First Order Digital Systems

Learning Outcomes

After reading this theory sheet, you should be able to

- recognise the difference between an analogue system and a digital system
- recognise the need for digital systems
- recognise the difference between an analogue differential equation and a digital difference equation
- convert a differential equation into a difference equation using a forward difference method
- solve a first order difference equation using any one or more of three methods.

First Order Analogue Systems

Example1

Consider a particle projected in a straight line through a resisting medium in which the resistance to motion is proportional to the velocity of the particle. This situation can be modelled using

(a) Newton's Second Law of Motion, $F = ma$, Force is mass times acceleration, and
(b) Instantaneous acceleration is given by the differential of velocity, v, with respect to time, t.

The mathematical model for this is given by

$$F = ma = m\frac{dv}{dt} = kv\,,$$

resulting in the ***ordinary, linear, first-order, homogeneous differential equation with constant coefficients***,

$$m\frac{dv}{dt} - kv = 0 \qquad (1.1)$$

where

i) m is the mass of the particle and
ii) k is the constant of proportionality which is necessarily less than zero since the resistance to motion produces a deceleration.

Example 2

Now consider an electrical circuit simply consisting of an inductance, L, and a resistance, R, in series with no applied voltage. The equation relating the current, i, in the circuit at any time, t, is given using Kirchhoff's Second Law, which states that the algebraic sum of all the potential differences in a circuit loop should be zero.

This results in the ***ordinary, linear, first-order, homogeneous differential equation with constant coefficients***

$$L\frac{di}{dt} + Ri = 0 \qquad (1.2)$$

Both of the above examples lead to essentially the same equations, for which we can now define the terms above (in bold italics).

The differential equation is

- *ordinary* because it has only one independent variable - in both cases, time, t (otherwise it would be a *partial* differential equation).
- *linear* because the dependent variable and its derivative are only first degree, i.e. there are no second order terms (or higher) such as $i^3(t)$ or $i(t) \times di/dt$
- *first-order* because the highest order derivative is first-order,
- *homogeneous* because the equation involves one dependent variable **only** (in the above examples velocity, $v(t)$ and current, $i(t)$) and the derivative(s) of the dependent variable, and
- has *constant coefficients* since the dependent variable and its derivative(s) are multiplied only by constants (m, k, L and R are not functions of t.)

These two differential equations can easily be made non-homogeneous by, for example, pushing the particle with a possibly time-dependent external force, $P(t)$, or by applying a possibly time-dependent external voltage, $e(t)$. The resulting governing equations will become

$$m\frac{dv}{dt} - kv = P(t) \qquad (1.3)$$

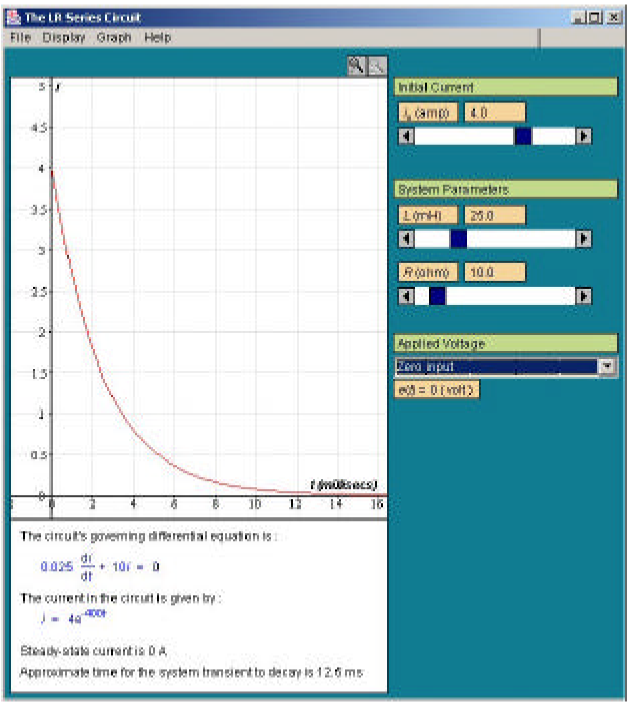
and

$$L\frac{di}{dt} + Ri = e(t) \qquad (1.4)$$

In each case there is now a second time-dependent variable; the equations are now *non-homogeneous* and are the types of equation that will be 'digitised' in the following sections.

All systems listed above, (1.1) → (1.4), can be solved to obtain a continuous-time, analogue solution which, when plotted will produce a continuous solution curve.

For the *particle* problem, the solution will be of the form $v(t)$ = some function of t
For the *circuit* problem the solution will be of the form $i(t)$ = some function of t



The diagram shown on the left is a screenshot from the ***MathinSite*** "LR Series Circuit" applet and shows a typical continuous curve (i.e. analogue) response from an LR circuit with no applied voltage (the homogeneous case). However, without some initial condition a circuit with no applied voltage would have zero response for all time, so here the applet specifies that there is already an initial current of 4 A in the circuit when the clock is started (at t = 0). This is called the ***initial condition*** and for a *first*-order system there needs to be just a *single* initial condition.

System Excitation

To get *some* response from a system there needs to be either

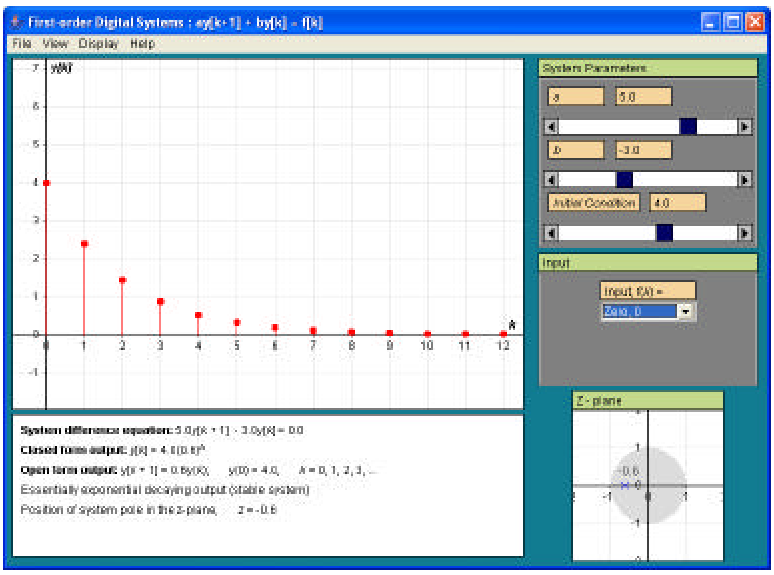
(a) an external driving source (a force or a voltage, for example) or
(b) some non-zero initial condition
(c) *or both.*

A **Zero-input system** is excited only by initial conditions (and has no external force/voltage/etc)
A **Zero-state system** is excited only by a source (and has zero initial conditions)

Digital Systems

The need to digitise signals is essential to engineers involved in, for example, signal processing, filter design and system control.

A **digital system** is one in which data is represented as a series of periodic pulses. The initial data source, often analogue data (represented as a continuous-time signal), is regularly sampled and converted into numerical values. This is usually achieved in practice using analogue-to-digital converters (ADC) and, if necessary, back again using digital-to-analogue converters (DAC). But why is this necessary? The main reason is for computer analysis and control. Computers store their data only in a digital form (at the most basic level as binary numbers) and are therefore are unable to deal with continuous data. A random signal would require the computer to hold an infinite amount of information - hence the need to sample the signal at regular, pre-defined time intervals, so producing a digital, or *discrete-time*, signal.



The diagram shown on the left is a screenshot from the ***MathinSite*** "First Order Digital Systems" applet and shows the equivalent digital output (solution) of the LR continuous-time system shown on page 2.

The 'ball' at the top of each spike is there merely to show the spikes more clearly.

For any system analysis or control by a computer, output/input signals have to be digitised. The LR series circuit above is effectively an electronic filter and to investigate this using a computer takes the engineer into *digital filter design*. The output of such a system will be some form of signal which, when digitised and analysed, takes the engineer into *digital signal processing* (DSP). Using a computer to control a robot arm (which may consist of many electromechanical parts including sensors and actuators), will require conversion of continuous-time displacements and voltages, for example, into digital displacements and voltages; typical digital signals that are used to drive *computer-controlled systems.*

Digitising Analogue Systems

Consider a system whose governing differential equation is the ordinary, linear, first-order, non-homogeneous differential equation with constant coefficients *A* and *B*,

$$A\frac{dy(t)}{dt} + By(t) = f(t) \qquad (1.5)$$

The lower case *t* indicates that the time is continuous and hence the system is an analogue system.

In order to digitise the system, samples have to be taken at regular intervals (usually denoted by the time, *T*, the *sampling time*).

In this case the signal will only exist at

$$0T,\ 1T,\ 2T,\ 3T,\ 4T, \ldots$$

where *T* is the sampling time.

For example if the sampling time is 15 milliseconds (ms), then the signal only has values at times

0 ms, 15 ms, 30 ms, 45 ms, 60 ms, …

In its digitised form, nothing is known of the signal between sample values, i.e. the signal takes no value in between sampling (e.g. at 13ms); *the signal does not exist other than at the sampling times*.

So, in order to make an analogue-to-digital time conversion, use is made of

$$t = kT \quad (k = 0, 1, 2, 3, 4, \ldots)$$

The value of the output and input of the above system are in

- analogue form, $y(t)$ and $f(t)$ respectively
- digital form, for sample k, $y[kT]$ and $f[kT]$ respectively.

[Note that square brackets are often used to indicate that variables relate to discrete time]

But how is the differential $dy(t)/dt$ digitised?

The very beginnings of differential calculus show that the gradient of a *chord* cutting a curve in two points (x_1, y_1) and (x_2, y_2) is given by

$$\text{Gradient of chord} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{dy}{dx}$$

For which the gradient of the *tangent* at (x_1, y_1) is given, as $x_2 - x_1 \rightarrow 0$, by

$$\text{Gradient of tangent} = \frac{dy}{dx} \approx \frac{y_2 - y_1}{x_2 - x_1}$$

Translating this to the above system, where the consecutive k^{th} and the $(k + 1)^{th}$ sampled points have coordinates $(kT, y[kT])$ and $((k+1)T, y[(k + 1)T])$ gives

$$\frac{dy}{dt} \approx \frac{y\left[(k+1)T\right] - y[kT]}{(k+1)T - kT}$$

or

$$\frac{dy}{dt} \approx \frac{y\left[(k+1)T\right] - y[kT]}{T}$$

- and this is called the ***forward difference approximation*** for the first derivative.

Note: There are others, including the backward difference and bilinear approximations. The forward difference approximation is used in the accompanying *MathinSite* applet.

Equation (1.5) can now be completely digitised:

$$A\left(\frac{y\left[(k+1)T\right] - y[kT]}{T}\right) + By[kT] = f[kT] \qquad (1.6)$$

It is convenient to specify $T = 1$ (1 unit of time) so that (1.6) simplifies to

$$A\left(y[k+1] - y[k]\right) + By[k] = f[k] \qquad k = 0, 1, 2, 3, 4, \ldots$$

or

$$Ay[k+1] + (B - A)\,y[k] = f[k] \qquad k = 0, 1, 2, 3, 4, \ldots$$

Writing $A = a$ and $(B - A) = b$ gives this equation in the format used by ***MathinSite***:

$$ay[k+1] + by[k] = f[k] \qquad k = 0, 1, 2, 3, 4, \ldots \qquad (1.7)$$

Equation (1.7) is a ***linear, first-order <u>difference</u> equation with constant coefficients*** and is the digital equivalent of the analogue ***linear, first-order, <u>differential</u> equation with constant coefficients*** given in Equation (1.5).

Solving First Order Difference Equations

There exist a variety of ways of solving difference equations of this type – three are considered here, using:

- a recursive formula
- the 'classical' complementary function and particular integral approach
- *z*-transforms.

The first method produces an 'open form' type of solution and the second and third methods a 'closed form' type of solution. These will be discussed as they are met.

This Theory Sheet is not intended to give an exhaustive description of how to solve such equations. One equation will be solved by the three methods as an example.

Consider the difference equation

$$4y[k+1] - 2y[k] = 5 \quad \text{with } y[0] = \tfrac{3}{2} \qquad (1.8)$$

(In order to obtain a numerical solution an initial condition is needed - in this example it is specified as $y[0] = 3/2$, i.e. the value of y is zero for $k = 0$.)

Here a = 4, b = -2 and $f[k]$ = 5. It is left as an exercise to the reader to show that the *differential* equation relating to the above *difference* equation is

$$4\frac{dy}{dt}+2y=5, \quad y(0)=\tfrac{3}{2}$$

This analogue equation has the analytical solution, using the classical "assumed solution method" or by using Laplace Transforms

$$y=\tfrac{1}{2}\left(5-2e^{-2t}\right)$$

This solution can be used to compare with the digital solution obtained below.

1. ***Solving the digital system*** (1.8) ***using a recurrence formula***

Equation (1.8) can be rearranged to give the 'recurrence' formula

$$y[k+1]=\tfrac{1}{4}\left(5+2y[k]\right) \text{ with the initial condition } y[0] = \tfrac{3}{2}$$

and this is used recursively, starting with $y[0] = \tfrac{3}{2}$ to obtain the solution sequence

when k = 0: $y[0]=\tfrac{1}{4}\left(5+2\times\tfrac{3}{2}\right)=2$

when k = 1: $y[1]=\tfrac{1}{4}\left(5+2\times 2\right)=\tfrac{9}{4}$

when k = 2: $y[2]=\tfrac{1}{4}\left(5+2\times\tfrac{9}{4}\right)=\tfrac{19}{8}$

when k = 3: $y[3]=\tfrac{1}{4}\left(5+2\times\tfrac{19}{8}\right)=\tfrac{39}{16}$

when k = 4: $y[4]=\tfrac{1}{4}\left(5+2\times\tfrac{39}{16}\right)=\tfrac{79}{16}$

etc

Note the disadvantages with using a recurrence formula
(a) there is no analytical ('nice formula") solution, and
(b) you can only find a value in the sequence by knowing ALL the previous values

This solution is in the form of a sequence of values,

$$\left\{\tfrac{3}{2}, 2, \tfrac{9}{4}, \tfrac{19}{8}, \tfrac{39}{16}, \ldots\right\}$$

and this type of solution is called an ***open-form*** solution.

2. ***Solving the digital system*** (1.8) ***using the 'classical' assumed solution method***

In order to solve (1.8) using this method it is necessary find the 'complementary function' (CF) first by solving the homogeneous equation

$$4y[k+1]-2y[k]=0$$

The solution here is assumed to be of the form $y[k]=Ar^k$ where A is some constant whose value is to be determined. If this *IS* a solution of the homogeneous equation it must *satisfy* the homogeneous equation. So, with $y[k+1]=Ar^{k+1}$,

$$4Ar^{k+1}-2Ar^k=0$$

giving

$$2\times Ar^k\times\left(2r-1\right)=0$$

Here, either

- 2 = 0 (not very likely!) or
- $Ar^k=0$ (this, the ***trivial*** solution, is discounted since it leads to the perfectly acceptable, but incredibly uninteresting, solution $y[k]$ = 0), or
- $2r-1=0$, giving r = ½

So the solution (CF) satisfying the homogeneous equation is $y[k] = A\left(\tfrac{1}{2}\right)^k$ (1.9)

For a 'particular integral' of the full, homogeneous equation given in (1.8) it is necessary to assume that there will be a solution that will be some constant, p, say.

(NOTE: *It is a characteristic of linear difference equations (systems) that whatever the type of input (in this case the constant value* 5*), the output will be of the same type.*)

So if the solution expected is $y[k] = p$ (and hence $y[k + 1] = p$ also) then, from (1.8),

$$4p - 2p = 5 \text{ or } \underline{p = \tfrac{5}{2}} \qquad (1.10)$$

The *full* solution for the system is now found by adding the particular integral (1.10) just found to the CF (1.9) giving the *general* solution

$$y[k] = A\left(\tfrac{1}{2}\right)^k + \tfrac{5}{2}$$

Even now, this is not the final solution since the value of A is still unknown. This is found using the initial condition, $y[0] = \tfrac{3}{2}$, i.e.

$$y[0] = A\left(\tfrac{1}{2}\right)^0 + \tfrac{5}{2}$$

or

$$\tfrac{3}{2} = A + \tfrac{5}{2} \text{ i.e. } A = -1$$

So the *particular* solution is

$$\boxed{y[k] = \tfrac{5}{2} - \left(\tfrac{1}{2}\right)^k}$$

This result should corroborate the result found using the recursive method. Consider just 3 values of k.

when $k = 0$: $y[0] = \tfrac{5}{2} - \left(\tfrac{1}{2}\right)^0 = \tfrac{3}{2}$

when $k = 2$: $y[2] = \tfrac{5}{2} - \left(\tfrac{1}{2}\right)^2 = \tfrac{9}{4}$

when $k = 4$: $y[0] = \tfrac{5}{2} - \left(\tfrac{1}{2}\right)^4 = \tfrac{79}{16}$ etc

Note the *advantages* with using this method
(a) there *is* 'nice formula' solution that allows
(b) you to find a value *anywhere* in the sequence *without* knowing ALL the previous values

The solution here is a ***closed-form*** solution; it finds *any* value in the output sequence without having to evaluate *any* of the previous values.

Before going any further, it should be pointed out that it is not necessarily a case of "open-form bad, closed-form good". Very often in, for example, real-time systems such as computer-controlled robots, the software programmer is more likely to use the open form for two major reasons:

- Counting clock cycles. Real-time software control systems have to be fast. In the above example, the closed form solution involved subtraction and raising the number ½ to a power - a notoriously 'slow' process for a computer when compared with the open-form solution, where it was only necessary to use multiplication, addition and division.
- The controlling software, in the very act of controlling a system, necessarily has to generate all the output values anyway.

3. *Solving the digital system* (1.8) *using z-transforms*

This method is likely to be met in a second-year undergraduate engineering course and is the most sophisticated of the three given here. It requires knowledge of how to use *z*-transforms, which are the digital, discrete-time equivalent of the analogue, continuous-time Laplace Transforms.

In solving (1.8), the following *z*-transforms are required:

- $Z\{y[k]\} = Y(z)$, $Z\{y[k+1]\} = zY(z) - zy[0]$,
- $Z\{A\} = A.\dfrac{z}{z-1}$, $Z\{Ar^k\} = A.\dfrac{z}{z-r}$

where A and r are constants.

So, taking *z*-transforms of (1.8) and using the initial condition straightaway (as opposed to having to leave it until the *end* in the previous method) gives

$$4\left(zY(z) - z\times\tfrac{3}{2}\right) - 2Y(z) = 5.\frac{z}{z-1}$$

or

$$4\left(z-\tfrac{1}{2}\right)Y(z) = 6z + \frac{5z}{z-1} = \frac{6z^2 - z}{z-1} = \frac{z(6z-1)}{z-1}$$

so

$$\frac{Y(z)}{z} = \frac{(6z-1)}{4(z-1)\left(z-\tfrac{1}{2}\right)} = \frac{5}{2}.\frac{1}{z-1} - \frac{1}{\left(z-\tfrac{1}{2}\right)}$$

using partial fractions, so

$$Y(z) = \frac{5}{2}.\frac{z}{z-1} - \frac{z}{\left(z-\tfrac{1}{2}\right)}$$

which, by taking inverse transforms gives, as before,

$$y[k] = \tfrac{5}{2} - \left(\tfrac{1}{2}\right)^k$$

Comparison of Analogue and Digital Solutions

	Analogue	**Digital**
Solution	$y(t) = \tfrac{5}{2} - e^{-2t}$	$y[k] = \tfrac{5}{2} - \left(\tfrac{1}{2}\right)^k$
Initial Condition, $t = kT = 0$	$y(0) = \tfrac{3}{2}$	$y[k] = \tfrac{3}{2}$
Solution at $t = kT = 0.5$	2.132121	1.792893
Solution at $t = kT = 1$	2.364665	2
Solution at $t = kT = 2$	2.481684	2.25
Solution at $t = kT = 5$	2.499955	2.46875
Steady State value	$\tfrac{5}{2}$	$\tfrac{5}{2}$
Type of Solution	Monotonic*, exponential growth to a limit	Monotonic, exponential growth to a limit

**Monotonic* means that the solution approaches the steady state value purely from one side (in this case from $\frac{3}{2}$ up to $\frac{5}{2}$), i.e. the solution does not oscillate about the steady state value as it is approached.

It is impossible to generalise from a single example. However, the above table does indicate some interesting points.

- Both solutions are effectively the same equation with e^{-2} in the analogue solution being approximated by ½ in the digital – already not a very good approximation
- An initial condition, $\frac{3}{2}$, and a steady state, $\frac{5}{2}$, is achieved by both, however
- The digital solution takes longer to reach steady state (for example, the digital solution at $t = kT = 5$ is about the same as the analogue solution at $t = kT = 1$)
- The error between analogue and digital at $t = kT = 0.5$ is approximately 16% but by $t = kT = 5$ this has dropped to about 1.2%. During the early part of the transient response, the digital approximation is worse than when steady state is approached.

Exercises

1. Will the digital solution always be 'similar' to the analogue solution? Will a monotonic analogue solution always result in a monotonic digital solution? Try the same analysis with these (or use the applet to do the investigation for you!).

- ❑ $5y[k+1]+3y[k]=2$
- ❑ $5y[k+1]-3y[k]=2$
- ❑ $3y[k+1]-3y[k]=2$
- ❑ $3y[k+1]+3y[k]=2$

noting in particular how the position of the system pole* in the *z*-plane relates to the type of solution obtained.

*The system pole for a first order system always lies on the real axis in the complex plane (Argand Diagram). Note that such a pole can either lie

- ❑ totally *within* the unit circle
- ❑ *on* the unit circle, or
- ❑ *outside* the unit circle

The type of response of the system (i.e. the type of solution of the difference equation) can be gauged from this pole position. Think in terms of stable, unstable, monotonic, oscillatory, constant, purely oscillatory, as the pole moves along the real axis.

2. Classify (perhaps in a table) how the system responds for particular ranges of values of the pole position along the real axis.