

This kind of decomposition is unique to discrete-time signals, and will prove useful subsequently.

Discrete-time systems can act on discrete-time signals in ways similar to those found in analog signals and systems. Because of the role of software in discrete-time systems, many more diferent systems can be envisioned and "constructed" with programs than can be with analog signals. In fact, a special class of analog signals can be converted into discrete-time signals, processed with software, and converted back into an analog signal, all without the incursion of error. For such signals, systems can be easily produced in software, with equivalent analog realizations difcult, if not impossible, to design.

#### Symbolic-valued Signals

 Available under [Creative Commons-ShareAlike 4.0 International License](http://creativecommons.org/licenses/by-sa/4.0/) ([http://creativecommon](http://creativecommons.org/licenses/by-sa/4.0/) [s.org/licenses/by-sa/4.0/](http://creativecommons.org/licenses/by-sa/4.0/)).

Another interesting aspect of discrete-time signals is that their values do not need to be real numbers. We do have real-valued discrete-time signals like the sinusoid, but we also have signals that denote the sequence of characters typed on the keyboard. Such characters certainly aren't real numbers, and as a collection of possible signal values, they have little mathematical structure other than that they are members of a set. More formally, each element of the symbolic-valued signal *s (n)* takes on one of the values {***a1*,...,*aK***} which comprise the **alphabet A**. This technical terminology does not mean we restrict symbols to being members of the English or Greek alphabet.

They could represent keyboard characters, bytes (8-bit quantities), integers that convey daily temperature. Whether controlled by software or not, discrete-time systems are ultimately constructed from digital circuits, which consist entirely of analog circuit elements. Furthermore, the transmission and reception of discrete-time signals, like e-mail, is accomplished with analog signals and systems. Understanding how discrete-time and analog signals and systems intertwine is perhaps the main goal of this course.

[MEDIA OBJECT](http://www.opentextbooks.org.hk/system/files/resource/9/9648/9678/media/discrete-time-signals-25.pdf) ([http://www.opentextbooks.org.hk/system/files/resource/9/96 8/9678/media/](http://www.opentextbooks.org.hk/system/files/resource/9/9648/9678/media/discrete-time-signals-25.pdf) [discrete-time-signals-25.pdf](http://www.opentextbooks.org.hk/system/files/resource/9/9648/9678/media/discrete-time-signals-25.pdf))

#### Introduction to Systems

 Available under [Creative Commons-ShareAlike 4.0 International License](http://creativecommons.org/licenses/by-sa/4.0/) ([http://creativecommon](http://creativecommons.org/licenses/by-sa/4.0/) [s.org/licenses/by-sa/4.0/](http://creativecommons.org/licenses/by-sa/4.0/)).

**Signals are manipulated by systems.** Mathematically, we represent what a system does by the notation *y (t)= S (x (t)*), with x representing the input signal and *y* the output signal.