Lab 6: Introduction to Signal Processing – Convolution, Transformations and Aliasing

This laboratory requires the following equipment:

- Matlab

The laboratory duration is approximately 3 hours. Although this laboratory is not graded, we encourage you to take your own personal notes as the lab verification test might leverage results acquired during this laboratory session. For any questions, please contact us at **sis-ta@groupes.epfl.ch**

1.1 Information

In the following, text you will find several exercises and questions.

- The notation **S** means that the question can be solved using only additional simulation.
- The notation **Q** means that the question can be answered theoretically, without any simulation.
- The notation **I** means that the problem has to be solved by implementing a piece of code and performing a simulation.
- The notation **B** means that the question is optional and should be answered if you have enough time at your disposal.

Outline

This lab continues on the subject of signal processing, and in particular, reviews the convolution, discusses the various transforms and the concept of aliasing. In Part 1, you will analyze the convolution in a bit more detailed manner. In Part 2, you will apply different continuous transforms to a first-order continuous system in order to find the step response and to analyze it. In Part 3, you will be given the discrete version of the system. You will apply discrete transformations to obtain the step response and analyze it. In Part 4, you will investigate the concept of aliasing. Every part includes both theoretical and MATLAB related questions. You can find the Laplace and Z-transform tables in the appendix if you need them.

Getting Started

To start with this lab, you will need to download the material available on Moodle. Download `lab06.tar.gz` or `lab06.zip` and extract it in your home directory (you can type `tar xvfz lab06.tar.gz`). Now start Matlab and change your “Current directory” to be `lab06/part01/`.

Please install Control System Toolbox if you have not installed it already. You can do so by pressing Add-Ons button in MATLAB main window.

Part 1: Convolution review

In this part, you will review the concept of convolution in more detail by writing your own convolution function.

Open the Matlab script `part_1.m` in the folder `part_1/`. A `rect` function is given to you along with the corresponding time series.

```
rect = [0 0 1 1 1 0 0];
t = [1 2 3 4 5 6 7];
```

1. (Q): Use the Matlab function `stem` to plot `rect` against time t.

2. **(Q):** Recall how convolution is performed in the discrete domain.

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m] \cdot g[n-m]$$

Perform the convolution of the `rect` function above by hand using the above equation and plot it on paper. What shape do you see? How many elements are present in the result?

Hint: n would go from -3 to 3, and the limits of the summation would also be from m=-3 to m=3. For any time outside this range, assume the function value is zero. Example: g[n-m] = g[-4] = 0.

3. **(I):** Use the Matlab function `conv` to do a convolution of `rect` with itself using the 'full' as well as 'same' parameter, just as you did in the previous lab. What shape do you see? What is the difference in the result when 'full' or 'same' is used? Why are the lengths different? What would be the corresponding time series in each case?

Note: The code is already present in the script but commented.

4. **(I):** Write the code for performing the convolution by yourself (i.e, without using the matlab function `conv`). For your convenience, two copies of `rect` have been made as a substitute of *f* and *g*: `rect_1`, `rect_2`.

Hint: You will need two for loops, one inside the other.

Part 2: Continuous-Time Transforms

In this part, you will apply Continuous-Time (CT) transformations (Fourier and Laplace transforms) to a first-order system in order to obtain its step response and analyze the response. Open the script `part_2_3.m` in the folder `part_2_3` and check the corresponding questions.

Step response of a system is defined as the time evolution of its output when its input is step function. This can be shown from the Fig. 1.

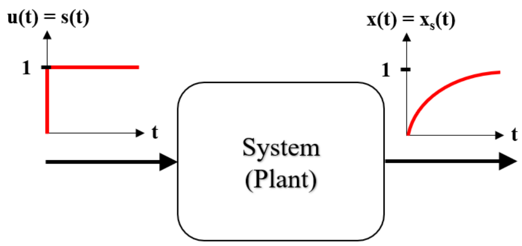

Figure 1. Step input s(t) and step response $x_s(t)$

Consider the analog circuitry given in Fig.2. This circuitry consists of an independent voltage source v_s, a resistor *R*, and a capacitor *C*, whose voltage is shown by v_c.

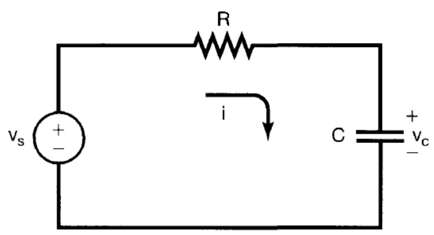

Figure 2. RC circuitry [1]

This system can be represented by a continuous first-order differential equation, if we select input as v_s and output as v_c. The equation can be written as follows:

$$\frac{dv_c(t)}{dt} = \frac{1}{RC}\,(v_s(t) - v_c(t))$$

If we define *u(t)* as input, *x(t)* as state and the multiplication of *R* and *C* as Γ, the equation has a new form as follows:

$$\frac{dx(t)}{dt} = \frac{1}{\Gamma}\,(u(t) - x(t))$$

Here Γ is known as time constant of the system and it is an indication of how fast the system responds. It is the duration where the system reaches (approximately) 63% of its steady state (final) value. (Pay attention to this definition, you will use this to extract time constants from the plots by adding data points on it.)

In order to find the step response (i.e. the state of the system when the input is a step function) in time domain, we should substitute the step function *s(t)* into *u(t)*. The definition of *s(t)* is given as follows:

$$s(t) = \begin{cases} 1\,, if\ t \geq 0 \\ 0\ \ , otherwise \end{cases}$$

Considering the complimentary and particular solutions of the first-order differential equations, the solution (step response) of this differential equation can be found as follows:

$$x_s(t) = \left(-e^{\left(-\frac{t}{\Gamma}\right)} + 1\right) \cdot s(t)$$

5. **(I):** Given that R = 1 Ω and C = 0.1 F, find the impulse response *h(t)* of the system from the step response. Plot both the step response and impulse response using MATLAB and observe the figure. Can you identify the time-constant from the figure?

   Hint-1: Since the system is linear, the impulse response of the system is the derivative of the step response with respect to time.
   Hint-2: Write the functions and variables symbolically using `syms`. *You can use* `heaviside()` *to represent the unit step input. You can use* `fplot()` *to plot symbolic functions.*
   Hint-3: To have more accurate unit step function use `sympref('HeavisideAtOrigin',1)` *before using step function.*
   Hint-3: You can plot all responses between the interval [0,0.6].

6. **(I):** Since you know the unit step input and found the impulse response in time domain. You can find the step response by using convolution of both analytically. However, there is no symbolic convolution function in MATLAB. Instead, we can use the definition of continuous convolution to calculate it.

$$x_s(t) = s(t) * h(t) = \int_{-\infty}^{\infty} s(\tau) \cdot h(t - \tau)\, d\tau$$

   Obtain the step response, plot and compare it with the one that you found in question **5(I)**. Do the overall shape and the time-constant differ?

   Hint-1: Use `int()` *function to calculate symbolic integration.*
   Hint-2: You can use `subs()` *function to substitute any value to a symbolic function.*
   Hint-3: You can use `inf, -inf` *variables to define integral bounds.*

Hint-4: Use `simplify()` *function to simplify the symbolic expression calculated by above functions. Sometimes, even if you use* `simplify()`, *you can obtain complicated expressions from symbolic tools. This is because symbolic tools consider all time axis. In order to further simplify expression for the time greater than zero, before calculation of the expression add* `assume(t>0)` *where t is your time variable. In order to remove assumption from the variable use* `assume(t,'clear')`.

7. **(I):** Analyze the frequency content of the step response $x_s(t)$ by applying the CT Fourier transform in MATLAB. Plot the magnitude of the response and observe it. Pay attention to the unit of frequency axis. Is it in Hertz or radians/sec?

   Hint-1: Use `fourier()` *function to calculate fourier transform symbolically.*
   Hint-2: Do not forget to obtain absolute value.

8. **(I):** Another way to obtain system response is by using Laplace transform. This transform is more generic version of the Fourier transform and generally used in system analysis. Considering the properties of Laplace transform, it is known that the convolution in time domain corresponds to the multiplication in the frequency domain, and this is valid also for the Laplace transform.

$$x_s(t) = s(t) * h(t) \leftrightarrow S(s) \cdot \mathrm{H(s)} = X_s(s)$$

   Where *S(s)* and *H(s)* are the Laplace transform of *s(t)* and *h(t)*. Therefore, by taking the inverse Laplace transform of the multiplication of *S(s)* and *H(s)*, we can obtain the step response in time domain.

$$x_s(t) = \mathcal{L}^{-1}(S(s) \cdot \mathrm{H(s)})$$

   Where $H(s) = \frac{X(s)}{U(s)}$ is also known as the transfer function of the CT system. Notice that the Laplace transform of the impulse response *h(t)* is the transfer function.

   a. **(B)** Compute *S(s)* and *H(s)* by using the definition of Laplace transform or Laplace transform tables in appendix. Find the step response $x_s(t)$ in time domain by using the definition of inverse Laplace transform or inverse Laplace transform tables.

   b. Apply the same procedure in MATLAB symbolically and find the step response $x_s(t)$. Plot and compare the result with the one you found in questions **5(I)** and **6(I)**.

      Hint-1: Use `laplace()` *function to calculate Laplace transform symbolically.*
      Hint-2: Use `ilaplace()` *function to calculate inverse Laplace transform symbolically.*
      Hint-3: Use `simplify()` *function to simplify the symbolic expression calculated by above functions. Sometimes, even if you use* `simplify()`, *you can obtain complicated expressions from symbolic tools. This is because symbolic tools consider all time axis. In order to further simplify expression for the time greater than zero, before calculation of the expression, add* `assume(t>0)` *where t is your time variable. In order to remove assumption from the variable use* `assume(t,'clear')`.

   c. Observe *H(s)* and represent it in MATLAB using `tf()` function. Use `step()` function together with resulting transfer function to obtain the step response. Plot and compare the result with the one you found in question **5(I), 6(I)** and **8.b (I)**. Do the overall shape and time-constant differ?

Hint-1: After you obtain H(s) symbolically, take note of the numerator and denominator coefficients and enter this into `tf()`*. For example if your H(s)=0.1/(0.4s-0.3) use Hs =* `tf(0.1,[0.4,-0.3])`.
Hint-2: `step(Hs)` *directly gives and plots the step response where Hs is your transfer function.*

Note: The CT of the Fourier transform is a special case of the Laplace transform. Convergence of the Laplace transform is broader than that of the CT Fourier transform, i.e. includes unstable systems. Therefore, while the Laplace transform is mostly used in system analysis, the CT Fourier transform is mostly used in continuous signal processing.

Part 3: Discrete-Time Transforms

In this part, you will apply some Discrete-Time (DT) transformations (Z-transform and DT Fourier Transform) to a first-order system in order to obtain step response of the system and analyze the response. Open the script `part_2_3.m` in the folder `part_2_3` and check the corresponding questions.

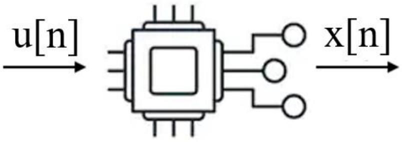

Figure 3. Digital system

Now, we would like to represent our continuous-time system (RC circuitry) in an embedded digital computer as in Fig.3. By using an appropriate discretization method, we can find the difference equation of the system as follows:

$$x[n] = \left(1 - \frac{\Delta t}{\Gamma}\right) x[n-1] + \left(\frac{\Delta t}{\Gamma}\right) u[n-1]$$

where, Δt is the sampling period and Γ is defined in previous questions. Take Δt as 0.01 seconds.

9. (I): In order to find the system response *x[n]* (solution to differential equation), we first need to find the transfer function *H(z)* of the system (in Z-domain). We will use Z-transform to find this since the system is represented in discrete time. Note that Z-transform has similar purpose for discrete functions and systems as the Laplace transform for continuous functions and systems.

a. **(B)** Take Z-transform of the both sides of the difference equation by using Z-transform tables given in appendix. Manipulate the equation so that obtain the transfer function *H(z)* $= \frac{X(z)}{U(z)}$ symbolically.

Hint-1: Use `ztrans()` *function to take z transform of the any function symbolically.*
Hint-2: Use "linearity" and "time shift" properties of z transform while solving the problem analytically.

b. By applying the method in part "a", it can be found that $H(z) = \frac{0.1}{z-0.9}$, represent it as MATLAB transfer function using `tf()`. Use `step()` function together with resulting transfer function to obtain the step response. Plot and compare the result with the one you found in questions **5(I), 6(I)** and **8(I)**. Is there any difference between the CT and DT counterparts? Do the overall shape and the time-constant differ?

Hint-1: After you obtain H(z) symbolically, take note of the numerator and denominator coefficients and enter this into `tf()`. *For example if your H(z)=0.1/(0.4z-0.3) use Hz =* `tf(0.1,[0.4,-0.3],Δt)` *where Δt is your sample time for discrete system.*
Hint-2: `step(Hz)` *directly gives and plots the step response where Hz is your transfer function.*

c. Similar to previous questions, take input *u[n]* as unit step input *s[n]* whose definition is given as

$$s[n] = \begin{cases} 1\,, if\ n \geq 0 \\ 0\ \ , otherwise \end{cases}$$

Similar to Laplace transform, the convolution in time domain is the multiplication in z-domain.

$$x[n] = s[n] * h[n] \ \leftrightarrow\ S(z) \cdot \mathrm{H(z)} = X(z)$$

Where *S(z)* and *H(z)* are the z-transforms of *s[n]* and *h[n]*. Therefore, by taking the inverse Z-transform of the multiplication, we can obtain the step response in time domain.

$$x_s[n] = \ Z^{-1}(S(z) \cdot \mathrm{H(z)})$$

Apply this procedure analytically by using transform tables **(B)** and in MATLAB to obtain the step response $x_s[n]$. Plot and compare the result with the one you found in questions **5(I), 6(I), 8 (I)** and **9(I).b**. Do the overall shape and the time-constant differ?

Hint-1: You already obtained H(z). Find the z-transform of s[n] by using `ztrans()`.
Multiply with H(z) and take the inverse transform symbolically by using `iztrans()`.
Hint-2: You can use `fplot()` *to plot symbolic functions.*

10. (I): Define the time series for *n* using the code below:

```
n = 0:1:60;
```

Represent the step response $x_s[n]$ (found in **9(I).c**) as time-series by using *n.* There is no symbolic function to calculate the DT Fourier transform in MATLAB. However, we can use `fft()` to calculate it. Use `calculate_DTFT_and_FFT()` to plot transforms in in the folder `part_3/` . Input your response series to the functions. Observe how `calculate_DTFT_and_FFT()` is utilizing MATLAB's built in function `fft()` to calculate the DT Fourier transform. Observe the differences between FFT and DT Fourier transform plots. Observe the frequency content of the step response and compare it with the one you found in question **7(I)**.

Note: the DT Fourier transform is a special case of Z-transform. The convergence of Z-transform is broader than that of the DT Fourier transform, i.e. it includes unstable systems. Therefore, while the Z-transform is mostly used in system analysis, the DT Fourier transform is mostly used in digital signal processing.

Part 4: Aliasing

In this part, you will play with sampling frequencies to understand aliasing. Open the script `part_4.m` in the folder `part_4/`. You will study what happens when the maximal frequency of the signal is higher than half of the sampling frequency, therefore violating the Nyquist-Shannon theorem for an error-free reconstruction of the signal.

The code contains a single sine wave with a single frequency. You can change the frequency of the sine wave, F, as well as the sampling frequency, Fs. If you run the code, it plots the true FFT as well as the FFT of the sampled signal.

Note: Do not worry about the magnitude of the signal, it may have an unexpected value in some cases due to noise. In those cases, you can see the noise on the FFT plot.

11. (S): You can change the `F` and `Fs` in lines 5,6,7 (the units are Hz). Run the code for the default values and verify that the FFT is plotted as expected.

12. (S): Repeat by setting `F` = 10 and `Fs` = 20, 18, 16, and 10. What do you see in the FFT of the sampled signal?

Note: The frequency at which you see the peak is the alias frequency, and it is different from `F`.

13. (S): Find a relation between the frequency of the signal, `F`, sampling frequency, `Fs`, and the alias frequency `Fa`.

14. (S): Recall from the lecture that aliasing can cause challenges during reconstruction because of overlapping of frequencies in the frequency domain. However, if there is no overlapping, you can benefit from aliasing.

Consider a situation where you want to find the frequency components of a signal. You know that the frequency components are in the range of 50-60Hz. You have a data acquisition system with a maximum sampling frequency, `Fs_max` = 50Hz. What sampling frequency will you use?

Appendix: Transform tables

Laplace Transform [1]

TABLE 9.1 PROPERTIES OF THE LAPLACE TRANSFORM

Section	Property	Signal	Laplace Transform	ROC
		$x(t)$ $x_1(t)$ $x_2(t)$	$X(s)$ $X_1(s)$ $X_2(s)$	R R_1 R_2
9.5.1	Linearity	$ax_1(t) + bx_2(t)$	$aX_1(s) + bX_2(s)$	At least $R_1 \cap R_2$
9.5.2	Time shifting	$x(t - t_0)$	$e^{-st_0}X(s)$	R
9.5.3	Shifting in the s-Domain	$e^{s_0 t}x(t)$	$X(s - s_0)$	Shifted version of R (i.e., s is in the ROC if $s - s_0$ is in R)
9.5.4	Time scaling	$x(at)$	$\frac{1}{\|a\|}X\left(\frac{s}{a}\right)$	Scaled ROC (i.e., s is in the ROC if s/a is in R)
9.5.5	Conjugation	$x^*(t)$	$X^*(s^*)$	R
9.5.6	Convolution	$x_1(t) * x_2(t)$	$X_1(s)X_2(s)$	At least $R_1 \cap R_2$
9.5.7	Differentiation in the Time Domain	$\frac{d}{dt}x(t)$	$sX(s)$	At least R
9.5.8	Differentiation in the s-Domain	$-tx(t)$	$\frac{d}{ds}X(s)$	R
9.5.9	Integration in the Time Domain	$\int_{-\infty}^{t} x(\tau)d(\tau)$	$\frac{1}{s}X(s)$	At least $R \cap \{\mathcal{R}e\{s\} > 0\}$

TABLE 9.2 LAPLACE TRANSFORMS OF ELEMENTARY FUNCTIONS

Transform pair	Signal	Transform	ROC
1	$\delta(t)$	1	All s
2	$u(t)$	$\frac{1}{s}$	$\mathcal{R}e\{s\} > 0$
3	$-u(-t)$	$\frac{1}{s}$	$\mathcal{R}e\{s\} < 0$
4	$\frac{t^{n-1}}{(n-1)!}u(t)$	$\frac{1}{s^n}$	$\mathcal{R}e\{s\} > 0$
5	$-\frac{t^{n-1}}{(n-1)!}u(-t)$	$\frac{1}{s^n}$	$\mathcal{R}e\{s\} < 0$
6	$e^{-\alpha t}u(t)$	$\frac{1}{s+\alpha}$	$\mathcal{R}e\{s\} > -\alpha$
7	$-e^{-\alpha t}u(-t)$	$\frac{1}{s+\alpha}$	$\mathcal{R}e\{s\} < -\alpha$
8	$\frac{t^{n-1}}{(n-1)!}e^{-\alpha t}u(t)$	$\frac{1}{(s+\alpha)^n}$	$\mathcal{R}e\{s\} > -\alpha$
9	$-\frac{t^{n-1}}{(n-1)!}e^{-\alpha t}u(-t)$	$\frac{1}{(s+\alpha)^n}$	$\mathcal{R}e\{s\} < -\alpha$
10	$\delta(t - T)$	e^{-sT}	All s
11	$[\cos \omega_0 t]u(t)$	$\frac{s}{s^2 + \omega_0^2}$	$\mathcal{R}e\{s\} > 0$
12	$[\sin \omega_0 t]u(t)$	$\frac{\omega_0}{s^2 + \omega_0^2}$	$\mathcal{R}e\{s\} > 0$
13	$[e^{-\alpha t}\cos \omega_0 t]u(t)$	$\frac{s+\alpha}{(s+\alpha)^2 + \omega_0^2}$	$\mathcal{R}e\{s\} > -\alpha$
14	$[e^{-\alpha t}\sin \omega_0 t]u(t)$	$\frac{\omega_0}{(s+\alpha)^2 + \omega_0^2}$	$\mathcal{R}e\{s\} > -\alpha$
15	$u_n(t) = \frac{d^n\delta(t)}{dt^n}$	s^n	All s
16	$u_{-n}(t) = \underbrace{u(t) * \cdots * u(t)}_{n \text{ times}}$	$\frac{1}{s^n}$	$\mathcal{R}e\{s\} > 0$

Z Transform [1]

TABLE 10.1 PROPERTIES OF THE z-TRANSFORM

Section	Property	Signal	z-Transform	ROC
		$x[n]$	$X(z)$	R
		$x_1[n]$	$X_1(z)$	R_1
		$x_2[n]$	$X_2(z)$	R_2
10.5.1	Linearity	$ax_1[n] + bx_2[n]$	$aX_1(z) + bX_2(z)$	At least the intersection of R_1 and R_2
10.5.2	Time shifting	$x[n - n_0]$	$z^{-n_0}X(z)$	R, except for the possible addition or deletion of the origin
10.5.3	Scaling in the z-domain	$e^{j\omega_0 n}x[n]$	$X(e^{-j\omega_0}z)$	R
		$z_0^n x[n]$	$X\left(\frac{z}{z_0}\right)$	$z_0 R$
		$a^n x[n]$	$X(a^{-1}z)$	Scaled version of R (i.e., $\|a\|R$ = the set of points $\{\|a\|z\}$ for z in R)
10.5.4	Time reversal	$x[-n]$	$X(z^{-1})$	Inverted R (i.e., R^{-1} = the set of points z^{-1}, where z is in R)
10.5.5	Time expansion	$x_{(k)}[n] = \begin{cases} x[r], & n = rk \\ 0, & n \neq rk \end{cases}$ for some integer r	$X(z^k)$	$R^{1/k}$ (i.e., the set of points $z^{1/k}$, where z is in R)
10.5.6	Conjugation	$x^*[n]$	$X^*(z^*)$	R
10.5.7	Convolution	$x_1[n] * x_2[n]$	$X_1(z)X_2(z)$	At least the intersection of R_1 and R_2
10.5.7	First difference	$x[n] - x[n-1]$	$(1 - z^{-1})X(z)$	At least the intersection of R and $\|z\| > 0$
10.5.7	Accumulation	$\sum_{k=-\infty}^{n} x[k]$	$\frac{1}{1 - z^{-1}}X(z)$	At least the intersection of R and $\|z\| > 1$
10.5.8	Differentiation in the z-domain	$nx[n]$	$-z\frac{dX(z)}{dz}$	R

TABLE 10.2 SOME COMMON z-TRANSFORM PAIRS

Signal	Transform	ROC
1. $\delta[n]$	1	All z
2. $u[n]$	$\frac{1}{1 - z^{-1}}$	$\|z\| > 1$
3. $-u[-n-1]$	$\frac{1}{1 - z^{-1}}$	$\|z\| < 1$
4. $\delta[n - m]$	z^{-m}	All z, except 0 (if $m > 0$) or ∞ (if $m < 0$)
5. $\alpha^n u[n]$	$\frac{1}{1 - \alpha z^{-1}}$	$\|z\| > \|\alpha\|$
6. $-\alpha^n u[-n-1]$	$\frac{1}{1 - \alpha z^{-1}}$	$\|z\| < \|\alpha\|$
7. $n\alpha^n u[n]$	$\frac{\alpha z^{-1}}{(1 - \alpha z^{-1})^2}$	$\|z\| > \|\alpha\|$
8. $-n\alpha^n u[-n-1]$	$\frac{\alpha z^{-1}}{(1 - \alpha z^{-1})^2}$	$\|z\| < \|\alpha\|$
9. $[\cos\omega_0 n]u[n]$	$\frac{1 - [\cos\omega_0]z^{-1}}{1 - [2\cos\omega_0]z^{-1} + z^{-2}}$	$\|z\| > 1$
10. $[\sin\omega_0 n]u[n]$	$\frac{[\sin\omega_0]z^{-1}}{1 - [2\cos\omega_0]z^{-1} + z^{-2}}$	$\|z\| > 1$
11. $[r^n \cos\omega_0 n]u[n]$	$\frac{1 - [r\cos\omega_0]z^{-1}}{1 - [2r\cos\omega_0]z^{-1} + r^2 z^{-2}}$	$\|z\| > r$
12. $[r^n \sin\omega_0 n]u[n]$	$\frac{[r\sin\omega_0]z^{-1}}{1 - [2r\cos\omega_0]z^{-1} + r^2 z^{-2}}$	$\|z\| > r$

References

[1] Oppenheim A.V. , Willsky A.S., Hamid S., Signals and Systems, Prentice Hall (1996)