3

Block diagrams and operators: Two new representations

The goals of this chapter are:

- to introduce two representations for discrete-time systems: block diagrams and operators;
- to introduce the whole-signal abstraction and to exhort you to use abstraction;
- to start manipulating operator expressions;
- to compare operator with difference-equation and block-diagram manipulations.

The preceding chapters explained the verbal-description and difference-equation representations. This chapter continues the theme of multiple representations by introducing two new representations: block diagrams and operators. New representations are valuable because they suggest new thoughts and often provide new insight; an expert engineer values her representations the way an expert carpenter values her tools. This chapter first introduces block diagrams, discusses the whole-signal abstraction and the general value of abstraction, then introduces the operator representation.

3.1 Disadvantages of difference equations

Chapter 2 illustrated the virtues of difference equations. When compared to the verbal description from which they originate, difference equations are compact, easy to analyze, and suited to computer implementation. Yet analyzing difference equations often involves chains of micro-manipulations from which insight is hard to find. As an example, show that the difference equation

$$d[n] = a[n] - 3a[n-1] + 3a[n-2] - a[n-3]$$

is equivalent to this set of equations:

$$\begin{aligned} d[n] &= c[n] - c[n-1] \\ c[n] &= b[n] - b[n-1] \\ b[n] &= a[n] - a[n-1]. \end{aligned}$$

As the first step, use the last equation to eliminate $b[n]$ and $b[n-1]$ from the $c[n]$ equation:

$$c[n] = \underbrace{(a[n] - a[n-1])}_{b[n]} - \underbrace{(a[n-1] - a[n-2])}_{b[n-1]} = a[n]-2a[n-1]+a[n-2].$$

Use that result to eliminate $c[n]$ and $c[n-1]$ from the $d[n]$ equation:

$$\begin{aligned} d[n] &= \underbrace{(a[n] - 2a[n-1] + a[n-2])}_{c[n]} - \underbrace{(a[n-1] - 2a[n-2] + a[n-3])}_{c[n-1]} \\ &= a[n] - 3a[n-1] + 3a[n-2] - a[n-3]. \end{aligned}$$

Voilà: The three-equation system is equivalent to the single difference equation. But what a mess. Each step is plausible yet the chain of steps seems random. If the last step had produced

$$d[n] = a[n] - 2a[n-1] + 2a[n-2] - a[n-3],$$

it would not immediately look wrong. We would like a representation where it would look wrong, perhaps not immediately but at least quickly. Block diagrams are one such representation.

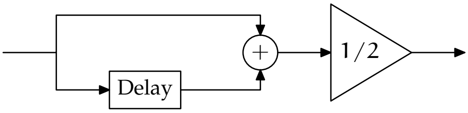
Exercise 12. Although this section pointed out a disadvantage of difference equations, it is also important to appreciate their virtues. Therefore, invent a verbal description (a story) to represent the single equation

$$d[n] = a[n] - 3a[n-1] + 3a[n-2] - a[n-3]$$

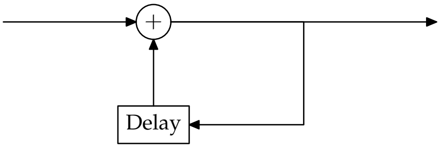
and then a verbal description to represent the equivalent set of three equations. Now have fun showing, without converting to difference equations, that these descriptions are equivalent!

3.2 Block diagrams to the rescue

Block diagrams visually represent a system. To show how they work, here are a few difference equations with corresponding block diagrams:



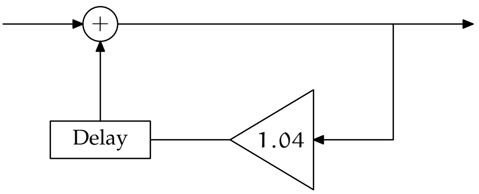
$y[n] = (x[n] + x[n-1])/2$
averaging filter



$y[n] = y[n-1] + x[n]$
account with 0% interest

Pause to try 13. Draw the block diagram for the endowment account from Section 2.2.

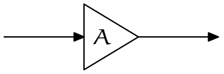
The endowment account is a bank account that pays 4% interest, so it needs a gain element in the loop, with gain equal to 1.04. The diagram is not unique. You can place the gain element before or after the delay. Here is one choice:



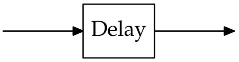
$$y[n] = 1.04\,y[n-1] + x[n]$$

endowment account from Section 2.2

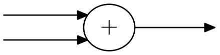
Amazingly, all systems in this course can be built from only two actions and one combinator:



action 1: multiply by α



action 2: delay one tick



combinator: add inputs

3.2.1 Block diagram for the Fibonacci system

To practice block diagrams, we translate (represent) the Fibonacci system into a block diagram.

Pause to try 14. Represent the Fibonacci system of Section 1.1 using a block diagram.

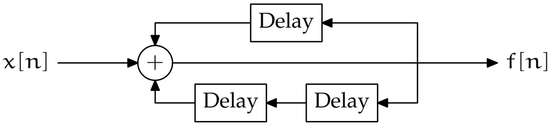
We could translate Fibonacci's description (Section 1.1) directly into a block diagram, but we worked so hard translating the description into a difference equation that we start there. Its difference equation is

$$f[n] = f[n-1] + f[n-2] + x[n],$$

where the input signal $x[n]$ is how many pairs of child rabbits enter the system at month n, and the output signal $f[n]$ is how many pairs of rabbits are in the system at month n. In the block diagram, it is convenient to let input signals flow in from the left and to let output signals exit at the right – following the left-to-right reading common to many languages.

Exercise 13. Do signals-and-systems textbooks in Hebrew or Arabic, which are written right to left, put input signals on the right and output signals on the left?

The Fibonacci system combines the input sample, the previous output sample, and the second-previous output sample. These three signals are therefore inputs to the plus element. The previous output sample is produced using a delay element to store samples for one time tick (one month) before sending them onward. The second-previous output sample is produced by using two delay elements in series. So the block diagram of the Fibonacci system is



3.2.2 Showing equivalence using block diagrams

We introduced block diagrams in the hope of finding insight not easily visible from difference equations. So use block diagrams to redo the proof that

$$d[n] = a[n] - 3a[n-1] + 3a[n-2] - a[n-3]$$

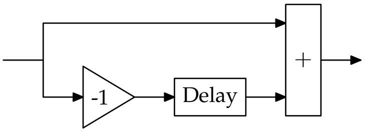
is equivalent to

$$\begin{aligned} d[n] &= c[n] - c[n-1], \\ c[n] &= b[n] - b[n-1], \\ b[n] &= a[n] - a[n-1]. \end{aligned}$$

The system of equations is a cascade of three equations with the structure

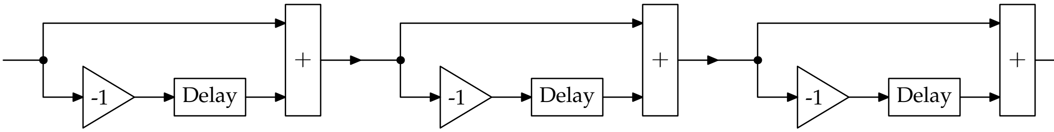
$$\text{output} = \text{this input} - \text{previous input}.$$

The block diagram for that structure is



where the gain of -1 produces the subtraction.

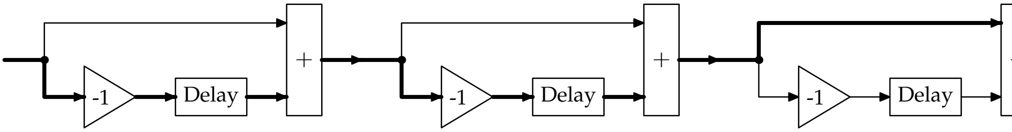
The cascade of three such structures has the block diagram



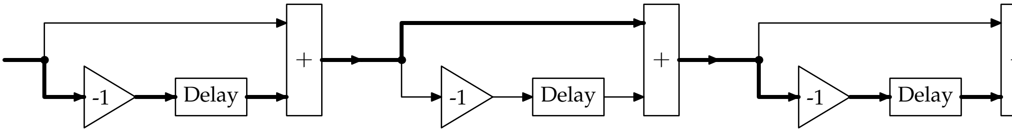
This diagram has advantages compared to the set of difference equations. First, the diagram helps us describe the system compactly. Each stage in the cascade is structurally identical, and the structural identity is apparent by looking at it. Whereas in the difference-equation representation, the common structure of the three equations is hidden by the varying signal names. Each stage, it turns out, is a discrete-time differentiator, the simplest discrete-time analog of a continuous-time differentiator. So the block diagram makes apparent that the cascade is a discrete-time triple differentiator.

Second, the block diagram helps rewrite the system, which we need to do to show that it is identical to the single difference equation. So follow a signal through the cascade. The signal reaches a fork three times (marked with a dot), and each fork offers a choice of the bottom or top branch. Three two-way branches means 2^3 or 8 paths through the system. Let's examine a few of them. Three paths accumulate two delays:

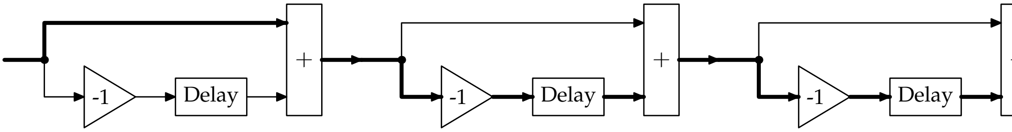
1. low road, low road, high road:



2. low road, high road, low road:



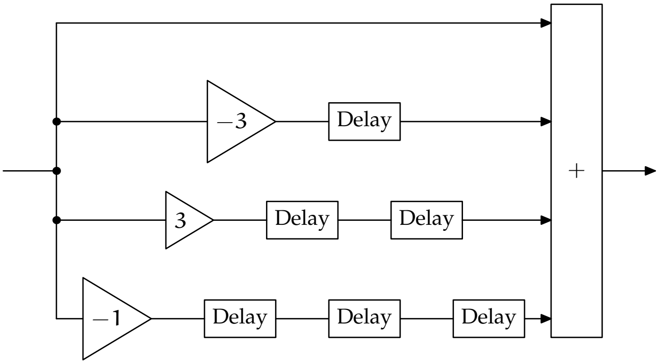
3. high road, low road, low road:



Besides the two delays, each path accumulates two gains of -1, making a gain of 1. So the sum of the three paths is a gain of 3 and a double delay.

Exercise 14. Show the other five paths are: three paths with a single delay and a gain of -1, one path with three delays and a gain of -1, and one path that goes straight through (no gain, no delay).

A block diagram representing those four groups of paths is



The single difference equation

$$d[n] = a[n] - 3a[n-1] + 3a[n-2] - a[n-3].$$

also has this block diagram.

The pictorial approach is an advantage of block diagrams because humans are sensory beings and vision is an important sense. Brains, over hundreds of millions of years of evolution, have developed extensive hardware to process sensory information. However, analytical reasoning and symbol manipulation originate with language, skill perhaps 100,000 years old, so our brains have much less powerful hardware in those domains.

Not surprisingly, computers are far more skilled than are humans at analytical tasks like symbolic algebra and integration, and humans are far more skilled than are computers at perceptual tasks like recognizing faces or speech. When you solve problems, amplify your intelligence with a visual representation such as block diagrams.

On the other side, except by tracing and counting paths, we do not know to manipulate block diagrams; whereas analytic representations lend themselves to transformation, an important property when redesigning systems. So we need a grammar for block diagrams. To find the rules of this grammar, we introduce a new representation for systems, the operator representation. This representation requires the whole-signal abstraction in which all samples of a signal combine into one signal. It is a subtle change of perspective, so we first discuss the value of abstraction in general, then return to the abstraction.

3.3 The power of abstraction

Abstraction is a great tools of human thought. All language is built on it: When you use a word, you invoke an abstraction. The word, even an ordinary noun, stands for a rich, subtle, complex idea. Take *cow* and try to program a computer to distinguish cows from non-cows; then you find how difficult abstraction is. Or watch a child's ability with language develop until she learns that 'red' is not a property of a particular object but is an abstract property of objects. No one knows how the mind manages these amazing feats, nor – in what amounts to the same ignorance – can anyone teach them to a computer.

Abstraction is so subtle that even Einstein once missed its value. Einstein formulated the theory of special relativity [7] with space and time as separate concepts that mingle in the Lorentz transformation. Two years later, the mathematician Hermann Minkowski joined the two ideas into the spacetime abstraction:

> The views of space and time which I wish to lay before you have sprung from the soil of experimental physics, and therein lies their strength. They are radical. Henceforth space by itself, and time by itself, are doomed to fade away into mere shadows, and only a kind of union of the two will preserve an independent reality.

See the English translation in [11] or the wonderful textbook *Spacetime Physics* [1], whose first author recently retired from the MIT physics department. Einstein thought that spacetime was a preposterous invention of mathematicians with time to kill. Einstein made a mistake. It is perhaps the fundamental abstraction of modern physics. The moral is that abstraction is powerful but subtle.

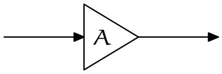
Exercise 15. Find a few abstractions in chemistry, biology, physics, and programming.

If we lack Einstein's physical insight, we ought not to compound the absence with his mistake. So look for and create abstractions. For example, in a program, factor out common code into a procedure and encapsulate common operations into a class. In general, organize knowledge into abstractions or chunks [15].

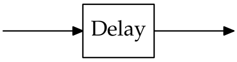
3.4 Operations on whole signals

For signals and systems, the whole-signal abstraction increases our ability to analyze and build systems. The abstraction is take all samples of a signal and lump them together, operating on the entire signal at once and as one object. We have not been thinking that way because most of our representations hinder this view. Verbal descriptions and difference equations usually imply a sample-by-sample analysis. For example, for the Fibonacci recurrence in Section 2.3.2, we found the zeroth sample $f[0]$, used $f[0]$ to find $f[1]$, used $f[0]$ and $f[1]$ to find $f[2]$, found a few more samples, then got tired and asked a computer to carry on.

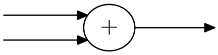
Block diagrams, the third representation, seem to imply a sample-by-sample analysis because the delay element holds on to samples, spitting out the sample after one time tick. But block diagrams live in both worlds and can also represent operations on whole signals. Just reinterpret the elements in the whole-signal view, as follows:



action 1: multiply whole signal by α



action 2: shift whole signal right one tick



combinator: add whole signals

To benefit from the abstraction, compactly represent the preceding three elements. When a signal is a single object, the gain element acts like ordinary multiplication, and the plus element acts like addition of numbers. If the delay element could also act like an arithmetic operation, then all three elements would act in a familiar way, and block diagrams could be manipulated using the ordinary rules of algebra. In order to bring the delay element into this familiar framework, we introduce the operator representation.

3.4.1 Operator representation

In operator notation, the symbol $\mathcal{R}$ stands for the right-shift operator. It takes a signal and shifts it one step to the right. Here is the notation for a system that delays a signal X by one tick to produce a signal Y:

$$Y = \mathcal{R}\{X\}.$$

Now forget the curly braces, to simplify the notation and to strengthen the parallel with ordinary multiplication. The clean notation is

$$Y = \mathcal{R}X.$$

Pause to try 15. Convince yourself that right-shift operator $\mathcal{R}$, rather than the left-shift operator $\mathcal{L}$, is equivalent to a delay.

Let's test the effect of applying $\mathcal{R}$ to the fundamental signal, the impulse. The impulse is

$$I = 1, 0, 0, 0, \ldots$$

Applying $\mathcal{R}$ to it gives

$$\mathcal{R}I = 0, 1, 0, 0, \ldots$$

which is also the result of delaying the signal by one time tick. So $\mathcal{R}$ rather than $\mathcal{L}$ represents the delay operation. In operator notation, the block-diagram elements are:

α	action 1 (gain)	multiply whole signal by α
$\mathcal{R}$	action 2 (delay)	shift whole signal right one tick
$+$	combinator	add whole signals

3.4.2 Using operators to rewrite difference equations

Let's try operator notation on the first example of the chapter: rewriting the single difference equation

$$d[n] = a[n] - 3a[n-1] + 3a[n-2] - a[n-3]$$

into the system of three difference equations

$$\begin{aligned} d[n] &= c[n] - c[n-1], \\ c[n] &= b[n] - b[n-1], \\ b[n] &= a[n] - a[n-1]. \end{aligned}$$

To turn the sample-by-sample notation into whole-signal notation, turn the left side of the long equation into the whole signal D, composed of the samples $d[0], d[1], d[2], \ldots$. Turn the samples on the right side into whole signals as follows:

$$\begin{aligned} a[n] &\to A, \\ a[n-1] &\to \mathcal{R}A, \\ a[n-2] &\to \mathcal{R}\mathcal{R}A, \\ a[n-3] &\to \mathcal{R}\mathcal{R}\mathcal{R}A. \end{aligned}$$

Now import compact notation from algebra: If $\mathcal{R}$ acts like a variable or number then $\mathcal{R}\mathcal{R}$ can be written $\mathcal{R}^2$. Using exponent notation, the translations are:

$$\begin{aligned} a[n] &\to A, \\ a[n-1] &\to \mathcal{R}A, \\ a[n-2] &\to \mathcal{R}^2A, \\ a[n-3] &\to \mathcal{R}^3A. \end{aligned}$$

With these mappings, the difference equation turns into the compact form

$$D = (1 - 3\mathcal{R} + 3\mathcal{R}^2 - \mathcal{R}^3)A.$$

To show that this form is equivalent to the system of three difference equations, translate them into an operator expression connecting the input signal A and the output signal D.

Pause to try 16. What are the operator versions of the three difference equations?

The system of equations turns into these operator expressions

$$\begin{aligned} d[n] &= c[n] - c[n-1] &\rightarrow\quad D &= (1-\mathcal{R})C,\\ c[n] &= b[n] - b[n-1] &\rightarrow\quad C &= (1-\mathcal{R})B,\\ b[n] &= a[n] - a[n-1] &\rightarrow\quad B &= (1-\mathcal{R})A. \end{aligned}$$

Eliminate B and C to get

$$D = (1-\mathcal{R})(1-\mathcal{R})(1-\mathcal{R})A = (1-\mathcal{R})^3 A.$$

Expanding the product gives

$$D = (1 - 3\mathcal{R} + 3\mathcal{R}^2 - \mathcal{R}^3)A,$$

which matches the operator expression corresponding to the single difference equation. The operator derivation of the equivalence is simpler than the block-diagram rewriting, and much simpler than the difference-equation manipulation.

Now extend the abstraction by dividing out the input signal:

$$\frac{D}{A} = 1 - 3\mathcal{R} + 3\mathcal{R}^2 - \mathcal{R}^3.$$

The operator expression on the right, being independent of the input and output signals, is a characteristic of the system alone and is called the **system functional**.

The moral of the example is that operators help you efficiently analyze systems. They provide a grammar for combining, for subdividing, and in general for rewriting systems. It is a familiar grammar, the grammar of algebraic expressions. Let's see how extensively operators follow these. In the next section we stretch the analogy and find that it does not break.

Exercise 16. What is the result of applying $1 - \mathcal{R}$ to the signal $1, 2, 3, 4, 5, \ldots$?

Exercise 17. What is the result of applying $(1 - \mathcal{R})^2$ to the signal $1, 4, 9, 16, 25, 36, \ldots$?

3.5 Feedback connections

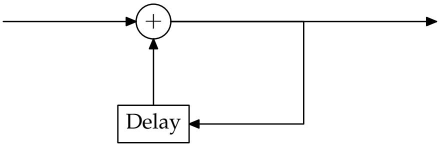
The system with $(1 - \mathcal{R})^3$ as its system functional used only feedforward connections: The output could be computed directly from a fixed number of inputs. However, many systems – such as Fibonacci or bank accounts – contain feedback, where the output depends on previous values of the output. Feedback produces new kinds of system functionals. Let's test whether they also obey the rules of algebra.

3.5.1 Accumulator

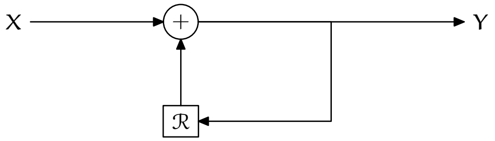
Here is the difference equation for the simplest feedback system, an **accumulator**:

$$y[n] = y[n-1] + x[n].$$

It is a bank account that pays no interest. The output signal (the balance) is the sum of the inputs (the deposits, whether positive or negative) up to and including that time. The system has this block diagram:



Now combine the visual virtues of block diagrams with the compactness and symbolic virtues of operators by using $\mathcal{R}$ instead of 'Delay'. The operator block diagram is



Pause to try 17. What is its system functional?

Either from this diagram or from the difference equation, translate into operator notation:

$$Y = \mathcal{R}Y + X.$$

Collect the Y terms on one side, and you find end up with the system functional:

$$\frac{Y}{X} = \frac{1}{1-\mathcal{R}}.$$

It is the reciprocal of the differentiator.

This operator expression is the first to include $\mathcal{R}$ in the denominator. One way to interpret division is to compare the output signal produced by the difference equation with the output signal produced by the system functional $1/(1-\mathcal{R})$. For simplicity, test the equivalence using the impulse

$$I = 1, 0, 0, 0, \ldots$$

as the input signal. So $x[n]$ is 1 for $n = 0$ and is 0 otherwise. Then the difference equation

$$y[n] = y[n-1] + x[n]$$

produces the output signal

$$Y = 1, 1, 1, 1, \ldots.$$

Exercise 18. Check this claim.

The output signal is the discrete-time **step function** θ. Now apply $1/(1-\mathcal{R})$ to the impulse I by importing techniques from algebra or calculus. Use synthetic division, Taylor series, or the binomial theorem to rewrite $1/(1-\mathcal{R})$ as

$$\frac{1}{1-\mathcal{R}} = 1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots.$$

To apply $1/(1-\mathcal{R})$ to the impulse, apply the each of terms 1, $\mathcal{R}$, $\mathcal{R}^2$, . . . to the impulse I:

$$\begin{aligned}
1\mathrm{I} &= 1,0,0,0,0,0,0,\ldots,\\
\mathcal{R}\mathrm{I} &= 0,1,0,0,0,0,0,\ldots,\\
\mathcal{R}^2\mathrm{I} &= 0,0,1,0,0,0,0,\ldots,\\
\mathcal{R}^3\mathrm{I} &= 0,0,0,1,0,0,0,\ldots,\\
\mathcal{R}^4\mathrm{I} &= 0,0,0,0,1,0,0,\ldots,\\
&\ldots
\end{aligned}$$

Add these signals to get the output signal Y.

Pause to try 18. What is Y?

For $n \geqslant 0$, the $y[n]$ sample gets a 1 from the $\mathcal{R}^n\mathrm{I}$ term, and from no other term. So the output signal is all 1's from $n = 0$ onwards. The signal with those samples is the step function:

$$\mathrm{Y} = 1,1,1,1,\ldots.$$

Fortunately, this output signal matches the output signal from running the difference equation. So, for an impulse input signal, these operator expressions are equivalent:

$$\frac{1}{1-\mathcal{R}} \quad \text{and} \quad 1 + \mathcal{R} + \mathcal{R}^2 + \mathcal{R}^3 + \cdots.$$

Exercise 19. If you are mathematically inclined, convince yourself that verifying the equivalence for the impulse is sufficient. In other words, we do not need to try all other input signals.

The moral is that the $\mathcal{R}$ operator follows the rules of algebra and calculus. So have courage: Use operators to find results, and do not worry.

3.5.2 Fibonacci

Taking our own advice, we now analyze the Fibonacci system using operators. The recurrence is:

$$\text{output} = \text{delayed output} + \text{twice-delayed output} + \text{input}.$$

Pause to try 19. Turn this expression into a system functional.

The output signal is F, and the input signal is X. The delayed output is $\mathcal{R}\mathsf{X}$, and the twice-delayed output is $\mathcal{R}\mathcal{R}\mathsf{X}$ or $\mathcal{R}^2\mathsf{X}$. So

$$\mathsf{F} = \mathcal{R}\mathsf{F} + \mathcal{R}^2\mathsf{F} + \mathsf{X}.$$

Collect all F terms on one side:

$$\mathsf{F} - \mathcal{R}\mathsf{F} - \mathcal{R}^2\mathsf{F} = \mathsf{X}.$$

Then factor the F:

$$(1 - \mathcal{R} - \mathcal{R}^2)\mathsf{F} = \mathsf{X}.$$

Then divide both sides by the $\mathcal{R}$ expression:

$$\mathsf{F} = \frac{1}{1 - \mathcal{R} - \mathcal{R}^2}\mathsf{X}.$$

So the system functional is

$$\frac{1}{1 - \mathcal{R} - \mathcal{R}^2}.$$

Exercise 20. Using `maxima` or otherwise, find the Taylor series for $1/(1 - \mathcal{R} - \mathcal{R}^2)$. What do you notice about the coefficients? Coincidence? `maxima` (maxima.sourceforge.net) is a powerful symbolic algebra and calculus system. It descends from `Macsyma`, which was created at MIT in the 1960's. `maxima` is free software (GPL license) and is available for most platforms.

3.6 Summary

Including the two system representations discussed in this chapter, you have four representation for discrete-time systems:

1. verbal descriptions,
2. difference equations,
3. block diagrams, and
4. operator expressions.

In the next chapter, we use the operator representation to decompose, and design systems.