Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

Chapter 7:

Discrete Systems

Contents

References

…update…

[1] P.D. Cha, J.I. Molinder, „Fundamentals of Signals and Systems – A Building Block Approach“, Cambridge University Press, 2006.

[2] G.Lekkas, J.Wild, „Signale und System“, ZHW-Vorlesung, 2007.

…introduction…

1. Discrete System Description with Difference Equations

Similarly to the modelling of continuous systems with a differential equation, discrete systems can be described with a difference equation. Before we look into examples let us define which kind of discrete systems we are dealing with.

Figure 2-1 below shows a continuous LTI system on the left, taking as input a continuous signal u(t) and producing an output signal y(t). The discrete system on the right is also linear and time invariant (LTD = LT(I)D : linear time-invariant and discrete system), and takes u[n] discrete signal u[n] as input and produces y[n] as output. The sampling period T_S is assumed to be known and constant for both input and output signals.

Figure 7-1 Representation of LTI and LTD systems with respective in/out-puts

Please find below the formulas explaining the main characteristics of our LTD systems, namely: linearity, time invariance and causality.

Linearity:

$$\begin{aligned} &u_1[n] \leftrightarrow y_1[n] \\ &u_2[n] \leftrightarrow y_2[n] \\ &(a \cdot u_1[n] + b \cdot u_2[n]) \leftrightarrow (a \cdot y_1[n] + b \cdot y_2[n]) \end{aligned} \tag{3}$$

Time Invariance:

$$\begin{aligned} &u[n] \leftrightarrow y[n] \\ &u[n-k] \leftrightarrow y[n-k] \end{aligned} \tag{4}$$

Causality:

$$y[n] \neq 0 \;\; \text{for any} \;\; u[k] \neq 0 \;\; \text{mit} \;\; k \leq n \tag{5}$$

Observation: k and n are integer indexes.

Similarly to the continuous systems we can also draw a graphical representation of the discrete systems with a block diagram, also called signal flow diagram. Here the basic elements used for the block diagram are: delay-block, sum-point and branching point. When a delay block receives a signal x[n] on its input, it generates in the output x[n-1], which corresponds to the value of x[n] in the previous time sample (1 x T_s earlier).

Let us now consider the systems (a) (b) (c) with the following difference equations:

(a) $y[n] = u[n] + u[n-1] + u[n-2] + u[n-3]$

(b) $y[n] = u[n] - u[n-1]$

(c) $y[n] = u[n] - u[n-1] + u[n-2] - u[n-3] + u[n-4] - u[n-5]$

System (a) is an example of a moving average filter, because it calculates the sum of the current sample plus the last 3 input samples. In order to get an output in the same range as the input values, one could scale the sum with a factor ¼ .

Question 7-1

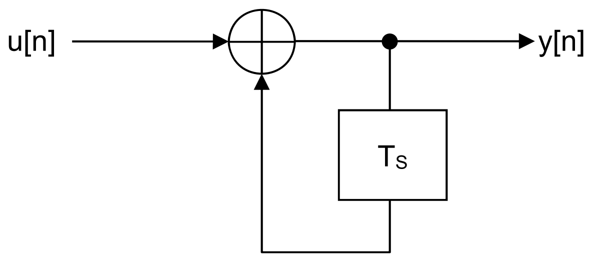
Which effect do you think systems (b) and (c) are having on their inputs? Check the output signals in figures 2-2 and 2-3 and justify your answer.

Question 7-2

Draw the block diagram of systems (a), (b) and (c).

Question 7-3

Determine the difference equation of system (d) with the following signal flow diagram:



Which continuous operation can be approximated with this discrete system?

Question 7-4

Write the difference equation of a moving average filter that calculates the sum of the last 10 input values. Use the sum operator Σ .

Question 7-5

Write the difference equation describing the balance of a saving account that gets 3% interest. You can consider new deposits in the account as the input signal. The sampling rate (when interests are accounted for) equals one year.

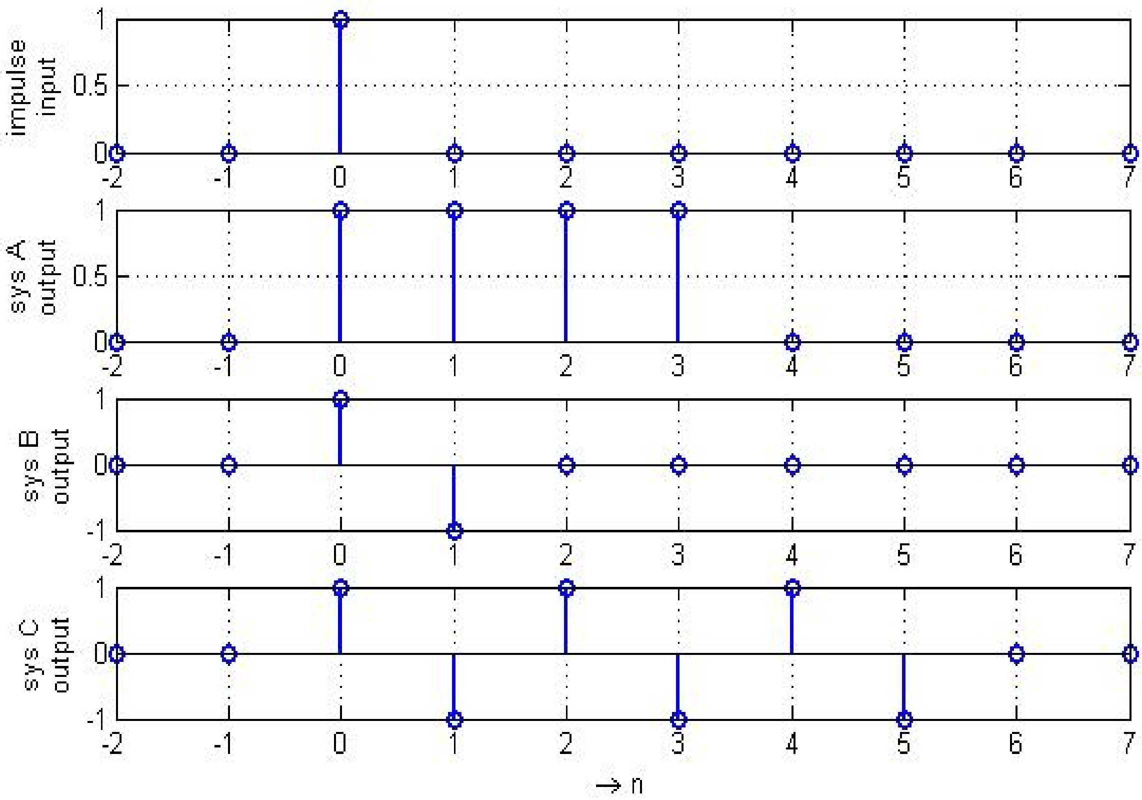


Figure 7-2 Response of discrete systems a, b, c to a unit impulse (Kronecker Delta)

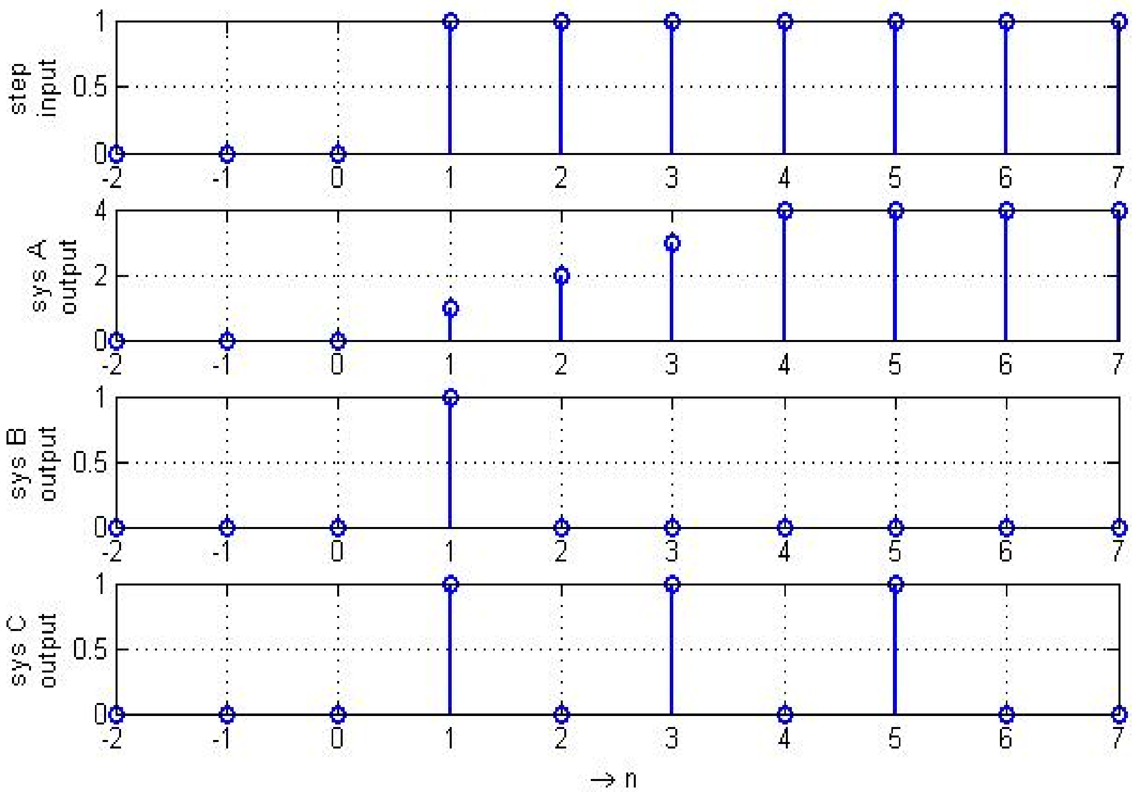


Figure 7-3 Response of discrete systems a, b, c to a unit step

2. System Modelling with Test Signals

2.1. Discrete Systems

The black-box modelling approach can also be deployed for discrete systems (as summarized in figure 2-8 below). An example of impulse and step responses is shown in figures 2-2 and 2-3 in the previous section.

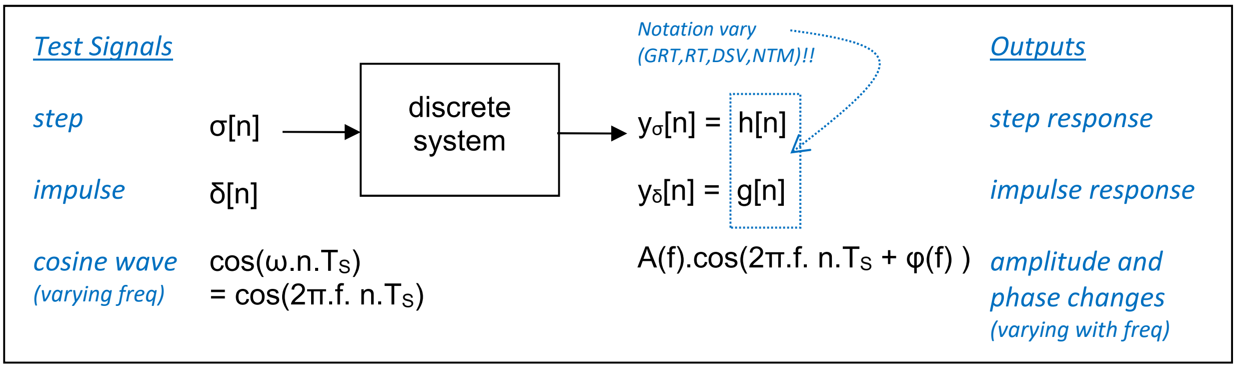


Figure 7-8 Empirical approach to model a black-box discrete system

For discrete systems the easiest and most informative test signal we can apply is a discrete unit impulse. For instance, there are two types of discrete systems that can be recognized by the length of their impulse response (how long non-zero values are produced after the pulse):

- systems with finite impulse responses, called FIR or non-recursive;
- systems with infinite impulse response, called IIR or recursive.

Question 2-9

Determine the impulse response of the discrete system from question 2-3. Based on this impulse response, is this system FIR or IIR? Check the block diagram and discuss with your colleague what could be the reason for the denomination recursive or non-recursive.

Let us suppose now, you have a continuous low pass filter, whose characteristics are well adapted to your application, and you would like to have these same characteristics on a discrete filter that you could implement in a microcontroller. Is this possible? If so, how can we proceed?

Considering that you have a microcontroller available and AD- DA- converters that are fast enough, the answer is yes. Next we discuss a possibility to implement this filter or system based on the impulse response[1].

The first step is to sample the impulse response of the continuous system you wish to match. We call the sample impulse response $g_S[n]$, where the s-subscript refers to the sampling period T_S.

[1] Other possibilities to implement discrete filters matching the characteristics of continuous filters will be learned in later courses (DSV1, using the Z-Transformation).

Now we know that your target discrete system should have an impulse response equal to $g_S[n]$. If you look back at figure 2-2 you will see that the coefficients of the difference equations of systems (a) (b) and (c) match the non-zero values of the impulse response.

But, how does this work? And how can I be sure that getting this impulse response will give the desired filter characteristics?

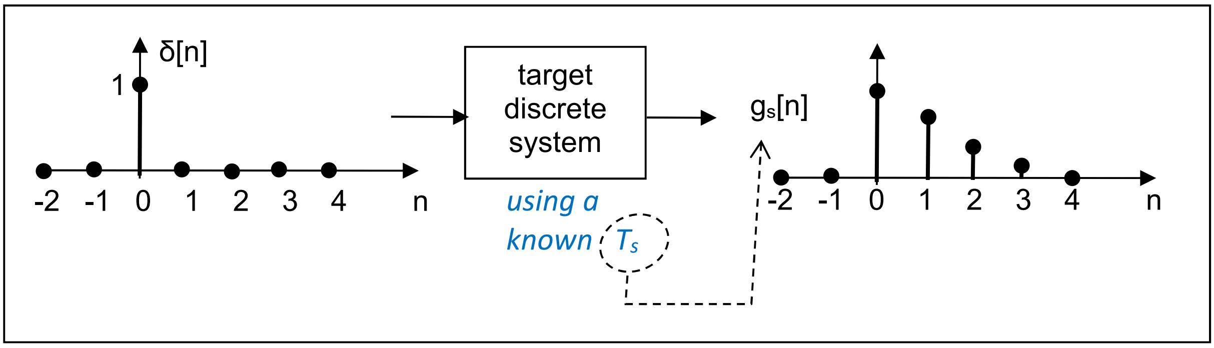


Figure 7-9 Target discrete system with a desired impulse response

In order to answer that, we need to recall some points from chapter 1:
- the discrete systems we deal with are linear and time-invariant (LTD);
- every discrete signal can be described as a sum of weighted and shifted delta impulses.

Therefore we can describe any arbitrary input signal as a sum of weighted and shifted delta impulses, and each of these impulses causes a weighted and shifted impulse response at the system output such that the system output is the sum of all these impulse responses. Figure 2-10 shows an example of this idea.

Now, is there a mathematical operation that carries on this sequence of operations: "sum-of-weighted-and-shifted-responses"? In fact yes, it is called convolution, and we shall discuss it in the next section.

3. Convolution with Impulse Response

The formula we used to calculate the system output signal in figure 2-10 is:

discrete convolution: $$y[n] = \sum_{k=-N_{\max}}^{n} u[k] \cdot g_s[n-k] \qquad (6)$$

In figure 2-10 we followed the integer variable k, giving the index of the input signal u[k] and checked for the non-zero input values which had already caused an impulse response at time n, and summed all contributions together (of the delayed and weighted impulse responses).

Now if we look at equation (6) from the point of view of the integer variable n, and take for example $n = N_1$ (a constant and positive integer value), then $y[N_1]$ corresponds to the sum of the product of two functions evolving with index k:
- u[k] the input signal, and
- $g_s[N_1-k]$ which corresponds to the impulse response mirrored and shifted.

Figure 7-11 represents this second point of view for the same functions used in figure 2-10.

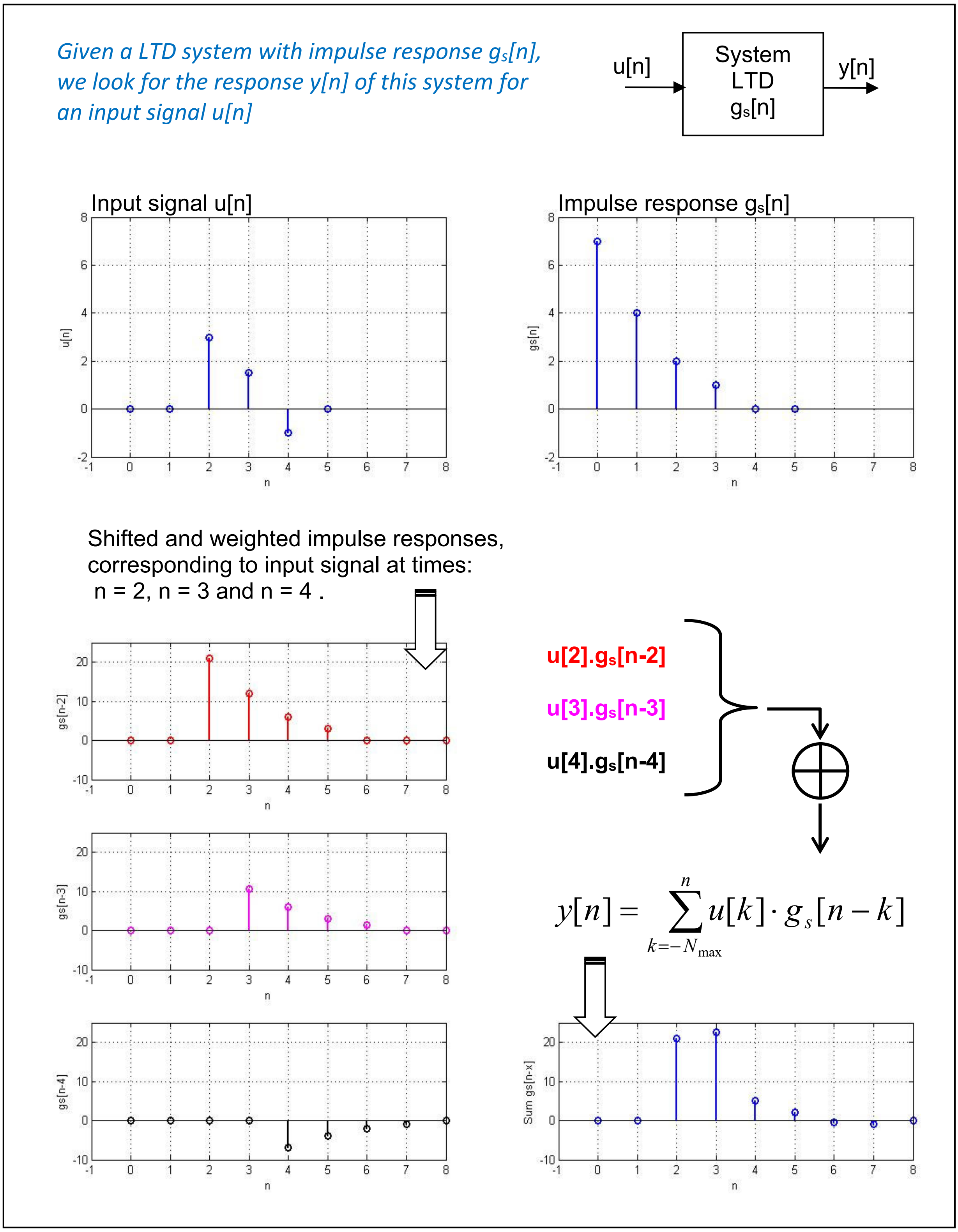


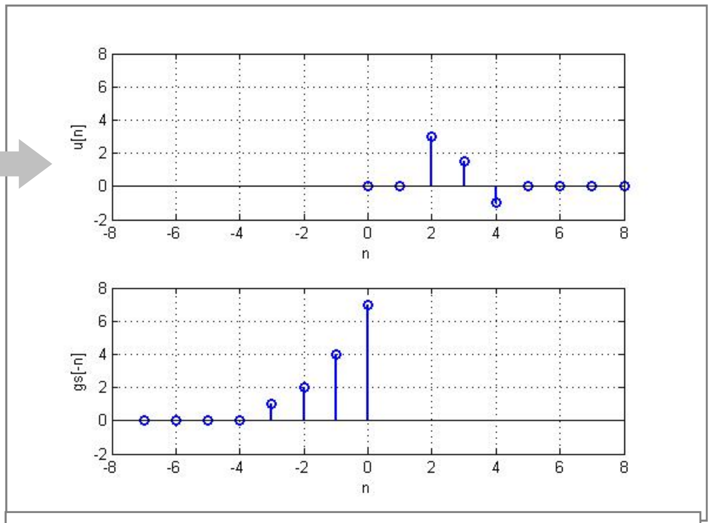
Figure 7-10 **Calculating the output of discrete system through sum of shifted and weighted impulse responses**

Discrete convolution:

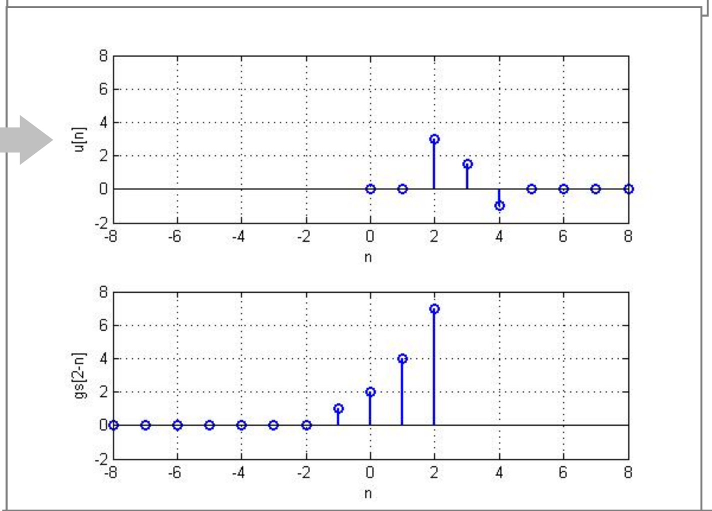
$$y[n] = \sum_{k=-N_{\max}}^{n} u[k] \cdot g_s[n-k]$$

For n = 0

$$y[0] = \sum_{k=-N_{\max}}^{n} u[k] \cdot g_s[0-k]$$



$$y[2] = \sum_{k=-N_{\max}}^{n} u[k] \cdot g_s[2-k]$$



$$y[3] = \sum_{k=-N_{\max}}^{n} u[k] \cdot g_s[3-k]$$

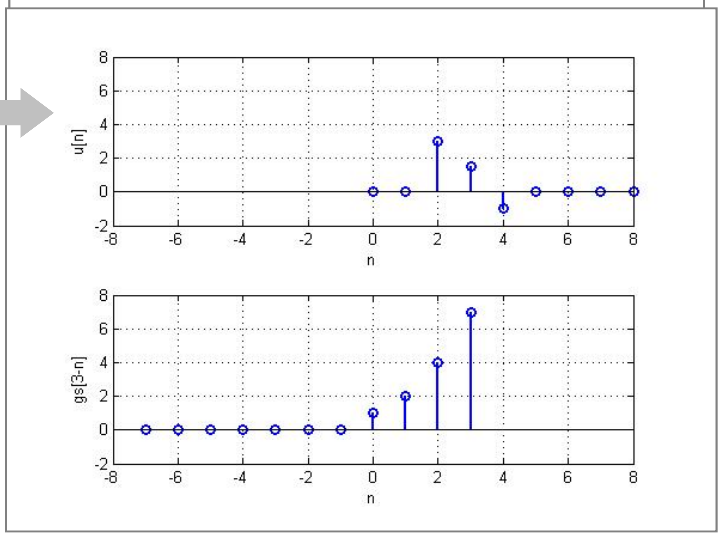


Figure 7-11 Discrete Convolution
2nd point of view: product of 1-original and 1-mirrowed-shifted function

The limits for the index in the sum sign for the index variable k are from $-N_{max}$ to n. Where do these come from? In fact the general definition of a discrete convolution would have as limits for k from $-\infty$ to $+\infty$. Here we can decrease them because:

- for a causal system the impulse response g[k] starts at k=0, and the flipped-shifted impulse response g[n-k] stops therefore at n;
- if the impulse response g[k] has a limited length of N_{max} non-zero values, then the flipped-shifted impulse response g[n-k] starts the earliest at $-N_{max}$ (in fact $-(N_{max} + n)$);
- furthermore if u[k]≠0 only for k≥0, then the limits of the sum can be from 0 up to n.

The convolution operation is in fact commutative, such that:

Discrete Convolution – commutative property

$$y[n]=\sum_{k=-\infty}^{+\infty}u[k]\cdot x[n-k]=\sum_{k=-\infty}^{+\infty}x[k]\cdot u[n-k]=x[n]*u[n]=u[n]*x[n] \qquad (7)$$

The star symbol is used to express the convolution operation in a more compact form.

Next we want to expand the concept of the convolution and the calculation of the system response by doing convolution with the impulse response to the continuous systems (LTI).

Convolution – definition and commutative property

$$y(t)=\int_{-\infty}^{+\infty}u(\lambda)\cdot x(t-\lambda)d\lambda=\int_{-\infty}^{+\infty}x(\lambda)\cdot u(t-\lambda)d\lambda=x(t)*u(t)=u(t)*x(t) \qquad (8)$$

The definition of the convolution operation for two continuous functions is given in equation (8) and figure 2-12 shows a comparison of the convolution with the impulse response for a continuous and a discrete system.

On the left we see a continuous function u(t) as input signal for a continuous LTI system with impulse response g(t). A possible approximation for u(t) is a discrete signal u[n], and an approximation for the integral of the convolution is the sum multiplied by the sample time T_s. When we want to have numerically comparable results for a continuous and discrete convolution, remember to take into account the factor T_s .

In conclusion, the convolution with the impulse response is a method to calculate the output signal for both continuous[2] and discrete systems.

Have a look at the following links and demos for images and animations showing the convolution operation of continuous and discrete functions:

- discrete:
http://lmb.informatik.uni-freiburg.de/lectures/old_lmb/bildverarbeitung/Faltung/disfaltung.html
Matlab demo: dconvdemo-v307 [3]

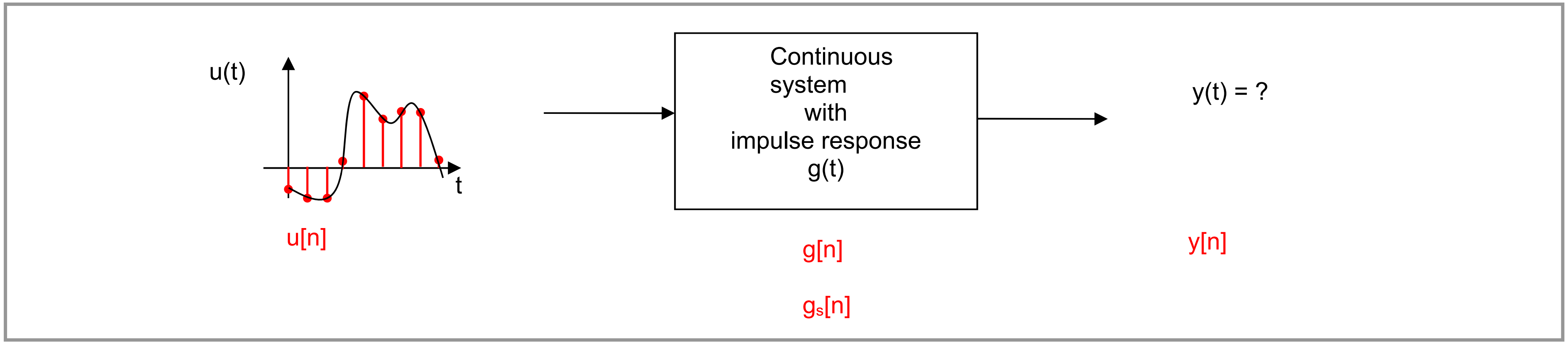
- continuous:
http://lmb.informatik.uni-freiburg.de/lectures/old_lmb/bildverarbeitung/Faltung/convolution_demo.htm
Matlab demo: cconvdemo-v212

We have exercises on this topic in list number 3.

---

[2] Another way to get to this result for the continuous systems would be to consider approximating the input signal by a sum of step signals shifted in time and weighted by a factor reflecting the change in amplitude of the input signal.

[3] Source of both demo codes: http://users.ece.gatech.edu/mcclella/matlabGUIs/



Continuous	Discrete
$y(t)=\int\limits_{-\infty}^{+\infty} u(\lambda)\cdot g(t-\lambda)\,d\lambda$	$y[n]=\sum\limits_{k=-\infty}^{+\infty} u[k]\cdot g[n-k]\cdot T_s = \sum\limits_{k=-\infty}^{+\infty} u[k]\cdot g_s[n-k]$
System causal	System causal
$y(t)=\int\limits_{-\infty}^{t} u(\lambda)\cdot g(t-\lambda)\,d\lambda$	$y[n]=\sum\limits_{k=-\infty}^{n} u[k]\cdot g_s[n-k]$
System stable → limited impulse response $\lim\limits_{t\to\infty} g(t)=0$	System stable → limited impulse response $\lim\limits_{n\to\infty} g[n]=0$
$g(\lambda_{max})\approx 0$	$g[N_{max}]\approx 0$
and if input signal "starts" at 0 : $u(t)\neq 0$ for $t\geq 0$	and if input signal "starts" at 0 : $u[n]\neq 0$ for $n\geq 0$
$y(t)=\int\limits_{0}^{t} u(\lambda)\cdot g(t-\lambda)\,d\lambda$	$y[n]=\sum\limits_{k=0}^{n} u[k]\cdot g_s[n-k]$

Figure 7-12 Comparing continuous and discrete convolution of input signal with impulse response

4. Vocabulary

convolution:	Faltung
differential equation:	Differentialgleichung (DGl)
difference equation:	Differenzgleichung (DzGl)
interests (for account):	Zinsen
saving account:	Sparkonto
signal flow diagram:	Signalflussdiagramm

ABBREVIATIONS:

LTD = LT(I)D : linear time-invariant and discrete system
LTI : linear time-invariant continuous system

5. Annexes

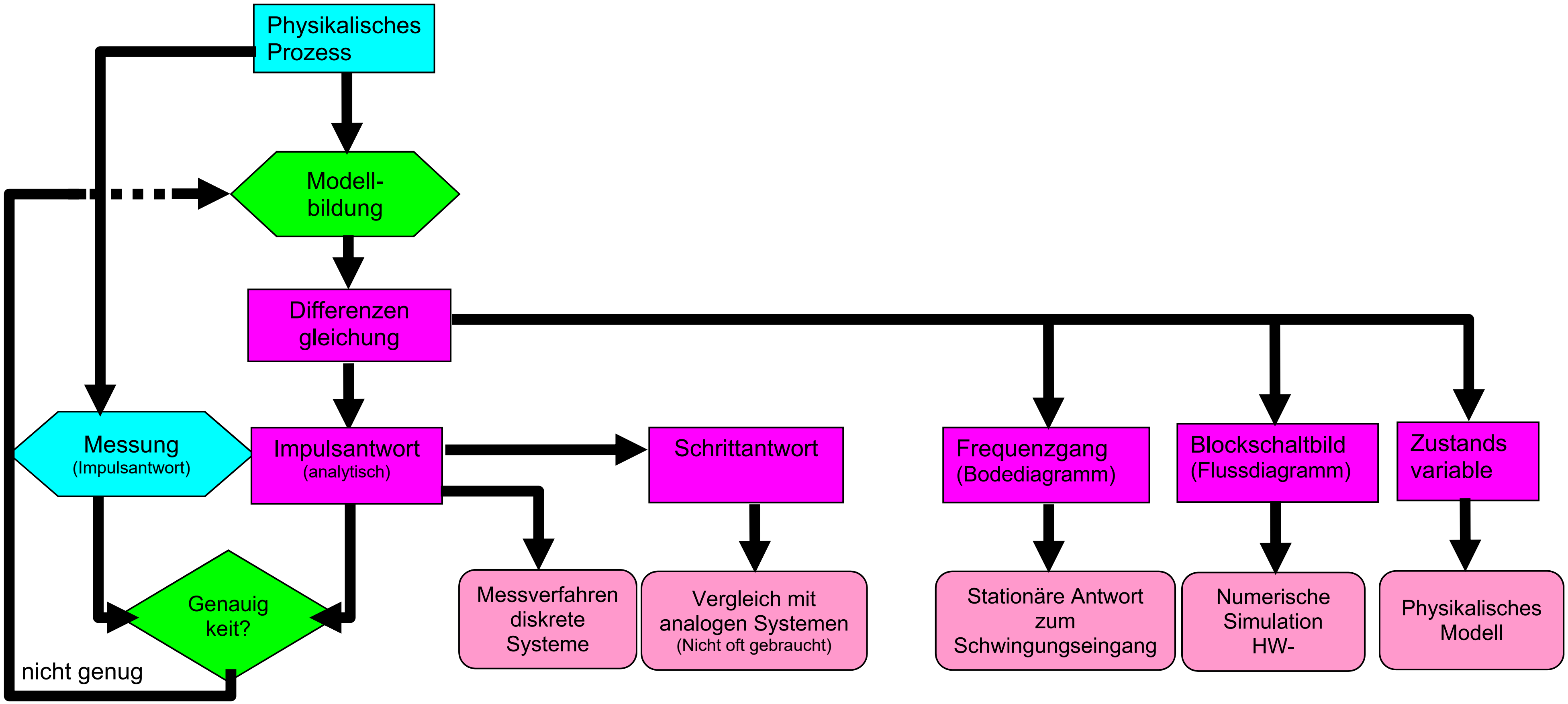
(7-A) Overview of LTD system representations

(7-B) Alternative Convolution representation with Flip-Flops

(7-C) Alternative Convolution representation with Cross-Table

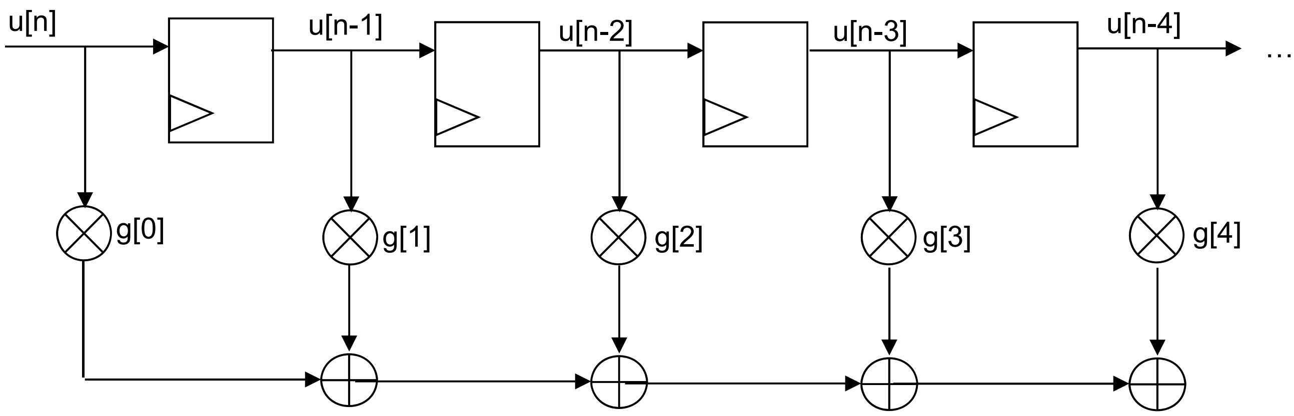
(7-A) Overview of LTD system representations (*Überblick LTD Beschreibungen)*

Die Abbildung unten gibt einen Überblick von Darstellungsarten für lineare, zeitinvariante, und **diskrete Systeme** (auf Englisch *linear time invariant and discrete LTD*).



(7-B) Alternative Convolution representation with Flip-Flops

% Faltung 3rd view : Delayline ('shiftregister') + weighting coefficients



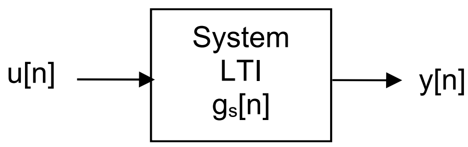

```
gs_n = [7 4 2 1 0 0];
```

u[n]	u[0].g_s[n-0]	u[1].g_s[n-1]	u[2].g_s[n-2]	u[3].g_s[n-3]	u[4].g_s[n-4]	y[n]
0	0					0
0	0	0				0
3	0	0	21			21
1.5	0	0	12	10.5		22.5
-1	0	0	6	6	-7	5
0	0	0	3	3	-4	2
		0	0	1.5	-2	-0.5
			0	0	-1	-1
				0	0	0
					0	0

$$y[n] = \sum_{k=-N_{\max}}^{n} u[k] \cdot g_s[n-k] = \sum_{k=-N_{\max}}^{n} u[n-k] \cdot g_s[k]$$

(7-C) Alternative Convolution representation with Cross-Table

% Faltung 4th view : Diagonal of product matrix



y[0] **y[1]** **y[2]** **y[3]** **y[4]**

y[n] → u[n] ↓	**7**	**4**	**2**	**1**	**0**	**0**
0	0	0	0	0	0	0
0	0	0	0	0	0	0
3	21	12	6	3	0	0
1.5	10.5	6	3	1.5	0	0
-1	-7	-4	-2	-1	0	0

$$y[n] = \sum_{k=-N_{\max}}^{n} u[k] \cdot g_s[n-k] = \sum_{k=-N_{\max}}^{n} u[n-k] \cdot g_s[k]$$