

# Chapter 4

## Development of Haptic Systems

Christian Hatzfeld and Thorsten A. Kern

**Abstract** This chapter deals with the general design processes for the development of task-specific haptic systems. Based on known mechatronic development processes such as the V-model, a specialized variant for haptic systems is presented that incorporates a strong focus on the intended interaction and the resulting impacts on the development process. Based on this model, a recommended order of technical decisions in the design process is derived. General design goals of haptic systems are introduced in this chapter as well. These include stability, haptic quality, and usability that have to be incorporated in several stages of the design process. A brief introduction to different forms of technical descriptions for electromechanical systems, control structures, and kinematics is also included in this chapter to provide a common basis for the second part of the book.

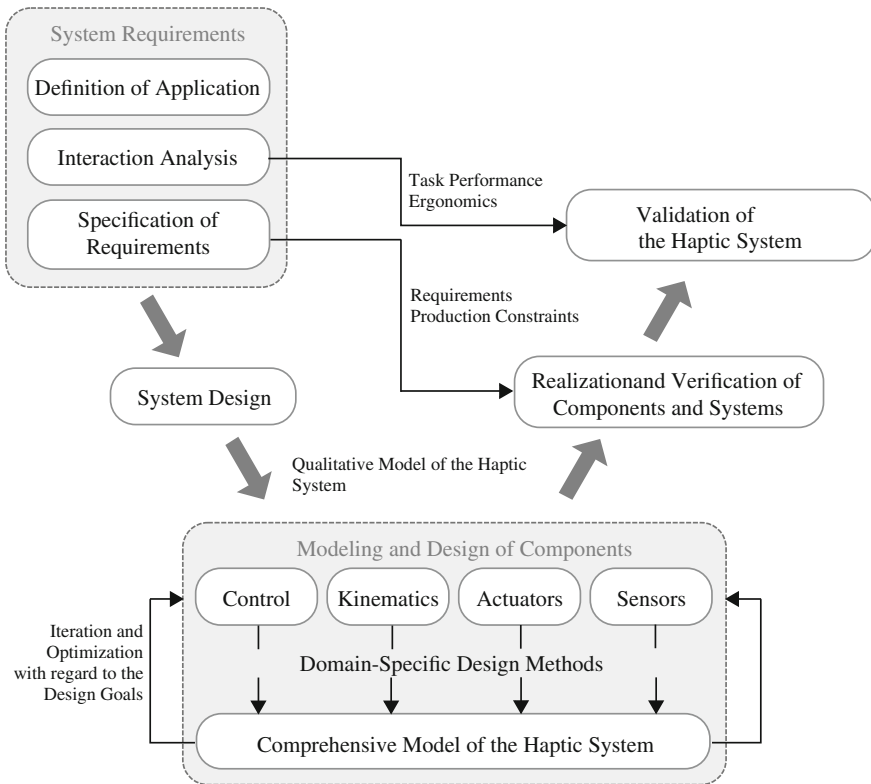
### 4.1 Application of Mechatronic Design Principles to Haptic Systems

Obviously, haptic systems are mechatronic systems incorporating powerful actuators, sophisticated kinematic structures, specialized sensors, and demanding control structures as well as complex software. The development of these parts is normally the focus of specialized areas of specialists, i.e., mechanical engineers, robotic specialists, sensor and instrumentation professionals, control and automation engineers, and software developers. A haptic system engineer should be at least able to understand the basic tasks and procedures of all of these professions, in addition to the required basic knowledge of psychophysics and neurobiology outlined in the previous chapters.

---

C. Hatzfeld (✉)  
Institute of Electromechanical Design, Technische Universität Darmstadt,  
Merckstr. 25, 64283 Darmstadt, Germany  
e-mail: c.hatzfeld@hapticdevices.eu

T.A. Kern  
Continental Automotive GmbH, VDO-Straße 1, 64832 Babenhausen, Germany  
e-mail: t.kern@hapticdevices.eu



**Fig. 4.1** Adaption of the V-model for the design of haptic systems

All of the above-mentioned professions use different methods, but generally agree on the same concepts in developing their parts of a haptic system. These can be integrated in some common known development design methods as for example, the V-MODEL for the development of mechatronic systems [16]. The model was originally developed for pure software design by the Federal Republic of Germany, but adapted to other domains as well. For the design of task-specific haptic systems, the authors detailed and extended some phases of the derivation of technical requirements based on [3] (Interaction Analysis) and [4] (Detailed Modeling of Mechatronic Systems). This adapted model is shown in Fig. 4.1. Based on this, five general stages are derived for the design of haptic systems. These stages are the basis for the further structure of this book and therefore detailed in the following sections.

The V-MODEL knows different variations, depending on the actual usage and scale of the developed systems. In this case, the above-mentioned variation was chosen over existing model variations, to be able to include additional steps in each stage of the V-MODEL. The resulting model is probably nearest to the W-MODEL for the design of adaptronic systems introduced by NATTERMANN AND ANDERL [8], because this model also includes an iteration in the modeling and design stage. It is further based

on a comprehensive data management system that includes not only information about interfaces and dependencies of individual components, but also a simulation model of each part. Since there is no comparable data basis for the design of haptic systems (that probably make use of a wider range of physical actuation and sensing principles than adaptronic systems till date), the W-MODEL approach is not directly transferable and more iterations in the modeling and design stage have to be accepted.

### ***4.1.1 Stage 1: System Requirements***

The first stage is used for the derivation of system requirements. For the design of task-specific haptic systems, a breakdown in three phases seems useful.

**Definition of Application** As described in Sect. 2.3, each haptic system should be assigned a well-defined application. This definition is the starting point of haptic system design and comes as a probably vague idea from the client ordering a task-specific haptic system and has to be detailed by the development engineer.

**Interaction Analysis** Based on the detailed application definition, the intended interaction of the user with the haptic system should be analyzed. For this step, the different concepts of interaction shown in Sect. 2.2 will provide useful vocabulary for the description of interactions. Based on this interaction, the intended grip configuration should be chosen and perceptual parameters for this configuration should be acquired, either from the known literature or by own psychophysical studies. At least, absolute thresholds and the  $\leftrightarrow$ JND should be known for the next steps, along with a model of the mechanical impedance of the intended grip configuration.

Another result of this phase is detailed and quantified interaction goal for application in terms of task performance and ergonomics. Possible categories of these goals are given in Chap. 13. If, for example, a communication system is designed, possible goals could be a certain amount of information transfer (IT) [5] and a decrease in cognitive load in an exemplary application scenario measured by the NASA Task Load Index [2].

**Specification of Requirements** Based on the predefined steps, a detailed analysis of technical requirements on the task-specific haptic system can be made. This should include all technical relevant parameters for the whole system and each component (i.e., actuators, sensors, kinematic structures, interfaces, control structure, and software design). Chapter 5 provides some helpful clusters depending on different interactions for the derivation of precise requirement definitions.

The result of this stage is at least a detailed requirement list. The necessary steps are detailed in Chap. 5. Further tools for the requirement of engineering can be used as well, but are not detailed further in this book.

### 4.1.2 Stage 2: System Design

In this stage, the general form and principles used in the system and its components have to be decided on. In general, one can find a vast number of different principles for components of haptic systems. During the technical development of haptic systems, the decisions on single components influence each other intensively. However, this influence is not identical between all components. For the engineer, it is necessary to proceed in the solution identification for each component, after having gained the knowledge of the requirements for the haptic system. It is obvious that according to a systematic development process, each solution has to be compared to the specifications concerning its advantages and disadvantages. The recommended procedure to deal with the components is the basis for the chapter structure of this section of the book and is summarized once again for completeness:

1. Decision about the control engineering structure of the haptic system based on the evaluation of the application (tactile or kinaesthetic), the impedance in idle state (masses  $>20$  g and friction acceptable), and the maximum impedance (stiffness  $>300$  N/m or smaller). This decision is based on the general structures described in Chap. 6 and the control structure of the haptic system described in Chap. 7.
2. Decision about the kinematics based on calculations of the workspace and the expected stiffness as detailed in Chap. 8.
3. Based on the now-known mechanical structure, the actuator design can be made. Chapter 9 deals with this topic, starting with an approximate decision about working principles based on performance criteria and detailed information about the common actuation principles for haptic systems.
4. Dependent on the chosen control engineering structure, the force sensor design can be performed parallel to the actuator design as detailed in Chap. 10.
5. Relatively uncritical for the design is the choice of the kinematic sensors (see Chap. 10).
6. The electronic interfaces are subordinate to all the decisions made before (see Chap. 11).
7. The software design of the haptic rendering itself, in many aspects, is so independent of prior technical decisions that it can be decoupled in almost all aspects from the rest of the design, when all specifications are made. Chapter 12 summarizes some typical topics for the design of this system component.

Nevertheless, it is vital to note that, e.g., the kinematics design cannot be realized completely decoupled from the available space for the device and the forces and torques—or rather the actuator. Additionally, kinematics directly influences any measurement technology as even displacement sensors have limitations on resolution and dynamics. The order suggested above for taking decisions has to be understood as a recommendation for processing the tasks; it does not free the design engineer from the responsibility to keep an overview of the subcomponents and their reciprocal influence.

A good possibility to keep track of this influence is the definition of clear interfaces between single components. This definition should include details about the form

of energy and data exchanged between the components and be further detailed in the course of the development process to include a clear definition of, for example, voltage levels, mechanical connections, standard interfaces, and connectors used.

### ***4.1.3 Stage 3: Modeling and Design of Components***

#### **4.1.3.1 Modeling of Components**

Based on the decisions from the preceding stage, the individual components can be modeled and designed. For this, general domain-specific methods and description forms are normally used, which are further described in Sect. 4.3. This step will first result in a model of the component that will include all relevant design parameters that influence the performance and design of the component. Some of these parameters can be chosen almost completely freely (i.e., control and filter parameters), while others are limited by purchased parts in the system component (one will, for example, only find sensors with different but fixed ranges as well as actuators with fixed supply voltages, etc.).

#### **4.1.3.2 Comprehensive Model of the Haptic System**

In a second step, a more general model of the component should be developed, which exhibits similar interfaces to adjacent components like those defined in the preceding Sect. 4.1.2. Furthermore, this model should only include the most relevant design parameters to avoid excessive parameter sets.

When the interfaces of adjacent components match, the models of all components can be combined to a comprehensive model of the haptic system with general haptic input and output definitions (see Fig. 2.33) and relevant design parameters for each individual component. Normally, a large number of components are involved in these comprehensive models. For a teleoperation system, one can roughly calculate two actuators, two kinematic structures, two positioning sensors for actuator control, one force sensor, and the corresponding power and signal processing electronics for *each*  $\leftrightarrow$ DoF with the resulting modeling and simulation effort.

Even if they are very large, such models are advisable to optimize the haptic system with respect to the below-mentioned design goals such as stability and haptic quality. Only with a comprehensive model can one evaluate the intercomponent influences on these design goals. Based on the descriptions of the system structure given in Chap. 7, the optimization of the comprehensive model will lead to additional requirements on the individual components or modifications of the prior-defined interfaces between components. These should also be documented in the requirement list.

One has to keep in mind that all parameters are prone to errors, especially variances with regard to the nominal value and differences between the real part and the (somewhat) simplified model. During optimization of the comprehensive model, robustness of the results with regard to these errors has to be kept in mind.

#### 4.1.3.3 Optimization of Components

Based on the results of the optimized comprehensive model, the individual components of a haptic system can be further optimized. This step is not only needed when there is a change of interface definitions and requirements of single components, but also when it is normally necessary to ensure certain requirements of the system that are not dependent on a single component only. Examples are the overall stiffness of the kinematic structure, the mass of the moving parts of the system, and, of course, the tuning parameters of control loops.

For the optimization of components, typical mechatronic approaches and techniques can be used (see for example [9, 18] and Sect. 4.3). Further aspects such as standard conformity, security, recycling, wearout, and suitability for production have to be taken into account in this stage, too.

In practice, the three parts of *Stage 3: Modeling and Design of Components* are not used sequentially, but with several iterations and branches. Experience and intuition of the developer will guide several aspects influencing the success and duration of this stage, especially the selection of meaningful parameters and the depth of modeling of each component. Currently, many software manufacturers work on a combination of different model abstraction levels (i.e.,  $\leftrightarrow$  single input, single output (SISO) systems, network parameter descriptions, finite element models) into a single CAE software with the ability not only to simulate, but also to optimize the model. While this is already possible to a certain amount in commercial software products (e.g., ANSYS<sup>TM</sup>), the ongoing development in these areas will be very useful for the design of haptic systems.

#### 4.1.4 Stage 4: Realization and Verification of Components and System

Based on the optimization, the components can be manufactured and the haptic system can be assembled. Each manufactured component and the complete haptic system should be tested against the requirements, i.e., a verification should be made. Additionally, other design goals such as control stability and transparency (if applicable) should be tested. Due to the above-mentioned interaction analysis (see Sect. 5.2 below for more details), this step ensures that the system generates perceivable haptic signals to the user without any disturbances due to errors. To compare the

developed haptic system with others, objective parameters as described in Chap. 13 can be measured.

### 4.1.5 Stage 5: Validation of the Haptic System

While step 4 ensures that the system is developed correctly with respect to the expected functions and the requirements, this step checks whether the correct system was developed. This is simply made by testing the evaluation criteria defined in the interaction analysis and comparison with other systems with haptic feedback in a user test.

This development process ensures that time-intensive and costly user tests are only conducted in the first and last stages, while all other steps only rely on models and typical engineering tools and process chains. With this detailing of the V-MODEL, the general mechatronic design process is extended in such a way that the interaction with the human user is incorporated in an optimized way in terms of effort and development duration.

## 4.2 General Design Goals

There are a couple basic goals for the design of haptic systems that can be applied with various extensions to all classes of applications. They do not lead to rigorous requirements, but are helpful to keep all of these in mind when designing an haptic system to ensure a successful product.

**Stability** Stability in the sense of control engineering should be archived by all haptic systems. It affects the safety of a haptic device as well as the task performance of a haptic system and the interactions performed with it. To ensure stability while improving haptic transparency is the main task of the haptic system control. This is further detailed in Chap. 7.

**Haptic Quality** To ensure sufficient haptic quality is the second design goal of a haptic system. In general, each system should be able to convey the haptic signals of human-machine interaction without conveying the own mechanical properties to the user. For teleoperation systems, one will find the term *haptic transparency* for this preferable behavior. Analogous to the visual transparency of an ideal window, an ideal haptic teleoperation system will let the user feel exactly the same mechanical properties that are exposed to the manipulator of the teleoperation system. Since physical parts of a haptic system exhibit real physical behavior that cannot be neglected, haptic quality is a control task as well to compensate for this real behavior. It is therefore detailed in Chap. 7.

**Usability** Since haptics is considered as an interaction modality in this book, all usability considerations of human–machine interfaces should be treated as a design goal. These goals are described in the ISO 9241 standard series<sup>1</sup> and demand *effectiveness* in fulfilling a given task, *efficiency* in handling the system, and *user satisfaction* when working with the system.

Usability has therefore been considered in almost all stages of the development process. This includes the selection of suitable grip configurations that prevent fatigue and allow comfortable use of the system, the definition of clearly distinguishable haptic icons that are not annoying when occurring repeatedly, and the integration of assistive elements such as arm rests. It is advisable to provide for individual adjustment, since this contributes to the usability of a system. This applies to mechanical parts such as adjustable arm rests as well as information-carrying elements such as haptic icons. Methods to assess some of these criteria mentioned are given in Chap. 13 as well as in the standard literature on usability for human–machine interaction, as for example [1].

For the design of haptic systems, the following design principles derived from PREIM’s principle for the design of interactive software systems can assist in the development of haptic systems with higher usability [10]:

- Get information about potential users and their tasks;
- Focus on the most important interactions;
- Clarify the interaction options;
- Show system states and make them distinguishable;
- Build an adaptive interface;
- Assist users in developing a mental model, i.e., by consistency of different task primitives;
- Avoid surprising the user;
- Avoid keeping a large amount of information in the user’s memory.

### 4.3 Technical Descriptions of Parts and System Components

Since the design of haptic systems involves several scientific disciplines, one has to deal with different description languages according to the discipline’s culture. This section gives a brief introduction into different description languages used in the design of control, kinematics, sensors, and actuators. It is not intended to be sufficient, but to give an insight into the usage and advantages of the different descriptions for components of haptic systems.

---

<sup>1</sup> The ISO 9241 primarily deals with human–computer interaction in a somewhat limited view of the term “computer” with a strong focus on standard workstations. The general concepts described in the standard series can be transferred to haptics nevertheless, and the ISO 9241-9xx series deals with haptics exclusively.



### 4.3.1 Single Input, Single Output Descriptions

One of the simplest forms of modeling for systems and components are  $\hookrightarrow$ SISO descriptions. They only consider a single input and a single output with time dependency, i.e., a time-varying force  $F(t)$ . The description also includes additional constant parameters and derivatives with respect to time of inputs and outputs. If considering a DC motor for example, a SISO description would be the relation between the output torque  $M_{\text{out}}(t)$  evoked by a current input  $i_{\text{in}}(t)$  as shown in Eq. 4.1.

$$\begin{aligned} M_{\text{out}}(t) &= k_M \cdot i_{\text{in}}(t) \\ \Rightarrow h(t) &= \frac{M_{\text{out}}(t)}{i_{\text{in}}(t)} = k_M \end{aligned} \quad (4.1)$$

The output torque is related to the input current by the transfer function  $h(t)$ . In this case, the transfer function is just the motor constant  $k_M$  that is calculated from the strength of the magnetic field, the number of poles and windings, and geometric parameters of the rotor among others. It is normally given in the data sheet of the motor.

SISO descriptions are mostly given in the LAPLACE domain, i.e., a transformation of the time-domain transfer function  $h(t)$  into the frequency-domain transfer function  $G(s) \bullet \dashrightarrow h(t)$  with the complex LAPLACE operator  $s = \sigma + j\omega$ . These system descriptions are widely used in control theory to assess the stability and quality of control. However, for the design of complex systems with different components, SISO descriptions have some drawbacks.

- Since only single input and output variables are used, one cannot describe the flow of energy by SISO descriptions accordingly. This is obvious from the above example of a DC motor: Usable output torque will decrease as the revolution speed of the motor increases, since the amount of energy available is limited by the thermal dissipation capabilities of the motor and the counter electromotive force. This behavior cannot be incorporated in Eq. (4.1), since it involves more than one time-dependent input variable.
- When using SISO descriptions for different components that are arranged in a signal and/or energy transmission chain, one has to adjust the interfaces between components accordingly. This complicates the exchange of single components in the transmission chain. Consider an actuator driving a kinematic structure. The exchange of an electrodynamic principle for a piezoelectric principle will require a new SISO description of the kinematic structure, since a input current to the actuator will evoke different kinds of outputs (a force in the case of the electrodynamic principle and an elongation for the piezoelectric principle).

To overcome these disadvantages, one can extend the SISO description to multiple input and multiple output systems (MIMO). For the description of haptic systems, a special class of MIMO systems is advisable, the description based on network parameters as outlined in the following Sect. 4.3.2.

**Table 4.1** Analogy between electrical and mechanical network descriptions

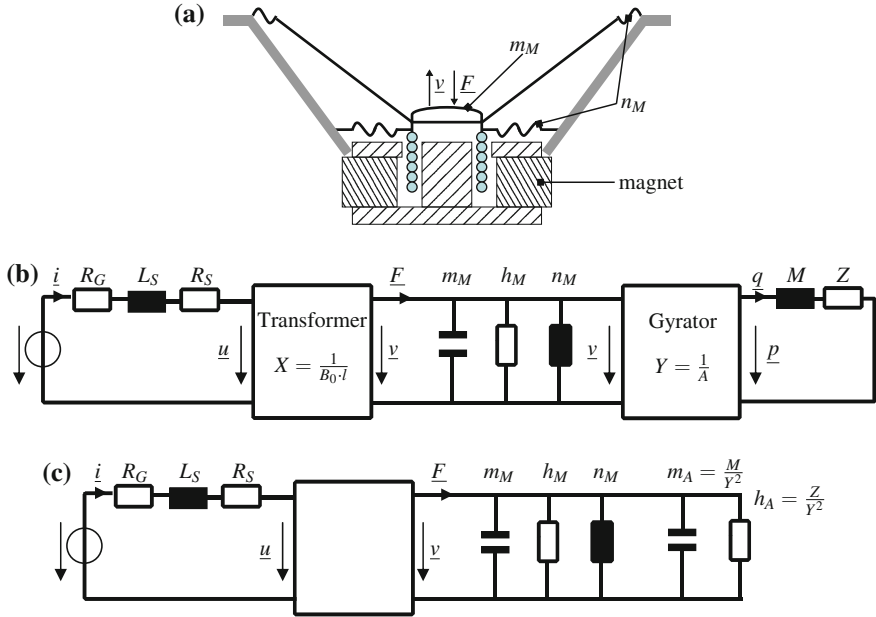
Electrical domain			Mechanical domain	
Parameter	Symbol		Parameter	Symbol
Voltage	$\underline{i}$	$\Leftrightarrow$	Velocity	$\underline{v}$
Current	$\underline{u}$	$\Leftrightarrow$	Force	$\underline{F}$
Inductivity	$L$	$\Leftrightarrow$	Compliance	$n$
Capacity	$C$	$\Leftrightarrow$	Mass	$m$
Resistance	$R$	$\Leftrightarrow$	Viscous damping/friction	$h = \frac{1}{r}$
Impedance	$\underline{Z} = \frac{\underline{u}}{\underline{i}}$	$\Leftrightarrow$	Admittance (mobility)	$\underline{h} = \frac{1}{\underline{Z}}$
Admittance	$\underline{Y} = \frac{1}{\underline{Z}}$	$\Leftrightarrow$	Impedance	$\underline{z} = \frac{\underline{F}}{\underline{v}}$

These drawbacks do not necessarily mean that SISO descriptions have no application in the modeling of haptic systems: Despite the usage in control design, they are also useful to describe system parts that are not involved in extensive exchange of energy, but primarily in the exchange of information. Consider a force sensor placed at the tip of the manipulator of a haptic system: While the sensor compliance will effect the transmission of mechanical energy from  $\leftrightarrow$ TCP to the kinematic structure of the manipulator (and should therefore be considered with a more detailed model than a SISO description), the transformation of forces into electrical signals is mainly about information. It is therefore sufficient to use a SISO description for this function of a force sensor.

### 4.3.2 Network Parameter Description

The description of mechanical, acoustic, fluidic, and electrical systems based on lumped network parameters is based on the similar topology of the differential equations in each of these domains. A system is described by several network elements, which are locally and functionally separated from each other and exchange energy via predefined terminals or ports. To describe the exchange of energy, each considered domain exhibits a flow variable in direct connection with neighboring ports (e.g., current in the electrical domain and force in translational mechanics) and an effort variable (e.g., voltage, respectively, velocity between two arbitrary ports of the network). Table 4.1 gives the mapping of electrical and translational mechanical elements. Historically, there are two analogies between these domains. The one used here depicts physical conditions the best. There is, however, a single incongruent point: The definition of the mechanical impedance is the quotient of flow variable and effort variable.

To couple different domains, lossless transducers are used. Because they are lossless, systems in different domains can be transformed into a single network, which can be simulated with an extensive number of simulation techniques known from electrical engineering like SPICE. The transducers can be divided into two general



**Fig. 4.2** **a** Network model of an electrodynamic loudspeaker. The system consists of an electrical system, the electrodynamic transducer with transformer constant  $X$ , the mechanical parts of the moving parts, the mechanical–acoustic transducer with gyration constant  $Y$ , and the properties of the acoustic system. **b** shows the corresponding network model and **c** the network model, when acoustic network elements are transformed in equivalent mechanical elements

classes. The first class called *transformer* links the effort variable of domain A with the effort variable of domain B. A typical example for a transformer is a electrodynamic transducer that can be described as shown in Eq. 4.2 with the transformer constant  $X = \frac{1}{B_0 \cdot l}$ :

$$\begin{pmatrix} v \\ F \end{pmatrix} = \begin{pmatrix} \frac{1}{B_0 \cdot l} & 0 \\ 0 & B_0 \cdot l \end{pmatrix} \cdot \begin{pmatrix} u \\ i \end{pmatrix} \tag{4.2}$$

$B_0$  denotes the magnetic flux density in the air gap of the transducer, and  $l$  denotes the length of the electrical conductor in this magnetic field. Further details about these transducers are given in Chap. 9. If different domain networks are transformed into each other by means of a transformer, the network topology stays the same and the transformed elements are weighted with the transformer constant. This is shown in Fig. 4.2 with the example of an electrodynamic loudspeaker and applied to electrodynamic actuators in Sect. 9.2.

The other class of transducers is called *gyrator*, coupling the flow variable from domain A with the effort variable from domain B and vice versa. The coupling is described with the transformer constant  $Y$ , and examples (not shown here) include electrostatic actuators and transducers that change mechanical energy into fluidic

energy. If different domain networks are transformed, the network topology changes, and series connections become parallel and vice versa. The single elements change as well; for a gyratory transformation between mechanical and electrical domains, an inductor will become a mass and a compliance will turn into a capacitance. A common application for gyratory transformations is the modeling of piezoelectric transducers. This is shown in Sect. 9.3.4.1 in the course of the book.

An advantage of this method is the consideration of influences from other parts in the network, a property that cannot be provided by the representation with SISO transfer functions. On the other side, this method will only work for linear time-invariant systems. Mostly, a linearization around an operating point is made to use network representations of electromechanical systems. Some time dependency can be introduced with switches connecting parts of the network at predefined simulation times. Another constraint is the size of the systems and components modeled by the network parameters. If size and wavelength of the flow and effort variables are in similar dimensions as the system itself, the basic assumption of lumped parameters cannot hold anymore. In that case, distributed forms of lumped parameter networks can be used to incorporate some wave-line transmission properties.

In haptics, network parameters are, for example, used for the description of the mechanical user impedance  $Z_{\text{user}}$  as shown in Chap. 3 and the condensed description of kinematic structures and the optimization of the mechanical properties of sensors and actuators as shown above. Further information about this method can be found in the work of TILMANN [14, 15] and LENK ET AL. [7], from which all information in this section was taken.

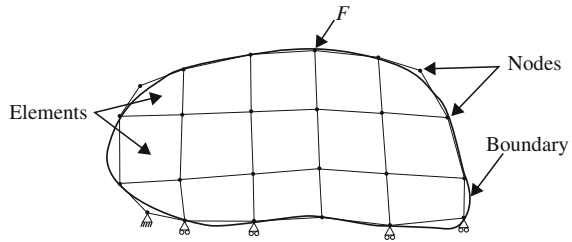
### 4.3.3 Finite Element Methods

↔ Finite element methods (FEM) are mathematical tools to evaluate ↔ partial differential equations (PDE). Since a lot of physical principles are described by partial differential equations, this technique is used throughout engineering to calculate mechanical, thermal, electromagnetic, and acoustic problems [6].

The use of the finite element method requires a discretization of the whole domain, thereby generating several finite elements with finite element nodes as shown in Fig. 4.3. Furthermore, boundary conditions have to be defined for the border of the domain, and external loads and effects are included in these boundary conditions.

Put very simply, FE analysis will run through the following steps: To solve the PDE on the chosen domain, first a partial integration is performed on the differential equations multiplied with a test function. This step leads to a weak formulation of the partial differential equation (also called natural formulation) that incorporates the NEUMANN boundary conditions. Discretization is performed on this natural formulation, leading to a set of PDEs that has to be solved on each single element of the discretized domain. By assuming a certain type of appropriate shape or interpolation function for the PDE on each element, a large but sparse linear matrix is constructed that can be solved with direct or iterative solvers, depending on the size of the matrix.

**Fig. 4.3** Domain, elements, nodes, and boundary conditions of a sample FEM problem formulation



There are a lot of commercial software products that perform FEM in different engineering fields. They normally include a preprocessor that takes care of discretization, material parameters and boundary conditions, a solver, and a post-processor that turns the solver's results into a meaningful output. For the quality of results of FEM, the choice of the element types depending on geometry of the considered domain and the kind of analysis and mathematical solver used is of great importance.

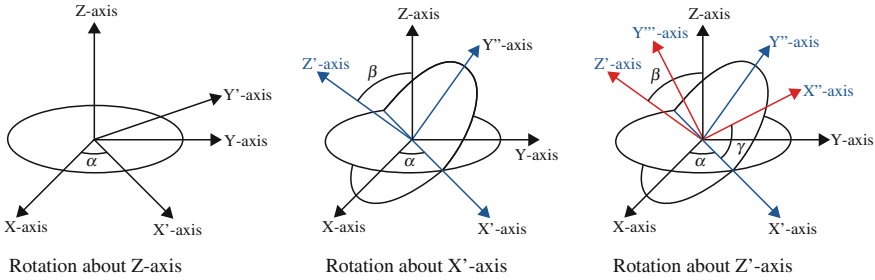
The advantages of the FE method are the treatment of nonlinear material properties, the application to complex geometries, and the versatile analysis possibilities that include static, transient, and harmonic analysis [19]. The aspect of discretization yields high computational effort, and also spatial resolution of the physical value on investigation.

To overcome the disadvantages of FEM, there are some extensions to the method: The *combined simulation* maps FE results onto network models that are further used in network-based simulations of complex systems [13, 20]. The advantage is the high spatial resolution of the calculation on the required parts only and the resulting higher speed. The data exchange between FE and network model is made by the user. The *coupled simulation* incorporates an automated data exchange between FE and network models at runtime of the simulation. At the moment, many companies work on the integration of this functionality in program packages for FE and network model analysis to allow for multidomain simulation of complex systems.

The application of  $\leftrightarrow$  finite element model (FEM) in haptics can be found in the design of force sensors (see Sect. 10.1), the evaluation of thermal behavior of actuators, and the structural strength of mechanical parts.

#### 4.3.4 Description of Kinematic Structures

A description of the pose, i.e., the position and orientation of a rigid body in space, is a basic requirement to deal with kinematic structures and to optimize their properties. If considering Euclidean space, six coordinates are required to describe the pose of a body. This is normally done by defining a fixed reference frame  $i$  with an origin  $O_i$  and three orthogonal basis vectors  $(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$ . The pose of a body with respect to the reference frame is described by the differences in position and orientation. The difference in position is also called displacement and describes the change in



**Fig. 4.4** Rotation of a coordinate frame based on Euler angles  $(\alpha, \beta, \gamma)$  in Z–X–Z order

position of the origin  $O_j$  of another coordinate frame  $j$  that is fixed to the body. The orientation is described by the angle differences between the two sets of basis vectors  $(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i)$  and  $(\mathbf{x}_j, \mathbf{y}_j, \mathbf{z}_j)$ . This rotation of the coordinate frame  $j$  with respect to the reference frame  $i$  can be described by the rotation matrix  ${}^j\mathbf{R}_i$  as given in Eq. 4.3.

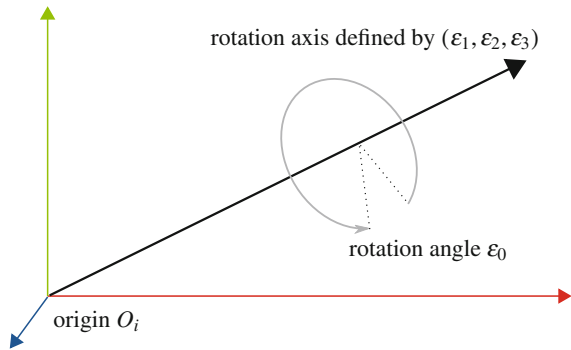
$${}^j\mathbf{R}_i = \begin{pmatrix} \mathbf{x}_i \cdot \mathbf{x}_j & \mathbf{y}_i \cdot \mathbf{x}_j & \mathbf{z}_i \cdot \mathbf{x}_j \\ \mathbf{x}_i \cdot \mathbf{y}_j & \mathbf{y}_i \cdot \mathbf{y}_j & \mathbf{z}_i \cdot \mathbf{y}_j \\ \mathbf{x}_i \cdot \mathbf{z}_j & \mathbf{y}_i \cdot \mathbf{z}_j & \mathbf{z}_i \cdot \mathbf{z}_j \end{pmatrix} \tag{4.3}$$

While the rotation matrix contains nine elements, only three parameters are needed to define the orientation of a body in space. Although there are some mathematical constraints on the elements of  ${}^j\mathbf{R}_i$  that ensure the equivalence, several minimal representations of rotations can be used to describe the orientation with less parameters (and therefore less computational effort is required when computing kinematic structures). In this book, only three representations are discussed further, the description by *Euler angles*, *fixed angles*, and *quaternions*.

**Euler Angles** To minimize the number of elements needed to describe a rotation, the Euler angle notation uses three angles  $(\alpha, \beta, \gamma)$  that each represents a rotation about the axis of a moving coordinate frame. Since each rotation depends on the prior rotations, the order of rotations has to be given as well. Typical orders are Z–Y–Z and Z–X–Z rotations shown in Fig. 4.4. The description by Euler angles exhibits singularities, when the first and last rotations occur about the same axis. This is a drawback when one has to describe several consecutive rotations and when describing motion, i.e., deriving velocities and accelerations.

**Fixed Angles** Fixed angle descriptions are basically the same as Euler angle descriptions, and the rotation angles  $(\psi, \theta, \phi)$  describe, however, the rotation about the fixed axes of the reference frame. Also known as *yaw*  $\psi$  around the  $\mathbf{x}_i$ -axis, *pitch*  $\theta$  around the  $\mathbf{y}_i$ -axis, and *roll*  $\phi$  around the  $\mathbf{z}_i$ -axis, the fixed angles exhibit the same singularity problem as the Euler angles.

**Fig. 4.5** Rotation of a frame defined by a quaternion



**Quaternions** To overcome this singularity problem, quaternions are used in the description of kinematic structures. Mathematically, also known as Hamilton numbers  $\mathbb{H}$ , they are an extension of the real number space  $\mathbb{R}$ . A quaternion  $\epsilon$  is defined as

$$\epsilon = \epsilon_0 + \epsilon_1 i + \epsilon_2 j + \epsilon_3 k$$

with the scalar components  $\epsilon_0, \epsilon_1, \epsilon_2,$  and  $\epsilon_3$  and the operators  $i, j,$  and  $k$ . The operators fulfill the combination rules shown in Eq. (4.4) and therefore allow associative, commutative, and distributive addition as well as associative and distributive multiplication of quaternions.

$$\begin{aligned} ii = jj = kk &= -1 \\ ij = k, \quad jk = i, \quad ki = j \\ ji = -k, \quad kj = -i, \quad ik = -j \end{aligned} \tag{4.4}$$

One can imagine a quaternion as the definition of a vector  $(\epsilon_1, \epsilon_2, \epsilon_3)$  that defines the axis; the frame is rotated about the scalar part  $\epsilon_0$  defining the amount of rotation. This is shown in Fig. 4.5. By dualization, quaternions can be used to describe the complete pose of a body in space, i.e., rotation and displacement. Further forms of kinematic descriptions, for example, the description based on screw theory can be found in [17], on which this section is based primarily and other works like [11, 12].

## Recommended Background Reading

- [4] K. Janschek: **Mechatronic systems design: methods, models, concepts**. Springer, Heidelberg, 2012.  
*Design methodologies for mechatronic systems that can also be applied to haptic systems.*
- [6] M. Kaltenbacher: **Numerical simulation of mechatronic sensors and actuators**. Springer, Heidelberg, 2007.  
*Broad overview about finite element methods and the application to sensors and actuators.*
- [7] A. Lenk et al. (Eds.): **Electromechanical Systems in Microtechnology and Mechatronics: Electrical, Mechanical and Acoustic Networks, their Interactions and Applications**. Springer, Heidelberg, 2011.  
*Introduction to the network element description methodology.*

## References

1. Boy GA (ed) (2011) The handbook of human-machine interaction—a human-centered approach. Ashgate, Farnham
2. Hart S, Staveland L (1988) Development of NASA-TLX (task load index): results of empirical and theoretical research. Human mental workload 1:139–183. doi:[10.1016/S0166-4115\(08\)62386-9](https://doi.org/10.1016/S0166-4115(08)62386-9)
3. Hatzfeld C (2013) Experimentelle analyse der menschlichen Kraftwahrnehmung als ingenieurtechnische Entwurfsgrundlage für haptische Systeme. Dissertation, Technische Universität Darmstadt. <http://tuprints.ulb-tu-darmstadt.de/3392/>. München: Dr. Hut Verlag, 2013. ISBN: 978-3-8439-1033-0
4. Janschek K (2012) Mechatronic systems design: methods, models, concepts. Springer, Berlin. ISBN: 978-3-642-17531-2
5. Jones L, Tan H (2013) Application of psychophysical techniques to haptic research. IEEE Trans Haptics 6:268–284. doi:[10.1109/TOH.2012.74](https://doi.org/10.1109/TOH.2012.74)
6. Kaltenbacher M (2007) Numerical simulation of mechatronic sensors and actuators. Springer, Berlin. ISBN: 978-3-540-71359-3. doi:[10.1007/978-3-540-71360-9](https://doi.org/10.1007/978-3-540-71360-9)
7. Lenk A (ed) (2011) Electromechanical systems in microtechnology and mechatronics: electrical, mechanical and acoustic networks, their interactions and applications. Springer, Heidelberg
8. Nattermann R, Anderl R (2013) The W-model-using systems engineering for adaptronics. Procedia Comput Sci 16:937–946. doi:[10.1016/j.procs.2013.01.098](https://doi.org/10.1016/j.procs.2013.01.098)
9. Pahl G, Wallace K, Blessing L (2007) Engineering design: a systematic approach. Springer, Berlin. ISBN: 978-3540199175
10. Preim B (1999) Entwicklung interaktiver Systeme. Grundlagen, Fallbeispiele und innovative Anwendungsfelder. Springer, Heidelberg. ISBN: 978-3540656487
11. Sciavicco L (2000) Modelling and control of robot manipulators. Springer, Berlin. ISBN: 978-1-4471-0449-0
12. Selig J (2005) Geometric fundamentals of robotics. Springer, Berlin. ISBN: 978-0-387-27274-0
13. Starke E (2009) Kombinierte Simulation—eine weitere Methode zur Optimierung elektromechanischer Systeme. Dissertation. Technische Universität Dresden. [http://tudresden.de/die\\_tu\\_dresden/fakultaeten/fakultaet\\_elektrotechnik\\_und\\_informationstechnik/ihtm/mst/Veroeffentlichungen\\_MST/Archiv\\_MST/mst\\_2010?fis\\_type=publikation&fis\\_id=b1847](http://tudresden.de/die_tu_dresden/fakultaeten/fakultaet_elektrotechnik_und_informationstechnik/ihtm/mst/Veroeffentlichungen_MST/Archiv_MST/mst_2010?fis_type=publikation&fis_id=b1847)



14. Tilmans H (1996) Equivalent circuit representation of electromechanical transducers: I. lumped parameter systems. *J Micromech Microeng* 6: 157. doi:[10.1088/0960-1317/6/1/036](https://doi.org/10.1088/0960-1317/6/1/036)
15. Tilmans H (1997) Equivalent circuit representation of electromechanical transducers: II. Distributed parameter systems. *J Micromech Microeng* 7(4) 285. doi:[10.1088/0960-1317/7/4/005](https://doi.org/10.1088/0960-1317/7/4/005)
16. VDI 2206: Entwicklungsmethodik für mechatronische Systeme. Berlin: Verband Deutscher Ingenieure (VDI), 2004. <http://www.vdi.de/technik/fachthemen/produkt-und-prozessgestaltung/fachbereiche/produktentwicklungund-mechatronik/themen/rilis-mechatronische-systeme/richtlinie-vdi-2206-entwicklungsmethodik-fuer-mechatronische-systeme/>
17. Waldron K, Schmiedeler J (2008) Kinematics In: Siciliano B, Khatib O (eds) Springer handbook of robotics. Springer, Berlin, pp 9–33. doi:[10.1007/978-3-540-30301-5\\_2](https://doi.org/10.1007/978-3-540-30301-5_2)
18. Janschek K (2012) Mechatronic systems design: methods, models, concepts. In: Design methodologies for mechatronic systems that can also be applied to haptic systems. Springer, Heidelberg.
19. Kaltenbacher M (2007) Numerical simulation of mechatronic sensors and actuators. In: Broad overview about finite element methods and the application to sensors and actuators. Springer, Berlin
20. Lenk A (ed) (2011) Electromechanical systems in microtechnology and mechatronics: electrical, mechanical and acoustic networks, their interactions and applications. Springer, Heidelberg