, i.e., for hip prosthesis implant. Yet, in minimally-invasive surgery, i.e., cardiac surgery, the surgeon operates while seated comfortably at a console viewing a 3D image of the surgical field, and operating the surgical instruments remotely by means of a haptic interface (Fig. 1.40). Further, in diagnostic and endoscopic surgery systems, small teleoperated robots travels through the cavities of human body, i.e., in the gastrointestinal system, bringing live images or intervening in situ for biopsy, dispensing drugs or removing neoplasms.

Fig. 3

Finally, in motor rehabilitation systems, a hemiplegic patient wears an exoskeleton, which actively interacts, sustains and corrects the movements according to the physiotherapist's programmed plan. Another wide market segment comes from entertainment, where robots are used as toy companions for children, and life companions for the elderly, such as humanoid robots and the pet robots being developed in Japan. It is reasonable to predict that service robots will be naturally integrated into our society.

III. WORKING AND CONSTRUCTION

A. *DIFFERENTIAL KINEMATICS AND STATICS*

In the previous chapter, direct and inverse kinematics equations establishing the relationship between the joint variables and the end-effector pose were derived. In this chapter, differential kinematics is presented which gives the relationship between the joint velocities and the corresponding end-effector linear and angular velocity. This mapping is described by a matrix, termed geometric Jacobian, which depends on the manipulator configuration. Alternatively, if the end-effector pose is expressed with reference to a minimal representation in the operational space, it is possible to compute the Jacobian matrix via differentiation of the direct kinematics function with respect to the joint variables. The resulting Jacobian, termed analytical Jacobian, in general differs from the geometric one. The Jacobian constitutes one of the most important tools for manipulator characterization; in fact, it is useful for finding singularities, analyzing redundancy, determining inverse kinematics algorithms, describing the mapping between forces applied to the end-effector and resulting torques at the joints (statics) and, as will be seen in the following chapters, deriving dynamic equations of motion and designing operational space control schemes. Finally, the kineto-statics duality concept is illustrated, which is at the basis of the definition of velocity and force manipulability ellipsoids.

B. *TRAJECTORY PLANNING*

For the execution of a specific robot task, it is worth considering the main features of motion planning algorithms. The goal of trajectory planning is to generate the reference inputs to the motion control system which ensures that the manipulator executes the planned trajectories. The user typically specifies a number of parameters to describe the desired trajectory. Planning consists of generating a time sequence of the values attained by an interpolating function (typically a polynomial) of the desired trajectory. This chapter presents some techniques for trajectory generation, both in the case when the initial and final point of the path are assigned (point-to-point motion), and in the case when a finite sequence of points are assigned along the path (motion through a sequence of points). First, the problem of trajectory planning in the joint space is considered, and then the basic concepts of trajectory planning in the operational space are illustrated. The treatment of the motion planning problem for mobile robots is deferred to Chap.

C. *PATH AND TRAJECTORY*

The minimal requirement for a manipulator is the capability to move from an initial posture to a final assigned posture. The transition should be characterized by motion laws requiring the actuators to exert joint generalized forces confusion between terms often used as synonyms, the difference between a path and a trajectory is to be explained. A path denotes the locus of points in the joint space, or in the operational space, which the manipulator has to follow in the execution of the assigned motion; a path is then a pure geometric description of motion. In principle, it can be conceived that the inputs to a trajectory planning algorithm are the path description, the path constraints, and the constraints imposed by manipulator dynamics, whereas the outputs are the end- effector trajectories in terms of a time sequence of the values attained by position, velocity and acceleration. A geometric path cannot be fully specified by the userfor obvious complexity reasons. Typically, a reduced number of parameters is specified such as extremal points, possible intermediate points, and geometric primitives interpolating the points. Also, the motion timing law is not typically specified at each point of the geometric path, but rather it regards the total trajectory time, the constraints on the maximum velocities and accelerations, and eventually the assignment of velocity and acceleration at points of particular interest. On the basis of the above information, the trajectory planning algorithm generates a time sequence of variables that describe end-effector position and orientation over time in respect of the imposed constraints. Since the control action on the manipulator is carried out in the joint space, a suitable inverse kinematics algorithm is to be used to reconstruct the time sequence of joint variables corresponding to the above sequence in the operational space. Trajectory planning in the operational space naturally allows the presence of path constraints to be accounted; these are due to regions of workspace which are forbidden to the manipulator, e.g., due to the presence of obstacles. In fact, such constraints are typically better described in the operational space, since their corresponding points in the joint space are difficult to compute.

D. JOINT SPACE TRAJECTORIES

A manipulator motion is typically assigned in the operational space in terms of trajectory parameters such as the initial and final end-effector pose, possible intermediate poses, and travelling time along particular geometric paths. If it is desired to plan a trajectory in the joint space, the values of the joint variables have to be determined first from the end-effector position and orientation specified by the user. It is then necessary to resort to an inverse kinematics algorithm, if planning is done off-line, or to directly measure the above variables, if planning is done by the teaching-by-showing technique.

The planning algorithm generates a function q(t) interpolating the given vectors of joint variables at each point, in respect of the imposed constraints. In general,a joint space trajectory planning algorithm is required to have the following features:

- the generated trajectories should be not very demanding from a computational viewpoint,
- the joint positions and velocities should be continuous functions of time(continuity of accelerations may be imposed, too),
- undesirable effects should be minimized,

IV. OBJECTIVE

In this paper, a drilling robot based on earthworm locomotion was developed for seafloor exploration. Seabed mineral resources are found on the bottom of the ocean. The drilling robot developed herein can excavate and obtain samples of seafloor soil.
Second, the drilling resistance was reduced by the adjustment of the penetration and rotational speeds of the drilling robot based on the drilling properties of underwater ground. Lastly, the shape of the earth auger was considered for the reduction of the drilling torque.

- DCAD model (Humanoid robot)
- Design of system
- Fabrication Process

In Chap. 4, trajectory planning techniques have been presented which allow the generation of the reference inputs to the motion control system. The problem of controlling a manipulator can be formulated as that to determine the time history of the generalized forces (forces or torques) to be developed by the joint actuators, so as to guarantee execution of the commanded task while satisfying given transient and steady-state requirements. The task may regard either the execution of specified motions for a manipulator operating in free space, or the execution of specified motions and contact forces for a manipulator whose end- effector is constrained by the environment. In view of problem complexity, the two aspects will be treated separately; first, motion control in free space, and then control of the interaction with the environment. The problem of motion control of a manipulator is the topic of this chapter.

A number of joint space control techniques are presented. These can be distinguished between decentralized control schemes, i.e., when the single manipulator joint is controlled independently of the others, and centralized control schemes, i.e., when the dynamic interaction effects between the joints are taken into account. Finally, as a premise to the interaction control problem, the basic features of operational space control schemes are illustrated.

A number of joint space control techniques are presented. These can be distinguished between decentralized control schemes, i.e., when the single manipulator joint is controlled independently of the others, and centralized control schemes, i.e., when the dynamic interaction effects between the joints

are taken into account. Finally, as a premise to the interaction control problem, the basic features of operational space control schemes are illustrated.

4, trajectory planning techniques have been presented which allow the generation of the reference inputs to the motion control system. The problem of controlling a manipulator can be formulated as that to determine the time history of the generalized forces (forces or torques) to be developed by the joint actuators, so as to guarantee

execution of the commanded task while satisfying given transient and steady-state requirements. The task may regard either the execution of specified motions for a manipulator operating in free space, or the execution of specified motions and contact forces for a manipulator whose end- effector is constrained by the environment. In view of problem complexity, the two aspects will be treated separately; first, motion control in free space, and then control of the interaction with the environment. The problem of motion control of a manipulator is the topic of this chapter.

A number of joint space control techniques are presented. These can be distinguished between decentralized control schemes, i.e., when the single manipulator joint is controlled independently of the others, and centralized control schemes, i.e., when the dynamic interaction effects between the joints are taken into account. Finally, as a premise to the interaction control problem, the basic features of operational space control schemes are illustrated.

interactive interface (VRFII) is first developed by projecting a virtual robot into the real scene with an augmented-reality (AR) device, aiming to implement offline teaching. Second, a visual-aid algorithm (VAA) is proposed to improve offline teaching accuracy. Third, a gesture and speech teaching fusion algorithm (GSTA) with the fingertip tactile force feedback is developed to obtain the natural teaching pattern and improve the interactive accuracy of teaching the real or virtual robot. More specifically, through the VRFII, the operator can use the GSTA and the VAA to teach the virtual robot naturally

H. PATH AND TRAJECTORY

I.

The minimal requirement for a manipulator is the capability to move from an initial posture to a final assigned posture. The transition should be characterized by motion laws requiring the actuators to exert joint generalized forces

confusion between terms often used as synonyms, the difference between a path and a trajectory is to be explained. A path denotes the locus of points in the joint space, or in the operational space, which the manipulator has to follow in the execution of the assigned motion; a path is then a pure geometric description of motion. On the other hand, a trajectory is a path on which a timing law is specified, for instance in terms of velocities and/or accelerations at each point.

In principle, it can be conceived that the inputs to a trajectory planning algorithm are the path description, the path constraints, and the constraints imposed by manipulator dynamics, whereas the outputs are the end-effector trajectories in terms of a time sequence of the values attained by position, velocity and acceleration. A geometric path cannot be fully specified by the userfor obvious complexity reasons. Typically, a reduced number of parameters is specified such as extremal points, possible intermediate points, and geometric primitives interpolating the points. Also, the motion timing law is not typically specified at each point of the geometric path, but rather it regards the total trajectory time, the constraints on the maximum velocities and accelerations, and eventually the assignment of velocity and acceleration at points of particular interest. On the basis of the above information, the trajectory planning algorithm generates a time sequence of variables that describe end-effector position and orientation over time in respect of the imposed constraints. Since the control action on the manipulator is carried out in the joint space, a suitable inverse kinematics algorithm is to be used to reconstruct the time sequence of joint variables corresponding to the above sequence in the operational space. Trajectory planning in the operational space naturally allows the presence of path constraints to be accounted; these are due to regions of workspace which are forbidden to the manipulator, e.g., due to the presence of obstacles. In fact, such constraints are typically better described in the operational space, since their corresponding points in the joint space are difficult to compute. With

regard to motion in the neighbourhood of singular configurations and presence of redundant DOFs, trajectory planning in the operational space may involve problems difficult to solve. In such cases, it may be advisable to specify the pathin the joint space, still in terms of a reduced number of parameters. Hence, a time sequence of joint variables has to be generated which satisfy the constraints imposed on the trajectory. For the sake of clarity, in the following, the case of joint space trajectory planning is treated first. The results will then be extended to the case of trajectories in the operational space.

J. JOINT SPACE TRAJECTORIES

A manipulator motion is typically assigned in the operational space in terms of trajectory parameters such as the initial and final end-effector pose, possible intermediate poses, and travelling time along particular geometric paths. If it is desired to plan a trajectory in the joint space, the values of the joint variables have to be determined first from the end-effector position and orientation specified by the user. It is then necessary to resort to an inverse kinematics algorithm, if planning is done off-line, or to directly measure the above variables, if planning is done by the teaching-by-showing technique.

The planning algorithm generates a function $q(t)$ interpolating the given vectors of joint variables at each point, in respect of the imposed constraints. In general, a joint space trajectory planning algorithm is required to have the following features:

- the generated trajectories should be not very demanding from a computational viewpoint,

- the joint positions and velocities should be continuous functions of time(continuity of accelerations may be imposed, too),
- undesirable effects should be minimized,

non smooth trajectories interpolating a sequence of pointson a path. At first, the case is examined when only the initial and final points on the path and the traveling time are specified (point-to-point); the results are then generalized to the case when also intermediate points along the path are specified (motion through a sequence of points). Without loss of generality, the single joint variable q(t) is considered.

K. POINT-TO-POINT MOTION

In point-to-point motion, the manipulator has to move from an initial to a final joint configuration in a given time tf . In this case, the actual end-effector path is of no concern. The algorithm should generate a trajectory which, in respect to the above general requirements, is also capable of optimizing some performance index whenthe joint is moved from one position to another. A suggestion for choosing the motion primitive may stem from the analysis of an incremental motion problem. Let I be the moment of inertia of a rigid body about its rotation axis. It is required to take the angle q from an initial value qi to a final value qf in a time tf . It is obvious that infinite solutions exist to this problem. Assumed that rotation is executed through a torque τ supplied by a motor, a solution can be found which minimizes the energy dissipated in the motor

prescribed path points or by generating the analytical motion primitive and the relative trajectory in a punctual way. In both cases, the time sequence of the values attained by the operational space variables is utilized in real time to obtain the corresponding sequence of values of the joint space variables, via an inverse kinematics algorithm. In this regard, the computational complexity induced by trajectory generation in the operational space and related kinematic inversion sets an upper limit on the maximum sampling rate to generate the above sequences. Since these sequences constitute the reference inputs to the motion control system, a linear microinterpolation is typically carried out. In this way, the frequency at which reference inputs are updated is increased so as to enhance dynamic performance of the system.

L. DYNAMICS

Derivation of the dynamic model of a manipulator plays an important role for simulation of motion, analysis of manipulator structures, and design of control algorithms. Simulating manipulator motion allows control strategies and motion planning techniques to be tested without the

need to use a physically available system. The analysis of the dynamic model can be helpful for mechanical design of prototype arms. Computation of the forces and torques required for the execution of typical motions provides useful information for designing joints, transmissions and actuators. The goal of this chapter is to present two methods for derivation of the equations of motion of a manipulator in the joint space. The first method is based on the Lagrange formulation and is conceptually simple and systematic. The second method is based on the Newton–Euler formulation and yields the model in a recursive form; it is computationally more efficient since it exploits the typically open structure of the manipulator kinematic chain. Then, a technique for dynamic parameter identification is presented. Further, the problems of direct dynamics and inverse dynamics are formalized, and a technique for trajectory dynamic scaling is

introduced, which adapts trajectory planning to the dynamic characteristics of the manipulator. The chapter ends with the derivation of the dynamic model of a manipulator in the operational space and the definition of the dynamic manipulability ellipsoid.

M. LAGRANGE FORMULATION

The dynamic model of a manipulator provides a description of the relationship between the joint actuator torques and the motion of the structure. With Lagrange formulation, the equations of motion can be derived in a systematic way independently of the reference coordinate frame. Once a set of variables qi,i = 1,...,n, termed generalized coordinates, are chosen which effectively describe the link positions of an n-DOF manipulator, the Lagrangian of the mechanical system can be defined as a function of the generalized coordinates

o. DYNAMIC PARAMETER IDENTIFICATION

The use of the dynamic model for solving simulation and control problems demands the knowledge of the values of dynamic parameters of the manipulator model. Computing such parameters from the design data of the mechanical structure is not simple. CAD modelling techniques can be adopted which allow

the computation of the values of the inertial parameters of the various components (links, actuators and transmissions) on the basis of their geometry and type of materials employed. Nevertheless, the estimates obtained by such techniques are inaccurate because of the simplification typically introduced by geometric modelling; moreover, complex dynamic effects, such as joint friction, cannot be taken into account. A heuristic approach could be to dismantle the various components of the manipulator and perform a series of measurements to evaluate the inertial parameters. Such technique is not easy to implement and may be

troublesome to measure the relevant quantities. In order to find accurate estimates of dynamic parameters, it is worth resorting to identification techniques which conveniently exploit the property of linearity (7.81) of the manipulator model with respect to a suitable set of dynamic parameters. Such

techniques allow the computation of the parameter j vector π from the measurements of joint torques τ and of relevant quantities for the evaluation of the matrix Y , when suitable motion trajectories are imposed to the manipulator. On the assumption that the kinematic parameters in the matrix Y are known with good accuracy, e.g., as a result of a kinematic calibration, measurements of joint positions q, velocities $\dot{q}$ and accelerations $\ddot{q}$ are required. Joint positions and velocities can be actually measured while numerical reconstruction of accelerations is needed; this can be performed on the basis of the position and velocity values recorded during the execution of the trajectories. The reconstructing filter does not work in real time and thus it can also be anti- causal, allowing an accurate reconstruction of the accelerations.

V OBJECTIVE

- **In this paper, a drilling robot based on earthworm locomotion was developed for seafloor exploration. Seabed mineral resources are found on the bottom of the ocean. The drilling robot developed herein can excavate and obtain samples of seafloor soil.**
- **Second, the drilling resistance was reduced by the adjustment of the penetration and rotational speeds of the drilling robot based on the drilling properties of underwater ground. Lastly, the shape of the earth auger was considered for the reduction of the drilling torque.**
- **DCAD model (Humanoid robot)**
- **Design of system**
- **Fabrication Process**

VI METHOLOGY

- **Robots can be fully autonomous, semi-autonomous or controlled by human operators.**
- **A fully autonomous robot will make all of its decisions and actions without direct human intervention.**
- **semi-autonomous robot will have some degree of autonomy (e.g. some tasks, like balancing or object tracking, will be fully automated, while others will be controlled by human operator)**
- **while non-autonomous robots are directly controlled by an operator**

In Chap. 4, trajectory planning techniques have been presented which allow the generation of the reference inputs to the motion control system. The problem of controlling a manipulator can be formulated as that to determine the time history of the generalized forces (forces or torques) to be developed by the joint actuators, so as to guarantee execution of the commanded task while satisfying given transient and steady-state requirements. The task may regard either the execution of specified motions for a manipulator operating in free space, or the execution of specified motions and contact forces for a manipulator whose end- effector is constrained by the environment. In view of problem complexity, the two aspects will be treated separately; first, motion control in free space, and then control of the interaction with the environment. The problem of motion control of a manipulator is the topic of this chapter.

A number of joint space control techniques are presented. These can be distinguished between decentralized control schemes, i.e., when the single manipulator joint is controlled independently of the others, and centralized control schemes, i.e., when the dynamic interaction effects between the joints are taken into account. Finally, as a premise to the interaction control problem, the basic features of operational space control schemes are illustrated.

6.1 THE CONTROL PROBLEM

Several techniques can be employed for controlling a manipulator. The technique followed, as well as the way it is implemented, may have a significant influence on the manipulator performance and then on the possible range of applications. For instance, the need for trajectory tracking control in the operational space may lead to hardware/software implementations, which differ from those allowing point-to-point control, where only reaching of the final position is of concern. On the other hand, the manipulator mechanical design.

FIG 4 . ARDUINO

A. *I^{2}C*

Inter-integrated circuit (I^{2}C), pronounced either "i-squared-c" or "i-two-c," is the final communication protocol we'll cover in this tutorial. Though its implementation is the most complicated of the three protocols, I^{2}C addresses several drawbacks in the other communication protocols, giving it an advantage over the others in some applications. These include:

- The ability to connect multiple masters to multiple slaves Synchronicity (just like SPI), which means higher speedcommunication.
- Simplicity: implementation only requires two wires andsome resistors

B. *ESP32*

ESP32 is a low-powered, low-cost microcontroller (MCU) board, with both Wi-Fi and Bluetooth built in, and is based on a dual-core processor mechanism. The first one is a powerful processor, such as a Xtensa LX6 (~240 MHz) with 512 KiB memory and the second an ultra-low coprocessor (ULP) with only 8 KiB memory designed to run when ESP32 is in deep- sleep mode.

Other components include around 48 I/O pins (variable); an array of peripheral interfaces including temperature, hall effect, and capacitive touch sensors; and an 8- centimeter LCD panel, prominently visible here in an ESP32-WROVER boardby Express if Systems.

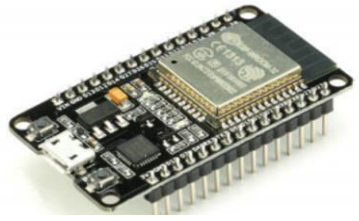


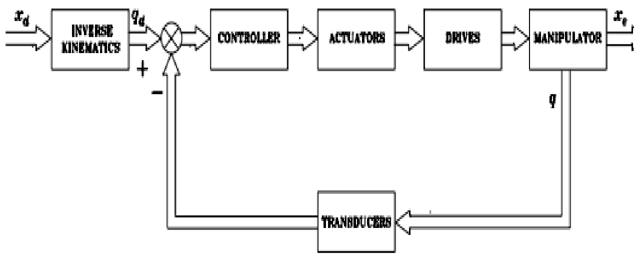
FIG 5. ESP32

C. *GPRS:*

GPRS stands for General Packet Radio Service. It is a mobile data service that enables cellular devices to transmit and receive data over the internet. GPRS is a packet- switched technology that divides the data into small packetsand transmits them over the network. This allows for faster data transfer rates than traditional circuit-switched technology.
GPRS is commonly used for mobile internet access, email, multimedia messaging, and other data-intensive applications. It is also used for machine-to-machine (M2M)communications, such as remote monitoring and control ofdevices.

D. *NRF24L01:*

The NRF24L01 is a 2.4 GHz wireless transceiver module, designed for low-power, short-range wireless communication in embedded systems. It is commonly used in applications such as wireless sensor networks, home automation, remote control, and wireless toys. The NRF24L01 module supports arange of data rates from 250 kbps to 2 Mbps, and can operate at distances up to 100 meters in open space. It uses a frequency-hopping technique to minimize interference with other wireless devices operating in the same frequency band.



the operational space, whereas control actions (joint actuator generalized forces) are performed in the joint space. This fact naturally leads to considering two kinds of general control schemes, namely, a joint space controlscheme (Fig. 8.1) and an operational space control scheme (Fig. 8.2). In both schemes, the control structure has closed loops to exploit the good features provided by feedback, i.e., robustness to modelling uncertainties andreduction of disturbance effects. In general terms, the following considerations should be made.

The joint space control problem is actually articulated in two subproblems. First, manipulator inverse kinematics is solved to transform the motion requirements xd from the operational space into the corresponding motion qd in the joint space. Then, a joint space control scheme is designed that allows the actual motion q to track the reference inputs. However, this solution has the drawback that a joint space control scheme does not influence the operational space variables xe which are controlled in an open-loop fashion through the manipulator mechanical structure. It is then clear that any uncertainty of the structure (construction tolerance, lack of calibration, gear backlash, elasticity) or any imprecision in the knowledge of the end-effector pose relative to an object to manipulate causes a loss of accuracy on the operational space variables. The operational space control problem follows a global approach that requires a greater algorithmic complexity; notice that inverse kinematics is now embedded into the feedback control loop. Its conceptual advantage regards the possibility of acting directly on operational space variables; this is somewhat only a potential advantage, since measurement of operational space variables is often performed not directly, but through the evaluation of direct kinematics functions starting from measured joint space variables. On the above premises, in the following, joint space control schemes for manipulator motion in the free space are presented first. In the sequel, operational space control schemes will be illustrated which are logically at the basis of control of the interaction with the environment.

a Cartesian manipulator is substantially different from that of an anthropomorphic manipulator. The driving system of the joints also has an effect on the type of control strategy used. If a manipulator is actuated by electric motors with reduction gears of high ratios, the presence of gears tendsto linearize system dynamics, and thus to decouple the joints in view of the reduction of nonlinearity effects.

The price to pay, however, is the occurrence of joint friction, elasticity and backlash that may limit system performance more than it is due to configuration-dependent inertia, Coriolis and centrifugal forces, and so forth. On the other hand, a robot actuated with direct drives eliminates the drawbacks due to friction, elasticity and backlash, but the weight of nonlinearities and couplings between the joints becomes relevant. As a consequence, different control strategies have to be thought of to obtain high performance. Without any concern to the specific type of mechanical manipulator, it is worth remarking that task specification (end-effector motion and forces) is usually carried out in the operational space, whereas control actions (joint actuator generalized forces) are performed in the joint space. This fact naturally leads to considering two kinds of general control schemes, namely, a joint space control scheme (Fig. 8.1) and an operational space control scheme (Fig. 8.2). In both schemes, the control structure has closed loops to exploit the good features provided by feedback, i.e., robustness to modelling uncertainties and reduction of disturbance effects. In general terms, the following considerations should be made.

Fig. 6

The joint space control problem is actually articulated in two subproblems. First, manipulator inverse kinematics is solved to transform the motion requirements xd from the operational space into the corresponding motion qd in the joint space. Then, a joint space control scheme is designed that allows the actual motion q to track the reference inputs. However, this solution has the drawback that a joint space control scheme does not influence the operational space variables xe which are controlled in an open-loop fashion through the manipulator mechanical structure. It isthen clear that any uncertainty of the structure (construction tolerance, lack of calibration, gear backlash, elasticity) or any imprecision in the knowledge of the end- effector pose relative to an object to manipulate causes a loss of accuracy on the operational space variables. The operational space control problem follows a global approach that requires a greater algorithmic complexity; notice that inverse kinematics is now embedded into the feedback contr;;o)l loop. Its conceptual advantage regards the possibility of ij
acting directly on operational space variables; this is somewhat only a potential advantage, since measurement of operational space variables is often performed not directly, but through the evaluation of direct kinematics functions starting from measured joint space variables. On the above premises, in the following, joint space control schemes for manipulator motion in the free space are presented first. In the sequel, operational space control schemes will be illustrated which are logically at the basis of control of the interaction with the environment.

DECENTRALIZED FEEDFORWARD COMPENSATION

When the joint control servos are required to track reference trajectories with high values of speed and acceleration, the tracking capabilities of the scheme in Fig. 8.6 are unavoidably degraded. The adoption of a decentralized feedforward compensation allows a reduction of the tracking error. Therefore, in view of the closed-loop input/output transfer functions in (8.24), (8.28), (8.35),

E. *MOTOR DRIVE & MOTOR*

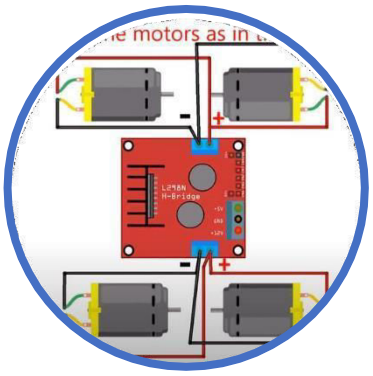


FIG 7. MOTOR DRIVE & MOTOR

i) Motor drive:

This L298N Motor Driver Module is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5Vregulator. L298N Module can control up to 4 DC motors, or 2 DC motors with directional andspeed control.

Motor:

A DC motor or Direct Current Motor converts electrical energy into mechanical energy. A direct current (DC) motor is a fairly simple electric motor that uses electricity and a magnetic field to produce torque, which turns the rotor and hence give mechanical work The presence of saturation blocks in the schemes of Figs. 8.12, 8.13, 8.14 is to be intended as intentional nonlinearities whose function is to limit relevant physical quantities during transients; the greater the number of feedback loops, the greater the number of quantities that can be limited (velocity, acceleration, and motor voltage). To this end, notice that trajectory tracking is obviously lostwhenever any of the above quantities saturates. This situation often occurs for industrial manipulators required to execute point-to-point motions; in this case, there is less concern about the actual trajectories followed, and the actuators are intentionally taken to operate at the current limits so as to realize the fastest possible motions.

OPERATIONAL SPACE CONTROL

In all the above control schemes, it was always assumed that the desired trajectory is available in terms of the time sequence of the values of joint position, velocity and acceleration. Accordingly, the error for the control schemes was expressed in the joint space. As often pointed out, motion specifications are usually assigned in the operational space, and then an inverse kinematics algorithm has to be utilized to transform operational space references into the corresponding joint space references. The process of kinematic inversion has an increasing computational load when, besides inversion of direct kinematics, inversion of first-order and second- order differential kinematics is also required to transformthe desired time history of end-effector position, velocity and acceleration into the corresponding quantities at the joint level. It is for this reason that current industrial robot control systems compute the joint positions through kinematics inversion, and then perform a numerical differentiation to compute velocities and accelerations. A different approach consists of considering control schemesdeveloped directly in the operational space. If the motion is specified in terms of operational space variables, the measured joint space variables can be transformed into the corresponding operational space variables through direct kinematics relations. Comparing the desired input with the reconstructed variables allows the design of feedback control loops where trajectory inversion is replaced with a suitable coordinate transformation embedded in the feedback loop

SQUARE TUBE

A hollow structural section (HSS) is a type of metal profile with a hollow cross section. The term is used predominantly in the United States, or other countries which follow US construction or engineering terminology.

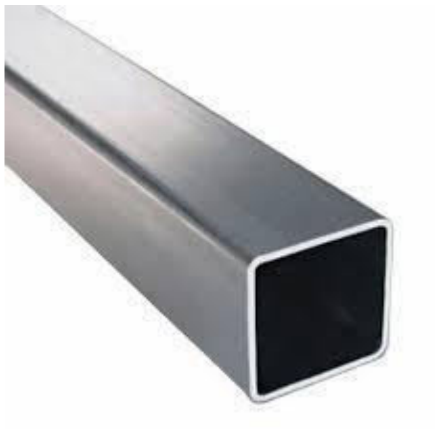


Fig. 8

HSS members can be circular, square, or rectangular sections, although other shapes such as elliptical are also available. HSS is only composed of structural steel per code.

HSS is sometimes mistakenly referenced as hollow structural steel. Rectangular and square HSS are also commonly called tube steel or box section. Circular HSS are sometimes mistakenly called steel pipe, although true steel pipe is actually dimensioned and classed differently from HSS. (HSS dimensions are based on exterior dimensions of the profile; pipes are also manufactured toan exterior tolerance, albeit to a different standard.) The corners of HSS are heavily rounded, having a radius which is approximately twice the wall thickness. The wall thickness is uniform around the section.

WIPER MOTOR

A DC motor is an electrical motor that uses direct current (DC) to produce mechanical force. The most common types rely on magnetic forces produced by currents in the coils. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of currentin part of the motor.

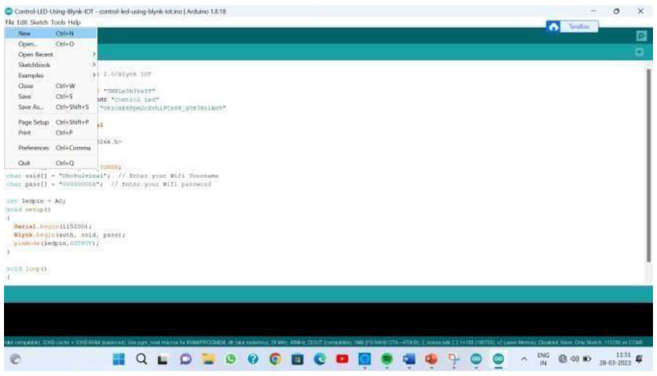
DC motors were the first form of motors widely used, as they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances. The universal motor, a lightweight brushed motor used for portable power tools and appliances can operate on direct current and alternating current. Larger DC motors are currently used in propulsion of electric vehicles, elevator and hoists, and in drives for steel rolling mills. The advent of power electronics has made replacement of DC motors with AC motors possible in many applications.

- **The network coverages of the four placement are 100%, 91%, 52%, 94%, respectively. The deployments are verified to have reasonable sensor placement and meet the requirements of optimization goals, which is as expected.**

IV. SOFTWARE SPECIFICATION

i) ARDUINO IDE – 1.8.5

arduino is an open-source electronics platform based on easy-to-use hardware and software. arduino boards are able to readinputs - light on a sensor, a finger on a button, or a twitter message - and turn it into an output - activating a motor, turning on an led, publishing something online. you can tell your board what to do by sending a set of instructions to the microcontroller on the board. to do so you use the arduino programming language (based on wiring), and the arduino software (ide), based on processing.



```
char ssid[] = "Chokulvinai";  // Enter your Wifi Username
char pass[] = "00000000k";  // Enter your Wifi password

int ledpin = A0;
void setup()
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  pinMode(ledpin,OUTPUT);
}

void loop()
{
```

FIG 9. ARDUINO IDE

EMBEDDED C:

Embedded c is a set of language extension for the C Programming language by the C Standards committee to address commonality issues that exist between C extensions for different embedded systems. Historically embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations.

Embedded C uses most of the syntax and semantics of standard C, e.g., main () function, variable definition, data type declaration, conditional statements (if, switch, case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc. During infancy years of microprocessor-based systems, programs were developed using assemblers and fused into the EPROMs. There used to be no mechanism to find what the program was doing. LEDs, switches, etc. were used to check for correct execution of the program. But they were too costly and were not quite reliable as well. As time progressed, use of microprocessor-specific assembly-only as the programming language reduced and embedded systems

6.1 ARDUINO UNO

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.[1] The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative

Commons Attribution Share-Alike 2.5 license and is available on the Arduinowebsite. Layout and production files for some versions of the hardware are also available.

The word "uno" means "one" in Italian and was chosen to mark the initial release of Arduino Software. The Uno board is the first in a series of USB- based Arduino boards; it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases. The ATmega328 on the board comes pre programmed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.

While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

GENERAL PIN FUNCTIONS

- **LED: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it is off.**
- **VIN: The input voltage to the Arduino/Genuino board when it is using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.**
- **5V: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.**
- **3V3: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.**
- **GND: Ground pins.**
- **IOREF: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source, or enable voltage translators on the outputs to work with the 5V or 3.3V.**
- **Reset: Typically used to add a reset button to shields that block the one on theboard.**
- **Key/EN: It is used to bring Bluetooth module in AT commands mode. If Key/EN pin is set to high, then this module will work in command mode. Otherwise by default it is in data mode. The default baud rate of HC-05 in command mode is 38400bps and 9600 in data mode.**

HC-05 module has two modes,

- **Data mode: Exchange of data between devices.**
- **Command mode: It uses AT commands which are used to change setting of HC-05. To send these commands to module serial (USART) port isused.**

- **VCC: Connect 5 V or 3.3 V to this Pin.**
- **GND: Ground Pin of module.**
- **TXD: Transmit Serial data (wirelessly received data by Bluetooth moduletransmitted out serially on TXD pin)**
- **RXD: Receive data serially (received data will be transmitted wirelesslyby Bluetooth module).**
- **State: It tells whether module is connected or not.**

EMBEDDED SYSTEM

Embedded systems are controllers with on chip control. They consist of microcontrollers, input and output devices, memories etc., on chip and they can be used for a specific application. A small computer designed in a single chip is called a single chip microcomputer. A single chip microcomputer typically includes a microprocessor RAM, ROM, timer, interrupt and peripheral controller in a single chip. This single chip microcomputer is also called as microcontroller; These Microcontrollers are used for variety of applications where it replaces the computer. The usage of this microcomputer for a specific application, in which the microcontrollers a part of application, is called embedded systems. Embedded systems are used for real time applications with high reliability, accuracy and precision, Embedded systems are operated with Real Time Operating systems like WinCE, RT Linux, VxWorks, PSOS, etc.., Embedded systems are very popular these days Most of the Electrical, Electronics, Mechanical, Chemical, Industrial, Medical, Space and many more areas have the embedded systems in their applications

ROLE OF EMBEDDED SYSTEM

Embedded systems are compact, smart, efficient, and economical and user friendly, they are closed systems and respond to the real world situation very fast, closed system means, everything required for a specific application is embedded on the chip and hence, they do not call for external requirement fortheir functioning.

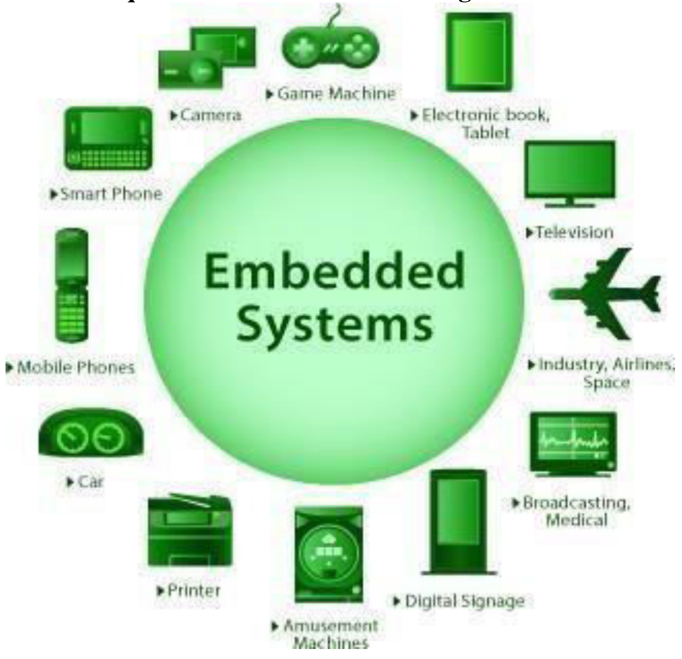


Fig. 10

6.1 *APPLICATIONS OF EMBEDDED SYSTEM*

- ❖ **Robotics**
- ❖ **Aviation**
- ❖ **Telecommunication and Broadcasting**
- ❖ **Mobile Phones and mobiles networking**
- ❖ **Satellite Communication**
- ❖ **Blue Tooth**
- ❖ **Electronic sensors**

- **Home Appliances etc.**

PERIPHERALS

Embedded Systems talk with the outside world via peripherals, such as:

- **Serial Communication Interfaces (SCI): RS-232, RS-422, RS-485 etc.**
- **Synchronous Serial Communication Interface: I2C, SPI, SSC and ESSI(Enhanced Synchronous Serial Interface)**
- **Multi Media Cards (SD Cards, Compact Flash etc.)**
- **Networks: Ethernet, Lon Works, etc.**
- **Fieldbuses: CAN-Bus, LIN-Bus, PROFIBUS, etc.**
- **Timers: PLL(s), Capture/Compare and Time Processing Units**
- **Discrete IO: aka General Purpose Input/Output (GPIO)**
- **Analog to Digital/Digital to Analog (ADC/DAC)**
- **Debugging: JTAG, ISP, ICSP, BDM Port, BITP, and DP9 ports.**

6.2 SOFTWARE

Embedded C is a set of language extensions for the C Programming the C standards committee to address commonality issues that exist between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed point arithmetic, multiple distinct memory banks and basic input output operations. In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing. Embedded C use most of the syntax and semantics of standard C, e.g., main () function.

Embedded Systems talk with the outside world via peripherals, such as:

- **Serial Communication Interfaces (SCI): RS-232, RS-422,RS-485 etc.**
- **Synchronous Serial Communication Interface: I2C, SPI,SSC and ESSI(Enhanced Synchronous Serial Interface)**
- **Universal Serial Bus (USB)**
- **Multi Media Cards (SD Cards, Compact Flash etc.)**
- **Networks: Ethernet, Lon Works, etc.**
- **Fieldbuses: CAN-Bus, LIN-Bus, PROFIBUS, etc.**
- **Timers: PLL(s), Capture/Compare and Time Processing Units**
- **Discrete IO: aka General Purpose Input/Output (GPIO)**
- **Analog to Digital/Digital to Analog (ADC/DAC)**
- **Debugging: JTAG, ISP, ICSP, BDM Port, BITP, and DP9 ports, variable definition, data type declaration, conditional statements (if, switch. case),**

In terms of communication, a robot will be better understood and accepted if its communication behavior more explicitly emulates that of a human. Common understanding should be reached by using the same conversational gestures used by humans, those of gaze, pointing, and hand and face gestures. Robots should also be able to interpret and display such behavior so that their communication appears natural to their human conversational partner actuation system which animates the mechanical components of the robot. The concept of such a system refers to the context of motion control, dealing with servomotors, drives and transmissions. The capability for perception is entrustedto a sensory system which can acquire data on the internal status of the mechanical system (proprioceptive sensors, such as position transducers) as well as on the external status of the environment (exteroceptive sensors, such as force sensors and cameras). The realization of such a system refers to the context of materials properties, signal conditioning, data processing, and information retrieval. The capability for connecting action to perception in an intelligent fashion is provided by a control system which can command the execution of the action in respect to the goals set by a task planning technique,as well as of the constraints imposed by the robot and the environment. The realization of such a system follows the same feedback principle devoted to control of human body functions, possibly exploiting the description of the robotic system's components (modelling). The context is that of cybernetics, dealing with control and supervision of robot motions, artificial intelligence and expert systems, the computational architecture and programming environment. Therefore, it can be recognized that robotics is an interdisciplinary subject concerning the cultural areas of mechanics, control, computers, and electronics.

ROBOT MECHANICAL STRUCTURE

The key feature of a robot is its mechanical structure. Robots can be classified as those with a fixed base, robot manipulators, and those with a mobile base,mobile robots. In the following, the geometrical features of the two classes are presented.

1.1 ROBOT MANIPULATORS

The mechanical structure of a robot manipulator consists of a sequence of rigid bodies (links) interconnected by means of articulations (joints); a manipulator is characterized by an arm that ensures mobility, a wrist that confers dexterity, and an end-effector that performs the task required of the robot. The fundamental structure of a manipulator is the serial or open kinematic chain. From a topological viewpoint, a kinematic chain is termed open when there is only one sequence of links connecting the two ends of the chain. Alternatively, a manipulator contains a closed kinematic chain when a sequence of links forms a loop. A manipulator's mobility is ensured by the presence of joints. The articulation between two consecutive links can be realized by means of either a prismatic or a revolute joint. In an open kinematic chain, each prismatic or revolute joint provides the structure with a single degree of freedom (DOF). A prismatic joint creates a relative translational motion between the two links, whereas a revolute joint creates a relative rotational motion between the two links.

Revolute joints are usually preferred to prismatic joints in view of their compactness and reliability. On the other hand, in a closed kinematic chain, the number of DOFs is less than the number of joints in view of the constraints imposed by the loop. The degrees of freedom should be properly distributed along the mechanical structure in order to have a sufficient number to execute a given task. In the most general case of a task consisting of arbitrarily positioning and orienting an object in three-dimensional (3D) space, six DOFs are required, three for positioning a point on the object and three for orienting the object with respect to a reference coordinate frame. If more DOFs than task variables are available, the manipulator is said to be redundant from a kinematic viewpoint. The workspace represents that portion of the environment the manipulator's end-effector can access. Its shape and volume depend on the

manipulator structure as well as on the presence of mechanical joint limits. The task required of the arm is to position the wrist which then is required to orient the end-effector. The type and sequence of the arm's DOFs, starting from the base joint, allows a classification of manipulators as Cartesian, cylindrical, spherical, SCARA, and anthropomorphic. Cartesian geometry is realized by three prismatic joints whose axes typically are mutually orthogonal (Fig. 1.2). Inview of the simple geometry, each DOF corresponds to a Cartesian space variable and thus it is natural to perform straight motions in space. The Cartesian structure offers very good mechanical stiffness. Wrist positioning accuracy is constant everywhere in the workspace. This is the volume enclosed by a rectangular parallel-piped

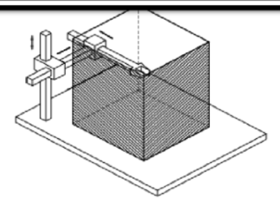


Fig. 1.2. Cartesian manipulator and its workspace

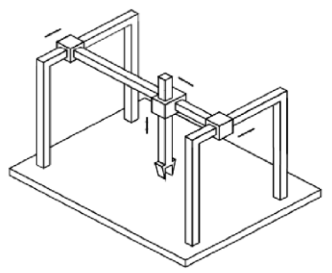


Fig. 1.3. Gantry manipulator

Fig. 11

As opposed to high accuracy, the structure has low dexterity since all the joints are prismatic. The direction of approach in order to manipulate an object is from the side. On the other hand, if it is desired to approach an object from the top, the Cartesian manipulator can be realized by a gantry structure as illustrated
Such a structure makes available a workspace with a large volume and enables the manipulation of objects of large dimensions and heavy weight. Cartesian manipulators are employed for material handling and assembly. The motors actuating the joints of a Cartesian manipulator are typically electric and occasionally pneumatic. Cylindrical geometry differs from Cartesian in that the first prismatic joint is replaced with a revolute joint (Fig. 1.4). If the task is described in cylindrical coordinates, in this case each DOF also corresponds to aCartesian space variable. The cylindrical structure offers good mechanical stiffness. Wrist positioning accuracy decreases as the horizontal stroke increases. The workspace is a portion of a hollow cylinder (Fig. 1.4). The horizontal prismatic joint makes the wrist of a cylindrical manipulator suitable to access horizontal cavities. Cylindrical manipulators are mainly employed for carrying objects even of large dimensions; in such a case the use of hydraulic motors is to be preferred to that of electric motors. Spherical geometry differs

from cylindrical in that the second prismatic joint is replaced with a revolute joint (Fig. 1.5). Each DOF corresponds to a Cartesian space variable provided that the task is described in spherical coordinates. Mechanical stiffness is lower than the above two geometries and mechanical construction is more complex. Wrist positioning accuracy decreases as the radial stroke increases.

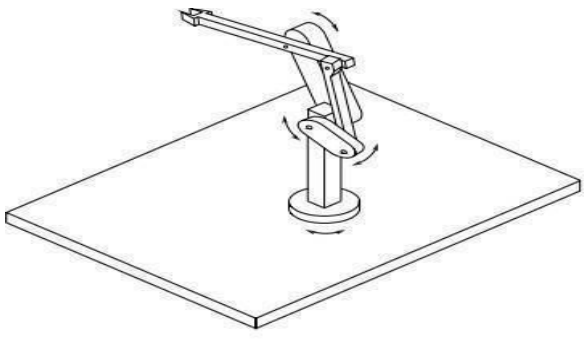


Fig. 1.8. Manipulator with parallelogram

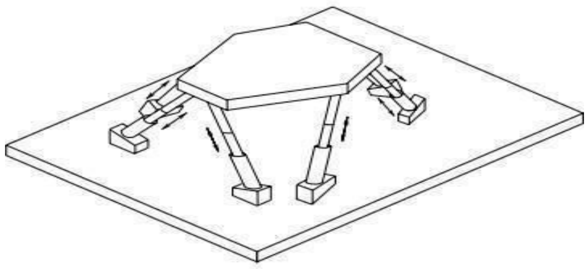


Fig. 1.9. Parallel manipulator

Fig. 12

it can allow manipulation of objects on the floor. Spherical manipulators are mainly employed for machining. Electric motors are typically used to actuate the joints. A special geometry is SCARA geometry that can be realized by disposing two revolute joints and one prismatic joint in such a way that all the axes of motion are parallel (Fig. 1.6). The acronym SCARA stands for Selective Compliance Assembly Robot Arm and characterizes the mechanical features of

a structure offering high stiffness to vertical loads and compliance to horizontal loads. As such, the SCARA structure is well-suited to vertical assembly tasks. The correspondence between the DOFs and Cartesian space variables is maintained only for the vertical component of a task described in Cartesian coordinates. Wrist positioning accuracy decreases as the distance of the wrist from the first joint axis increases. The typical workspace is illustrated

geometry illustrated in Fig. 1.10 is of hybrid type, since it consists of a parallel arm and a serial kinematic chain. This structure is suitable for the execution of manipulation tasks requiring large values of force along the vertical direction. The manipulator structures presented above are required to position the wrist which is then required to orient the manipulator's end-effector. If arbitrary orientation in 3D space is desired, the wrist must possess at least three DOFs provided by revolute joints. Since the wrist constitutes the terminal part of the manipulator, it has to be compact; this often complicates its mechanical design. Without entering into construction details, the realization endowing the wrist with the highest dexterity is one where the three revolute axes intersect at a single point. In such a case, the wrist is called a spherical wrist, as represented. The key feature of a spherical wrist is the decoupling between position and orientation of the end-effector; the arm is entrusted with the task of positioning the above point of intersection, whereas the wrist determines the end-effector orientation. Those realizations where the wrist is not spherical are simpler from a mechanical viewpoint, but position and orientation are coupled, and this complicates the coordination between the motion of the

CONCLUSION

In terms of communication, a robot will be better understood and accepted if its communication behavior more explicitly emulates that of a human. Common understanding should be reached by using the same conversational gestures used by humans, those of gaze, pointing, and hand and face gestures. Robots should also be able to interpret and display such behavior so that their communication appears natural to their human conversational partner

[1] Lee, E.A. Cyber Physical Systems: DesignChallenges. In Proceedings of the 11th IEEE InternationalSymposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC), Orlando, FL, USA, 5–7 May 2008; pp. 363–369.

[2] Bordel, B.; Alcarria, R.; Robles, T.; Martín, D. Cyber–physical systems: Extending pervasive sensing from control theory to the Internet of Things. Pervasive Mob. Comput. 2017, 40, 156–184.

[3] Mell, P.; Grance, T. The NIST Definition of Cloud Computing, Version 15, 10-7-09. National Institute of Standards and Technology. Information Technology Laboratory. Available online: https://csrc.nist.gov/ publications/detail/sp/800-145/final (accessed on 2 July 2020).

[4] Rane, MsDeweshvree, P. G. Scholar-VLSI, and Sevagram BDCE. "Review paper based on automatic irrigation system based on RF module." PG Scholar-VLSI, Sevagram, Wardha, india, IJAICT, ISSN (2014): 2348-9928.

[5] Abbasi, Abu Zafar, Noman Islam, and Zubair Ahmed Shaikh. "A review of wireless sensors and networks' applications in agriculture." Computer Standards & Interfaces 36.2 (2014): 263-270.

[6] Kamalaskar, H. N., and P. H. Zope. "INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY Survey of Smart Irrigation System.".

[7] Kansara, Karan, et al. "Sensor based Automated Irrigation System with IOT: A Technical Review." International Journal of Computer Science and Information Technologies 6.6 (2015)