Linear Systems Description

Amir Karniel and Gideon F. Inbar

Introduction

The systems approach is a widely used practice in modeling artificial as well as natural phenomena. Each process or sub-process is viewed as an input-output system, as described graphically in Fig.1.

This approach is used extensively in engineering, for example in modeling electronic and mechanical systems and in chemical process description. In this chapter we describe this approach and its application to biological systems in general and the nervous system in particular. The systems approach can be used as a modeling tool to comprehend the function of the system and to produce a hypothetical model which can be tested in experiments. It is useful in describing and characterising experimental results, at times by relating the anatomical and physiological properties to the measured variables (see for example the muscle spindle transfer function, Houk 1963). Mathematical modeling of part of the neurological system can be used to study that and other parts by simulation. (See McRuer et al. 1968 for an example of combination of models to the motor neurons, the muscle and the muscle spindle in a closed loop). The systems approach modeling is also useful for building interfaces to engineering systems in order to develop measurement devices or artificial organs such as hearing aids, pacemakers and artificial limbs.

Linear systems are highly popular models due to their simplicity and since they are very convenient for mathematical analysis. Beyond the above technical advantage, many systems can be modeled as linear systems at least for a limited range of operation.

Let us begin with a short description of the terminology of this field. Figure 1 describes the general notion of an input-output system in a block diagram. The input is u and the output is y. They usually describe a physical quantity as potential, current force or position. In this chapter their value may be scalar real numbers, or vectors of real

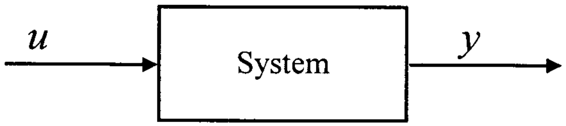


Fig. 1. Input-Output system. u is the input and y is the output. They can be scalars or vectors that represent physical values such as potential, current, force, or position. They can also be functions of time, that is trajectories. The output is generally a function of the input and possibly of the state of the system.

Amir Karniel, Technion – Israel Institute of Technology, Department of Electrical Engineering, Haifa, Israel (phone: 972-4–8294662; fax: 972-4–8323041; e-mail: karniel@tx.technion.ac.il)
Gideon F. Inbar, Technion – Israel Institute of Technology, Department of Electrical Engineering, Haifa, Israel (phone: 972-4–8294718; fax: 972-4–8323041; e-mail: inbar@ee.technion.ac.il)

numbers in the case of multiple inputs and outputs. The inputs and outputs can also be functions of time, that is trajectories, either discrete or continuous. We will concentrate on deterministic mapping systems, which means that for a specific input there is a specific singular output, i.e., the output is a function of the input, $y=f(u)$. Let us add two important qualifiers for a system:

- A system is *time-invariant*, roughly, if the system properties do not change with time.
- A system is *linear* if it satisfies the property of superposition, that is, for any couple of inputs and outputs $y^1=f(u^1)$ and $y^2=f(u^2)$, the equation $ay^1+by^2=f(au^1+bu^2)$ will be satisfied for any couple of scalars a and b.

A system that satisfies both of these properties is naturally called a linear time-invariant (LTI) system. All the systems in this chapter are LTI unless otherwise mentioned.

In this chapter we introduce linear systems, static and dynamic, then we move to a detailed description of each step in describing, modeling and analyzing linear systems, as is outlined in the next section.

Outline

In this section we combine a short outline of this chapter with a description of the biological system modeler work. The main stages are illustrated in Fig. 2.

The first step in biological modeling by the systems approach is to choose or define the inputs and outputs. This can be done by inspection of the anatomical structure of the system and incorporating prior knowledge about the physiological function of the modeled system. Such inspection can lead to an electrical or mechanical model, or sometimes directly to an equation that describes relations between inputs and outputs. For example, by looking at the physiological structure of a small region in the retina, one can choose the input to be the light intensity and the output to be the firing rate of a related axon. Then one can suggest a simple linear model such as $f = a \cdot I$, where f is the firing rate, I is the light intensity, and a is a constant that is sometimes called a "gain" . Another suggestion may include a more sophisticated electrical model of the nerve cell, which finally produces a differential equation that relates the output to the inputs. Part 1 deals with the first suggestion, i.e., static relations, where linear artificial neural networks are described and an example for an associative memory is given. The major part of the chapter is about dynamic models, that is, where the inputs and outputs are functions of time and the output may be a function of previous events and not only of the current input. Part 2 introduces dynamic linear systems and Part 3 deals with electrical and mechanical models, and how to derive the differential equations from the graphic description of the systems model; these procedures are based on Kirchhoff's and Newton's Laws. This part contains various examples for modeling the nervous system, synapses and muscles. Once we have a model, that is, a set of equations that describes the biological system, we can check the behavior of the model in various cases in order to produce hypotheses that can be later checked on real data from the biological system. Laplace and Z transforms are powerful tools to analyze and manipulate linear systems and they are the subjects of Part 4.

A model usually contains some parameters, for example, the parameter a in the simple model of the retina above. One of the objectives of the modeler is to estimate the values of these parameters. This estimation is based on measurements of the system's input and output. An estimation method for linear systems is described in Part 5. Part 6 describes how to integrate linear models of subsystems in a block diagram in order to get a model of the complete system; this method is used extensively in control theory, and therefore examples from the field of motor control and of temperature regulation are given. Models of artificial means can also be incorporated, such as measurement devices, artificial organs or functional neuromuscular stimulation for the paralyzed patient. Recently it has become a fashion to discuss nonlinear models and chaos, which seems to appear in many natural systems. This observation is correct; however, in many cases, the powerful linear system description tools can still be used in order to describe and analyze nonlinear systems. This is the subject of Part 7. The most common tool to handle nonlinear systems is linearization, which is finding a linear model that is similar to the nonlinear system in some area of interest. Other options are linear time-varying or parameter-varying models such as the Hodgkin and Huxley membrane model; two other options are pre- or post-processing of the linear system which are the terms that are used in the field of neural computation. One should note that the work of the modeler usually consists of a few iterations of improving the model, designing new experiments to obtain new data, estimating the parameters and analyzing the results, as illustrated graphically in Fig. 2.

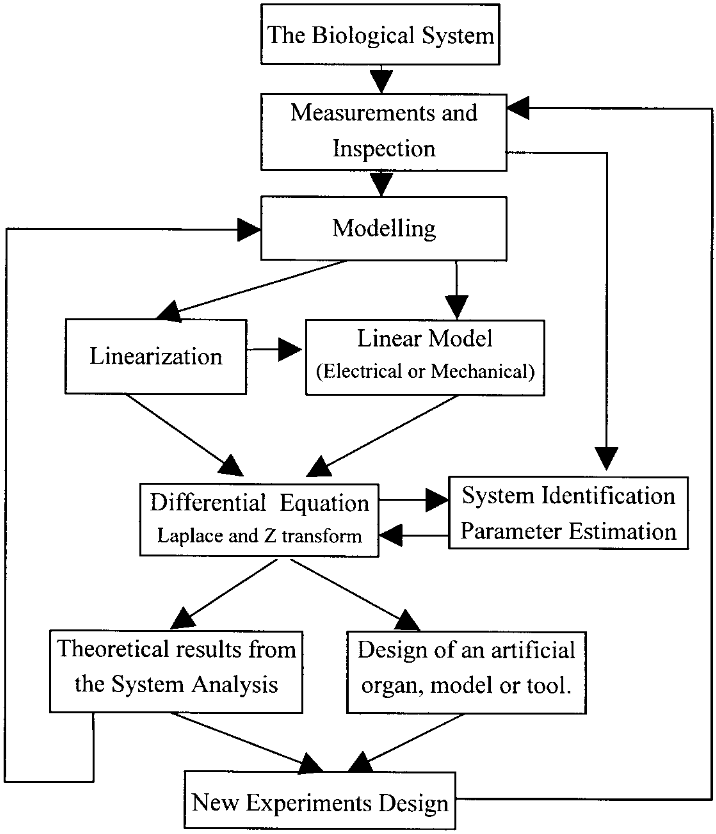


Fig. 2. Outline of linear systems description: The first step is to choose or define the inputs and the outputs and to acquire data about the system from measurements and inspection of the physical (i.e., anatomical) structure of the system. Then one can move directly to system identification by trying to fit a linear model to the data, or first draw a physical or electrical model and then estimate its parameters. The given system may be linear or nonlinear; in the case of a linear system we can describe it with a mechanical or an electrical equivalent model and then write the differential equations. In the case of a nonlinear system we can use a linear approximation and then continue as if we had a linear system. The linear time-invariant difference equation can be transformed to the Laplace domain to get the transfer function. These functions can be used for various goals, such as system identification, artificial control and modeling in order to anticipate the system behaviour, and analyzing its properties. Finally, one can use the results of this procedure to design a new experiment and go back to the measurement step.

Part 1: Static Linear Systems

In a static system, the output depends on the input only and does not depend on time. The simplest linear input-output static system is the system $y = a \cdot u$ where a is a constant. If we wish to extend this system to multiple inputs and outputs, we can use vector notation and write $Y = A \cdot U$ where Y and U are the output and input vectors and A is the transfer matrix. We restrict our description to homogeneous systems, i.e., those characterized by zero in the input producing zero in the output. However, it is easy to move to the general case by introducing a new input that is constant and then, for one dimension, the relation would be $y = a_0 + a_1 \cdot u$.

A basic element in many neural network models and in artificial neural networks is such a linear relation as illustrated in Fig. 3. The inputs u can model the activity of neurons that influence the modeled neuron; the constants w_i represent the synaptic strength or position; and the output can represent the neuron potential or firing rate.

In the case of multiple outputs, that is, multiple neurons, we can construct an artificial neural network (ANN), as illustrated in Fig. 4. The relation between the output and the input is $y_j = \sum_i w_{ij} u_i$. For m outputs and n inputs one can write the input-output relation as $Y = W^T \cdot U$ using the following matrixes notation:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \quad W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{am} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix} \quad W^T = \begin{bmatrix} w_{11} & w_{21} & \cdots & w_{n1} \\ w_{12} & w_{22} & \cdots & w_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1m} & w_{am} & \cdots & w_{mn} \end{bmatrix}$$

Each single weight represents the relation or association between a specific pair of input and output. Therefore this network is sometimes called "associative network". By implementing the rule of Hebb (Hebb 1949), i.e., adding strength to connections between neurons that act simultaneously, one can construct a basic model for associative memory.

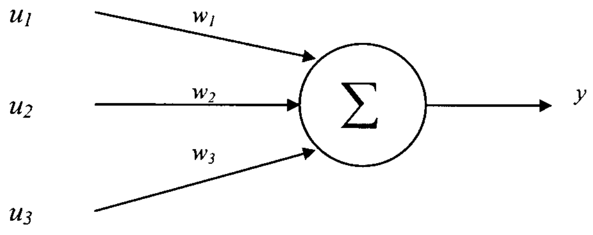


Fig. 3. Static linear system: y= $u_1 w_1 + u_2 w_2 + u_3 w_3$

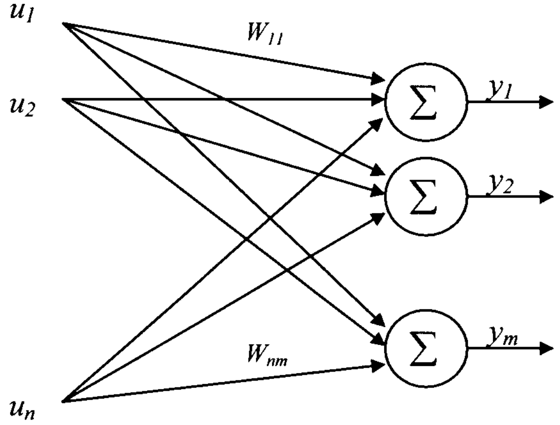


Fig. 4. An associative linear artificial neural network

The technical definition of memory is a device that can store and recall information, the input is called an address and the output is the data. Associative memory can recall the data in face of an inaccurate address, as long as the corrupted address is close enough to the true address. This property is very similar to the human brain memory. The next example describes a simple implementation of associative memory with the linear neural network.

Example 1: Linear Neural Network as an Associative Memory

This example demonstrates the use of a linear neural network as an associative memory. The input can be referred to as the address and the output as the data. For example we can think of the problem of face recognition, where the input would represent the face (maybe a vector of bitmap from a camera, or better a vector of features) and the output would be an identity number of the person. The structure of the network described in Figure 4 suggests that $y_j = \sum_i w_{ij} \cdot u_i$. Let us choose the domain of the inputs and outputs to be the binary range {-1,+1} and denote the items that we wish to insert into the memory with superscript *l*=1,2…,*L* where *L* is the number of items (these items are sometimes referred to as the "learning examples"). According to the rule of Hebb, the weights represent the correlation between two neurons, and in this case the correlation between the output and the input. This can be done mathematically by the equation $w_{ij} = \varepsilon \cdot \sum_{l=1}^{L} x_i^l \cdot y_i^l$ where the constant ε is chosen to be *1/n*.

Let us look at a simple numerical example of two memory items.

$$u^1 = [-1,1,1,-1] \qquad y^1 = [1]$$
$$u^2 = [1,1,1,1] \qquad y^2 = [-1]$$

In this case the indexes are in the range $i \in \{1,2,3,4\}$ $j = 1$ $l \in \{1,2\}$ and the weights matrix will be:

$$W = \left\{ \varepsilon \cdot \sum_{i=1}^{L} u_k^1 y_j^1 \right\} = \frac{1}{4} \begin{bmatrix} -1\cdot 1 + 1\cdot -1 \\ 1\cdot 1 + 1\cdot -1 \\ 1\cdot 1 + 1\cdot -1 \\ -1\cdot 1 + 1\cdot -1 \end{bmatrix} = \begin{bmatrix} -1/2 \\ 0 \\ 0 \\ -1/2 \end{bmatrix}$$

One can verify that the output of each memory item is correct. The next question is related to the generalization capability, that is, what will be the result of a new input vector that was not learned?

Associative networks should produce the nearest stored item. Let us check it for the following vector: $u = [-1,-1,-1,-1]$. For this vector, the output is $y = \sum_i w_i \cdot u_i = 1/2 + 1/2 = 1$, which is the result of the first item above, and one can see that the new item is closer to the first item (only two bits were inverse, compared with four bits in the second item).

This associative memory has many drawbacks, the stored data should be binary orthogonal vectors, there are a lot of connections and the capacity is low. There are other nonlinear associative memories, but they lack the simplicity and the mathematical tractability of the linear model and they are beyond the scope of this chapter. For more information about this and other ANN architectures see Chapter 25 and Fausett (1994)

The architecture described above and illustrated in Fig. 4 is the most general static linear neural network, since adding more layers of neutrons will not change the capability of this architecture. Note that this is not the case for nonlinear neural networks, where adding layers can enhance the capabilities of the architecture.

Part 2: Dynamic Linear Systems

Most of this chapter will deal with dynamic linear systems. In this section we will explain the notion of dynamic linear systems and the common methods to describe them. In the following sections specific procedures and examples will be described.

In dynamic linear systems, the inputs and outputs are functions of time, i.e., trajectories. Five common methods to describe such a system are described below:

- *Electronic or mechanical diagram*. One common way to describe a system is through graphic description of its physical elements and their connection. This method provides comprehensive description of the system's structure and an easy way to get qualitative understanding of the system by an expert. However, it is not always easy to predict the exact behavior of the system by examining the graphic diagram; therefore it is usually numerically simulated or transformed to a set of equations that can be mathematically analyzed and manipulated.
- *Differential or difference equation*. A system can be described by a set of differential equations relating the input and the output (or difference equation in the discrete case), the output will be the solution of the equation. A simple way to represent a linear dynamic system as a differential equation is the following standard differential equation: $y = \sum_i w_i \cdot u_i = y(t) = w_1 u(t) + w_2 \dot{u}(t) + \cdots + w_{N+1} \dot{y}(t) + w_{N+2} \ddot{y}(t) \cdots$ where a dot over a variable represents the time derivative, i.e.,

  $$\dot{u}(t) = \frac{\partial u(t)}{\partial t}, \quad \ddot{u}(t) = \frac{\partial^2 u(t)}{\partial t^2}$$

  etc. or in the discrete time: $y(t) = w_1 u(t) + w_2 u(t-1) + \cdots + w_{N+1} y(t-1) + w_{N+2} y(t-2) \cdots$ where t is a natural number, i.e., $t \in N$.
- *State space description*. The notion of state of the system helps us in separating the dynamic part of the system. One introduces a set of new variables that represent the state of the system in a way that the output is a static function of the state and possibly the inputs. The behavior of the variables is dominated by its own differential equation. The state variables may have a physical interpretation, such as the potential of a capacitor. In the linear case these equations are linear and can be written with matrix notation as follows:

  $$\dot{x} = Ax + Bu$$
  $$y = Cx + Du$$

  where x is the state, u is the input, and y is the output.
- *Impulse response*. As noted above, the basic property of linear systems is the superposition property; that is, the response of a linear system to the sum of two inputs is equal to the sum of the system's responses to each input. Thus, if we had a simple input, such that any other input could be produced as a linear sum of that input, and if we knew the system's response to that simple input, we could calculate its response to any other input. Such an input function is the impulse (also known as the delta function, $\delta(t)$), and the system's response to the impulse is called "impulse response". So if we know the impulse response, we practically know everything about the system. That is one of the major beauties of linear systems. Let us first describe the impulse function and then see how to calculate any response when the impulse response is given. Strictly speaking, an impulse is an abstract mathematical concept. To understand this concept, imagine a rectangular pulse lasting from $t=0$ to $t=\Delta t$ (duration Δt) and amplitude $1/\Delta t$, so that its area equals 1 (unit pulse). Now let Δt approach zero. In the limit, the pulse will be of infinitely short duration and infinitely large amplitude and is called unit impulse (because of its unit area) or delta function, $\delta(t)$, which

is a singular function. Impulses of other areas are obtained by appropriate multiplication with a factor. Unless specified otherwise, impulses are assumed to occur at zero time. For example, $\delta(t-t_o)$ is an impulse that occur at $t=t_o$.

The most important property of the impulse is that, for any regular function that is continuous at $t=0$, $\int_{-\infty}^{\infty} \delta(t)\cdot\phi(t)=\phi(0)$. That is, the impulse "highlights" the function at $t=0$. This function can therefore be thought of as being composed of sequences of such highlights generated by integrals whose integrands are products of and successive, infinitely closely spaced delta functions. Any linear system's response *y(t)* to an input *u(t)* therefore is the superposition of the system's impulse response *h(t)* to all the successive function highlights. This superposition is called a *convolution*. The convolution of *h* and *u* is defined as $y(t)=h(t)*u(t)=\int_{-\infty}^{\infty} h(t-\tau)\cdot u(\tau)\cdot d\tau$.

In the discrete case, the delta function is much simpler, its value is one at *t=0* and zero otherwise. The convolution in the discrete case is defined as $y(t)=h(t)*u(t)=\sum_{m=-\infty}^{\infty} h(t-m)\cdot u(m)$ where *t* and *m* are natural numbers. More information about the delta function and the convolution integral can be found in most of the advanced linear systems textbooks, such as Kwakernaak and Sivan (1991) and Lathi (1974), which also contains a graphic view of the convolution integral.

Comments: (i) Many other functions can be used as inputs to a linear system and produce all the information about the system, actually any function that contains all the frequencies. In many cases the step function and the step response are used, and sometimes a random noise signal is used. (ii) The impulse response is a very useful mathematical tool. However, in experiments of the biological system, it is usually not recommended to try to introduce an impulse. In fact, it is not possible to introduce a pure impulse but even an approximated high-energy impulse can cause damage to the system. The biological system is seldom linear in all the frequencies and an impulse can activate nonlinear modes of the system, therefore it is recommended to test and model the system only in its linear regions.

– *Transfer function in the Laplace or Z transform domain*. Given the impulse response and an arbitrary input, one can calculate the output as mentioned in the previous method, but the calculation involves evaluating the convolution integral, which may be a hard task. The main idea of the transforms is to move to another space where the convolution becomes a simple multiplication. The impulse response is transformed to a function that is called the transfer function, and the output in the Laplace domain is the Laplace domain input multiplied by the transfer function. In this way one can combine subsystems to a complex system in a block diagram, as will be described later in this chapter.

In the next parts we further describe, explain and demonstrate how to use these mathematical description tools.

Coherence

Before we start to build models and fit them to our data, we need to validate our assumption that we do have a linear system. A practical method to check the linearity of an unknown system is by calculating the coherence function between the input and the output signals. The coherence function, Γ, is defined as follows:

$$\Gamma(z)=\begin{cases}\dfrac{S_{xy}\cdot S_{yx}}{S_{xx}\cdot S_{yy}} & S_{xx}\cdot S_{yy}\neq 0\\ 0 & S_{xx}\cdot S_{yy}=0\end{cases}$$

where S_{uv} is the cross spectrum of the signals *u* and *v*. Most of the mathematical software, such as MATLAB, has the toolboxes and command to calculate the coherence function. (See Chapter 18 for more information about the coherence function.) For an LTI system without noise, the value of the coherence function is one. Therefore, if we find small values of the coherence function, we cannot be sure whether the system is not linear or very noisy. In both of these cases there is no point in estimating parameters for an LTI model. If the coherence function is close to one at the frequencies of interest, one can go on to estimate the parameters, build an LTI model and expect small errors. See Cadzo and Solomon (1987) for a thorough description of linear modeling and the coherence function, and Inbar (1996) for an example of typical coherence values in EMG measurements to estimate mechanical transfer function.

Note: One should notice the estimation procedure in order to get an accurate value of the coherence function, see Benignus (1969) for an estimation procedure.

Part 3: Physical Components of Linear Systems

Dynamic linear systems can be described schematically with simple basic physical components, electrical or mechanical. This method is natural for the description and design of physical systems, and therefore was widely used and developed by mechanical and electrical engineers. The advantage of this description is in the graphic description that is more comprehensible then differential equations due to the correspondence between the graphic elements and the modeled system. There are many simulation programs, such as SPICE, which provide graphic description as well as numerical simulation of such models (see Conant 1993, and Nilsson and Riedel 1996). In this section we will introduce the basic elements of electrical and mechanical systems, and the procedure to get the differential equation from the graphic description, which is based on the well-known laws of Newton and Kirchhoff. For more information about circuit theory and about modeling dynamic systems, see for example Charles and Kuh (1969) and Dorny (1993).

All the examples are from the field of neuromuscular system modeling.

Electrical Models

The basic elements of electrical models are resistor (R), capacitor (C), inductor (L) and sources of potential (V) or current (I). In some cases it is convenient to use conductance (g) instead of resistor, but the relation is simple, they are just the inverse of one another, i.e., g=1/R. From a graphic scheme of the electrical model one can extract the flow of current and the potential at each place in the model. The following Procedure 3.1 and Fig. 5 describe the methods and steps required in order to extract the differential equations out of the schematic description.

Procedure 3.1: Writing the Differential Equation of a Linear Electrical Circuit

1. Using an arrow, mark the current flow direction in each branch of the model.
2. For each node, write the Kirchhoff current law, stating that the sum of incoming currents equals zero: $\sum I = 0$
3. Replace each current by its value according to Fig. 5.
4. Solve or simplify the set of equations.

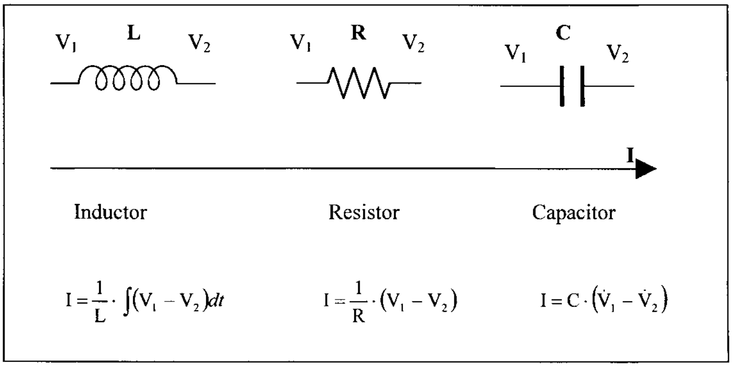


Fig. 5. The basic components of linear electrical circuits.

Example 3.1: Linear Neuron Model

The simplest dynamic linear model of a nerve cell is depicted in Fig. 6. The dendritic tree is represented by the resistors R_1, R_2 and R_3, which transmit the currents generated by the input voltages V_1, V_2 and V_3 to the cell body. The potentials V_1, V_2, and V_3 are due to synapses from other neurons. The currents generated in the dendritic tree are integrated by the capacitor C, representing the cell body membrane capacity. This integrator is "leaky", as represented by the membrane resistance R_4, this model therefore being referred to as "leaky integrator". Let us write the differential equation of this model according to Procedure 3.1.

1. Mark all the currents with an arrow that is going out of the point V_c
2. Due to the Kirchhoff current law we can write:

$$I_{R_1} + I_{R_2} + I_{R_3} + I_{R_4} + I_C = 0$$

3. Now replace the currents by their values according to Fig. 5:

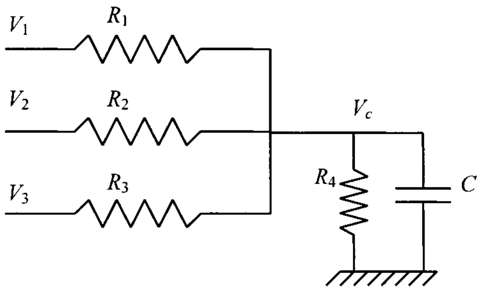
$$\frac{V_c - V_1}{R_1} + \frac{V_c - V_2}{R_2} + \frac{V_c - V_3}{R_3} + \frac{V_c}{R_4} + C\frac{dV_c}{dt} = 0$$

4. Finally, some simplification can be made to get a standard first-order differential equation:

$$\left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4}\right)V_c - \left(\frac{V_1}{R_1} + \frac{V_2}{R_2} + \frac{V_3}{R_3}\right) + C\frac{dV_c}{dt} = 0$$

This first-order equation can be solved analytically or numerically, and can also be transformed to the Laplace domain for further systems modeling and integrating, as will be described in the next sections. At this stage, let us consider some more examples for the execution of Procedure 3.1.

Fig. 6. A leaky integrator model for the nerve



Example 3.2: Linear Membrane Model

The membrane of nerves and muscles, when the potential is near the resting potential, can be modeled as the linear electric circuit in Fig. 7. C_m is the membrane capacity, and each branch represents a current of a single ion type. The potential sources V_K, V_{Na}, and V_{Cl} represent the Nernst potentials of potassium, sodium and chloride, respectively. The resistors R_K, R_{Na}, and R_{Cl} represent the resistance of the membrane to currents of potassium, sodium and chloride, respectively. The resistance is the macroscopic manifestation of the microscopic state of channels within the cell membrane. Note the arrows beside each voltage variable (V_m, V_{Cl}, V_{Na}, V_K). Each voltage variable denotes the potential difference between the arrow's head and its tail. For example, the membrane potential is defined as $V_m = V_{in} - V_{out}$. The direction of the potential sources (the longer line is positive) represents the typical value of the ion potential, as chloride and potassium have negative Nernst potentials and sodium has a positive Nernst potential. In writing the equations of such a model we regard only the arrow's direction; for a biologically plausible model the given data or the results of the calculations are expected to be compatible with the sources' directions as represented by the long and short lines.

Let us write the differential equation of this model according to Procedure 3.1:

$$I_c + I_{Na} + I_K + I_{CL} = 0$$

$$C_m \cdot \frac{\partial V_m}{\partial t} + \frac{(V_m - V_{Na})}{R_{Na}} + \frac{(V_m - V_K)}{R_K} + \frac{(V_m - V_{Cl})}{R_{Cl}} = 0$$

Let us check what happens in the steady state, i.e., when V_m does not change. In this case the time derivative of V_m is equal to zero and one can write the value of the membrane potential as follows:

$$V_m = \left(\frac{V_{Na}}{R_{Na}} + \frac{V_K}{R_K} + \frac{V_{Cl}}{R_{Cl}}\right) \cdot \left(\frac{1}{R_{Na}} + \frac{1}{R_K} + \frac{1}{R_{Cl}}\right)^{-1}$$

Notice that this is a weighted average of the Nernst potentials of the ions, where the weights depend on the membrane resistivity.

This result is similar to the Goldman equation (see Plonsey and Barr, Capter 3):

$$V_m = +\frac{K \cdot T}{q} \cdot \ln\left(\frac{P_K \cdot [K]_e + P_{Na} \cdot [Na]_e + P_{Cl} \cdot [CL]_i}{P_K \cdot [K]_i + P_{Na} \cdot [Na]_i + P_{Cl} \cdot [CL]_e}\right)$$

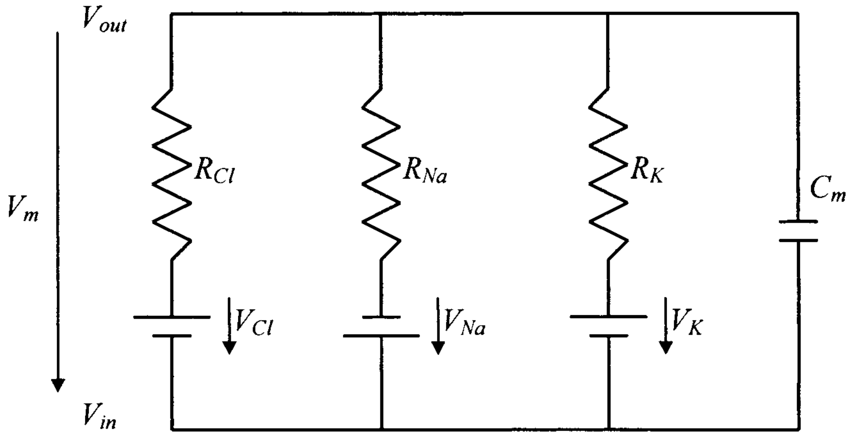


Fig. 7. The subthreshold membrane model

where P_k, P_{Na}, P_{Cl} are the permeability of potassium, sodium and chloride, respectively, [X] stands for the concentration of ion *X* in the fluid, index *i* stands for internal fluid and *e* stands for external fluid. KT/q is a constant equal to 26mV at room temperature.

In the extreme case, such as when the membrane is permeable to one ion only, the results are the same, i.e., the membrane potential equals the ions' Nernst potential. However, these two equations are not identical since their development is based on different sets of assumptions.

Example 3.3: Model of the Postsynaptic Membrane

Many chemical synapses in the nervous system can be described as follows:
- The presynaptic neuron releases neurotransmitters that operate on special sites in the postsynaptic membrane that open channels to specific ions. These ions flow according to the electrodiffusion forces and change the postsynaptic potential.
- The following electrical model (Fig. 8) describes the changes in the postsynaptic potential as a result of the change in the number of opened channels. The potential V_s represents the Nernst potential of the ion, and Δg_srepresents the change in the membrane admittance as a result of one channel that has been opened.

Let us write the equation of this model according to Procedure 3.1. Note that this is a static model (without capacitors or inductors), and remember that the admittance is the inverse of the resistance.

The number of open channels, that is, the number of close switches in Fig. 8, is denoted by *n*, each close switch adding one more branch to the circuit. Therefore, with I_{g_0} flowing through g_o and I_g flowing through Δg_s:

$$I_{g_0} + \sum_{i=1}^{n} I_g = 0$$

$$g_0 \cdot V_{Post} + n \cdot \Delta g_s \cdot (V_{Post} - V_s) = 0$$

$$V_{post} = \frac{n \cdot \Delta g_s \cdot V_s}{g_0 + n \cdot \Delta g_s}$$

This is a linear model with respect to its currents and potential; however, notice that the relation between the number *n* of channels and the postsynaptic voltage is nonlinear.

Note: One can add an exponential relation between the presynaptic potential and the number of channels opened. The result is that the relation between post- and presynaptic potential is a sigmoidal function which is very common in ANN models.

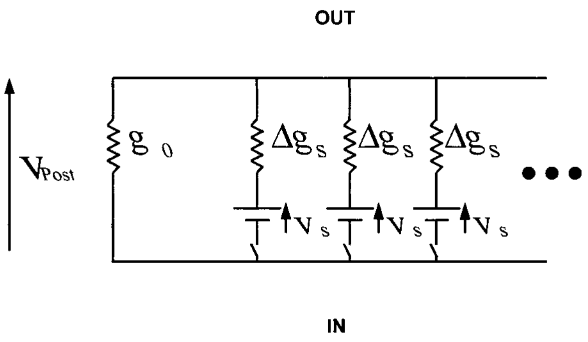


Fig. 8. A model of the postsynaptic membrane

Example 3.4: Cable Model of the Passive Nerve Fiber

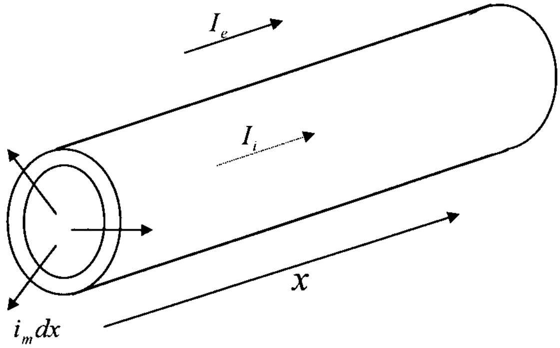
This example introduces a widely used model for propagation of potentials within axons and dendrites. This model is valid for the same regime as in Example 3.2, i.e., in the subthreshold region where the membrane can be viewed as a linear system.

In this model fibers are idealized as having a cylindrical geometry as described in Fig. 9 (see Plonsey and Barr 1988, Chapter 6, for a thorough description).

Let us look at the continuous fiber, as if it were constructed from small segments. With slight abuse of the term dx, we can refer to the length of each segment as dx, and then when dx approaches zero, it becomes the integral and differential operator. The electrical model is described in Fig. 10, with the following definitions:

- $r_i \cdot dx$ is the resistance of the internal liquid of the fiber segment to axial current;
- $r_e \cdot dx$ is the resistance of the extracellular liquid of the segment;
- r_m / dx is the resistance of the segment to current through the membrane;
- $C_m dx$ is the capacity of the membrane segment;
- $i_m dx$ is the current through the membrane in one segment of the fiber;
- $i_p dx$ is the current due to an external electrode in one segment of the fiber; it is positive for current entering the extracellular space via polarising electrodes.
- I_i is the axial current in the fiber,
- I_e is the current outside the fiber.

Fig. 9. The nerve fiber as a cylindrical fiber and the currents related to it.



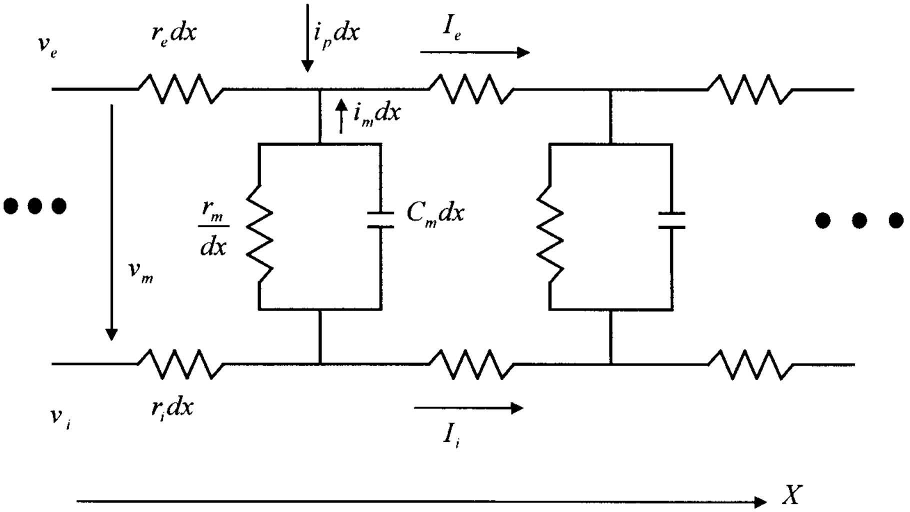


Fig. 10. The electrical linear model of the passive nerve fiber.

Now we can use Procedure 3.1 to write the current law and find out the differential equation that governs the membrane potential in the passive fiber.

From Kirchhoff's current law, we can write the following three equations:

$$i_p dx + i_m dx - dI_e = 0$$
$$i_m dx + dI = 0_i$$
$$i_m = \frac{V_m}{r_m} + C_m \frac{dV_m}{dt}$$

From Ohm's law (or from the definition of the resistive element) we have:

$$\frac{dV_e}{dx} = -I_e r_e$$
$$\frac{dV_i}{dx} = -I_i r_i$$

Let us recall the definition of the membrane potential, $V_m \equiv V_i - V_e$, apply differentiation with respect to x to both sides, and use the above relations.

$$\frac{dV_m}{dx} = \frac{dV_i}{dx} - \frac{dV_e}{dx} = -I_i r_i + I_e r_e$$

With a second differentiation and the three current law equations above we can write the following differential equation for the membrane potential:

$$\frac{d^2V_m}{dx^2} = -\frac{dI_i}{dx} r_i + \frac{dI_e}{dx} r_e = i_m r_i + \left(i_m + i_p\right) r_e = r_e i_p + (r_i + r_e) C_m \frac{dV_m}{dt} + (r_i + r_e) \frac{V_m}{r_m}$$

Let us introduce some useful notations, which make that equation more compact:

$$\tau \equiv r_m \cdot c_m \qquad \lambda^2 \equiv \frac{r_m}{r_i + r_e} \qquad D \equiv \frac{\lambda^2}{\tau} \qquad q(x,t) \equiv -D \cdot r_e \cdot i_p$$

$$D \cdot \frac{d^2 v_m}{dx^2} - \frac{dv_m}{\mathrm{dt}} - \frac{v_m}{\tau} = -q(x,t)$$

The last equation is known as the cable differential equation, which can be solved analytically or numerically.

If we introduce an impulse in *q(x,t)*, with some mathematical manipulation, we get the following impulse response:

$$V_h(x,t) = \frac{1}{2\sqrt{\pi \cdot D \cdot t}} \cdot e^{-\frac{x^2}{4 \cdot D \cdot t}} \cdot e^{-\frac{t}{\tau}}$$

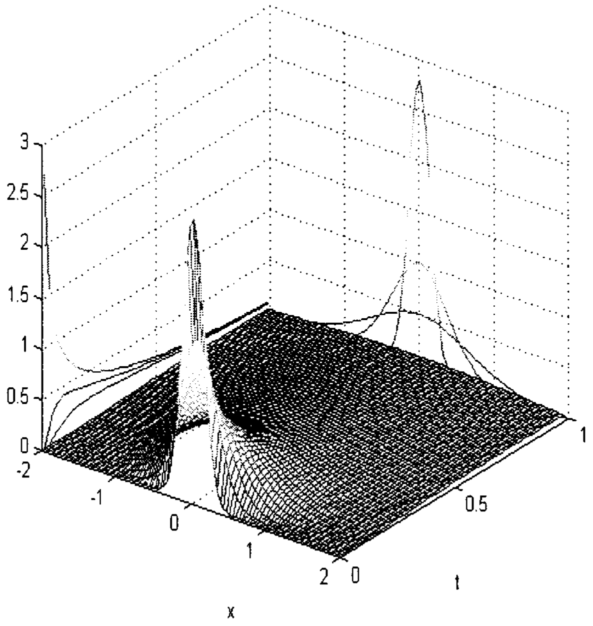
See Fig. 11 for an illustration of this impulse response function.

With the impulse response, one can calculate the response of the system to any given input, $q(x,t)$, by the convolution operator, which in this two-dimensional case is the following integral:

$$V_m(x,t) = (V_h ** q)(x,t) = \int_{\eta=0}^{t} \int_{\xi=-\infty}^{\infty} \frac{q(\xi,\eta)}{2\sqrt{\pi \cdot D \cdot (t-\eta)}} \cdot e^{-\frac{(x-\xi)^2}{4 \cdot D \cdot (t-\eta)}} \cdot e^{-\frac{(t-\eta)}{\tau}} \cdot d\eta \cdot d\xi$$

This integral can be solved analytically for some simple cases and numerically for practically any input function.

Fig. 11. Two-dimensional impulse response of the passive fiber. X is in units of $[1/\sqrt{D}]$ and t is in units of $[1/\tau]$. The sections on the sides of the picture are at times 0.01, 0.07 and 0.21$[1/\tau]$, and in distances 0.03, 0.5 and 0.8 $[1/\sqrt{D}]$.



Results

The passive fiber model above was used extensively to check the conduction velocity of nervous fibers with and without myelin, to calculate the optimal distance between the Ranvier nodes, to analyze the propagation of postsynapses potential in the dentritic tree and the propagation of action potentials in the axon. For further examples see Plonsey and Barr (1988) and Stein (1980).

Mechanical Models

There is great interest in modeling muscle and joint dynamics. There are two main reasons for this. One is that muscle is the main motor output of the nervous system and therefore an important window into the operation of the nervous system. Another reason lies in building prostheses and artificial limbs and in external excitation of muscles in paralyzed patients, which is called "functional neuromuscular stimulation" (FNS) (see, for example, Allin and Inbar 1986). All these fields require the construction of a model for the system. Below are two examples for muscle and joint modeling with mechanical elements.

The basic elements of mechanical models are:
- Spring (K), also known as elastic element;
- Damper (B), also called friction element;
- Mass (M);
- Force or tension generator (F, P or T).

The position (X) can be fixed to one location or be free to change according to the forces acting on it. From a graphic scheme one can extract the position velocities and forces at each place in the model. The following Procedure 3.2 and Fig. 12 describe the methods and steps required in order to extract the differential equations from the schematic description.

Procedure 3.2: Writing the Differential Equation of a Linear Mechanical System

1. Using an arrow, mark the force direction at each branch.
2. For each node, apply Newton's second law, stating that the forces applied to a mass equal the acceleration multiplied by the mass: $\sum F = M \cdot \ddot{x}$
3. Replace each force with its value according to Fig. 12.
4. Solve or simplify the equations.

Example 3.5: Second-order Mechanical Muscle Model

Figure 13 depicts a linear lumped model, approximating muscle behavior for a small signal (see McRuer et al. 1968). In this model,
- P represents the internal force in the muscle that is the result of the neural excitation;
- K and B are the elastic and viscous elements that represent the passive mechanical properties of the muscle tissue;
- M is the mass of the muscles and the joint.

According to Procedure 3.2:
1. Mark all the force directions to the right.
2. Of interest in this model is the position of the mass, so write Newton's law for that point: $F_p + F_K + F_B = M \cdot \ddot{x}$
3. Replace the forces according to Fig. 12: $-P + K \cdot (0 - x) + B \cdot (0 - \dot{x}) = M \cdot \ddot{x}$
4. The simplification stage is trivial here and leads to the following second-order equation: $P + K \cdot x + B \cdot \dot{x} + M \cdot \ddot{x} = 0$

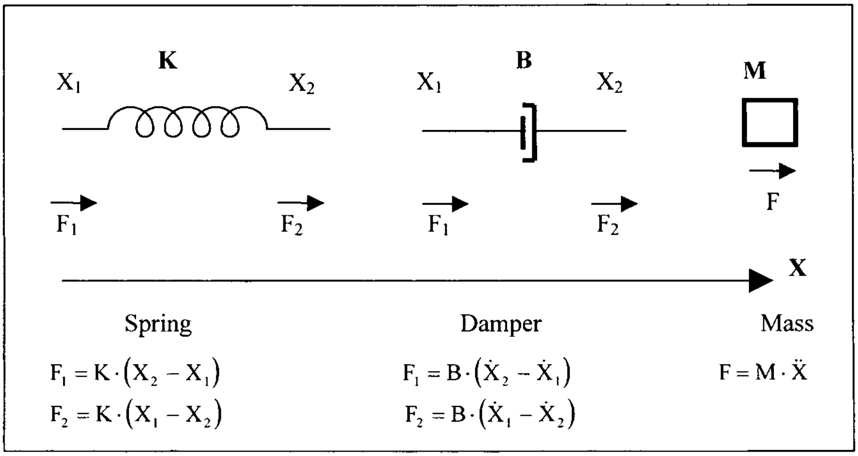
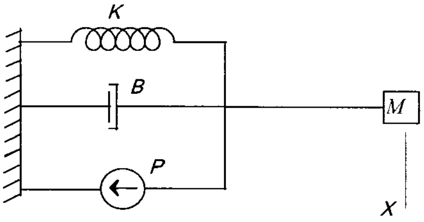


Fig. 12. The basic components of a linear mechanical system.

Fig. 13. Second-order mechanical model of muscles and joint



Example 3.6: A More Complex Mechanical Muscle Model

This is a slightly more complicated mechanical model of the muscle. This model is a linearized version of the Hill model, which will be discussed later in this chapter. The model represents one muscle, so it has to be combined with other muscles that act on a specific joint and with the joint mass in order to get a complete model for specific movements. In this model (see Fig. 14),

- P represents the internal force in the muscle resulting from neural excitation;
- B is the viscous element that represents the relation between force and velocity in the muscle;
- K_s is a serial elastic element that represents the mechanical property of the tendon;
- K_p and B_p represent the mechanical properties of other tissues around the muscle and the joint;
- F is the force between the joint and the muscle.

Let us write the equations according to Procedure 3.2.

There are two points of interest, X and $X_{1,}$ the latter being the connection point of B, P and K_s. Both points are not associated with a mass, leading to the following two equations:

$$F - B_p\dot{x} - k_p x + k_s(x_1 - x) = 0$$
$$-P - B\dot{x}_1 + k_s(x - x_1) = 0$$

Extracting x_1 from the first equation and inserting it into the second leads to

$$x_1 = \frac{-F + B_p\dot{x} + k_p x}{k_s} + x$$
$$-P - B\left(\frac{-\dot{F} + B_p\ddot{x} + k_p\dot{x}}{k_s} + \dot{x}\right) + k_s\left(\frac{F - B_p\dot{x} - k_p x}{k_s}\right) = 0$$
$$-Pk_s - B\left(-\dot{F} + B_p\ddot{x} + k_p\dot{x} + k_s\dot{x}\right) + k_s\left(F - B_p\dot{x} - k_p x\right) = 0$$

This final equation is a third-order differential equation that can be solved numerically or transformed to the Laplace domain for further manipulation or incorporation in a larger model as will be done in the next section.

The reader has surely noticed the similarity between Procedures 3.1 and 3.2. One can transform a mechanical model to an electrical model and vice versa according to Table 1. This transform can be useful if one is an expert in one kind of scheme, or if one has good simulation software for a specific kind of modeling scheme.

This equivalence underscores the major advantage of linear systems modeling. Linear modeling of any kind of system, mechanical, electrical, hydraulic, chemical or other,

Fig. 14. A more complex linear mechanical muscle model

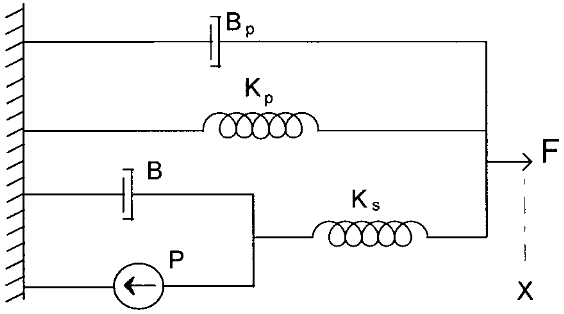


Table 1. The equivalence of electrical and mechanical components

Mechanical	F	$\dot{X}$	B	K	M
Electrical	I	V	1/R	1/L	C

is always reduced to a differential equation that has standard solutions and can be transformed to the Laplace domain and treated with the same tools. This contrasts with non-linear models that are usually unique to a specific system and therefore require a special theory to be built for each case.

Part 4: Laplace and Z Transform

Linear systems can be described and analyzed in the frequency domain. For linear signal analyses the Fourier transform is very popular, and for linear systems description the Laplace and Z transforms are used for continuous and discrete description, respectively. The main advantage of this description is in finding the transfer function of the system, which is the Laplace transform of the impulse response, as mentioned briefly in Part 2. In the Laplace domain, complex operations simplify, such that e.g. differentiation reduces to multiplication.

Let us begin with the definition of the transform and an introductory example.

The Laplace transform of the continuous signal *x(t)* and the Z transform of the discrete signal are the following:

$$X(s) = \int_{-\infty}^{\infty} x(t) \cdot e^{-s \cdot t} dt \qquad X(z) = \sum_{n=-\infty}^{\infty} x(n) \cdot z^{-n}$$

The domain of the variables *s* and *z* consists of all the complex numbers for which the integral (or sum) above converges. For example, let us look at a system that executes an integration of its input, that is, the system $y(t) = \int_{-\infty}^{t} u(t) dt$. The impulse response of this system is a step function and the Laplace transform of the impulse response is 1/*s*. Therefore, in the Laplace domain the relation between the output and the input is $Y(s) = U(s)/s$, which is a much simpler relation than the integration above. Figure 15 describes this idea graphically.

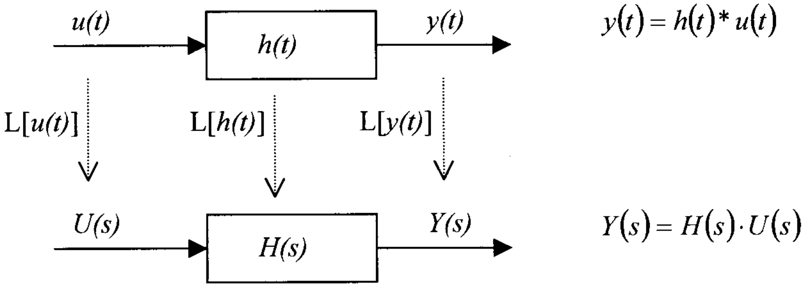


Fig. 15. Linear input-output system: In the time domain, top, *h(t)* is the impulse response and * denotes the convolution operation. In the Laplace domain, bottom, *H(s)* is the transfer function.

Procedure 4.1: Determining Laplace and Z Transforms

The calculation of the transform of a specific function requires some practice. However, for practical purposes, we can just look up a table of Laplace or Z transforms. A short table is given below (see Table 2) and detailed tables can be found in Kwakernaak and Sivan (1991). Another practical approach is to use numerical or symbolic software such as MATLAB that has built-in functions to these transforms.

The following subprotocols introduce some standard properties of these transforms that are used for systems description and analysis. The comprehensive theory and practice of these transforms is beyond the scope of this chapter, and the interested reader is referred to Kwakernaak and Sivan (1991).

Procedure 4.2: Transfer Function from Differential Equation

This simple procedure is based on the following property of the Laplace transform:

$$L\left\{\frac{df(t)}{dt}\right\} = s \cdot L\{f(t)\}$$

This property is correct when one assumes zero initial conditions, i.e., *f(0)=0*.

Therefore the transform of the following general differential equation will be:

$$y(t) = w_1 u(t) + w_2 \dot{u}(t) + \cdots + w_N u^{(N-1)}(t) + w_{N+1}\dot{y}(t) + w_{N+2}\ddot{y}(t) \cdots + w_{N+M} y^{(M)}(t)$$
$$Y(s) = w_1 U(s) + w_2 sU(s) + \cdots + w_N s^{N-1} U(s) + w_{N+1} sY(s) + w_{N+2} s^2 Y(s) + \cdots + w_{N+M} s^M Y(s)$$

Table 2. Laplace and Z transforms: Some useful functions and properties.

Continous Time $f(t)$	Laplace Transform $F(s)$	Discrete Time $f(n)$	Z Transform $F(z)$
$\delta(t)$	1	$\Delta(n)$	1
$f(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}$	$\frac{1}{s}$	$f(n) = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$	$\frac{z}{z-1}$
$e^{-a \cdot t}$	$\frac{1}{s+a}$	a^n	$\frac{z}{z-a}$
$e^{-a \cdot t}\sin(w \cdot t)$	$\frac{w}{(s+a)^2 + w^2}$	$a^n \sin(\Omega \cdot n)$	$\frac{a \cdot \sin(\Omega) \cdot z}{z^2 - 2a\cos(\Omega)z + a^2}$
$a \cdot f(t) + b \cdot g(t)$	$a \cdot F(s) + b \cdot G(s)$	$a \cdot f(n) + b \cdot g(n)$	$a \cdot F(z) + b \cdot G(z)$
$\frac{d}{dt} f(t)$	$s \cdot F(s) - f(0)$	$f(n+1)$	$z \cdot F(z)$
$f(t) * g(t)$	$F(s) \cdot G(s)$	$f(n) * g(n)$	$F(z) \cdot F(z)$

From the last expression one can directly extract the transfer function of the system:

$$\frac{Y(s)}{U(s)}=\frac{w_1+w_2s+\cdots+w_Ns^{N-1}}{1+w_{N+1}s+w_{N+2}s^2+\cdots+w_{N+M}s^M}$$

A similar procedure holds for the Z transform, that is, the difference equation

$$y(n)=\sum_{i=0}^{N}a_i\cdot x(n-i)-\sum_{j=1}^{M}b_j\cdot y(n-j)$$

will be transform to the following algebraic equation

$$Y(z)=\sum_{i=0}^{N}a_i\cdot z^{-i}\cdot X(z)-\sum_{j=1}^{M}b_j\cdot z^{-j}\cdot Y(z)$$

Comment: Rational Transfer Functions

The notion of transfer function is illustrated in Fig. 15. We just saw how to transform a differential equation to a transfer function where the numerator and denominator were polynomial functions. The transfer function can therefore be described as follows:

$$H(s)=\frac{OUT(s)}{IN(s)}=k\cdot\frac{\Pi(s-z_i)}{\Pi(s-p_j)}$$

where k is called the "gain", the z_i are called "zeros", and the p_i are called "poles".

This formalization is easy to analyze. There is a vast literature about the influence of the location of the poles and zeros on system behavior, and there are many names for all kinds of such systems. If there are only poles, the system is called "auto-recursive" (AR); if there are only zeros, the system is called "moving average" (MA); and the general case is called "auto-recursive moving average (ARMA) system". The case of zeros only is also called "finite impulse response" (FIR) because the influence of the impulse is gone after a short period, while adding poles produces an infinite impulse response (IIR).

Another useful property of the Laplace transform enables the calculation of the steady state of the system in the Laplace domain:

$$\lim_{t\to\infty}f(t)=\lim_{s\to 0}s\cdot F(s)$$

All these properties have their equivalent in the Z transform domain for discrete signals and systems.

Procedure 4.3: Discretization

Since we usually use a computer and discrete measurements, it is very useful to have a discrete model. Nevertheless, the physical and biological world is a continuous world. Discretization is the procedure of converting a continuous model to a discrete one. There are different procedures for discretization just as there are many ways of numerical integration. The simplest method, the Euler's forward method, is to move to the Z transform by replacing each s by $(z-1)/T$. This method is demonstrated in Example 4.2 (see Santina et al. 1996 for more details about discretization methods).

Procedure 4.4: Check for Stability

A *system* is stable if a bounded input produces a bounded output. In the Laplace domain, there is a simple procedure to check the stability of a system.

1. Write the transfer function as a rational function.
2. Find the roots of the denominator, that is, the poles.
3. If the real part of all the poles is negative, then the system is stable. Otherwise the system is not stable.

Example 4.1: Stability and Step Response

This example serves to practice the linear systems description and the transform tools. We will show how to find the conditions for a second-order system to be stable, find an expression for its response to a step input and find the time to reach the maximum in a response to a step input.

For this purpose, let us first highlight some pertinent properties of first- and second-order systems. A first-order system is most common in biological modeling. Most of the examples in this chapter are of a first-order system starting with the leaky integrator of Example 3.1. The transfer function of a typical first-order system is $1/(s+a)$, and from Table 2, one can see that the impulse response of such a system is an exponential decay. The step response of this system is illustrated on the left side of Fig. 16. This function is characterised by its time constant $\tau=1/a$, which is the time when the response has reached about 2/3 of its final value.

Second-order systems can also produce oscillatory behavior. The muscle and joint model in Example 3.5 is a second-order system. A description of the step response of a typical second-order system is illustrated on the right side of Fig. 16. In order to describe the parameters of such a system, let us write the transfer function of a second-order system in a standard form:

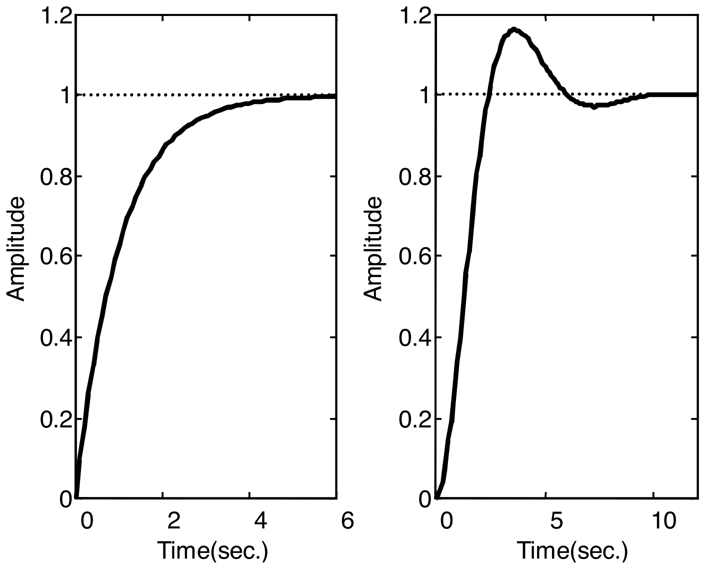


Fig. 16. Step response of a first-order system (left) and of a second-order system (right). The transfer functions of these systems are $1/(1+s)$ and $1/(s^2+s+1)$, respectively.

$$H(s) = \frac{w_n^2}{s^2 + 2 \cdot \xi \cdot w_n \cdot s + w_n^2}$$

This system has been analyzed extensively, and there are expressions for every possible feature of it. For example, the overshoot, which is the ratio of the response that is beyond the steady state response to the latter, is the following function of the parameter

$$\xi : \text{O.S.} = e^{-\frac{\pi \cdot \xi}{\sqrt{1-\xi^2}}}.$$

The frequency of the oscillations is

$$f_d = \frac{w_n}{2\pi} \cdot \sqrt{1-\xi^2},$$

the time constant is

$$\tau = \frac{1}{\xi \cdot w_n}$$

and the settling time, which is the time to reach a region of 2% of the final value and stay there, is

$$t_s \cong \frac{4}{\xi \cdot w_n}.$$

Let us now check the stability of the above second-order system according to Procedure 4.4. For that purpose, we have to find the location of the poles, that is, the roots of the denominator of the transfer function, which are:

$$s_{1,2} = -\xi \cdot w_n \pm j w_n \cdot \sqrt{1-\xi^2}$$

For stability, both poles must have a negative real part. Therefore the requirement for stability is: $\xi \cdot w_n > 0$. Notice that the second-order system has an oscillatory behavior when the poles have imaginary parts.

Let us now move to the calculation of the step response. We know that in the Laplace domain the output is the input multiplied by the transfer function. According to Table 2, the Laplace transform of the step function is 1/s, and the output will be the second-order systems transfer function divided by *s:*

$$Y(s) = \frac{w_n^2}{s \cdot \left(s^2 + 2 \cdot \xi \cdot w_n \cdot s + w_n^2\right)}$$

The inverse transform of this function can be calculated using Table 2 and some mathematical manipulations, by looking at an extended Laplace transforms table or by using mathematical software. The resulting function is the response to a step function, which is:

$$y(t) = 1 - \frac{e^{-\xi \cdot w_n \cdot t}}{\sqrt{1-\xi^2}} \cdot \sin\left[w_n \cdot t \cdot \sqrt{1-\xi^2} + \tan^{-1}\left(\frac{\sqrt{1-\xi^2}}{\xi} \right) \right]$$

The maximum is the first place where the tangent is flat, that is,

$$\frac{dy(t)}{dt} = 0.$$

The time of this event is:

$$t_p = \frac{\pi}{w_n \cdot \sqrt{1-\xi^2}}$$

Example 4.2: Second-order Mechanical Muscle Model

The second-order mechanical muscle model was introduced in Example 3.5 (see Fig. 13). With Procedure 4.1 we can transform the differential equation to the following transfer function:

$$\frac{X(s)}{P(s)} = \frac{-1}{M \cdot s^2 + B \cdot s + K}$$

One can derive a similar relation for external force and its relation to the position, or any other desired relationship. With Procedure 4.3, we can transform the above function to the Z transform domain:

$$\frac{X(z)}{P(z)} = \left.\frac{-1}{M \cdot s^2 + B \cdot s + K}\right|_{s=\frac{z-1}{T}} =$$

$$\frac{-T^2}{M \cdot z^2 + (B \cdot T - 2 \cdot M) \cdot z + M - B \cdot T + K \cdot T^2}$$

From the Z transform we can move to discrete time by applying Procedure 4.1 inversely:

$$X(n) = \frac{-T^2}{M} \cdot P(n-2) - \frac{(B \cdot T - 2 \cdot M)}{M} \cdot X(n-1) - \frac{(M - B \cdot T + K \cdot T^2)}{M} \cdot X(n-2)$$

This difference equation may be useful for parameter estimation and handling of sampled data from this system, as will be demonstrated in Example 5.2. The last difference equation can be formalized as follows: $X(n) = w_1 \cdot P(n-2) + w_2 \cdot X(n-1) + w_3 \cdot X(n-2)$, where w_i are the parameters. If the sample time T is given, it is equivalent to know w_i or M,B,K. Therefore, for each time step, the system can be viewed as a simple static system with three inputs and one output, as described graphically in Fig. 3.

Example 4.3: Complex Mechanical Muscle Model

This example demonstrates how to extract the transfer function and the steady-state behavior of the Hill-type mechanical muscle model that was introduced in Example 3.6. From the graphic model in Fig. 14, we derived the following differential equation that can be further simplified:

$$-Pk_s - B\left(-\dot{F} + B_p\ddot{x} + k_p\dot{x} + k_s\dot{x}\right) + k_s\left(F - B_p\dot{x} - k_p x\right) = 0$$

From this equation we can move to the Laplace domain as described above in Procedure 4.1:

$$(k_s + Bs)F - Pk_s - \left(BB_p s^2 + Bk_p s + Bk_s s + k_s B_p s + k_s k_p\right)X = 0$$

Now we can extract any transfer function we need according to the input and output we define. In an isometric experiment when the length is held constant, we can subtract the constant force due to the constant length and find the transfer function between the force, F, and the neural excitation to the muscle, which is related to P. (In the linear model we assume a linear relation between the firing rate of the motor neuron and the hypothetical internal force P.)

$$\frac{F}{P} = \frac{k_s}{(k_s + Bs)}$$

This is a first-order system and one can investigate its behavior according to the description in Example 4.1 above. For example, the force response to a step in the input will look like the left graph in Fig. 16.

The same manipulation can be done in order to investigate an isotonic experiment, where the force is held constant.

The steady-state behavior of the model can be calculated by practically replacing each s with a zero, which brings us to the following relation:

$$\frac{F - P}{k_p} = x$$

One should notice that the viscous elements have no role in the steady-state behavior, since they produce forces only at times of change in position.

A good example for the application of these mathematical procedures to the physiological neuromuscular system is the study of eye movements. The time constant of the mechanical model of the system is of the order of seconds while we know that the eye can move from one position to the other in tenths of a second. Therefore, a simple step in neural excitation will not satisfy the observations of eye movement. An alternative hypothesis is that the neural excitation signal contains an initial pulse added to the step to accelerate the movement. This was found to be consistent both with the model and the measurements. This control strategy of pulse plus step neural excitation is also used in limb movement control models.

Comment: Notice that when a step neural excitation is mentioned, we relate to the firing rate and not to the nerve cell potential, that is, a unit step at time zero means that the cell starts to fire an action potential once a second from time zero and thereafter.

Part 5: System Identification and Parameter Estimation

In many disciplines of science and technology we frequently face data from an unknown system and our aim is to find a model of this system. A parametric model belongs to a family of models characterized by a finite number of parameters. The modeler's task is first to choose a proper model family and then to estimate the parameter values. In this section we will first describe the estimation problem in general and then concentrate on linear models.

The general problem of parameter estimation can be formalized as follows: Let $\Theta(u,a)$ be a family of parametric functions, that is, for each parameter vector a_0, $y = \Theta(u,a_0)$ is a static input/output function or a transfer function in the Laplace domain, where u is the input and y is the output. Suppose that we have an unknown system $F(u)$ that is assumed to belong to the above function family, that is, $F(u)=\Theta(u,a_0)$ for a specific but unknown parameter vector a_0. As a result of an experiment on this unknown system, we collect a group of measurements of input/output pairs $\{u_i,y_i\}$ that naturally satisfy $y_i=F(u_i)$. (In the presence of measurement noise or uncertainty in the generating function, that is, if we are not positive about the assumption that the unknown system belongs to the selected family of parametric functions, we can relax the requirements from the data to $|y_i-F(u_i)|<n$, where n represents the noise or the uncertainty in the fitness of the model to the system). The problem is to find the vector of parameters a that will best fit the measurement pairs according to a given criterion. If one uses the least-squares criterion, the problem is to solve the following minimization:

$$\hat{a} = \underset{a}{\arg\min} \sum_i \left(y_i - \Theta(x_i, a)\right)^2$$

There are many methods to solve this problem and to formalize parametric groups of functions (see for example Sjoberg et al. 1995). In this chapter we concentrate on the linear group of functions, implying that the function can be transformed to the Laplace domain, resulting in a transfer function. In the discrete case the same can be done with a difference equation and the Z transform.

Procedure 5.1: Estimation Scheme

The basic way to estimate the parameters of a linear model is the following:
The linear model is $y=\sum_i w_i \cdot u_i$ or in matrix notation $y=W^TU$. The real system may not be linear and the data we have may be noisy; however, the optimal linear model according to the least-squares criterion is the following:

$$W_{OPT} = \Phi^{-1} \cdot P$$

$$P \equiv E[Y \cdot U] \qquad \Phi \equiv E\left[U \cdot U^T\right]$$

where E stands for expectation. For the origin and proof, see any textbook on linear parameter estimation (e.g., Porat 1994).

In practice we estimate the expectation as a numerical average over the measurements, that is,

$$P = \frac{1}{N}\sum_{l=1}^{N} Y^l \cdot U^l \qquad \Phi = \frac{1}{N}\sum_{l=1}^{N} U^l \cdot U^{l^T}$$

Example 5.1: Two Inputs-One Output System

This is a simple synthetic numerical example to demonstrate the use of Procedure 5.1. Assume that we have a static system with two inputs and one output and we wish to find an optimal linear model for this system. The model will be $y = w_1 \cdot u_1 + w_2 \cdot u_2$, the input vector will be $U=[u_1,u_2]^T$ and the parameter vector will be $W=[w_1,w_2]^T$. In order to estimate the parameters of the model, an experiment was conducted and the following four measurements were obtained:

u_1	u_2	y
1	1	3.1
1	−1	1.2
−1	1	−0.8
−1	−1	−2.9

Let us calculate the estimation of P and Φ according to procedure 5.1.

$$P \equiv E[Y \cdot U] = E\begin{bmatrix} y \cdot u_1 \\ y \cdot u_2 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 3.1+1.2+0.8+2.9 \\ 3.1-1.2-0.8+2.9 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\Phi \equiv E\left[U \cdot U^T\right] = E\begin{bmatrix} u_1 \cdot u_1 & u_1 \cdot u_2 \\ u_2 \cdot u_1 & u_2 \cdot u_2 \end{bmatrix} = \cdots = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Now we can calculate the optimal parameters:

$$W_{OPT} = \Phi^{-1} \cdot P = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

and with these parameters we can calculate the model's output to the measurement data and check the fitness of the model to the data: One can see that the model outputs are

u	u_2	y_m
1	1	3
1	−1	1
−1	1	−1
−1	−1	−3

similar to the actual data.

Comment: In practice, the measurements may be random and there may be more noise, therefore more examples are needed in order to get a good estimation of the parameters.

Example 5.2: Estimation of Muscle Model

In Example 4.2 we found the following relation for the linear muscle model that was introduced in Example 3.5:

$$X(n) = w_1 \cdot P(n-2) + w_2 \cdot X(n-1) + w_3 \cdot X(n-2)$$

In this example we demonstrate the parameter estimation procedure for this dynamic model. We can combine our input components $X(n-2)$, $X(n-1)$, $P(n-2)$ to form an input vector U, and denote the output vector, which in this case has just one element, $X(n)$, by the letter Y. Now we can use the optimal solution of procedure 5.1.

Let us illustrate this estimation scheme by a simulation example. A random sequence of P was chosen (normally distributed noise with standard deviation (STD) equal to one, and zero mean). The length X was calculated according to the model with the following nominal value of the parameters: M=5, B=3, K=2, T=0.1, that is, W_1=−0.002, W_2=1.94, W_3=−0.944.

Figure17 shows the results of the simulation. The first graph is the random input P, the second is the calculated X. An additional random noise was added to simulate measurement noise or uncertainties in the model (normally distributed noise with STD=0.01 and zero mean); this sequence appears in the third graph. Then the optimal parameters were calculated according to Procedure 5.1, and the results were W_1= −0.0016, W_2= 1.744, W_3= −0.739, which is close to the nominal parameters, as expected. Finally, the output of the estimated model for each time step was calculated, and it appears in the fourth graph being similar to the second graph, which is the actual model output.

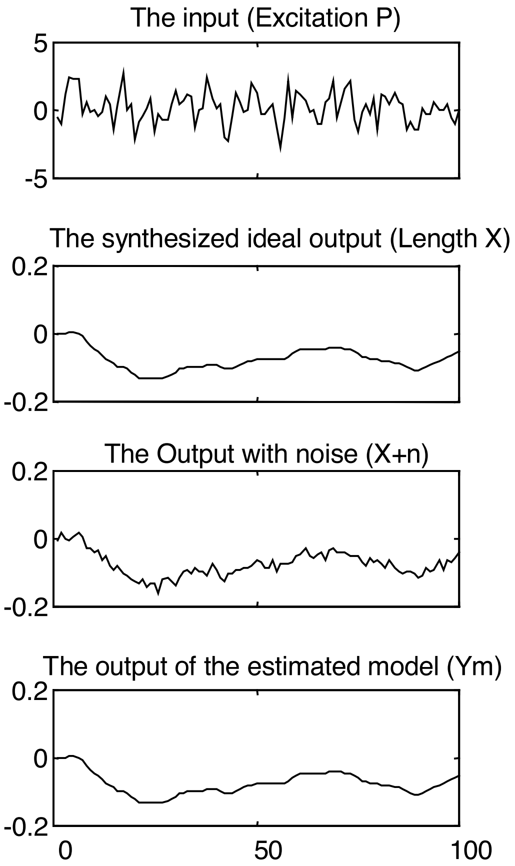


Fig. 17. An example of parameter estimation of the linear (ARMA) muscle model. The simulation is discrete, therefore the abscissa consists of time steps. The length of each time step in this simulation is 0.1 second.

Comments: The above example is a straightforward implementation of Procedure 5.1. There are various pitfalls that are listed below:

- This example regards the discrete data as a set of independent examples of a static model and the optimal model is checked for each couple of input-output independently. In practice, the error combines from one time step to the other, since the model may use its own output to estimate the next time step, and not the real system outputs. This problem can be severe when the system has some unstable poles, then the error might grow very fast.
- The calculated model, that is, the estimated parameters, should be checked on a new data set and not only on the data that was used for parameter estimation. This check is called "generalization check" and can assist in avoiding over-fitting of the data. We discuss this method of validation in the following subsection.
- One should remember that the biological system is generally a time-varying system. For example, muscles can change their properties due to fatigue. Therefore the duration of the experiment must be short in order to justify the assumption that the system is time-invariant.
- Finally, we must mention here that the simple optimal parameter calculation in Procedure 5.1 is not always stable numerically. There are many improvements and practical methods that can be found in modern numerical software, such as MATLAB (see Ljung 1986).

Problem: Choosing the Right Model Order

In the last example the structure of the model was known and the only problem was estimating the parameters, but in most biological cases the model is unknown.

In order to estimate the parameters, we first need to establish the order of the model, which means, for example in the ARMA case, choosing N and M in the following discrete model:

$$y(n)=\sum_{i=0}^{N}a_i\cdot x(n-i)-\sum_{j=1}^{M}b_j\cdot y(n-j)$$

At first glance, one might suppose that the more parameters the model has, the better it will fit the actual system, but this is not the case (see Paiss and Inbar 1987 for an extensive treatment of the model order selection problem for the case of surface electromyography). Too many parameters are not only a computational burden but they may cause errors in the model. One can be wrong by either choosing too many or too few parameters. See Fig. 25 for a description of the pitfalls in choosing the wrong number of parameters.

Many approaches have been suggested for choosing the proper order. For linear models a commonly used approach is the Akaike information criterion (AIC) which is based on a discrepancy measure. For the ARMA model it will take the following form:

$$N_T^{\;-1}\cdot AIC(N,M)=\hat{\sigma}^2+\frac{2\cdot(N+M+1)}{N_T}$$

where N and M are the model size, see the discrete ARMA model above, N_T is the total number of samples and $\hat{\sigma}^2$ is the estimation of the error.

Since the first term, the estimation of the error, $\hat{\sigma}^2$, monotonically decreases with increasing model size and the second term increases, one can find an optimal model size by finding the minimal value of the *AIC*. Another method to choose the order of the model is validation. This method is commonly used in pattern recognition and classification where part of the data is kept from the learning phase (in our case this will be the fitting phase), and then the model is chosen for its generalization capabilities checked on the kept data. For more information about parameter estimation and systems identification see Porat (1994), Sjoberg et al. (1995) and Ljung (1986).

Part 6: Modeling The Nervous System Control

We have seen that linear systems can be described in the Laplace domain by their transfer functions. These transfer functions make it easy to analyze complex systems that include many modules. This section describes how to use block diagrams in modeling the nervous system.

Procedure 6.1: Block Diagrams

There are only two basic elements in linear block diagrams: summer and transfer function block. The summation element is usually symbolized by a circle and a sign at each one of its inputs that determines whether is should be added or subtracted. The transfer function is represented as a block with the transfer function or an impulse response function in it. A special case of transfer function is a mere gain that is sometimes symbolized by a triangle. Each block is connected to other blocks by arrows that symbolize the direction of information flow.

The procedure of writing the transfer function between two points in a block diagram is as follows:

1. Name each arrow that has no name with a unique variable.
2. Write an equation for each variable. For example, if the variable is y and the input(s) to the block before it is u, then write the following:
   For an output of a summation element, write $y = \sum_i u_i$.
   For an output of a transfer function H, write $y=Hu$ (in the Laplace domain).
3. Simplify the set of equations in order to get a transfer function between the input and the output or any other relation needed.

Example 6.1: Feedback Control

Feedback control is based on using the outcome of the process or the controlled system, which is usually called the "Plant", in order to control it, in other words, using the error between the desired output, y_d, and the actual output, y, in order to reduce it. This scheme is widely used to describe the nervous system control of the musculoskeletal system.

The analogy of the feedback scheme, Fig. 18, to motor control is the following: The plant corresponds to the muscles, the bones and the dynamics of the environment; the feedback corresponds to the output of the sensory systems, and the controller corresponds to the nervous system. Let us follow Procedure 6.1 in order to find the transfer function of the complete system in Fig. 18. (The Laplace variable s is omitted for simplicity.)

1. Let us call the output of the feedback x_1, and the output of the sum x_2.
2. There are four elements and therefore four equations:

$$Y = P \cdot u \quad u = C \cdot x_2 \quad x_1 = F \cdot Y \quad x_2 = Y_d - x_1$$

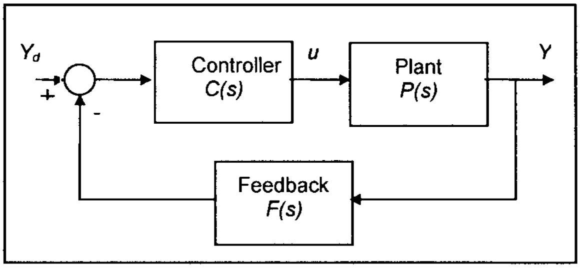
3. From the above equations one can extract the following transfer function:

$$\frac{Y}{Y_d} = \frac{P \cdot C}{P \cdot C \cdot F + 1}$$

One major advantage of the feedback control scheme is the reduced sensitivity to changes in the parameters of the plant, and to changes in the environment. The sensitivity of system H to changes in the parameter k is defined as follows:

$$S_H^k \equiv \left| \frac{\partial H}{\partial k} \cdot \frac{k}{H} \right|$$

Fig. 18. Feedback control



When the value of the sensitivity function is zero, the system is insensitive to changes in the parameter. Let us look at the system without feedback, where H is the transfer function, and k is a gain parameter. The system in an open loop is $H = k \cdot P$. The sensitivity of the system would be:

$$S_H^k \equiv \left| \frac{\partial H}{\partial k} \cdot \frac{k}{H} \right| = P \cdot \frac{k}{k \cdot P} = 1$$

However, in closed-loop mode, the system is $H = \dfrac{k \cdot P}{k \cdot P \cdot F + 1}$ and the sensitivity will be:

$$S_H^k \equiv \left| \frac{\partial H}{\partial k} \cdot \frac{k}{H} \right| = \frac{P \cdot (k \cdot P \cdot F + 1) - k \cdot (P)^2 \cdot F}{(k \cdot P \cdot F + 1)^2} \cdot \frac{k \cdot P \cdot F + 1}{P} = \frac{1}{k \cdot P \cdot F + 1} < 1$$

So when the loop gain, k, is high, the sensitivity to changes is low.

There is a vast literature on the stability of such systems and on methods to choose a controller when the specifications of the desired performances are given (see for example Kwakernaak and Sivan 1991 and section III in Levine 1996).

Comments: The first problem in using this simple feedback scheme to model biological systems occurs when one tries to measure the loop gain. In biological systems, one often finds very low loop gains in the order of one. Therefore reduced sensitivity to changes in the parameters frequently does not occur in biological systems. Another problem results from delays in biological systems that can cause instability and oscillation. See Karniel and Inbar (in press) for a review of these and other problems in biological motor control.

Example 6.2: Multiple Feedback Loops

The importance of the loop gain in reducing the sensitivity to parameter changes was mentioned in the previous example. The loop gain can also be a major factor in determining the stability of the system. In order to measure the loop gain, one should open the loop, introduce a test input at one end and measure the output at the other. However, in biological systems there are typically multiple parallel feedbacks (see, e.g., Milgram and Inbar 1976; Windhorst 1996). For example, in the temperature regulation system, there are sensors in the skin, in the core of the body and in the hypothalamus, and they all influence the temperature regulation mechanisms (see Brown and Brengelmann 1970). In movement control, there are feedback loops from sensors in the muscles, joints and skin (i.e., muscle spindles, Golgi tendon organs, pressure transducers etc.), and furthermore there are many sensors of each type operating in parallel. The primary advantage of such multiple loops, and of any redundancy, is robustness. That is, if one subsystem fails, there are other options to operate the system. There is a great danger in trying to estimate the loop gain in such a system because there may be loops that we

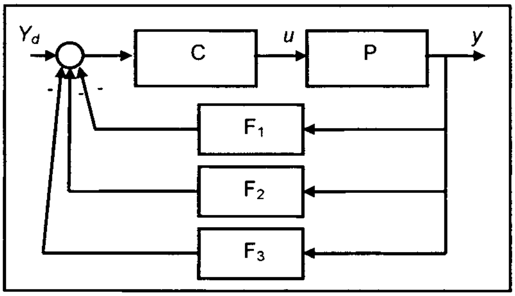


Fig. 19. Multiple feedback loops

cannot open or are unaware of. In such a case we may underestimate the loop gain. For example, if we open the first two loops in Fig. 19 and leave F_3 connected, the transfer function from Y_d to Y will be $P \cdot C / (P \cdot C \cdot F_3 + 1)$ instead of $P \cdot C$ when there is no additional loop. So one should be aware of these multiple loops.

Part 7: Modeling Nonlinear Systems with Linear Systems Description Tools

Many physical and virtually all biological systems are not linear, and many are time-varying systems. Still, we may wish to use the powerful linear systems description tools that were described in this chapter. In this section we will broaden our scope to illustrate the application of linear systems description tools to nonlinear systems. Linearization is a method to find a linear system that is similar to the modeled nonlinear system, at least for a small-signal region and a short time. We will show how nonlinear systems can be described as time-varying or parameter-varying systems, and then we will discuss nonlinear systems as a linear sum of nonlinear functions or as a nonlinear function of a linear system.

Linearization

Linearization is the procedure of finding a linear system that is similar to the modeled nonlinear system in some domain near a point that is called the "working point". Graphically one can imagine the linearized model as a tangent of the nonlinear function. Mathematically the linearization represents the first two terms in the Taylor expansion of the function.

Procedure 7.1: Static Systems Linearization

Static systems linearization simply means to take the first two components in the Taylor expansion. For a single input and single output system, this means taking the following linear estimation $F_L(u)$ of the nonlinear system $F(u)$ near the point $\hat{u}$ which is called the *working point*.

$$F_L(u) = F(\hat{u}) + \left.\frac{dF}{du}\right|_{u=\hat{u}} \cdot (u - \hat{u})$$

This estimation is good for smooth functions, near the working point, and it may be very poor for distant points. The same estimation can be implemented for multiple input static systems.

Procedure 7.2: Dynamic Systems Linearization

Dynamic systems can be represented as a set of differential equations in the state space. If we assume continuity of the function, we can write the first two parts of the Taylor's expansion of $\dot{X} = F(X)$ near the point of interest $\hat{X}$, which is:

$$\dot{x} = f_i(x_1,\ldots,x_n) = f_i(\hat{x}_1,\ldots,\hat{x}_n) + \left.\frac{\partial f_i}{\partial x_1}\right|_{X=\hat{X}} (X_1 - \hat{X}_1) + \ldots + \left.\frac{\partial f_i}{\partial xn}\right|_{X=\hat{X}} (X_n - \hat{X}_n)$$
$$+ O\left(\left\|X - \hat{X}\right\|^2\right)$$

The Jacobean matrix is:

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{X=\hat{X}}$$

And by defining a new variable $Z = X - \hat{X}$, we can get the following linear state function: $\dot{z} = A \cdot z$

The solution of this differential equation is an exponential function. It is interesting to know whether the solution decays, that is, whether the system is stable. We can answer this question by checking the eigenvalues λ_i of the Jacobean matrix A: The system is stable if and only if the real part of all the eigenvalues λ_i is negative. The eigenvalues can be calculated by finding the roots of the equation $|A - \lambda \cdot I| = 0$, or by just writing the proper command in MATLAB or in any other mathematical software.

Example 7.1: Relation between Force and Length of the Muscle

Striated muscles consist of actin and myosin filaments that slide one over the other. As a result of this infrastructure, there is an optimal length, L_o, at which the muscle can produce maximum force. Therefore, the relation between the length of a muscle and its force is nonlinear and can be approximated by the following nonlinear equation:

$$F = F_{\max} - (L_0 - x)^2$$

Suppose that we want a linear model of this muscle near a working point $\hat{x} = L_0 / 2$; we can then linearize the above relation according to Procedure 7.1 as follows:

$$F_L(x) = F(\hat{x}) + \left.\frac{dF}{dX}\right|_{x=\hat{x}} \cdot (x - \hat{x}) = F_{\max} - \left(L_0 - \frac{L_0}{2}\right)^2 + 2 \cdot \frac{L_0}{2} \cdot \left(x - \frac{L_0}{2}\right) =$$

$$= F_{\max} - \frac{3 \cdot L_0^{\ 2}}{4} + L_0 \cdot x$$

Comments

1. The model in Example 3.5 is actually a linearized model of the muscle (see McRuer et al. 1968).
2. The membrane model in Example 3.2 can be seen as linearization of the membrane properties near equilibrium points.
3. One should notice that the linearized model is close to the modeled function only for small perturbations near the working point.

Nonlinear Systems as Linear Time- or Parameter-varying Systems

In Procedures 3.1 and 3.2we considered electrical and mechanical models of linear systems. These procedures can be used to describe nonlinear systems if one allows change in the values of the elements as a function of time or of other values in the system. We give here two examples for such a modeling approach: the membrane electric model and the muscle mechanical model.

Example 7.2: Hodgkin-Huxley Model

This model was introduced in Example 3.2, where the values of the resistive elements were fixed. However, the membrane is not linear and resistance to each ion current is not a constant (see Hodgkin and Huxley (1952) for a comprehensive description of this nonlinear model).

This model is illustrated in Fig. 20, where the arrows on the resistors mean that the resistance is not a constant. The membrane resistance to sodium and its resistance to potassium change as a function of the membrane potential and as a function of time.

The equations that describe this model are the following set of nonlinear differential equations:

$$I_m = C_m \frac{\partial V_m}{\partial t} + g_{Na} \cdot (V_m - V_{Na}) + g_K \cdot (V_m - V_K) + g_{cl} \cdot (V_m - V_{cl})$$
$$g_{Na} = \bar{g}_{Na} \cdot m^3 \cdot h$$
$$g_K = \bar{g}_K \cdot n^4$$
$$g_{cl} = \bar{g}_{cl} = const$$
$$\dot{n}(t) = \alpha_n \cdot (1-n) - \beta_n \cdot n$$
$$\dot{m}(t) = \alpha_m \cdot (1-m) - \beta_m \cdot m$$
$$\dot{h}(t) = \alpha_h \cdot (1-h) - \beta_h \cdot h$$

The first equation can be written according to Procedure 3.1, it is the differential equation of the electrical model in Fig. 20, which is similar to the equation in Example 3.2, where resistance, R, was used instead of conductance, g. We will not go into the details of the change in membrane resistivity, but one can notice in the equations above that this change is also described by linear systems description tools, in fact by a simple first-order differential equation.

Example 7.3: Hill-Type Muscle Model

Let us consider the Hill-type mechanical muscle model in Fig. 21. This model is taken from Zangemeister et al. (1981), with minor changes (see Karniel and Inbar 1997). Note that this model combines a mechanical description with a block diagram in the Laplace domain. It is similar to the model in Example 3.6, but shows three differences: The parallel spring was omitted, a first-order filter was added to describe the excitation-con-

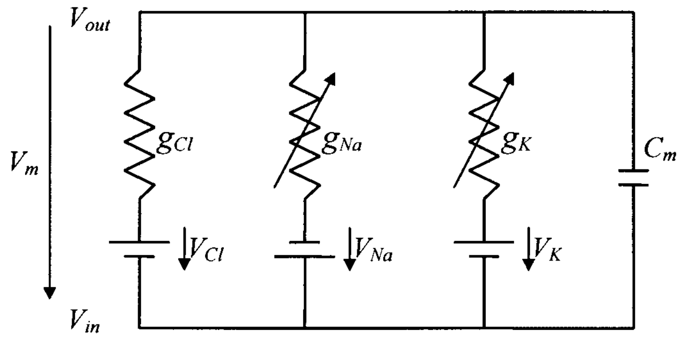


Fig. 20. The Hodgkin-Huxley electrical model of the membrane.

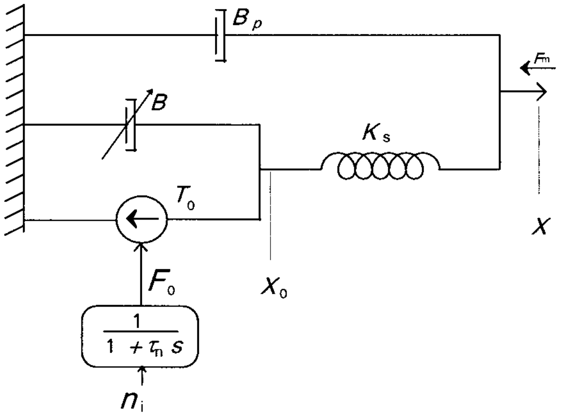


Fig. 21. Mechanical muscle model. n_i is the neural input. The first-order filter represents the excitation-contraction coupling. T_0 is the hypothetical force in the muscle. B represents the relation between force and velocity from Hill's model. The other elements represent the mechanical properties of the tendon and other connective tissues around the joint.

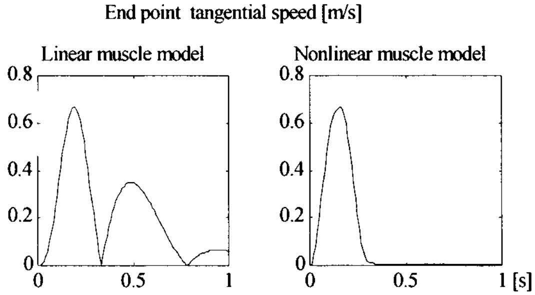


Fig. 22. A comparison of the speed profile of the end point of a two-degrees-of-freedom anthropomorphic arm with a linear muscle model (left) and with a nonlinear muscle model (right) in response to typical rectangular pulse activation of the muscles. Only the nonlinear muscle model yields a bell-shaped speed profile with a smooth stop (for more details, see Karniel and Inbar 1997)

traction coupling and the recruitment of the motor units, and the main difference is the viscous element which is not a constant.

Following are the differential equations of this mechanical model:

$$\dot{F}_0 = \frac{1}{\tau_n} \cdot (n_i - F_0)$$

$$T_0 = F_0 \cdot F_{\max}$$

$$\dot{X}_0 = \frac{\left(K_s \cdot (X - X_0) - T_0\right)}{B}$$

$$F_m = B_P \cdot \dot{X} + K_s \cdot (X - X_0)$$

This model was derived from the Hill model where the value of the viscous element, B, depends on the internal force and on the contraction velocity:

$$B = \begin{cases} (a \cdot T_0)/(b + v) & v \geq 0 \\ a' \cdot T_0 & v < 0 \end{cases}$$

The value of B was taken as a constant in several models, for the sake of simplicity, in order to get a linear model of the muscle. This linear model is under-damped and therefore overshoots, and oscillations are most likely to appear in the controlled movement. This problem is avoided by the use of the more realistic nonlinear model, as demonstrated in Fig. 22 for a very basic movement, the reaching movement.

This example demonstrates how nonlinearity might be exploited advantageously by nature. However, in order to simulate and analyze this phenomenon, we exploited the linear systems description tools.

Comment Another useful model is the second-order muscle model which was introduced in Example 3.5. In this model, in order to get a more realistic behavior, the stiffness must change as a function of the activation, the length and the velocity of the muscle (see for example Inbar 1996).

Pre-processing or Post-processing

In this subsection we describe two simple ways of combining linear systems in nonlinear modeling. The first one is to describe a linear combination of nonlinear fixed functions, which is called a pre-processing, since the inputs are processed prior to their entrance to the linear model, see Fig. 23. The second way is to describe a nonlinear function of a linear combination, which is called "post-processing" since the output of the linear model is nonlinearly processed, see Fig. 24. Both models can take advantage of the linear parameter estimation tools in order to estimate the parameters of the linear part of the model.

Example 7.3: Artificial Neural Networks

The growing field of neural computation is based on combinations of linear and nonlinear elements. The perceptron which is the basic threshold element of neural networks is built as in Fig. 24, where the function $F(x)$ is a step function or any other sigmoidal function.

It is well known that any function can be approximated with the Taylor expansion as a polynomial function. So one can choose the functions $F_i(u_i)$ in Fig. 23 to be: 1, u, u^2, u^3, etc. and then this model can estimate any continuous function. The field of neural computation contains numerous examples for these kinds of models, see Chapter 25.

Comment: Overfit and Underfit

The problem of choosing the order of the model raised in the previous section on linear systems is a major problem in the field of neural computation. Both too many or too few parameters should be avoided. *Under-fit* is the situation where the model is less complex

Fig. 23. Linear combination of nonlinear functions

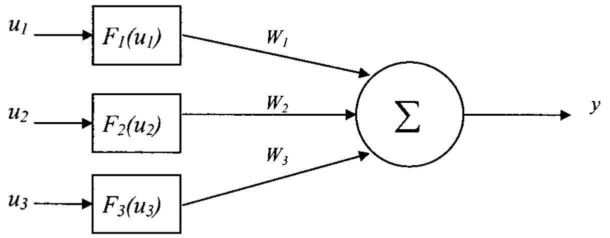
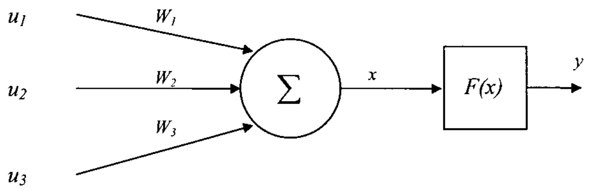


Fig. 24. Nonlinear function of a linear sum



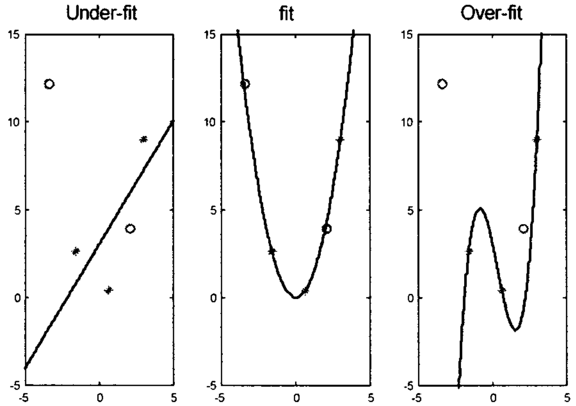


Fig. 25. Fitting a model to data. In this illustration, the three stars are the data taken from an underlying unknown function. On the left, a linear function was fitted to the data. In the middle, a quadratic function was fitted; and on the right, a third-order polynomial function was fitted. After the fitting was completed, two more examples were taken from the same underlying function (the two circles). One can see that the left model is too simple, i.e., under-fits the data, and the right model is too complex, i.e., over-fits the data, but unfortunately does not fit the underlying system.

than the actual system. In this case, the model is unable to fit the data. See Fig. 25on the left.

Over-fit is the situation where the model is more complex than the actual system. In this case, the model will fit the observations. However, if there is noise or insufficient observations (that is, the number of independent observations is smaller then the number of parameters), then the model will not fit the actual system, and in the validation process it may fail to predict the outcome of the system. (In the validation process we check the generalization, that is, the ability of the identified model to deal with examples that were not used for the fitting). See Fig. 25on the right.

Example 7.5: Single-sign Integrated Pulse Frequency Modulation

The transformation of graded membrane potentials into sequences of action potentials in nerve cells is often modeled by single-sign integrated pulse frequency modulation (IPFM), as illustrated in Fig. 26. The input (membrane potential) is integrated (the block *1/s*), and when the value of the integrator reaches a threshold A, the pulse shaper (P.S.), which can be anatomically related to the axon hillock, produces an action potential that resets the integrator and is the output of the system. This model has a linear part which is the integration, and a nonlinear part which is the threshold.

This model can also be combined with the model in Example 3.1 in order to account for multiple inputs from different synapses that influence the value of the integrator.

Let us analyze this nonlinear model and perform linearization in order to find a similar linear model.

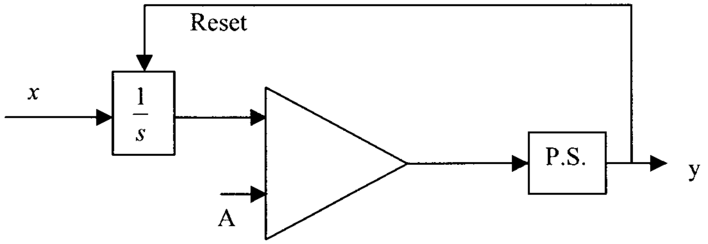


Fig. 26. The integrated pulse frequency modulation (IPFM) model for neural coding

The output is a series of pulses, and we are interested only in the time of each pulse. We also know that the integrator is reset exactly when a new pulse is generated, therefore we know that the integral of the input between two pulses is equal to the threshold, that is:

$$A = \int_{t_k}^{t_{k+1}} x(t) \cdot dt = X(t_{k+1}) - X(t_k) \qquad \text{where } x(t) \equiv \frac{dX(t)}{dt}$$

Let us write the first elements of the Taylor expansion for $X(t_{k+1})$ near t_k:

$$X(t_{k+1}) \cong X(t_k) + x(t_k) \cdot (t_{k+1} - t_k)$$

Let us define the frequency of the output as one over the interval between two pulses and combine the above equations to get

$$A = X(t_{k+1}) - X(t_k) \cong x(t_k) \cdot (t_{k+1} - t_k) = \frac{x(t_k)}{f}$$

Therefore we can conclude that the relation between the frequency of the output and the input signal is

$$f(t) \cong \frac{1}{A} \cdot x(t)$$

We have arrived at the simplest linear system there is. We can take this IPFM as a model of a piece of the retina, the input x as the intensity of light and the output as the firing rate of the optic nerve, and therefore we can close this chapter with the very first example that opened it.

Conclusions

Linear systems description is a very powerful tool used extensively in all branches of science and technology. Biological systems are generally not linear, and a purely linear model is thus seldom satisfactory. Nevertheless, linear systems description tools have important advantages due to their simplicity, analyzability and tractability. They can also be used in one or more of the following model types: locally linear, short-term linear, linear time-varying, linear parameter-varying, linear combinations of nonlinear functions and nonlinear functions of linear combinations. Therefore linear systems description tools are not expected to become obsolete in the near future.

References

Allin J, Inbar GF (1986) FNS Parameter Selection and Upper Limb Characterization. IEEE transactions on biomedical engineering 33:809–817

Benignus VA (1969) Estimation of the coherence spectrum and its confidence interval using the fast fourier transform, IEEE Trans. Audio Electroacoustics, 17:145–150 and correction in 18:320.

Brown AC, Brengelmann GL (1970) The interaction of peripheral and central inputs in the temperature regulation system. In JD Hardy, AP Gagge, JAJ Stolwijk (Eds) Physiological and Behavioral Temperature Regulation, Chapter 47, pp 684–702.

Cadzo JA and Solomon OM (1987) Linear Modeling and the Coherence Function. IEEE Trans. On Acoustic, Speech, and Signal Processing ASSP-35:19–28

Conant R (1993) Engineering Circuit Analysis with Pspice and Probe. McGraw-Hill, Inc, New York.

Charles AD, Kuh ES (1969) Basic Circuit Theory, International student edition, McGraw-Hill KogaKusha, Ltd.

Dorny CN (1993) Understanding Dynamic Systems. Prentice-Hall, Inc.
Fausett Laurene (1994) Fundamentals of Neural Networks. Prentice Hall International, Inc.
Hebb DO (1949) The organization of behaviour, New York: Wiley.
Hodgkin AL, Huxley AF (1952) A Quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol. 117:500–544
Houk JC (1963) A mathematical model of the stretch reflex in human muscle systems. M.S. Thesis, Massachusetts Institute of Technology, Cambridge.
Inbar GF (1996) Estimation of Human Elbow Joint Mechanical Transfer Function During Steady State and During Cyclical Movements. In:Gath I & Inbar GF (eds.) Advances in Processing and Pattern Analysis of Biological Signals. Plenum Press.
Karniel A, Inbar GF (1997) A Model for Learning Human Reaching Movements. Biological Cybernetics 77:173–183.
Karniel A, Inbar GF (in press) Human Motor Control: Learning to control a Time-Varying Non-Linear Many-to-One System. Accepted for publication in IEEE Transactions on system, man, and cybernetics Part C.
Kwakernaak H. and R. Sivan (1991) Modern signals and systems. Prentice-Hall International, Inc.
Lathi BP (1974) Signals, Systems, and Controls. Intext Educational Publishers, New York.
Levine WS Ed.(1996) The control handbook. CRC press.
Ljung L (1986) System Identification Toolbox: The manual. The Mathworks Inc. 1st Ed., (4th ed. 1994) Natick, MA.
McRuer DT, Magdaleno RE, Moore GP (1968) A Neuromuscular Actuation System Model. IEEE Trans. On man-machine systems 9:61–71
Milgram P, Inbar GF. (1976) Distortion Suppression in Neuromuscular Information Transmission Due to Interchannel Dispersion in Muscle Spindle Firing Thresholds. IEEE transactions on biomedical engineering 23:1–15
Nilsson JW, Riedel SA (1996) Using Computer Tools for Electric Circuits. Fith Edition Addison-Wesley, Inc.
Paiss O, Inbar GF (1987) Autoregressive Modeling of Surface EMG and Its Spectrum with Application to Fatigue. IEEE transactions on biomedical engineering 34:761–770
Plonsey R. and R.C. Barr (1988) Bioelectricity A quantitative approach. Plenum Press, New York and London.
Porat B. (1994) Digital processing of random signals: theory and methods. Prentice-Hall, Inc, New Jersey.
Santina SM, Stubberud AR, Hostetter GH (1996) Discrete-Time Equivalents to Continuous-Time Systems. In Levine WS (Ed) The Control Handbook , CRC press Inc. pp:265–279.
Sjoberg J, Zhang Q, Ljung L, Benveniste A, Delyon B, Glorennec P, Hjalmarsson H, and Juditsky A (1995) Nonlinear Black-box Modeling in System Identification: a Unified Overview. Automatica, 31:1691–1724.
Stein RB (1980) Nerve and Muscle membranes, cells, and systems. Plenum Press, New York.
Windhorst U (1996) Chapter 1. Regulatory principles in physiology. In: Greger R, Windhorst U (eds) Comprehensive human physiology. From cellular mechanisms to integration, pp. 21–42. Springer-Verlag, Berlin Heidelberg
Zangemeister WH, Lehman S, Stark L (1981) Simulation of Head Movement Trajectories: Model and Fit to Main Sequence. Biol. Cybern. 41:19–32