

Chapter 8

Kinematic Design



Fady Youssef and Sebastian Kassner

Abstract One aspect in haptic devices is the design of the kinematics. The kinematics of a mechanism is the key to implement and accomplish the design goals, like transmitting dynamic feedback in the form of forces or torques, or allowing a sufficient workspace for the user to interact with environment. This chapter introduces the steps of the kinematic design. The chapter consists of five main sections. The first section gives an overview on some basic definitions and the main types of mechanisms. In the second section, the first step in the design, defining the structure of the mechanism, is introduced. This is accompanied with an example. After choosing the most applicable structure for the desired application, the second step takes place, where the kinematic equations are solved. These equations are used to describe the relation between the operating point of the mechanism and the base at any point in time. Different approaches are used to solve those equations depending on the type of the mechanism used. The third and final step in the design process is introduced in the fourth section. This step contains the optimization process of the mechanism in order to achieve a desired operation of the mechanism. Last but not least, the importance of modeling and simulation is discussed in the last section.

8.1 Introduction

The introduction to the topic of kinematic design begins with mentioning the major goals behind the kinematic design. Then, some basic definitions are introduced, followed by a classification of the mechanisms used in haptic interfaces. Finally an introduction to the design steps is given.

F. Youssef (✉)
Hamburg University of Technology, Eißendorfer Str. 38, 21073 Hamburg, Germany
e-mail: f.youssef@tuhh.de

S. Kassner
Knorr-Bremse Systeme für Schienenfahrzeuge GmbH, Moosacher Str. 80, 80809 München, Germany
e-mail: sebastian.kassner@kassner-net.de

© The Author(s) 2023
T. A. Kern et al. (eds.), *Engineering Haptic Devices*, Springer Series on Touch and Haptic Systems, https://doi.org/10.1007/978-3-031-04536-3_8

8.1.1 Major Design Goals

Kinematics has a big role in haptic devices. It's the mechanical interface between the user and the environment. Different design goals are available depending on the application. In some applications, the goal of the design is to transmit dynamic feedback, e.g. forces or torques, to the user. Other applications require sufficient workspace to ensure the interaction between the user and the environment.

8.1.2 Basic Definitions

In the area of haptic interfaces, some definitions are valid apart from the type of the mechanism used.

Kinematics

Kinematics is the branch in mechanics that studies the motion of points and bodies in space in terms of position, velocity and acceleration without taking into consideration the cause of this motion, e.g forces or torques.

Dynamics

Dynamics, sometimes referred as kinetics, is the branch in mechanics that studies the motion in space with the cause of the motion is considered.

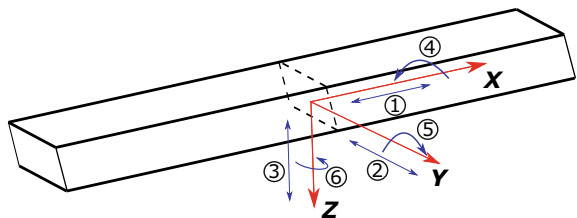
Degree of Freedom

Degrees of freedom are the number of independent motions a body or a mechanism is able to carry out. A free body has 3 DoF in 2D, two translations and one rotation, while in 3D has 6 DoF, three translations and three rotations. Figure 8.1 shows the possible independent motions a body can do in space.

Joints

Joints are used to connect two or more bodies together. Depending on the type of the joint used, the number of allowed relative DoF between the connected bodies is defined. The commonly used joints allow either 1,2 or 3 DoF. Figure 8.2 shows several types of joints.

Fig. 8.1 Independent motions of a free body in space. 3 translations and 3 rotations along and about the 3 axes (x , y , z)



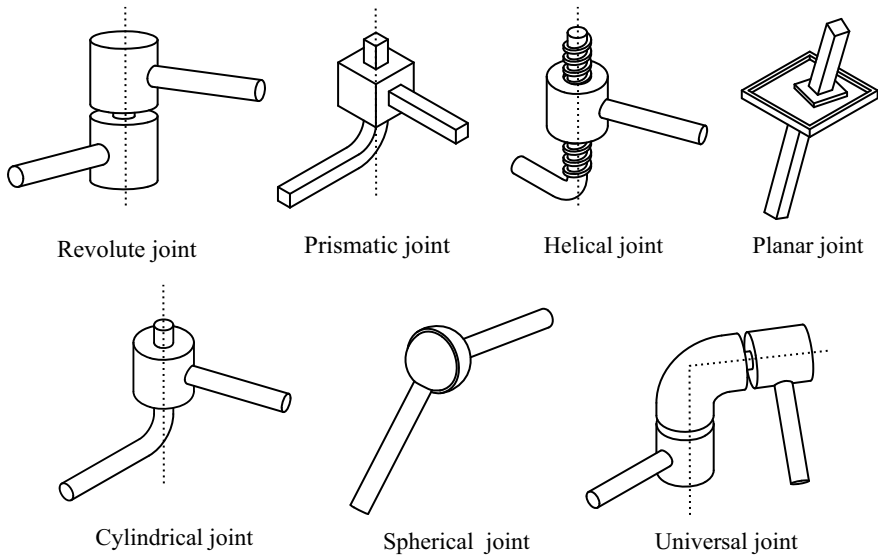


Fig. 8.2 Different types of joints. 1. DoF: Revolute (R), Prismatic (P), Helical (H). 2. DoF: Planar (E), Cylindrical (C), and Universal (U). 3. DoF: Spherical (S)

Active Joints

These are the actuated joints in a mechanism.

Passive Joints

These are the non-actuated joints in a mechanism.

Base

The base is the reference platform of the mechanism. All the calculations are performed relative to this platform. Positions, velocities, and accelerations of any point on the mechanism are given with respect to the base.

Tool Center Point

TCP is the point where the user, or the environment, interacts with the mechanism. Usually it's the end-point of the mechanism.

Workspace

The workspace is the set of all positions in space the TCP can reach.

Singularity

Position in the mechanism's workspace where the mechanism loses the control of one or more DoF. This can be seen in having non-solvable equations. Singularities will be further discussed in Sect. 8.4.2.

Translational Parallel Machine

TPM is a mechanism whose TCP is allowed only to move in Cartesian space (x, y, z) .

8.1.3 Classification of Mechanisms

There are three main configurations of mechanisms used in haptic interfaces: Serial, Parallel and Hybrid. This chapter will focus on both serial and parallel mechanisms. Figures 8.3 and 8.4 shows the different configurations of mechanisms used.

Serial Mechanisms

Serial mechanisms are open kinematic chains, in other words there is only one path from the base platform to the TCP. A typical serial mechanism consists of only active joints connected with rods (links). Usually one DoF joints are used. The number of intended DoF of the TCP defines the number of joints in the mechanism. Figure 8.5 shows UR10E from the company *Universal Robots*. The robot has 6 active revolute joints. As can be seen from the figure, there is only one path from the base to the TCP. The joints are connected in series. The advantages of serial mechanism are their simple design, their relatively large workspace and their relatively easy control especially in positioning tasks. On the other hand, the major disadvantage of serial mechanisms is that one actuator carries the load of the all the next actuators. In

Fig. 8.3 Basic elements of a mechanism

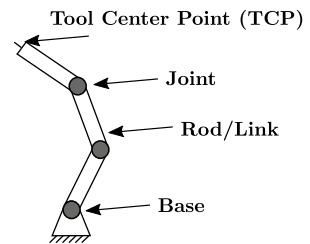


Fig. 8.4 Basic kinematic structures

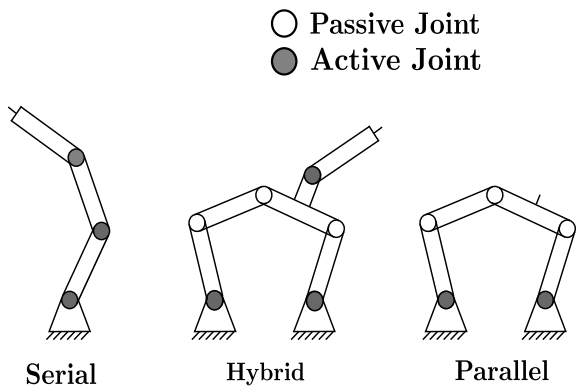


Fig. 8.5 Example of Serial mechanism, UR10E from the company *Universal Robots*

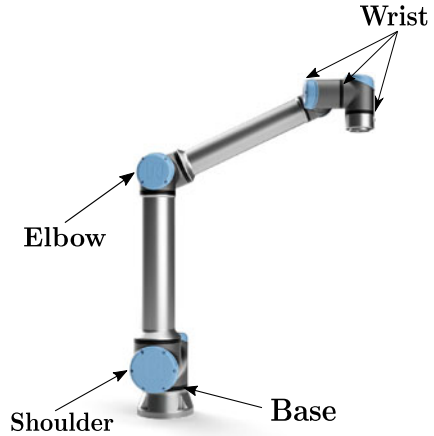
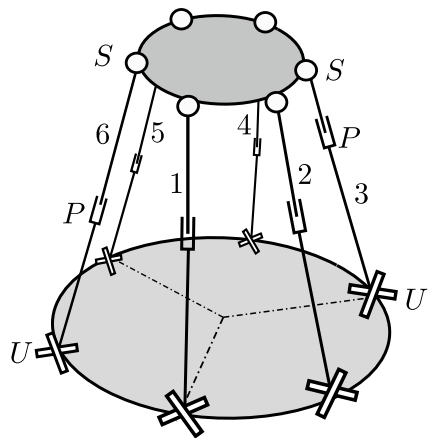


Fig. 8.6 STEWART- GOUGH mechanism



other words, any actuator should overcome its own inertia, the inertia of all the next actuators in the chain, and finally the load acting on the TCP. This drawback affects the dynamic behaviour of serial mechanisms. This drawback can reflect on the mechanism to have an overall low structural stiffness with respect to its own weight.

Parallel Mechanisms

Parallel mechanisms are closed chains mechanisms. In parallel mechanisms, there are at least two paths from the base platform to the TCP. A typical parallel mechanism consists of both active and passive joints. The number of active joints is defined by the intended DoF of the TCP. The most famous parallel mechanism is the STEWART-GOUGH platform Fig. 8.6. Another example of parallel mechanism, OMEGA6 haptic device from the company *Force Dimension*, is shown in Fig. 8.7. OMEGA6 is a pen-shaped force-feedback device. From the figure, one can see that there are three paths

Fig. 8.7 OMEGA6 haptic device, © 2022 Force Dimension, used with permission



from the base to the TCP. The main advantage of parallel mechanisms is that the load on the TCP is distributed on multiple kinematic chains, this will lead to a lightweight design, yet a high structural stiffness. The same advantage leads to high transparent transmission of haptic feedback, which is the reason why this configuration is of a great importance in haptic interfaces. On the other hand, the main disadvantages are small workspace and relative complexity of solving the kinematic equations compared to serial mechanisms, in addition to singular positions which is discussed later in this chapter in Sect. 8.4.2.

Hybrid Mechanisms

Hybrid mechanisms are a combination of both serial and parallel mechanisms. It contains both open and closed chains. The most well known example of a hybrid mechanism is the Tricept (Fig. 8.8). It is composed of parallel part that create the translation in the workspace, followed by a serial part in order to create the orientation of the TCP in space [19]. Decoupling the serial part from the parallel part simplifies the design of the mechanism. Hybrid mechanisms stands in between the serial and parallel mechanisms in terms of advantages and disadvantages; they can have a lightweight design compared to pure serial mechanisms and a larger workspace compared to pure parallel mechanisms. In this chapter we will focus on pure serial and pure parallel mechanisms.

8.1.4 Design Steps

Designing the kinematic mechanism of haptic interfaces passes through three steps:

- Defining the structure (Sect. 8.2): In this step, the type of mechanism, the number of appropriate DoF of the joints are defined. This is done based on the application.

Fig. 8.8 TRICEPT T606 parallel kinematic, © 2022 PKMtricept, used with permission



- Solving the kinematic equations (Sect. 8.3): In this step the basic foundation for controlling the mechanism is defined. These equations are used later in the last design step.
- Finally dimensioning the mechanism (Sect. 8.4): In this step, the optimization of the mechanism in terms of lengths is done. The optimization process is done based on the intended tasks.

If a serial, a parallel or a hybrid mechanism is suitable for the design of a haptic interface should be decided on a case-by-case basis. All are used in haptic applications.

8.2 Design Step 1: Defining the Mechanism's Structure

The first step in designing the mechanism is the definition of the structure. It leads to the basic configuration of joints, rods and actuators. While the basic structure of the haptic interface is defined in this step, the topological synthesis has to be carried out very thoroughly.

The topological synthesis should be based on an analysis of the specific task. At least the following issues should be addressed:

- Degrees of freedom: In how many DoF should the user interact with the haptic interface? Which DoF are required (e.g. one pure rotatory as in a jog wheel, three to mimic spatial interaction or even six to display three translations and three rotations)?
- Adaption of existing structures: Should the device adapt the structure of the task (e.g. a controlled robot) or of the user (e.g. the user's finger or arm)?

- **Workspace:** How large is the desired workspace, the TCP has to move in? Are there any restrictions (e.g. areas of the workspace which should not be accessed)?
- **Mobility:** Is the haptic interface designed as a device which is standing on a fixed place, e.g. on a table, or is designed as a portable device?

The analysis of these requirements lays the foundation for the design of an easy-to-use and ergonomic haptic interface which will be accepted by the user.

8.2.1 *Synthesis of Serial Mechanisms*

A serial mechanism is not less or more than a sequence of rods and actuators, whereas the actuators can be regarded as driven (active) joints. Whether the actuators are linear or rotary is of no importance for the complexity of the kinematic problem. For the workspace and the orientation of the TCP however this aspect is of highest importance.

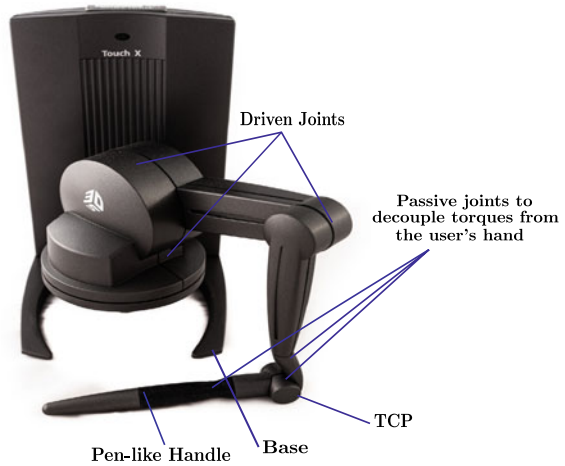
A widely used design in serial kinematics is to split the joints into two groups: the first group is responsible for the translation of the TCP, and the second group is responsible for the orientation of the TCP. In Fig. 8.5, one can see that the base, shoulder, and elbow joints are mainly responsible for the translation, while the three joints of the wrist are responsible for the orientation.

If it is not intended to generate a torque as output to the user, the handle attached to this serial mechanism has to be equipped with a passive universal joint. Such a realisation as haptic device can be found in Fig. 8.9, the torques are decoupled from the hand. The handle does not have to be placed exactly in the TCP, as the moments are eliminated by the passive joints. Force vectors can be displaced arbitrarily within space. As a result the hand experiences the same forces like the TCP.

As human beings are equipped with many serial kinematic chains (e.g. arms, legs) the working area of a serial kinematic chain can be understood intuitively. This makes it simple to design a corresponding haptic control unit. This is however not the only criterion and will be further addressed in Sect. 8.4. The design can be done geometrically “with circle and ruler”, however the following should be considered:

- **Actuators add inertia and moving masses to the mechanism.** In serial mechanisms all actuators are placed in series. This has negative influence on the dynamics of force transmission. Approaches for a dynamic analysis will be discussed later in this chapter in Sect. 8.4.4.
- **A simple design criterion could be, to place actuators as near as possible to the base platform of the system and use transmission elements to the points/joints of actuation.**

Fig. 8.9 TOUCHX haptic device from the company 3D Systems



8.2.2 Synthesis of Parallel Mechanisms

The synthesis of a parallel mechanism in general is a less intuitive process than the synthesis of a serial mechanism.

Since a parallel structure comprises several kinematic chains, the first step is to determine the required number of kinematic chains with respect to the desired degrees of freedom of the mechanism. This can be done using the ratio of the number of chains k and the degrees of freedom F of the mechanism leading to the degree of parallelism [7].

$$P_g = \frac{k}{F} \tag{8.1}$$

A mechanism is considered fully parallel, most common case, if $P_g = 1$. Partially parallel mechanism has $P_g < 1$, while highly parallel mechanism has $P_g > 1$. This means that for a fully parallel mechanism, the number of chains (legs) is equal to the desired number of DoF of the mechanism.

As mentioned earlier, parallel mechanisms consist of both active and passive joints. The relation between the joints (active and passive), and mechanism's DoF is done using the GRUEBLER- KUTZBACH- CHEBYCHEFF mobility criterion:

$$F = \lambda \cdot (n - g - 1) + \sum_{i=1}^g f_i - f_{id} + s \tag{8.2}$$

where:

- F Mechanism's DoF
- λ Spatial factor: 3 for 2D and 6 for 3D mechanisms
- n Number of bodies

- g Number of joints
- f_i DoF of the i joint
- f_{id} Sum of identical condition
- s Sum of constraints

An identical link is given for example when a rod has universal joints at both of its ends. The rod will be able to rotate around its axis, without violating any constraints. Another example are two coaxial oriented linear joints.

Constraints appear whenever conditions have to be fulfilled to enable the movement. If five joint axes have to be parallel to a (6th) axis to enable a movement, then $s = 5$. Another example for a passive condition are two driving rods which have to be placed in parallel to enable a motion.

At this stage of the design, Eq. (8.2) can't be applied directly, as n and g aren't known yet. There exists a correlation between the number of chains (legs) k , joints g and elements n is defined according to:

$$n = g - k + 2 \quad (8.3)$$

Assuming spatial mechanism with no identical conditions and no constraints, applying Eqs. (8.3) in (8.2), the total number of joints' DoF to be distributed are:

$$\sum_{i=1}^g f_i = F + 6 \cdot (k - 1) \quad (8.4)$$

As a simple rule of thumb:

- Concentrate joint DoF: Universal joints and Spherical joints simplify the design and the transfer characteristics compared to serially placed joints with just one DoF each. Especially the kinematic transfer characteristics are much easier to formulate with concentrated joints.
- Usage of actuators fixed to the frame: With the actuators being fixed to the frame the dynamic properties of the design would be increased, as their mass isn't included in the inertia of the device anymore.
- DoF should be distributed symmetrically: A symmetrical design should be preferred compared to an asymmetrical one. The kinematic transfer functions get significantly simpler.

8.2.2.1 Special Case: Parallel Mechanisms with Pure Translation Motion

An important task of many haptic interfaces is the displaying pure three-dimensional spatial sensation. An example is the interaction with a pen-like tool where only forces in (x, y, z) should be displayed to the user. A special class of 3-DoF parallel mechanisms is used for those applications is TPM. This is achieved by kinematic

chains which are blocking one or more rotatory DoF of the TCP and being able to perform translational motion in all directions.

According to CARRICATO [2, 3] two restrictions have to be fulfilled to ensure a parallel kinematic mechanism with pure translational motion:

- Spherical joints shall not be used.
- The axis of rotation of rotatory joints shall not be parallel to the axis of a degree of freedom which should be constrained.

Neglecting over-determined configurations, this results in so-called T5-mechanisms, each comprising four or five rotatory joints. Each joint constrains the rotation of the TCP about one axis. More details are found in [2, 3].

8.2.2.2 Example: DELTA Mechanism

One of the most common topologies to display spatial interaction is the parallel DELTA mechanism (Fig. 8.7). Due to its relevance in the field of haptic interfaces it is used as an example for the topological synthesis. Let us assume the design goal of a parallel kinematic haptic interface for a spatial interaction in (x, y, z) . Thus a mechanism with three degrees of freedom is required. Using Eq. (8.1) for a fully parallel mechanism $P_g = 1$ on $F = 3$ haptic degrees of freedom leads to a mechanism with $k = 3$ kinematic chains or legs.

In a second step we have to determine the the required joint degrees of freedom using GRUEBLER'S formula (Eq. (8.2)). This leads to the sum of $\sum_{i=1}^g f_i = 15$ joint degrees of freedom.

Regarding an equal behavior in all spatial directions it is self-evident to distribute the 15 joint degrees of freedom with five degrees in each leg. This leads to the topologies in Table 8.1. The topologies are denominated according to the joints in one leg starting from the base of the mechanism to the TCP, e.g. a UUP mechanism comprises of a universal joint, followed by another universal joint, and finally one prismatic joint. The selection of an appropriate topology then can be carried on by a systematic reduction of the 3-DoF topologies in Table 8.1. The reduction is based on the following criteria:

- Functionality as a TPM: Criteria like the number of R-joints or the existence of a S-joint eliminate a large number of topologies.
- Position of actuators: Rotatory, linear or piston actuators (e.g. in a hydraulic system) act as R-, P- or C-joints. When having topologies with an U-joint attached to the base platform this would lead to actuators which are located within the kinematic chain. The required acceleration to move the actuators with relatively high masses then would inhibit that the dynamic advantages of a parallel mechanism have the fullest effect.
- Number of joints: A concentration of two R-joints into one U-joint and a R- and P-joint in into a C-joint respectively simplifies the mechanisms geometry and thereby its kinematic equations.

Table 8.1 Topologies for 3-DoF mechanisms with 5 DoF in each leg

Joints per leg	Topologies
1 × 1 DoF, 2 × 2 DoF	UUP, UPU, PUU, UUR, URU, RUU, CUP, CPU, CUR, CRU, RCU, UCP, UPC, PCU, UCR,URC, RUC, CCP, CPC, PCC, CCR, CRC, RCC
2 × 1 DoF, 1 × 3 DoF	SPP, SRR, SPR, SRP, PSP, RSP, PSR, RSR, PPS, RRS, RPS, PRS
3 × 1 DoF, 1 × 2 DoF	RRRU, RRUR, RURR, URRR, RRPU, RRUP, RURP, URRP, RPRU, RPUR, RUPR, URPR, URPR, PRUR, PURR, UPRR, RPPU, RPUP, RUPP, URPP, PRPU, PRUP, PURP, UPRP, PPPU, PPUP, PUPP, UPPP, RRRC, RRCR, RCRR, CRRR, RRPC, RRCP, RCRP, CRRP, RPRC, RPCR, RCPR, CRPR, PRRC, PRCR, PCRR, CPRR, RPPC, RPCP, RCPP, PRPC, PCRP, PPPC,PPCP, PCPP, CPPP
5 × 1 DoF	32 iterations of P- and R-joints

Taking into account the above mentioned criteria, the remaining configurations are: UPU, PUU, CUR, CRU, RUU and RUC (Fig. 8.10). Table 8.2 shows the eliminated topologies based on the different criteria. Looking carefully at these topologies in Fig. 8.10, one recognizes that only RUU and RUC have rotatory joint attached to the base platform. Thus these are the only two topologies that can be reasonably driven by a rotatory electrical motor. What makes the RUU (DELTA) mechanism special is that there are only joints with rotatory degrees of freedom within the kinematic chains. All forces and torques are converted into rotatory motion and there is no chance for the mechanism to cant. DELTA mechanisms have singular positions within the workspace. This has to be considered when dimensioning the mechanism Sect. 8.4. The RUU/DELTA was introduced in 1988 by CLAVEL [4]. Besides acting as a spatial haptic interface (Fig. 8.7), the mechanism is kind of widely used in robotic applications (e.g. pick-and-place tasks). In these devices with mainly kinaesthetic feedback, a mechanical mechanism is used to link the user and the feedback generating actuators. Furthermore the user’s input commands are often given by moving a mechanical mechanism.

8.3 Design Step 2: Kinematic Equations

The second step in designing a mechanism is finding the relation between the base and the TCP at any point in time. This is done by solving the kinematic equations. There are two main types of kinematic equations; forward kinematic and inverse kinematic. Before addressing the kinematic equations, some basic definitions should be introduced.

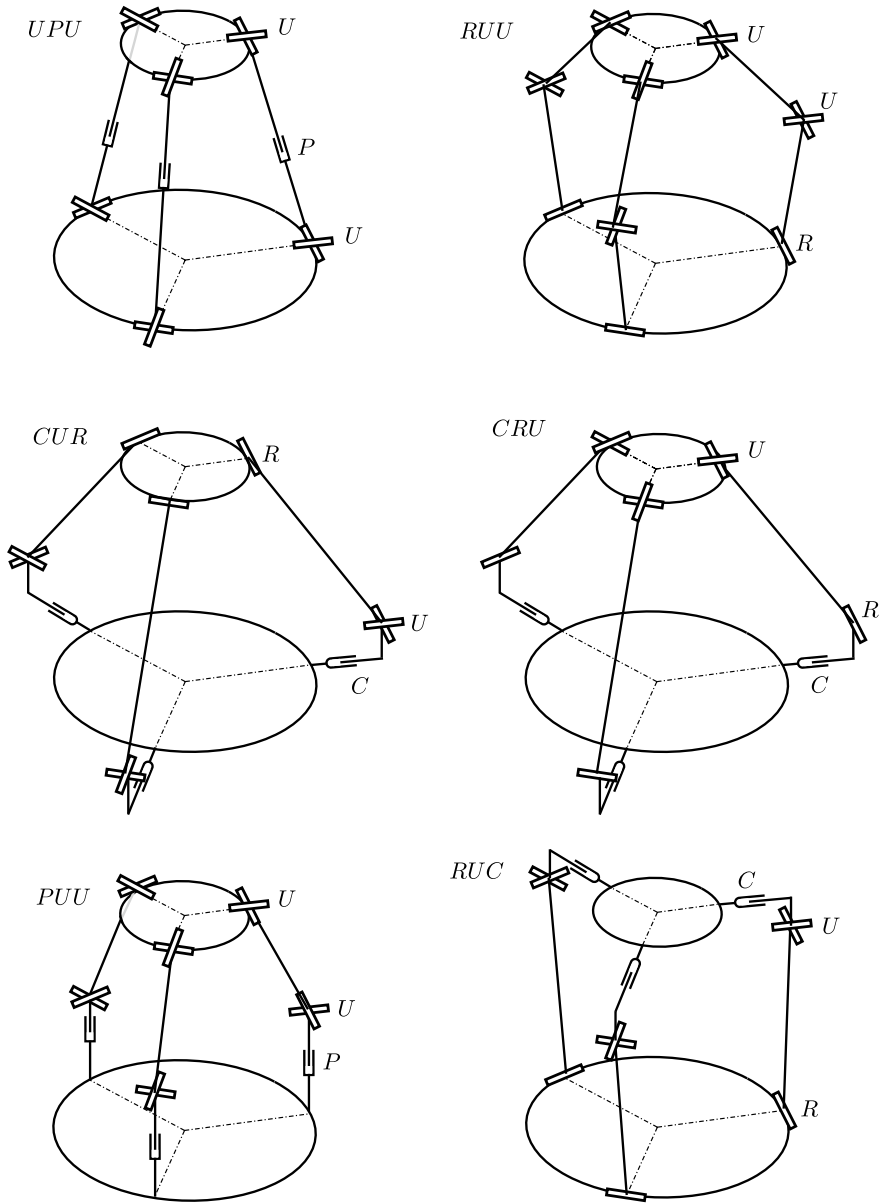


Fig. 8.10 Possible TPM mechanisms

Table 8.2 Eliminated topologies, sorted by the distribution of the 5 DoF in each leg

Elimination criterion	5×1 DoF	3×1 DoF, 1×2 DoF	2×1 DoF, 1×3 DoF	1×1 DoF, 2×2 DoF
No TPM	RRRRP, RRPRR, RRPPR, RRPPP, RPRRR, RPRRP, RPRPR, RPRPP, RPPRR, RPPRP, RPPPR, RPPPP, PRRRR, PRRRP, PRRPR, PRRPP, PRPRR, PRPRP, PRPPR, PPRRP, PPRPR, PPRPP, PPPRR, PPPRP, PPPPR, PPPPP, PRPPP	RPPU, RPUP, RUPP, URPU, PURP, UPRP, PPPU, PPUP, UPP, UPPP, RRPC, RRCP, RCRP, CRRP, RPRC, RPCR, RCPR, CRPR, PRRC, PRCR, PCRR, CPRR, RPPC, RPCP, RCPP, CRPP, PRPC, PRCP, PCRP, CPRP, PPPC, PPCP, PCPP, CPPP	SPP, SRR, SPR, SRP, PSP, RSP, PSR, RSR, PPS, RRS, RPS, PRS	CUP, CPU, RCU, UCP, UPC, PCU, PUC, UCR, CCP, CPC, PCC, CCR, CRC, RCC
High number of joints	RRRRR, RRRPR, RRPRP, PRRRR	RRRU, RRUR, RURR, RRRU, RRUP, RURP, RPRU, RPUR, RURP, PRRU, PRUR, PURR, UPRR, RRRC, RRCR, RCRR, CRRR		
Base joint can't be used as an actuator		URRR, URRP, URPR		UUP, UUR, URU, URC

Forward Kinematics

Forward kinematics is defined as giving the joints' angles/positions $q = (q_1, q_2, \dots, q_n)$ as input and calculating the pose (position and orientation) $p = (p_1, p_2, \dots, p_m)$ of the TCP.

$$p = f(q)$$

In serial kinematics, solving the forward kinematics is usually done analytically. On the other hand, for parallel mechanisms the direct kinematic problem can only be solved numerically. However, there are exceptions that can be seen later in this chapter.

An important application of the forward kinematic problem is the calculation of an input command in impedance controlled devices.

Inverse Kinematics

Inverse kinematics is the opposite to the forward kinematics. The pose of the TCP is given, and the joints' angles/positions are calculated.

$$q = f^{-1}(p)$$

Geometric, algebraic, and numerical methods are used to solve the inverse kinematics problem. The method used depends on the type of mechanism. Numerical methods can be applied to any type of mechanisms. Inverse kinematics in parallel mechanisms is usually easier to calculate compared to serial mechanisms.

In admittance controlled devices, inverse kinematics is used to calculate the required evasive movement in order to regulate a desired contact force between user and the haptic interface.

Coordinate Frames

Coordinate frame i , Fig. 8.11, or simply frame i , is composed of an origin O_i and three mutually orthogonal base vectors $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$, that is fixed to a particular body [22]. The pose of each body (rod) in a mechanism is always expressed relative to another body. In other words, the pose can be expressed as the relation between two frames, each frame is stick to one body. The pose consists of two parts, position and rotation (orientation). In a mechanism, the most two important frames are the tool and base frames. The pose of the TCP, or any frame inside the mechanism, is usually given relative to the base frame.

Position Vector

Position vector is the vector connecting the origins of two frames. The 3×1 position vector of frame j relative to frame i is given as:

$${}^i p_j = \begin{bmatrix} {}^i p_j^x \\ {}^i p_j^y \\ {}^i p_j^z \end{bmatrix}$$

The components of this vector are the Cartesian coordinates of O_j in frame i . This gives the translation between the two origins.

Rotation Matrix

Orientation of frame j relative to frame i is expressed using rotation matrix. A rotation matrix is 3×3 . It is composed as follows:

$${}^i R_j = \begin{bmatrix} \hat{x}_j \cdot \hat{x}_i & \hat{y}_j \cdot \hat{x}_i & \hat{z}_j \cdot \hat{x}_i \\ \hat{x}_j \cdot \hat{y}_i & \hat{y}_j \cdot \hat{y}_i & \hat{z}_j \cdot \hat{y}_i \\ \hat{x}_j \cdot \hat{z}_i & \hat{y}_j \cdot \hat{z}_i & \hat{z}_j \cdot \hat{z}_i \end{bmatrix}$$

For example, a simple rotation of frame j around \hat{z}_i only by an angle θ (Fig. 8.12) gives the following rotation matrix:

Fig. 8.11 Example of an arbitrary coordinate frame

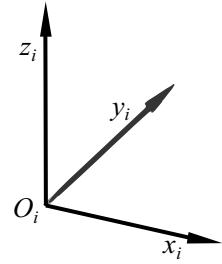
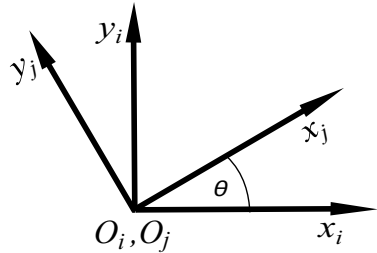


Fig. 8.12 Simple rotation of one frame around an axis of another frame



$${}^iR_j = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The different representations of multiple rotations can be found in [22].

Homogeneous Transformation Matrix

With homogeneous transformations, position vectors and rotation matrices are combined together in a compact notation. Homogeneous transformation matrix is a 4×4 matrix, and is given as follows:

$${}^iT_j = \begin{bmatrix} {}^iR_j & {}^ip_j \\ 0 & 1 \end{bmatrix}$$

The matrix iT_j transforms vectors from frame j to coordinate frame i . Its inverse ${}^jT_i = {}^iT_j^{-1}$ transforms vectors from coordinate frame i to frame j .

Remarks:

- Matrix multiplications are associative, but they are not commutative. Thus the order for multiplication is highly important for the calculations, and specially in the forward kinematics.
- The numbers (0, 1) in the last row of the matrix guarantee that rotations and translations do not influence each other. With this feature a simple algorithm can perform

rotations and translations with a single matrix multiplication. This increases the clarity of an implementation and may be one reason why homogeneous coordinate transformations are widespread within robotics and even virtual reality programming.

8.3.1 Solving Kinematic Equations in Serial Mechanisms

In order to solve the forward kinematic equations in serial mechanisms, the DH convention is used. This convention was introduced by JACQUES DENAVIT and RICHARD HARTENBERG in 1955.

Denavit-Hartenberg Convention

In [5, 10, 22] the different variants of the DH convention, proximal and distal, are well differentiated. The convention is based on attaching frames to each link in the mechanism and performing two translations and two rotations to jump from one frame to the next one. Regardless the variant used, there are common steps as follow:

1. Defining and attaching a coordinate frame on each link (rod) according to the variant used, starting from the base to the TCP. The frames should be oriented in such way that frame $i + 1$ can be reached from frame i after performing the four operations, two translations and two rotations.
2. Composing the DH-table using the DH-parameters (θ, d, a, α) . These parameters represent the four operations (e.g. Table 8.4).
3. Formulation of the homogeneous transformation matrix ${}^i T_{i+1}$ that relates frame $i+1$ to frame i . Each transformation matrix represents a row in the DH-table.
4. Multiplying all the transformation matrices to calculate the total transformation matrix that relates the TCP to the base, ${}^{base} T_{TCP}$.

$${}^{base} T_{TCP} = {}^{base} T_1 \cdot {}^1 T_2 \cdot \dots \cdot {}^{n-1} T_n \cdot {}^n T_{TCP} \quad (8.5)$$

${}^{base} T_{TCP}$ gives the pose of the TCP in the base frame. The matrix is function of the active joints' values and the dimensions of the links. Substituting with the actuators angles/positions gives the TCP pose in result to the given set of values.

Inverse Kinematics in Serial Mechanisms

There are multiple approaches to solve the inverse kinematics problem in serial mechanisms. Generally, the inverse kinematics problem is nonlinear. A lot of questions rise when solving the inverse kinematics such as, whether there is a solution at all or the existence of multiple solutions. The two main approaches are closed-form and numerical solutions.

Pieper [20] introduced an approach to solve the inverse kinematics (closed-form) of a six DoF serial manipulator, where three axes meet at one point.

8.3.1.1 Example: UR10E

UR10E (Fig. 8.5) is one example of serial mechanisms used in haptic interfaces. The forward kinematics will be discussed in this part. The proximal variant (Modified DH convention) is used in this analysis. In the proximal variant:

- The frames start from the base as frame 0.
- The total number of frames is $n + 1$, where is n the number of links.
- The definition of the four parameters are different than in the classic convention as seen in Table 8.3.
- The transformation matrix relating any two frames $i - 1$ and i is given as follows:

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\ \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -d_i \sin \alpha_{i-1} \\ \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & d_i \cos \alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.6)$$

The robotic arm has six revolute DoF, so we have a total seven frames and total of six rows of the DH-table (Table 8.4). The frames are given in Fig. 8.13. The next step is formulating the six transformation matrices as follow:

Table 8.3 Definitions of Modified DH parameters

Parameter	Description
a_{i-1}	Distance from Z_{i-1} to Z_i along X_{i-1}
α_{i-1}	Angle from Z_{i-1} to Z_i about X_{i-1}
d_i	Distance from X_{i-1} to X_i along Z_i
θ_i	Angle from X_{i-1} to X_i about Z_i

Table 8.4 DH Table of UR10E

i	a_{i-1} (m)	α_{i-1} (rad)	d_i (m)	θ_i (rad)
1	0	0	$d_1 = 0.1807$	θ_1
2	0	$\pi/2$	0	θ_2
3	$a_2 = -0.6127$	0	0	θ_3
4	$a_3 = -0.57155$	0	$d_4 = 0.17415$	θ_4
5	0	$\pi/2$	$d_5 = 0.11985$	θ_5
6	0	$-\pi/2$	$d_6 = 0.11655$	θ_6

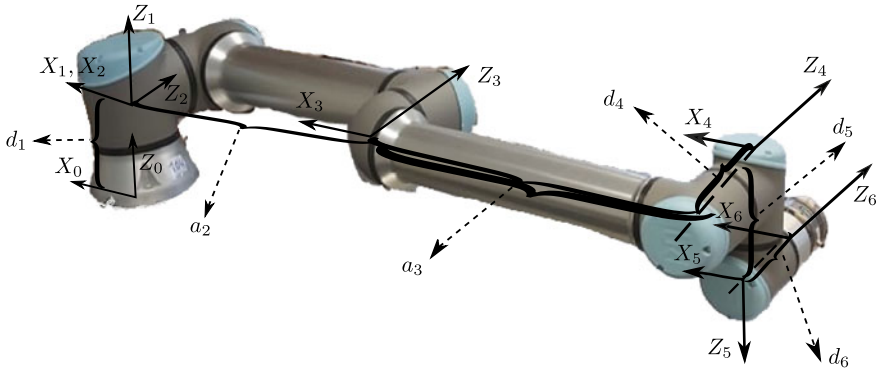


Fig. 8.13 Coordinate frames of UR10E according to modified DH convention

$${}^0T_1 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1T_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

⋮

$${}^5T_6 = \begin{bmatrix} \cos \theta_6 & -\sin \theta_6 & 0 & 0 \\ 0 & 0 & -1 & d_6 \\ -\sin \theta_6 & \cos \theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The total transformation matrix is given as follows:

$${}^0T_6 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6 \tag{8.7}$$

The closed-form approach to solve the inverse kinematics of this robot is discussed in details in [11]. The numerical approach will be discussed later in this chapter in Sect. 8.5.3.

8.3.2 Solving Kinematic Equations in Parallel Mechanisms

Solving kinematic equations in parallel mechanisms is somehow different compared to serial mechanisms. The main goal remains the same, to get the relation between the pose of the TCP and the values of the joints' angles/positions. The presence of both active and passive joints adds complexity to the kinematic equations. Also, if the joints aren't distributed equally on all chains (legs), the kinematics gets more complicated.

Forward Kinematics

In contrast to serial mechanisms, for parallel mechanisms the direct kinematic problem can only be solved numerically. However, there are exceptions as can be seen later. As mentioned earlier, the STEWART- GOUGH platform (Fig. 8.6) is one of the most famous parallel mechanisms. Solving the forward kinematics of this platform may end with 40 possible solutions [21] and [16]. Many approaches were introduced to solve the kinematics problem in general, like elimination [9], interval analysis [14], continuation [21]. Recently, other algorithms were introduced to cope with the real-time constraints, such as, using Neural networks [18], or using the information from the inverse kinematics and the small changes in the motion of the TCP [24].

Inverse Kinematics

The procedure of calculating the inverse kinematic problem can be split up into the following three steps:

1. Formulation of closed vector chains for each leg, starting at the coordinate system enclosing the TCP and going back to the reference coordinate system, usually the base.
2. Splitting the vector chains in all Cartesian movement directions of the individual leg.
3. Solving the resulting system of equations according to the TCP coordinates.

8.3.2.1 Example: RUU/DELTA Mechanism

TPM is a special case in parallel mechanisms. Solving the forward and inverse kinematics is somehow not complicated. One example of TPM is the RUU/DELTA mechanism.

Forward Kinematics

Figure 8.14 shows the necessary dimensions and angles to derive the kinematic equations. It is desired to express all these equations with respect to the world frame in the middle of the base platform. The x axis points towards the first leg. A local frame $(x_{A_i}, y_{A_i}, z_{A_i})$ with the origin A_i is fixed at the first joint of the i^{th} leg. This local coordinate system is rotated by $\phi_i = (i - 1) \cdot 120$ degrees, with $i = 1, 2, 3$

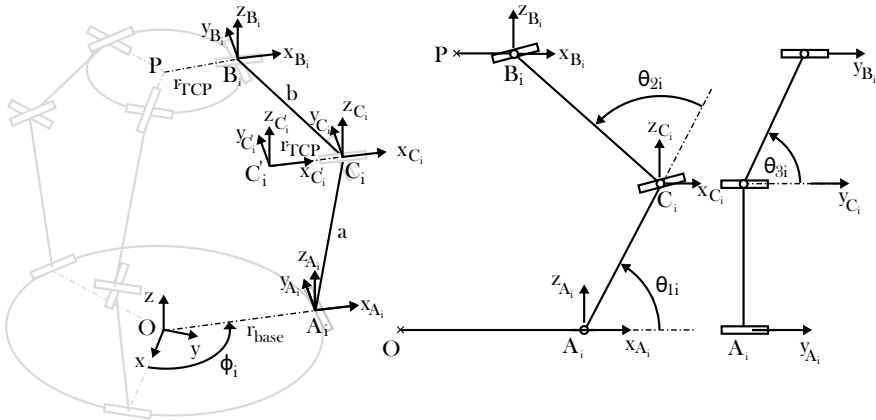


Fig. 8.14 Coordinate frames of DELTA mechanism according to [23]

with respect to the world frame. The transformation between the base frame and the A_i frame is as follows:

$${}^{base}T_{A_i} = \begin{bmatrix} \cos(-\phi_i) & \sin(-\phi_i) & 0 & 0 \\ -\sin(-\phi_i) & \cos(-\phi_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & r_{base} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.8)$$

$${}^{base}T_{A_i} = \begin{bmatrix} \cos(-\phi_i) & \sin(-\phi_i) & 0 & r_{base} \cdot \cos(-\phi_i) \\ -\sin(-\phi_i) & \cos(-\phi_i) & 0 & -r_{base} \cdot \sin(-\phi_i) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.9)$$

The transformation is a rotation around the z axis in the world frame by angle $-\phi_i$ and then a translation in the x_i direction by distance r_{base} .

Another frame is attached to the point C_i . This frame has the same orientation as the A_i frame, the relative position is dependent on the angle θ_{1i} . The transformation between these two frames is:

$${}^{A_i}T_{C_i} = \begin{bmatrix} 1 & 0 & 0 & a \cdot \cos \theta_{1i} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a \cdot \sin \theta_{1i} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.10)$$

As mentioned above the forward kinematic problem in general cannot be solved for parallel kinematic mechanisms. In case of the DELTA mechanism it is different. Here the method of trilateration can be applied. This approach is based on the fact that, if looking at one leg, all points B_i are on the surface of a sphere with radius b and the

center point C_i . The surface is given by sphere equation:

$$(x - x_{C_i})^2 + (y - y_{C_i})^2 + (z - z_{C_i})^2 = b^2 \quad (8.11)$$

with the center coordinates $(x_{C_i}, y_{C_i}, z_{C_i})$ of the sphere.

In order to use the trilateration method more easily, a virtual frame C'_i will be placed with a distance of $-r_{TCP}$ along the x axis of frame C_i . The transformation between the two frames is:

$${}^{C_i}T_{C'_i} = \begin{bmatrix} 1 & 0 & 0 & -r_{TCP} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.12)$$

As mentioned before, the usage of homogeneous transformation matrices gives the ability to jump from one frame to another. The same idea could be used in expressing compound transformations. For example, the transformation between the base frame and the C'_i can be expressed as jumping from the base frame to frame A_i , then jump from frame A_i to frame C_i and finally a jump from frame C_i to frame C'_i . This compound transformation can be expressed as follows:

$${}^{base}T_{C'_i} = {}^{base}T_{A_i} \cdot {}^{A_i}T_{C_i} \cdot {}^{C_i}T_{C'_i} \quad (8.13)$$

The reason behind attaching the virtual frame C'_i is that, considering three spheres with radius b and center C'_i , the three sphere surfaces of the three legs intersect in the point P , which is the solution of the forward kinematics. Additionally, the other two angles of each leg, θ_{2i} and θ_{3i} , aren't known. The equation of the three spheres can be formulated as follows:

The solution of the set of equations leads to two points of intersection of the spheres, only one of them is geometrically meaningful.

Inverse Kinematics

The DELTA mechanism is especially known from impedance-controlled devices. In this mode of operation the inverse kinematics problem is not needed. However it is a very useful tool in the design process to determine the available workspace which is shown later in Sect. 8.4.

A frame is attached to the TCP at point P with the same orientation as the base frame. Three other frames are attached to each point B_i with the same orientation as the frames at A_i and C_i . To solve the inverse kinematics of each leg, we can use the following compound transformation:

$${}^{base}T_{TCP} = {}^{base}T_{A_i} \cdot {}^{A_i}T_{C_i} \cdot {}^{C_i}T_{B_i} \cdot {}^{B_i}T_{TCP} \quad (8.14)$$

where:

$$\begin{aligned}
 {}^{base}T_{TCP} &= \begin{bmatrix} 1 & 0 & 0 & x_P \\ 0 & 1 & 0 & y_P \\ 0 & 0 & 1 & z_P \\ 0 & 0 & 0 & 1 \end{bmatrix} {}^{C_i}T_{B_i} = \begin{bmatrix} 1 & 0 & 0 & b \cdot \sin \theta_{3i} \cdot \cos (\theta_{1i} + \theta_{2i}) \\ 0 & 1 & 0 & b \cdot \cos \theta_{3i} \\ 0 & 0 & 1 & b \cdot \sin \theta_{3i} \cdot \sin (\theta_{1i} + \theta_{2i}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^{B_i}T_{TCP} &= \begin{bmatrix} \cos (\phi_i) & \sin (\phi_i) & 0 & r_{TCP} \cdot \cos (\phi_i) \\ -\sin (\phi_i) & \cos (\phi_i) & 0 & -r_{TCP} \cdot \sin (\phi_i) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.15)
 \end{aligned}$$

The unknown matrices are ${}^{A_i}T_{C_i}$ and ${}^{C_i}T_{B_i}$, that are function of the three unknown angles, θ_{1i}, θ_{2i} , and θ_{3i} . In order to solve for the angles, all the known matrices should be on one side and the unknown should be on the other side. We get:

$${}^{base}T_{A_i}^{-1} \cdot {}^{base}T_{TCP} \cdot {}^{B_i}T_{TCP}^{-1} = {}^{A_i}T_{C_i} \cdot {}^{C_i}T_{B_i} \quad (8.16)$$

Multiplying the two matrices on the right hand side give:

$${}^{A_i}T_{B_i} = \begin{bmatrix} 1 & 0 & 0 & a \cdot \cos \theta_{1i} + b \cdot \sin \theta_{3i} \cdot \cos (\theta_{1i} + \theta_{2i}) \\ 0 & 1 & 0 & b \cdot \cos \theta_{3i} \\ 0 & 0 & 1 & a \cdot \sin \theta_{1i} + b \cdot \sin \theta_{3i} \cdot \sin (\theta_{1i} + \theta_{2i}) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_{B_i} \\ 0 & 1 & 0 & y_{B_i} \\ 0 & 0 & 1 & z_{B_i} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.17)$$

This leads to, according to [23]:

$$\theta_{3i} = \arccos \frac{y_{B_i}}{b} \quad (8.18)$$

$$\theta_{2i} = \arccos \frac{x_{B_i}^2 + y_{B_i}^2 + z_{B_i}^2 - a^2 - b^2}{2ab \sin \theta_{3i}} \quad (8.19)$$

$$\theta_{1i} = \arctan \frac{x_{B_i} - b \sin \theta_{3i} \cos (\theta_{1i} + \theta_{2i})}{z_{B_i} - b \sin \theta_{3i} \sin (\theta_{1i} + \theta_{2i})} \quad (8.20)$$

Equations (8.18)–(8.20) are the solution to the inverse kinematic equation for each leg.

8.4 Design Step 3: Dimensioning a Haptic Kinematic

The last step in the design is dimensioning. Optimizing the dimensions of the mechanism, lengths of the rods/links defined in step 1 (Sect. 8.2), affects the workspace of the mechanism, the transmission of forces/torques, and the velocities. The goal of

the optimization is to reach a specific optimum performance. This may be a maximized workspace with homogeneous transfer characteristics of forces from the TCP to the actuators. Dimensioning procedure in parallel mechanisms is usually more complicated compared to that in serial mechanisms. According to MERLET, parallel mechanisms with well designed dimensions can perform better than one with better suited topology but worse dimensions [15]. An important parameter in haptic interfaces is the impedance. In order to calculate the impedance of the system, the values of the velocities and the forces have to be known. In this part of the chapter, an introduction will be given on how the dimensioning procedure is performed.

Jacobian Matrix

In both the forward and inverse kinematic problems, the vectors q and p are linked via the mechanism's gearing properties. Those properties are represented by the Jacobian matrix J . For the mechanism's kinematics, the Jacobian matrix represents the transmission matrix of the first order. It carries all information regarding dimensions and transmission properties. J is defined by the partial derivative of TCP coordinates with respect to the joints' coordinates. However, generally, the Jacobian could be calculated for any frame in the mechanism, but usually the TCP frame is the important frame to be considered. The size of the matrix is $m \times n$, where m is the number of TCP coordinates and n is the number of joints' coordinates. For example, the TCP of UR10E robot has $m = 6$ ($x_p, y_p, z_p, \alpha_p, \beta_p, \gamma_p$) and $n = 6$ ($\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$). So the Jacobian matrix of this mechanism consists of m rows and n columns:

$$J = \begin{bmatrix} \frac{\partial x_p}{\partial \theta_1} & \frac{\partial x_p}{\partial \theta_2} & \frac{\partial x_p}{\partial \theta_3} & \frac{\partial x_p}{\partial \theta_4} & \frac{\partial x_p}{\partial \theta_5} & \frac{\partial x_p}{\partial \theta_6} \\ \frac{\partial y_p}{\partial \theta_1} & \frac{\partial y_p}{\partial \theta_2} & \frac{\partial y_p}{\partial \theta_3} & \frac{\partial y_p}{\partial \theta_4} & \frac{\partial y_p}{\partial \theta_5} & \frac{\partial y_p}{\partial \theta_6} \\ \frac{\partial z_p}{\partial \theta_1} & \frac{\partial z_p}{\partial \theta_2} & \frac{\partial z_p}{\partial \theta_3} & \frac{\partial z_p}{\partial \theta_4} & \frac{\partial z_p}{\partial \theta_5} & \frac{\partial z_p}{\partial \theta_6} \\ \frac{\partial \alpha_p}{\partial \theta_1} & \frac{\partial \alpha_p}{\partial \theta_2} & \frac{\partial \alpha_p}{\partial \theta_3} & \frac{\partial \alpha_p}{\partial \theta_4} & \frac{\partial \alpha_p}{\partial \theta_5} & \frac{\partial \alpha_p}{\partial \theta_6} \\ \frac{\partial \beta_p}{\partial \theta_1} & \frac{\partial \beta_p}{\partial \theta_2} & \frac{\partial \beta_p}{\partial \theta_3} & \frac{\partial \beta_p}{\partial \theta_4} & \frac{\partial \beta_p}{\partial \theta_5} & \frac{\partial \beta_p}{\partial \theta_6} \\ \frac{\partial \gamma_p}{\partial \theta_1} & \frac{\partial \gamma_p}{\partial \theta_2} & \frac{\partial \gamma_p}{\partial \theta_3} & \frac{\partial \gamma_p}{\partial \theta_4} & \frac{\partial \gamma_p}{\partial \theta_5} & \frac{\partial \gamma_p}{\partial \theta_6} \end{bmatrix} \quad (8.21)$$

More details are found in [12].

The Jacobian matrix is used to express various relations between the inputs and outputs of a mechanism, such as the relation between the velocities of the joints compared to the velocity of the TCP, and the relation between the torques applied to the joints and the forces on the TCP. These relations are discussed later in this section.

8.4.1 *Workspace*

The dimensions of the mechanism affect the workspace. To perform an optimization the following steps should be taken:

1. Definition of parameters and their span of values (e.g. rod lengths)
2. Analytical or numerical description of the optimization problem
3. Mathematical optimization, e.g. via a gradient approach or evolutionary algorithms

These steps are discussed and covered in [8, 17]. The key challenge is the formal description of the optimum. This process should be done using a computer software which will be discussed later in Sect. 8.5. Within the optimization process, the measurement value for an optimum has to be determined by scanning the complete workspace and optimizing relevant parameters of the mechanism between each scanning process. In [1] several optimizations are given using the singular values of the Jacobian matrix as a measure.

8.4.2 *Isotropy and Singular Positions*

The dimensioning process continues by taking into account the best desirable working points of the TCP inside the workspace and what positions should be avoided.

Isotropy describes the optimum working points in the workspace, these are the configurations where the servo transmissions are highly coupled, meaning that the error between the input and the output is minimised.

On the other hand, singularities are the configurations that should be avoided. In singular positions, the control of one or more of the mechanism's DoF are lost. If a mechanism approaches a singular position its transmission or gear ratio changes quickly until the mechanism is locked in the singular position. Singularities are divided into two main types [5].

Workspace-Boundary Singularities

This type of singularity occurs when the mechanism is fully stretched to the edge of the workspace. This applies to all types of mechanisms.

Workspace-Interior Singularities

This type occurs inside the workspace. In serial mechanisms, one of these singularities happens especially in six DoF mechanisms where the axes of the last three joints (wrist) intersect in on point. Usually this happens when two axes are coincident.

Figure 8.15 shows examples of both types of singular positions. The key to analyze the isotropy and singularity is based on the properties of the Jacobian matrix. A key performance index which is derived from the Jacobian matrix is the condition number κ .

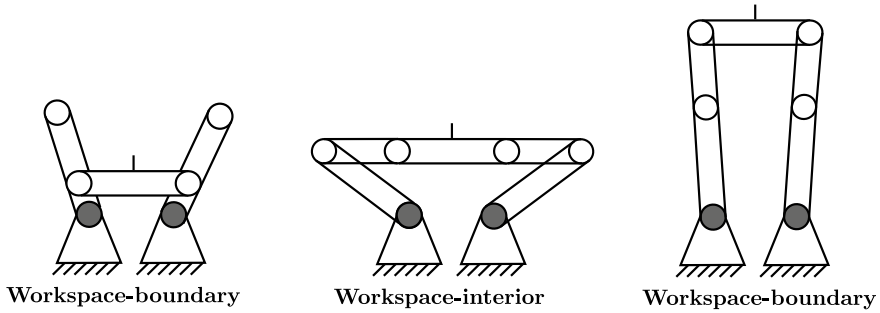


Fig. 8.15 Different singular positions

The Conditioning Number

The kinematic transmission behavior is rated by the singular values σ_i of the inverse Jacobian matrix J^{-1} . In general the singular values of a matrix A are defined as:

$$\sigma_i(A) = \sqrt{\lambda_i(A^T A)} \tag{8.22}$$

The role of the singular values can be shown by GOLUB’S method of singular value decomposition [6]. It is based on the fact that for a real $m \times n$ matrix A , with $m \geq n$, and rank r can be fractioned in the following product:

$$A = U \cdot \Sigma \cdot V^T \tag{8.23}$$

Where U consists of n orthonormalized eigenvectors of the n largest eigenvalues of AA^T and V consists of the orthonormalized eigenvectors of $A^T A$. Σ is a $m \times n$ diagonal matrix as follows:

$$\Sigma = \left(\begin{array}{ccc|ccc} \sigma_1 & & & & \vdots & \\ & \ddots & & & & \\ & & & & \vdots & \\ & & \sigma_r & & \vdots & \\ \hline & & & & & \\ & \vdots & & & & \\ \dots & 0 & \dots & \dots & 0 & \dots \\ & \vdots & & & \vdots & \end{array} \right) \tag{8.24}$$

Where $\sigma_1 \geq \dots \geq \sigma_r > 0$. The conditioning number is defined as:

$$\kappa = \frac{\sigma_{max}}{\sigma_{min}} \tag{8.25}$$

As a function of the Jacobian matrix κ changes with respect to the mechanism's position. The conditioning number can reach values from $\frac{1}{\kappa} = 0 \dots 1$.

The goal is to have a highly isotropic transmission, which means a conditioning number of 1. On the other hand, singular positions should be avoided. In terms of the Jacobian matrix, the rank of the matrix decreases in the singular position. This will translated into a conditioning number of ∞ or $\frac{1}{\kappa} = 0$.

For the two types of singularities introduced earlier, the loss of rank of the Jacobian matrix is characterized by:

- Workspace-boundary singularity: $\det(J) = 0$
- Workspace-interior singularity: $\det(J^{-1}) = 0$.

Optimization Criteria

Besides the analysis of isotropy and singular positions, another aspect one has to take care of in the design process is the transmission of force and speed.

Recalling Eq. (8.35), to limit the maximal required force and torque and thereby limit also the size of the used actuators, it is important to reach a good transmission of forces and torques even in cases of a disadvantageous σ_i . We can derive the criterion as follows:

$$\sigma_{min}(J^{-1}) \rightarrow max \tag{8.26}$$

For maximizing the speed transmission, the criterion could be as follows:

$$\sigma_{max}(J^{-1}) \rightarrow min \tag{8.27}$$

Table 8.5 sums up various design optimization criteria.

One major drawback of Eq. (8.25) is that it rates the mechanism for Jacobian matrix or position. The pure optimization of $1/\kappa$ would in fact lead to one single position where the mechanism reaches high isotropy. However one cannot draw the conclusion that the whole workspace in total has an optimized transmission behavior.

Table 8.5 Summary of optimization criteria

Design aspect	Criterion
Force transmission	$\sigma_{min}(\mathbf{J}) \rightarrow \max$
No singular positions	$\sigma_{min}(\mathbf{J}) \rightarrow \max$
High stiffness	$\sigma_{min}(\mathbf{J}) \rightarrow \max$
Speed transmission	$\sigma_{max}(\mathbf{J}) \rightarrow \min$
Isotropy	$\frac{\sigma_{min}(\mathbf{J})}{\sigma_{max}(\mathbf{J})} \rightarrow \max$

What is needed is a measure to rate $1/\kappa$ of a whole workspace. This measure is provided by the global conditioning index as in [15].

$$v = \frac{\int_W \frac{1}{\kappa} dW}{\int_W dW} \quad (8.28)$$

The global conditioning index can be optimized using computer algorithms.

8.4.3 Velocities

The velocities of the joints and those of the TCP are related with the Jacobian matrix of the mechanism:

$$dp = J \cdot dq \quad (8.29)$$

Equation (8.29) gives the output velocity of the TCP with respect to the joints' velocities. The same idea could be done to get the desired joints' velocities in order to have a required velocity of the TCP:

$$dq = J^{-1} \cdot dp \quad (8.30)$$

The optimization process should involve the desired velocities of the TCP, this will affect the motors used to drive the joints.

8.4.4 Dynamics

For the design and operation of haptic interfaces there is another equation of high importance related to the transformation of forces and torques by a mechanism. In order to express the dynamics of a mechanism, the equations of motion of the links are to be calculated. The goal is to find the required torques/forces on the joints. CRAIG [5] divided the approaches to calculate the equations of motions into: Iterative (numerical) and closed form (analytical).

Iterative Approach: NEWTON-EULER Dynamics Algorithm

One example of iterative methods is the NEWTON- EULER dynamics algorithm. This algorithm is split into parts, outward and inward iterations. The algorithm is as follows:

1. Outward iterations are computed to calculate the velocities and accelerations (linear and rotational) of the center of mass of each link/rod in the mechanism. The iterations start with the first link and ends with the last link.

2. Using the calculated velocities and accelerations of each link, the forces and the torques acting on the link are calculated using the NEWTON- EULER equations. This step occurs along with the outward iterations.
3. Finally the torques/forces on each joint are calculated using the inward iterations. These iterations start with the final link and move back to the first link.

The usage of a numerical approach can be applied to any robot. It only needs the inertia tensors of each link, position vectors that connect the links with each other and the rotation matrices between each two links. On the other hand, sometimes the information about the gravity and the non-inertial effects are important. A closed form equation should be introduced in that case.

Closed Form Approach

Closed form approaches express the dynamics of a mechanism in more detail. There are a lot of methods that can be used to express the equations of motion analytically. Two of these methods are discussed in this chapter: NEWTON- EULER equation and Lagrangian dynamic formulation.

The general form of a NEWTON- EULER equation for a link is as follows:

$$\tau = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) \quad (8.31)$$

where:

- τ Vector of torques applied on the joints
- M Mass matrix of the mechanism
- V Vector includes the Coriolis and centrifugal terms
- G Vector includes the gravitational terms

Another method that is widely used is the Lagrangian dynamic formulation. The NEWTON- EULER equation is considered a force balance approach, while, on the other hand, Lagrangian formulation is considered as an energy approach. This method uses the energy of the system to express the equations of motions.

A scalar function called the Lagrangian (L) is defined as:

$$L(\theta, \dot{\theta}) = k(\theta, \dot{\theta}) - u(\theta) \quad (8.32)$$

where:

- L Lagrangian function
- k Sum of the kinetic energy of the mechanism
- u Sum of the potential energy of the mechanism

The equations of motion are given as follow:

$$\tau = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} \quad (8.33)$$

The number of equations of motion obtained using the Lagrangian dynamics formulation depends on the number of the generalized coordinates. The generalized coordinates are the parameters needed to express the configuration of a mechanism. In our case the joints' values are the generalized coordinates. This means that the Lagrangian function L should be expressed only in terms of the generalized coordinates.

The equations showed (8.31) and (8.33) include only the forces as a result of rigid body mechanics, the most important factor that isn't included is friction. There are multiple ways to model friction forces; the two most important models are viscous friction and Coulomb friction. An additional part F_f is added to Eq. (8.31) to model the friction:

$$\tau = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) + F_f(\theta, \dot{\theta}) \quad (8.34)$$

Equations (8.31) and (8.33) give the same output. Also, both of them are expressed in terms of the joints' positions, velocities, accelerations or, in other words, in the joint space. In order to express the forces on the TCP, the Jacobian matrix can be used as follows:

$$\tau = J^T(\theta) \cdot F \quad (8.35)$$

Combining Eqs. (8.31) and (8.35) results to:

$$J^{-T}\tau = J^{-T}M(\theta)\ddot{\theta} + J^{-T}V(\theta, \dot{\theta}) + J^{-T}G(\theta) \quad (8.36)$$

Which results to:

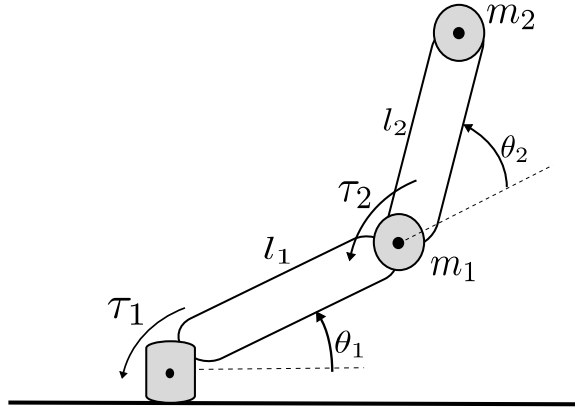
$$F = J^{-T}M(\theta)\ddot{\theta} + J^{-T}V(\theta, \dot{\theta}) + J^{-T}G(\theta) \quad (8.37)$$

Other methods are available to express the equations of motion of mechanisms. Malvezzi et al. [13] made a qualitative comparison between three approaches to express the dynamics of a serial mechanism. The dynamics in serial mechanisms gets complicated with the increase in number of links, however, it's not as complicated as the case of parallel mechanisms.

8.4.4.1 Example: Equations of Motion of 2-DoF Serial Mechanism

To show how obtaining the equations of motion is complicated, a simple mechanism will be discussed. Figure 8.16 shows a 2-DoF serial mechanism. The mechanism is simplified such as the masses m_1 and m_2 are considered to be point masses at the end of each link and the links are considered massless. Also the friction forces aren't taken into consideration. The equations of motion are obtained using the Lagrange dynamic formulation. The generalized coordinates of this mechanism are θ_1 and θ_2 . Recalling Eq. (8.33), the two equations of motion are as follows:

Fig. 8.16 2-DoF serial mechanism



$$\tau_1 = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} \tag{8.38}$$

$$\tau_2 = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2} \tag{8.39}$$

Where each equation represents the torque on each motor that creates the motion of each angle.

The total kinetic energy of the mechanism is:

$$k_{Total} = k_{m_1} + k_{m_2} = \frac{1}{2} \cdot m_1 \cdot (v_{m_1})^2 + \frac{1}{2} \cdot m_2 \cdot (v_{m_2})^2 \tag{8.40}$$

$$v_{m_1}^2 = (l_1 \cdot \dot{\theta}_1)^2$$

To get v_{m_2} , one can obtain first the position of m_2 in Cartesian space and convert it to the joint space:

$$x_{m_2} = l_1 \cdot \cos \theta_1 + l_2 \cdot \cos (\theta_1 + \theta_2)$$

$$y_{m_2} = l_1 \cdot \sin \theta_1 + l_2 \cdot \sin (\theta_1 + \theta_2)$$

$$v_{m_2}^2 = (\dot{x}_{m_2})^2 + (\dot{y}_{m_2})^2$$

This leads to:

$$v_{m_2}^2 = l_1^2 \dot{\theta}_1^2 + (\dot{\theta}_1 + \dot{\theta}_2)^2 l_2^2 + 2\dot{\theta}_1 l_1 l_2 (\dot{\theta}_1 + \dot{\theta}_2) [\sin \theta_1 \sin (\theta_1 + \theta_2) + \cos \theta_1 \cos (\theta_1 + \theta_2)]$$

Using the angle addition trigonometric function, v_{m_2} is given as follows:

$$v_{m_2}^2 = l_1^2 \dot{\theta}_1^2 + (\dot{\theta}_1 + \dot{\theta}_2)^2 l_2^2 + 2\dot{\theta}_1 l_1 l_2 (\dot{\theta}_1 + \dot{\theta}_2) [\cos \theta_2]$$

The total kinetic energy of the mechanism is:

$$k_{Total} = \frac{1}{2} \cdot m_1 \cdot (l_1 \cdot \dot{\theta}_1)^2 + \frac{1}{2} \cdot m_2 \cdot l_1^2 \dot{\theta}_1^2 + (\dot{\theta}_1 + \dot{\theta}_2)^2 l_2^2 + 2\dot{\theta}_1 l_1 l_2 (\dot{\theta}_1 + \dot{\theta}_2) [\cos \theta_2]$$

The potential energy of the mechanism is:

$$u_{Total} = m_1 \cdot g \cdot y_{m_1} + m_2 \cdot g \cdot y_{m_2} \quad (8.41)$$

$$u_{Total} = m_1 \cdot g \cdot l_1 \sin \theta_1 + m_2 \cdot g \cdot [l_1 \sin \theta_1 + l_2 \sin (\theta_1 + \theta_2)]$$

Hence, the Lagrangian function L is defined as:

$$L = k_{Total} - u_{Total} \quad (8.42)$$

The torques to be applied on the joints, recalling Eqs. (8.38) and (8.39), are as follow:

$$\begin{aligned} \tau_1 = & (m_1 + m_2) l_1^2 \ddot{\theta}_1 + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 \cos \theta_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) - m_2 l_1 l_2 \sin \theta_2 \dot{\theta}_2^2 \\ & - 2m_2 l_1 l_2 \sin \theta_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g \cos (\theta_1 + \theta_2) + (m_1 + m_2) l_1 g \cos \theta_1 \end{aligned} \quad (8.43)$$

$$\tau_2 = m_2 l_2 [(\ddot{\theta}_1 + \ddot{\theta}_2) l_2 + l_1 \cos \theta_2 \ddot{\theta}_1 + l_1 \sin \theta_2 \dot{\theta}_1^2 + g \cos (\theta_1 + \theta_2)] \quad (8.44)$$

Or in matrix form like Eq. (8.34):

$$\begin{aligned} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = & \begin{bmatrix} (m_1 + m_2) l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos \theta_2 & m_2 l_2^2 + m_2 l_1 l_2 \cos \theta_2 \\ m_2 l_2^2 + m_2 l_1 l_2 \cos \theta_2 & m_2 l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \\ & \begin{bmatrix} -m_2 l_1 l_2 \sin \theta_2 \dot{\theta}_2^2 - 2m_2 l_1 l_2 \sin \theta_2 \dot{\theta}_1 \dot{\theta}_2 \\ m_2 l_1 l_2 \sin \theta_2 \dot{\theta}_1^2 \end{bmatrix} + \begin{bmatrix} m_2 l_2 g \cos (\theta_1 + \theta_2) + (m_1 + m_2) l_1 g \cos \theta_1 \\ m_2 l_2 g \cos (\theta_1 + \theta_2) \end{bmatrix} \end{aligned} \quad (8.45)$$

As mentioned earlier, the torque equations are expressed in joint space. In order to express the forces acting on the TCP, the Jacobian matrix is to be used. The Jacobian matrix of this mechanism is:

$$J = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} -l_1 \sin \theta_1 - l_2 \sin (\theta_1 + \theta_2) & -l_2 \sin (\theta_1 + \theta_2) \\ l_1 \cos \theta_1 + l_2 \cos (\theta_1 + \theta_2) & l_2 \cos (\theta_1 + \theta_2) \end{bmatrix} \quad (8.46)$$

recalling Eq. (8.36), J^{-T} is expressed as:

$$J^{-T} = \frac{1}{l_1 l_2 \sin \theta_2} \begin{bmatrix} l_2 \cos(\theta_1 + \theta_2) & -l_1 \cos \theta_1 - l_2 \cos(\theta_1 + \theta_2) \\ l_2 \sin(\theta_1 + \theta_2) & -l_1 \sin \theta_1 - l_2 \sin(\theta_1 + \theta_2) \end{bmatrix} \quad (8.47)$$

Recalling Eq. (8.37):

$$J^{-T} M(\theta) = \frac{1}{l_1 l_2 \sin \theta_2} \begin{bmatrix} M'_{11} & M'_{12} \\ M'_{21} & M'_{22} \end{bmatrix} \quad (8.48)$$

Where:

$$M'_{11} = (m_1 + m_2)l_1^2 l_2 \cos(\theta_1 + \theta_2) + m_2 l_1 l_2^2 \cos \theta_2 \cos(\theta_1 + \theta_2) - m_2 l_1 l_2^2 \cos \theta_1 - m_2 l_1^2 l_2 \cos \theta_1 \cos \theta_2 \quad (8.49)$$

$$M'_{12} = m_2 l_1 l_2^2 (\cos \theta_2 \cos(\theta_1 + \theta_2) - \cos \theta_1) \quad (8.50)$$

$$M'_{21} = (m_1 + m_2)l_1^2 l_2 \sin(\theta_1 + \theta_2) + m_2 l_1 l_2^2 \cos \theta_2 \sin(\theta_1 + \theta_2) - m_2 l_1 l_2^2 \sin \theta_1 - m_2 l_1^2 l_2 \sin \theta_1 \cos \theta_2 \quad (8.51)$$

$$M'_{22} = m_2 l_1 l_2^2 (\cos \theta_2 \sin(\theta_1 + \theta_2) - \sin \theta_1) \quad (8.52)$$

Accordingly:

$$J^{-T} V(\theta, \dot{\theta}_1) = \frac{1}{l_1 l_2 \sin \theta_2} \begin{bmatrix} V'_1 \\ V'_2 \end{bmatrix} \quad (8.53)$$

Where:

$$V'_1 = -m_2 l_1 l_2^2 \sin \theta_2 \cos(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)^2 - m_2 l_1^2 l_2 \cos \theta_1 \sin \theta_2 \dot{\theta}_1^2 \quad (8.54)$$

$$V'_2 = -m_2 l_1 l_2^2 \sin \theta_2 \sin(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)^2 - m_2 l_1^2 l_2 \sin \theta_1 \sin \theta_2 \dot{\theta}_1^2 \quad (8.55)$$

And:

$$J^{-T} G(\theta) = \frac{1}{l_1 l_2 \sin \theta_2} \begin{bmatrix} G'_1 \\ G'_2 \end{bmatrix} \quad (8.56)$$

Where:

$$G'_1 = m_1 l_1 l_2 g \cos \theta_1 \cos(\theta_1 + \theta_2) \quad (8.57)$$

$$G'_2 = (m_1 + m_2) l_1 l_2 g \cos \theta_1 \sin(\theta_1 + \theta_2) - m_2 l_1 l_2 g \sin \theta_1 \cos(\theta_1 + \theta_2) \quad (8.58)$$

If we assume from Eq. (8.37), that $F = [F_x \ F_y]^T$, this leads to the force acting on the TCP in x- and y-direction:

$$F_x = \frac{1}{l_1 l_2 \sin \theta_2} [M'_{11} \ddot{\theta}_1 + M'_{12} \ddot{\theta}_2 + V'_1 + G'_1] \quad (8.59)$$

$$F_y = \frac{1}{l_1 l_2 \sin \theta_2} [M'_{21} \ddot{\theta}_1 + M'_{22} \ddot{\theta}_2 + V'_2 + G'_2] \quad (8.60)$$

Equations (8.59) and (8.60) express the forces with respect to the angular accelerations of the joints $\ddot{\theta}$. The same forces could be expressed with respect to the accelerations of the Cartesian variables \ddot{X} . The general form is found in [5]:

$$F = M_x(\theta) \ddot{X} + V_x(\theta, \dot{\theta}) + G_x(\theta) \quad (8.61)$$

Where:

$$M_x = J^{-T}(\theta) M(\theta) J^{-1}(\theta) \quad (8.62)$$

$$V_x = J^{-T}(\theta) (V(\theta, \dot{\theta}) - M(\theta) J^{-1}(\theta) \dot{J}(\theta) \dot{\theta}) \quad (8.63)$$

$$G_x = J^{-T}(\theta) G(\theta) \quad (8.64)$$

From this example:

- Although the mechanism is just a simple 2-DoF planar one, the equations of motion are already complicated. The equations get more complicated as the number of DoF increases. This issue is somehow solved by obtaining the equations of motion by a computer-based algorithm. This is the reason, why the role of simulation is so important (Sect. 8.5).
- Looking at the equation of τ_1 , it can be seen that the required torque on the first joint contains many terms depending on both angles (θ_1 and θ_2). This explains the statement related to serial mechanisms in particular that each joint is responsible for the dynamics of the following links.

8.5 Role of Simulation

The design steps of kinematic mechanisms are introduced in the previous sections. The example introduced in Sect. 8.4.4.1 shows how complex the equations of motion are. This complexity, not only in the equations of motion, but rather in the whole dimensioning process, is the reason behind using computer-based simulations.

Figure 8.17 shows a block diagram of the usage of kinematic mechanisms in a general application. Taking a pick-and-place application, the goal of the manipulator is to follow a certain trajectory in Cartesian space. The desired trajectory is the input in our case.

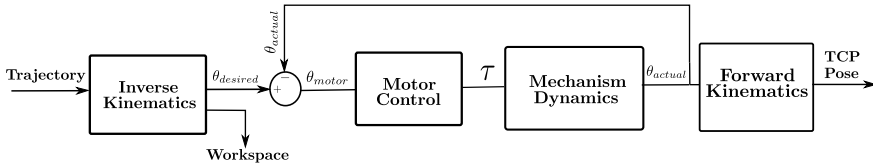


Fig. 8.17 Block diagram of the usage of a kinematic mechanism in a general application

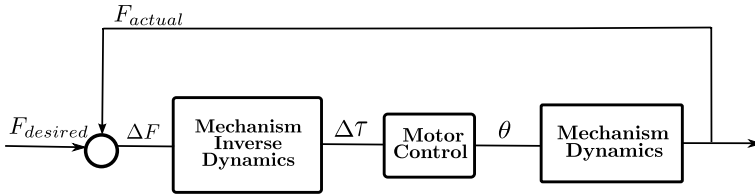


Fig. 8.18 Block diagram of the usage of a kinematic mechanism in a haptic application

Next comes the role of the inverse kinematics to transfer the desired trajectory into desired joint angles. Generally, inverse kinematics has another importance, that is the definition of the manipulator workspace. Any point inside the workspace has a solution for the inverse kinematics equations of the mechanism.

The desired angles are subtracted from the actual joint angles and the difference is fed to the control block. For the scope of this chapter, the control block isn't discussed. What is important for this chapter, is that the output of this block is the torques applied to the motors.

The torques are the inputs to the manipulator dynamics block, that contains the equations of motion of the manipulator. The output of this block are the actual angles of the joints. The actual joints angles complete the feedback loop to the desired angles summation point.

The actual angles can be also used to get the pose of the TCP using the forward kinematics of the manipulator.

For haptic applications, the general diagram in Fig. 8.17 doesn't perfectly match. In haptic applications, sometimes, the goal isn't to follow a specific trajectory, but rather to have a desired force on the TCP, as shown in Fig. 8.18.

The desired forces are subtracted from the actual forces sensed on the TCP. The difference is fed to the control block. The output of the control block are the torques applied on the motors.

The torques are fed to the manipulator dynamics block. This block doesn't only contain the mechanism dynamics, but also the dynamics of the user are modeled. The simplest model of the user is a mass-spring-damper system. The outputs of the this block are the actual forces on the TCP.

8.5.1 Example of Software Used in Simulation

There are many software on the market that model and simulate the kinematics of haptic interfaces. For haptic interfaces, the inverse kinematics is of high importance and it limits the software to be used. Generally, all the software have many features in common:

- Model the kinematic mechanism, apart from the programming language used or the way of modeling, whether using programming scripts or blocks.
- Model the environment of the application with all the external inputs.
- Visualize and simulate the movement of the mechanism inside the environment during performing a predefined application.

Comparing between different software isn't the scope of this chapter. We will focus in this section on Matlab[®] and its offered toolboxes to give only an example on how the modeling and simulation are implemented. The example shown in Sect. 8.4.4.1 will be discussed in terms of optimizing the workspace and how the kinematic and dynamic equations are solved.

8.5.2 Optimizing the Workspace

As mentioned earlier, any point inside the workspace has a solution for the inverse kinematics of the mechanism. Referring to the example in Sect. 8.4.4.1, the workspace depends on the lengths of the two links and the limits of the two revolute joints. Figure 8.19 shows the workspace of the mechanism with two different combinations of link lengths. In both combinations the limits of the joints are as follow:

$$0^\circ \leq \theta_1 \leq 90^\circ$$

$$0^\circ \leq \theta_2 \leq 90^\circ$$

The lengths in the first combination are ($l_1 = l_2 = 10$ cm), where in the second combination the lengths are ($l_1 = l_2 = 20$ cm). For more complex mechanisms, more variables will be included in the optimization process. One has to keep in mind that by changing the lengths, the dynamics of the mechanism change as well. This will be discussed in the following sections.

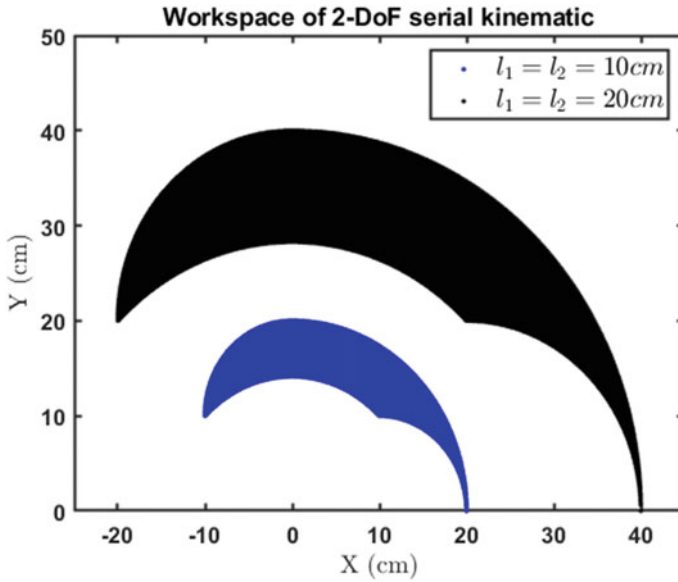


Fig. 8.19 Workspace of 2-DoF mechanism with different combinations of link lengths

8.5.3 Solving Kinematic and Dynamic Equations

In Matlab[®] there are many ways to model the kinematic equations. One way is to model the mechanism as a rigid-body tree. This is done by defining all the links/legs, and the joints. The rigid-body tree approach supports serial mechanisms, however, parallel mechanisms aren't directly supported.

In haptic applications, another approach is more applicable. As mentioned earlier in Fig. 8.18, the goal is to maintain certain forces on the TCP. This can be modeled using Simulink[®] and Simscape Multibody[™] toolbox.

Generally, forward and inverse kinematics equations, Jacobian matrix and equations of motion could be modeled using Matlab or Simulink, such that the optimization criteria listed in Table 8.5 could be implemented.

In Simscape Multibody[™] toolbox, the designer has the option to either import the mechanism from a CAD software, or to use the predefined model blocks in the toolbox. The toolbox offers variety of joints and sensors. Figure 8.20 shows a simple modeling of our 2-DoF mechanism. The two links (l_1 and l_2) and the two masses (m_1 and m_2) are modeled using the predefined blocks, frames are attached to the each end of the links and the masses. The revolute joints connect the frames and constraint the motion in the specified direction only, rotation about the z-axis. The solver of the toolbox solves the kinematic equations defined by connection of the

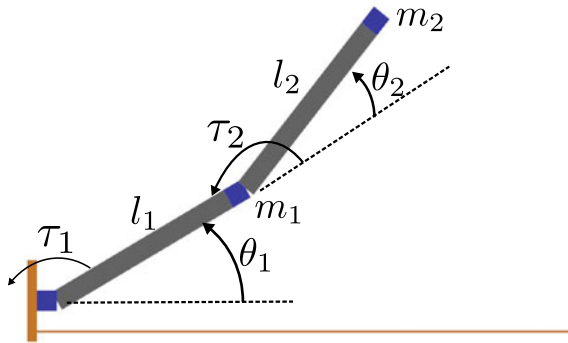


Fig. 8.20 Simscape Multibody™ model of 2-DoF serial mechanism

blocks, and singular positions are also detected. Modeling the mechanism using the toolbox solves the equations of motion. The torques applied on the joints and the joints' values are sensed using predefined sensors in the joint block. Applying Eq. (8.35), the forces on the TCP are calculated.

As mentioned earlier, changing the lengths of the links affects the dynamics of the mechanism. Consider the two link length combinations used in Sect. 8.5.2 and the masses (m_1 and m_2) are set to be 0.1 kg.

A simplified haptic scenario is when the user is obliged to follow a certain trajectory. An example of such trajectory could be that θ_1 is fixed, and the input to θ_2 is in the form of a sine wave. To simulate the torques applied to the joints and the result forces on the TCP for both combinations, the trajectory of the simplified scenario is applied to the joints. Fig. 8.21 shows the required torques on the joints for both combinations of lengths. From Fig. 8.21, it can be seen that the τ_1 is higher than τ_2 in both length combinations. The reason behind it, as mentioned earlier, in serial mechanisms one actuator carries the load of all the following actuators. Subsequently, the forces on the TCP can be calculated using Eq. (8.35). The forces are represented in Fig. 8.22.

The Simulink® model of this simplified scenario is shown in Fig. 8.23. The model consists of two main subsystems; the first contains the model of the mechanism: links, masses, joints, and the environment. The second subsystem contains the Jacobian matrix of the mechanism in order to calculate the forces on the TCP. Generally, more subsystems are added that contain, for example, the controller. From the values represented in Figs. 8.21 and 8.22, one can decide on the driver, the motors in this example, that will be able to apply the required torques to the joints.

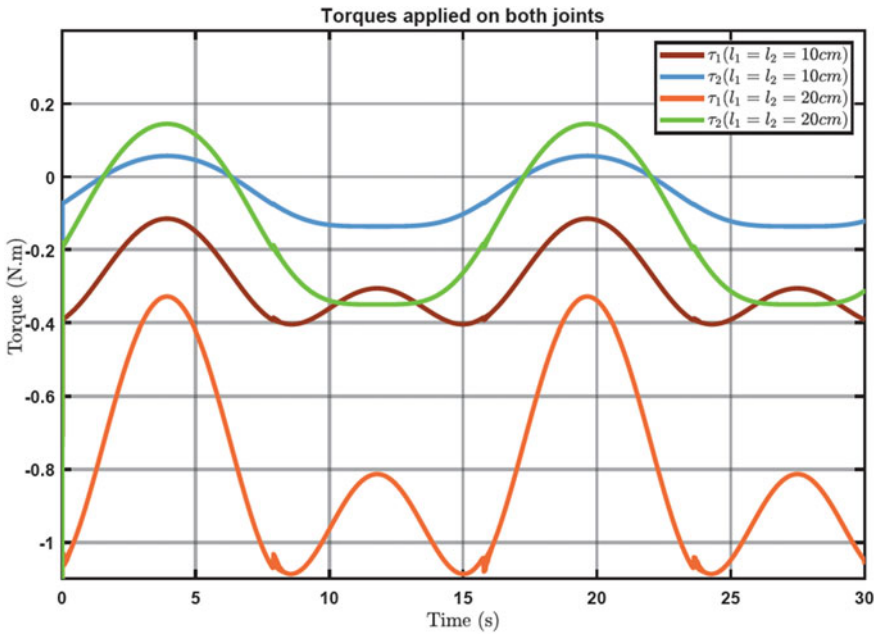


Fig. 8.21 Torques on both joints

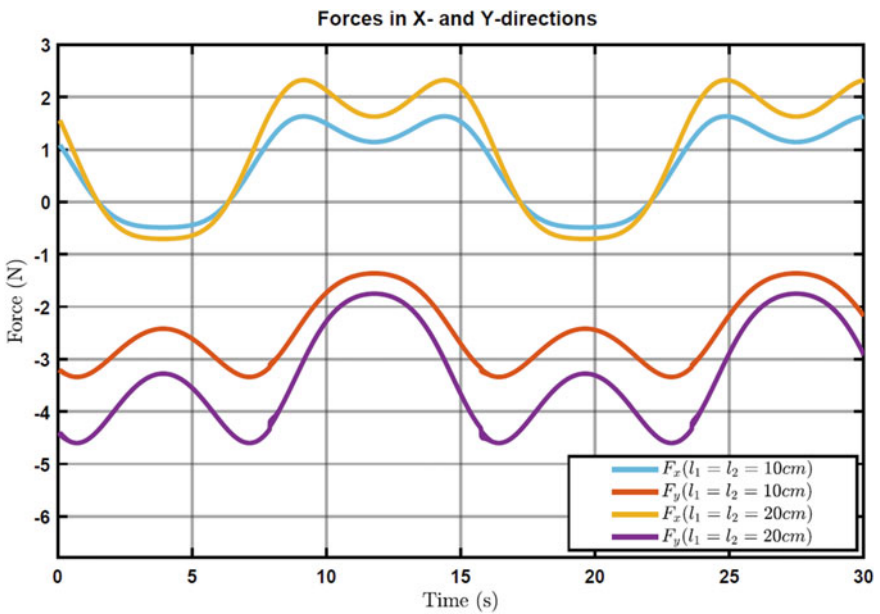


Fig. 8.22 Forces on the TCP

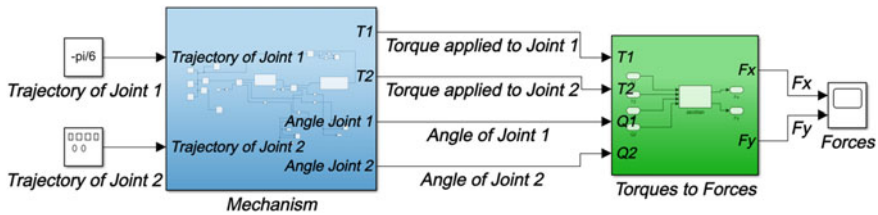


Fig. 8.23 Simulink[®] model of the mechanism in a simplified haptic scenario

References

- Bleicher F (2003) Parallelkinematische Werkzeugmaschinen. Verlag, NWV-Neuer Wiss. 3708301188, 9783708301181
- Carricato M, Parenti-Castelli V (2002) Singularity-free fully-isotropic translational parallel mechanisms. *Int J Robot Res* 21(2):161–174. <https://doi.org/10.1177/027836402760475360>
- Carricato M, Parenti-Castelli V (2003) A family of 3-DOF translational parallel manipulators. *J Mech Des* 125 (2):302–307. <https://doi.org/10.1115/1.1563635>
- Clavel, R. (1988). DELTA, a fast robot with parallel geometry. In C. W. Burckhardt (ed.), *Proc of the 18th International Symposium on Industrial Robots* (p./pp. 91–100), New York: Springer-Verlag
- Craig JJ (2005) *Introduction to robotics: mechanics and control*. Pearson Prentice Hall. ISBN: 9789702607724
- Golub GH, Reinsch C (1971) Singular value decomposition and least squares solutions. Springer, pp 134–151. https://doi.org/10.1007/978-3-662-39778-7_10
- Husty M et al (2013) *Kinematik und robotik*. Springer. ISBN: 3642590292, 9783642590290
- Kirchner J (2001) Mehrkriterielle optimierung von parallelkinematiken. Verlag Wiss, Scripten
- Lee T-Y, Shim J-K (2003) Improved dialytic elimination algorithm for the forward kinematics of the general Stewart-Gough platform. In: *Mechanism and machine theory*, vol 38, no 6, pp 563–577. [https://doi.org/10.1016/S0094-114X\(03\)00009-0](https://doi.org/10.1016/S0094-114X(03)00009-0)
- Lipkin, H. "A Note on Denavit-Hartenberg Notation in Robotics." *Proceedings of the ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol 7: 29th Mechanisms and Robotics Conference, Parts A and B. Long Beach, California, USA. September 24–28, 2005. pp. 921–926. ASME. <https://doi.org/10.1115/DETC2005-85460>
- Q. Liu, D. Yang, W. Hao and Y. Wei, *Research on Kinematic Modeling and Analysis Methods of UR Robot*, (2018) *IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2018, pp. 159–164, <https://doi.org/10.1109/ITOEC.2018.8740681>
- López M et al (2006) Delta robot: inverse, direct, and intermediate Jacobians. In: *Proceedings of the institution of mechanical engineers, Part C: Journal of mechanical engineering science*, vol 220, no 1 (2006), pp 103–109. <https://doi.org/10.1243/095440606X78263>
- Malvezzi, F., Orsino, R.M.M., Coelho, T.A.H. (2019). Lagrange's, Maggi's and Kane's Equations Applied to the Dynamic Modelling of Serial Manipulator. In: Fleury, A., Rade, D., Kurka, P. (eds) *Proceedings of DINAME 2017*. DINAME 2017. *Lecture Notes in Mechanical Engineering*(.). Springer, Cham. https://doi.org/10.1007/978-3-319-91217-2_20
- Merlet J-P (2004) Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis. *Int J Robot Res* 23(3):221–235. <https://doi.org/10.1177/0278364904039806>
- Merlet J-P (2005) *Parallel robots*, vol 128. Springer Science & Business Media. ISBN: 9781402041327
- Mourrain B (1993) The 40 "generic" positions of a parallel robot. In: *Proceedings of the 1993 international symposium on Symbolic and algebraic computation*, pp 173–182

17. Neugebauer R (2006) *Parallelkinematische Maschinen: Entwurf, Konstruktion. Anwendung.* Springer, Berlin, Heidelberg, New York. ISBN 103-540-20991-3
18. Parikh PJ, Lam SS (2009) Solving the forward kinematics problem in parallel manipulators using an iterative artificial neural network strategy. *Int J Adv Manuf Technol* 40(5):595–606. <https://doi.org/10.1007/s00170-007-1360-x>
19. Pham P (2009) Design of hybrid-kinematic mechanisms for machine tools. EPFL. <https://doi.org/10.5075/epfl-thesis-4314>
20. Pieper DL (1969) The kinematics of manipulators under computer control. Stanford University
21. Raghavan M (1991) The Stewart platform of general geometry has 40 configurations. In: vol 97768, *Proceedings of the ASME 1991 Design Technical Conferences. 17th Design Automation Conference: Vol 2 Computer-Aided Design, Mechanical Systems Simulation, and Analysis, Mechanisms, and Robotics.* Miami, Florida, USA. September 22–25, 1991. pp. 397–402. ASME. <https://doi.org/10.1115/1.2919188>
22. Siciliano B, Khatib O, Kröger T (2008) *Springer handbook of robotics*, vol 200. Springer. <https://doi.org/10.1007/978-3-540-30301-5>
23. Tsai L-W, Joshi S (2002) Kinematic analysis of 3-DOF position mechanisms for use in hybrid kinematic machines. *J Mech Des* 124(2):245–253. <https://doi.org/10.1115/1.1468860>
24. Wang Y (2006) “An incremental method for forward kinematics of parallel manipulators.” 2006 IEEE Conference on Robotics, Automation and Mechatronics, pp 1–5. <https://doi.org/10.1109/RAMECH.2006.252629>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

