Introduction to Robotics

- 2.12 Lecture Notes -

H. Harry Asada

Ford Professor of Mechanical Engineering

Chapter 1
Introduction

Many definitions have been suggested for what we call a *robot.* The word may conjure up various levels of technological sophistication, ranging from a simple material handling device to a humanoid. The image of robots varies widely with researchers, engineers, and robot manufacturers. However, it is widely accepted that today's robots used in industries originated in the invention of a programmed material handling device by George C. Devol. In 1954, Devol filed a U.S. patent for a new machine for part transfer, and he claimed the basic concept of *teach-in/playback* to control the device. This scheme is now extensively used in most of today's industrial robots.

1.1 Era of Industrial Robots

Devol's industrial robots have their origins in two preceding technologies: *numerical control* for machine tools, and *remote manipulation.* Numerical control is a scheme to generate control actions based on stored data. Stored data may include coordinate data of points to which the machine is to be moved, clock signals to start and stop operations, and logical statements for branching control sequences. The whole sequence of operations and its variations are prescribed and stored in a form of memory, so that different tasks can be performed without requiring major hardware changes. Modern manufacturing systems must produce a variety of products in small batches, rather than a large number of the same products for an extended period of time, and frequent changes of product models and production schedules require *flexibility* in the manufacturing system. The transfer line approach, which is most effective for mass production, is not appropriate when such flexibility is needed (Figure 1-1). When a major product change is required, a special-purpose production line becomes useless and often ends up being abandoned, despite the large capital investment it originally involved. Flexible automation has been a central

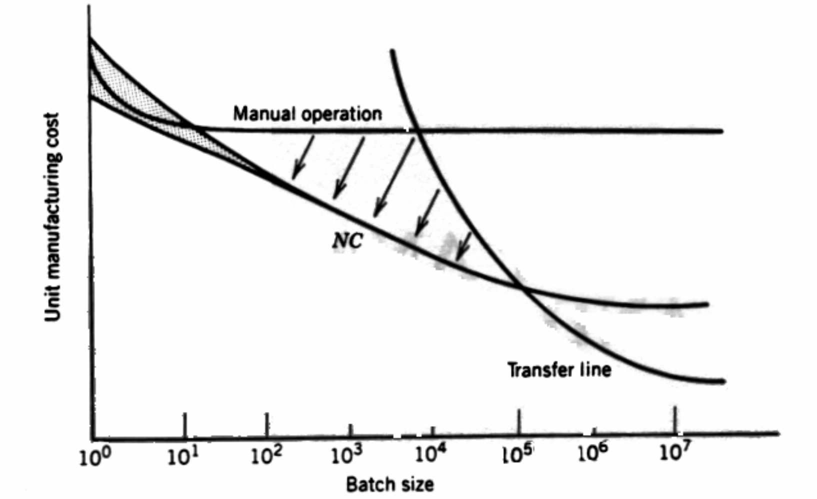


Figure 1-1 General trend of manufacturing cost vs. batch size

issue in manufacturing innovation for a few decades, and numerical control has played a central role in increasing system flexibility. Contemporary industrial robots are programmable machines that can perform different operations by simply modifying stored data, a feature that has evolved from the application of numerical control.

Another origin of today's industrial robots can be found in remote manipulators. A remote manipulator is a device that performs a task at a distance. It can be used in environments that human workers cannot easily or safely access, e.g. for handling radio-active materials, or in some deep sea and space applications. The first *master-slave manipulator* system was developed by 1948. The concept involves an electrically powered mechanical arm installed at the operation site, and a control joystick of geometry similar to that of the mechanical arm (Figure 1-2). The joystick has position transducers at individual joints that measure the motion of the human operator as he moves the tip of the joystick. Thus the operator's motion is transformed into electrical signals, which are transmitted to the mechanical arm and cause the same motion as the one that the human operator performed. The joystick that the operator handles is called the *master* manipulator, while the mechanical arm is called the *slave* manipulator, since its motion is ideally the replica of the operator's commanded motion. A master-slave manipulator has typically six degrees of freedom to allow the gripper to locate an object at an arbitrary position and orientation. Most joints are revolute, and the whole mechanical construction is similar to that of the human arm. This analogy with the human arm results from the need of replicating human motions. Further, this structure allows dexterous motions in a wide range of workspaces, which is desirable for operations in modern manufacturing systems.

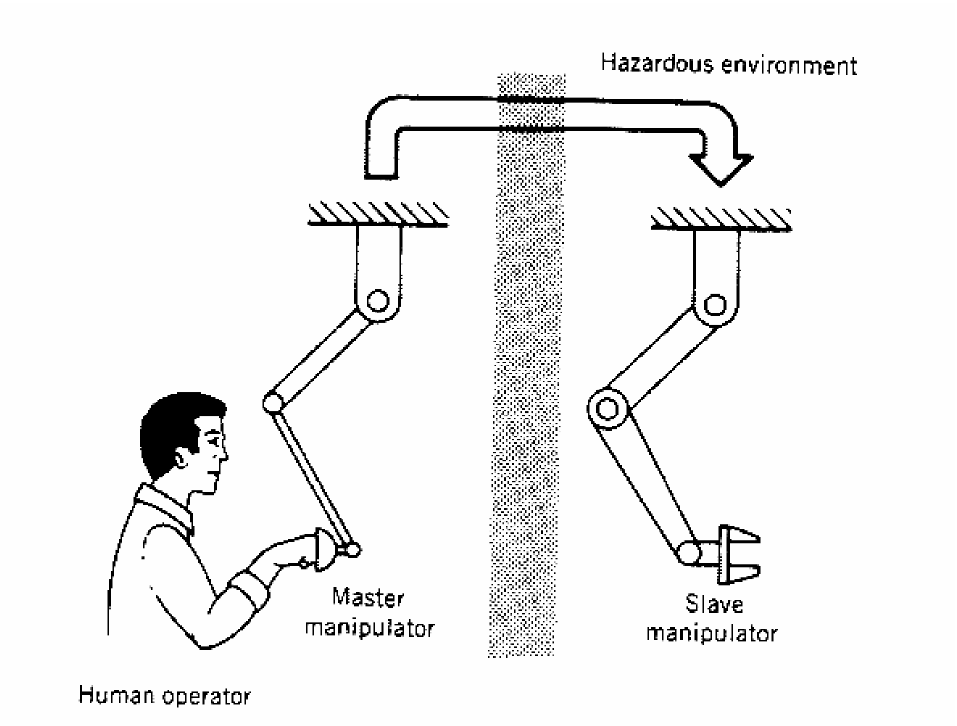


Figure 1-2 Master-slave remote manipulator system

Contemporary industrial robots retain some similarity in geometry with both the human arm and remote manipulators. Further, their basic concepts have evolved from those of numerical control and remote manipulation. Thus a widely accepted definition of today's industrial robot is that of a numerically controlled manipulator, where the human operator and the master manipulator in the figure are replaced by a numerical controller.

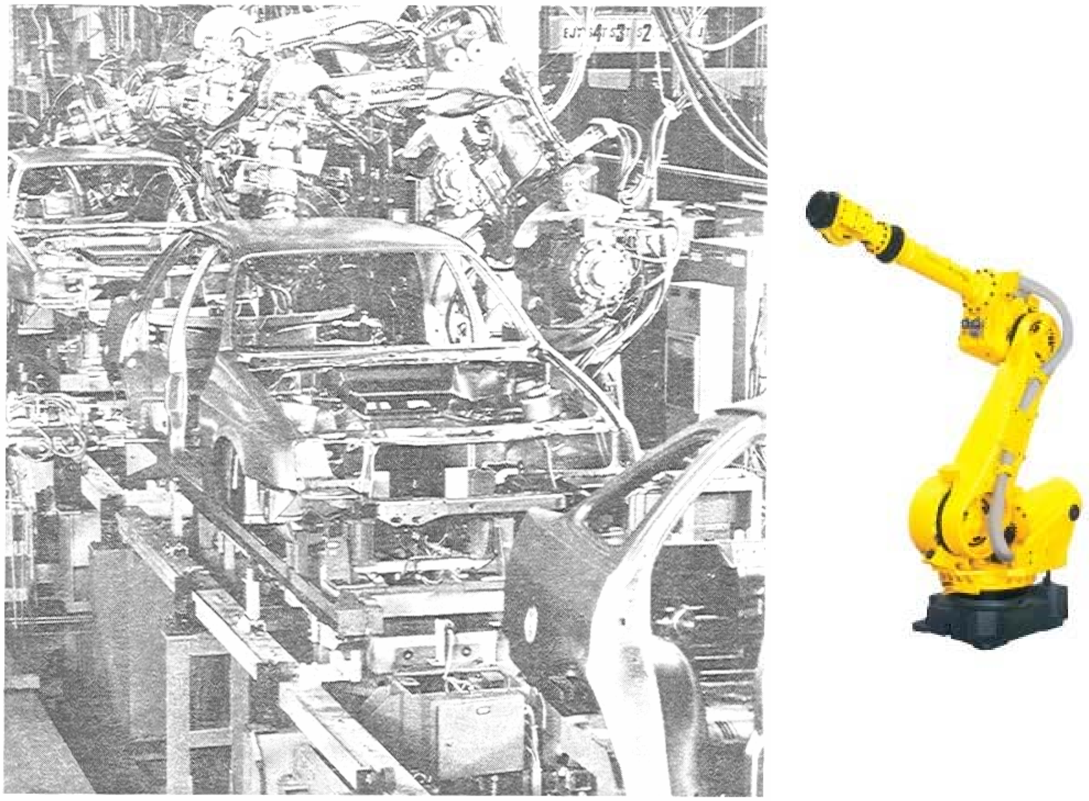


Figure 1-3 White body assembly lines using spot welding robots

1.2 Creation of Robotics

The merge of numerical control and remote manipulation created a new field of engineering, and with it a number of scientific issues in design and control which are substantially different from those of the original technologies have emerged.

Robots are required to have much higher *mobility and dexterity* than traditional machine tools. They must be able to work in a large reachable range, access crowded places, handle a variety of workpieces, and perform flexible tasks. The high mobility and dexterity requirements result in the unique mechanical structure of robots, which parallels the human arm structure. This structure, however, significantly departs from traditional machine design. A robot mechanical structure is basically composed of cantilevered beams, forming a sequence of arm links connected by hinged joints. Such a structure has inherently poor mechanical stiffness and accuracy, hence is not appropriate for the heavy-duty, high-precision applications required of machine tools. Further, it also implies a serial sequence of servoed joints, whose errors accumulate along the linkage. In order to exploit the high mobility and dexterity uniquely featured by the serial linkage, these difficulties must be overcome by advanced design and control techniques.

The serial linkage geometry of manipulator arms is described by complex nonlinear equations. Effective analytical tools are necessary to understand the geometric and kinematic behavior of the manipulator, globally referred to as the manipulator *kinematics.* This represents an important and unique area of robotics research, since research in kinematics and design has traditionally focused upon single-input mechanisms with single actuators moving at constant speeds, while robots are multi-input spatial mechanisms which require more sophisticated analytical tools.

The *dynamic* behavior of robot manipulators is also complex, since the dynamics of multi-input spatial linkages are highly coupled and nonlinear. The motion of each joint is significantly affected by the motions of all the other joints. The inertial load imposed at each joint varies widely depending on the configuration of the manipulator arm. Coriolis and centrifugal effects are prominent when the manipulator arm moves at high speeds. The kinematic and dynamic complexities create unique control problems that are not adequately handled by standard linear control techniques, and thus make effective *control system design* a critical issue in robotics.

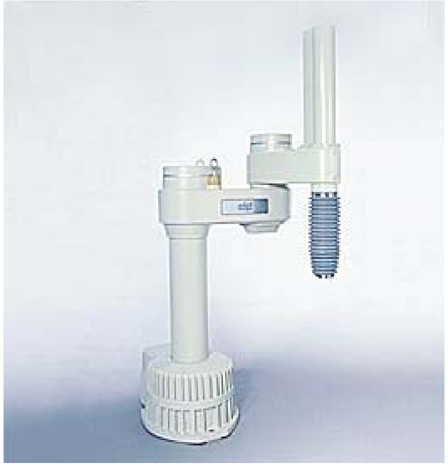


Figure 1-4 Adept Direct-Drive robot

Finally, robots are required to *interact* much more heavily with peripheral devices than traditional *numerically-controlled* machine tools. Machine tools are essentially self-contained systems that handle workpieces in well-defined locations. By contrast, the environment in which robots are used is often poorly structured, and effective means must be developed to identify the locations of the workpieces as well as to communicate to peripheral devices and other machines in a coordinated fashion. Robots are also critically different from master-slave manipulators, in that they are *autonomous* systems. Master-slave manipulators are essentially manually controlled

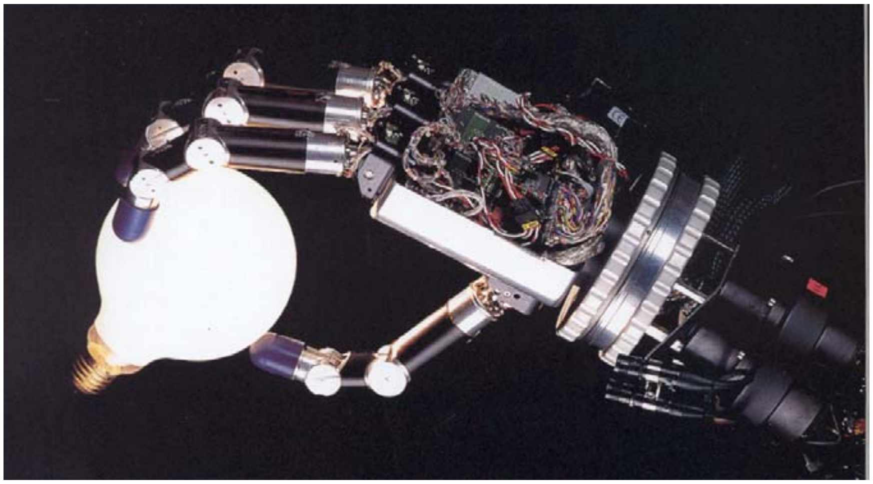


Figure 1-5 Dexterous fingers

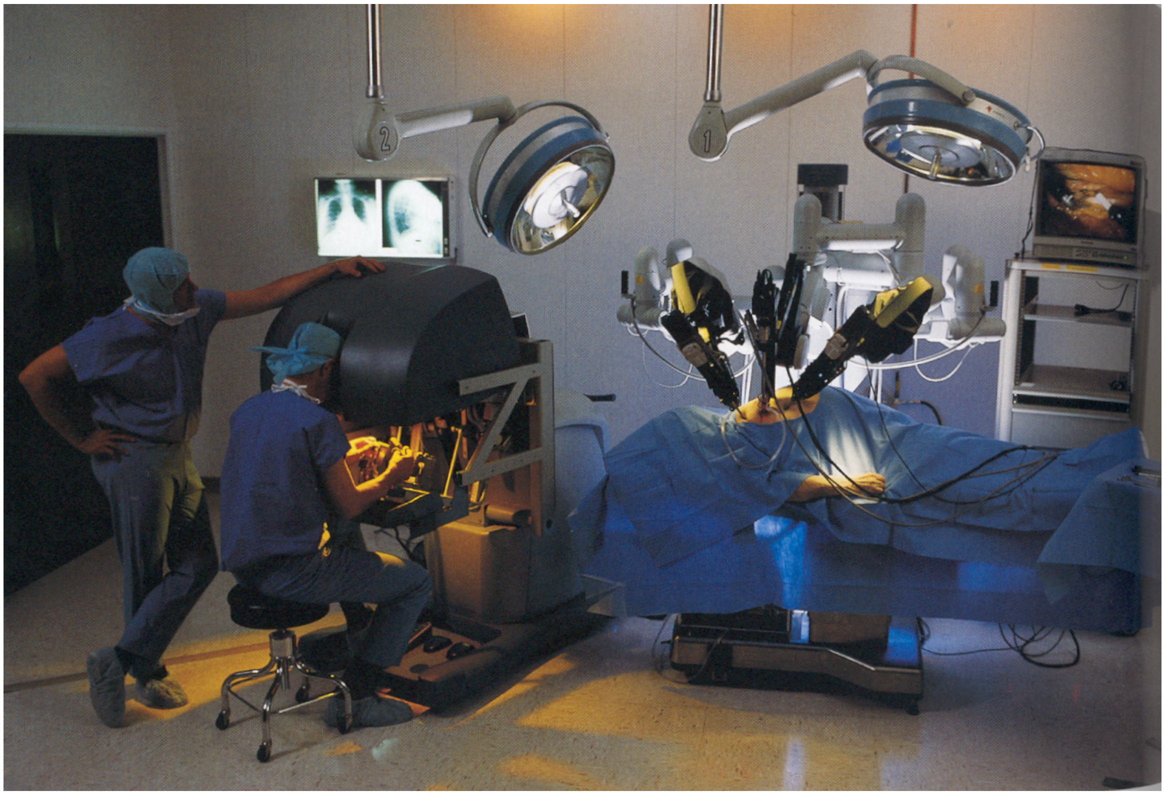


Figure 1-6 Medical robots for minimally invasive surgery

systems, where the human operator takes the decisions and applies control actions. The operator interprets a given task, finds an appropriate strategy to accomplish the task, and plans the procedure of operations. He/she devises an effective way of achieving the goal on the basis of his/her experience and knowledge about the task. His/her decisions are then transferred to the slave manipulator through the joystick. The resultant motion of the slave manipulator is monitored by the operator, and necessary adjustments or modifications of control actions are provided when the resultant motion is not adequate, or when unexpected events occur during the operation. The human operator is, therefore, an essential part of the control loop. When the operator is eliminated from the control system, all the planning and control commands must be generated by the machine itself. The detailed procedure of operations must be set up in advance, and each step of motion command must be generated and coded in an appropriate form so that the robot can interpret it and execute it accurately. Effective means to store the commands and manage the data file are also needed . Thus, *programming and command generation* are critical issues in robotics. In addition, the robot must be able to fully monitor its own motion. In order to adapt to disturbances and unpredictable changes in the work environment, the robot needs a variety of *sensors,* so as to obtain information both about the environment (using *external* sensors, such as cameras or touch sensors) and about itself (using *internal* sensors, such as joint encoders or joint torque sensors). Effective sensor-based strategies that incorporate this information require advanced control algorithms. But they also imply a detailed understanding of the task.

1.3. Manipulation and Dexterity

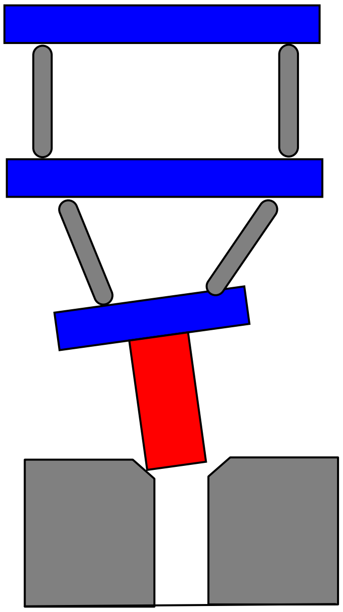


Figure 1-7 Remote-center compliance hand

Contemporary industrial needs drive the applications of robots to ever more advanced tasks. Robots are required to perform highly skilled jobs with minimum human assistance or intervention. To extend the applications and abilities of robots, it becomes important to develop a sound understanding of the *tasks* themselves.

In order to devise appropriate arm mechanisms and to develop effective control algorithms, we need to precisely understand how a given task should be accomplished and what sort of motions the robot should be able to achieve. To perform an assembly operation, for example, we need to know how to guide the assembly part to the desired location, mate it with another part, and secure it in an appropriate way. In a grinding operation, the robot must properly position the grinding wheel while accommodating the contact force. We need to analyze the grinding process itself in order to generate appropriate force and motion commands.

A detailed understanding of the underlying principles and "know-how" involved in the task must be developed in order to use industrial robots effectively, while there is no such need for making control strategies *explicit* when the assembly and grinding operations are performed by a human worker. Human beings perform sophisticated manipulation tasks without being aware of the control principles involved. We have *trained* ourselves to be capable of skilled jobs, but in general we do not know what the acquired skills are exactly about. A sound and explicit understanding of manipulation operations, however, is essential for the long-term progress of robotics. This scientific aspect of manipulation has never been studied systematically before, and represents an emerging and important part of robotics research.

1.4 Locomotion and Navigation

Robotics has found a number of important application areas in broad fields beyond manufacturing automation. These range from space and under-water exploration, hazardous waste disposal, and environment monitoring to robotic surgery, rehabilitation, home robotics, and entertainment. Many of these applications entail some locomotive functionality so that the robot can freely move around in an unstructured environment. Most industrial robots sit on a manufacturing floor and perform tasks in a structured environment. In contrast, those robots for non-manufacturing applications must be able to move around on their own. See Figure 1-8.

Locomotion and navigation are increasingly important, as robots find challenging applications in the field. This opened up new research and development areas in robotics. Novel mechanisms are needed to allow robots to move through crowded areas, rough terrain, narrow channels, and even staircases. Various types of legged robots have been studied, since, unlike standard wheels, legs can negotiate with uneven floors and rough terrain. Among others, biped robots have been studied most extensively, resulting in the development of humanoids, as shown in Figure 1-9. Combining leg mechanisms with wheels has accomplished superior performance in both flexibility and efficiency. The Mars Rover prototype shown below has a rocker-buggy mechanism combined with advanced wheel drives in order to adapt itself to diverse terrain conditions. See Figure 1-10.

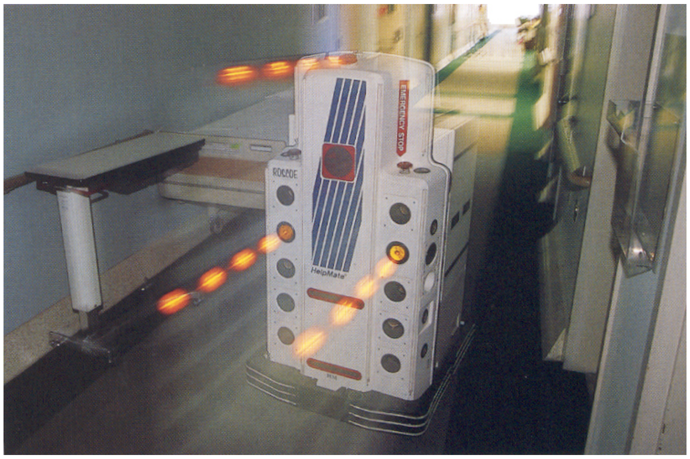


Figure 1-8 Automatically guided vehicle for meal delivery in hospitals

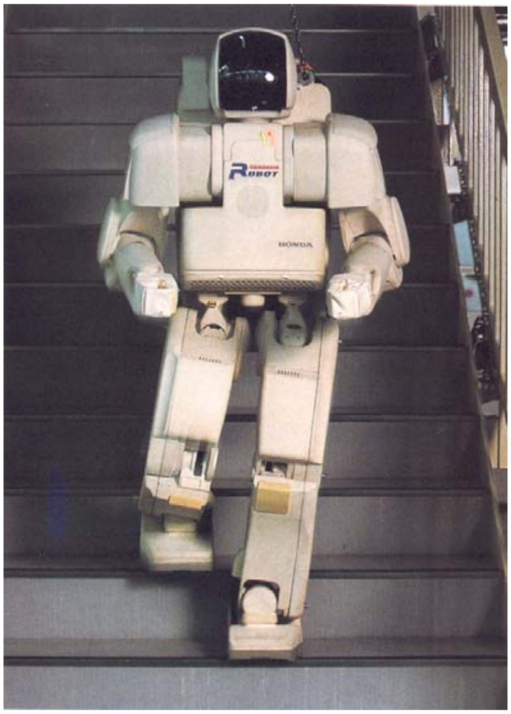


Figure 1-9 Honda’s P3 humanoid robot

Navigation is another critical functionality needed for mobile robots, in particular, for unstructured environment. Those robots are equipped with range sensors and vision system, and are capable of interpreting the data to locate themselves. Often the robot has a map of the environment, and uses it for estimating the location. Furthermore, based on the real-time data obtained in the field, the robot is capable of updating and augmenting the map, which is incomplete and uncertain in unstructured environment. As depicted in Figure 1-10, location estimation and map building are simultaneously executed in the advanced navigation system. Such Simultaneous Localization And Mapping (SLAM) is exactly what we human do in our daily life, and is an important functionality of intelligent robots.

The goal of robotics is thus two-fold: to extend our understanding about manipulation, locomotion, and other robotic behaviors and to develop engineering methodologies to actually perform desired tasks. The goal of this book is to provide entry-level readers and experienced engineers with fundamentals of understanding robotic tasks and intelligent behaviors as well as with enabling technologies needed for building and controlling robotic systems.

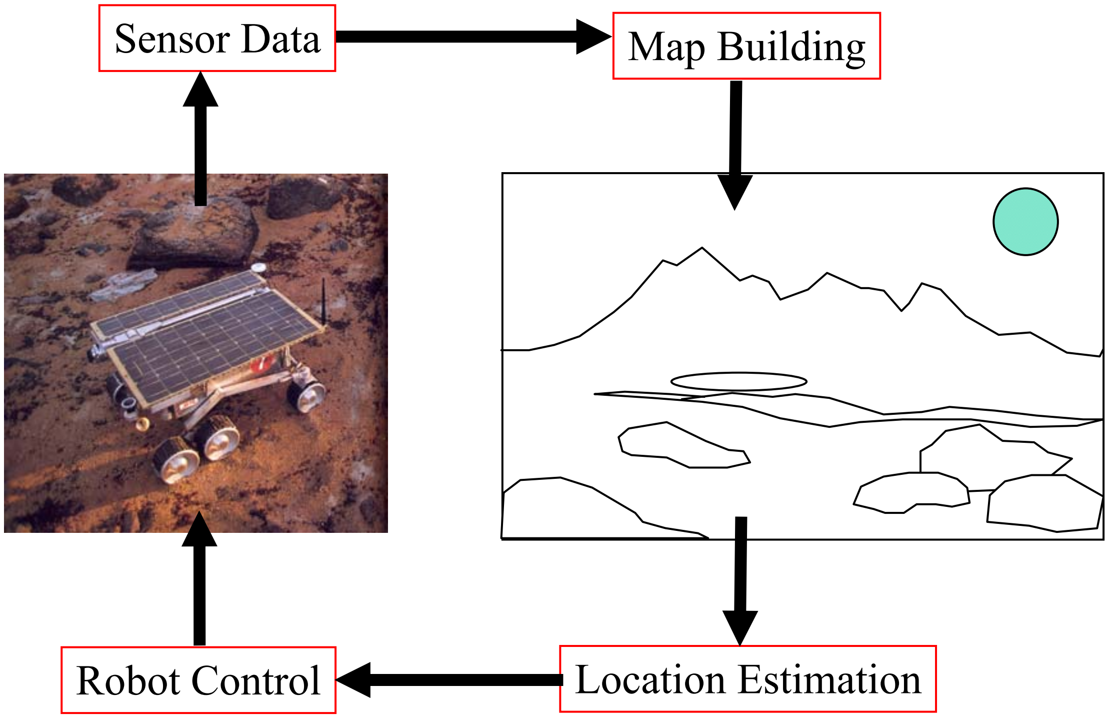


Figure 1-10 JPL’s planetary exploration robot: an early version of the Mars Rover

Chapter 2
Actuators and Drive Systems

Actuators are one of the key components contained in a robotic system. A robot has many degrees of freedom, each of which is a servoed joint generating desired motion. We begin with basic actuator characteristics and drive amplifiers to understand behavior of servoed joints.

Most of today's robotic systems are powered by electric servomotors. Therefore, we focus on electromechanical actuators.

2.1 DC Motors

Figure 2-1 illustrates the construction of a DC servomotor, consisting of a stator, a rotor, and a commutation mechanism. The stator consists of permanent magnets, creating a magnetic field in the air gap between the rotor and the stator. The rotor has several windings arranged symmetrically around the motor shaft. An electric current applied to the motor is delivered to individual windings through the brush-commutation mechanism, as shown in the figure. As the rotor rotates the polarity of the current flowing to the individual windings is altered. This allows the rotor to rotate continually.

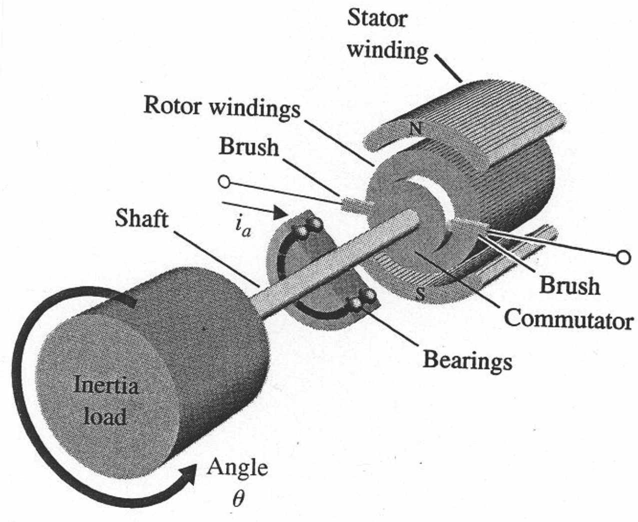


Figure 2.1.1 Construction of DC motor

Let τ_m be the torque created at the air gap, and *i* the current flowing to the rotor windings. The torque is in general proportional to the current, and is given by

$$\tau_m = K_t \cdot i \tag{2.1.1}$$

where the proportionality constant K_t is called the ***torque constant***, one of the key parameters describing the characteristics of a DC motor. The torque constant is determined by the strength of the magnetic field, the number of turns of the windings, the effective area of the air gap, the radius of the rotor, and other parameters associated with materials properties.

In an attempt to derive other characteristics of a DC motor, let us first consider an idealized energy transducer having no power loss in converting electric power into mechanical power. Let *E* be the voltage applied to the idealized transducer. The electric power is then given by $E \cdot i$, which must be equivalent to the mechanical power:

$$P_{in} = E \cdot i = \tau_m \cdot \omega_m \tag{2.1.2}$$

where ω_m is the angular velocity of the motor rotor. Substituting eq.(1) into eq.(2) and dividing both sides by *i* yield the second fundamental relationship of a DC motor:

$$E = K_t \omega_m \tag{2.1.3}$$

The above expression dictates that the voltage across the idealized power transducer is proportional to the angular velocity and that the proportionality constant is the same as the torque constant given by eq.(1). This voltage *E* is called the back emf (electro-motive force) generated at the air gap, and the proportionality constant is often called the back emf constant.

Note that, based on eq.(1), the unit of the torque constant is *Nm/A* in the metric system, whereas the one of the back emf constant is *V/rad/s* based on eq.(2).

Exercise 2.1 Show that the two units, *Nm/A* and *V/rad/s*, are identical.

The actual DC motor is not a loss-less transducer, having resistance at the rotor windings and the commutation mechanism. Furthermore, windings may exhibit some inductance, which stores energy. Figure 2.1.2 shows the schematic of the electric circuit, including the windings resistance *R* and inductance *L*. From the figure,

$$u = R \cdot i + L\frac{di}{dt} + E \tag{2.1.4}$$

where *u* is the voltage applied to the armature of the motor.

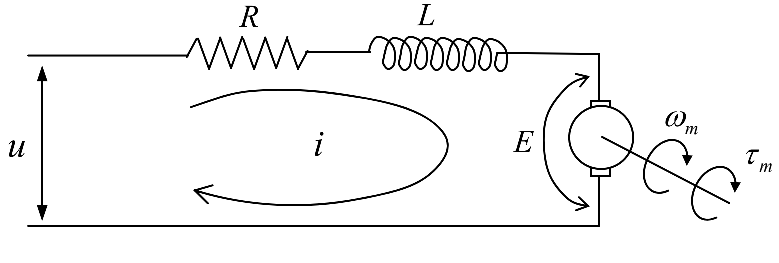


Figure 2.1.2 Electric circuit of armature

Combining eqs.(1), (3) and (4), we can obtain the actual relationship among the applied voltage *u,* the rotor angular velocity ω_m, and the motor torque τ_m.

$$\frac{K_t}{R}u = \tau_m + T_e\frac{d\tau_m}{dt} + \frac{K_t^2}{R}\omega_m \tag{2.1.5}$$

where time constant $T_e = \dfrac{L}{R}$, called the motor reactance, is often negligibly small. Neglecting this second term, the above equation reduces to an algebraic relationship:

$$\tau_m = \frac{K_t}{R} u - \frac{K_t^{\,2}}{R} \omega_m \tag{2.1.6}$$

This is called the torque-speed characteristic. Note that the motor torque increases in proportion to the applied voltage, but the net torque reduces as the angular velocity increases. Figure 2.1.3 illustrates the torque-speed characteristics. The negative slope of the straight lines, $-K_t^{\,2}/R$, implies that the voltage-controlled DC motor has an inherent damping in its mechanical behavior.

The power dissipated in the DC motor is given by

$$P_{dis} = R \cdot i^2 = \frac{R}{K_t^{\,2}} \tau_m^{\,2} \tag{2.1.7}$$

from eq.(1). Taking the square root of both sides yields

$$\sqrt{P_{dis}} = \frac{\tau_m}{K_m}, \quad K_m = \frac{K_t}{\sqrt{R}} \tag{2.1.8}$$

where the parameter K_m is called the motor constant. The motor constant represents how effectively electric power is converted to torque. The larger the motor constant becomes, the larger the output torque is generated with less power dissipation. A DC motor with more powerful magnets, thicker winding wires, and a larger rotor diameter has a larger motor constant. A motor with a larger motor constant, however, has a larger damping, as the negative slope of the torque-speed characteristics becomes steeper, as illustrated in Figure 2.1.3.

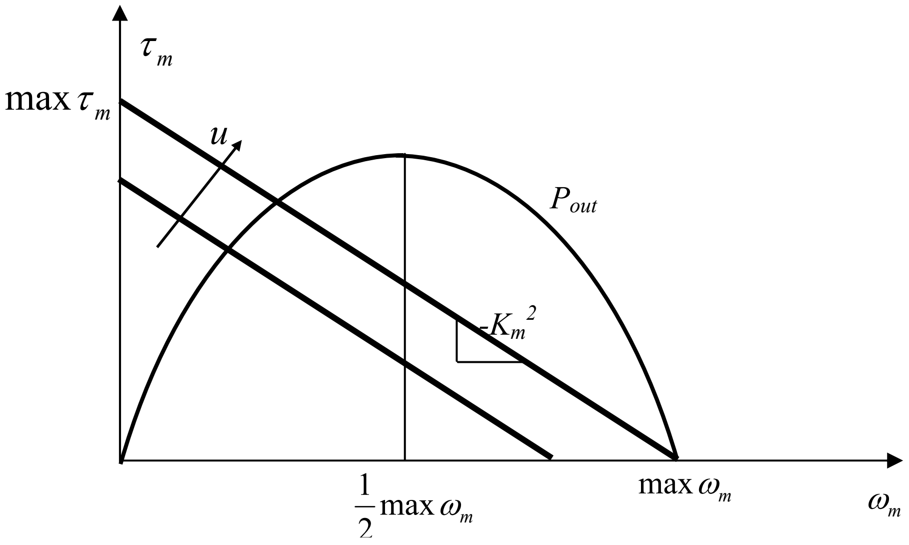


Figure 2.1.3 Torque-speed characteristics and output power

Taking into account the internal power dissipation, the net output power of the DC motor is given by

$$P_{out} = \tau_m \cdot \omega_m = (\frac{K_t}{R} u - K_m{}^2 \omega_m)\, \omega_m \qquad (2.1.9)$$

This net output power is a parabolic function of the angular velocity, as illustrated in Figure 2.1.3. It should be noted that the net output power becomes maximum in the middle point of the velocity axis, i.e. 50 % of the maximum angular velocity for a given armature voltage *u.* This implies that the motor is operated most effectively at 50 % of the maximum speed. As the speed departs from this middle point, the net output power decreases, and it vanishes at the zero speed as well as at the maximum speed. Therefore, it is important to select the motor and gearing combination so that the maximum of power transfer be achieved.

2.2 Dynamics of Single-Axis Drive Systems

DC motors and other types of actuators are used to drive individual axes of a robotic system. Figure 2.2.1 shows a schematic diagram of a single-axis drive system consisting of a DC motor, a gear head, and arm links[1]. An electric motor, such as a DC motor, produces a relatively small torque and rotates at a high speed, whereas a robotic joint axis in general rotates slowly, and needs a high torque to bear the load. In other words, the impedance of the actuator:

$$Z_m = \frac{torque}{angular\ velocity} = \frac{\tau_m}{\omega_m} \qquad (2.2.1)$$

is much smaller than that of the load.

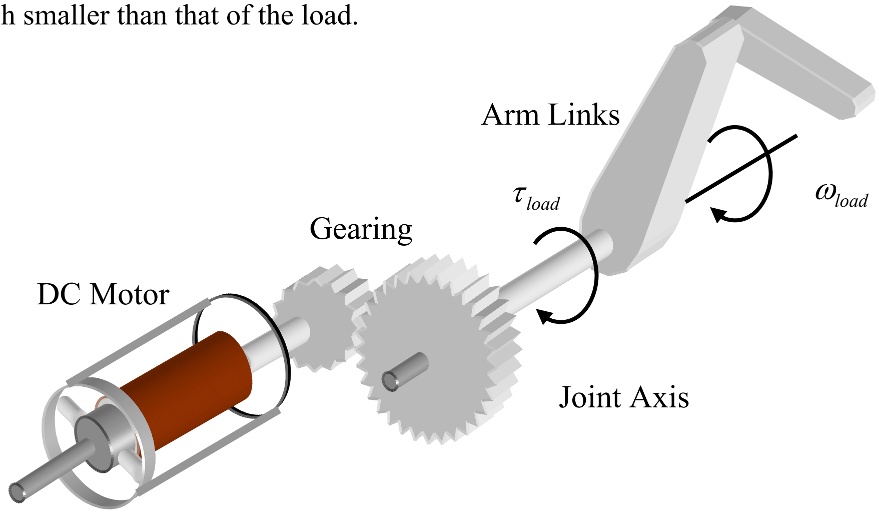


Figure 2.2.1 Joint axis drive system

[1] Although a robotic system has multiple axes driven by multiple actuators having dynamic interactions, we consider behavior of an independent single axis in this section, assuming that all the other axes are fixed.

To fill the gap we need a gear reducer, as shown in Figure 2.2.1. Let $r > 1$ be a gear reduction ratio (If d_1 and d_2 are diameters of the two gears, the gear reduction ratio is $r = d_2 / d_1$). The torque and angular velocity are changed to:

$$\tau_{load} = r \cdot \tau_m, \quad \omega_{load} = \frac{1}{r}\omega_m \tag{2.2.2}$$

where τ_{load} and ω_{load} are the torque and angular velocity at the joint axis, as shown in the figure. Note that the gear reducer of gear ratio *r* increases the impedance r^2 times larger than that of the motor axis Z_m:

$$Z_{load} = r^2 \cdot Z_m \tag{2.2.3}$$

Let I_m be the inertia of the motor rotor. From the free body diagram of the motor rotor,

$$I_m \dot{\omega}_m = \tau_m - \frac{1}{r}\tau_{load} \tag{2.2.4}$$

where $-\frac{1}{r}\tau_{load}$ is the torque acting on the motor shaft from the joint axis through the gears, and $\dot{\omega}_m$ is the time rate of change of angular velocity, i.e. the angular acceleration. Let I_l be the inertia of the arm link about the joint axis, and *b* the damping coefficient of the bearings supporting the joint axis. Considering the free body diagram of the arm link and joint axis yields

$$I_l \dot{\omega}_{load} = \tau_{load} - b\omega_{load} \tag{2.2.5}$$

Eliminating τ_{load} from the above two equations and using eq.(2.1.6) and (2.2.2) yields

$$I\dot{\omega}_{load} + B\omega_{load} = k \cdot u \tag{2.2.6}$$

where *I, B, k* are the effective inertia, damping, and input gain reflected to the joint axis:

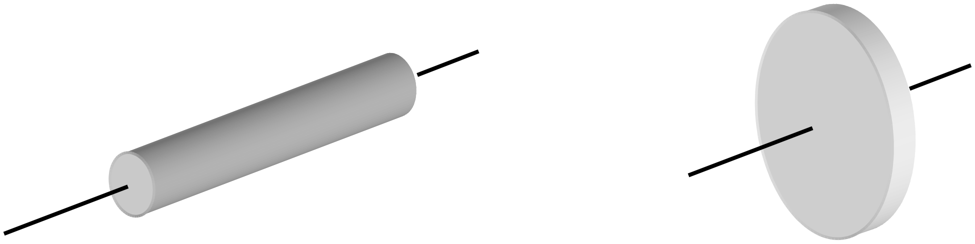
$$I = I_l + r^2 I_m \tag{2.2.7}$$

$$B = b + r^2 K_m{}^2 \tag{2.2.8}$$

$$k = r\frac{K_t}{R} \tag{2.2.9}$$

Note that the effective inertia of the motor rotor is r^2 times larger than the original value I_m when reflected to the joint axis. Likewise, the motor constant becomes r^2 times larger when reflected to the joint axis. The gear ratio of a robotic system is typically 20 ~ 100, which means that the effective inertia and damping becomes 400 ~ 10,000 times larger than those of the motor itself.

For fast dynamic response, the inertia of the motor rotor must be small. This is a crucial requirement as the gear ratio gets larger, like robotics applications. There are two ways of reducing the rotor inertia in motor design. One is to reduce the diameter and make the rotor longer, as shown in Figure 2.2.2-(a). The other is to make the motor rotor very thin, like a pancake, as shown in Figure 2.2.2-(b).



(a) Long and slender (b) Pancake

Figure 2.2.2 Two ways of reducing the motor rotor inertia

Most robots use the long and slender motors as Figure (a), and some heavy-duty robots use the pancake type motor. Figure 2.2.3 shows a pancake motor by Mavilor Motors, Inc.

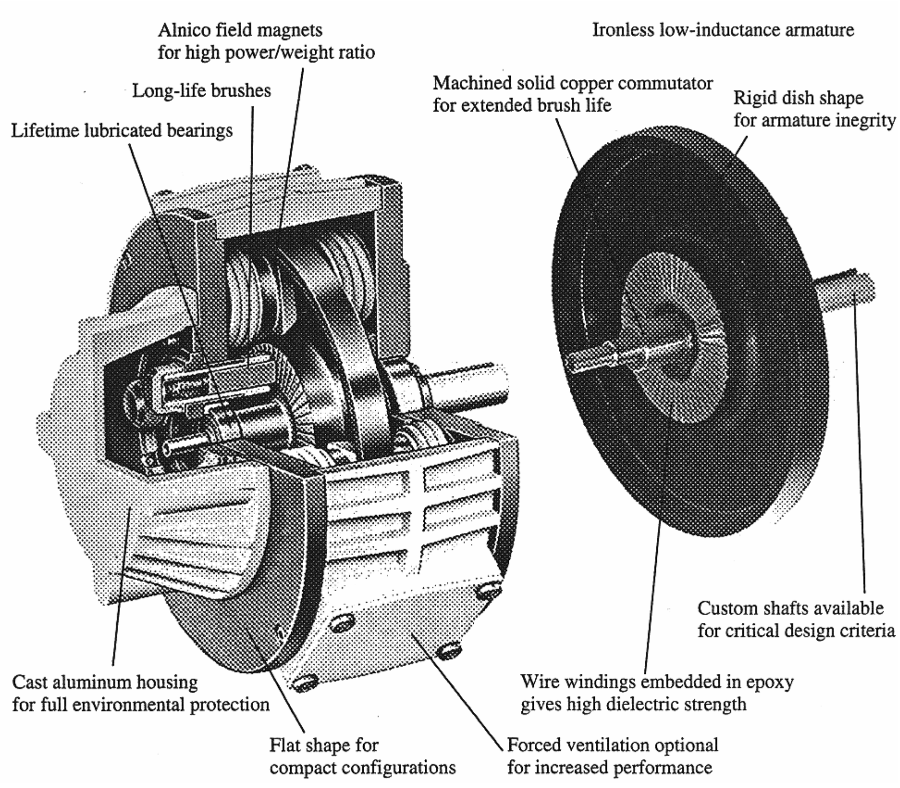

Alnico field magnets
for high power/weight ratio
Ironless low-inductance armature
Long-life brushes
Machined solid copper commutator
for extended brush life
Rigid dish shape
for armature inegrity
Lifetime lubricated bearings
Custom shafts available
for critical design criteria
Cast aluminum housing
for full environmental protection
Wire windings embedded in epoxy
gives high dielectric strength
Flat shape for
compact configurations
Forced ventilation optional
for increased performance

Figure 2.2.3 Pancake DC motor

Exercise 2-2 Assuming that the angular velocity of a joint axis is approximately zero, obtain the optimal gear ratio *r* in eq.(7) that maximizes the acceleration of the joint axis.

2.3 Power Electronics

Performance of servomotors used for robotics applications highly depends on electric power amplifiers and control electronics, broadly termed power electronics. Power electronics has shown rapid progress in the last two decades, as semiconductors became faster, more powerful, and more efficient. In this section we will briefly summarize power electronics relevant to robotic system development.

2.3.1 Pulse width modulation (PWM)

In many robotics applications, actuators must be controlled precisely so that desired motions of arms and legs may be attained. This requires a power amplifier to drive a desired level of voltage (or current indirectly) to the motor armature, as discussed in the previous section. Use of a linear amplifier (like an operational amplifier), however, is power-inefficient and impractical, since it entails a large amount of power loss. Consider a simple circuit consisting of a single transistor for controlling the armature voltage, as shown in Figure 2.3.1. Let *V* be the supply voltage connected to one end of the motor armature. The other end of the armature is connected to the collector of the transistor. As the base voltage varies the emitter-collector voltage varies, and thereby the voltage drop across the motor armature, denoted *u* in the figure, varies accordingly. Let *i* be the collector current flowing through the transistor. Then the power loss that is dissipated at the transistor is given by

$$P_{loss} = (V-u)\cdot i = \frac{1}{R}(V-u)\cdot u \tag{2.3.1}$$

where *R* is the armature resistance. Figure 2.3.2 plots the internal power loss at the transistor against the armature voltage. The power loss becomes the largest in the middle, where half the supply voltage *V/2* acts on the armature. This large heat loss is not only wasteful but also harmful, burning the transistor in the worst case scenario. Therefore, this type of linear power amplifier is seldom used except for driving very small motors.

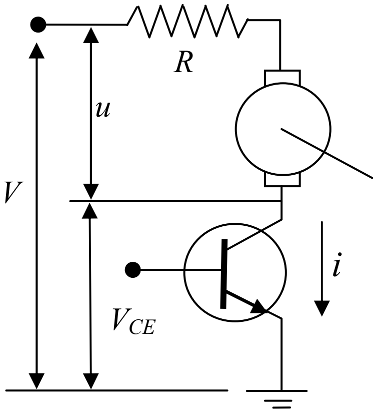


Figure 2.3.1 Analogue power amplifier for driving the armature voltage

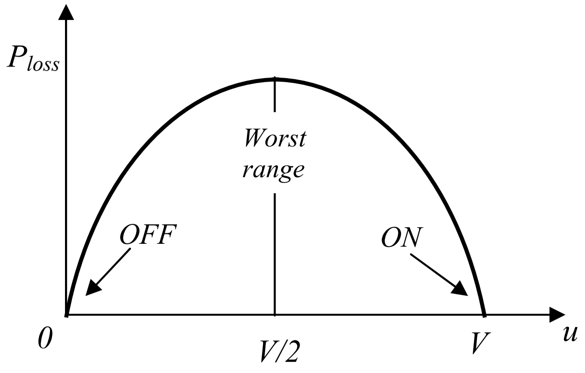


Figure 2.3.2 Power loss at the transistor vs. the armature voltage.

An alternative is to control the voltage via ON-OFF switching. Pulse Width Modulation, or PWM for short, is the most commonly used method for varying the average voltage to the motor. In Figure 2.3.2 it is clear that the heat loss is zero when the armature voltage is either *0* or *V*. This means that the transistor is completely shutting down the current (OFF) or completely admitting the current (ON). For all armature voltages other than these complete ON-OFF states, some fraction of power is dissipated in the transistor. Pulse Width Modulation (PWM) is a technique to control an effective armature voltage by using the ON-OFF switching alone. It varies the ratio of time length of the complete ON state to the complete OFF state. Figure 2.3.3 illustrates PWM signals. A single cycle of ON and OFF states is called the PWM period, whereas the percentage of the ON state in a single period is called *duty rate*. The first PWM signal is of 60% duty, and the second one is 25 %. If the supply voltage is *V=10 volts,* the average voltage is *6 volts* and *2.5 volts*, respectively.

The PWM period is set to be much shorter than the time constant associated with the mechanical motion. The PWM frequency, that is the reciprocal to the PWM period, is usually 2 ~ 20 kHz, whereas the bandwidth of a motion control system is at most 100 Hz. Therefore, the discrete switching does not influence the mechanical motion in most cases.

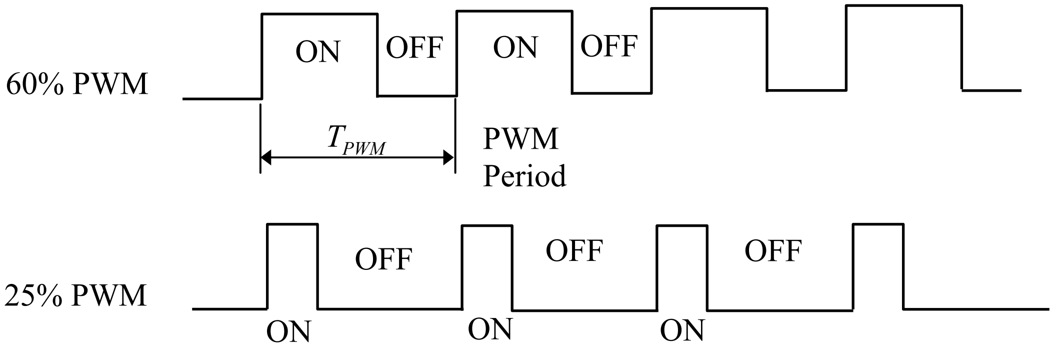


Figure 2.3.3 Pulse width modulation

As modeled in eq.(2.1.4), the actual rotor windings have some inductance *L*. If the electric time constant T_e is much larger than the PWM period, the actual current flowing to the motor armature is a smooth curve, as illustrated in Figure 2.3.4-(a). In other words, the inductance works as a low-pass filter, filtering out the sharp ON-OFF profile of the input voltage. In contrast, if the electric time constant is too small, compared to the PWM period, the current profile becomes zigzag, following the rectangular voltage profile, as shown in Figure 2.3.4-(b). As a result, unwanted high frequency vibrations are generated at the motor rotor. This happens for some types of pancake motors with low inductance and low rotor inertia.

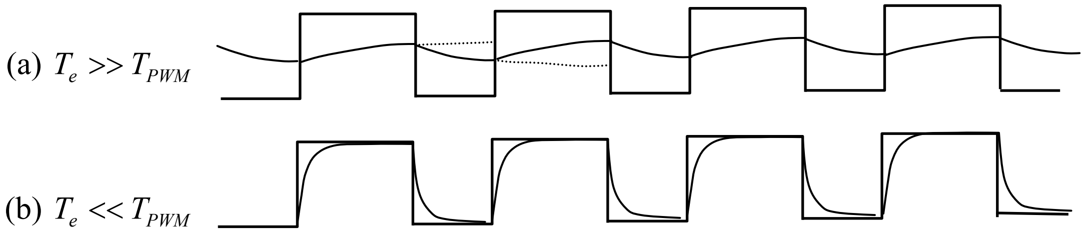


Figure 2.3.4 Current to the motor is smoothed due to inductance

2.3.2 PWM switching characteristics

As the PWM frequency increases, the current driven to the motor becomes smoother, and the nonlinearity due to discrete switching disappears. Furthermore, high PWM frequencies cause no audible noise of switching. The noise disappears as the switching frequency becomes higher than the human audible range, say *15 kHz*. Therefore, a higher PWM frequency is in general desirable. However, it causes a few adverse effects. As the PWM frequency increases:

- The heat loss increases and the transistor may over-heat,
- Harmful large voltage spikes and noise are generated, and
- Radio frequency interference and electromagnetic interference become prominent.

The first adverse effect is the most critical one, which limits the capacity of a PWM amplifier. Although no power loss occurs at the switching transistor when it is completely ON or OFF, a significant amount of loss is caused during transition. As the transistor state is switched from OFF to ON or vise versa, the transistor in Figure 2.3.1 goes through intermediate states, which entail heat loss, as shown in Figure 2.3.2. Since it takes some finite time for a semiconductor to make a transition, every time it is switched, a certain amount of power is dissipated. As the PWM frequency increases, more power loss and, more importantly, more heat generation occur. Figure 2.3.5 illustrates the turn-on and turn-off transitions of a switching transistor. When turned on, the collector current I_c increases and the voltage V_{ce} decreases. The product of these two values provides the switching power loss as shown by broken lines in the figure. Note that turn-off takes a longer time, hence it causes more heat loss.

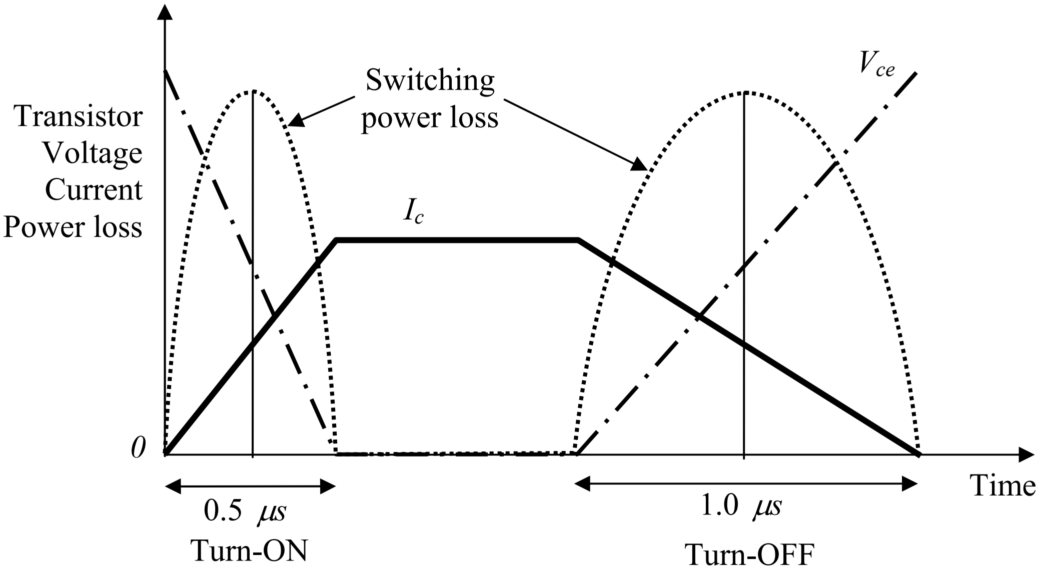


Figure 2.3.5 Transient responses of transistor current and voltage and associated power loss during turn-on and turn-off state transitions

From Figure 2.3.5 it is clear that a switching transistor having fast turn-on and turn-off characteristics is desirable, since it causes less power loss and heat generation. Power MOSFETs (Metal-Oxide-Semiconductor Field-Effect Transistors) have very fast switching characteristics, enabling 15 ~ 100 kHz of switching frequencies. For relatively small motors, MOSFETs are widely used in industry due to their fast switching characteristics. For larger motors, IGBTs (Insulated Gate Bipolar Transistor) are the rational choice because of their larger capacity and relatively fast response.

As the switching speed increases, the heat loss becomes smaller. However, fast switching causes other problems. Consider eq.(2.1.4) again, the dynamic equation of the armature:

$$u = R \cdot i + L\frac{di}{dt} + E \tag{2.1.4}$$

High speed switching means that the time derivative of current *i* is large. This generates a large inductance-induced kickback voltage $L\frac{di}{dt}$ that often damages switching semiconductors. As illustrated in Figure 2.3.6-(a), a large spike is induced when turning on the semiconductor. To get rid of this problem a free-wheeling-diode (FWD) is inserted across the motor armature, as shown in Figure 2.3.6-(b). As the voltage across the armature exceeds a threshold level, FWD kicks in to bypass the current so that the voltage may be clamped, as shown in figure (c).

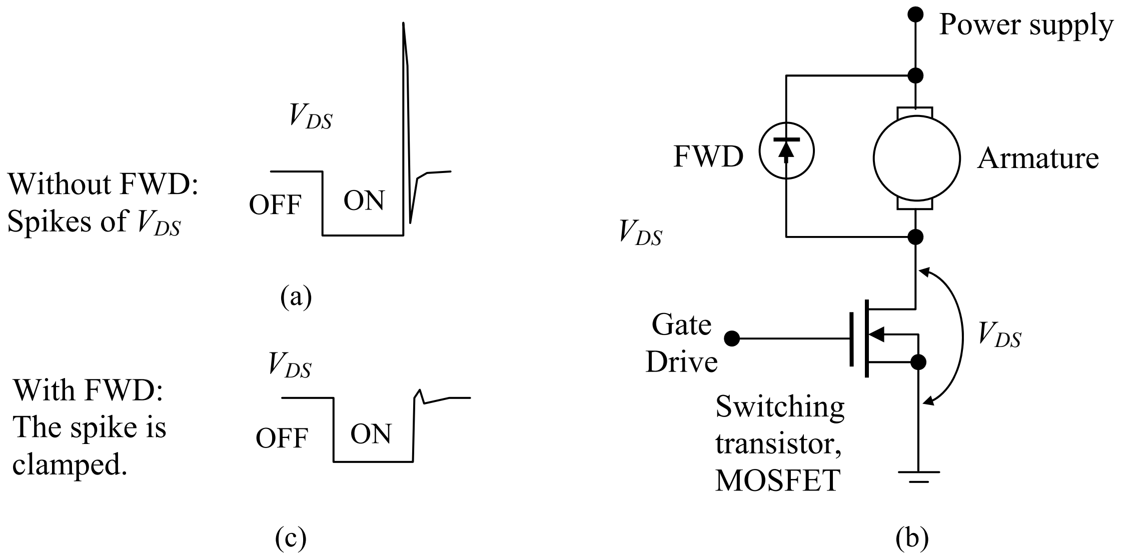
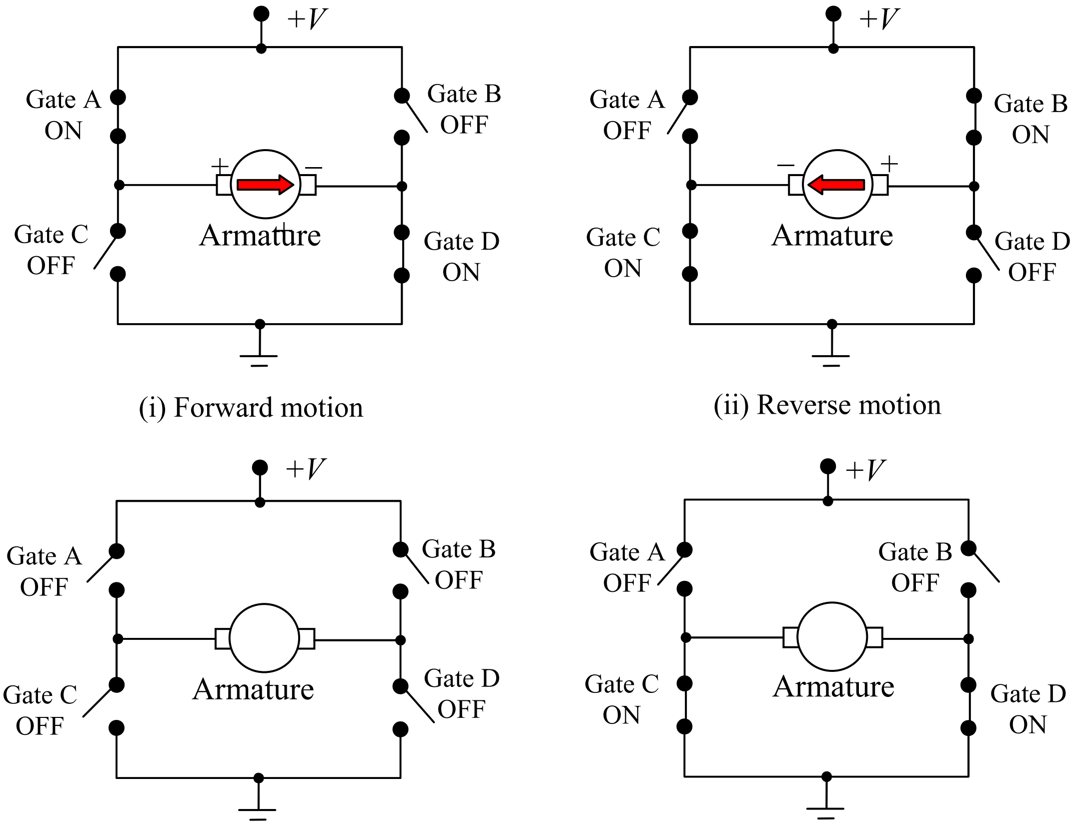


Figure 2.3.6 Voltage spike induced by inductance (a), free-wheeling diode (b), and the clamped spike with FWD (c)

High speed PWM switching also generates Electromagnetic Interference (EMI), particularly when the wires between the PWM amplifier and the motor get longer. Furthermore, high speed PWM switching may incur Radio-Frequency Interference (RFI). Since the PWM waveforms are square, significant RFI can be generated. Whenever PWM switching edges are faster than 10 μs, RFI is induced to some extent. An effective method for reducing EMI and RFI is to put the PWM amplifier inside the motor body. This motor architecture, called Integrated Motor or Smart Motor, allows confining EMI and RFI within the motor body by minimizing the wire length between the motor armature and the power transistors.

2.3.3 The H-bridge and bipolar PWM amplifiers

In most robotics applications, bi-directional control of motor speed is necessary. This requires a PWM amplifier to be bipolar, allowing for both forward and backward rotations. The architecture described in the previous section needs to be extended to meet this bipolar requirement. The H-Bridge architecture is commonly used for bipolar PWM amplifiers. As shown in Figure 2.3.7, the H-Bridge architecture resembles the letter H in the arrangement of switching transistors around the motor armature. Switching transistors A and B are pulled up to the supply voltage *V,* whereas transistors C and D are connected to ground. Combinations of these four switching transistors provide a variety of operations. In figure (i), gates A and D are ON, and B and C are OFF. This gate combination delivers a current to the armature in the forward direction. When the gate states are reversed, as shown in figure (ii), the direction of current is reversed. Furthermore, the motor coasts off when all the gates are turned OFF, since the armature is totally isolated or disconnected as shown in figure (iii). On the other hand, the armature windings are shortened, when both gates C and D are turned ON and A and B are turned OFF. See figure (iv). This shortened circuit provides a “braking” effect, when the motor rotor is rotating.



(i) Forward motion

(ii) Reverse motion

(iii) The motor armature coasts off

(iv) The motor windings are shortened causing a braking effect.

Figure 2.3.7 H-bridge and four quadrant control

It should be noted that there is a fundamental danger in the H-bridge circuit. A direct short circuit can occur if the top and bottom switches connected to the same armature terminal are turned on at the same time. A catastrophic failure results when one of the switching transistors on the same vertical line in Figure 2.3.7 fails to turn off before the other turns on. Most of H-bridge power stages commercially available have several protection mechanisms to prevent the direct short circuit.

2.4 Robot Controls and PWM Amplifiers of the 2.12 Laboratory

DC motors and PWM amplifiers, the two most important components involved in a robot power train, have been described. Now we are ready to introduce the specific drive system and controls to be used for building robots for the design project.

This term we will use a wireless real-time controller, WEECS (Wireless Ethernet Embedded Control System) by Quanser (www.quanser.com), and bipolar PWM amplifiers by OkaTech. See Figures 2.4.1 through 3.

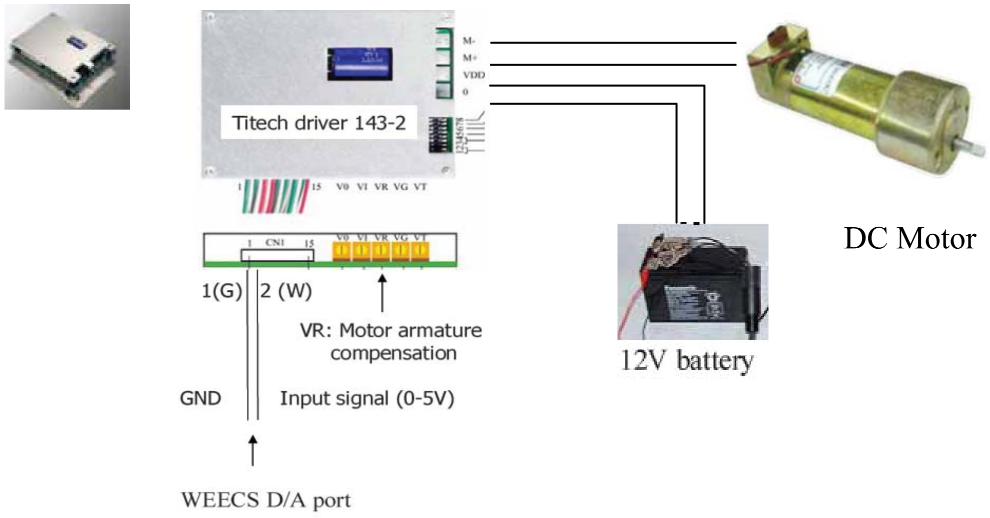
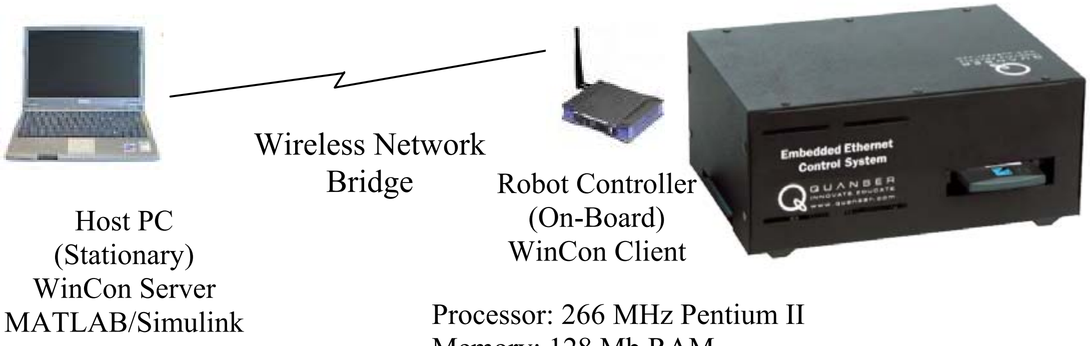


Figure 2.4.1 Bipolar PWM amplifier by OkaTech.


Wireless Network Bridge
Robot Controller (On-Board) WinCon Client
Host PC (Stationary) WinCon Server MATLAB/Simulink
Embedded Ethernet Control System
QUANSER

Processor: 266 MHz Pentium II
Memory: 128 Mb RAM
File System: 256 Mb Compact Flash Solid State Disk
Operating System: Windows XP with VentureCom
RTX real-time extension
Control System Software: WinCom

Figure 2.4.2 On-board and stationary controllers, Quanser.

- 8 A/Ds
- 8 D/As
- 8 digital outputs
- 8 digital inputs
- 8 encoder channels
- 2 clock sources
- Supports keyboard, mouse, video, LAN, USB, printer, and 2 serial ports

Figure 2.4.3 On-board controller

The system consists of a stationary PC, an WEECS onboard controller, a wireless network bridge, several PWM amplifiers, DC Motors, and optical shaft encoders (explained next). The system runs on Windows XP with VentureCom's real-time extension, which handles data I/O for time critical control applications. Eight channels of A/D, D/A, digital I/O and encoder counters are involved in the WEECS controller. The software that runs WEECS is WinCon 5.0, which allows you to run codes generated by MATLAB/Simulink. The WinCon software consists of two parts: WinCon Server and WinCon Client. The former is a graphic user interface on a host PC, while the latter runs in hard real-time on the robot side. The two are wirelessly connected using the TCP/IP protocol. Details will be explained in the laboratory.

2.5 Optical Shaft Encoders

The servomechanism described in the previous section is based on analogue feedback technology, using a potentiometer and a tachometer generator. These analogue feedbacks, although simple, are no longer used in industrial robots and other industrial applications, due to limited reliability and performance. A potentiometer, for example, is poor in reliability, resolution, accuracy, and signal to noise ratio. The output tap of the variable resistance slides on a track of resistive material, making a mechanical contact all the time. This slide contact causes not only electric noise but also wear of the contacting surfaces. The resolution and S/N ratio of the sensor are also limited by the mechanical contact. Furthermore, linearity depends on the uniformity of the resistive material coated on the substrate, and that is a limiting factor of a potentiometer's accuracy. Today's industrial standard is optical shaft encoders, having no sliding contact. This will be discussed next.

2.5.1 Basic principle

An optical encoder consists of a rotating disk with grids, light sources, photodetectors, and electronic circuits. As shown in Figure 2.5.1, a pattern of alternating opaque and translucent grids is printed on the rotating disk. A pair of light source and photodetector is placed on both sides of the rotating disk. As an opaque grid comes in, the light beam is blocked, while it is transmitted through the disk, when the translucent part comes in. The light beam is then detected by the photodetector. The disk is coupled to a motor shaft or a robot joint to measure. As it rotates, an alternating ON-OFF signal is obtained with the photodetector. The number of grids passing through the optical elements represents the distance traveled.

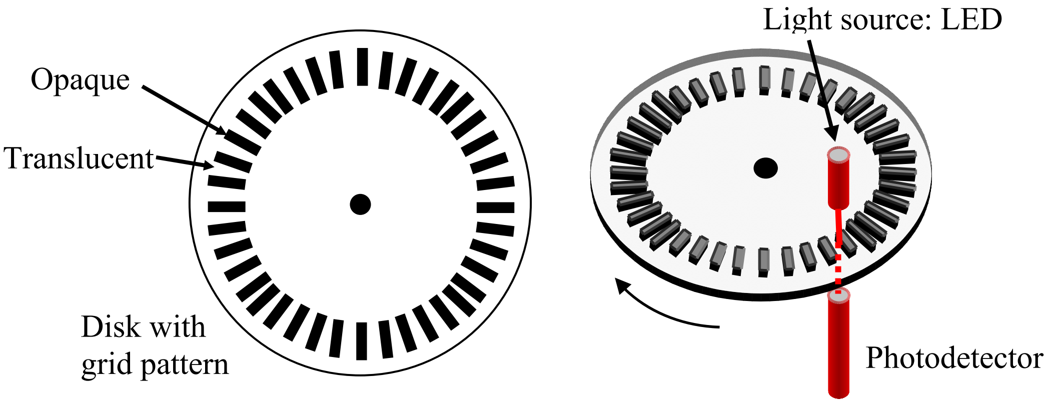


Figure 2.5.1 Basic construction of optical shaft encoder

This optical shaft encoder has no mechanical component making a slide contact, and has no component wear. An optical circuit is not disturbed by electric noise, and the photodetector output is a digital signal, which is more stable than an analogue signal. These make an optical shaft encoder reliable and robust; it is a suitable choice as a feedback sensor for servomotors.

2.5.2 Position measurement

One problem with the above optical encoder design is that the direction of rotation cannot be distinguished from the single photodetector output. The photodetector output is the same for both clockwise and counter-clockwise rotations. There is no indication as to which way the disk is rotating. Counting the pulse number merely gives the total distance the shaft has rotated back and forth. To measure the angular "position", the direction of rotation must be distinguished.

One way of obtaining the directional information is to add another pair of light source/photodetector and a second track of opaque/translucent grids with 90 degrees of phase difference from the first track. Figure 2.5.2 illustrates a double track pattern and resultant output signals for clockwise and counter-clockwise rotations. Note that track A leads track B by 90 degrees for clockwise rotation and that track B leads track A for counter-clockwise rotation. By detecting the phase angle the direction of rotation can be distinguished, and this can be done easily with an up-down counter.

By simply feeding both A phase and B phase encoder signals to an up-down counter, the direction of rotation is first detected, and the number of rising edges and falling edges of both signals is counted in such a way that the counter adds the incoming edge number for clockwise rotation and subtract the edge numbers for counter-clockwise rotation. The up-down counter indicates the cumulative number of edges, that is, the angular "position" of the motor. The output of the up-down counter is binary *n*-bit signals ready to be sent to a digital controller without A/D conversion.

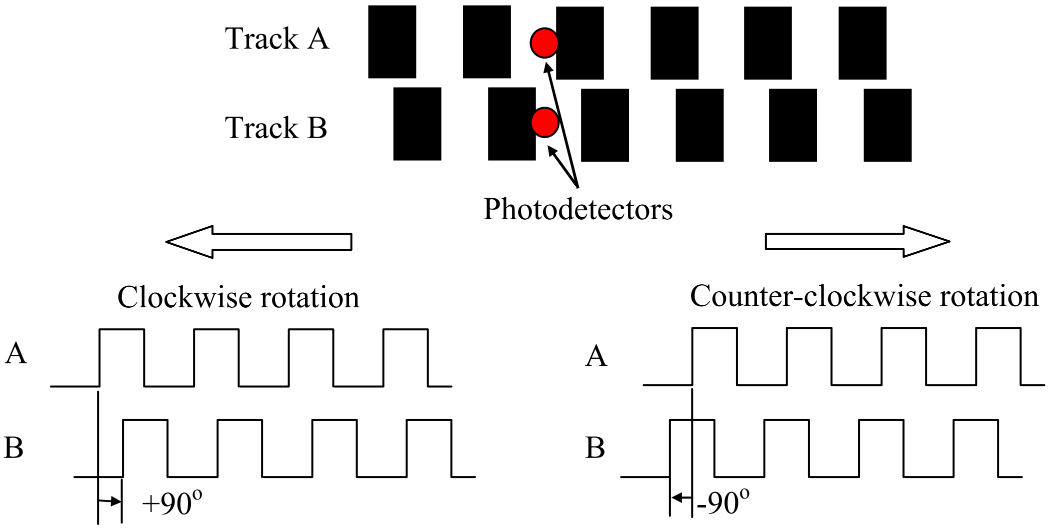


Figure 2.5.2 Double track encode for detection of the direction of rotation

It should be noted that this type of encoder requires initialization of the counter prior to actual measurement. Usually a robot is brought to a home position and the up-down counters are set to the initial state corresponding to the home position. This type of encoder is referred to as an ***incremental encoder***, since A-phase and B-phase signals provide relative displacements from an initial point. Whenever the power supply is shut down, the initialization must be performed for incremental encoders.

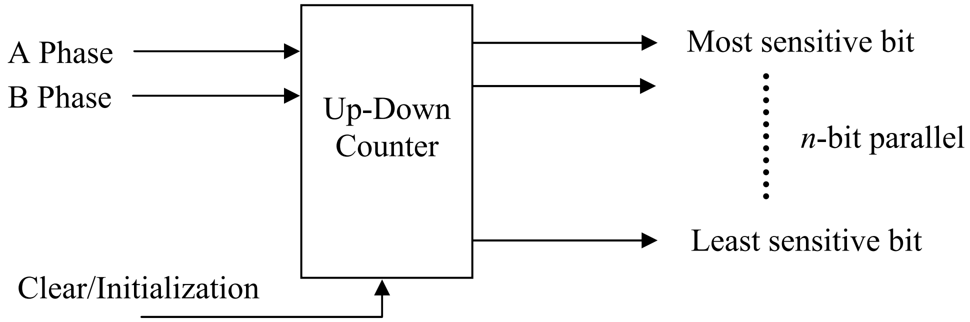
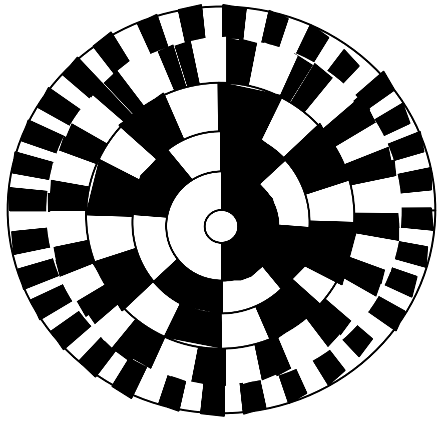


Figure 2.5.3 Up-down counter for an incremental encoder

An ***absolute encoder*** provides an n-bit absolute position as well as the direction of rotation without use of an up-down counter and initialization. As shown in Figure 2.5.4, the rotating disk has n-tracks of opaque-translucent grid patterns and n pairs of light sources and photodetectors. The n-tracks of grid patterns differ in the number of grids; the innermost track has only $1=2^0$ pair of opaque and translucent slits, the second track has $2=2^1$ pairs, and the i-th track has 2^{i-1} pairs. The n outputs from the photodetectors directly indicate the n-bit absolute position of the rotating disk. In general, absolute encoders are more complex and expensive than incremental encoders. In case of power failure, incremental encoders need a laborious initialization procedure for recovery. For quick recovery as well as for safety, absolute encoders are often needed in industrial applications.



Sorry for this poor drawing. I did in haste.

Figure 2.5.4 Absolute encoder

2.5.3 Velocity estimate

Velocity feedback is needed for improving accuracy of speed control as well as for compensating for system dynamics. A salient feature of optical encoders is that velocity information can be obtained along with position measurement. Without use of a dedicated tachometer generator, velocity measurement can be attained by simply processing pulse sequences generated by an optical encoder.

Figure 2.5.5 shows a pulse sequence coming from an optical encoder.[2] Each pulse indicates a rising edge or a falling edge of phase A & B signals. Therefore, the density of this pulse train, i.e. the pulse frequency, is approximately proportional to the angular velocity of the

[2] For simplicity only an incremental encoder is considered.

rotating shaft. The pulse density can be measured by counting the number of incoming pulses in every fixed period, say *T*=10 *ms,* as shown in the figure. This can be done with another up-down counter that counts A phase and B phase pulses. Counting continues only for the fixed sampling period *T*, and the result is sent to a controller at the end of every sampling period. Then the counter is cleared to re-start counting for the next period.

As the sampling period gets shorter, the velocity measurement is updated more frequently, and the delay of velocity feedback gets shorter. However, if the sampling period is too short, discretization error becomes prominent. The problem is more critical when the angular velocity is very small. Not many pulses are generated, and just a few pulses can be counted for a very short period. As the sampling period gets longer, the discretization error becomes smaller, but the time delay may cause instability of the control system.

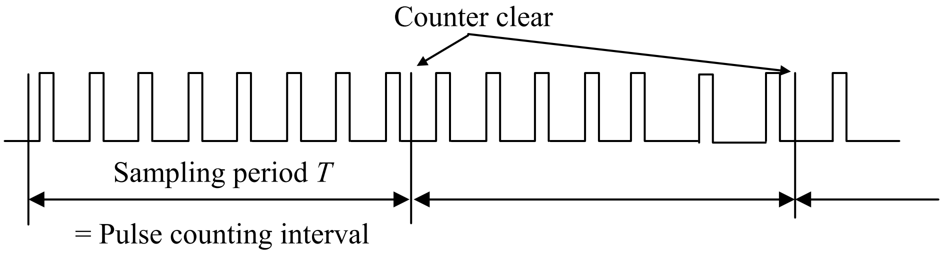


Figure 2.5.5 Velocity estimate based on pulse frequency measurement

An effective method for resolving these conflicting requirements is to use a dual mode velocity measurement. Instead of counting the number of pulses, the *interval* of adjacent pulses is measured at low speed. The reciprocal to the pulse interval gives the angular velocity. As shown in Figure 2.5.6, the time interval can be measured by counting clock pulses. The resolution of this pulse interval measurement is much higher than that of encoder pulse counting in a lower speed range. In contrast, the resolution gets worse at high speed, since the adjacent pulse interval becomes small. Therefore, these two methods supplement to each other. The dual mode velocity measurement uses both counters and switches them depending on the speed.

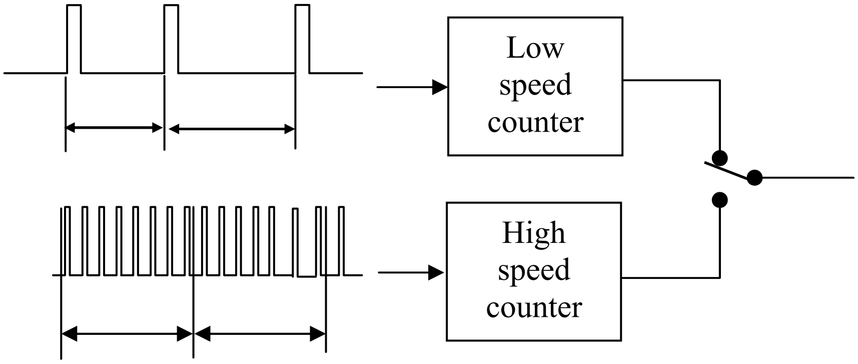


Figure 2.5.6 Dual mode velocity measurement

2.6 Brushless DC Motors

The DC motor described in the previous section is the simplest, yet efficient motor among various actuators applied to robotic systems. Traditional DC motors, however, are limited in reliability and robustness due to wear of the brush and commutation mechanism. In industrial applications where a high level of reliability and robustness is required, DC motors have been replaced by brushless motors and other types of motors having no mechanical commutator. Since brushless motors, or AC synchronous motors, are increasingly used in robotic systems and other automation systems, this section briefly describes its principle and drive methods.

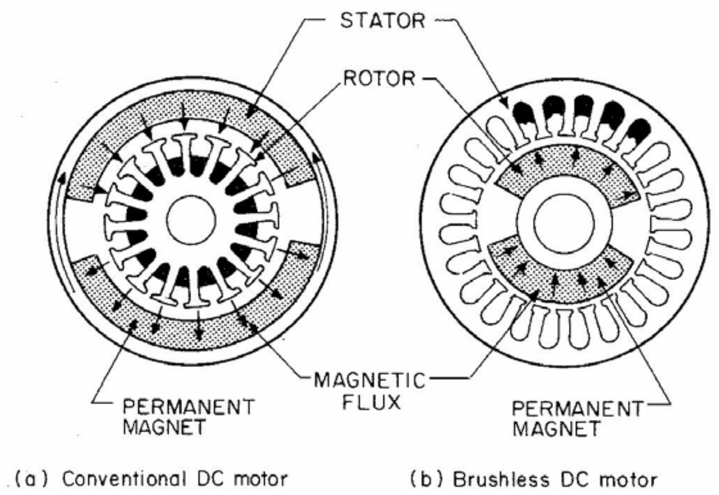


Figure 2.6.1 Construction of brushless DC motor and conventional DC motor

In the brushless motor, the commutation of currents is performed with an electronic switching system. Figure 2.6.1 shows the construction of a brushless motor, compared with a traditional DC motor. In the brushless motor, the rotor and the stator are swapped. Unlike the traditional DC motor, the stator of the brushless motor consists of windings, whereas the rotor comprises permanent magnets. The commutation is accomplished by measuring the rotor position using a position sensor. Depending on the rotor position, currents are delivered to the corresponding windings though electronic switching circuits. The principle of torque generation remains the same, and the torque-speed characteristics and other properties are mostly preserved. Therefore, the brushless motor is highly efficient with added reliability.

A drawback of this brushless motor design is that the torque may change discontinuously when switches are turned on and off as the rotor position changes. In the traditional DC motor this torque ripple is reduced by simply increasing the commutator segments and dividing the windings to many segments. For the brushless motor, however, it is expensive to increase the number of electronic switching circuits. Instead, in the brushless motor the currents flowing into individual windings are varied continuously so that the torque ripple be minimum. A common construction of the windings is that of a three-phase windings, as shown in Figure 2.6.2.

Let I_A, I_B and I_C be individual currents flowing into the three windings shown in the figure. These three currents are varies such that:

$$\begin{aligned} I_A &= I_O \sin\theta \\ I_B &= I_O \sin(\theta + \frac{2}{3}\pi) \\ I_C &= I_O \sin(\theta + \frac{4}{3}\pi) \end{aligned} \tag{2.6.1}$$

where I_O is the scalar magnitude of desired current, and θ is the rotor position. The torque generated is the summation of the three torques generated at the three windings. Taking into account the angle between the magnetic field and the force generated at each air gap, we obtain

$$\tau_m = k_O[I_A \sin\theta + I_B \sin(\theta + \frac{2}{3}\pi) + I_C \sin(\theta + \frac{4}{3}\pi)] \qquad (2.6.2)$$

where k_0 is a proportionality constant. Substituting eq.(1) into eq.(2) yields

$$\tau_m = \frac{2}{3} k_O I_O \qquad (2.6.3)$$

The above expression indicates a linear relationship between the output torque and the scalar magnitude of the three currents. The torque-current characteristics of a brushless motor are apparently the same as the traditional DC motor.

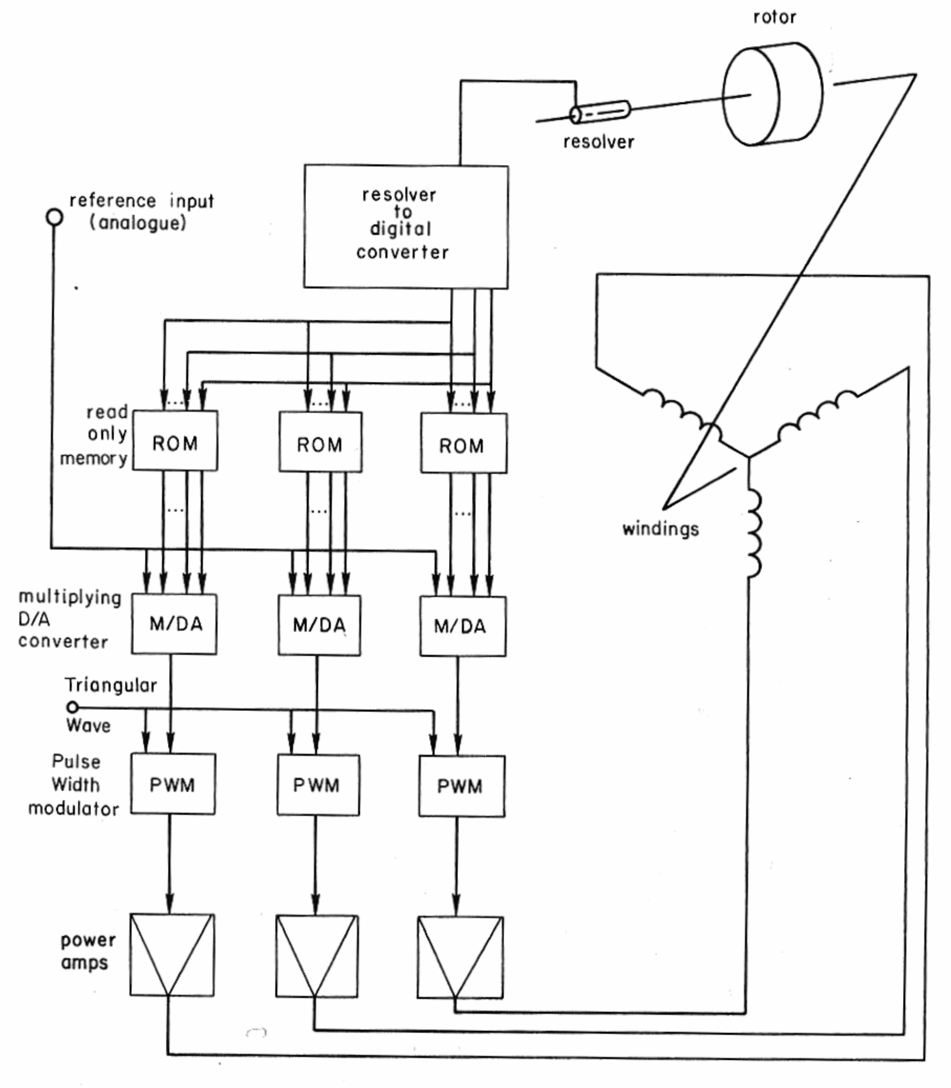


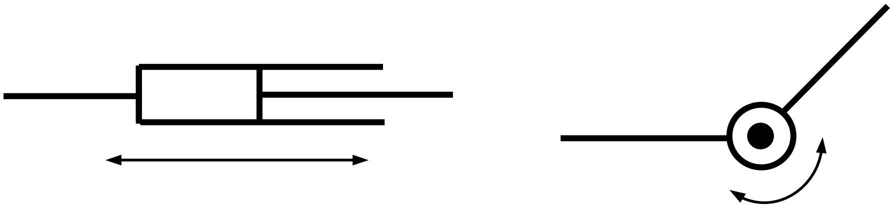
Figure 2.6.2 Brushless DC motor and drive amplifier

Chapter 3
Robot Mechanisms

A robot is a machine capable of physical motion for interacting with the environment. Physical interactions include manipulation, locomotion, and any other tasks changing the state of the environment or the state of the robot relative to the environment. A robot has some form of mechanisms for performing a class of tasks. A rich variety of robot mechanisms has been developed in the last few decades. In this chapter, we will first overview various types of mechanisms used for generating robotic motion, and introduce some taxonomy of mechanical structures before going into a more detailed analysis in the subsequent chapters.

3.1 Joint Primitives and Serial Linkages

A robot mechanism is a multi-body system with the multiple bodies connected together. We begin by treating each body as rigid, ignoring elasticity and any deformations caused by large load conditions. Each rigid body involved in a robot mechanism is called a link, and a combination of links is referred to as a linkage. In describing a linkage it is fundamental to represent how a pair of links is connected to each other. There are two types of primitive connections between a pair of links, as shown in Figure 3.1.1. The first is a prismatic joint where the pair of links makes a translational displacement along a fixed axis. In other words, one link slides on the other along a straight line. Therefore, it is also called a sliding joint. The second type of primitive joint is a revolute joint where a pair of links rotates about a fixed axis. This type of joint is often referred to as a hinge, articulated, or rotational joint.[1]



(a) Prismatic joint (b) Revolute joint

Figure 3.1.1 Primitive joint types: (a) a prismatic joint and (b) a revolute joint

Combining these two types of primitive joints, we can create many useful mechanisms for robot manipulation and locomotion. These two types of primitive joints are simple to build and are well grounded in engineering design. Most of the robots that have been built are combinations of only these two types. Let us look at some examples.

Robot mechanisms analogous to coordinate systems

One of the fundamental functional requirements for a robotic system is to locate its end-effecter, e.g. a hand, a leg, or any other part of the body performing a task, in three-dimensional space. If the kinematic structure of such a robot mechanism is analogous to a coordinate system,

[1] It is interesting to note that all biological creatures are made of revolute type joints; there are no sliding joints involved in their extremities.

it may suffice this positioning requirement. Figures 3.1.2 ~ 4 show three types of robot arm structures corresponding to the Cartesian coordinate system, the cylindrical coordinate system, and the spherical coordinate system respectively. The Cartesian coordinate robot shown in Figure 3.1.2 has three prismatic joints, corresponding to three axes denoted *x, y ,* and *z.* The cylindrical robot consists of one revolute joint and two prismatic joints, with θ, *r,* and *z* representing the coordinates of the end-effecter. Likewise, the spherical robot has two revolute joints denoted θ and ϕ and one prismatic joint denoted *r.*

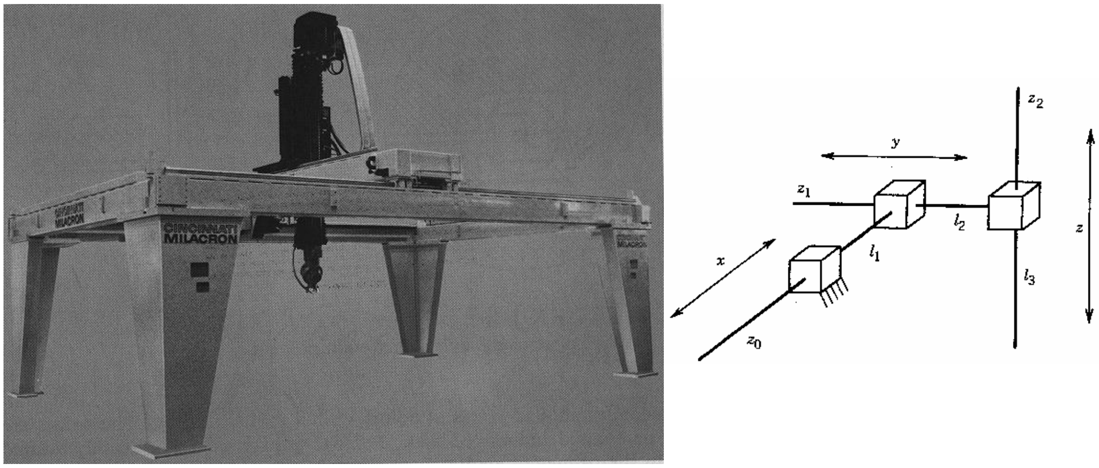


Figure 3.1.2 Cartesian coordinate robot

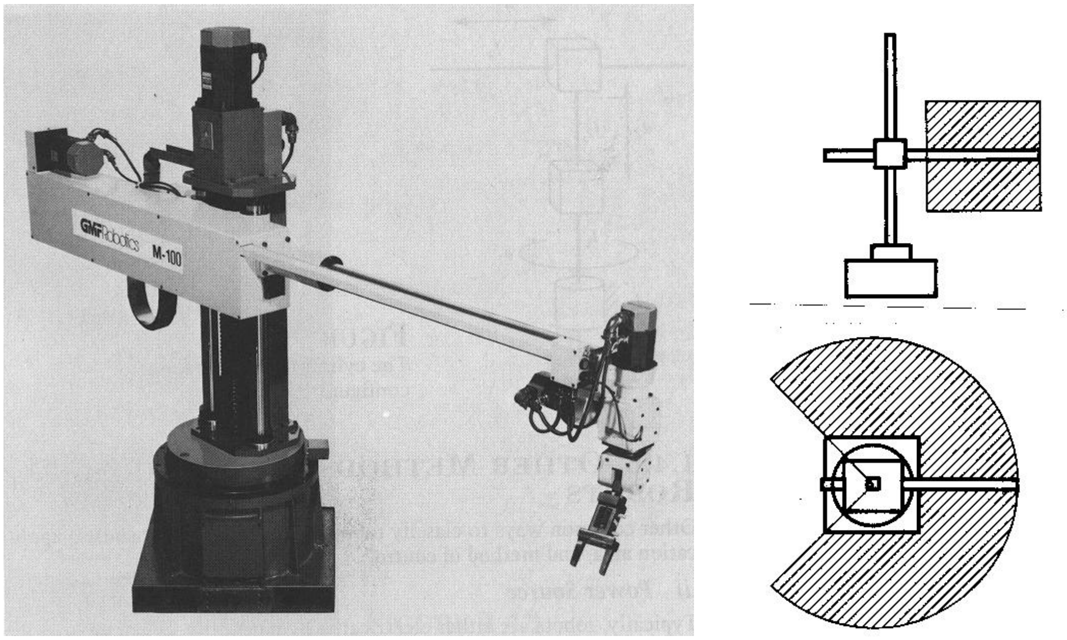


Figure 3.1.3 Cylindrical coordinate robot

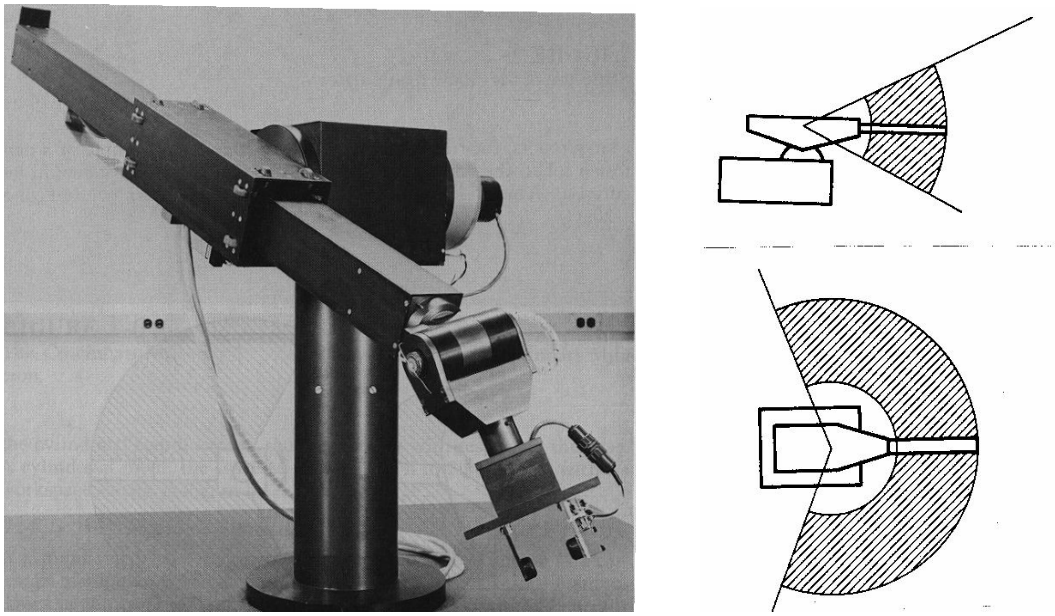


Figure 3.1.4 Spherical coordinate robot

There are many other ways of locating an end-effecter in three-dimensional space. Figure 3.1.5 ~ 7 show three other kinematic structures that allow the robot to locate its end-effecter in three-dimensional space. Although these mechanisms have no analogy with common coordinate systems, they are capable of locating the end-effecter in space, and have salient features desirable for specific tasks. The first one is a so-called SCALAR robot consisting of two revolute joints and one prismatic joint. This robot structure is particularly desirable for assembly automation in manufacturing systems, having a wide workspace in the horizontal direction and an independent vertical axis appropriate for insertion of parts.

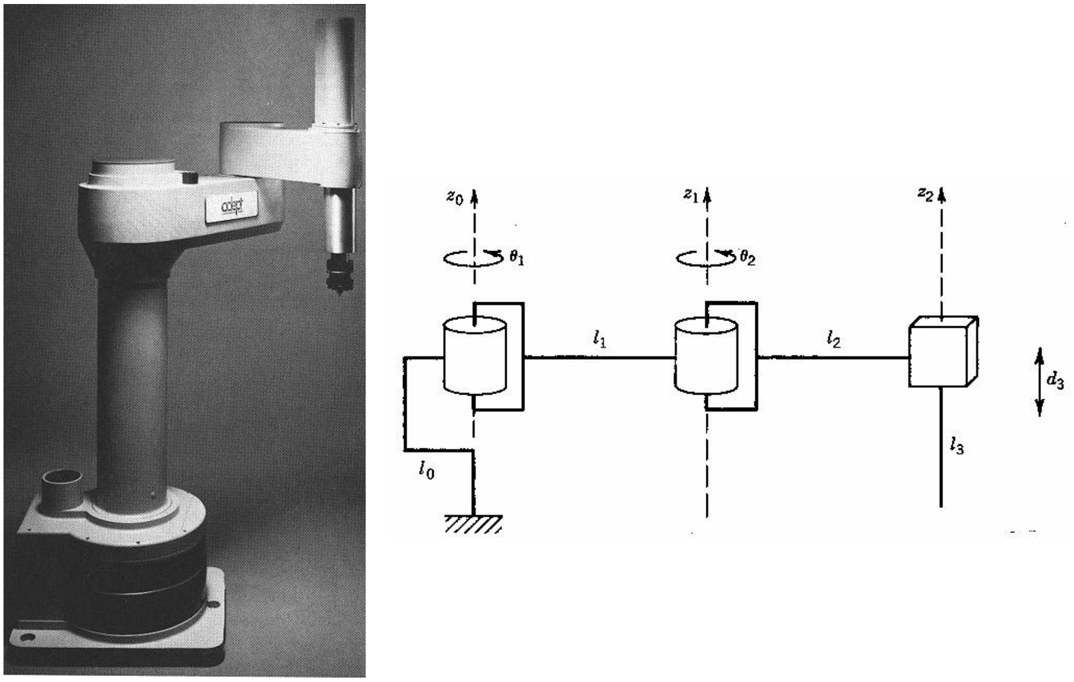


Figure 3.1.5 SCALAR type robot

The second type, called an articulated robot or an elbow robot, consists of all three revolute joints, like a human arm. This type of robot has a great degree of flexibility and versatility, being the most standard structure of robot manipulators. The third kinematic structure, also consisting of three revolute joints, has a unique mass balancing structure. The counter balance at the elbow eliminates gravity load for all three joints, thus reducing toque requirements for the actuators. This structure has been used for the direct-drive robots having no gear reducer.

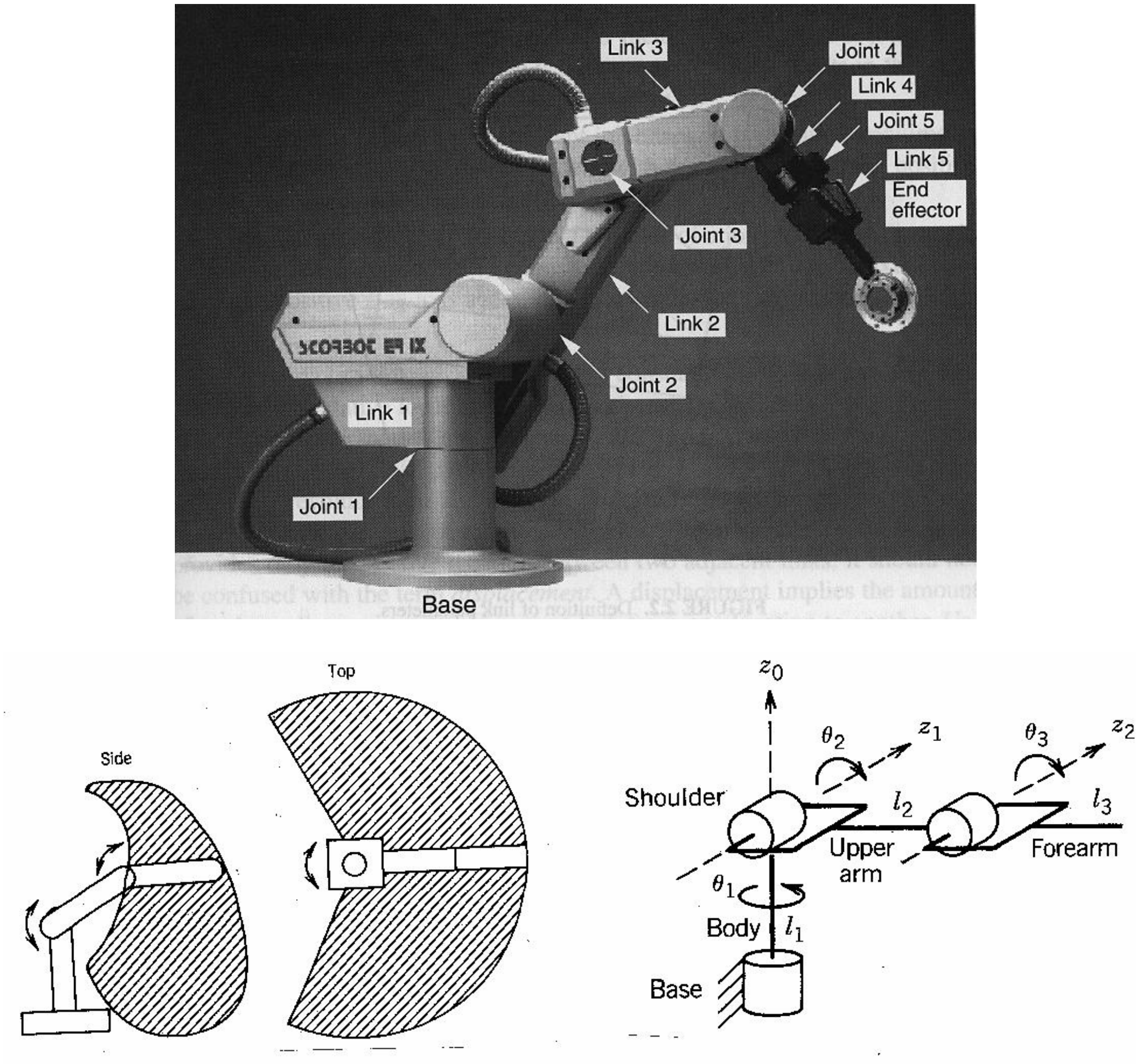


Figure 3.1.6 Articulated robot

Note that all the above robot structures are made of serial connections of primitive joints. This class of kinematic structures, termed a serial linkage, constitutes the fundamental makeup of robot mechanisms. They have no kinematic constraint in each joint motion, i.e. each joint displacement is a generalized coordinate. This facilitates the analysis and control of the robot mechanism. There are, however, different classes of mechanisms used for robot structures. Although more complex, they do provide some useful properties. We will look at these other mechanisms in the subsequent sections.

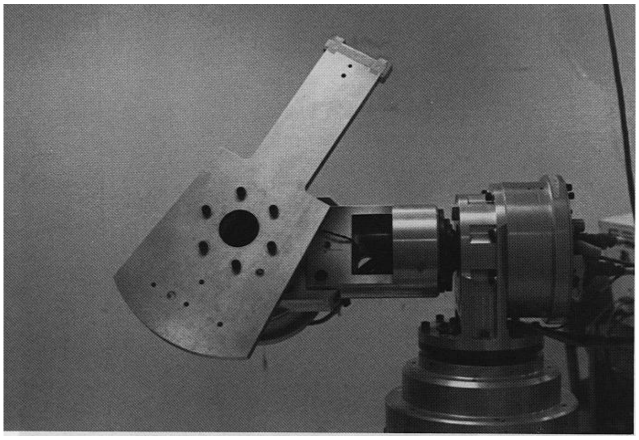


Figure 3.1.7 Gravity-balanced robot with three-revolute joints

3.2 Parallel Linkages

Primitive joints can be arranged in parallel as well as in series. Figure 3.2.1 illustrates such a parallel link mechanism. It is a five-bar-linkage consisting of five links, including the base link, connected by five joints. This can be viewed as two serial linkage arms connected at a particular point, point A in the figure. It is important to note that there is a closed kinematic chain formed by the five links and, thereby, the two serial link arms must conform to a certain geometric constraint. It is clear from the figure that the end-effecter position is determined if two of the five joint angles are given. For example, if angles θ_1 and θ_3 of joints 1 and 3 are determined, then all the link positions are determined, as is the end-effecter's. Driving joints 1 and 3 with two actuators, we can move the end-effecter within the vertical plane. It should be noted that, if more than two joints were actively driven by independent actuators, a conflict among three actuators would occur due to the closed-loop kinematic chain. Three of the five joints should be ***passive joints***, which are free to rotate. Only two joints should be ***active joints,*** driven by independent actuators.

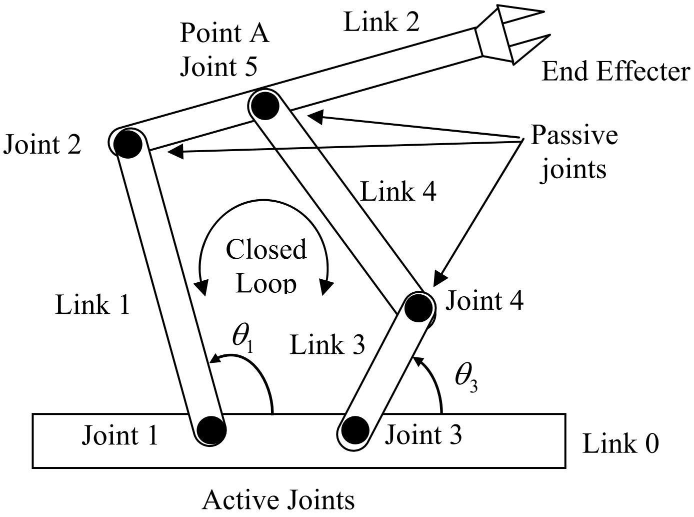


Figure 3.2.1 Five-bar-link parallel link robot

This type of parallel linkage, having a closed-loop kinematic chain, has significant features. First, placing both actuators at the base link makes the robot arm lighter, compared to the serial link arm with the second motor fixed to the tip of link 1. Second, a larger end-effecter load can be born with the two serial linkage arms sharing the load. Figure 3.2.2 shows a heavy-duty robot having a parallel link mechanism.

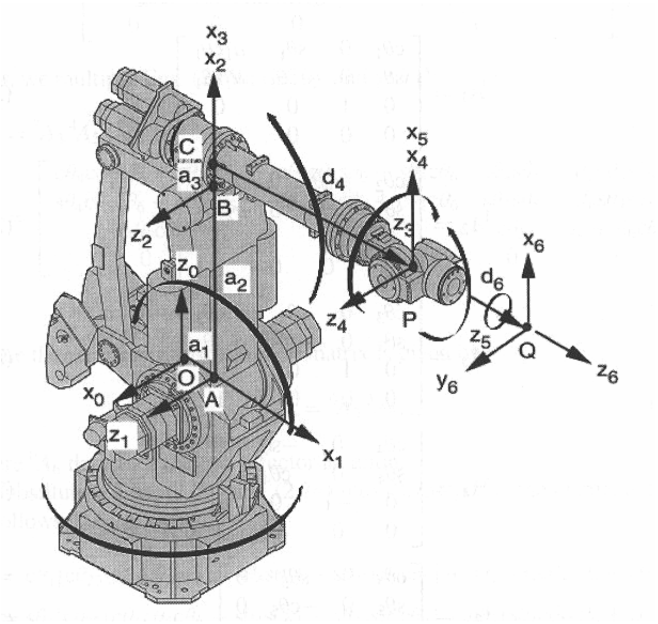


Figure 3.2.2 Heavy-duty robot with parallel link mechanism

Figure 3.2.3 shows the Stewart mechanism, which consists of a moving platform, a fixed base, and six powered cylinders connecting the moving platform to the base frame. The position and orientation of the moving platform are determined by the six independent actuators. The load acting on the moving platform is born by the six "arms". Therefore, the load capacity is generally large, and dynamic response is fast for this type of robot mechanisms. Note, however, that this mechanism has spherical joints, a different type of joints than the primitive joints we considered initially.

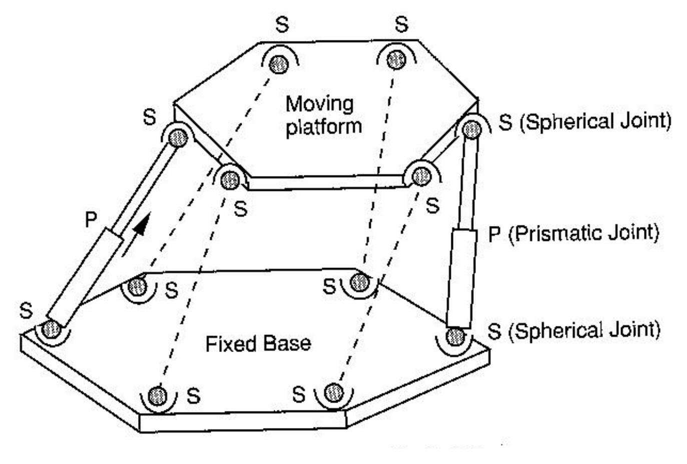


Figure 3.2.3 Stewart mechanism parallel-link robot

Chapter 4
Kinematics

Kinematics is *Geometry of Motion*. It is one of the most fundamental disciplines in robotics, providing tools for describing the structure and behavior of robot mechanisms. In this chapter, we will discuss how the motion of a robot mechanism is described, how it responds to actuator movements, and how the individual actuators should be coordinated to obtain desired motion at the robot end-effecter. These are questions central to the design and control of robot mechanisms.

To begin with, we will restrict ourselves to a class of robot mechanisms that work within a plane, i.e. *Planar Kinematics*. Planar kinematics is much more tractable mathematically, compared to general three-dimensional kinematics. Nonetheless, most of the robot mechanisms of practical importance can be treated as planar mechanisms, or can be reduced to planar problems. General three-dimensional kinematics, on the other hand, needs special mathematical tools, which will be discussed in later chapters.

4.1 Planar Kinematics of Serial Link Mechanisms

Example 4.1 Consider a three degree-of-freedom, planar robot arm shown in Figure 4.1.1. The arm consists of one fixed link and three movable links that move within the plane. All the links are connected by revolute joints whose joint axes are all perpendicular to the plane of the links. There is no closed-loop kinematic chain; hence, it is a serial link mechanism.

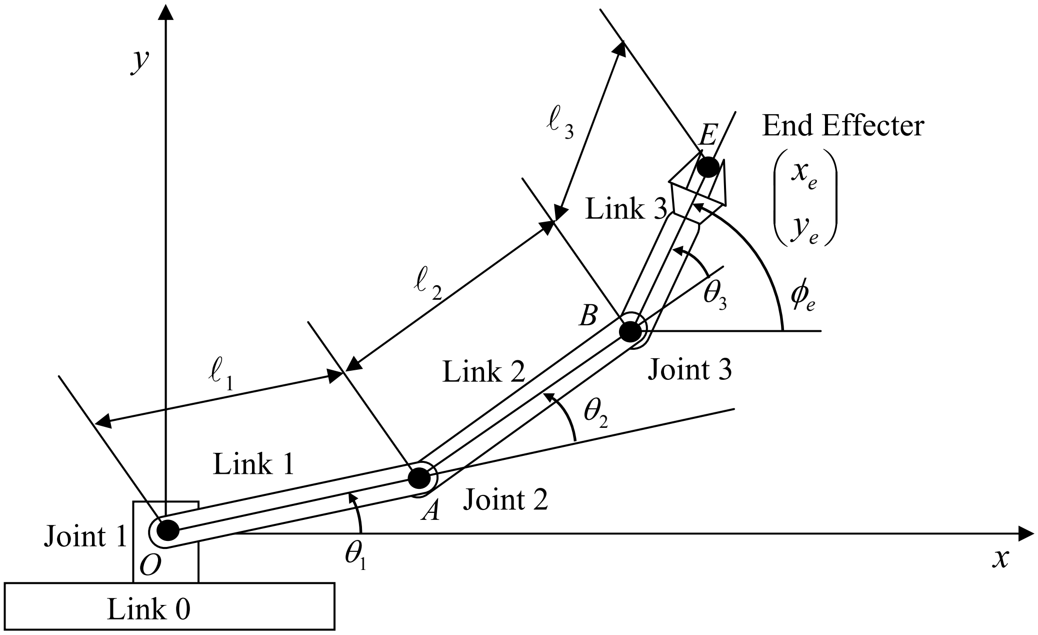


Figure 4.1.1 Three dof planar robot with three revolute joints

To describe this robot arm, a few geometric parameters are needed. First, the length of each link is defined to be the distance between adjacent joint axes. Let points *O, A,* and *B* be the locations of the three joint axes, respectively, and point E be a point fixed to the end-effecter. Then the link lengths are $\ell_1 = \overline{OA}$, $\ell_2 = \overline{AB}$, $\ell_3 = \overline{BE}$. Let us assume that Actuator 1 driving link 1 is fixed to the base link (link 0), generating angle θ_1, while Actuator 2 driving link 2 is fixed to the tip of Link 1, creating angle θ_2 between the two links, and Actuator 3 driving Link 3 is fixed to the tip of Link 2, creating angle θ_3, as shown in the figure. Since this robot arm performs task by moving its end-effecter at point E, we are concerned with the location of the end-effecter. To describe its location, we use a coordinate system, *O-xy,* fixed to the base link with the origin at the first joint, and describe the end-effecter position with coordinates x_e and y_e. We can relate the end-effecter coordinates to the joint angles determined by the three actuators by using the link lengths and joint angles defined above:

$$x_e = \ell_1 \cos\theta_1 + \ell_2 \cos(\theta_1 + \theta_2) + \ell_3 \cos(\theta_1 + \theta_2 + \theta_3) \tag{4.1.1}$$

$$y_e = \ell_1 \sin\theta_1 + \ell_2 \sin(\theta_1 + \theta_2) + \ell_3 \sin(\theta_1 + \theta_2 + \theta_3) \tag{4.1.2}$$

This three dof robot arm can locate its end-effecter at a desired orientation as well as at a desired position. The orientation of the end-effecter can be described with the angle of the centerline of the end-effecter measured from the positive *x* coordinate axis. This end-effecter orientation ϕ_e is related to the actuator displacements as

$$\phi_e = \theta_1 + \theta_2 + \theta_3 \tag{4.1.3}$$

□

The above three equations describe the position and orientation of the robot end-effecter viewed from the fixed coordinate system in relation to the actuator displacements. In general, a set of algebraic equations relating the position and orientation of a robot end-effecter, or any significant part of the robot, to actuator displacements, or displacements of active joints, is called ***Kinematic Equations,*** or more specifically, ***Forward Kinematic Equations*** in the robotics literature.

Exercise 4.1

Shown below in Figure 4.1.2 is a planar robot arm with two revolute joints and one prismatic joint. Using the geometric parameters and joint displacements, obtain the kinematic equations relating the end-effecter position and orientation to the joint displacements.

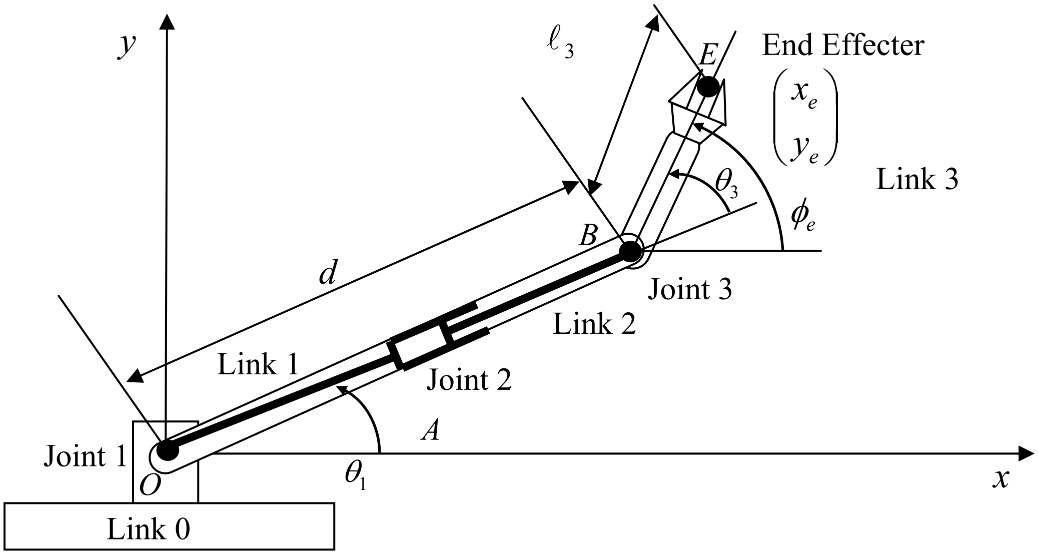


Figure 4.1.2 Three dof robot with two revolute joints and one prismatic joint

Now that the above Example and Exercise problems have illustrated kinematic equations, let us obtain a formal expression for kinematic equations. As mentioned in the previous chapter, two types of joints, prismatic and revolute joints, constitute robot mechanisms in most cases. The displacement of the *i-th* joint is described by distance d_i if it is a prismatic joint, and by angle θ_i for a revolute joint. For formal expression, let us use a generic notation: q_i. Namely, joint displacement q_i represents either distance d_i or angle θ_i depending on the type of joint.

$$q_i = \begin{cases} d_i & \text{Prismatic joint} \\ \theta_i & \text{Revolute joint} \end{cases} \tag{4.1.4}$$

We collectively represent all the joint displacements involved in a robot mechanism with a column vector: $q = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix}^T$, where *n* is the number of joints. Kinematic equations relate these joint displacements to the position and orientation of the end-effecter. Let us collectively denote the end-effecter position and orientation by vector ***p.*** For planar mechanisms, the end-effecter location is described by three variables:

$$p = \begin{bmatrix} x_e \\ y_e \\ \phi_e \end{bmatrix} \tag{4.1.5}$$

Using these notations, we represent kinematic equations as a vector function relating ***p*** to ***q***:

$$p = f(q), \quad p \in \Re^{3x1}, q \in \Re^{nx1} \tag{4.1.6}$$

For a serial link mechanism, all the joints are usually active joints driven by individual actuators. Except for some special cases, these actuators uniquely determine the end-effecter position and orientation as well as the configuration of the entire robot mechanism. If there is a link whose location is not fully determined by the actuator displacements, such a robot mechanism is said to be ***under-actuated***. Unless a robot mechanism is under-actuated, the collection of the joint displacements, i.e. the vector ***q,*** uniquely determines the entire robot configuration. For a serial link mechanism, these joints are independent, having no geometric constraint other than their stroke limits. Therefore, these joint displacements are ***generalized coordinates*** that locate the robot mechanism uniquely and completely. Formally, the number of generalized coordinates is called ***degrees of freedom.*** Vector ***q*** is called joint coordinates, when they form a complete and independent set of generalized coordinates.

4.2 Inverse Kinematics of Planar Mechanisms

The vector kinematic equation derived in the previous section provides the functional relationship between the joint displacements and the resultant end-effecter position and orientation. By substituting values of joint displacements into the right-hand side of the kinematic equation, one can immediately find the corresponding end-effecter position and orientation. The problem of finding the end-effecter position and orientation for a given set of joint displacements is referred to as the *direct kinematics problem.* This is simply to evaluate the right-hand side of the kinematic equation for known joint displacements. In this section, we discuss the problem of moving the end-effecter of a manipulator arm to a specified position and orientation. We need to find the joint displacements that lead the end-effecter to the specified position and orientation. This is the inverse of the previous problem, and is thus referred to as the *inverse kinematics problem.* The kinematic equation must be solved for joint displacements, given the end-effecter position and orientation. Once the kinematic equation is solved, the desired end-effecter motion can be achieved by moving each joint to the determined value.

In the direct kinematics problem, the end-effecter location is determined uniquely for any given set of joint displacements. On the other hand, the inverse kinematics is more complex in the sense that multiple solutions may exist for the same end-effecter location. Also, solutions may not always exist for a particular range of end-effecter locations and arm structures. Further, since the kinematic equation is comprised of nonlinear simultaneous equations with many trigonometric functions, it is not always possible to derive a closed-form solution, which is the explicit inverse function of the kinematic equation. When the kinematic equation cannot be solved analytically, numerical methods are used in order to derive the desired joint displacements.

Example 4.2 Consider the three dof planar arm shown in Figure 4.1.1 again. To solve its inverse kinematics problem the kinematic structure is redrawn in Figure 4.2.1. The problem is to find three joint angles $\theta_1, \theta_2, \theta_3$ leading the end effecter to desired position and orientation, x_e, y_e, ϕ_e. We take a two-step approach. First, we find the position of the wrist, point B, from x_e, y_e, ϕ_e. Then we find θ_1, θ_2 from the wrist position. Angle θ_3 can be determined immediately from the wrist position.

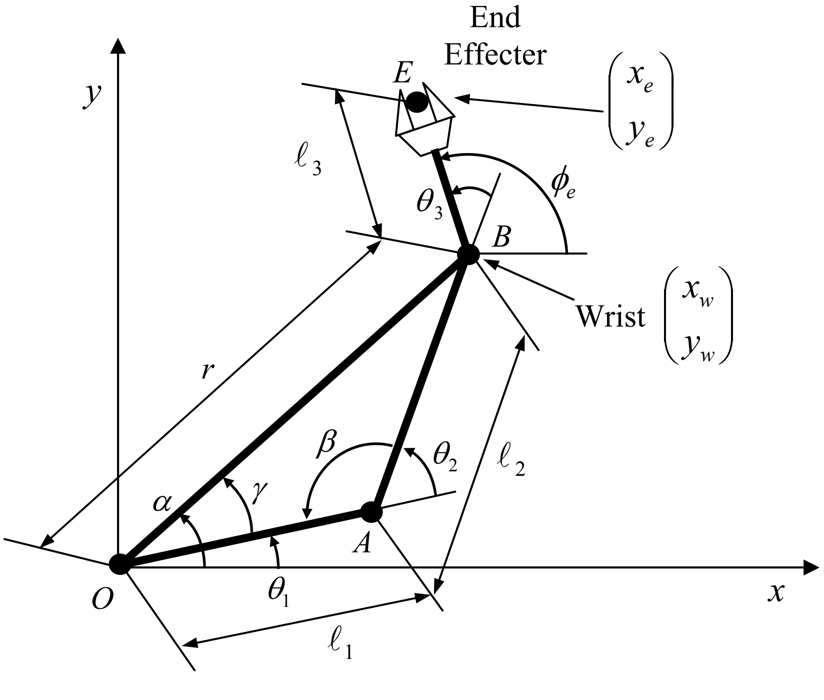


Figure 4.2.1 Skeleton structure of the robot arm of Example 4.1

Let x_w and y_w be the coordinates of the wrist. As shown in Figure 4.2.1, point B is at distance ℓ_3 from the given end-effecter position E. Moving in the opposite direction to the end effecter orientation ϕ_e, the wrist coordinates are given by

$$x_w = x_e - \ell_3 \cos\phi_e$$
$$y_w = y_e - \ell_3 \sin\phi_e \tag{4.2.1}$$

Note that the right hand side of the above equations is functions of x_e, y_e, ϕ_e alone. From these wrist coordinates, we can determine the angle α shown in the figure.[1]

$$\alpha = \tan^{-1}\frac{y_w}{x_w} \tag{4.2.2}$$

Next, let us consider the triangle *OAB* and define angles β, γ , as shown in the figure. This triangle is formed by the wrist *A*, the elbow *B*, and the shoulder *O*. Applying the cosine law to the elbow angle β yields

$$\ell_1^2 + \ell_2^2 - 2\ell_1\ell_2\cos\beta = r^2 \tag{4.2.3}$$

where $r^2 = p_x^2 + p_y^2$, the squared distance between O and B. Solving this for angle β yields

$$\theta_2 = \pi - \beta = \pi - \cos^{-1}\frac{\ell_1^2 + \ell_2^2 - x_w^2 - y_w^2}{2\ell_1\ell_2} \tag{4.2.4}$$

Similarly,

$$r^2 + \ell_1^2 - 2r\ell_1\cos\gamma = \ell_2^2 \tag{4.2.5}$$

Solving this for γ yields

$$\theta_1 = \alpha - \gamma = \tan^{-1}\frac{y_w}{x_w} - \cos^{-1}\frac{x_w^2 + y_w^2 + \ell_1^2 - \ell_2^2}{2\ell_1\sqrt{x_w^2 + y_w^2}} \tag{4.2.6}$$

From the above θ_1, θ_2 we can obtain

$$\theta_3 = \phi_e - \theta_1 - \theta_2 \tag{4.2.7}$$

Eqs. (4), (6), and (7) provide a set of joint angles that locates the end-effecter at the desired position and orientation. It is interesting to note that there is another way of reaching the same end-effecter position and orientation, i.e. another solution to the inverse kinematics problem. Figure 4.2.2 shows two configurations of the arm leading to the same end-effecter location: the elbow down configuration and the elbow up configuration. The former corresponds to the solution obtained above. The latter, having the elbow position at point *A'*, is symmetric to the former configuration with respect to line *OB*, as shown in the figure. Therefore, the two solutions are related as

$$\begin{aligned} \theta_1' &= \theta_1 + 2\gamma \\ \theta_2' &= -\theta_2 \\ \theta_3' &= \phi_e - \theta_1' - \theta_2' = \theta_3 + 2\theta_2 - 2\gamma \end{aligned} \tag{4.2.8}$$

Inverse kinematics problems often possess multiple solutions, like the above example, since they are nonlinear. Specifying end-effecter position and orientation does not uniquely determine the whole configuration of the system. This implies that vector ***p***, the collective position and orientation of the end-effecter, cannot be used as generalized coordinates.

The existence of multiple solutions, however, provides the robot with an extra degree of flexibility. Consider a robot working in a crowded environment. If multiple configurations exist for the same end-effecter location, the robot can take a configuration having no interference with

[1] Unless noted specifically we assume that the arc tangent function takes an angle in a proper quadrant consistent with the signs of the two operands.

the environment. Due to physical limitation, however, the solutions to the inverse kinematics problem do not necessarily provide feasible configurations. We must check whether each solution satisfies the constraint of movable range, i.e. stroke limit of each joint.

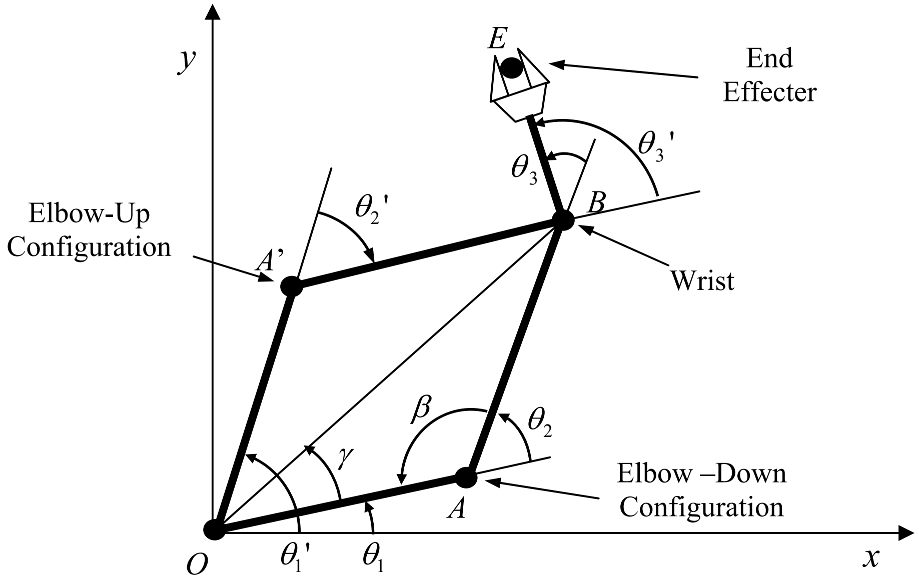


Figure 4.2.2 Multiple solutions to the inverse kinematics problem of Example 4.2

4.3 Spatial Kinematics of a Class of Robots

Kinematics of general three-dimensional robots is much more complicated than that of planar robots. However, a class of spatial robot structures that have been used in many applications can be dealt with in the same way as planar kinematics. Both forward and inverse problems can be solved by applying the same approach as we have obtained for planar robots. The following exercise problem of a spherical coordinate robot illustrates this.

Exercise 4.2 **Spherical Coordinate Robot**

Shown below is the schematic of a three dof spherical coordinate robot. Although this arm looks three-dimensional, its kinematic equations can be obtained by applying the same planar kinematic equations as obtained for Exercise 4.1. For joints 2 and 3 alone, consider a vertical plane containing links 2 and 3. It is identical to the first two joints of Exercise 4.1. As for joint 1, consider the projection of the endpoint onto the *xy* plane. Again it is a planar kinematics problem. Answer the following questions, using the notation shown in the figure.

(a) Obtain the kinematic equations relating the endpoint coordinates, x_e, y_e, z_e, to joint angles θ_1, θ_2, d_3.

(b) Solve the inverse kinematics problem, i.e. obtain the joint coordinates, given the endpoint coordinates. Obtain all of the multiple solutions, assuming that each revolute joint is allowed to rotate 360 degrees and that the prismatic joint is restricted to $d_3 \geq 0$.

(c) Sketch the arm configuration for each of the multiple solutions.

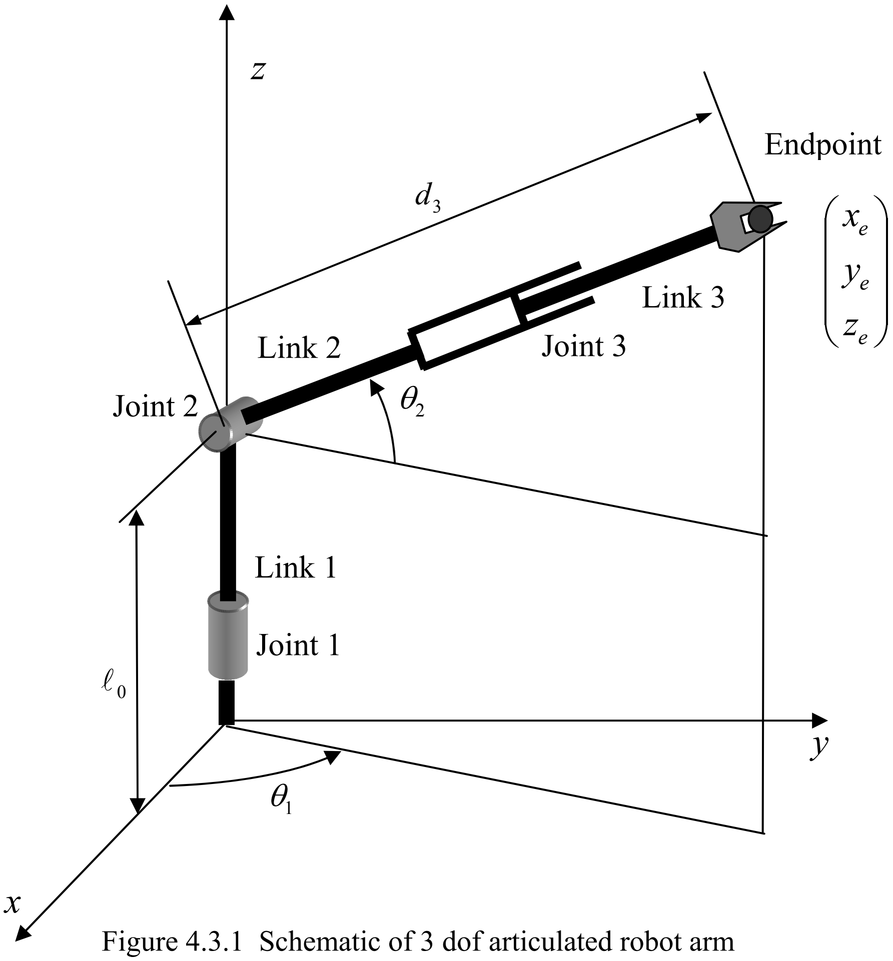


Figure 4.3.1 Schematic of 3 dof articulated robot arm

4.4 Kinematics of Parallel Link Mechanisms

As discussed in Chapter 3, parallel link mechanisms have been used for many applications, particularly for heavy duty and precision applications. Since a parallel link mechanism contains a closed kinematic chain, formulating kinematic equations is more involved. In general, closed-form kinematic equations relating its end effecter location to joint displacements cannot be obtained directly. A standard procedure for obtaining kinematic equations includes:

- Break down the closed kinematic chain into multiple open kinematic chains,
- Formulate kinematic equations for each of the open kinematic chain, and
- Solve the set of simultaneous kinematic equations for independent joint displacements and the end point coordinates.

Let us apply this procedure to a five-bar-link robot with a closed kinematic chain.

Example 4.3 Consider the five-bar-link planar robot arm shown in Figure 4.4.1. Joints 1 and 3 are active joints driven by independent actuators, but the other joints are free joints, which are determined by the active joints. Let us first break the closed kinematic chain at Joint 5, and create two open kinematic chains; Links 1 and 2 vs. Links 3 and 4 . From the figure, we obtain:

$$\begin{aligned} x_e &= \ell_1 \cos\theta_1 + \ell_2 \cos\theta_2 \\ y_e &= \ell_1 \sin\theta_1 + \ell_2 \sin\theta_2 \end{aligned} \tag{4.4.1}$$

for Links 1 and 2, and

$$\begin{aligned} x_A &= \ell_3 \cos\theta_3 + \ell_4 \cos\theta_4 \\ y_A &= \ell_3 \sin\theta_3 + \ell_4 \sin\theta_4 \end{aligned} \tag{4.4.2}$$

for Links 3 and 4.

Note that, in Eq. (1), Joint 2 is a passive joint. Hence, angle θ_2 is a dependent variable. Using θ_2, however, we can obtain the coordinates of point A:

$$\begin{aligned} x_A &= \ell_1 \cos\theta_1 + \ell_5 \cos\theta_2 \\ y_A &= \ell_1 \sin\theta_1 + \ell_5 \sin\theta_2 \end{aligned} \tag{4.4.3}$$

Equating (2) and (3) yields two constraint equations:

$$\begin{aligned} \ell_1 \cos\theta_1 + \ell_5 \cos\theta_2 &= \ell_3 \cos\theta_3 + \ell_4 \cos\theta_4 \\ \ell_1 \sin\theta_1 + \ell_5 \sin\theta_2 &= \ell_3 \sin\theta_3 + \ell_4 \sin\theta_4 \end{aligned} \tag{4.4.4}$$

Note that there are four variables and two constraint equations. Therefore, two of them, e.g. θ_1, θ_3, are independent. Solving Eq.(4) for θ_2 and θ_4 and substituting θ_2 into Eq.(1) yield forward kinematic equations relating the end point coordinates to independent joint displacements. It should be noted that multiple solutions exist for these constraint equations (4).

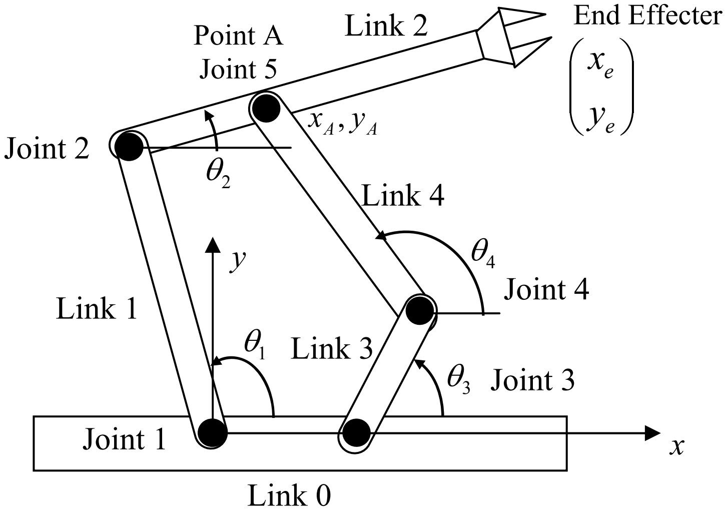


Figure 4.4.1 Five-bar-link mechanism

Although the forward kinematic equations are difficult to write out explicitly, the inverse kinematic equations can be obtained for this parallel link mechanism. The problem is to find θ_1, θ_3 that lead the endpoint to a desired position: x_e, y_e. We can take the following procedure:

Step 1 Given x_e, y_e, find θ_1, θ_2 by solving the two-link inverse kinematics problem.

Step 2 Given θ_1, θ_2, obtain x_A, y_A. This is a forward kinematics problem.

Step 3 Given x_A, y_A, find θ_3, θ_4 by solving another two-link inverse kinematics problem.

Exercise 4.3 Obtain the joint angles of the dog's legs, given the body position and orientation.

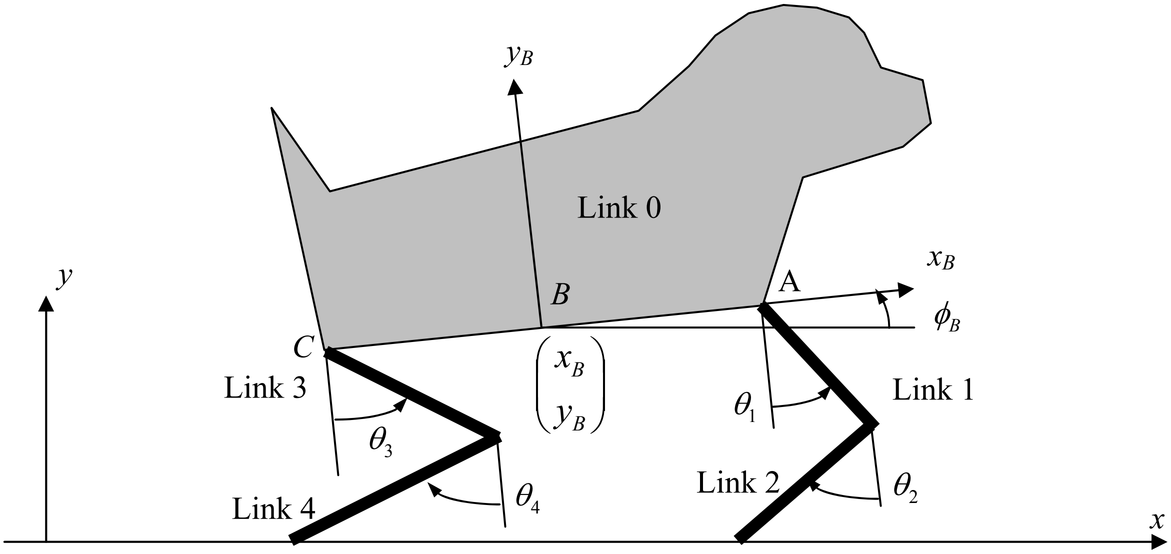


Figure 4.4.2 A doggy robot with two legs on the ground

The inverse kinematics problem:

Step 1 Given x_B, y_B, ϕ_B, find x_A, y_A and x_C, y_C

Step 2 Given x_A, y_A, find θ_1, θ_2

Step 3 Given x_C, y_C, find θ_3, θ_4

4.5 Redundant mechanisms

A manipulator arm must have at least six degrees of freedom in order to locate its end-effecter at an arbitrary point with an arbitrary orientation in space. Manipulator arms with less than 6 degrees of freedom are not able to perform such arbitrary positioning. On the other hand, if a manipulator arm has more than 6 degrees of freedom, there exist an infinite number of solutions to the kinematic equation. Consider for example the human arm, which has seven degrees of freedom, excluding the joints at the fingers. Even if the hand is fixed on a table, one can change the elbow position continuously without changing the hand location. This implies that there exist an infinite set of joint displacements that lead the hand to the same location. Manipulator arms with more than six degrees of freedom are referred to as *redundant manipulators.* We will discuss redundant manipulators in detail in the following chapter.

Chapter 5
Differential Motion

In the previous chapter, the position and orientation of the manipulator end-effecter were evaluated in relation to joint displacements. The joint displacements corresponding to a given end-effecter location were obtained by solving the kinematic equation for the manipulator. This preliminary analysis permitted the robotic system to place the end-effecter at a specified location in space. In this chapter, we are concerned not only with the final location of the end-effecter, but also with the *velocity* at which the end-effecter moves. In order to move the end-effecter in a specified direction at a specified speed, it is necessary to *coordinate* the motion of the individual joints. The focus of this chapter is the development of fundamental methods for achieving such coordinated motion in multiple-joint robotic systems. As discussed in the previous chapter, the end-effecter position and orientation are directly related to the joint displacements; hence, in order to coordinate joint motions, we derive the *differential* relationship between the joint displacements and the end-effecter location, and then solve for the individual joint motions.

5.1 Differential Relationship

We begin by considering a two degree-of-freedom planar robot arm, as shown in Figure 5.1.1. The kinematic equations relating the end-effecter coordinates x_e and y_e to the joint displacements θ_1 and θ_2 are given by

$$x_e(\theta_1, \theta_2) = \ell_1 \cos\theta_1 + \ell_2 \cos(\theta_1 + \theta_2) \qquad (5.1.1)$$

$$y_e(\theta_1, \theta_2) = \ell_1 \sin\theta_1 + \ell_2 \sin(\theta_1 + \theta_2) \qquad (5.1.2)$$

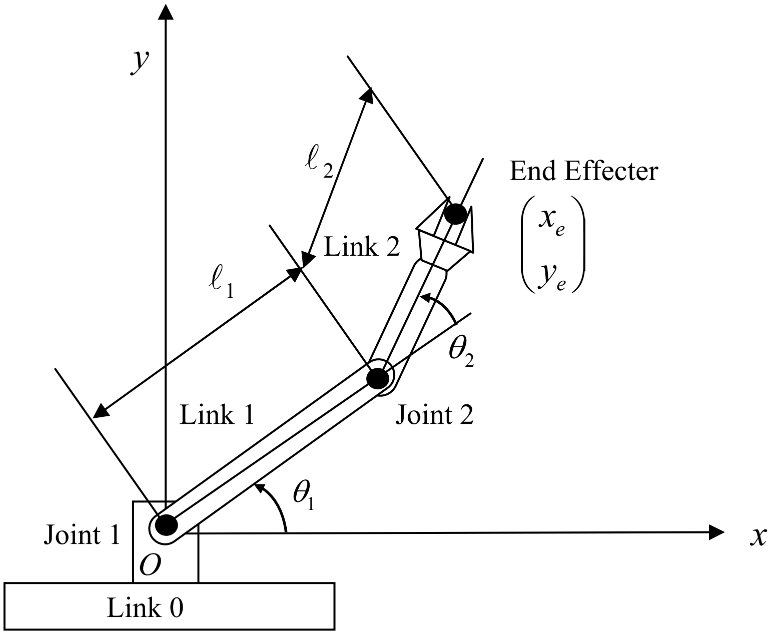


Figure 5.1.1 Two dof planar robot with two revolute joints

We are concerned with "small movements" of the individual joints at the current position, and we want to know the resultant motion of the end-effecter. This can be obtained by the total derivatives of the above kinematic equations:

$$dx_e = \frac{\partial x_e(\theta_1,\theta_2)}{\partial\theta_1}d\theta_1 + \frac{\partial x_e(\theta_1,\theta_2)}{\partial\theta_2}d\theta_2 \tag{5.1.3}$$

$$dy_e = \frac{\partial y_e(\theta_1,\theta_2)}{\partial\theta_1}d\theta_1 + \frac{\partial y_e(\theta_1,\theta_2)}{\partial\theta_2}d\theta_2 \tag{5.1.4}$$

where x_e, y_e are variables of both θ_1 and θ_2, hence two partial derivatives are involved in the total derivatives. In vector form the above equations reduce to

$$d\mathbf{x} = \mathbf{J}\cdot d\mathbf{q} \tag{5.1.5}$$

where

$$d\mathbf{x} = \begin{pmatrix} dx_e \\ dy_e \end{pmatrix}, \quad d\mathbf{q} = \begin{pmatrix} d\theta_1 \\ d\theta_2 \end{pmatrix} \tag{5.1.6}$$

and **J** is a 2 by 2 matrix given by

$$\mathbf{J} = \begin{pmatrix} \dfrac{\partial x_e(\theta_1,\theta_2)}{\partial\theta_1} & \dfrac{\partial x_e(\theta_1,\theta_2)}{\partial\theta_2} \\ \dfrac{\partial y_e(\theta_1,\theta_2)}{\partial\theta_1} & \dfrac{\partial y_e(\theta_1,\theta_2)}{\partial\theta_2} \end{pmatrix} \tag{5.1.7}$$

The matrix **J** comprises the partial derivatives of the functions $x_e(\theta_1,\theta_2)$ and $y_e(\theta_1,\theta_2)$ with respect to joint displacements θ_1 and θ_2. The matrix **J**, called the ***Jacobian Matrix,*** represents the differential relationship between the joint displacements and the resulting end-effecter motion. Note that most robot mechanisms have a multitude of active joints, hence a matrix is needed for describing the mapping of the vectorial joint motion to the vectorial end-effecter motion.

For the two-dof robot arm of Figure 5.1.1, the components of the Jacobian matrix are computed as

$$\mathbf{J} = \begin{pmatrix} -\ell_1\sin\theta_1 - \ell_2\sin(\theta_1+\theta_2) & -\ell_2\sin(\theta_1+\theta_2) \\ \ell_1\cos\theta_1 + \ell_2\cos(\theta_1+\theta_2) & \ell_2\cos(\theta_1+\theta_2) \end{pmatrix} \tag{5.1.8}$$

By definition, the Jacobian collectively represents the *sensitivities* of individual end-effecter coordinates to individual joint displacements. This sensitivity information is needed in order to coordinate the multi dof joint displacements for generating a desired motion at the end-effecter.

Consider the instant when the two joints of the robot arm are moving at joint velocities $\dot{\mathbf{q}} = (\dot{\theta}_1, \dot{\theta}_2)^T$, and let $\mathbf{v_e} = (\dot{x}_e, \dot{y}_e)^T$ be the resultant end-effecter velocity vector. The Jacobian provides the relationship between the joint velocities and the resultant end-effecter velocity. Indeed, dividing eq.(5) by the infinitesimal time increment *dt* yields

$$\frac{d\mathbf{x}_e}{dt} = \mathbf{J}\frac{d\mathbf{q}}{dt}, \quad \text{or} \quad \mathbf{v}_e = \mathbf{J}\cdot\dot{\mathbf{q}} \tag{5.1.9}$$

Thus the Jacobian determines the velocity relationship between the joints and the end-effecter.

5.2 Properties of the Jacobian

The Jacobian plays an important role in the analysis, design, and control of robotic systems. It will be used repeatedly in the following chapters. It is worth examining basic properties of the Jacobian, which will be used throughout this book.

We begin with dividing the 2-by-2 Jacobian of eq.(5.1.8) into two column vectors:

$$\mathbf{J} = (\mathbf{J}_1, \mathbf{J}_2), \quad \mathbf{J}_1, \mathbf{J}_2 \in \Re^{2\times 1} \tag{5.2.1}$$

Then eq.(5.1.9) can be written as

$$\mathbf{v}_e = \mathbf{J}_1 \cdot \dot{\theta}_1 + \mathbf{J}_2 \cdot \dot{\theta}_2 \tag{5.2.2}$$

The first term on the right-hand side accounts for the end-effecter velocity induced by the first joint only, while the second term represents the velocity resulting from the second joint motion only. The resultant end-effecter velocity is given by the vectorial sum of the two. Each column vector of the Jacobian matrix represents the end-effecter velocity generated by the corresponding joint moving at a unit velocity *when all other joints are immobilized.*

Figure 5.2.1 illustrates the column vectors $\mathbf{J}_1, \mathbf{J}_2$ of the 2 dof robot arm in the two-dimensional space. Vector $\mathbf{J}_2$, given by the second column of eq.(5.1. 8), points in the direction perpendicular to link 2. Note, however, that vector $\mathbf{J}_1$ is not perpendicular to link 1 but is perpendicular to line *OE*, the line from joint 1 to the endpoint *E.* This is because $\mathbf{J}_1$ represents the endpoint velocity induced by joint 1 when joint 2 is immobilized. In other words, links 1 and 2 are rigidly connected, becoming a single rigid body of link length *OE,* and $\mathbf{J}_1$ is the tip velocity of the link *OE.*

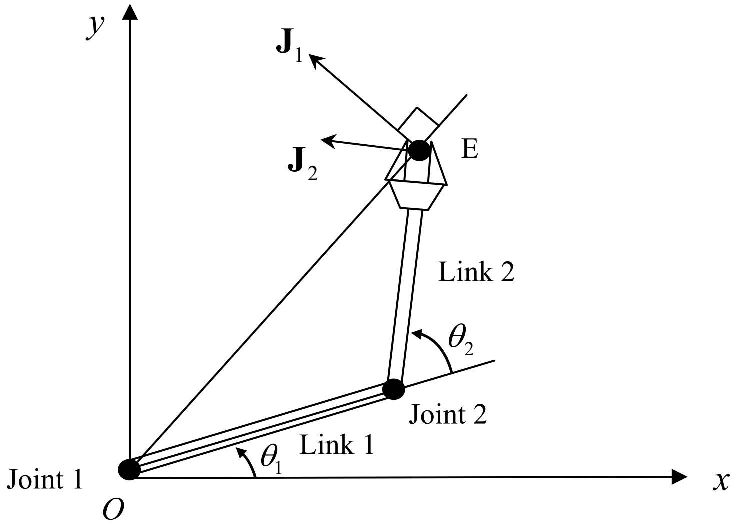


Figure 5.2.1 Geometric interpretation of the column vectors of the Jacobian

In general, each column vector of the Jacobian represents the end-effecter velocity and angular velocity generated by the individual joint velocity while all other joints are immobilized. Let $\dot{\mathbf{p}}$ be the end-effecter velocity and angular velocity, or the end-effecter velocity for short, and

$\mathbf{J}_i$ be the *i-th* column of the Jacobian. The end-effecter velocity is given by a linear combination of the Jacobian column vectors weighted by the individual joint velocities.

$$\dot{\mathbf{p}} = \mathbf{J}_1 \cdot \dot{q}_1 + \cdots + \mathbf{J}_n \cdot \dot{q}_n \tag{5.2.3}$$

where n is the number of active joints. The geometric interpretation of the column vectors is that $\mathbf{J}_i$ is the end-effecter velocity and angular velocity when all the joints other than joint i are immobilized and only *i-th* joint is moving at a unit velocity.

Exercise Consider the two-dof articulated robot shown in Figure 5.2.1 again. This time we use "absolute" joint angles measured from the positive *x*-axis, as shown in Figure 5.2.2. Note that angle θ_2 is measured from the fixed frame, i.e. the *x*-axis, rather than a relative frame, e.g. link 1. Obtain the 2-by-2 Jacobian and illustrate the two column vectors on the *xy* plane. Discuss the result in comparison with the previous case shown in Figure 5.2.1.

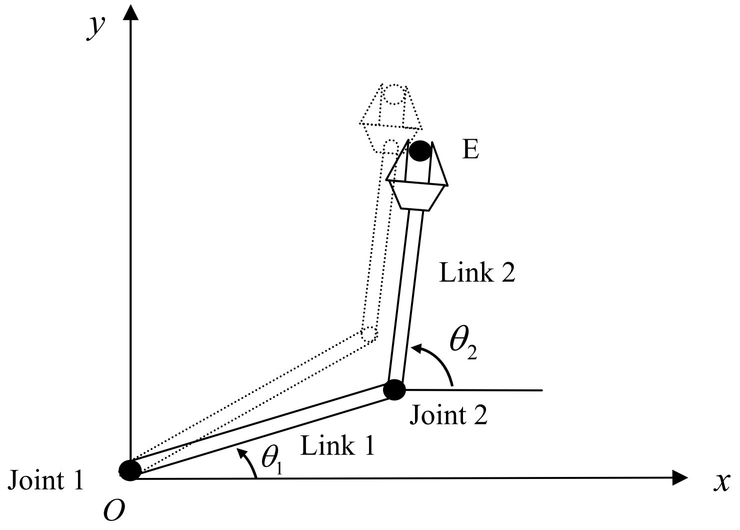


Figure 5.2.2 Absolute joint angles measured from the *x*-axis.

Note that the elements of the Jacobian are functions of joint displacements, and thereby vary with the arm configuration. As expressed in eq.(5.1.8), the partial derivatives, $\partial x_e / \partial \theta_i$, $\partial y_e / \partial \theta_i$, are functions of θ_1 and θ_2. Therefore, the column vectors $\mathbf{J}_1$, $\mathbf{J}_2$ vary depending on the arm posture. Remember that the end-effecter velocity is given by the linear combination of the Jacobian column vectors $\mathbf{J}_1$, $\mathbf{J}_2$. Therefore, the resultant end-effecter velocity varies depending on the direction and magnitude of the Jacobian column vectors $\mathbf{J}_1$, $\mathbf{J}_2$ spanning the two dimensional space. If the two vectors point in different directions, the whole two-dimensional space is covered with the linear combination of the two vectors. That is, the end-effecter can be moved in an arbitrary direction with an arbitrary velocity. If, on the other hand, the two Jacobian column vectors are aligned, the end-effecter cannot be moved in an arbitrary direction. As shown in Figure 5.2.3, this may happen for particular arm postures where the two links are fully contracted or extended. These arm configurations are referred to as singular configurations. Accordingly, the Jacobian matrix becomes ***singular*** at these singular configurations. Using the determinant of a matrix, this condition is expressed as

$$\det \mathbf{J} = 0 \tag{5.2.4}$$

In fact, the Jacobian degenerates at the singular configurations, where joint 2 is 0 or 180 degrees. Substituting $\theta_2 = 0, \pi$ into eq.(5.1.8) yields

$$\det \mathbf{J} = \begin{pmatrix} -(\ell_1 \pm \ell_2)\sin\theta_1 & \mp \ell_2 \sin\theta_1 \\ (\ell_1 \pm \ell_2)\cos\theta_1 & \pm \ell_2 \cos\theta_1 \end{pmatrix} = 0 \qquad (5.2.5)$$

Note that both column vectors point in the same direction and thereby the determinant becomes zero.

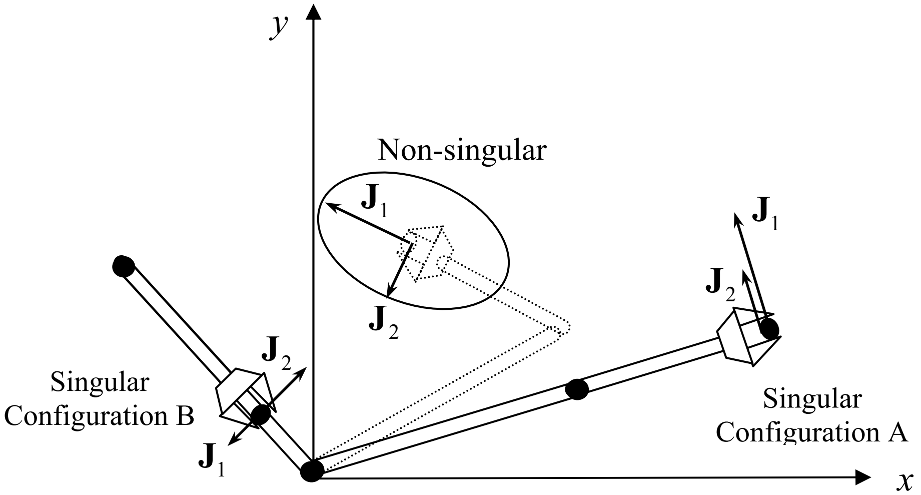


Figure 5.2.3 Singular configurations of the two-dof articulated robot

5.3 Inverse Kinematics of Differential Motion

Now that we know the basic properties of the Jacobian, we are ready to formulate the inverse kinematics problem for obtaining the joint velocities that allow the end-effecter to move at a given desired velocity. For the two dof articulated robot, the problem is to find the joint velocities $\dot{\mathbf{q}} = (\dot{\theta}_1, \dot{\theta}_2)^T$, for the given end-effecter velocity $\mathbf{v}_\mathbf{e} = (v_x, v_y)^T$. If the arm configuration is not singular, this can be obtained by taking the inverse to the Jacobian matrix in eq.(5.1.9),

$$\dot{\mathbf{q}} = \mathbf{J}^{-1} \cdot \mathbf{v}_\mathbf{e} \qquad (5.3.1)$$

Note that the solution is unique. Unlike the inverse kinematics problem discussed in the previous chapter, the *differential* kinematics problem has a unique solution as long as the Jacobian is non-singular.

The above solution determines how the end-effecter velocity $\mathbf{v}_\mathrm{e}$ must be decomposed, or *resolved*, to individual joint velocities. If the controls of the individual joints regulate the joint velocities so that they can track the resolved joint velocities $\dot{\mathbf{q}}$, the resultant end-effecter velocity will be the desired one $\mathbf{v}_\mathrm{e}$. This control scheme is called Resolved Motion Rate Control, attributed to Daniel Whitney (1969). Since the elements of the Jacobian matrix are functions of joint displacements, the inverse Jacobian varies depending on the arm configuration. This means that although the desired end-effecter velocity is constant, the joint velocities are not. Coordination is thus needed among the joint velocity control systems in order to generate a desired motion at the end-effecter.

Example Consider the two dof articulated robot arm again. We want to move the endpoint of the robot at a constant speed along a path staring at point *A* on the *x*-axis, (+2, 0), go around the origin through points *B* (+ε, 0) and *C* (0, +ε), and reach the final point *D* (0, +2) on the *y*-axis. See Figure 5.3.1. For simplicity each arm link is of unit length. Obtain the profiles of the individual joint velocities as the end-effecter tracks the path at the constant speed.

Substituting $\mathbf{v_e} = (v_x, v_y)^T$ into eq.(1) yields

$$\dot{\theta}_1 = \frac{v_x \cos(\theta_1 + \theta_2) + v_y \sin(\theta_1 + \theta_2)}{\sin\theta_2} \tag{5.3.2}$$

$$\dot{\theta}_2 = \frac{v_x[\cos\theta_1 + \cos(\theta_1 + \theta_2)] + v_y[\sin\theta_1 + \sin(\theta_1 + \theta_2)]}{\sin\theta_2} \tag{5.3.3}$$

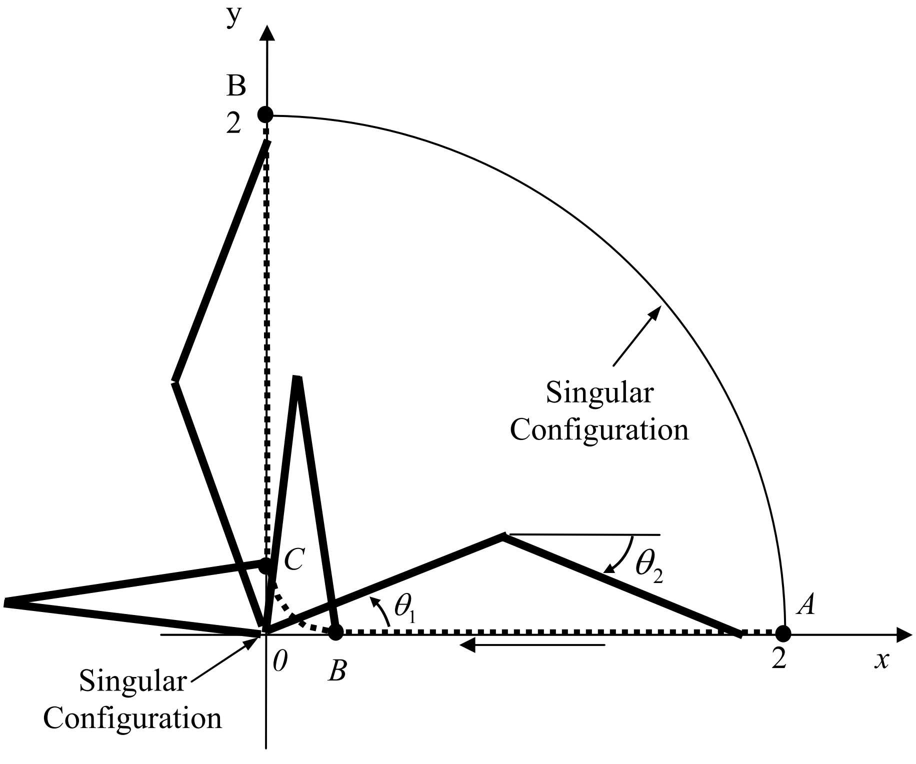


Figure 5.3.1 trajectory tracking near the singular points

Figure 5.3.2 shows the resolved joint velocities $\dot{\theta}_1, \dot{\theta}_2$ computed along the specified trajectory. Note that the joint velocities are extremely large near the initial and final points, and are unbounded at points A and D. These are at the arm's singular configurations, $\theta_2 = 0$. As the end-effecter gets close to the origin, the velocity of the first joint becomes very large in order to quickly turn the arm around from point B to C. At these configurations, the second joint is almost –180 degrees, meaning that the arm is near singularity. This result agrees with the singularity condition using the determinant of the Jacobian:

$$\det \mathbf{J} = \sin\theta_2 = 0, \quad \therefore \theta_2 = k\pi, \quad k = 0, \pm 1, \pm 2, \cdots \tag{5.3.4}$$

In eqs.(2) and (3) above, the numerators are divided by $\sin\theta_2$, the determinant of the Jacobian. Therefore, the joint velocities $\dot{\theta}_1, \dot{\theta}_2$ blow out as the arm configuration gets close to the singular configuration.

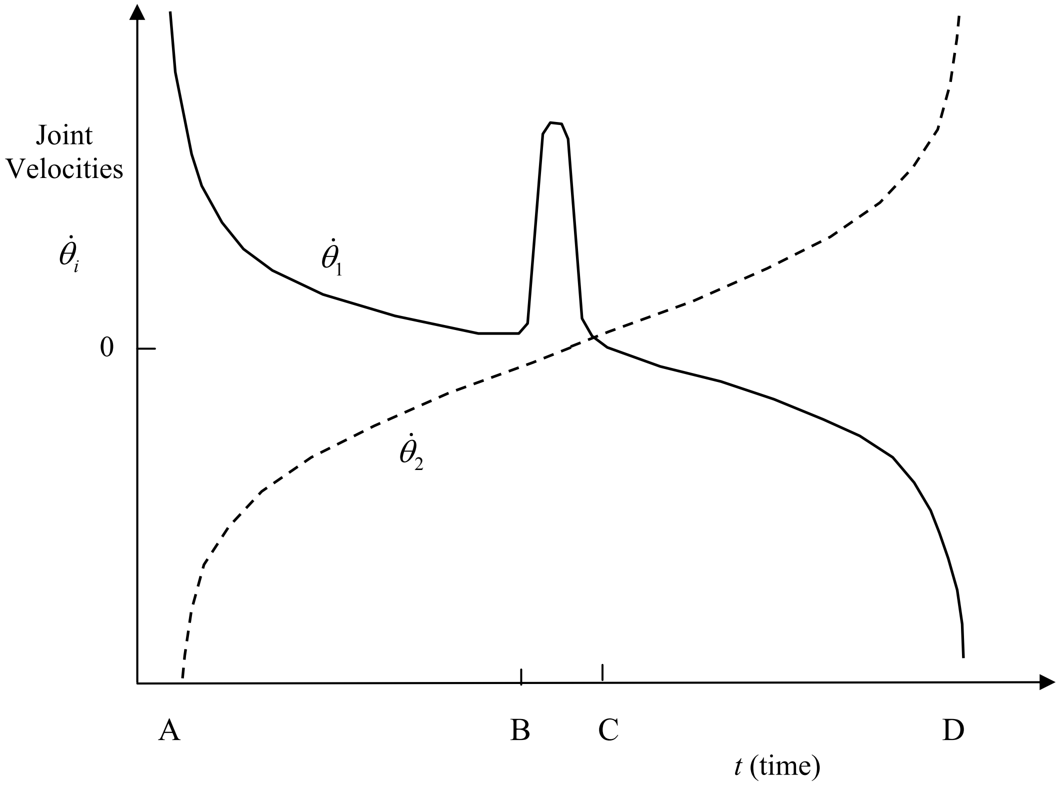


Figure 5.3.2 Joint velocity profiles for tracking the trajectory in Figure 5.3.1

Furthermore, the arm's behavior near the singular points can be analyzed by substituting $\theta_2 = 0, \pi$ into the Jacobian, as obtained in eq.(5.2.5). For $\ell_1 = \ell_2 = 1$ and $\theta_2 = 0$, the Jacobian column vectors reduce to the ones in the same direction:

$$\mathbf{J}_1 = \begin{pmatrix} -2\sin\theta_1 \\ 2\cos\theta_1 \end{pmatrix}, \ \mathbf{J}_2 = \begin{pmatrix} -\sin\theta_1 \\ \cos\theta_1 \end{pmatrix}, \quad for\ \theta_2 = 0 \tag{5.3.5}$$

As illustrated in Figure 5.2.3 (singular configuration A), both joints $\dot{\theta}_1, \dot{\theta}_2$ generate the endpoint velocity along the same direction. Note that no endpoint velocity can be generated in the direction perpendicular to the aligned arm links. For $\theta_2 = \pi$,

$$\mathbf{J}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{J}_2 = \begin{pmatrix} \sin\theta_1 \\ -\cos\theta_1 \end{pmatrix}, \quad for\ \theta_2 = \pi \tag{5.3.6}$$

The first joint cannot generate any endpoint velocity, since the arm is fully contracted. See singular configuration B in Figure 5.2.3.

At a singular configuration, there is at least one direction in which the robot cannot generate a non-zero velocity at the end-effecter. This agrees with the previous discussion; the Jacobian is degenerate at a singular configuration, and the linear combination of the Jacobian column vectors cannot span the whole space.

Exercise 5.2

A three-dof spatial robot arm is shown in the figure below. The robot has three revolute joints that allow the endpoint to move in the three dimensional space. However, this robot mechanism has singular points inside the workspace. Analyze the singularity, following the procedure below.

Step 1 Obtain each column vector of the Jacobian matrix by considering the endpoint velocity created by each of the joints while immobilizing the other joints.
Step 2 Construct the Jacobian by concatenating the column vectors, and set the determinant of the Jacobian to zero for singularity: $\det \mathbf{J} = 0$.
Step 3 Find the joint angles that make $\det \mathbf{J} = 0$.
Step 4 Show the arm posture that is singular. Show where in the workspace it becomes singular. For each singular configuration also show in which direction the endpoint cannot have a non-zero velocity.

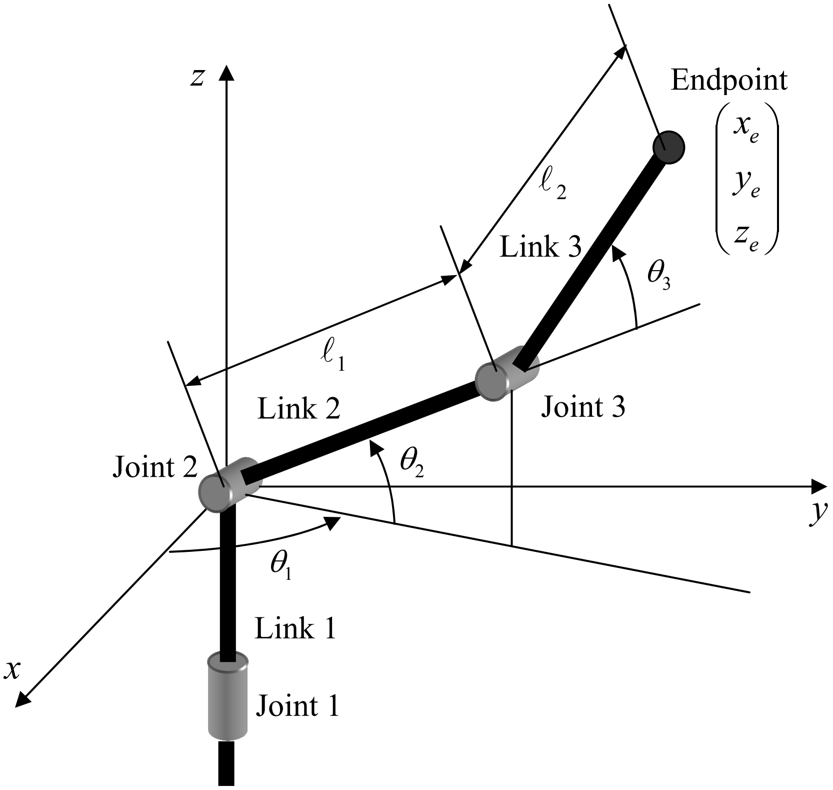


Figure 5.3.3 Schematic of a three dof articulated robot

5.4 Singularity and Redundancy

We have seen in this chapter that singular configurations exist for many robot mechanisms. Sometimes such singular configurations exist in the middle of the workspace, seriously degrading mobility and dexterity of the robot. At a singular point the robot cannot move in certain directions with a non-zero velocity. To overcome this difficulty, several methods can be considered. One is to plan a trajectory of the robot motion such that it will not go into singular configurations. Other method is to have more degrees of freedom, so that even when some degrees of freedom are lost at a certain configuration, the robot can maintain necessary degrees of freedom. Such a robot is referred to as a redundant robot. In this section we will discuss singularity and redundancy, and obtain general properties of differential motion for general *n* degrees of freedom robots.

As studied in Section 5.3, a unique solution exists to the differential kinematic equation, (5.1.15), if the arm configuration is non-singular. However, when a planar (spatial) robot arm has more than three (six) degrees of freedom, we can find an infinite number of solutions that provide the same motion at the end-effecter. Consider for instance the human arm, which has seven degrees of freedom excluding the joints at the fingers. When the hand is placed on a desk and fixed in its position and orientation, the elbow position can still vary continuously without moving the hand. This implies that a certain ratio of joint velocities exists that does not cause any velocity at the hand. This specific ratio of joint velocities does not contribute to the resultant endpoint velocity. Even if these joint velocities are superimposed to other joint velocities, the resultant end-effecter velocity is the same. Consequently, we can find different solutions of the instantaneous kinematic equation for the same end-effecter velocity. In the following, we investigate the fundamental properties of the differential kinematics when additional degrees of freedom are involved.

To formulate a differential kinematic equation for a general *n* degrees-of-freedom robot mechanism, we begin by modifying the definition of the vector $d\mathbf{x}_\mathbf{e}$ representing the end-effecter motion. In eq. (5.1.6), $d\mathbf{x}_\mathbf{e}$ was defined as a two-dimensional vector that represents the infinitesimal translation of an end-effecter. This must be extended to a general *m*-dimensional vector. For planar motion, *m* may be 3, and for spatial motion, *m* may be six. However, the number of variables that we need to specify for performing a task is not necessarily three or six. In arc welding, for example, only five independent variables of torch motion need be controlled. Since the welding torch is usually symmetric about its centerline, we can locate the torch with an arbitrary orientation about the centerline. Thus five degrees of freedom are sufficient to perform arc welding. In general, we describe the differential end-effecter motion by *m* independent variables $d\mathbf{p}$ that must be specified to perform a given task.

$$d\mathbf{p} = \begin{bmatrix} dp_1 & dp_2 & \cdots & dp_m \end{bmatrix}^T \in \Re^{m\times 1} \tag{5.4.1}$$

Then the differential kinematic equation for a general *n* degree-of-freedom robot is given by

$$d\mathbf{p} = \mathbf{J}\cdot d\mathbf{q} \tag{5.4.2}$$

where the dimension of the Jacobian **J** is *m* by *n*; $\mathbf{J}\in\Re^{m\times n}$. When *n* is larger than *m* and **J** is of full rank, there are *(n-m)* arbitrary variables in the general solution of eq.(2). The robot is then said to have *(n-m) redundant degrees of freedom* for the given task.

Associated with the above differential equation, the velocity relationship can be written as

$$\dot{\mathbf{p}} = \mathbf{J}\cdot\dot{\mathbf{q}} \tag{5.4.3}$$

where $\dot{\mathbf{p}}$ and $\dot{\mathbf{q}}$ are velocities of the end effecter and the joints, respectively.

Equation (3) can be regarded as a linear mapping from n-dimensional vector space $\boldsymbol{V^n}$ to m-dimensional space $\boldsymbol{V^m}$. To characterize the solution to eq.(3), let us interpret the equation using the linear mapping diagram shown in Figure 5.4.1. The subspace $R(\mathbf{J})$ in the figure is the range space of the linear mapping. The range space represents all the possible end-effecter velocities that can be generated by the n joints at the present arm configuration. If the rank of **J** is of full row rank, the range space covers the entire vector space $\boldsymbol{V}^m$. Otherwise, there exists at least one direction in which the end-effecter cannot be moved with non-zero velocity. The subspace $N(\mathbf{J})$ of Figure 5.4.1 is the null space of the linear mapping. Any element in this subspace is mapped into the zero vector in $\boldsymbol{V}^m$. Therefore, any joint velocity vector $\dot{\mathbf{q}}$ that belongs to the null space does not produce any velocity at the end-effecter. Recall the human arm discussed before. The elbow can move without moving the hand. Joint velocities for this motion are involved in the null space, since no end-effecter motion is induced. If the Jacobian is of full rank, the dimension of the null space, dim N(J), is the same as the number of redundant degrees of freedom *(n-m).* When the Jacobian matrix is degenerate, i.e. not of full rank, the dimension of the range space, dim R(J), decreases, and at the same time the dimension of the null space increases by the same amount. The sum of the two is always equal to *n:*

$$\dim R(\mathbf{J}) + \dim N(\mathbf{R}) = n \tag{5.4.4}$$

Let $\dot{\mathbf{q}}\,*$ be a particular solution of eq.(3) and $\dot{\mathbf{q}}_0$ be a vector involved in the null space, then the vector of the form $\dot{\mathbf{q}} = \dot{\mathbf{q}}\,* + k\,\dot{\mathbf{q}}_0$ is also a solution of eq.(3), where k is an arbitrary scalar quantity. Namely,

$$\mathbf{J}\dot{\mathbf{q}} = \mathbf{J}\dot{\mathbf{q}}\,* + k\mathbf{J}\dot{\mathbf{q}}_0 = \mathbf{J}\dot{\mathbf{q}}* = \dot{\mathbf{p}} \tag{5.4.5}$$

Since the second term $k\dot{\mathbf{q}}_0$ can be chosen arbitrarily within the null space, an infinite number of solutions exist for the linear equation, unless the dimension of the null space is 0. The null space accounts for the arbitrariness of the solutions. The general solution to the linear equation involves the same number of arbitrary parameters as the dimension of the null space.

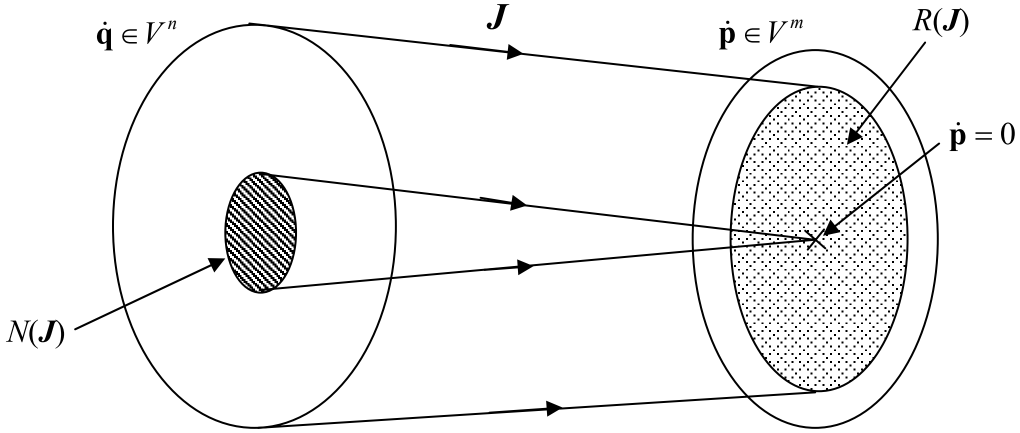


Figure 5.4.1 Linear mapping diagram

Chapter 6
Statics

Robots physically interact with the environment through mechanical contacts. Mating work pieces in a robotic assembly line, manipulating an object with a multi-fingered hand, and negotiating a rough terrain in leg locomotion are just a few examples of mechanical interactions. All of these tasks entail controls of the contacts and interference between the robot and the environment. Force and moment acting between the robot end-effecter and the environment must be accommodated in order to control the interactions. In this chapter we will analyze the force and moment that act on the robot when it is at rest.

A robot generates a force and a moment at its end-effecter by controlling individual actuators. To generate a desired force and moment the torques of the multiple actuators must be coordinated. As seen in the previous chapter, the sensitivities of the individual actuators upon the end-effecter motion, i.e. the Jacobian matrix, are essential in relating the actuator (joint) torques to the force and moment at the end-effecter. We will obtain a fundamental theorem of force and moment acting on a multi degree-of-freedom robot, which we will find is analogous to the differential kinematics discussed previously.

6.1 Free Body Diagram

We begin by considering the free body diagram of an individual link involved in an open kinematic chain. Figure 6.1.1 shows the forces and moments acting on link *i,* which is connected to link *i-1* and link *i+1* by joints *i-1* and *i+1*, respectively. Let O_i be a point fixed to link *i* located on the joint axis *i+1* and O_{i-1} be a point fixed to link *i-1* on the joint axis *i*. Through the connections with the adjacent links, link *i* receives forces and moments from both sides of links. Let $\boldsymbol{f}_{i-1,i}$ be a three-dimensional vector representing the linear force acting from link *i-1* to link *i*. Likewise let $\boldsymbol{f}_{i,i+1}$ be the force from link *i* to link *i+1*. The force applied to link *i* from link *i+1* is then given by $-\boldsymbol{f}_{i+1,i}$. The gravity force acting at the mass centroid C_i is denoted $m_i\boldsymbol{g}$, where m_i is the mass of link *i* and $\boldsymbol{g}$ is the 3x1 vector representing the acceleration of gravity. The balance of linear forces is then given by

$$\mathbf{f}_{i-1,i} - \mathbf{f}_{i,i+1} + m_i\mathbf{g} = \mathbf{0}, \quad i = 1, \cdots, n \tag{6.1.1}$$

Note that all the vectors are defined with respect to the base coordinate system *O-xyz.*

Next we derive the balance of moments. The moment applied to link *i* by link *i-1*is denoted $\boldsymbol{N}_{i-1,i}$, and therefore the moment applied to link *i* by link *i+1* is $-\boldsymbol{N}_{i,i+1}$. Furthermore, the linear forces $\boldsymbol{f}_{i-1,i}$ and $-\boldsymbol{f}_{i+1,i}$ also cause moments about the centroid C_i. The balance of moments with respect to the centroid C_i is thus given by

$$\mathbf{N}_{i-1,i} - \mathbf{N}_{i,i+1} - (\mathbf{r}_{i-1,i} + \mathbf{r}_{i,Ci}) \times \mathbf{f}_{i-1,i} + (-\mathbf{r}_{i,Ci}) \times (-\mathbf{f}_{i,i+1}) = \mathbf{0}, \quad i = 1, \cdots, n \tag{6.1.2}$$

where $\boldsymbol{r}_{i-1,i}$ is the 3x1 position vector from point O_{i-1} to point O_i with reference to the base coordinate frame, and $\boldsymbol{r}_{i,Ci}$ represents the position vector from point O_i to the centroid C_i.

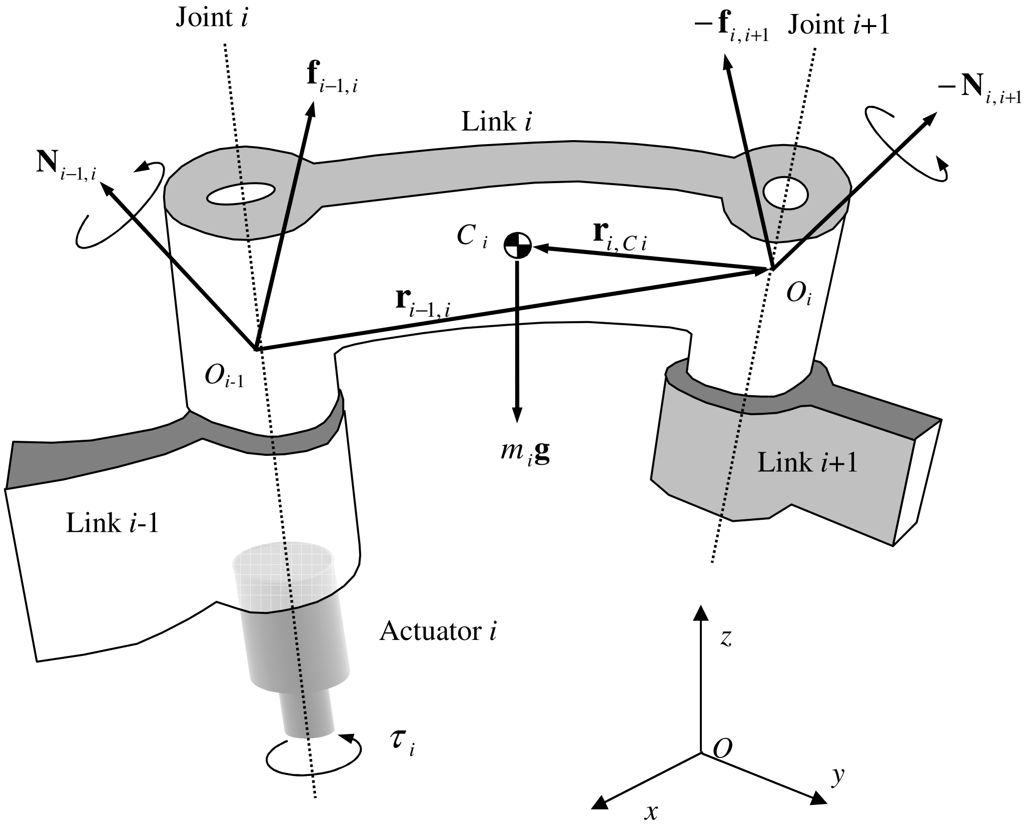


Figure 6.1.1 Free body diagram of the *i-th* link

The force $\mathbf{f}_{i-1,i}$ and moment $\mathbf{N}_{i-1,i}$ are called the coupling force and moment between the adjacent links *i* and *i*-1. For *i*=1, the coupling force and moment are $\mathbf{f}_{0,1}$ and $\mathbf{N}_{0,1}$. These are interpreted as the reaction force and moment applied to the base link to which the arm mechanism is fixed. See Figure 6.1.2-(a). When $i = n,$ on the other hand, the above coupling force and moment become $\mathbf{f}_{n,n+1}$ and $\mathbf{N}_{n,n+1}$. As the end-effecter, i.e. link *n* contacts the environment, the reaction force acts on the end-effecter. See Figure 6.1.2-(b). For convenience, we regard the environment as an additional link, numbered *n*+1, and represent the reaction force and moment by $\text{-}\mathbf{f}_{n,n+1}$ and $\text{-}\mathbf{N}_{n,n+1}$, respectively.

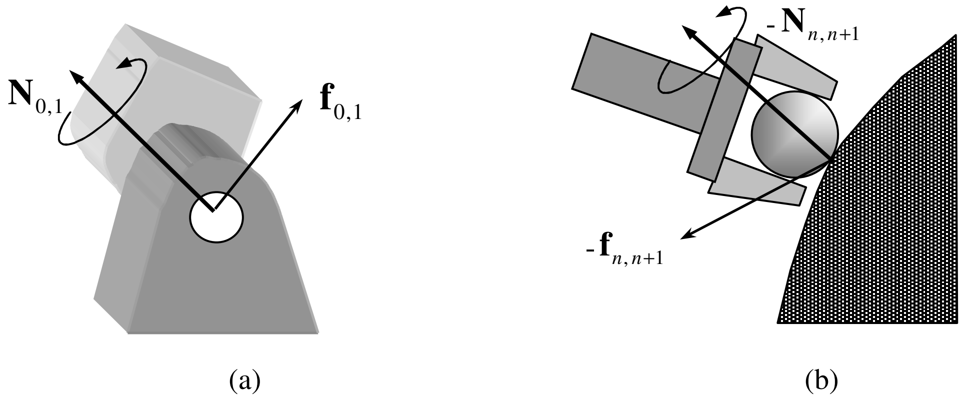


Figure 6.1.2 Force and moment that the base link exerts on link 1 (a), and the ones that the environment exerts on the end-effecter, the final link

The above equations can be derived for all the link members except for the base link, i.e. i=1,2, …, n. This allows us to form 2n simultaneous equations of 3x1 vectors. The number of coupling forces and moments involved is 2(n+1). Therefore two of the coupling forces and moments must be specified; otherwise the equations cannot be solved. The final coupling force and moment, $\mathbf{f}_{n,n+1}$ and $\mathbf{N}_{n,n+1}$, are the force and moment that the end-effecter applies to the environment. It is this pair of force and moment that the robot needs to accommodate in order to perform a given task. Thus, we specify this pair of coupling force and moment, and solve the simultaneous equations. For convenience we combine the force $\mathbf{f}_{n,n+1}$ and the moment $\mathbf{N}_{n,n+1}$, to define the following six-dimensional vector:

$$\mathbf{F} = \begin{pmatrix} \mathbf{f}_{n,n+1} \\ \mathbf{N}_{n,n+1} \end{pmatrix} \tag{6.1.3}$$

We call the vector **F** the endpoint force and moment vector, or the endpoint force for short.

6.2 Energy Method and Equivalent Joint Torques

In this section we will obtain the functional relationship between the joint torques and the endpoint force, which will be needed for accommodating interactions between the end-effecter and the environment. Such a functional relationship may be obtained by solving the simultaneous equations derived from the free body diagram. However, we will use a different methodology, which will give an explicit formula relating the joint torques to the endpoint force without solving the simultaneous equations. The methodology we will use is ***energy method***, or sometimes it is called ***indirect method.*** Since the simultaneous equations based on the balance of forces and moments are complex and difficult to solve, we will find that the energy method is the right choice for dealing with complex robotic systems.

In energy method, we describe a system with respect to energy and work. Therefore, terms associated with forces and moments that do not produce, store, or dissipate energy are eliminated in its basic formula. In the free body diagram shown in Figure 6.1.1, many components of the forces and moments are so called "constraint forces and moments" merely joining adjacent links together. Therefore, constraint forces and moments do not participate in energy formulation. This significantly reduces the number of terms and, more importantly, will provide an explicit formula relating the joint torques to the endpoint force.

To apply the energy method, two preliminary formulations must be performed. One is to separate the net force or moment generating energy from the constraint forces and moments irrelevant to energy. Second, we need to find independent displacement variables that are geometrically admissible satisfying kinematic relations among the links.

Figure 6.2.1 shows the actuator torques and the coupling forces and moments acting at adjacent joints. The coupling force $\mathbf{f}_{i-1,i}$ and moment $\mathbf{N}_{i-1,i}$ are the resultant force and moment acting on the individual joint, comprising the constraint force and moment as well as the torque generated by the actuator. Let $\mathbf{b}_{i-1}$ be the 3x1 unit vector pointing in the direction of joint axis i, as shown in the figure. If the i-th joint is a revolute joint, the actuator generates joint torque τ_i about the joint axis. Therefore, the joint torque generated by the actuator is one component of the coupling moment $\mathbf{N}_{i-1,i}$ along the direction of the joint axis:

$$\tau_i = \mathbf{b}_{i-1}{}^T \cdot \mathbf{N}_{i-1,i} \tag{6.2.1}$$

For a prismatic joint, such as the (j+1)-st joint illustrated in Figure 6.2.1, the actuator generates a linear force in the direction of the joint axis. Therefore, it is the component of the linear coupling force $\mathbf{f}_{i-1,i}$ projected onto the joint axis.

$$\tau_i = \mathbf{b}_{i-1}{}^T \cdot \mathbf{f}_{i-1,i} \tag{6.2.2}$$

Note that, although we use the same notation as that of a revolute joint, the scalar quantity τ_i has the unit of a linear force for a prismatic joint. To unify the notation we use τ_i for both types of joints, and call it a joint torque regardless the type of joint.

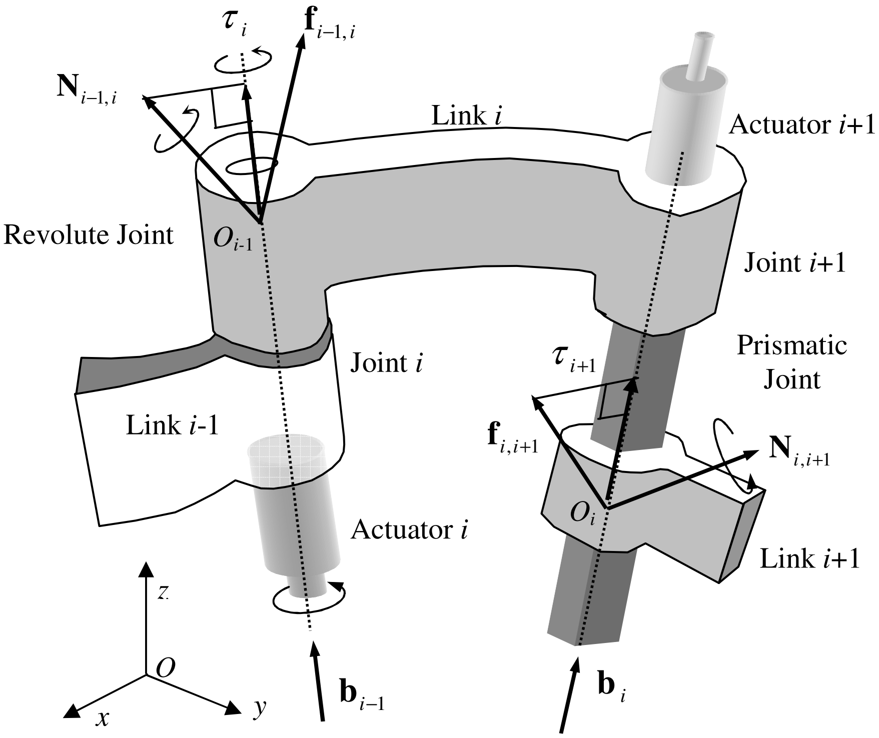


Figure 6.2.1 Joint torques as components of coupling force and moment

We combine all the joint toques from joint 1 through joint n to define the nx1 joint torque vector:

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_1 & \tau_2 & \cdots & \tau_n \end{pmatrix}^T \tag{6.2.3}$$

The joint torque vector collectively represents all the actuators' torque inputs to the linkage system. Note that all the other components of the coupling force and moment are borne by the mechanical structure of the joint. Therefore, the constraint forces and moments irrelevant to energy formula have been separated from the net energy inputs to the linkage system.

In the free body diagram the individual links are *disjointed*, leaving constraint forces and moments at both sides of the link. The freed links are allowed to move in any direction. In the energy formulation, we describe the link motion using independent variables alone. Remember that in a serial link open kinematic chain joint coordinates $\mathbf{q} = \begin{pmatrix} q_1 & \cdots & q_n \end{pmatrix}^T$ are a complete and independent set of generalized coordinates that uniquely locate the linkage system with independent variables. Therefore, these variables conform to the geometric and kinematic constraints. We use these joint coordinates in the energy-based formulation.

The explicit relationship between the n joint torques and the endpoint force **F** is given by the following theorem:

Theorem 6.1

Consider a n degree-of-freedom, serial link robot having no friction at the joints. The joint torques $\boldsymbol{\tau} \in \Re^{n \times 1}$ that are required for bearing an arbitrary endpoint force $\mathbf{F} \in \Re^{6x1}$ are given by

$$\boldsymbol{\tau} = \mathbf{J}^T \cdot \mathbf{F} \tag{6.2.4}$$

where **J** is the 6 x n Jacobian matrix relating infinitesimal joint displacements $d\mathbf{q}$ to infinitesimal end-effecter displacements $d\mathbf{p}$**:**

$$d\mathbf{p} = \mathbf{J} \cdot d\mathbf{q} \tag{6.2.5}$$

□

Note that the joint torques in the above expression do not account for gravity and friction. They are the *net* torques that balances the endpoint force and moment. We call $\boldsymbol{\tau}$ of eq.(2) the equivalent joint torques associated with the endpoint force **F**.

Proof

We prove the theorem by using the *Principle of Virtual Work*. Consider virtual displacements at individual joints, $\delta\mathbf{q} = (\delta q_1, \cdots, \delta q_n)^T$, and at the end-effecter, $\delta\mathbf{p} = (\delta\mathbf{x}_e{}^T, \delta\boldsymbol{\varphi}_e{}^T)^T$, as shown in Figure 6.2.2. Virtual displacements are arbitrary infinitesimal displacements of a mechanical system that conform to the geometric constraints of the system. Virtual displacements are different from actual displacements, in that they must only satisfy *geometric* constraints and do not have to meet other laws of motion. To distinguish the virtual displacements from the actual displacements, we use the Greek letter δ rather than the roman *d.* We assume that joint torques $\boldsymbol{\tau} = \begin{pmatrix}\tau_1 & \tau_2 & \cdots & \tau_n\end{pmatrix}^T$ and endpoint force and moment, **-F,** act on the serial linkage system, while the joints and the end-effecter are moved in the directions geometrically admissible. Then, the virtual work done by the forces and moments is given by

$$\begin{aligned} \delta Work &= \tau_1 \cdot \delta q_1 + \tau_2 \cdot \delta q_2 + \cdots + \tau_n \cdot \delta q_n - \mathbf{f}_{n,n+1}{}^T \cdot \delta\mathbf{x}_e - \mathbf{N}_{n,n+1}{}^T \cdot \delta\boldsymbol{\varphi}_e \\ &= \boldsymbol{\tau}^T \delta\mathbf{q} - \mathbf{F}^T \delta\mathbf{p} \end{aligned} \tag{6.2.6}$$

According to the principle of virtual work, the linkage system is in equilibrium if, and only if, the virtual work $\delta Work$ vanishes for arbitrary virtual displacements that conform to geometric constraints. Note that the virtual displacements $\delta\boldsymbol{q}$ and $\delta\boldsymbol{p}$ are not independent, but are related by the Jacobian matrix given in eq.(5). The kinematic structure of the robot mechanism dictates that the virtual displacements $\delta\boldsymbol{p}$ is completely dependent upon the virtual displacement of the joints, $\delta\boldsymbol{q}$**.** Substituting eq.(5) into eq.(6) yields

$$\delta Work = \boldsymbol{\tau}^T \delta\mathbf{q} - \mathbf{F}^T \mathbf{J} \cdot \delta\mathbf{q} = (\boldsymbol{\tau} - \mathbf{J}^T \mathbf{F})^T \cdot \delta\mathbf{q} \tag{6.2.7}$$

Note that the vector of the virtual displacements $\delta\boldsymbol{q}$ consists of all independent variables, since the joint coordinates of an open kinematic chain are generalized coordinates that are complete and independent. Therefore, for the above virtual work to vanish for arbitrary virtual displacements we must have:

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{F}$$

This is eq.(6.2.4), and the theorem has been proven.

□

The above theorem has broad applications in robot mechanics, design, and control. We will use it repeatedly in the following chapters.

Example 6.1

Figure 6.2.1 shows a two-dof articulated robot having the same link dimensions as the previous examples. The robot is interacting with the environment surface in a horizontal plane. Obtain the equivalent joint torques $\boldsymbol{\tau} = (\tau_1, \tau_2)^T$ needed for pushing the surface with an endpoint force of $\mathbf{F} = (F_x, F_y)^T$. Assume no friction.

The Jacobian matrix relating the end-effecter coordinates x_e and y_e to the joint displacements θ_1 and θ_2 has been obtained in the previous chapter:

$$\mathbf{J} = \begin{pmatrix} -\ell_1 \sin\theta_1 - \ell_2 \sin(\theta_1 + \theta_2) & -\ell_2 \sin(\theta_1 + \theta_2) \\ \ell_1 \cos\theta_1 + \ell_2 \cos(\theta_1 + \theta_2) & \ell_2 \cos(\theta_1 + \theta_2) \end{pmatrix} \qquad (5.1.8)$$

From ***Theorem 6.1*** the equivalent joint torques are obtained by simply taking the transpose of the Jacobian matrix.

$$\begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} = \begin{pmatrix} -\ell_1 \sin\theta_1 - \ell_2 \sin(\theta_1 + \theta_2) & \ell_1 \cos\theta_1 + \ell_2 \cos(\theta_1 + \theta_2) \\ -\ell_2 \sin(\theta_1 + \theta_2) & \ell_2 \cos(\theta_1 + \theta_2) \end{pmatrix} \cdot \begin{pmatrix} F_x \\ F_y \end{pmatrix} \qquad (6.2.8)$$

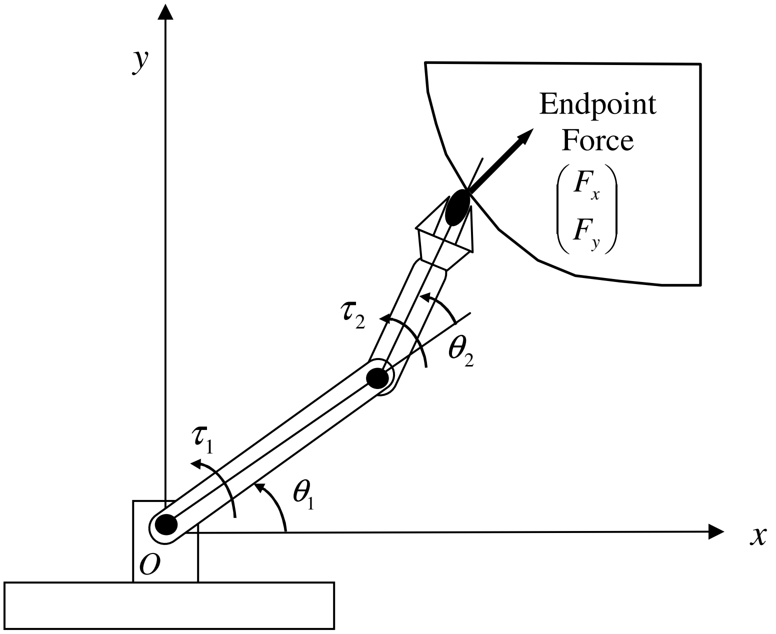
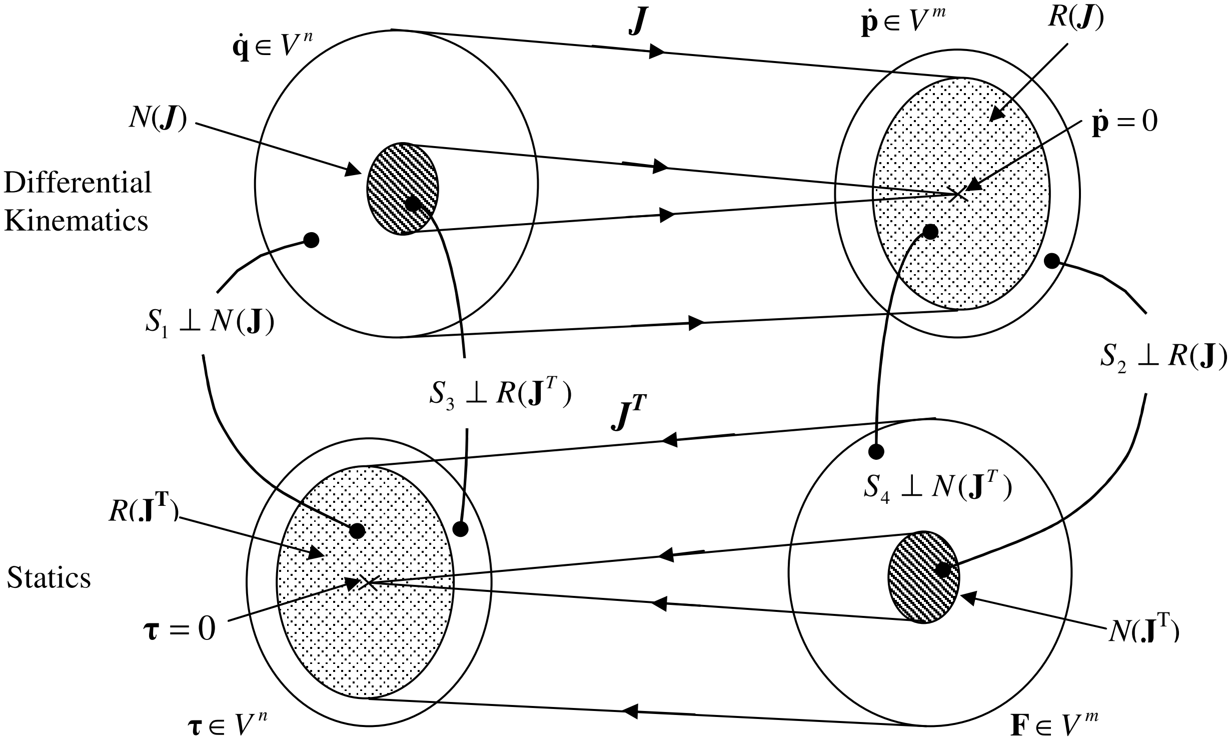


Figure 6.2.1 Two-dof articulated robot pushing the environment surface

6.3 Duality of Differential Kinematics and Statics

We have found that the equivalent joint torques is related to the endpoint force by the Jacobian matrix, which is the same matrix that relates the infinitesimal joint displacements to the end-effecter displacement. Thus, the static force relationship is closely related to the differential kinematics. In this section we discuss the physical meaning of this observation.

To interpret the similarity between differential kinematics and statics, we can use the linear mapping diagram of Figure 5.4.1. Recall that the differential kinematic equation can be regarded as a linear mapping when the Jacobian matrix is fixed at a given robot configuration. Figure 6.3.1 reproduces Figure 5.4.1 and completes it with a similar diagram associated with the static analysis. As before, the range space $R(\mathbf{J})$ represents the set of all the possible end-effecter velocities generated by joint motions. When the Jacobian matrix is degenerate or the robot configuration is singular, the range space does not span the whole vector space**.** Namely, there exists a direction in which the end-effecter cannot move with a non-zero velocity. See the subspace S_2 in the figure. The null space $N(\mathbf{J})$, on the other hand, represents the set of joint velocities that do not produce any velocity at the end-effecter. If the null space contains a non-zero element, the differential kinematic equation has an infinite number of solutions that cause the same end-effecter velocity.



$\dot{\mathbf{q}} \in V^n$
$\boldsymbol{J}$
$\dot{\mathbf{p}} \in V^m$
$R(\boldsymbol{J})$
$N(\boldsymbol{J})$
$\dot{\mathbf{p}} = 0$
Differential Kinematics
$S_1 \perp N(\mathbf{J})$
$S_3 \perp R(\mathbf{J}^T)$
$S_2 \perp R(\mathbf{J})$
$\boldsymbol{J}^{\boldsymbol{T}}$
$S_4 \perp N(\mathbf{J}^T)$
$R(\mathbf{J}^{\mathrm{T}})$
Statics
$\boldsymbol{\tau} = 0$
$N(\mathbf{J}^{\mathrm{T}})$
$\boldsymbol{\tau} \in V^n$
$\mathbf{F} \in V^m$

Figure 6.3.1 Duality of differential kinematics and statics

The lower half of Figure 6.3.1 is the linear mapping associated with the static force relationship given by eq.(6.2.4). Unlike differential kinematics, the mapping of static forces is given by the transpose of the Jacobian, generating a mapping from the m-dimensional vector space $\boldsymbol{V^m}$, associated with the Cartesian coordinates of the end-effecter, to the n-dimensional vector space V^n, associated with the joint coordinates. Therefore the joint torques $\boldsymbol{\tau}$ are always determined uniquely for any arbitrary endpoint force $\mathbf{F}$. However, for given joint torques, a balancing endpoint force does not always exist. As in the case of the differential kinematics, let us define the null space $N(\mathbf{J}^T)$ and the range space $R(\mathbf{J}^T)$ of the static force mapping. The null space $N(\mathbf{J}^T)$ represents the set of all endpoint forces that do not require any torques at the joints to bear the corresponding load. In this case the endpoint force is borne entirely by the structure of the linkage mechanism, i.e. constraint forces. The range space $R(\mathbf{J}^T)$, on the other hand, represents the set of all the possible joint torques that can balance the endpoint forces.

The ranges and null spaces of $\mathbf{J}$ and $\mathbf{J}^T$ are closely related. According to the rules of linear algebra, the null space $N(\mathbf{J})$ is the orthogonal complement of the range space $R(\mathbf{J}^T)$. Namely, if a non-zero n-vector $\mathbf{x}$ is in $N(\mathbf{J})$, it cannot also belong to $R(\mathbf{J}^T)$, and vice-versa. If we denote by S_1 the orthogonal complement of $N(\mathbf{J})$, then the range space $R(\mathbf{J}^T)$ is identical to S_1, as shown in the figure. Also, space S_3, i.e., the orthogonal complement of $R(\mathbf{J}^T)$ is identical to $N(\mathbf{J})$. What this implies is that, in the direction in which joint velocities do not cause any end-effecter velocity, the joint torques cannot be balanced with any endpoint force. In order to maintain a stationary configuration, the joint torques in this space must be zero.

There is a similar correspondence in the end-effecter coordinate space $V^{m.}$ The range space $R(\mathbf{J})$ is the orthogonal complement to the null space $N(\mathbf{J}^T)$. Hence, the subspace S_2 in the figure is identical to $N(\mathbf{J}^T)$, and the subspace S_4 is identical to $R(\mathbf{J})$. Therefore, no joint torques are required to balance the end point force when the external force acts in the direction in which the end-effecter cannot be moved by joint movements. Also, when the external endpoint force is applied in the direction along which the end-effecter can move, the external force must be borne entirely by the joint torques. When the Jacobian matrix is degenerate or the arm is in a singular configuration, the null space $N(\mathbf{J}^T)$ has a non-zero dimension, and the external force can be borne in part by the mechanical structure. Thus, differential kinematics and statics are closely related. This relationship is referred to as the *duality* of differential kinematics and statics.

6.4 Closed-Loop Kinematic Chains

The relationship between joint torques and the endpoint force obtained in Theorem 6.1 can be extended to a class of parallel-link mechanisms with closed kinematic-chains. It can also be extended to multi-fingered hands, leg locomotion, and other robot mechanisms having closed kinematic chains. In this section we discuss classes of closed kinematic chains based on the principle of virtual work.

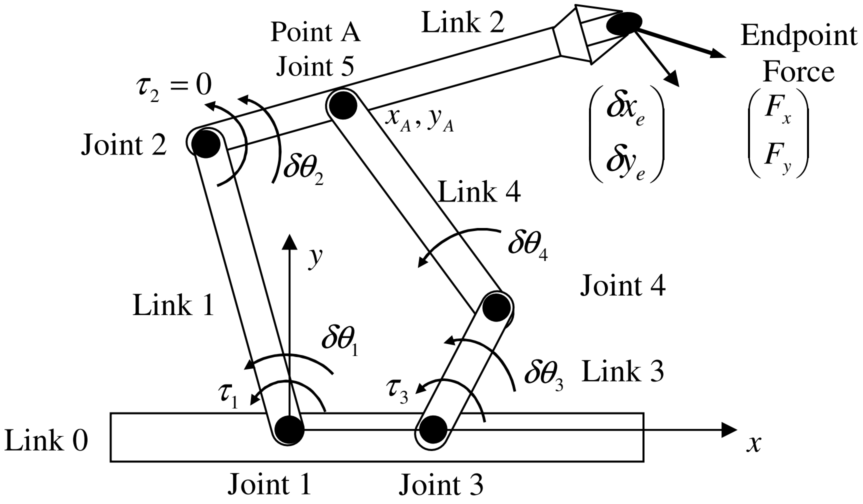


Figure 6.4.1 Five-bar-link robot exerting endpoint force

We begin by revisiting the five-bar-link planar robot shown in Figure 6.4.1. This robot has two degrees of freedom, comprising two active joints, Joints 1 and 3, and three passive joints, Joints 2, 4, and 5. Therefore the virtual work associated with the endpoint force and joint toques is given by

$$\delta Work = \tau_1 \delta\theta_1 + \tau_2 \delta\theta_2 + \tau_3 \delta\theta_3 + \cdots \tau_5 \delta\theta_5 - F_x \delta x_e - F_y \delta y_e \tag{6.4.1}$$

We assume no friction at the joints. Therefore the three passive joints cannot bear any torque load about their joint axis. Substituting $\tau_2 = \tau_4 = \tau_5 = 0$ into the above yields

$$\delta Work = \begin{pmatrix} \tau_1 & \tau_3 \end{pmatrix}^T \begin{pmatrix} \delta\theta_1 \\ \delta\theta_3 \end{pmatrix} - \begin{pmatrix} F_x & F_y \end{pmatrix}^T \begin{pmatrix} \delta x_e \\ \delta y_e \end{pmatrix}. \tag{6.4.2}$$

For any given configuration of the robot, the virtual displacements of the end-effecter are uniquely determined by the virtual displacements of Joints 1 and 3. In fact, the former is related to the latter via the Jacobian matrix:

$$\mathbf{J} = \begin{pmatrix} \dfrac{\partial x_e}{\partial \theta_1} & \dfrac{\partial x_e}{\partial \theta_3} \\ \dfrac{\partial y_e}{\partial \theta_1} & \dfrac{\partial y_e}{\partial \theta_3} \end{pmatrix} \tag{6.4.3}$$

Using this Jacobian,

$$\delta Work = \boldsymbol{\tau}^T \delta\mathbf{q} - \mathbf{F}^T \mathbf{J} \cdot \delta\mathbf{q} = (\boldsymbol{\tau} - \mathbf{J}^T \mathbf{F})^T \cdot \delta\mathbf{q} = 0, \quad \forall \delta\mathbf{q} \tag{6.4.4}$$

where

$$\delta\mathbf{q} = \begin{pmatrix} \delta\theta_1 & \delta\theta_3 \end{pmatrix}^T, \quad \delta\mathbf{p} = \begin{pmatrix} \delta\, x_e & \delta\, y_e \end{pmatrix}^T \tag{6.4.5}$$

Eq.(5) implies

$$\boldsymbol{\tau} = \mathbf{J}^T \cdot \mathbf{F} \tag{6.4.6}$$

which is the same form as eq.(6.2.4).

In general the following Corollary holds.

Corollary 6.1

Consider a *n* degree-of-freedom robot mechanism with *n* active joints. Assume that all the joints are frictionless, and that, for a given configuration of the robot mechanism, there exists a unique Jacobian matrix relating the virtual displacements of its end-effecter, $\delta\mathbf{p} \in \Re^{\mathbf{m \times 1}}$, to the virtual displacements of the active joints, $\delta\mathbf{q} \in \Re^{\mathbf{n \times 1}}$,

$$\delta\mathbf{p} = \mathbf{J}\delta\mathbf{q}\,. \tag{6.4.7}$$

Then the equivalent joint torques $\boldsymbol{\tau} \in \Re^{n\times 1}$ to bear an arbitrary endpoint force $\mathbf{F} \in \Re^{m\times 1}$ is given by

$$\boldsymbol{\tau} = \mathbf{J}^T \cdot \mathbf{F} \tag{6.4.8}$$

□

Note that the joint coordinates associated with the active joints are not necessarily generalized coordinates that uniquely locate the system. For example, the arm configuration of the five-bar-link robot shown in Figure 6.4.1 is not uniquely determined with joint angles θ_1 and θ_3 alone. There are two configurations for given θ_1 and θ_3. The corollary requires the differential relation to be defined uniquely in the vicinity of the given configuration.

6.5 Over-Actuated Systems

If a *n* degree-of-freedom robot system has more than *n* active joints, or less than *n* active joints, the above corollary does not apply. These are called ***over-actuated*** and ***under-actuated systems***, respectively. Over-actuated systems are of particular importance in many manipulation and locomotion applications. In the following we will consider the static relationship among joint torques and endpoint forces for a class of over-actuated systems.

Figure 6.4.2 shows a two-fingered hand manipulating an object within a plane. Note that both fingers are connected at the fingertips holding the object. While holding the object, the system has three degrees of freedom. Since each finger has two active joints, the total number of active joints is four. Therefore the system is over-actuated.

Using the notation shown in the figure, the virtual work is given by

$$\delta Work = \tau_1\delta\theta_1 + \tau_2\delta\theta_2 + \tau_3\delta\theta_3 + \tau_4\delta\theta_4 - F_x\delta x_e - F_y\delta y_e \tag{6.5.1}$$

Note that only three virtual displacements of the four joint angles are independent. There exists a differential relationship between one of the joints, say θ_4, and the other three due to the kinematic constraint. Let us write it as

$$\delta\theta_4 = \mathbf{J}_c \cdot \delta\mathbf{q} \tag{6.5.2}$$

where $\delta\mathbf{q} = (\delta\theta_1 \quad \delta\theta_2 \quad \delta\theta_3)^T$ are independent, and $\mathbf{J}_c$ is the 1x3 Jacobian associated with the constraint due to the closed kinematic chain. Substituting this equation together with the Jacobian relating the end effecter displacements to the tree joint displacements into eq.(1),

$$\delta Work = \boldsymbol{\tau}^T\delta\mathbf{q} + \tau_4\mathbf{J}_c\delta\mathbf{q} - \mathbf{F}^T\mathbf{J}\delta\mathbf{q} = 0, \quad \forall\delta\mathbf{q} \tag{6.5.3}$$

The virtual work vanished for an arbitrary $\delta\mathbf{q}$ only when

$$\boldsymbol{\tau} = -\mathbf{J}_c^{\ T}\tau_4 + \mathbf{J}^T\mathbf{F} \tag{6.5.4}$$

The two-fingered hand is at equilibrium only when the above condition is met. When the external endpoint force is zero: **F=0,** we obtain

$$\boldsymbol{\tau}_0 = -\mathbf{J}_c^{\ T}\tau_4 \tag{6.5.5}$$

This gives a particular combination of joint torques that do not influence the force balance with the external endpoint load **F.** The joint torques having this particular proportion generate the internal force applied to the object, as illustrated in the figure. This internal force is a grasp force that is needed for performing a task.

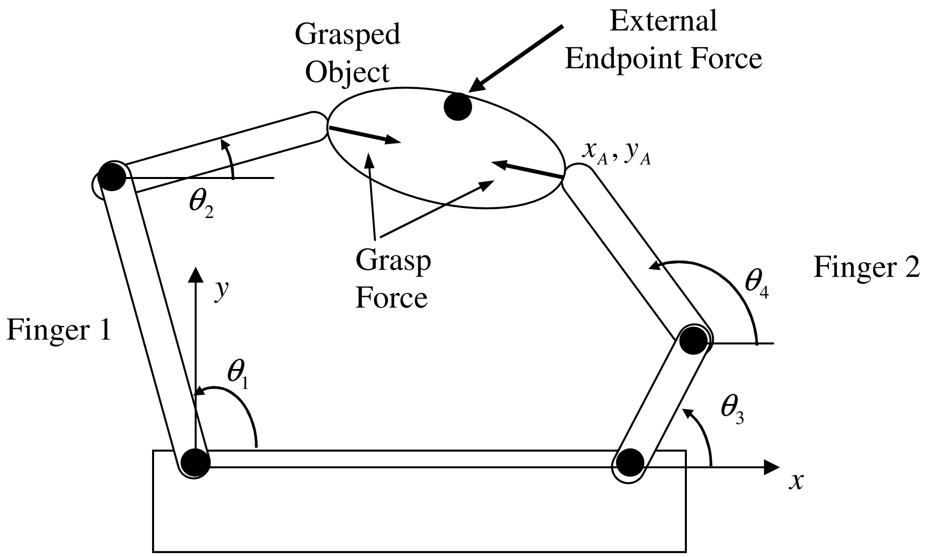


Figure 6.5.1 Two-fingered hand manipulating a grasped object

Exercise 6.2

Define geometric parameters needed in Figure 6.5.1, and obtain the two Jacobian matrices associated with the two-fingered hand holding an object. Furthermore, obtain the grasp force using the Jacobian matrices and the joint torques.

Chapter 7
Dynamics

In this chapter, we analyze the dynamic behavior of robot mechanisms. The dynamic behavior is described in terms of the time rate of change of the robot configuration in relation to the joint torques exerted by the actuators. This relationship can be expressed by a set of differential equations, called *equations of motion,* that govern the dynamic response of the robot linkage to input joint torques. In the next chapter, we will design a control system on the basis of these equations of motion.

Two methods can be used in order to obtain the equations of motion: the *Newton-Euler formulation,* and the *Lagrangian formulation.* The Newton-Euler formulation is derived by the direct interpretation of Newton's Second Law of Motion, which describes dynamic systems in terms of force and momentum. The equations incorporate all the forces and moments acting on the individual robot links, including the coupling forces and moments between the links. The equations obtained from the Newton-Euler method include the constraint forces acting between adjacent links. Thus, additional arithmetic operations are required to eliminate these terms and obtain explicit relations between the joint torques and the resultant motion in terms of joint displacements. In the Lagrangian formulation, on the other hand, the system's dynamic behavior is described in terms of work and energy using generalized coordinates. This approach is the extension of the indirect method discussed in the previous chapter to dynamics. Therefore, all the workless forces and constraint forces are automatically eliminated in this method. The resultant equations are generally compact and provide a closed-form expression in terms of joint torques and joint displacements. Furthermore, the derivation is simpler and more systematic than in the Newton-Euler method.

The robot's equations of motion are basically a description of the relationship between the input joint torques and the output motion, i.e. the motion of the robot linkage. As in kinematics and in statics, we need to solve the inverse problem of finding the necessary input torques to obtain a desired output motion. This *inverse dynamics* problem is discussed in the last section of this chapter. Efficient algorithms have been developed that allow the dynamic computations to be carried out on-line in real time.

7.1 Newton-Euler Formulation of Equations of Motion

7.1.1. Basic Dynamic Equations

In this section we derive the equations of motion for an individual link based on the direct method, i.e. Newton-Euler Formulation. The motion of a rigid body can be decomposed into the translational motion with respect to an arbitrary point fixed to the rigid body, and the rotational motion of the rigid body about that point. The dynamic equations of a rigid body can also be represented by two equations: one describes the translational motion of the centroid (or center of mass), while the other describes the rotational motion about the centroid. The former is Newton's equation of motion for a mass particle, and the latter is called Euler's equation of motion.

We begin by considering the free body diagram of an individual link. Figure 7.1.1 shows all the forces and moments acting on link i. The figure is the same as Figure 6.1.1, which describes the static balance of forces, except for the inertial force and moment that arise from the dynamic motion of the link. Let $\mathbf{v}_{ci}$ be the linear velocity of the centroid of link i with reference to the base coordinate frame *O-xyz,* which is an inertial reference frame. The inertial force is then given by $-m_i\dot{\mathbf{v}}_{ci}$, where m_i is the mass of the link and $\dot{\mathbf{v}}_{ci}$ is the time derivative of $\mathbf{v}_{ci}$. Based on D'Alembert's principle, the equation of motion is then obtained by adding the inertial force to the static balance of forces in eq.(6.1.1) so that

$$\mathbf{f}_{i-1,i} - \mathbf{f}_{i,i+1} + m_i\mathbf{g} - m_i\dot{\mathbf{v}}_{ci} = \mathbf{0}, \qquad i = 1,\cdots,n \tag{7.1.1}$$

where, as in Chapter 6, $\mathbf{f}_{i-1,i}$ and $-\mathbf{f}_{i,i+1}$ are the coupling forces applied to link *i* by links *i*-1 and *i*+1, respectively, and **g** is the acceleration of gravity.

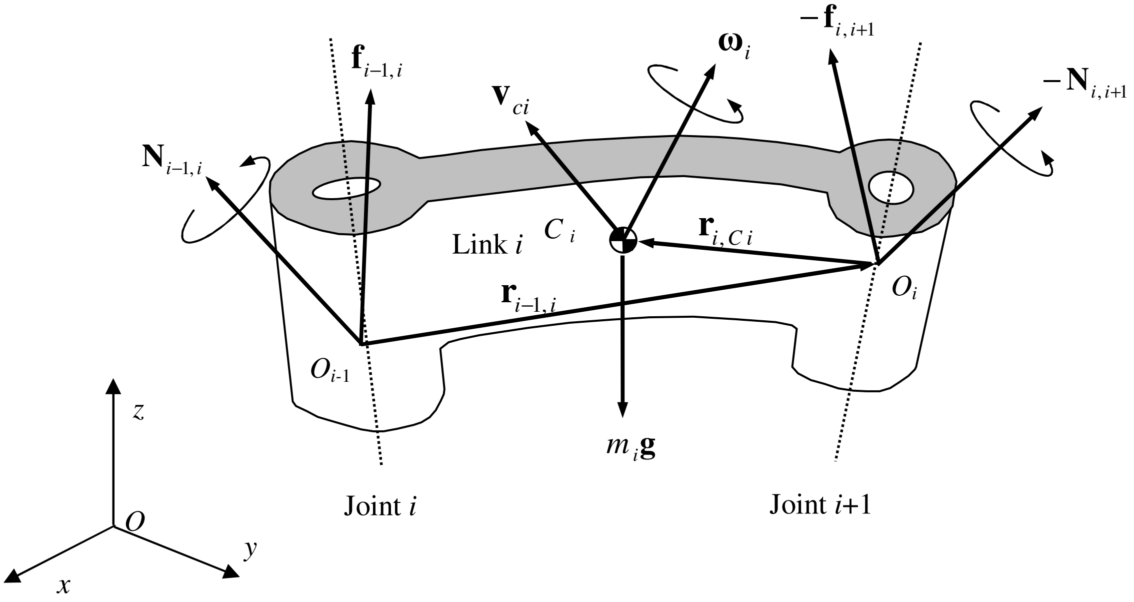


Figure 7.1.1 Free body diagram of link *i* in motion

Rotational motions are described by Euler's equations. In the same way as for translational motions, adding "inertial torques" to the static balance of moments yields the dynamic equations. We begin by describing the mass properties of a single rigid body with respect to rotations about the centroid. The mass properties are represented by an *inertia tensor,* or an *inertia matrix,* which is a 3 x 3 symmetric matrix defined by

$$\mathbf{I} = \begin{pmatrix} \int_{body}\{(y-y_c)^2+(z-z_c)^2\}\rho\,dV & -\int_{body}(x-x_c)(y-y_c)\rho\,dV & -\int_{body}(z-z_c)(x-x_c)\rho\,dV \\ -\int_{body}(x-x_c)(y-y_c)\rho\,dV & \int_{body}\{(z-z_c)^2+(x-x_c)^2\}\rho\,dV & -\int_{body}(y-y_c)(z-z_c)\rho\,dV \\ -\int_{body}(z-z_c)(x-x_c)\rho\,dV & -\int_{body}(y-y_c)(z-z_c)\rho\,dV & \int_{body}\{(x-x_c)^2+(y-y_c)^2\}\rho\,dV \end{pmatrix} \tag{7.1.2}$$

where ρ is the mass density, x_c, y_c, z_c are the coordinates of the centroid of the rigid body, and each integral is taken over the entire volume *V* of the rigid body. Note that the inertia matrix varies with the orientation of the rigid body. Although the inherent mass property of the rigid body does not change when viewed from a frame fixed to the body, its matrix representation when viewed from a fixed frame, i.e. inertial reference frame, changes as the body rotates.

The inertial torque acting on link *i* is given by the time rate of change of the angular momentum of the link at that instant. Let $\boldsymbol{\omega}_i$ be the angular velocity vector and $\mathbf{I}_i$ be the centroidal inertia tensor of link *i*, then the angular momentum is given by $\mathbf{I}_i\boldsymbol{\omega}_i$. Since the inertia tensor varies as the orientation of the link changes, the time derivative of the angular momentum includes not only the angular acceleration term $\mathbf{I}_i\dot{\boldsymbol{\omega}}_i$, but also a term resulting from changes in the inertia tensor viewed from a fixed frame. This latter term is known as the *gyroscopic torque* and is given by $\boldsymbol{\omega}_i \times (\mathbf{I}_i\,\boldsymbol{\omega}_i)$. Adding these terms to the original balance of moments (4-2) yields

$$\mathbf{N}_{i-1,i} - \mathbf{N}_{i,i+1} - (\mathbf{r}_{i-1,i} + \mathbf{r}_{i,Ci}) \times \mathbf{f}_{i-1,i} + (-\mathbf{r}_{i,Ci}) \times (-\mathbf{f}_{i,i+1}) - \mathbf{I}_i\dot{\boldsymbol{\omega}}_i - \boldsymbol{\omega}_i \times (\mathbf{I}_i\,\boldsymbol{\omega}_i) = \mathbf{0}, \quad i = 1, \cdots, n \tag{7.1.3}$$

using the notations of Figure 7.1.1. Equations (2) and (3) govern the dynamic behavior of an individual link. The complete set of equations for the whole robot is obtained by evaluating both equations for all the links, $i = 1, \cdot\ ,n.$

7.1.2. Closed-Form Dynamic Equations

The Newton-Euler equations we have derived are not in an appropriate form for use in dynamic analysis and control design. They do not explicitly describe the input-output relationship, unlike the relationships we obtained in the kinematic and static analyses. In this section, we modify the Newton-Euler equations so that explicit input-output relations can be obtained. The Newton-Euler equations involve coupling forces and moments $\mathbf{f}_{i-1,i}$ and $\mathbf{N}_{i-1,i}$. As shown in eqs.(6.2.1) and (6.2.2), the joint torque τ_i, which is the input to the robot linkage, is included in the coupling force or moment. However, τ_i is not *explicitly* involved in the Newton-Euler equations. Furthermore, the coupling force and moment also include workless constraint forces, which act *internally* so that individual link motions conform to the geometric constraints imposed by the mechanical structure. To derive explicit input-output dynamic relations, we need to separate the input joint torques from the constraint forces and moments. The Newton-Euler equations are described in terms of centroid velocities and accelerations of individual arm links. Individual link motions, however, are not independent, but are coupled through the linkage. They must satisfy certain kinematic relationships to conform to the geometric constraints. Thus, individual centroid position variables are not appropriate for output variables since they are not independent.

The appropriate form of the dynamic equations therefore consists of equations described in terms of all independent position variables and input forces, i.e., joint torques, that are explicitly involved in the dynamic equations. Dynamic equations in such an explicit input- output form are referred to as *closed-form dynamic equations.* As discussed in the previous chapter, joint displacements $\mathbf{q}$ are a complete and independent set of generalized coordinates that locate the whole robot mechanism, and joint torques $\boldsymbol{\tau}$ are a set of independent inputs that are separated from constraint forces and moments. Hence, dynamic equations in terms of joint displacements $\mathbf{q}$ and joint torques $\boldsymbol{\tau}$ are closed-form dynamic equations.

Example 7.1

Figure 7.1.1 shows the two dof planar manipulator that we discussed in the previous chapter. Let us obtain the Newton-Euler equations of motion for the two individual links, and then derive closed-form dynamic equations in terms of joint displacements θ_1 and θ_2, and joint torques τ_1 and τ_2. Since the link mechanism is planar, we represent the velocity of the centroid of each link by a 2-dimensional vector $\mathbf{v}_i$ and the angular velocity by a scalar velocity ω_i . We assume that the centroid of link *i* is located on the center line passing through adjacent joints at a distance ℓ_{ci} from joint *i,* as shown in the figure. The axis of rotation does not vary for the planar linkage. The inertia tensor in this case is reduced to a scalar moment of inertia denoted by I_i.

From eqs. (1) and (3), the Newton-Euler equations for link 1 are given by

$$\begin{aligned}&\mathbf{f}_{0,1} - \mathbf{f}_{1,2} + m_1\mathbf{g} - m_1\dot{\mathbf{v}}_{c1} = \mathbf{0},\\ &\mathbf{N}_{0,1} - \mathbf{N}_{1,2} + \mathbf{r}_{1,c1}\times\mathbf{f}_{1,2} - \mathbf{r}_{0,c1}\times\mathbf{f}_{0,1} - I_1\dot{\omega}_1 = 0\end{aligned} \qquad (7.1.4)$$

Note that all vectors are 2 x 1, so that moment N $_{i-1,i}$ and the other vector products are scalar quantities. Similarly, for link 2,

$$\begin{aligned}&\mathbf{f}_{1,2} + m_2\mathbf{g} - m_2\dot{\mathbf{v}}_{c2} = \mathbf{0},\\ &\mathbf{N}_{1,2} - \mathbf{r}_{1,c2}\times\mathbf{f}_{1,2} - I_2\dot{\omega}_2 = 0\end{aligned} \qquad (7.1.5)$$

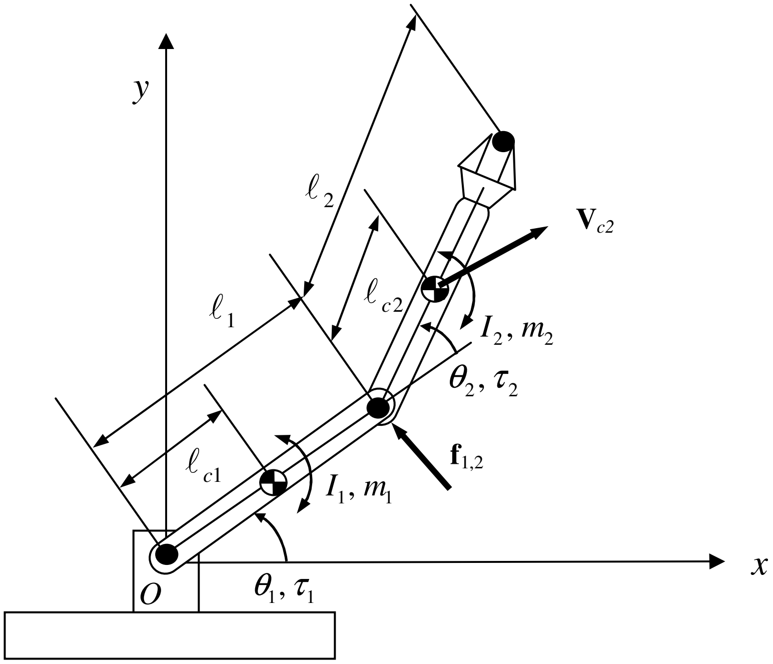


Figure 7.1.2 Mass properties of two dof planar robot

To obtain closed-form dynamic equations, we first eliminate the constraint forces and separate them from the joint torques, so as to explicitly involve the joint torques in the dynamic equations. For the planar manipulator, the joint torques τ_1 and τ_2 are equal to the coupling moments:

$$N_{i-1,i} = \tau_i, \quad i = 1, 2 \qquad (7.1.6)$$

Substituting eq.(6) into eq.(5) and eliminating $\mathbf{f}_{1,2}$ we obtain

$$\tau_2 - \mathbf{r}_{1,c2} \times m_2 \dot{\mathbf{v}}_{c2} + \mathbf{r}_{1,c2} \times m_2 \mathbf{g} - I_2 \dot{\omega}_2 = 0 \tag{7.1.7}$$

Similarly, eliminating $\mathbf{f}_{0,1}$ yields,

$$\tau_1 - \tau_2 - \mathbf{r}_{0,c1} \times m_1 \dot{\mathbf{v}}_{c1} - \mathbf{r}_{0,1} \times m_2 \dot{\mathbf{v}}_{c2} + \mathbf{r}_{0,c1} \times m_1 \mathbf{g} + \mathbf{r}_{0,1} \times m_2 \mathbf{g} - I_1 \dot{\omega}_1 = 0 \tag{7.1.8}$$

Next, we rewrite $\mathbf{v}_{ci}$, ω_i, and $\mathbf{r}_{i,i+1}$ using joint displacements θ_1 and θ_2, which are independent variables. Note that ω_2 is the angular velocity relative to the base coordinate frame, while θ_2 is measured relative to link 1. Then, we have

$$\omega_1 = \dot{\theta}_1, \quad \omega_2 = \dot{\theta}_1 + \dot{\theta}_2 \tag{7.1.9}$$

The linear velocities can be written as

$$\begin{aligned} \mathbf{v}_{c1} &= \begin{pmatrix} -\ell_{c1}\dot{\theta}_1 \sin\theta_1 \\ \ell_{c1}\dot{\theta}_1 \cos\theta_1 \end{pmatrix} \\ \mathbf{v}_{c2} &= \begin{pmatrix} -\{\ell_1 \sin\theta_1 + \ell_{c2}\sin(\theta_1+\theta_2)\}\dot{\theta}_1 - \ell_{c2}\sin(\theta_1+\theta_2)\}\dot{\theta}_2 \\ \{\ell_1 \cos\theta_1 + \ell_{c2}\cos(\theta_1+\theta_2)\}\dot{\theta}_1 + \ell_{c2}\cos(\theta_1+\theta_2)\}\dot{\theta}_2 \end{pmatrix} \end{aligned} \tag{7.1.10}$$

Substituting eqs. (9) and (10) along with their time derivatives into eqs. (7) and (8), we obtain the closed-form dynamic equations in terms of θ_1 and θ_2 :

$$\tau_1 = H_{11}\ddot{\theta}_1 + H_{12}\ddot{\theta}_2 - h\dot{\theta}_2^2 - 2h\dot{\theta}_1\dot{\theta}_2 + G_1 \tag{7.1.11-a}$$

$$\tau_2 = H_{22}\ddot{\theta}_2 + H_{21}\ddot{\theta}_1 + h\dot{\theta}_1^2 + G_2 \tag{7.1.11-b}$$

where

$$H_{11} = m_1\ell_{c1}^2 + I_1 + m_2(\ell_1^2 + \ell_{c2}^2 + 2\ell_1\ell_{c2}\cos\theta_2) + I_2 \tag{7.1.12-a}$$

$$H_{22} = m_2\ell_{c2}^2 + I_2 \tag{7.1.12-b}$$

$$H_{12} = m_2(\ell_{c2}^2 + \ell_1\ell_{c2}\cos\theta_2) + I_2 \tag{7.1.12-c}$$

$$h = m_2\ell_1\ell_{c2}\sin\theta_2 \tag{7.1.12-d}$$

$$G_1 = m_1\ell_{c1}\, g\cos\theta_1 + m_2\, g\{\ell_{c2}\cos(\theta_1+\theta_2) + \ell_1\cos\theta_1\} \tag{7.1.12-e}$$

$$G_2 = m_2\, g\ell_{c2}\cos(\theta_1+\theta_2) \tag{7.1.12-f}$$

The scalar *g* represents the acceleration of gravity along the negative *y*-axis. □

More generally, the closed-form dynamic equations of an *n*-degree-of-freedom robot can be given in the form

$$\tau_i = \sum_{j+1}^{n} H_{ij}\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}\dot{q}_j\dot{q}_k + G_i, \qquad i = 1,\cdots,n \tag{7.1.13}$$

where coefficients H_{ij} , h_{ijk}, and G_i are functions of joint displacements $q_1, q_2, \cdots, q_n$. When external forces act on the robot system, the left-hand side must be modified accordingly.

7.1.3. Physical Interpretation of the Dynamic Equations

In this section, we interpret the physical meaning of each term involved in the closed-form dynamic equations for the two-dof planar robot.

The last term in each of eqs. (11-a, b), G_i , accounts for the effect of gravity. Indeed, the terms G_1 and G_2, given by (12-e, f), represent the moments created by the masses m_1 and m_2 about their individual joint axes. The moments are dependent upon the arm configuration. When the arm is fully extended along the *x*-axis, the gravity moments become maximums.

Next, we investigate the first terms in the dynamic equations. When the second joint is immobilized, i.e. $\dot{\theta}_2 = 0$ and $\ddot{\theta}_2 = 0$, the first dynamic equation reduces to $\tau_1 = H_{11}\ddot{\theta}_1$, where the gravity term is neglected. From this expression it follows that the coefficient H_{11} accounts for the moment of inertia seen by the first joint when the second joint is immobilized. The coefficient H_{11} given by eq. (12-a) is interpreted as the total moment of inertia of both links reflected to the first joint axis. The first two terms, $m_1\ell_{c1}^{\ 2} + I_1$, in eq. (12-a), *represent* the moment of inertia of link *1* with respect to joint *1*, while the other terms are the contribution from link *2*. The inertia of the second link depends upon the distance *L* between the centroid of link *2* and the first joint axis, as illustrated in Figure 7.1.3. The distance *L* is a function of the joint angle θ_2 and is given by

$$L^2 = \ell_1^{\ 2} + \ell_{c2}^{\ 2} + 2\ell_1\ell_{c2}\cos\theta_2 \tag{7.1.14}$$

Using the parallel axes theorem of moment of inertia (Goldstein, 1981), the inertia of link *2* with respect to joint *1* is $m_2L^2+I_2$, which is consistent with the last two terms in eq. (12-a). Note that the inertia varies with the arm configuration. The inertia is maximum when the arm is fully extended ($\theta_2 = 0$), and minimum when the arm is completely contracted ($\theta_2 = \pi$).

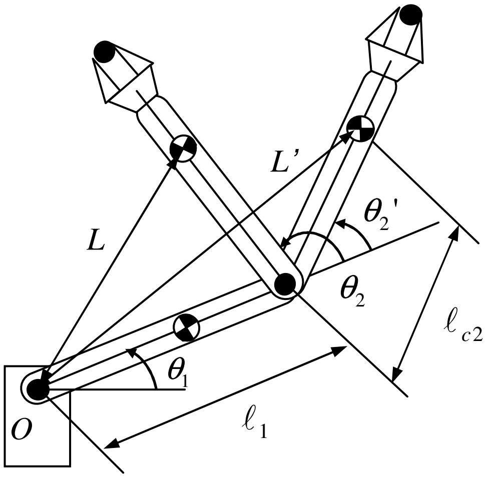


Figure 7.1.3 Varying inertia depending on the arm configuration

Let us now examine the second terms on the right hand side of eq. (11). Consider the instant when $\dot{\theta}_1 = \dot{\theta}_2 = 0$ and $\ddot{\theta}_1 = 0$, then the first equation reduces to $\tau_1 = H_{12}\ddot{\theta}_2$, where the gravity term is again neglected. From this expression it follows that the second term accounts for the effect of the second link motion upon the first joint. When the second link is accelerated, the reaction force and torque induced by the second link act upon the first link. This is clear in the original Newton-Euler equations (4), where the coupling force -$\mathbf{f}_{1,2}$ and moment -$N_{1,2}$ from link *2* are involved in the dynamic equation for link *1*. The coupling force and moment cause a torque τ_{int} about the first joint axis given by

$$\begin{aligned}\tau_{\text{int}} &= -N_{1,2} - \mathbf{r}_{0,1} \times \mathbf{f}_{1,2} \\ &= -I_2\dot{\omega}_2 - \mathbf{r}_{0,c2} \times m_2\dot{\mathbf{v}}_{c2} \\ &= -\{I_2 + m_2(\ell_{c2}{}^2 + \ell_1\ell_{c2}\cos\theta_2)\}\ddot{\theta}_2\end{aligned} \tag{7.1.15}$$

where $N_{1,2}$ and $\mathbf{f}_{1,2}$ are evaluated using eq. (5) for $\dot{\theta}_1 = \dot{\theta}_2 = 0$ and $\ddot{\theta}_1 = 0$. This agrees with the second term in eq. (11-a). Thus, the second term accounts for the interaction between the two joints.

The third terms in eq. (11) are proportional to the square of the joint velocities. We consider the instant when $\dot{\theta}_2 = 0$ and $\ddot{\theta}_1 = \ddot{\theta}_2 = 0$, as shown in Figure 7.1.4-(a). In this case, a centrifugal force acts upon the second link. Let $\mathbf{f}_{\text{cent}}$ be the centrifugal force. Its magnitude is given by

$$\left|\mathbf{f}_{cent}\right| = m_2\, L\dot{\theta}_1{}^2 \tag{7.1.16}$$

where L is the distance between the centroid C_2 and the first joint O. The centrifugal force acts in the direction of position vector $\mathbf{r}_{O,C2}$. This centrifugal force causes a moment τ_{cent} about the second joint. Using eq. (16), the moment τ_{cent} is computed as

$$\tau_{cent} = \mathbf{r}_{1,c2} \times \mathbf{f}_{cent} = -m_2\ell_1\ell_{c2}\dot{\theta}_1{}^2 \tag{7.1.17}$$

This agrees with the third term $h\dot{\theta}_1^2$ in eq. (11-b). Thus we conclude that the third term is caused by the centrifugal effect on the second joint due to the motion of the first joint. Similarly, rotating the second joint at a constant velocity causes a torque of $-h\dot{\theta}_2^2$ due to the centrifugal effect upon the first joint.

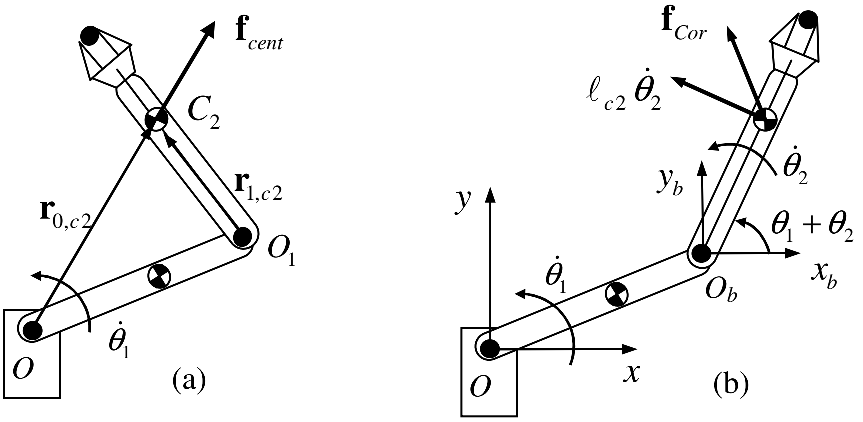


Figure 7.1.4 Centrifugal (a) and Coriolis (b) effects

Finally we discuss the fourth term of eq. (11-a), which is proportional to the product of the joint velocities. Consider the instant when the two joints rotate at velocities $\dot{\theta}_1$ and $\dot{\theta}_2$ at the same time. Let *O*$_b$-*x*$_b$*y*$_b$ be the coordinate frame attached to the tip of link *1*, as shown in Figure 7.1.4-(b). Note that the frame *O*$_b$-*x*$_b$*y*$_b$ is parallel to the base coordinate frame at the instant shown. However, the frame rotates at the angular velocity $\dot{\theta}_1$ together with link 1. The mass centroid of link *2* moves at a velocity of $\ell_{c2}\dot{\theta}_2$ relative to link 1, i.e. the moving coordinate frame *O*$_b$-*x*$_b$*y*$_b$. When a mass particle *m* moves at a velocity of $\mathbf{v}_b$ relative to a moving coordinate frame rotating at an angular velocity ω, the mass particle has the so-called *Coriolis force* given by $-2m(\boldsymbol{\omega}\times\mathbf{v}_b)$. Let $\mathbf{f}_{Cor}$ be the force acting on link 2 due to the Coriolis effect. The Coriolis force is given by

$$\mathbf{f}_{Cor} = \begin{pmatrix} 2m_2\,\ell_{c2}\,\dot{\theta}_1\,\dot{\theta}_2\cos(\theta_1+\theta_2) \\ 2m_2\,\ell_{c2}\,\dot{\theta}_1\,\dot{\theta}_2\sin(\theta_1+\theta_2) \end{pmatrix} \tag{7.1.18}$$

This Coriolis force causes a moment τ_{Cor} about the first joint, which is given by

$$\tau_{Cor} = \mathbf{r}_{0,c2}\times\mathbf{f}_{Cor} = 2m_2\,\ell_1\,\ell_{c2}\,\dot{\theta}_1\,\dot{\theta}_2\sin\theta_2 \tag{7.1.19}$$

The right-hand side of the above equation agrees with the fourth term in eq. (11-a). Since the Coriolis force given by eq. (18) acts in parallel with link *2*, the force does not create a moment about the second joint in this particular case.

Thus, the dynamic equations of a robot arm are characterized by a configuration-dependent inertia, gravity torques, and interaction torques caused by the accelerations of the other joints and the existence of centrifugal and Coriolis effects.

7.2. Lagrangian Formulation of Robot Dynamics

7.2.1. Lagrangian Dynamics

In the Newton-Euler formulation, the equations of motion are derived from Newton's Second Law, which relates force and momentum, as well as torque and angular momentum. The resulting equations involve constraint forces, which must be eliminated in order to obtain closed-form dynamic equations. In the Newton-Euler formulation, the equations are not expressed in terms of independent variables, and do not include input joint torques explicitly. Arithmetic operations are needed to derive the closed-form dynamic equations. This represents a complex procedure that requires physical intuition, as discussed in the previous section.

An alternative to the Newton-Euler formulation of manipulator dynamics is the Lagrangian formulation, which describes the behavior of a dynamic system in terms of work and energy stored in the system rather than of forces and moments of the individual members involved. The constraint forces involved in the system are automatically eliminated in the formulation of Lagrangian dynamic equations. The closed-form dynamic equations can be derived systematically in any coordinate system.

Let $q_1,\cdots,q_n$ be generalized coordinates that completely locate a dynamic system. Let *T* and *U* be the total kinetic energy and potential energy stored in the dynamic system. We define the Lagrangian *L* by

$$L(q_i,\dot{q}_i) = T(q_i,\dot{q}_i) - U(q_i) \qquad (7.2.1)$$

Note that the potential energy is a function of generalized coordinates q_i and that the kinetic energy is that of generalized velocities $\dot{q}_i$ as well as generalized coordinates q_i. Using the Lagrangian, equations of motion of the dynamic system are given by

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i, \qquad i = 1,\cdots,n \qquad (7.2.2)$$

where Q_i is the generalized force corresponding to the generalized coordinate q_i. Considering the virtual work done by non-conservative forces can identify the generalized forces acting on the system.

7.2.2 Planar Robot Dynamics

Before discussing general robot dynamics in three-dimensional space, we consider the 2 dof planar robot, for which we have derived the equations of motion based on Newton-Euler Formulation. Figure 7.2.1 shows the same robot mechanism with a few new variables needed for the Lagrangian Formulation.

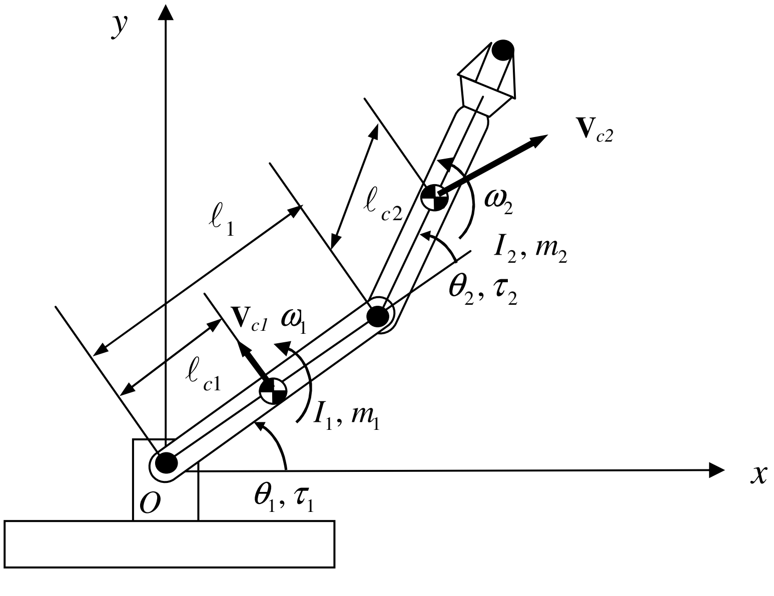


Figure 7.2.1 Two dof robot

The total kinetic energy stored in the two links moving at linear velocity $\mathbf{v}_{ci}$ and angular velocity ω_i at the centroids, as shown in the figure, is given by

$$T = \sum_{i=1}^{2}\left(\frac{1}{2}m_i\left|\mathbf{v}_{ci}\right|^2 + \frac{1}{2}I_i\,\omega_i^2\right) \qquad (7.2.3)$$

where $\left|\mathbf{v}_{ci}\right|$ represents the magnitude of the velocity vector. Note that the linear velocities and the angular velocities are not independent variables, but are functions of joint angles and joint angular velocities, i.e. the generalized coordinates and the generalized velocities that locate the dynamic state of the system uniquely. We need to rewrite the above kinetic energy so that it is with respect to θ_i and $\dot{\theta}_{\mathrm{i}}$. The angular velocities are given by

$$\omega_1 = \dot{\theta}_1, \quad \omega_2 = \dot{\theta}_1 + \dot{\theta}_2 \tag{7.2.4}$$

The linear velocity of the first link is simply

$$|\mathbf{v}_{c1}|^2 = \ell_{c1}{}^2 \dot{\theta}_1{}^2 \tag{7.2.5}$$

However, the centroidal linear velocity of the second link $\mathbf{v}_{c2}$ needs more computation. Treating the centroid C_2 as an endpoint and applying the formula for computing the endpoint velocity yield the centroidal velocity. Let $\mathbf{J}_{c2}$ be the 2x2 Jacobian matrix relating the centroidal velocity vector to joint velocities. Then,

$$|\mathbf{v}_{c2}|^2 = |\mathbf{J}_{c2}\dot{\mathbf{q}}|^2 = \dot{\mathbf{q}}^T \mathbf{J}_{c2}{}^T \mathbf{J}_{c2}\dot{\mathbf{q}} \tag{7.2.6}$$

where $\dot{\mathbf{q}} = (\dot{\theta}_1 \quad \dot{\theta}_2)^T$. Substituting eqs.(4~6) to eq.(3) yields

$$T = \frac{1}{2}H_{11}\dot{\theta}_1^2 + H_{12}\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}H_{22}\dot{\theta}_2^2 = \frac{1}{2}(\dot{\theta}_1 \quad \dot{\theta}_2)^T \begin{pmatrix} H_{11} & H_{12} \\ H_{12} & H_{22} \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} \tag{7.2.7}$$

where coefficients H_{ij} are the same as the ones in eq.(7.1.12).

$$H_{11} = m_1\ell_{c1}^2 + I_1 + m_2(\ell_1^2 + \ell_{c2}^2 + 2\ell_1\ell_{c2}\cos\theta_2) + I_2 = H_{11}(\theta_2) \tag{7.1.12-a}$$

$$H_{22} = m_2\ell_{c2}^2 + I_2 \tag{7.1.12-b}$$

$$H_{12} = m_2(\ell_{c2}^2 + \ell_1\ell_{c2}\cos\theta_2) + I_2 = H_{12}(\theta_2) \tag{7.1.12-c}$$

Note that coefficients H_{11} and H_{12} are functions of θ_2.

The potential energy stored in the two links is given by

$$U = m_1 g \ell_{c1}\sin\theta_1 + m_2 g\{\ell_1\sin\theta_1 + \ell_{c2}\sin(\theta_1+\theta_2)\} \tag{7.2.8}$$

Now we are ready to obtain Lagrange's equations of motion by differentiating the above kinetic energy and potential energy. For the first joint,

$$\frac{\partial L}{\partial q_1} = -\frac{\partial U}{\partial q_1} = -[m_1\ell_{c1}\, g\cos\theta_1 + m_2\, g\{\ell_{c2}\cos(\theta_1+\theta_2) + \ell_1\cos\theta_1\}] = -G_1 \tag{7.2.9}$$

$$\begin{aligned} &\frac{\partial L}{\partial \dot{q}_1} = H_{11}\dot{\theta}_1 + H_{12}\dot{\theta}_2 \\ &\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} = H_{11}\ddot{\theta}_1 + H_{12}\ddot{\theta}_2 + \frac{\partial H_{11}}{\partial \theta_2}\dot{\theta}_2\dot{\theta}_1 + \frac{\partial H_{12}}{\partial \theta_2}\dot{\theta}_2{}^2 \end{aligned} \tag{7.2.10}$$

Substituting the above two equations into eq.(2) yields the same result as eq.(7.1.11-a). The equation of motion for the second joint can be obtained in the same manner, which is identical to eq.(7.1.11-b). Thus, the same equations of motion have been obtained based on Lagrangian Formulation. Note that the Lagrangian Formulation is simpler and more systematic than the Newton-Euler Formulation. To formulate kinetic energy, velocities must be obtained, but accelerations are not needed. Remember that the acceleration computation was complex in the Newton-Euler Formulation, as discussed in the previous section. This acceleration computation is automatically dealt with in the computation of Lagrange's equations of motion. The difference between the two methods is more significant when the degrees of freedom increase, since many workless constraint forces and moments are present and the acceleration computation becomes more complex in Newton-Euler Formulation.

7.2.3 Inertia Matrix

In this section we will extend Lagrange's equations of motion obtained for the two d.o.f. planar robot to the ones for a general *n* d.o.f. robot. Central to Lagrangian formulation is the derivation of the total kinetic energy stored in all of the rigid bodies involved in a robotic system. Examining kinetic energy will provide useful physical insights of robot dynamic. Such physical insights based on Lagrangian formulation will supplement the ones we have obtained based on Newton-Euler formulation.

As seen in eq.(3) for the planar robot, the kinetic energy stored in an individual arm link consists of two terms; one is kinetic energy attributed to the translational motion of mass m_i and the other is due to rotation about the centroid. For a general three-dimensional rigid body, this can be written as

$$T_i = \frac{1}{2} m_i \mathbf{v}_{ci}{}^T \mathbf{v}_{ci} + \frac{1}{2} \boldsymbol{\omega}_i{}^T \mathbf{I}_i \boldsymbol{\omega}_i , \qquad i = 1, \cdots, n \tag{7.2.11}$$

where $\boldsymbol{\omega}_i$ and $\mathbf{I}_i$ are, respectively, the 3x1 angular velocity vector and the 3x3 inertia matrix of the *i*-th link viewed from the base coordinate frame, i.e. inertial reference. The total kinetic energy stored in the whole robot linkage is then given by

$$T = \sum_{i=1}^{n} T_i \tag{7.2.12}$$

since energy is additive.

The expression for the kinetic energy is written in terms of the velocity and angular velocity of each link member, which are not independent variables, as mentioned in the previous section. Let us now rewrite the above equations in terms of an independent and complete set of generalized coordinates, namely joint coordinates $\mathbf{q} = [q_1, .. , q_n]^{\mathrm{T}}$. For the planar robot example, we used the Jacobian matrix relating the centroid velocity to joint velocities for rewriting the expression. We can use the same method for rewriting the centroidal velocity and angular velocity for three-dimensional multi-body systems.

$$\begin{aligned} \mathbf{v}_{ci} &= \mathbf{J}_i^L \dot{\mathbf{q}} \\ \boldsymbol{\omega}_i &= \mathbf{J}_i^A \dot{\mathbf{q}} \end{aligned} \tag{7.2.13}$$

where $\mathbf{J}^L{}_i$ and $\mathbf{J}^A{}_i$ are, respectively, the 3 x n Jacobian matrices relating the centroid linear velocity and the angular velocity of the i-th link to joint velocities. Note that the linear and angular velocities of the i-th link are dependent only on the first i joint velocities, and hence the last n-i columns of these Jacobian matrices are zero vectors. Substituting eq.(13) into eqs.(11) and (12) yields

$$T = \frac{1}{2}\sum_{i=1}^{n}(m_i \dot{\mathbf{q}}^T \mathbf{J}_i^{L^T} \mathbf{J}_i^L \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{J}_i^{A^T} \mathbf{I}_\mathbf{i} \mathbf{J}_i^A \dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{H}\dot{\mathbf{q}} \tag{7.2.14}$$

where **H** is a n x n matrix given by

$$\mathbf{H} = \sum_{i=1}^{n}(m_i \mathbf{J}_i^{L^T} \mathbf{J}_i^L + \mathbf{J}_i^{A^T} \mathbf{I}_i \mathbf{J}_i^A) \tag{7.2.15}$$

The matrix **H** incorporates all the mass properties of the whole robot mechanism, as reflected to the joint axes, and is referred to as the *Multi-Body Inertia Matrix.* Note the difference between the multi-body inertia matrix and the 3 x 3 inertia matrices of the individual links. The former is an aggregate inertia matrix including the latter as components. The multi-body inertia matrix, however, has properties similar to those of individual inertia matrices. As shown in eq. (15), the multi-body inertia matrix is a symmetric matrix, as is the individual inertia matrix defined by eq. (7.1.2). The quadratic form associated with the multi-body inertia matrix represents kinetic energy, so does the individual inertia matrix. Kinetic energy is always strictly positive unless the system is at rest. The multi-body inertia matrix of eq. (15) is positive definite, as are the individual inertia matrices. Note, however, that the multi-body inertia matrix involves Jacobian matrices, which vary with linkage configuration. Therefore the multi-body inertia matrix is *configuration-dependent* and represents the instantaneous composite mass properties of the whole linkage at the current linkage configuration. To manifest the configuration-dependent nature of the multi-body inertia matrix, we write it as $H(\mathbf{q})$, a function of joint coordinates **q**.

Using the components of the multi-body inertia matrix $\mathbf{H}=\{H_{ij}\}$, we can write the total kinetic energy in scalar quadratic form:

$$T = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} H_{ij}\dot{q}_i\dot{q}_j \tag{7.2.16}$$

Most of the terms involved in Lagrange's equations of motion can be obtained directly by differentiating the above kinetic energy. From the first term in eq.(2),

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{q}_i} = \frac{d}{dt}(\sum_{j=1}^{n} H_{ij}\dot{q}_j) = \sum_{j=1}^{n} H_{ij}\ddot{q}_j + \sum_{j=1}^{n}\frac{dH_{ij}}{dt}\dot{q}_j \tag{7.2.17}$$

The first term of the last expression, $\sum_{j=1}^{n} H_{ij}\ddot{q}_j$, comprises the diagonal term $H_{ii}\ddot{q}_i$ as well as off-diagonal terms $\sum_{i\neq j}^{n} H_{ij}\ddot{q}_j$, representing the dynamic interactions among the multiple joints due to accelerations, as discussed in the previous section. It is important to note that a pair of joints, i and j, have the same coefficient of the dynamic interaction, $H_{ij}=H_{ji}$, since the multi-body inertia matrix **H** is symmetric. In vector-matrix form these terms can be written collectively as

$$\mathbf{H}\ddot{\mathbf{q}} = \begin{array}{c} \\ i> \\ j> \\ \\ \end{array}\begin{pmatrix} H_{11} & \cdots & \cdots & H_{1n} \\ \vdots & \ddots & H_{ij} & \vdots \\ \vdots & H_{ji} & \ddots & \vdots \\ H_{n1} & \cdots & \cdots & H_{nn} \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_i \\ \dot{q}_j \\ \vdots \\ \dot{q}_n \end{pmatrix} \qquad (7.2.18)$$

It is clear that the interactive inertial torque $H_{ij}\dot{q}_j$ caused by the *j*-th joint acceleration upon the *i*-th joint has the same coefficient as that of $H_{ji}\dot{q}_i$ caused by joint *i* upon joint *j*. This property is called Maxwell's Reciprocity Relation.

The second term of eq.(17) is non-zero in general, since the multi-body inertia matrix is configuration-dependent, being a function of joint coordinates. Applying the chain rule,

$$\frac{dH_{ij}}{dt} = \sum_{k=1}^{n} \frac{\partial H_{ij}}{\partial q_k}\frac{dq_k}{dt} = \sum_{k=1}^{n} \frac{\partial H_{ij}}{\partial q_k}\dot{q}_k \qquad (7.2.19)$$

The second term in eq.(2), Lagrange's equation of motion, also yields the partial derivatives of H_{ij}. From eq.(16),

$$\frac{\partial T}{\partial q_i} = \frac{\partial}{\partial q_i}\left(\frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} H_{jk}\dot{q}_j\dot{q}_k\right) = \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{\partial H_{jk}}{\partial q_i}\dot{q}_j\dot{q}_k \qquad (7.2.20)$$

Substituting eq.(19) into the second term of eq.(17) and combining the resultant term with eq.(20), let us write these nonlinear terms as

$$h_i = \sum_{j=1}^{n}\sum_{k=1}^{n} C_{ijk}\dot{q}_j\dot{q}_k \qquad (7.2.21)$$

where coefficients C_{ijk} is given by

$$C_{ijk} = \frac{\partial H_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial H_{jk}}{\partial q_i} \qquad (7.2.22)$$

This coefficient C_{ijk} is called Christoffel's Three-Index Symbol. Note that eq.(21) is nonlinear, having products of joint velocities. Eq.(21) can be divided into the terms proportional to square joint velocities, i.e. *j=k,* and the ones for $j \neq k$: the former represents centrifugal torques and the latter Coriolis torques.

$$h_i = \sum_{j=1}^{n} C_{ijj}\dot{q}_j^{\ 2} + \sum_{k \neq j}^{n} C_{ijk}\dot{q}_j\dot{q}_k = \text{(Centrifugal)} + \text{(Coriolis)} \qquad (7.2.23)$$

These centrifugal and Coriolis terms are present only when the multi-body inertia matrix is configuration dependent. In other words, the centrifugal and Coriolis torques are interpreted as nonlinear effects due to the configuration-dependent nature of the multi-body inertia matrix in Lagrangian formulation.

7.2.4 Generalized Forces

Forces acting on a system of rigid bodies can be represented as conservative forces and non-conservative forces. The former is given by partial derivatives of potential energy U in Lagrange's equations of motion. If gravity is the only conservative force, the total potential energy stored in n links is given by

$$U = -\sum_{i=1}^{n} m_i \mathbf{g}^T \mathbf{r}_{0,ci} \tag{7.2.24}$$

where $\mathbf{r}_{0,ci}$ is the position vector of the centroid C_i that is dependent on joint coordinates. Substituting this potential energy into Lagrange's equations of motion yields the following gravity torque seen by the i-th joint:

$$G_i = \frac{\partial U}{\partial q_i} = -\sum_{j=1}^{n} m_j \mathbf{g}^T \frac{\partial \mathbf{r}_{0,cj}}{\partial q_i} = -\sum_{j=1}^{n} m_j \mathbf{g}^T \mathbf{J}_{j,i}^{L} \tag{7.2.25}$$

where $\mathbf{J}_{j,i}^{L}$ is the i-th column vector of the 3 x 1 Jacobian matrix relating the linear centroid velocity of the j-th link to joint velocities.

Non-conservative forces acting on the robot mechanism are represented by generalized forces Q_i in Lagrangian formulation. Let $\delta Work$ be virtual work done by all the non-conservative forces acting on the system. Generalized forces Q_i associated with generalized coordinates q_i, e.g. joint coordinates, are defined by

$$\delta Work = \sum_{i=1}^{n} Q_i \delta q_i \tag{7.2.26}$$

If the virtual work is given by the inner product of joint torques and virtual joint displacements, $\tau_1 \delta q_1 + \cdots + \tau_n \delta q_n$, the joint torque itself is the generalized force corresponding to the joint coordinate. However, generalized forces are often different from joint torques. Care must be taken for finding correct generalized forces. Let us work out the following example.

Example 7.2

Consider the same 2 d.o.f. planar robot as Example 7.1. Instead of using joint angles θ_1 and θ_2 as generalized coordinates, let us use the absolute angles, ϕ_1 and ϕ_2, measured from the positive x-axis. See the figure below. Changing generalized coordinates entails changes to generalized forces. Let us find the generalized forces for the new coordinates.

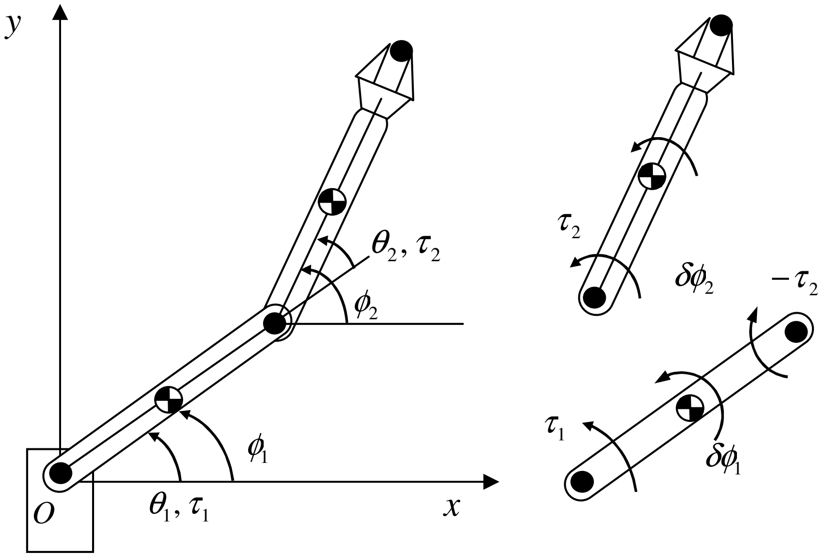


Figure 7.2.2 Absolute joint angles ϕ_1 and ϕ_2 and disjointed links

As shown in the figure, joint torque τ_2 acts on the second link, whose virtual displacement is $\delta\phi_2$, while joint torque τ_1 and the reaction torque $-\tau_2$ act on the first link for virtual displacement $\delta\phi_1$. Therefore the virtual work is

$$\delta Work = (\tau_1 - \tau_2)\delta\phi_1 + \tau_2\delta\phi_2 \tag{7.2.27}$$

Comparing this equation with eq.(26) where generalized coordinates are $\phi_1 = q_1, \phi_2 = q_2$, we can conclude that the generalized forces are:

$$Q_1 = \tau_1 - \tau_2, \quad Q_2 = \tau_2 \tag{7.2.28}$$

The two sets of generalized coordinates θ_1 and θ_2 vs. ϕ_1 and ϕ_2 are related as

$$\phi_1 = \theta_1, \quad \phi_2 = \theta_1 + \theta_2 \tag{7.2.29}$$

Substituting eq.(29) into eq.(27) yields

$$\delta Work = (\tau_1 - \tau_2)\delta\theta_1 + \tau_2\delta(\theta_1 + \theta_2) = \tau_1\delta\theta_1 + \tau_2\delta\theta_2 \tag{7.2.30}$$

This confirms that the generalized forces associated with the original generalized coordinates, i.e. joint coordinates, are τ_1 and τ_2.

Non-conservative forces acting on a robot mechanism include not only these joint torques but also any other external force $\mathbf{F}_{ext}$. If an external force acts at the endpoint, the generalized forces **Q=**$(Q_1,\ldots, Q_n)^T$ associated with generalized coordinates **q** are, in vector form, given by

$$\delta Work = \boldsymbol{\tau}^T\delta\mathbf{q} + \mathbf{F}_{ext}{}^T\delta\mathbf{p} = (\boldsymbol{\tau} + \mathbf{J}^T\mathbf{F}_{ext})^T\delta\mathbf{q} = \mathbf{Q}^T\delta\mathbf{q}$$

$$\mathbf{Q} = \boldsymbol{\tau} + \mathbf{J}^T\mathbf{F}_{ext} \tag{7.2.31}$$

When the external force acts at position **r**, the above Jacobian must be replaced by

$$\mathbf{J}_r = \frac{d\mathbf{r}}{d\mathbf{q}} \tag{7.2.32}$$

Note that, since generalized coordinates **q** can uniquely locate the system, the position vector **r** must be written as a function of **q** alone**.**

Chapter 9
Force and Compliance Controls

A class of simple tasks may need only trajectory control where the robot end-effecter is moved merely along a prescribed time trajectory. However, a number of complex tasks, including assembly of parts, manipulation of tools, and walking on a floor, entail the control of physical interactions and mechanical contacts with the environment. Achieving a task goal often requires the robot to comply with the environment, react to the force acting on the end-effecter, or adapt its motion to uncertainties of the environment. Strategies are needed for performing those tasks.

Force and compliance controls are fundamental task strategies for performing a class of tasks entailing the accommodation of mechanical interactions in the face of environmental uncertainties. In this chapter we will first present *hybrid position/force control*: a basic principle of strategic task planning for dealing with geometric constraints imposed by the task environment. An alternative approach to accommodating interactions will also be presented based on *compliance or stiffness control*. Characteristics of task compliances and force feedback laws will be analyzed and applied to various tasks.

9.1 Hybrid Position/Force Control

9.1.1 Principle

To begin with let us consider a daily task. Figure 9.1.1 illustrates a robot drawing a line with a pencil on a sheet of paper. Although we humans can perform this type of task without considering any detail of hand control, the robot needs specific control commands and an effective control strategy. To draw a letter, "A", for example, we first conceive a trajectory of the pencil tip, and command the hand to follow the conceived trajectory. At the same time we accommodate the pressure with which the pencil is contacting the sheet of paper. Let *o-xyz* be a coordinate system with the *z*-axis perpendicular to the sheet of paper. Along the *x* and *y* axes, we provide positional commands to the hand control system. Along the *z*-axis, on the other hand, we specify a force to apply. In other words, controlled variables are different between the horizontal and vertical directions. The controlled variable of the former is *x* and *y* coordinates, i.e. a position, while the latter controlled variable is a force in the *z* direction. Namely, two types of control loops are combined in the hand control system, as illustrated in Figure 9.1.2.

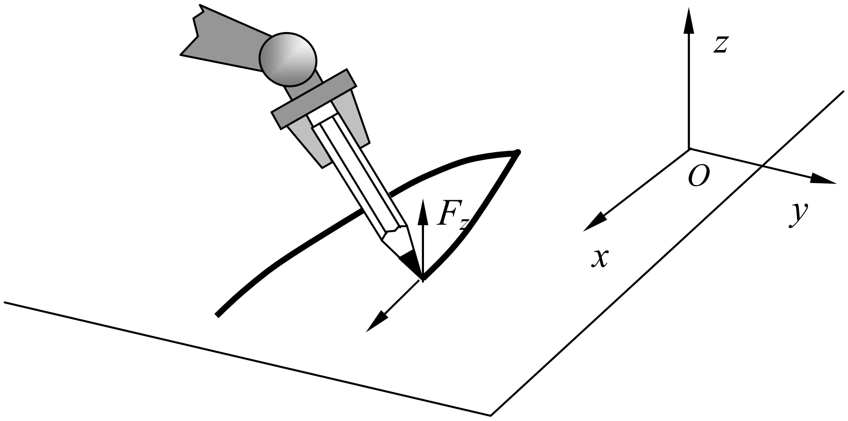


Figure 9.1.1 Robot drawing a line with a pencil on a sheet of paper

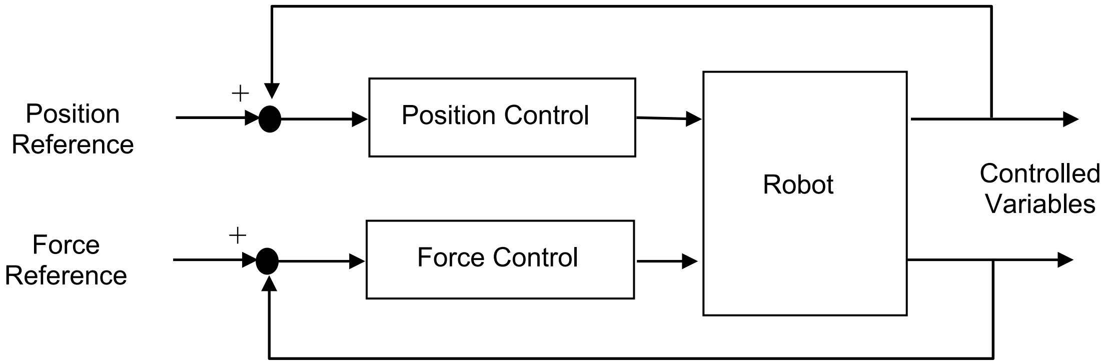


Figure 9.1.2 Position and force control loops

The above example is one of the simplest tasks illustrating the need for integrating different control loops in such a way that the control mode is consistent with the geometric constraint imposed to the robot system. As the geometric constraint becomes more complex and the task objective is more involved, an intuitive method may not suffice. In the following we will obtain a general principle that will help us find proper control modes consistent with both geometric constraints and task objectives. Let us consider the following six-dimensional task to derive a basic principle behind our heuristics and empiricism.

Example 9.1

Shown below is a task to pull up a peg from a hole. We assume that the peg can move in the vertical direction without friction when sliding in the hole. We also assume that the task process is quasi-static in that any inertial force is negligibly small. A coordinate system *O-xyz*, referred to as *C-frame*, is attached to the task space, as shown in the figure. The problem is to find a proper control mode for each of the axes: three translational and three rotational axes.

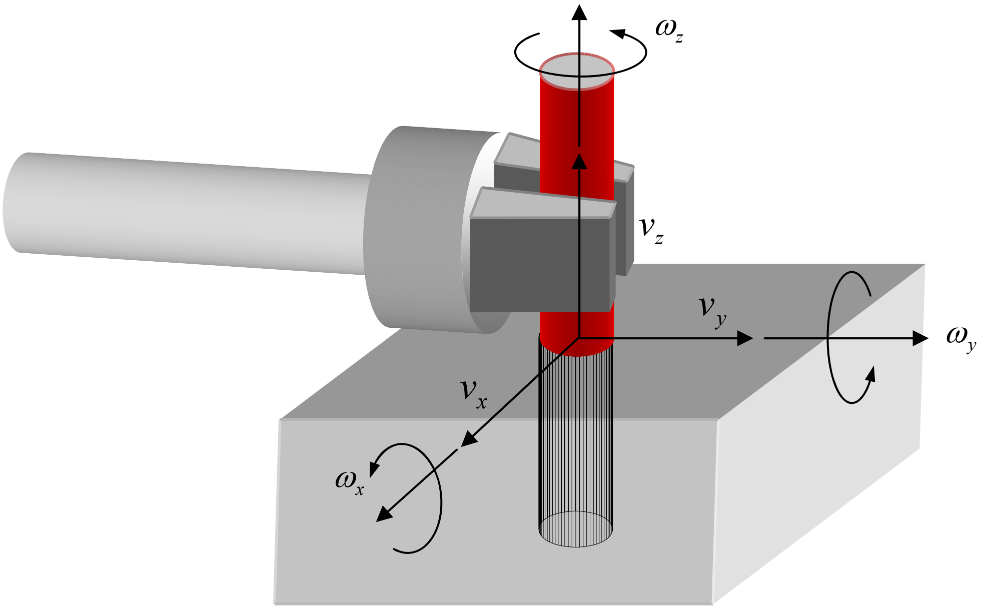


Figure 9.1.3 Pulling up a peg from a hole

The key question is how to assign a control mode, position control or force control, to each of the axes in the C-frame in such a way that the control action may not conflict with the geometric constraints and physics. M. Mason addressed this issue in his seminal work on hybrid position/force control. He called conditions dictated by physics *Natural Constraints*, and conditions determined by task goals and objectives *Artificial Constraints.* Table 9.1.1 summarizes these conditions.

From Figure 9.1.3 it is clear that the peg cannot be moved in the *x* and *y* directions due to the geometric constraint. Therefore, the velocities in these directions must be zero: $v_x = 0,\ v_y = 0$. Likewise, the peg cannot be rotated about the *x* and *y* axes. Therefore, the angular velocities are zero: $\omega_x = 0,\ \omega_y = 0$. These conditions constitute the natural constraints in the kinematic domain. The remaining directions are linear and angular *z* axes. Velocities along these two directions can be assigned arbitrarily, and may be controlled with position control mode. The reference inputs to these position control loops must be determined such that the task objectives may be satisfied. Since the task is to pull up the peg, an upward linear velocity must be given: $v_z = V > 0$. The orientation of the peg about the z-axis, on the other hand, doesn't have to be changed. Therefore, the angular velocity remains zero: $\omega_z = 0$. These reference inputs constitute the artificial constraints in the kinematic domain.

Table 9.1.1 Natural and artificial constraints of the peg-in-the-hole problem

	Kinematic	Static
Natural Constraints	$v_x = 0$ $v_y = 0$ $\omega_x = 0$ $\omega_y = 0$	$f_z = 0$ $\tau_z = 0$
Artificial Constraints	$v_z = V > 0$ $\omega_z = 0$	$f_x = 0$ $f_y = 0$ $\tau_x = 0$ $\tau_y = 0$

In the statics domain, forces and torques are specified in such a way that the quasi-static condition is satisfied. This means that the peg motion must not be accelerated with any unbalanced force, i.e. non-zero inertial force. Since we have assumed that the process is friction-less, no resistive force acts on the peg in the direction that is not constrained by geometry. Therefore, the linear force in the *z* direction must be zero: $f_z = 0$. The rotation about the *z* axis, too, is not constrained. Therefore, the torque about the *z* axis must be zero: $\tau_z = 0$. These conditions are dictated by physics, and are called the natural constraints in the statics domain. The remaining directions are geometrically constrained. In these directions, forces and torques can be assigned arbitrarily, and may be controlled with force control mode. The reference inputs to these control loops must be determined so as to meet the task objectives. In this task, it is not required to push the peg against the wall of the hole, nor twist it. Therefore, the reference inputs are set to zero: $f_x = 0, f_y = 0, \tau_x = 0, \tau_y = 0$. These constitute the artificial constraints in the statics domain.

In the above example, it is clear that the axes involved in the natural constraints and the artificial constraints are orthogonal to each other in both kinematic and static domains. Moreover, the axes involved in the natural kinematic constraints and the artificial static constraints are the same, and the ones listed in the natural static constraints and the artificial kinematic constraints are the same. These relationships are rather obvious in the above example where the direction of each C-frame axis is aligned with the direction along which each control mode, position or force, is assigned. If such a C-frame exists, these orthogonality properties are simply the consequence of the following assumptions and rule:

- Each C-frame axis must have only one control mode, either position or force,
- The process is quasi-static and friction less, and
- The robot motion must conform to geometric constraints.

In general, the axes of a C-frame are not necessarily the same as the direction of a separate control mode. Nevertheless, the orthogonality properties hold in general. We prove this next.

Let V^6 be a six-dimensional vector space, and $V_a \subset V^6$ be an admissible motion space, that is, the entire collection of admissible motions conforming to the geometric constraints involved in a given task. Let V_c be a constraint space that is the orthogonal complement to the admissible motion space:

$$V_c = V_a^{\perp} \tag{9.1.1}$$

Let $\mathbf{F} \in V^6$ be a six-dimensional endpoint force acting on the end-effecter, and $\Delta\mathbf{p} \in V^6$ be an infinitesimal displacement of the end-effecter. The work done on the end-effecter is given by

$$\Delta Work = \mathbf{F}^T \Delta\mathbf{p} \tag{9.1.2}$$

Decomposing each vector to the one in the admissible motion space and the one in the constraint space,

$$\begin{aligned} &\mathbf{F} = \mathbf{F}_a + \mathbf{F}_c; \qquad &\mathbf{F}_a \in V_a, \mathbf{F}_c \in V_c \\ &\Delta\mathbf{p} = \Delta\mathbf{p}_a + \Delta\mathbf{p}_c; \quad &\Delta\mathbf{p}_a \in V_a, \Delta\mathbf{p}_c \in V_c \end{aligned} \tag{9.1.3}$$

and substituting them to eq.(2) yield

$$\begin{aligned} \Delta Work &= (\mathbf{F}_a + \mathbf{F}_c)^T(\Delta\mathbf{p}_a + \Delta\mathbf{p}_c) = \mathbf{F}_a^{\ T}\Delta\mathbf{p}_a + \mathbf{F}_a^{\ T}\Delta\mathbf{p}_c + \mathbf{F}_c^{\ T}\Delta\mathbf{p}_a + \mathbf{F}_c^{\ T}\Delta\mathbf{p}_c \\ &= \mathbf{F}_a^{\ T}\Delta\mathbf{p}_a + \mathbf{F}_c^{\ T}\Delta\mathbf{p}_c \end{aligned} \tag{9.1.4}$$

since $\mathbf{F}_a \perp \Delta\mathbf{p}_c, \quad \mathbf{F}_c \perp \Delta\mathbf{p}_a$ by definition. For the infinitesimal displacement $\Delta\mathbf{p}$ to be a virtual displacement $\delta\mathbf{p}$, its component in the constraint space must be zero: $\Delta\mathbf{p}_c = 0$. Then, $\Delta\mathbf{p}_a = \delta\mathbf{p}$ becomes a virtual displacement, and eq.(4) reduces to virtual work. Since the system is in a static equilibrium, the virtual work must vanish for all virtual displacements $\delta\mathbf{p}_a$.

$$\delta Work = \mathbf{F}_a{}^T \delta\mathbf{p}_a = 0, \quad \forall \delta\mathbf{p}_a \tag{9.1.5}$$

This implies that any force and moment in the admissible motion space must be zero, i.e. the natural static constraints:

$$0 = \mathbf{F}_a \in V_a \tag{9.1.6}$$

Furthermore, the properties of artificial static constraints can be derived from eqs.(4) and (5). Since in eq.(4) $\Delta\mathbf{p}_c = 0$, the static equilibrium condition holds, although $\mathbf{F}_c \in V_c$ takes an arbitrary value. This implies that to meet a task goal we can assign arbitrary values to the force and moment in the constraint space, i.e. the artificial static constraints.

$$arbitrary : \mathbf{F}_c \in V_c \tag{9.1.7}$$

Converting infinitesimal displacements to velocities, $\dot{\mathbf{p}}_a, \dot{\mathbf{p}}_c$, we can obtain the natural and artificial kinematic constraints:

$$\begin{aligned} &arbitrary : \dot{\mathbf{p}}_a \in V_a, \\ &0 = \dot{\mathbf{p}}_c \in V_c \end{aligned} \tag{9.1.8}$$

Table 9.1.2 summarizes the above results.

Table 9.1.2 Mason's Principle of Hybrid Position/Force Control

	Kinematic	Static
Natural Constraints	$0 = \dot{\mathbf{p}}_c \in V_c$	$0 = \mathbf{F}_a \in V_a$
Artificial Constraints	$arbitrary : \dot{\mathbf{p}}_a \in V_a$	$arbitrary : \mathbf{F}_c \in V_c$

The reader will appreciate Mason's Principle when considering the following exercise problem, in which the partition between admissible motion space and constraint space cannot be described by a simple division of *C*-frame axes. Rather the admissible motion space lies along an axis where a translational axis and a rotational axis are coupled.

Exercise 9.2 (The same as PS)

9.1.2 Architecture of Hybrid Position/Force Control System

Based on Mason's Principle, a hybrid position/force control system can be constructed in such a way that the robot control system may not have a conflict with the natural constraints of the task process, while performing the task towards the task goal. Figure 9.1.5 shows the block diagram of a hybrid position/force control system. The upper half of the diagram is position control loops, where artificial kinematic constraints are provided as reference inputs to the system and are compared with the actual position of the end-effecter. The lower half of the diagram is force control loops, where artificial static constraints are provided as reference inputs to the feedback loops and are compared with the actual force and moment at the end-effecter. Note that feedback signals are described in an appropriate *C*-frame attached to the end-effecter.

If the feedback signals are noise-less and the *C*-frame is perfectly aligned with the actual task process, the position signal of the feedback loop must lie in the admissible motion space, and the force being fed back must lie in the constraint space. However, the feedback signals are in general corrupted with sensor noise and the *C*-frame may be misaligned. Therefore, the position signal may contain some component in the constraint space, and some fraction of the force signal may be in the admissible motion space. These components are contradicting with the natural constraints, and therefore should not be fed back to the individual position and force controls. To filter out the contradicting components, the feedback errors are projected to their own subspaces, i.e. the positional error $\mathbf{e}_p$ mapped to the admissible motion space V_a and the force feedback error $\mathbf{e}_f$ mapped to the constraint space V_c. In the block diagram these filters are shown by projection matrices, $\mathbf{P}_a$ and $\mathbf{P}_c$:

$$\widehat{\mathbf{e}}_p = \mathbf{P}_a \mathbf{e}_p, \quad \widehat{\mathbf{e}}_f = \mathbf{P}_c \mathbf{e}_f \tag{9.1.9}$$

When the C-frame axes are aligned with the directions of the individual position and force control loops, the projection matrices are diagonal, consisting of only 1 and 0 in the diagonal components. In the case of the peg-in-the-hole problem, they are:

$$\mathbf{P}_a = diag(1\,1\,0\,1\,1\,0), \quad \mathbf{P}_c = diag(0\,0\,1\,0\,0\,1) \tag{9.1.10}$$

In case of Example 9.2 where the C-frame axes are not the directions of the individual position and force control loops, the projection matrices are not diagonal.

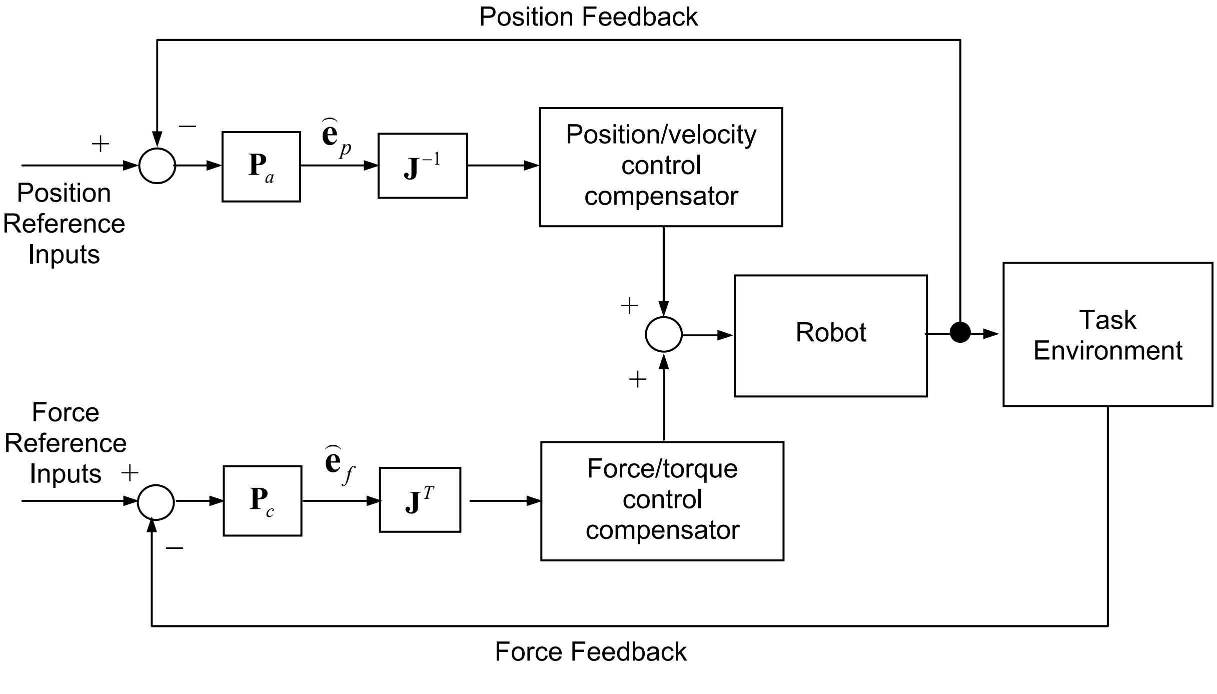


Figure 9.1.4 Block diagram of hybrid position/force control system

These feedback errors, $\widehat{\mathbf{e}}_p$ and $\widehat{\mathbf{e}}_f$, are in the C-frame, hence they must be converted to the joint space in order to generate control commands to the actuators. Assuming that the positional error vector is small and that the robot is not at a singular configuration, the position feedback error in joint coordinates is given by

$$\mathbf{e}_q = \mathbf{J}^{-1}\widehat{\mathbf{e}}_p \tag{9.1.11}$$

where **J** is the Jacobian relating the end-effecter velocities in the C-frame to joint velocites. The force feedback error in the joint coordinates, on the other hand, is obtained based on the duality principle:

$$\mathbf{e}_\tau = \mathbf{J}^T\widehat{\mathbf{e}}_f \tag{9.1.12}$$

These two error signals in the joint coordinates are combined after going through dynamic compensation in the individual joint controls.

9.2 Compliance Control

9.2.1 Task strategy

Use of both position and force information is a unique feature in the control of robots physically interacting with the environment. In hybrid position/force control, separation was made explicitly between position and force control loops through projections of feedback signals onto admissible motion space and constraint space. An alternative to this space separation architecture is to control a *relationship* between position and force in the task space. *Compliance Control* is a basic control law relating the displacement of the end-effecter to the force and moment acting on it. Rather than totally separating the task space into subspaces of either position or force control, compliance control reacts to the endpoint force such that a given functional relationship, typically a linear map, is held between the force and the displacement. Namely, a functional relationship to generate is given by

$$\Delta\mathbf{p} = \mathbf{CF} \tag{9.2.1}$$

where **C** is an *m* x *m* Compliance Matrix, and $\Delta\mathbf{p}$ and $\mathbf{F}$ are endpoint displacement and force represented in an *m*-dimensional, task coordinate system. Note that the inverse to the compliance matrix is a stiffness matrix:

$$\mathbf{K} = \mathbf{C}^{-1} \tag{9.2.2}$$

if the inverse exists.

The components of the compliance matrix, or the stiffness matrix, are design parameters to be determined so as to meet task objectives and constraints. Opening a door, for example, can be performed with the compiance illustrated in Figure 9.2.1. The trajectory of the doorknob is geometrically constrained to the circle of radius R centered at the door hinge. The robot hand motion must comply to the constrained doorknob trajectory, although the trajectory is not exactly known. The robot must not break the doorknob, although the conceived trajectory is different from the actual trajectory. This task requirement can be met by assigining a small stiffness, i.e. a high compliance, to the radial direction perpendicular to the trajectory. As illustrated in the figure, such a small spring constant generates only a small restoring force in response to the discrepancy between the actual doorknob trajectory and the reference trajectory of the robot hand. Along the direction tangent to the doorknob trajectory, on the other hand, a large stiffness, or a small compliance, is assigned. This is to force the doorknob to move along the trajectory despite friction and other resistive forces. The stiffness matrix is therefore given by

$$\mathbf{K} = \begin{pmatrix} k_x & 0 \\ 0 & k_y \end{pmatrix}; \quad k_x << 1, k_y >> 1 \tag{9.2.3}$$

with reference to the task coordinate system *O-xy*. Using this stiffness with which the doorknow is held, the robot can open the door smoothly and dexterously, although the exact trajectory of the doorknob is not known.

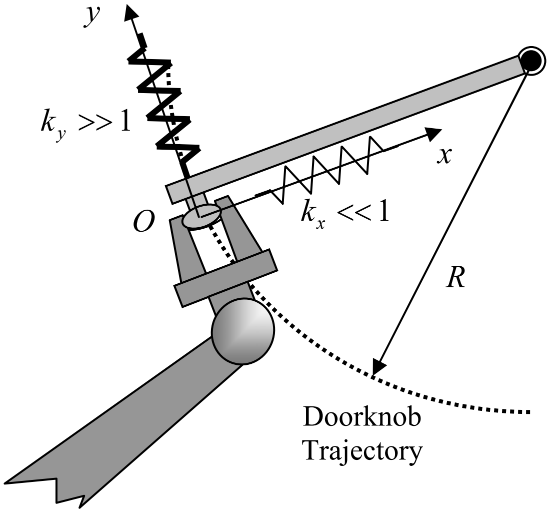


Figure 9.2.1 Door opening with compliance control

9.2.2 Compliance control synthesis

Now that a desired compliance is given, let us consider the method of generating the desired compliance. There are multiple ways of synthesizing a compliance control system. The simplest method is to accommodate the proportional gains of joint feedback controls so that desired restoring forces are generated in proportion to discrepancies between the actual and reference joint angles. As shown in Figure 9.2.2, a feedback control error e_i is generated when a disturbance force or torque acts on the joint. At steady state a ststic balance is made, as an actuator torque τ_i proportional to the control error e_i cancels out the disturbance torque. The proportionality constant is determined by the position feedback gain k_i, when friction is neglected. Therefore a desired stiffness or compliance can be obtained by tuning the position feedback gain.

Compliance synthesis is trivial for single joint control systems. For general *n* degree-of-freedom robots, however, multiple feedback loops must be coordinated. We now consider how to generate a desired *m x m* compliance or stiffness *matrix* specified at the endpoint by tuning joint feedback gains.

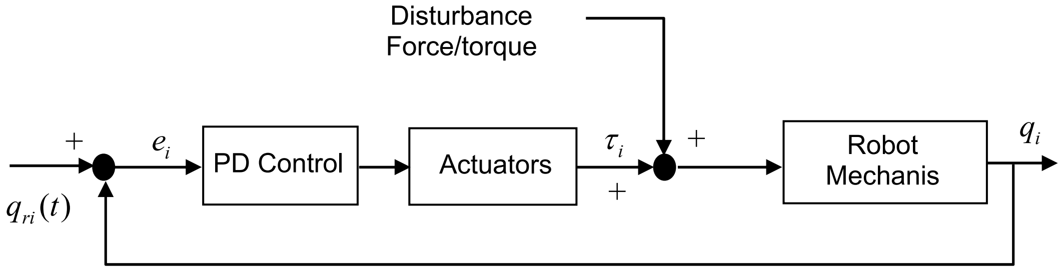


Figure 9.2.2 Single joint position feedback control system

Theorem Let **J** be the Jacobian relating endpoint velocity $\dot{\mathbf{p}} \in R^{m\,x1}$ to joint velocities $\dot{\mathbf{q}} \in R^{n\,x1}$, and $\boldsymbol{\tau} \in R^{n\,x1}$ be joint torques associated with joint coordinates **q**. Let $\Delta\mathbf{p} \in R^{m\,x1}$ be a *m* x 1 vector of the endpoint displacement measured from a nominal position $\overline{\mathbf{p}}$, and $\mathbf{F} \in R^{m\,x1}$ be the endpoint force associated with the endpoint displacement $\Delta\mathbf{p}$. Let $\mathbf{K_p}$ be a desired endpoint stiffness matrix defined as:

$$\mathbf{F} = \mathbf{K}_p \Delta\mathbf{p} \tag{9.2.4}$$

The necessary condition for joint feedback gain $\mathbf{K_q}$ to generate the endpoint stiffness $\mathbf{K_p}$ is given by

$$\mathbf{K}_q = \mathbf{J}^T \mathbf{K}_p \mathbf{J} \tag{9.2.5}$$

assuming no friction at the joints and linkage mechanisms.

Proof

Using the Jacobian and the duality principle as well as eq.(4),

$$\boldsymbol{\tau} = \mathbf{J}^T\mathbf{F} = \mathbf{J}^T\mathbf{K}_p\Delta\mathbf{p} = \mathbf{J}^T\mathbf{K}_p\mathbf{J}\cdot\Delta\mathbf{q} \tag{9.2.6}$$

Using eq.(5), the above relationship reduces to

$$\boldsymbol{\tau} = \mathbf{K}_q\Delta\mathbf{q} \tag{9.2.7}$$

This implies that $\mathbf{K}_q$ is the joint feedback gain matrix.

Example 9.2.1 Consider a two-link, planar robot arm with absolute joint angles and joint torques, as shown in Figure 9.2.3. Obtain the joint feedback gain matrix producing the endpoint stiffness $\mathbf{K}_p$:

$$\mathbf{K}_p = \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix} \tag{9.2.8}$$

Assuming that the link length is 1 for both links, the Jacobian is given by

$$\mathbf{J} = \begin{pmatrix} -s_1 & -s_2 \\ c_1 & c_2 \end{pmatrix} \tag{9.2.9}$$

From eq.(5),

$$\mathbf{K}_q = \mathbf{J}^T \mathbf{K}_p \mathbf{J} = \begin{pmatrix} k_{q1} & k_{q3} \\ k_{q3} & k_{q2} \end{pmatrix} \tag{9.2.10}$$

where

$$\begin{aligned} k_{q1} &= k_1 s_1^2 + k_2 c_1^2 \\ k_{q2} &= k_1 s_2^2 + k_2 c_2^2 \\ k_{q3} &= k_1 s_1 s_2 + k_2 c_1 c_2 \end{aligned} \tag{9.2.11}$$

Note that the joint feedback gain matrix $\mathbf{K}_q$ is symmetric and that the matrix $\mathbf{K}_q$ degenerates when the robot is at a singular configuration. If it is non-singular, then

$$\Delta\mathbf{p} = \mathbf{J}\Delta\mathbf{q} = \mathbf{J}\mathbf{K}_q^{-1}\boldsymbol{\tau} = \mathbf{J}\mathbf{K}_q^{-1}\mathbf{J}^T\mathbf{F} = \mathbf{J}(\mathbf{J}^T\mathbf{K}_p\mathbf{J})^{-1}\mathbf{J}^T\mathbf{F} = \mathbf{K}_p^{-1}\mathbf{F} = \mathbf{C}\mathbf{F} \tag{9.2.12}$$

Therefore, the obtained joint feedback gain provides the desired endpoint stiffness given by eq.(8), or the equivalent compliance.

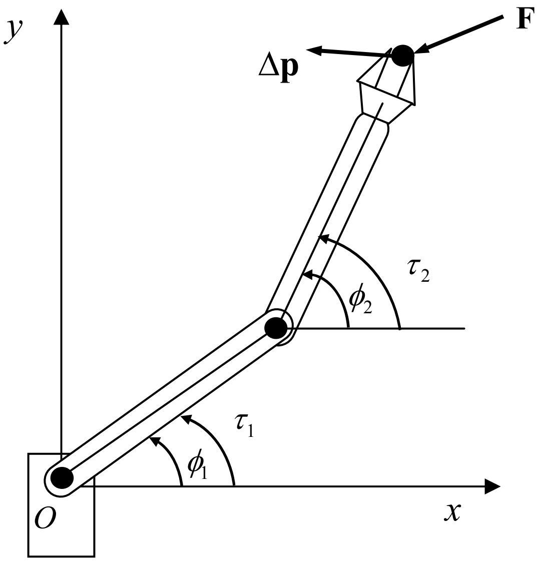


Figure 9.2.3 Two link robot