McGill

ECSE 421 Lecture 8: Embedded Hardware: Sensors

ESD Chapter 3

© Peter Marwedel, Brett H. Meyer

Last Time

- Von Neumann MoC
  - Advantages?
  - Disadvantages?
- Comparing different MoCs
- Combining different MoCs
- Modeling Levels

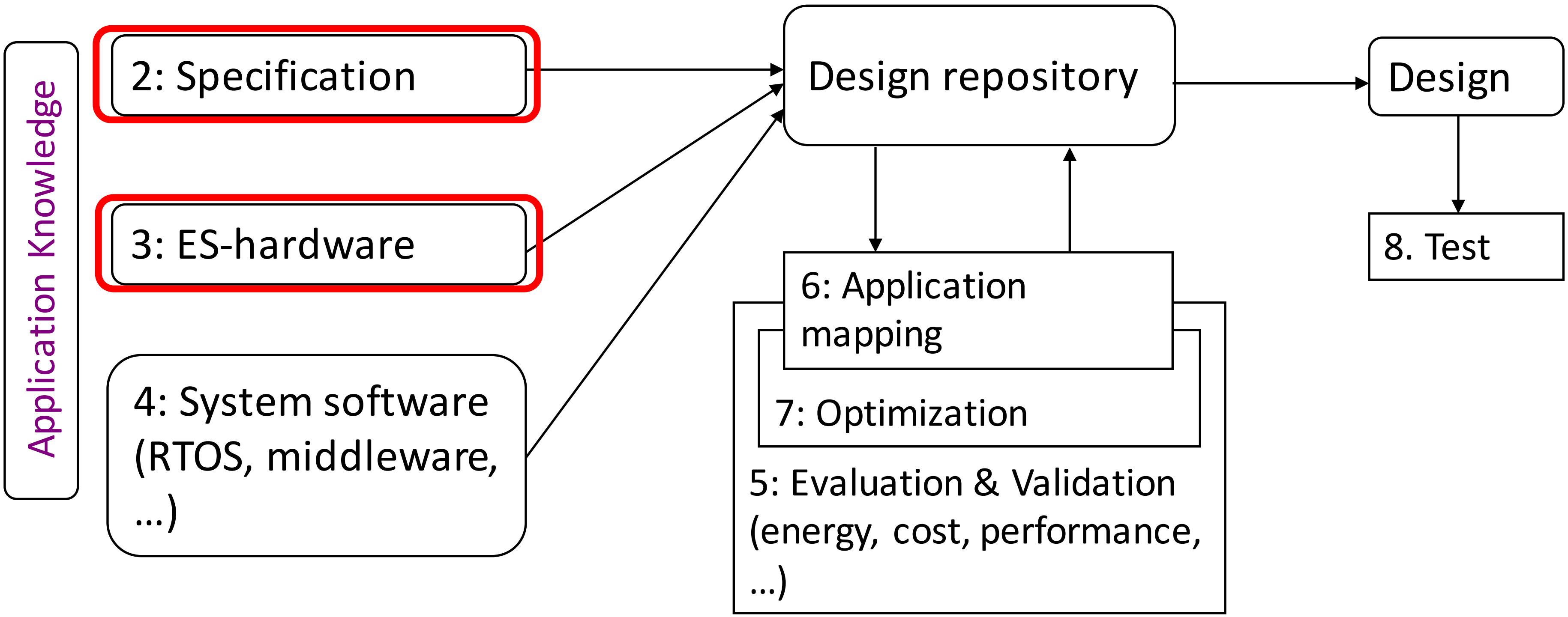
Where Are We?

W	D	Date		Topic	ESD	PES	Out	In	Notes
1	T	12-Jan-2016	L01	Introduction to Embedded System Design	1.1-1.4				
	R	14-Jan-2016		Introduction to Embedded System Design	1.1-1.4				
2	T	19-Jan-2016	L02	Specifying Requirements / MoCs / MSC	2.1-2.3				
	R	21-Jan-2016	L03	CFSMs	2.4				
3	T	26-Jan-2016	L04	Data Flow Modeling	2.5	3.1-5,7	LA1		
	R	28-Jan-2016	L05	Petri Nets	2.6				Thru Slide 21
4	T	2-Feb-2016	L06	Discrete Event Models	2.7	4			G: Zaid Al-bayati
	R	4-Feb-2016	L07	DES / Von Neumann Model of Computation	2.8-2.10	5	LA2	LA1	
5	T	9-Feb-2016	L08	Sensors	3.1-3.2	7.3,12.1-6			
	R	11-Feb-2016	L09	Processing Elements	3.3	12.6-12			
6	T	16-Feb-2016	L10	More Processing Elements / FPGAs			LA3	LA2	
	R	18-Feb-2016	L11	Memories, Communication, Output	3.4-3.6				
7	T	23-Feb-2016	L12	Embedded Operating Systems	4.1				
	R	25-Feb-2016		***Midterm exam: in-class, closed book***			P	LA3	Chapters 1-3
	T	1-Mar-2016		***No class***					Winter break
	R	3-Mar-2016		***No class***					Winter break
8	T	8-Mar-2016	L13	Middleware	4.4-4.5				
	R	10-Mar-2016	L14	Performance Evaluation	5.1-5.2				
9	T	15-Mar-2016	L15	More Evaluation and Validation	5.3-5.8				
	R	17-Mar-2016	L16	Introduction to Scheduling	6.1-6.2.2				
10	T	22-Mar-2016	L17	Scheduling Aperiodic Tasks	6.2.3-6.2.4				
	R	24-Mar-2016	L18	Scheduling Periodic Tasks	6.2.5-6.2.6				

Today

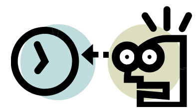
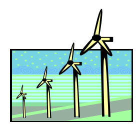
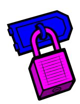
- Embedded Hardware (Chapter 3)
  - Sensors
  - Chapter 3.2

Hypothetical Design Flow



© 2016 Marwedel, Meyer

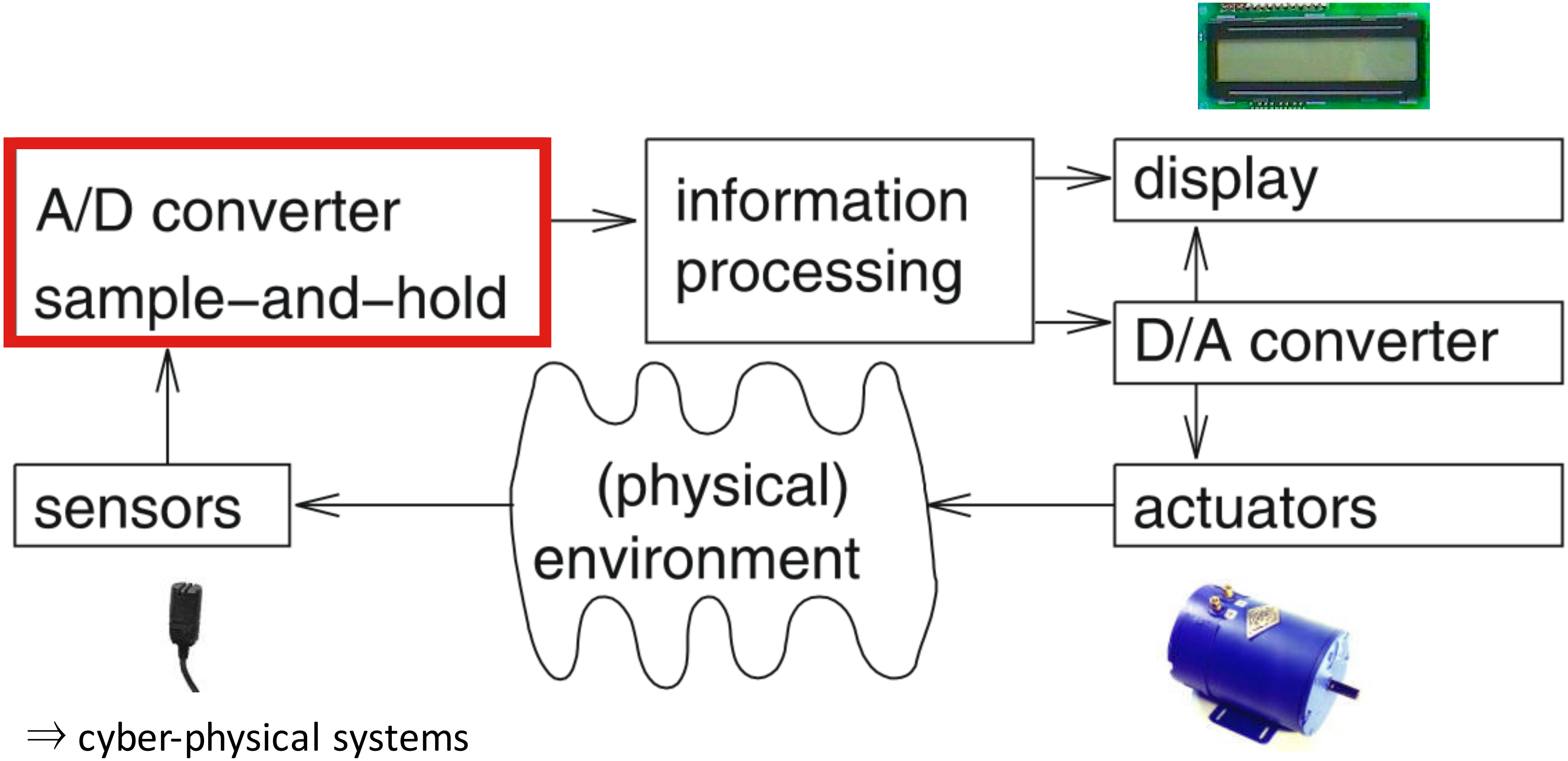
Motivation

- Embedded and Cyber-physical systems are implemented using hardware and software
  - Real-time behavior 
  - Efficiency
    - Energy 
    - …
  - Security 
  - Reliability 
  - …

© 2016 Marwedel, Meyer

Embedded System Hardware

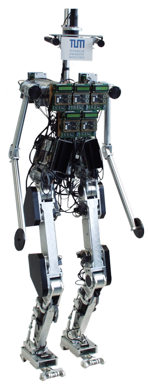
- Embedded system hardware is frequently used in a loop (“*hardware in a loop*”):



⇒ cyber-physical systems

Many Examples of Such Loops

- Heating
- Lights
- Engine control
- Power supply
- …
- Robots



Heating: www.masonsplumbing.co.uk/images/heating.jpg
Robot:: Courtesy and ©: H.Ulbrich, F. Pfeiffer, TU München

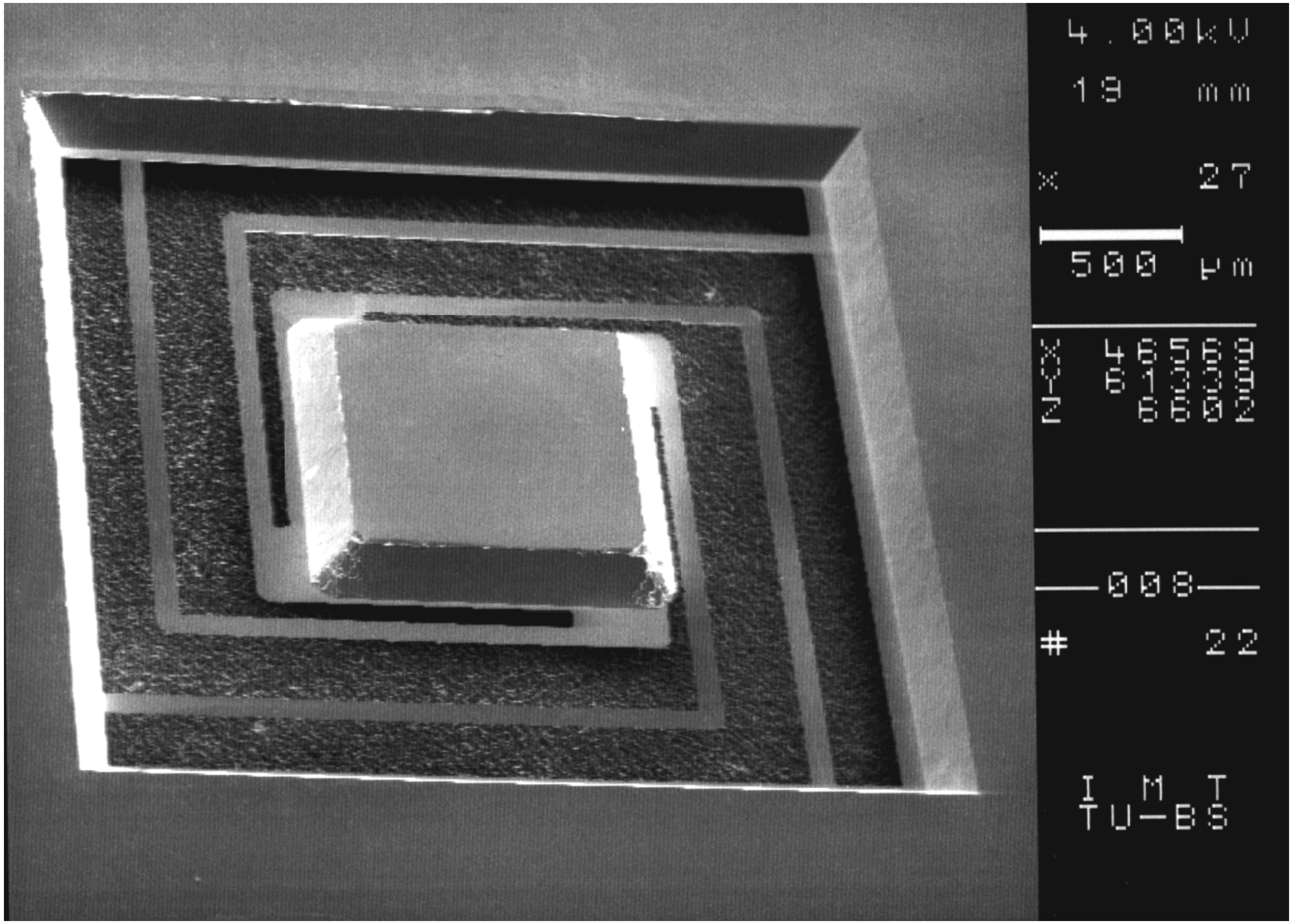
© 2016 Marwedel, Meyer

Sensors

- Sensors are needed to capture the data processed by embedded systems
- Sensors can be designed for virtually anything
  - Physical quantities: weight, velocity, acceleration, electrical current, voltage, temperatures, etc.
  - Chemical and biological compounds
- Many physical effects are used to construct sensors
  - Induction (generation of voltage in an electric field)
  - Photoelectric effects

© 2016 Marwedel, Meyer

Example: Acceleration Sensor

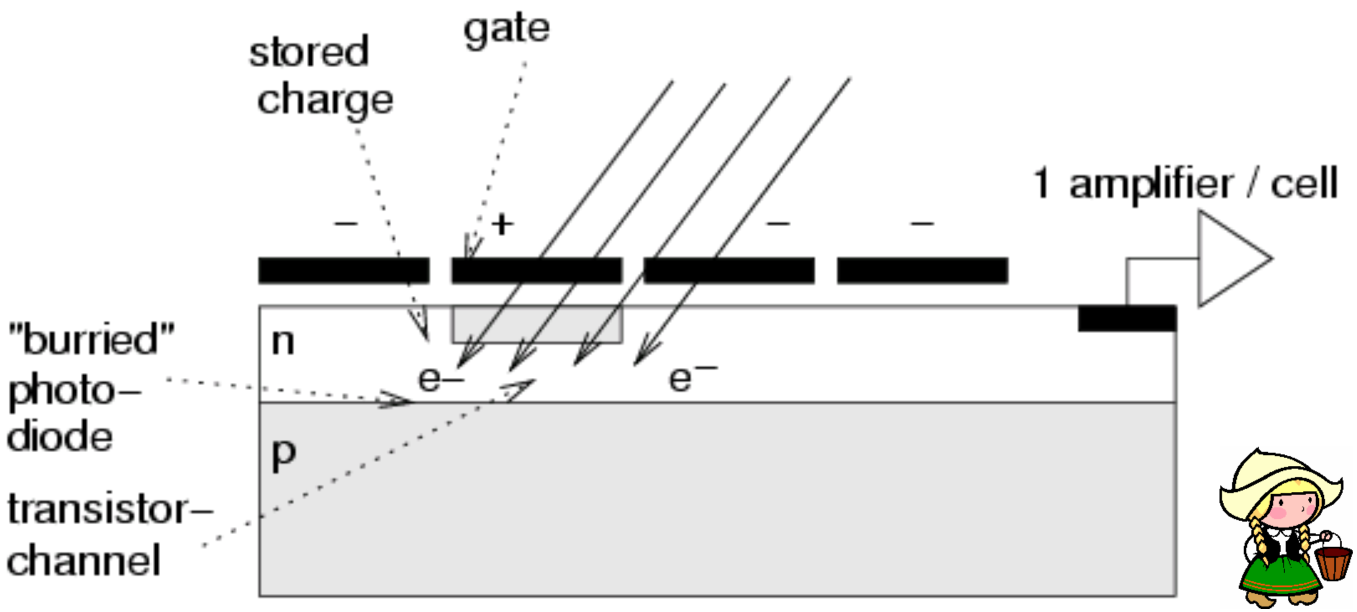


[How does it work? See http://www.youtube.com/watch?v=KZVgKu6v808]

Courtesy & ©: S. Bütgenbach, TU Braunschweig

Charge-coupled Devices (CCD)

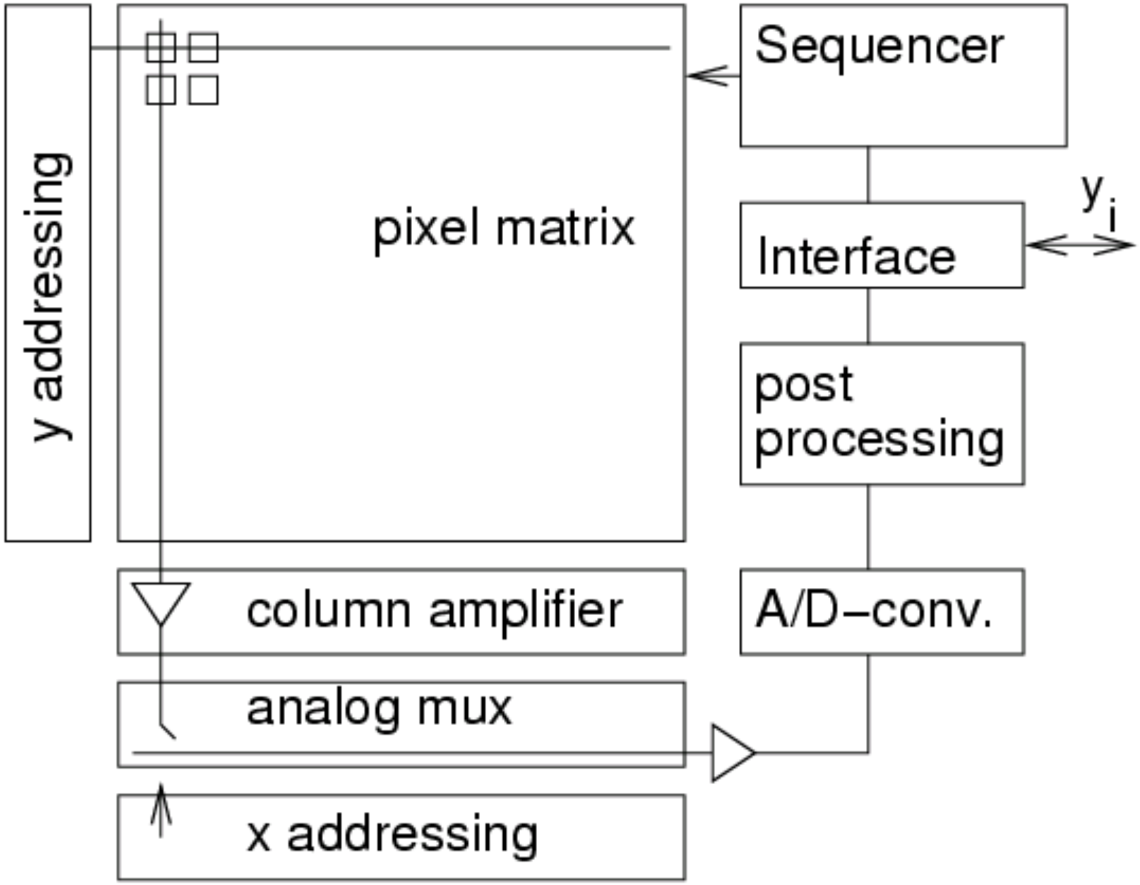
Image sensors based on charge transfer to next pixel cell



Corresponding to “bucket brigade device”
(German: “*Eimerkettenschaltung*”)

CMOS Image Sensors

- Based on standard CMOS
- Can integrate with other ICs:
  - On-chip image processing



Comparison CCD/CMOS sensors

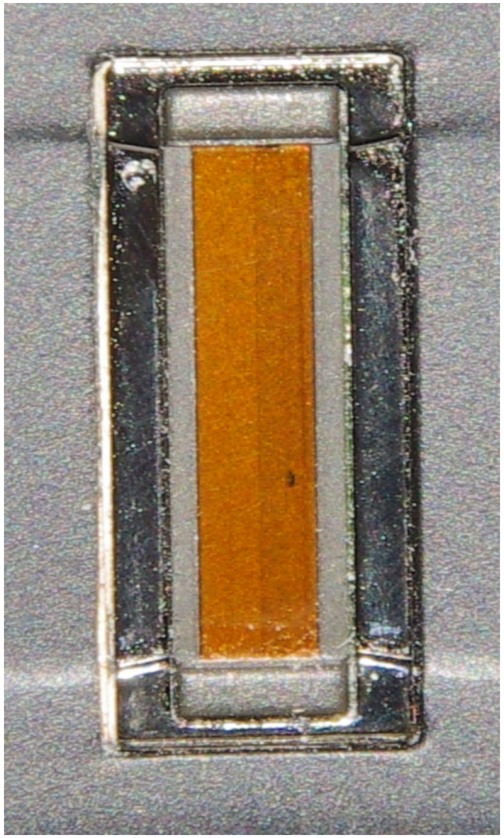
Property	CCD	CMOS
Technology optimized for	Optics	VLSI technology
Technology	Special	Standard
Smart sensors	No, no logic on chip	Logic elements on chip
Access	Serial	Random
Size	Limited	Can be large
Power consumption	Low	Larger
Applications	Compact cameras	Low cost devices, SLR cameras

See also B. Diericks: CMOS image sensor concepts. Photonics West 2000 Short course (Web)

© 2016 Marwedel, Meyer

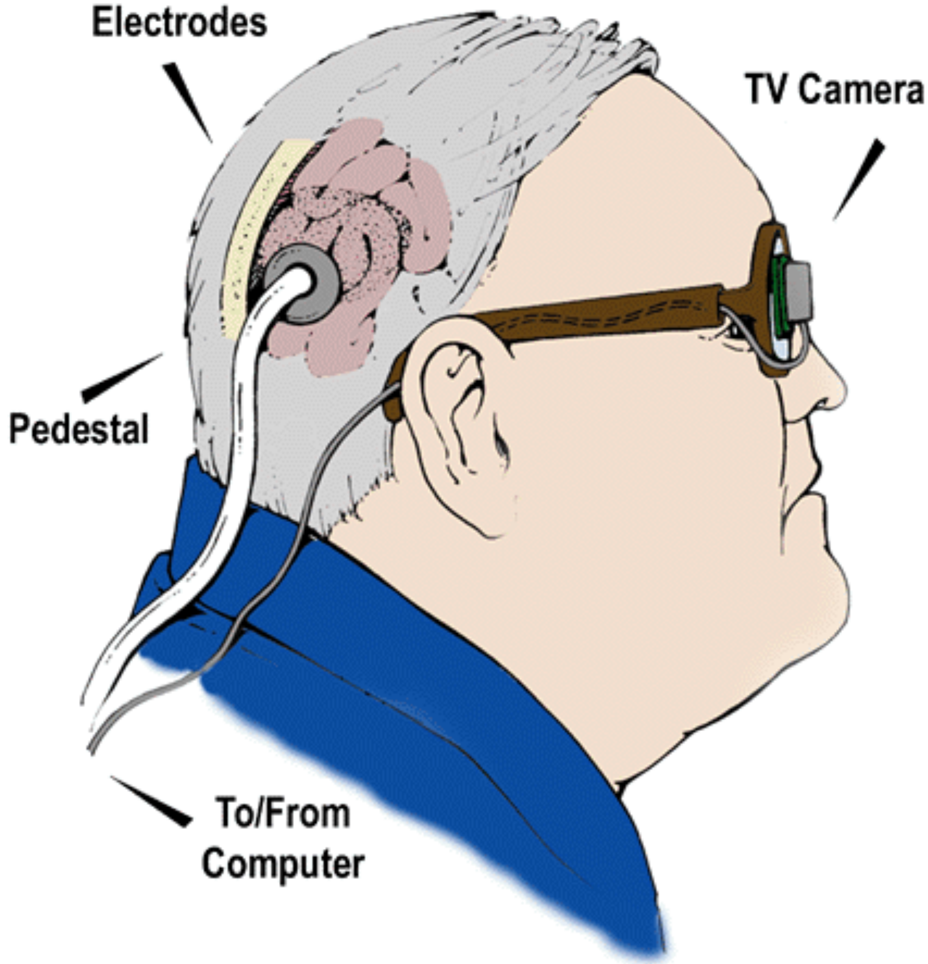
Example: Biometric Sensors

E.g.: Fingerprint sensor



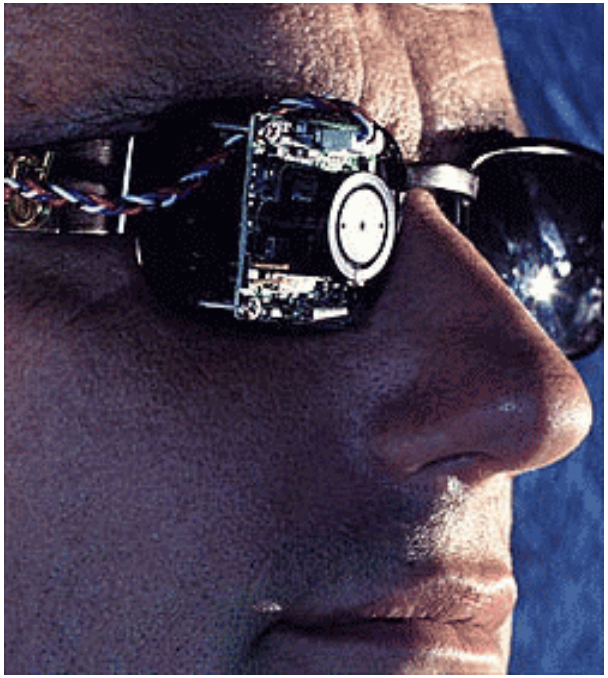
© P. Marwedel, 2010

Artificial Eyes



© Dobelle Institute
(was at www.dobelle.com)

Artificial Eyes (2)



"He looks hale, hearty, and healthy — except for the wires. …. From a distance the wires look like long ponytails."

© Dobelle Institute

Artificial Eyes (3)

- Translation into sound; resolution claimed to be good [http://www.seeingwithsound.com/etumble.htm]

Other Sensors

- Rain sensors for wiper control
  (“Sensors multiply like rabbits” [ITT automotive])
- Pressure sensors
- Proximity sensors
- Engine control sensors
- Hall effect sensors

Signals

- Sensors generate *signals*

Def.: a **signal** s is a mapping from the time domain D_T to a value domain D_V:

$$s: D_T \rightarrow D_V$$

- D_T : continuous or discrete time domain
- D_V : continuous or discrete value domain

© 2016 Marwedel, Meyer

Discretization of Time: Sampling

- Digital computers require discrete sequences of physical values

$$s: D_T \to D_V$$

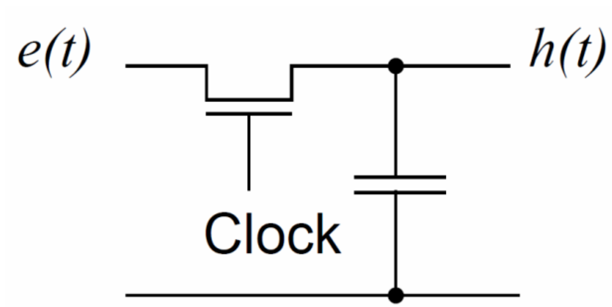
Discrete time domain

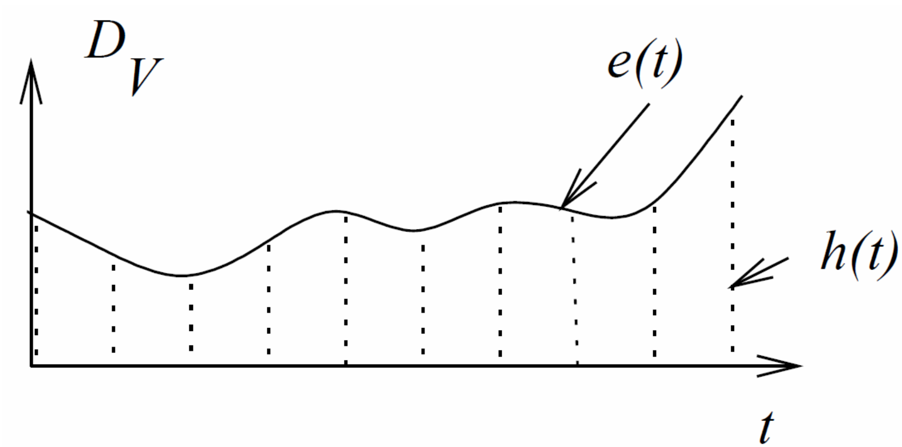
$\Rightarrow$ Sample-and-hold circuits

© 2016 Marwedel, Meyer

Sample-and-hold Circuits

- Implemented with a clocked transistor + capacitor
- Capacitor stores sequence a sequence of values





- $e(t)$ is a mapping $\mathbb{R} \rightarrow \mathbb{R}$
- $h(t)$ is a **sequence** of values or a mapping $\mathbb{Z} \rightarrow \mathbb{R}$

© 2016 Marwedel, Meyer

Does Sampling Lose Information?

- Can we reconstruct input signals from samples?

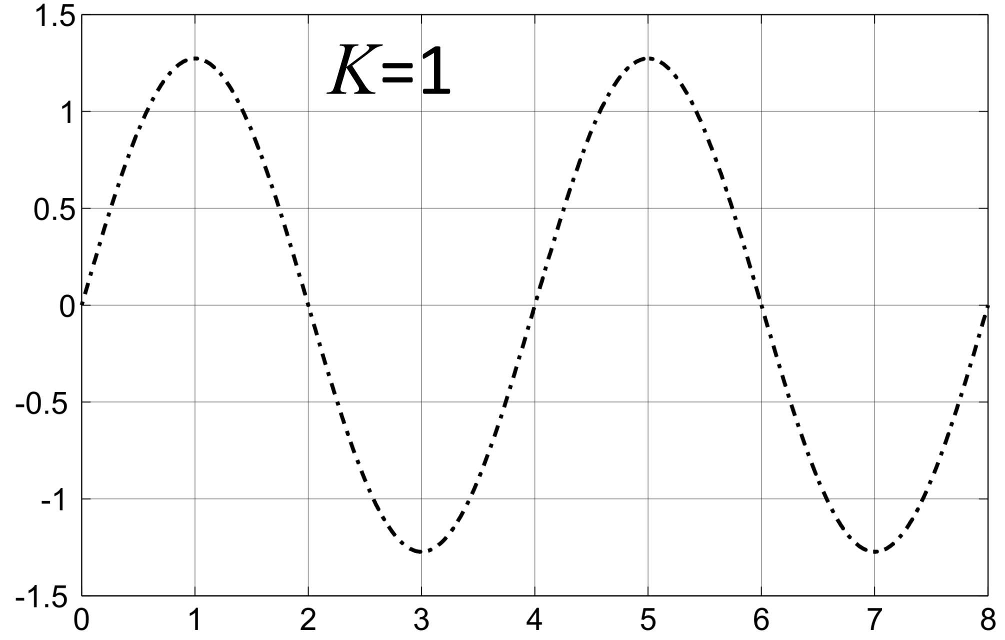
$\Rightarrow$ Approximation of signals by sine waves

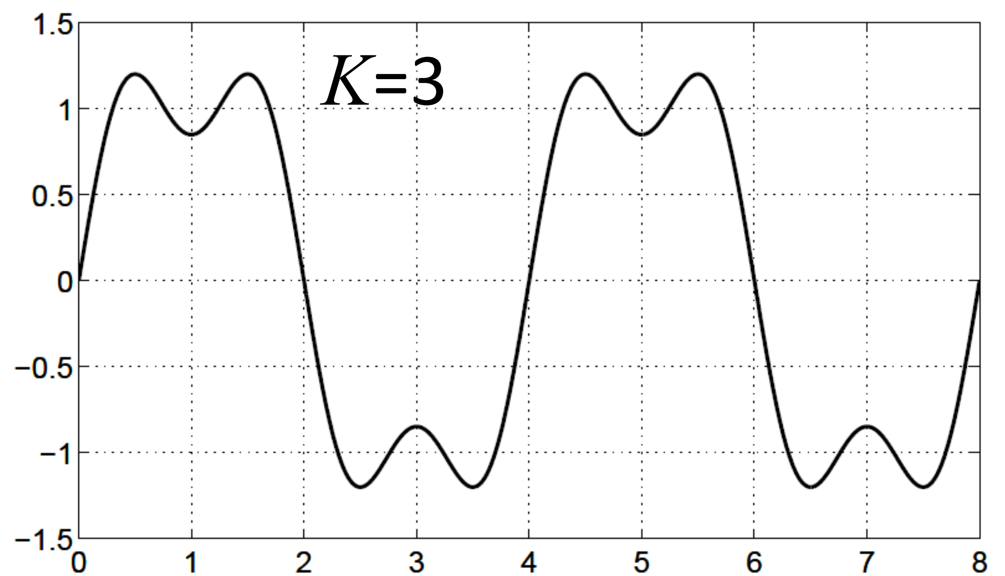
Approximation of a Square Wave (1)

- Target: square wave with period p_1=4

$$e'_K(t) = \sum_{k=1,3,5,..}^{K} \frac{4}{\pi k} \sin\left(\frac{2\pi\, t}{p_k}\right)$$

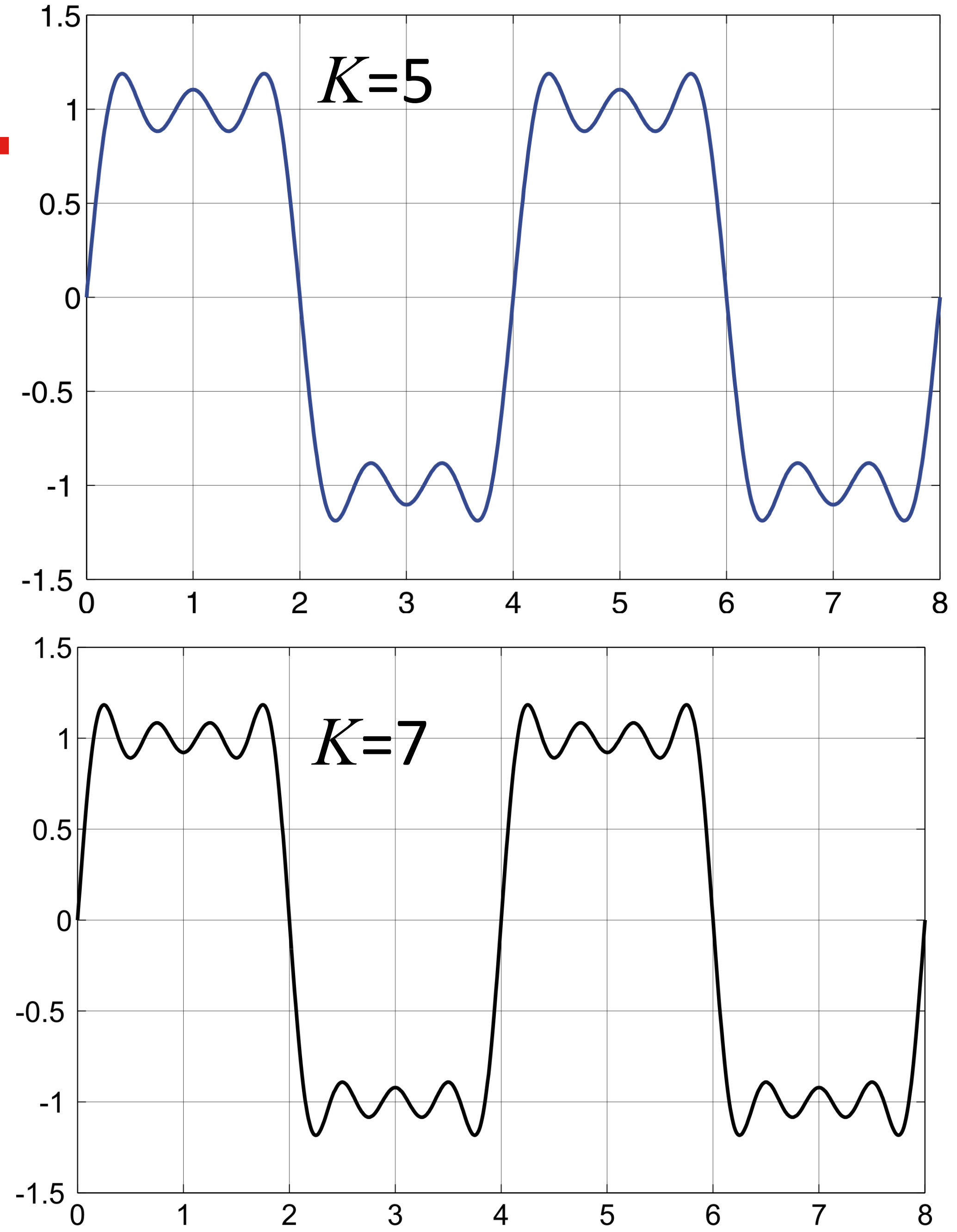
- $\forall k$: $p_k = p_1/k$: periods that contribute to e'





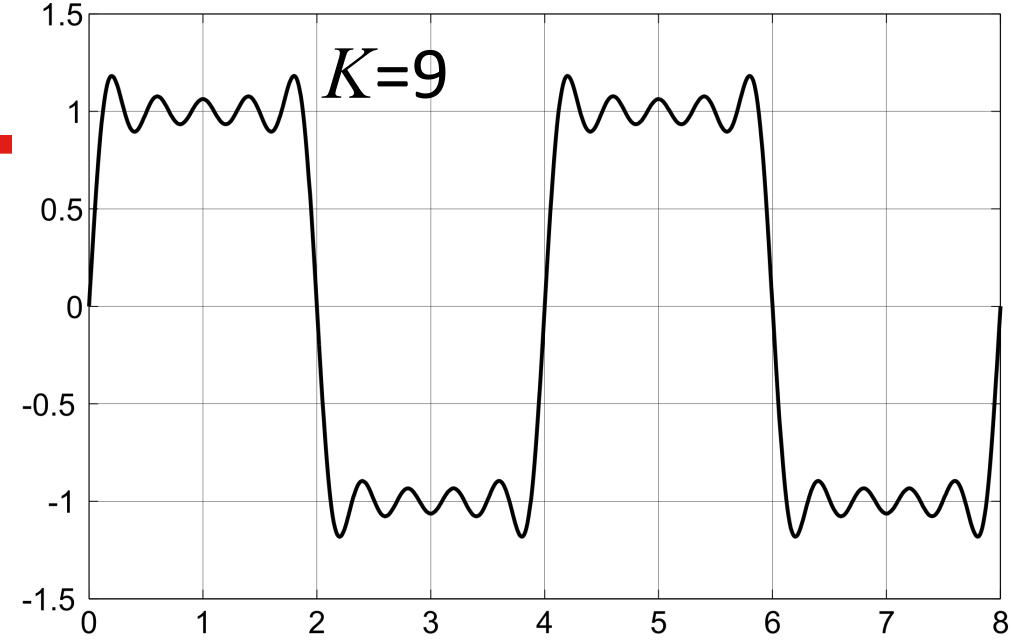
Approximation of a Square Wave (2)

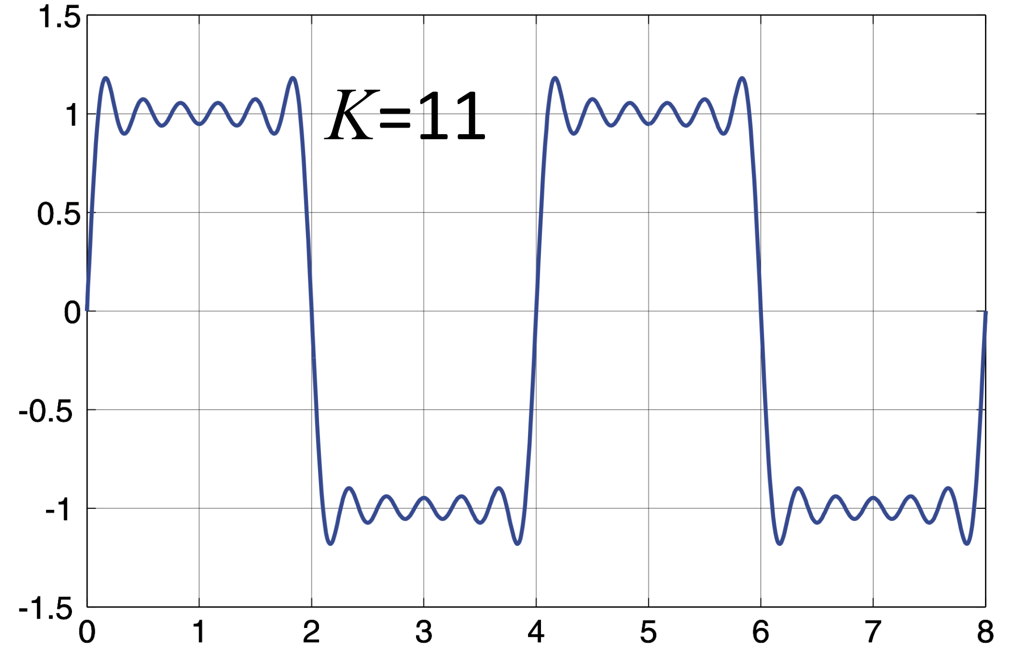
$$e'_K(t) = \sum_{k=1,3,5,..}^{K} \frac{4}{\pi k} \sin\left(\frac{2\pi\, t}{4/k}\right)$$



Approximation of a Square Wave (3)

$$e'_K(t) = \sum_{k=1,3,5,..}^{K} \frac{4}{\pi k}\sin\left(\frac{2\pi\, t}{4/k}\right)$$





Linear Transformations

- Let $e_1(t)$ and $e_2(t)$ be signals

Def.: A transformation Tr of signals is linear iff

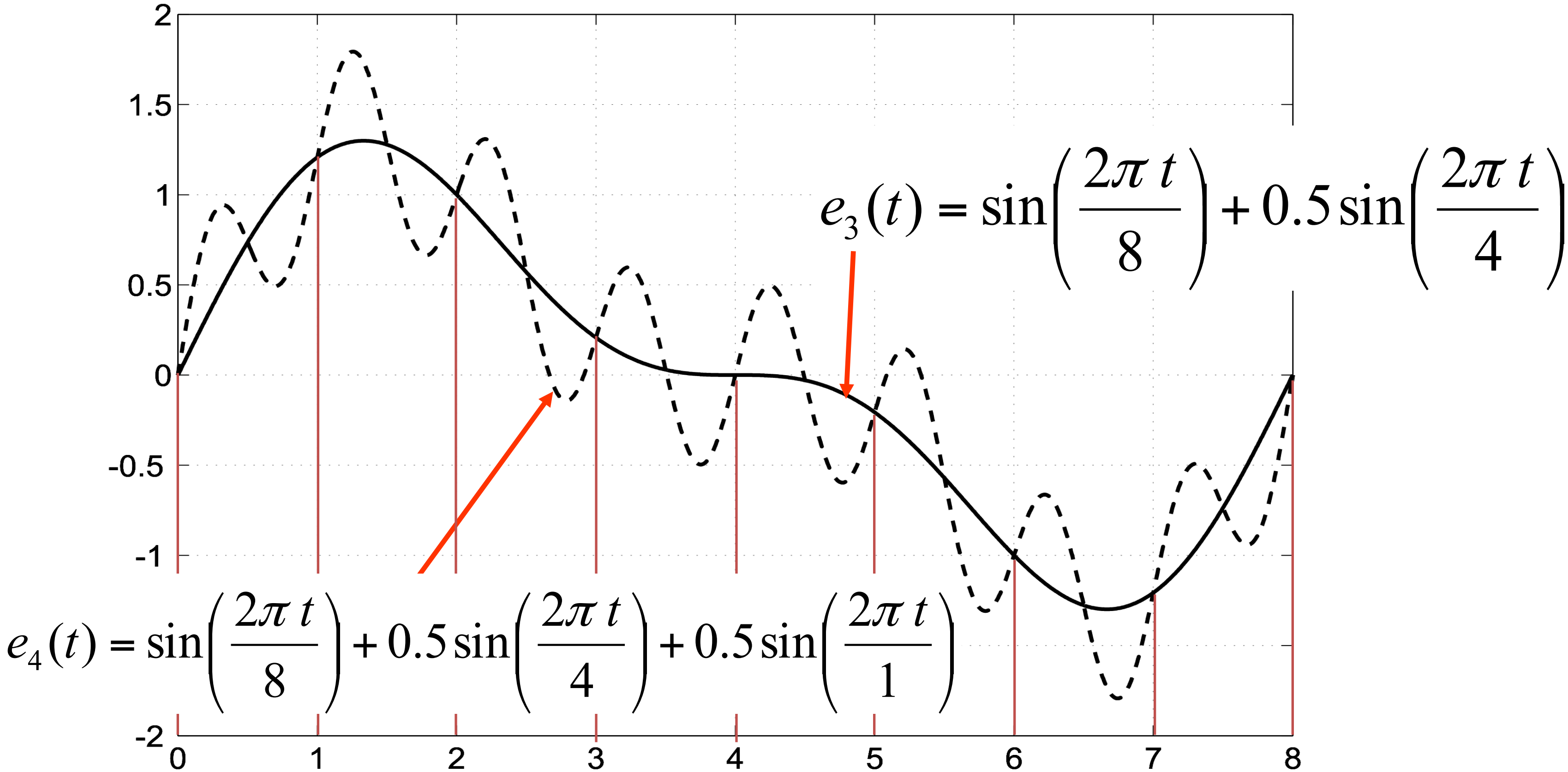
$$Tr(e_1 + e_2) = Tr(e_1) + Tr(e_2)$$

- In the following, we will consider linear transformations
  - We will consider sums of sine waves
    - Not the original signals
  - Given sine waves, can we reconstruct signals?

McGill

© 2016 Marwedel, Meyer

Aliasing (1)



- Periods of 8,4,1
- Indistinguishable if sampled at integer times, p_s=1

© 2016 Marwedel, Meyer

Aliasing (2)

- Reconstruction is impossible without adequate sampling
  - How frequently do we have to sample?
- *Nyquist criterion* (from sampling theory):
  - Aliasing can be avoided if we restrict the frequencies of the incoming signal to **less than half** of the sampling rate

$p_s < \frac{1}{2}\, p_N$; p_N is the period of the "fastest" sine wave, or

$f_s > 2\, f_N$; f_N is the frequency of the "fastest" sine wave

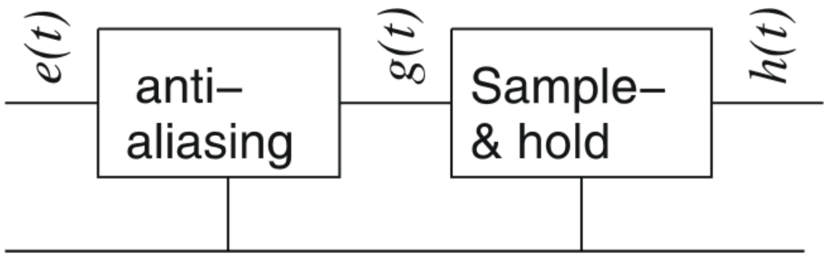
f_N: the **Nyquist frequency**; f_s: the **sampling rate**

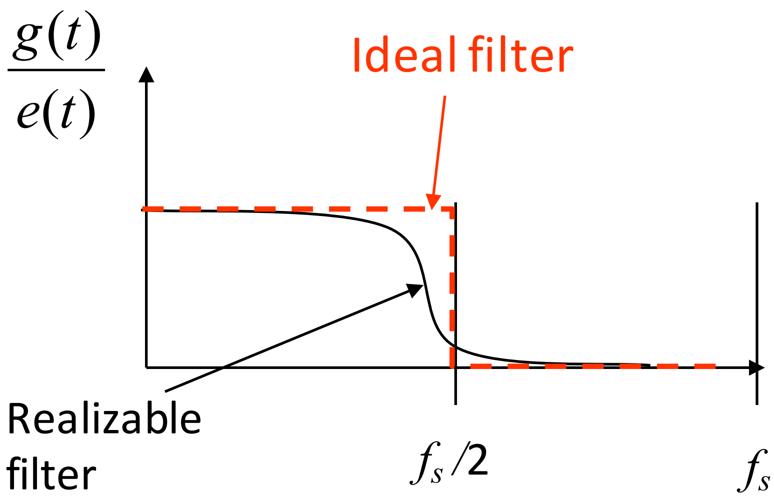
See e.g. [Oppenheim/Schafer, 2009]

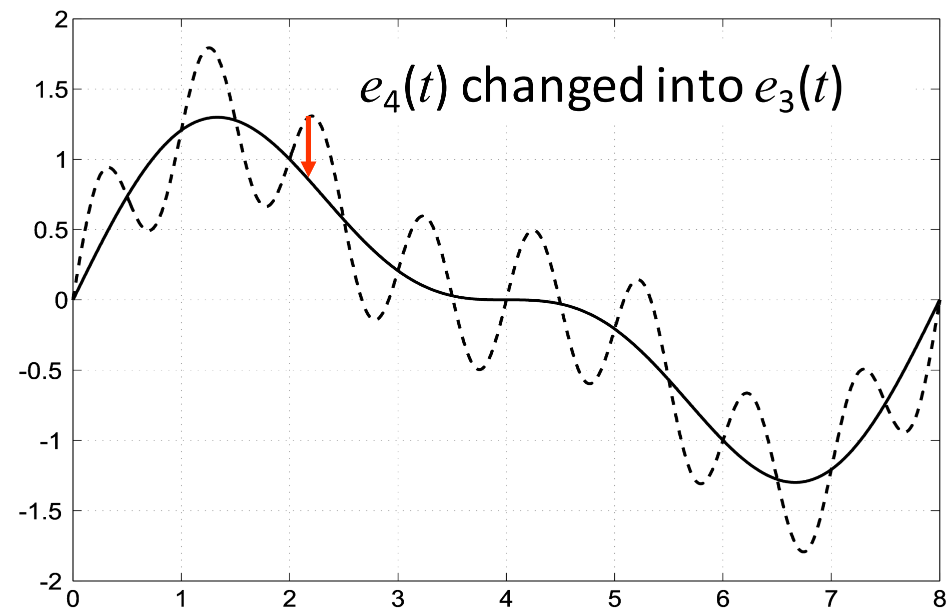
© 2016 Marwedel, Meyer

Anti-aliasing Filter

- A filter is needed to remove high frequencies



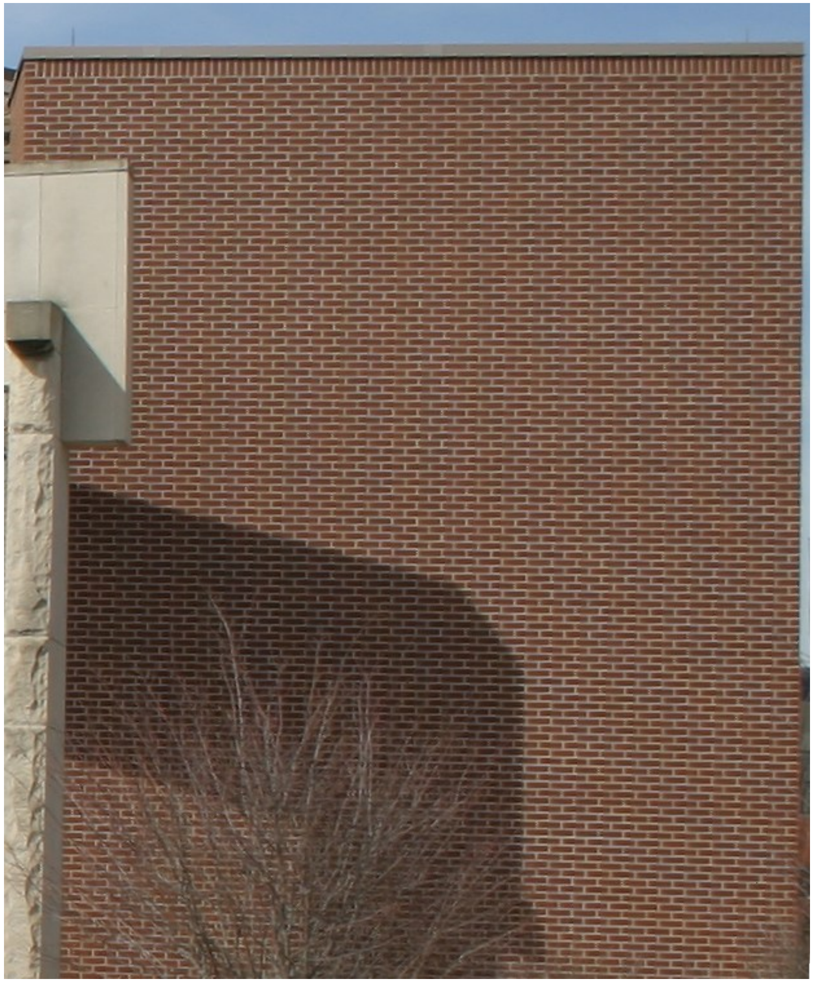




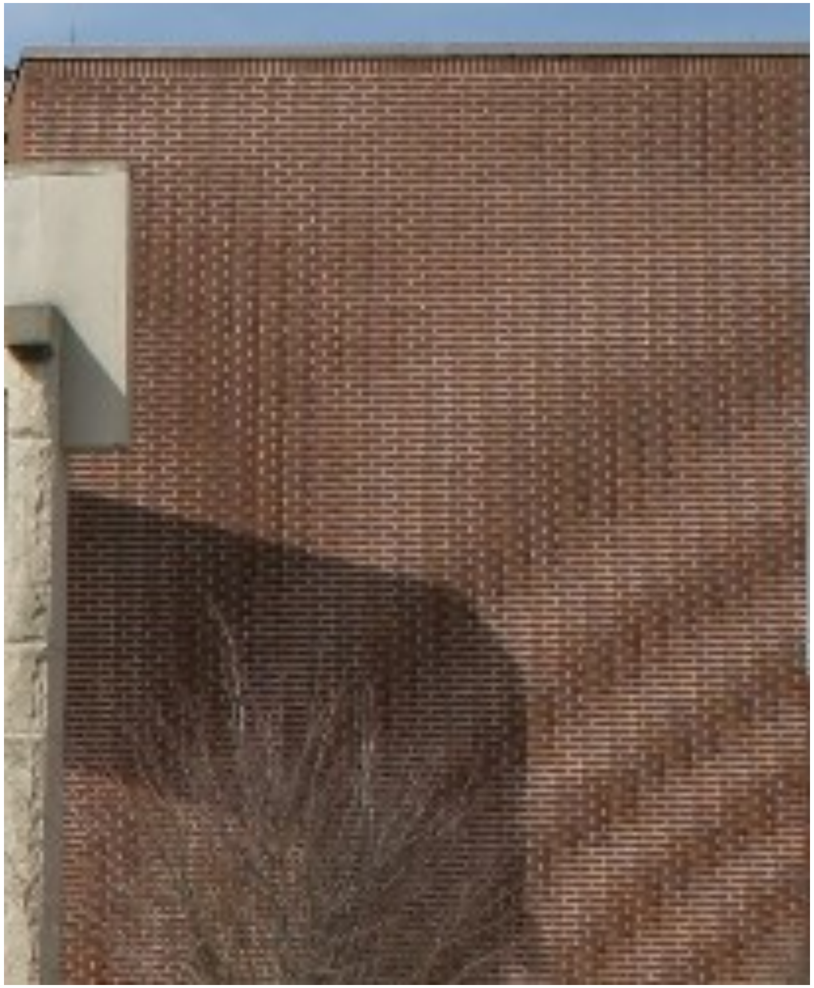
© 2016 Marwedel, Meyer

Aliasing in Computer Graphics (1)

- Original



- Sub-sampled, no filtering

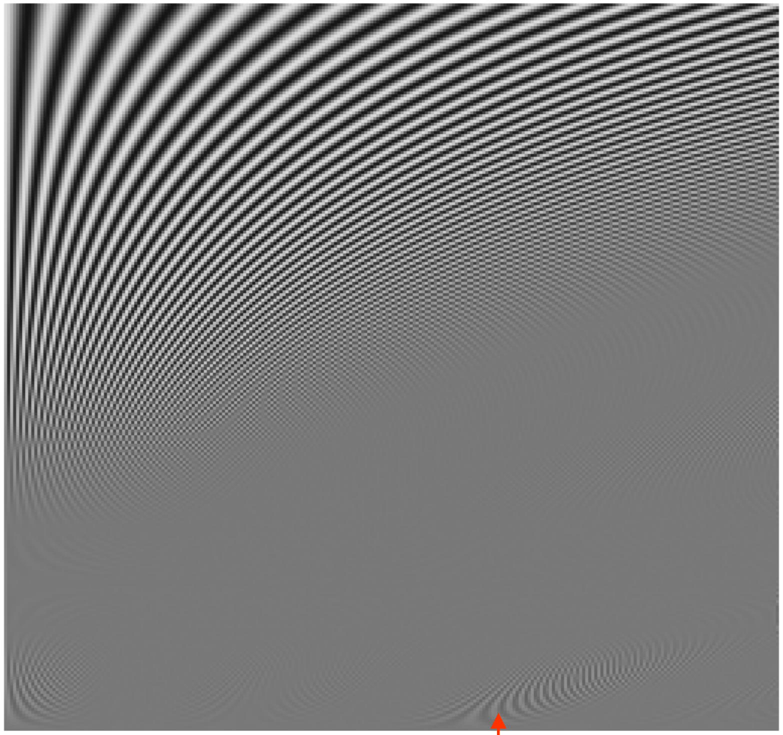


http://en.wikipedia.org/wiki/Image:
Moire_pattern_of_bricks_small.jpg

© 2016 Marwedel, Meyer

Aliasing in Computer Graphics (2)

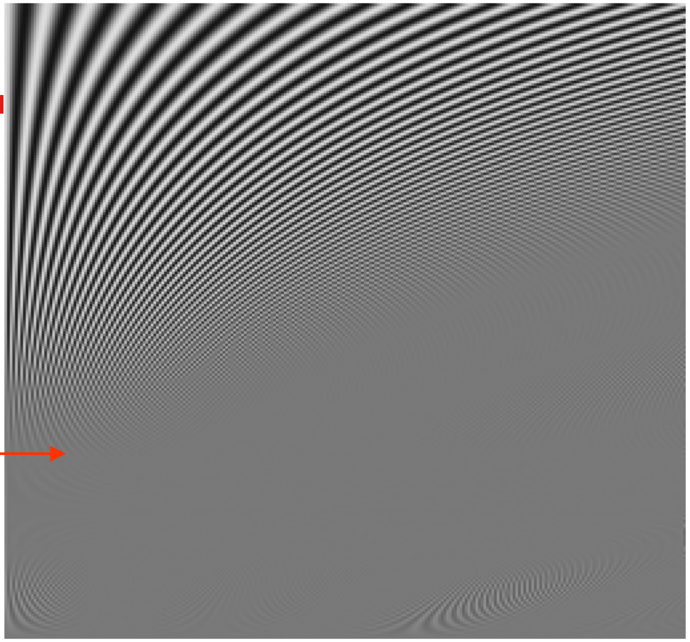
Original (pdf screen copy)



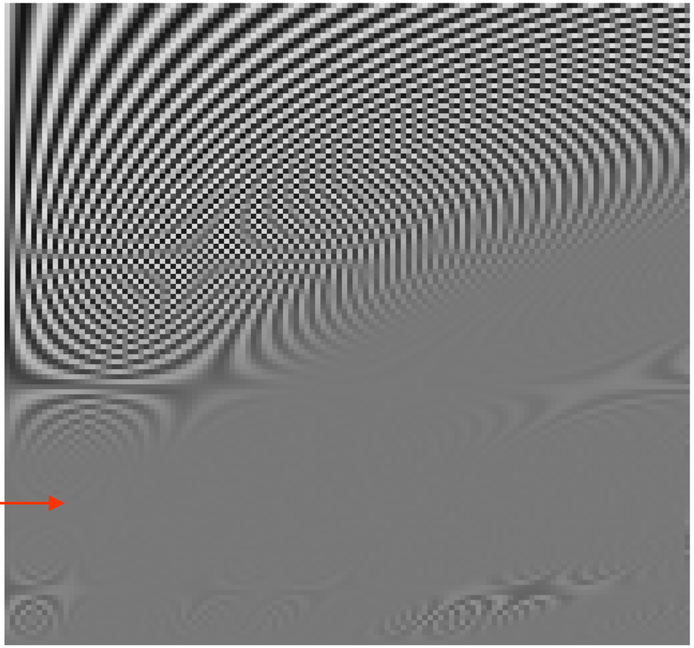
Impact of rasterization

http://www.niirs10.com/Resources/Reference Documents/Accuracy in Digital Image Processing.pdf

Filtered & sub-sampled



Sub-sampled, no filtering



Discretization of Values: A/D-Converters

- Digital computers require digital form of physical values

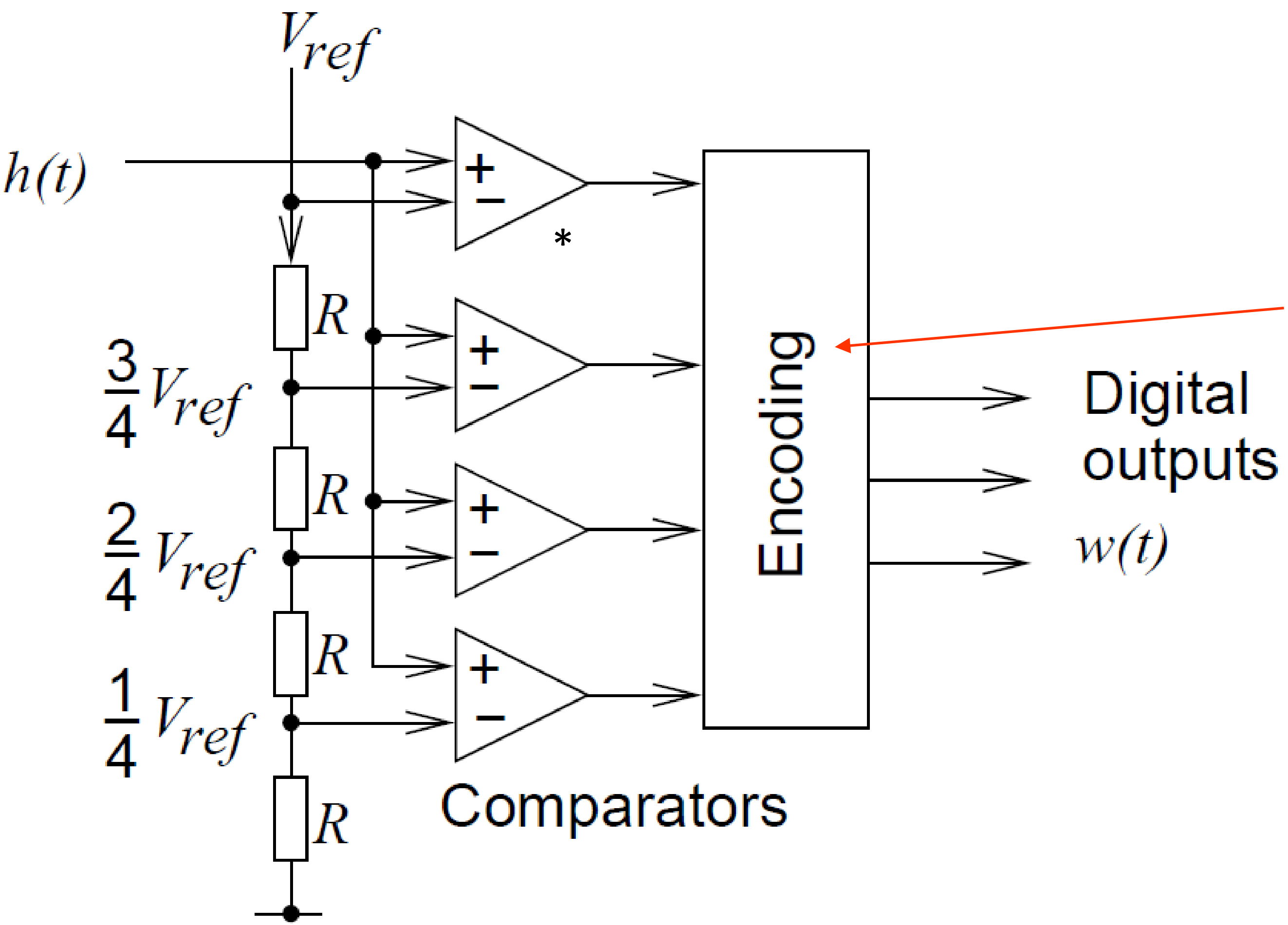
$$s: D_T \rightarrow D_V$$

Discrete value domain

$\Rightarrow$ A/D-conversion; many methods, different rates

© 2016 Marwedel, Meyer

Flash A/D Converter

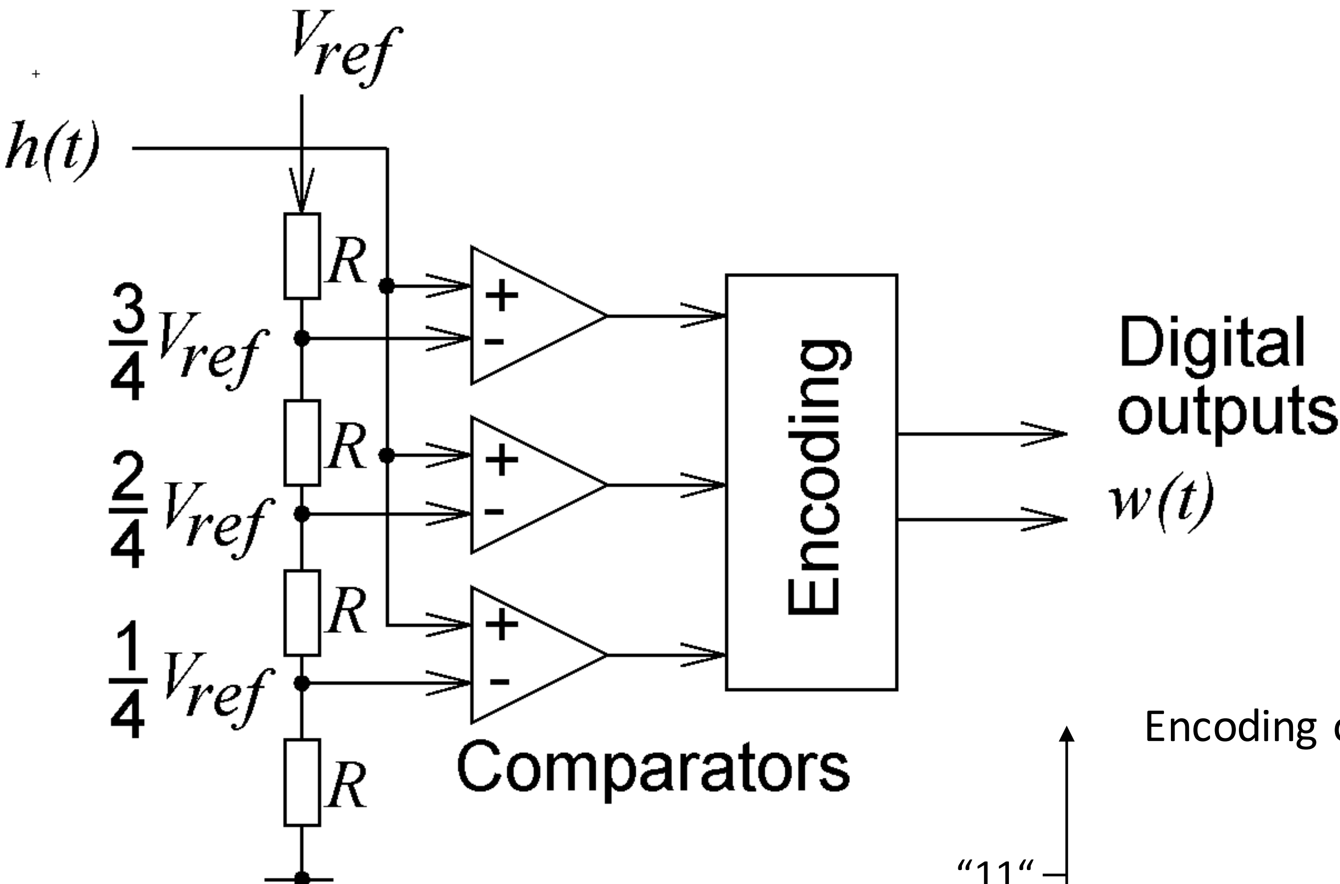


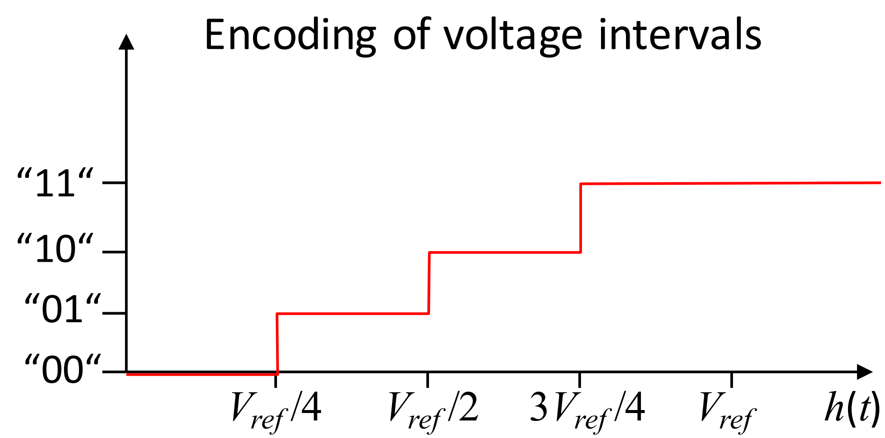
- Encodes input using priority encoder
- Position of most significant '1' ⇒ unsigned number:

"1111" -> "100",
"0111" -> "011",
"0011" -> "010",
"0001" -> "001",
"0000" -> "000"

* Frequently, the case $h(t) > V_{ref}$ would not be decoded

Assuming $0 \leq h(t) \leq V_{ref}$





© 2016 Marwedel, Meyer

Conversion Resolution

- Resolution (in bits): number of bits produced
- Resolution Q (in volts): difference between two input voltages causing a change in output

$$Q = \frac{V_{FSR}}{n} \quad \text{where}$$

Q:	resolution in volts per step
V_{FSR}:	difference between largest and smallest voltage
n:	number of voltage intervals

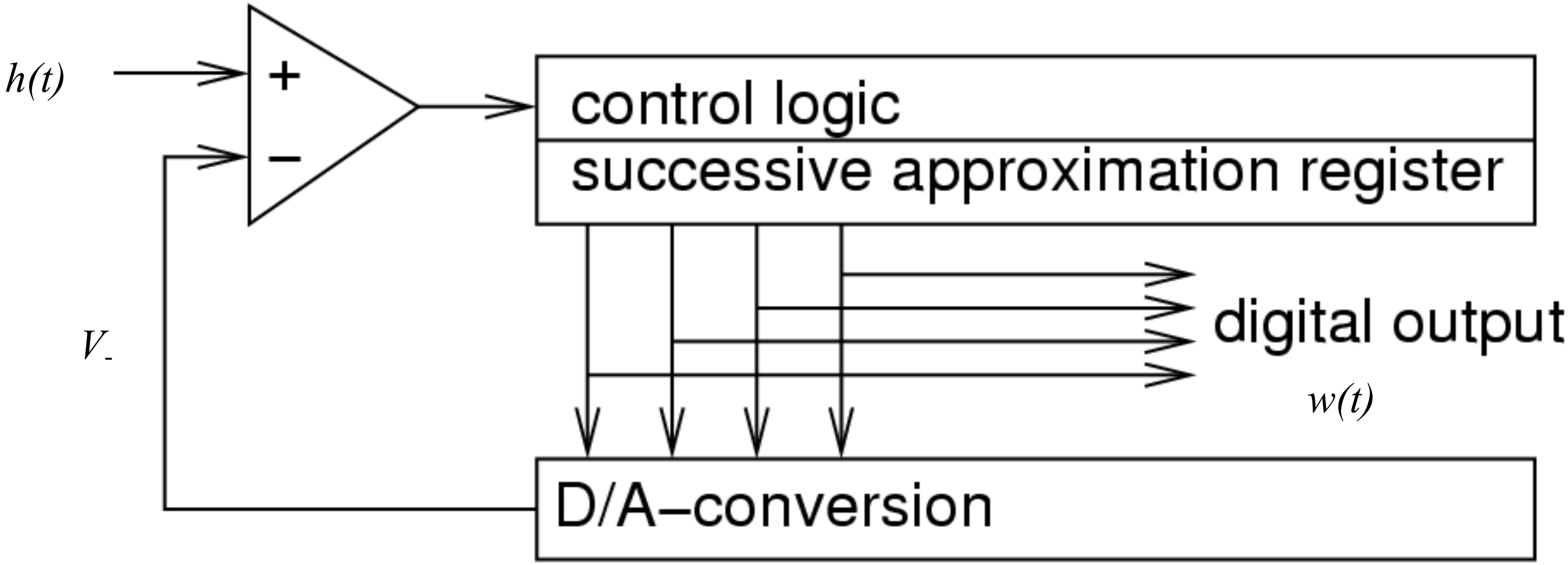
Example:
$Q = V_{ref}/4$ for the previous slide, assuming $0 \le h(t) \le V_{ref}$

Flash ADC Resolution and Speed

- **Parallel comparison with reference voltage**
- Speed: $O(1)$
- Hardware complexity: $O(n)$
- **Applications**: *e.g.* in video processing

© 2016 Marwedel, Meyer

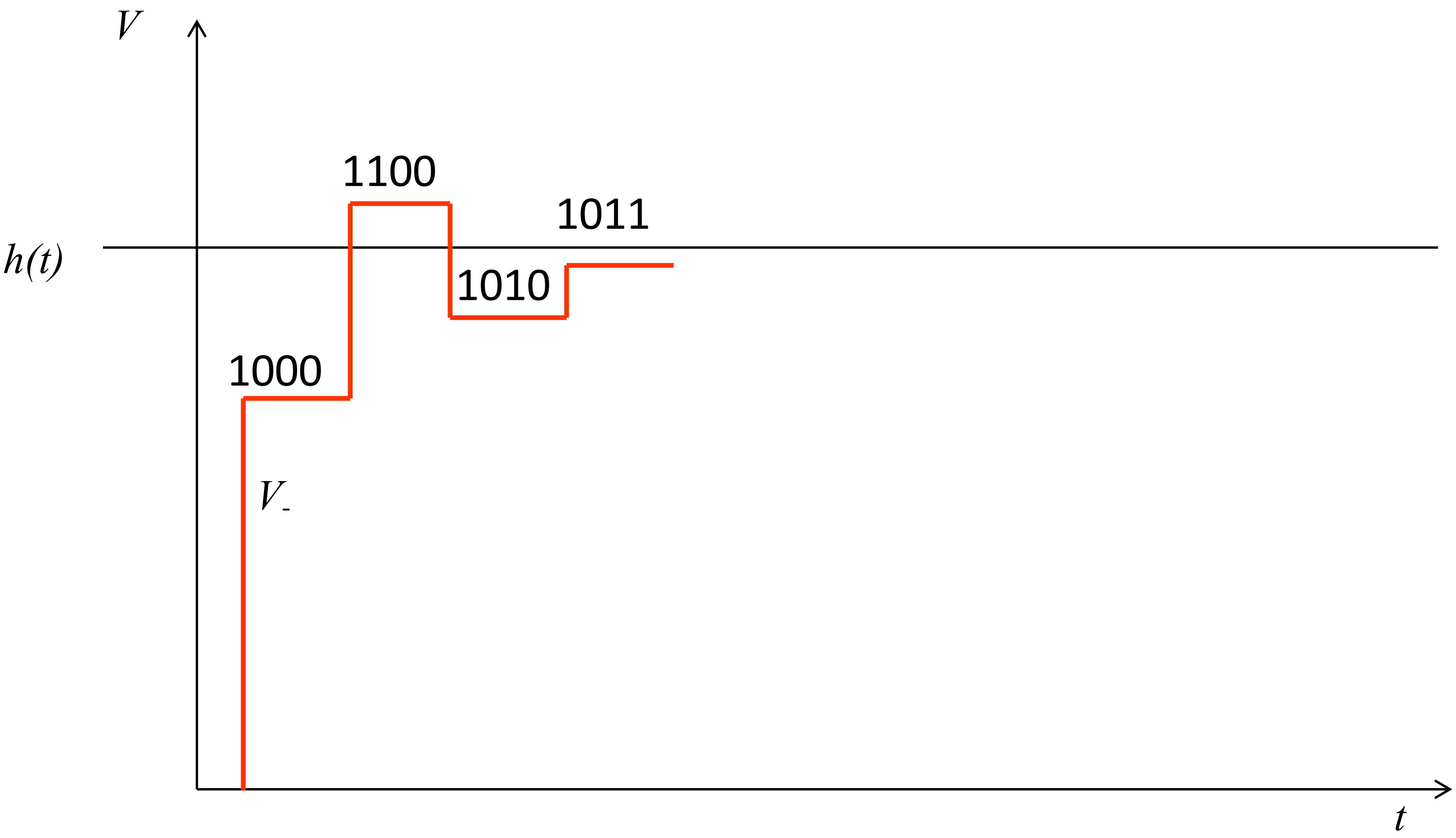
Successive Approximation (1)



- Key idea: binary search
- Set MSB='1'
  - if too large: reset MSB
- Set MSB-1='1'
  - if too large: reset MSB-1
- ...

- Speed: $O(\log_2(n))$
- HW Complexity: $O(\log_2(n))$
- n = # of different voltage levels;
- slow, but high precision possible

Successive Approximation (2)



© 2016 Marwedel, Meyer

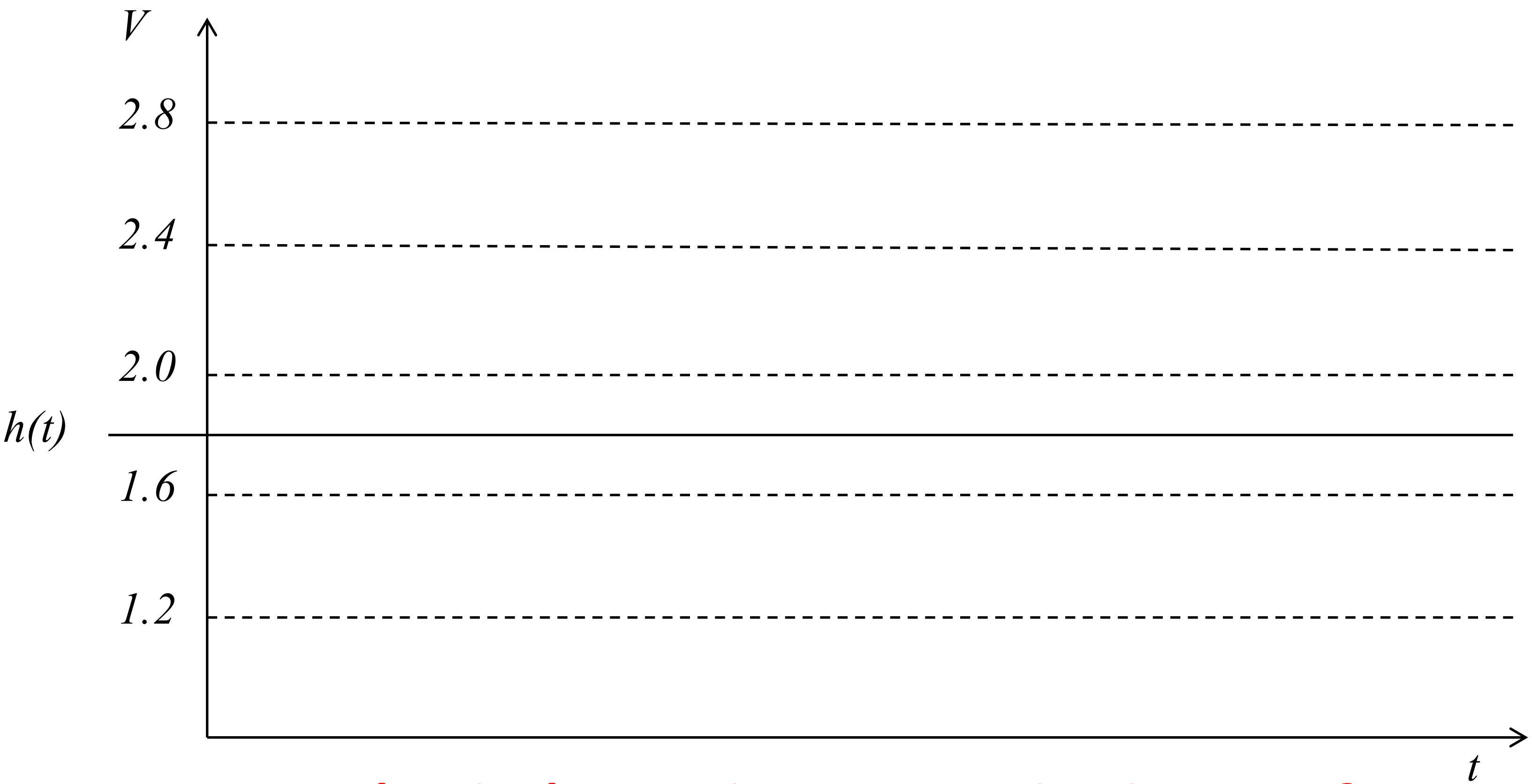
Example

- $[U_{min}, U_{max}] = [1.2\ V, 2.8\ V]$
- Assume a 2-bit output encoder
- What are V_{FSR}, n, and Q?
- What voltage intervals correspond to each digital output?

© 2016 Marwedel, Meyer

Example, Continued

- Given an input U_{in} = 1.8 V, perform successive approximation to find the appropriate encoding



What is the maximum quantization error?

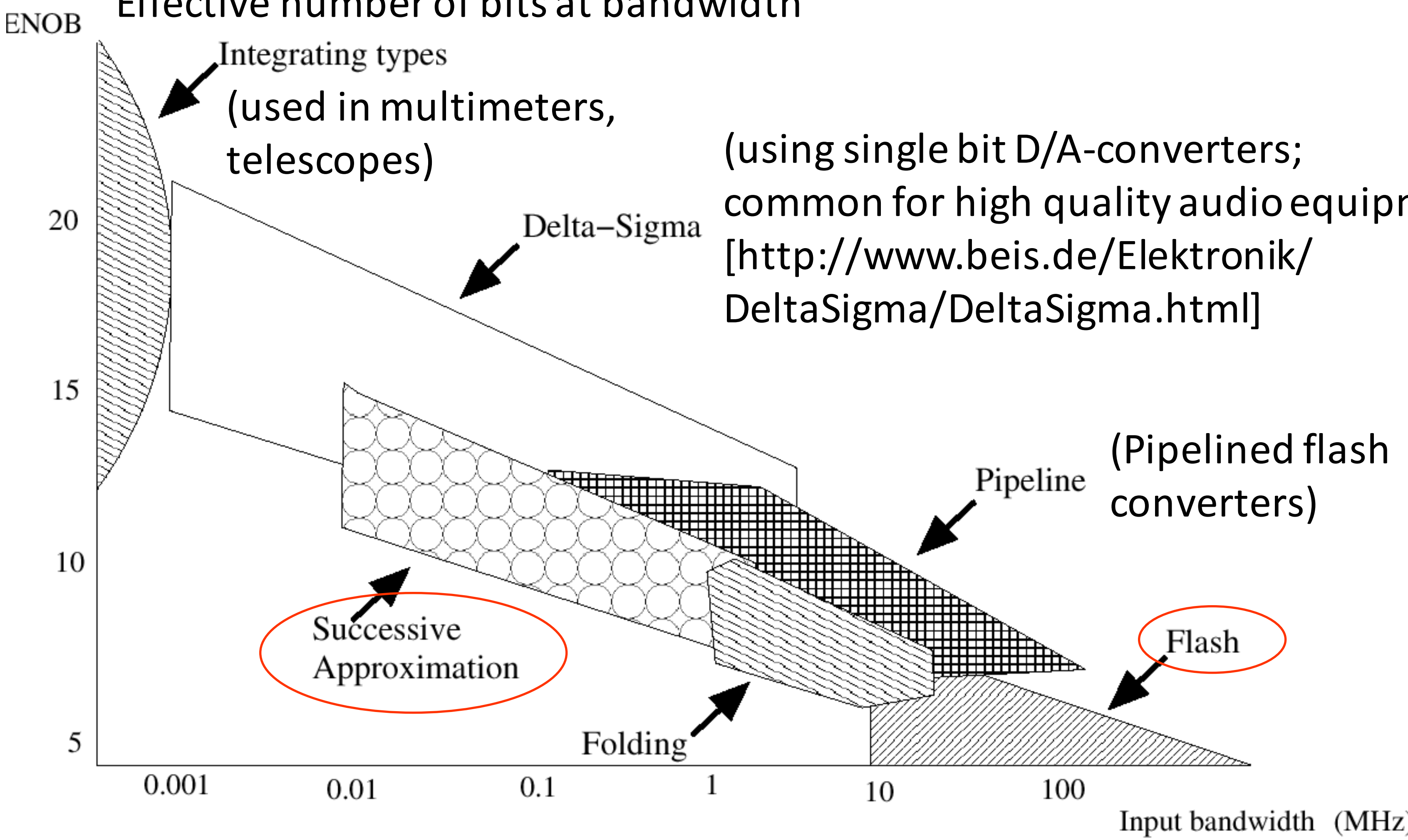
Application Areas for ADC

Effective number of bits at bandwidth

(used in multimeters, telescopes)

(using single bit D/A-converters; common for high quality audio equipment) [http://www.beis.de/Elektronik/DeltaSigma/DeltaSigma.html]

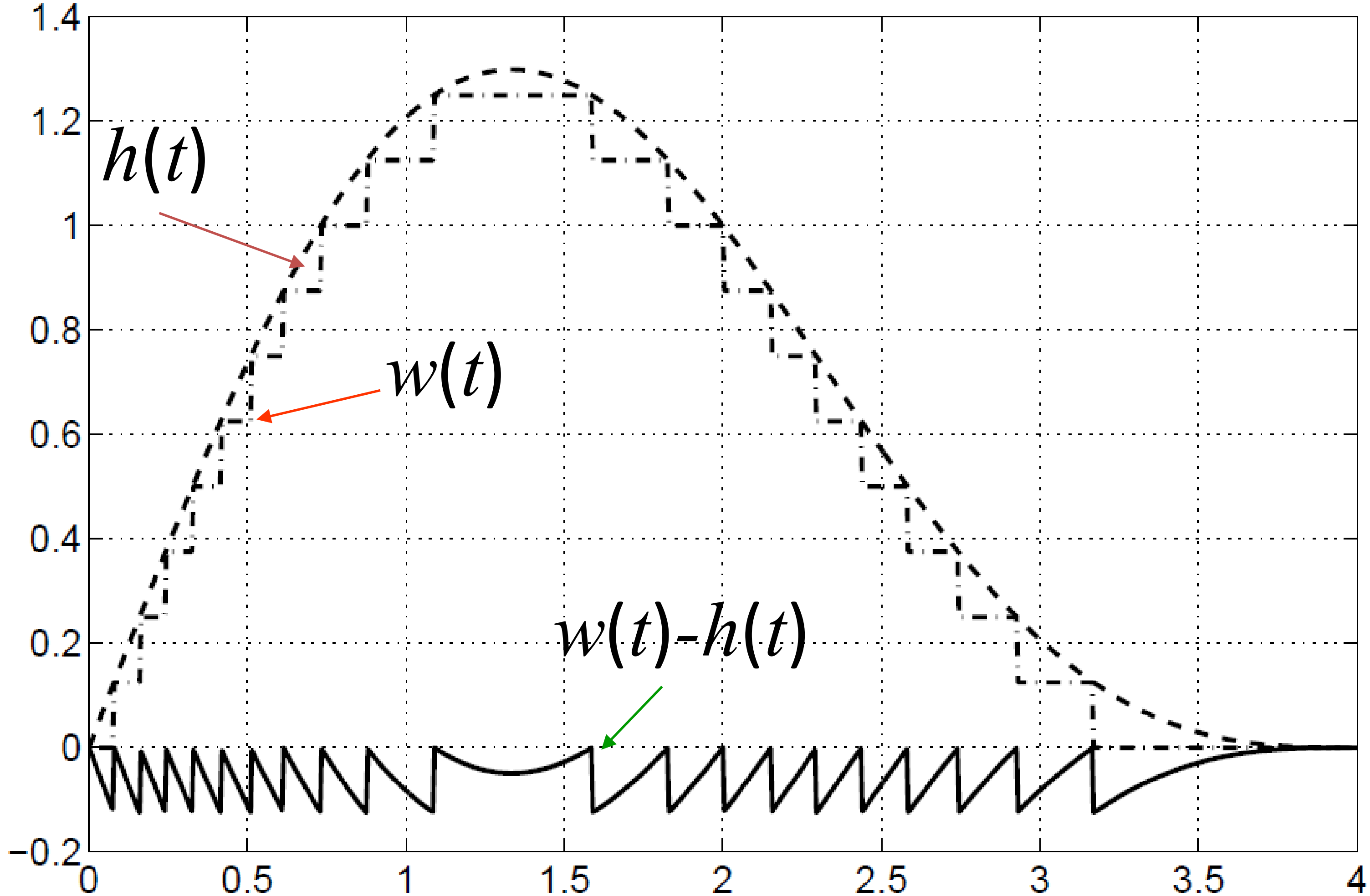
(Pipelined flash converters)



[Gielen et al., DAC 2003]

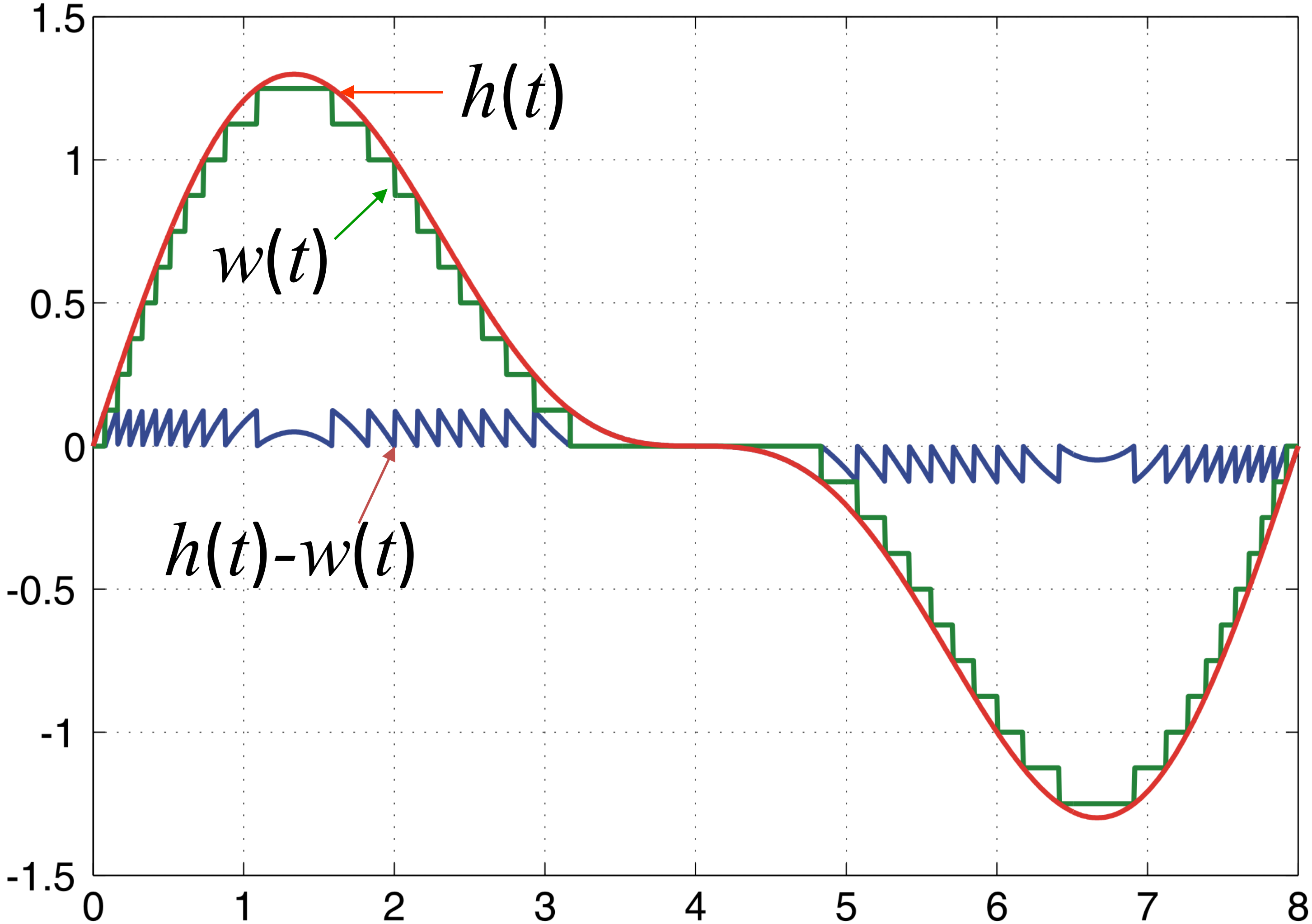
Quantization Noise (1)

- Assuming "rounding" (truncating) towards 0

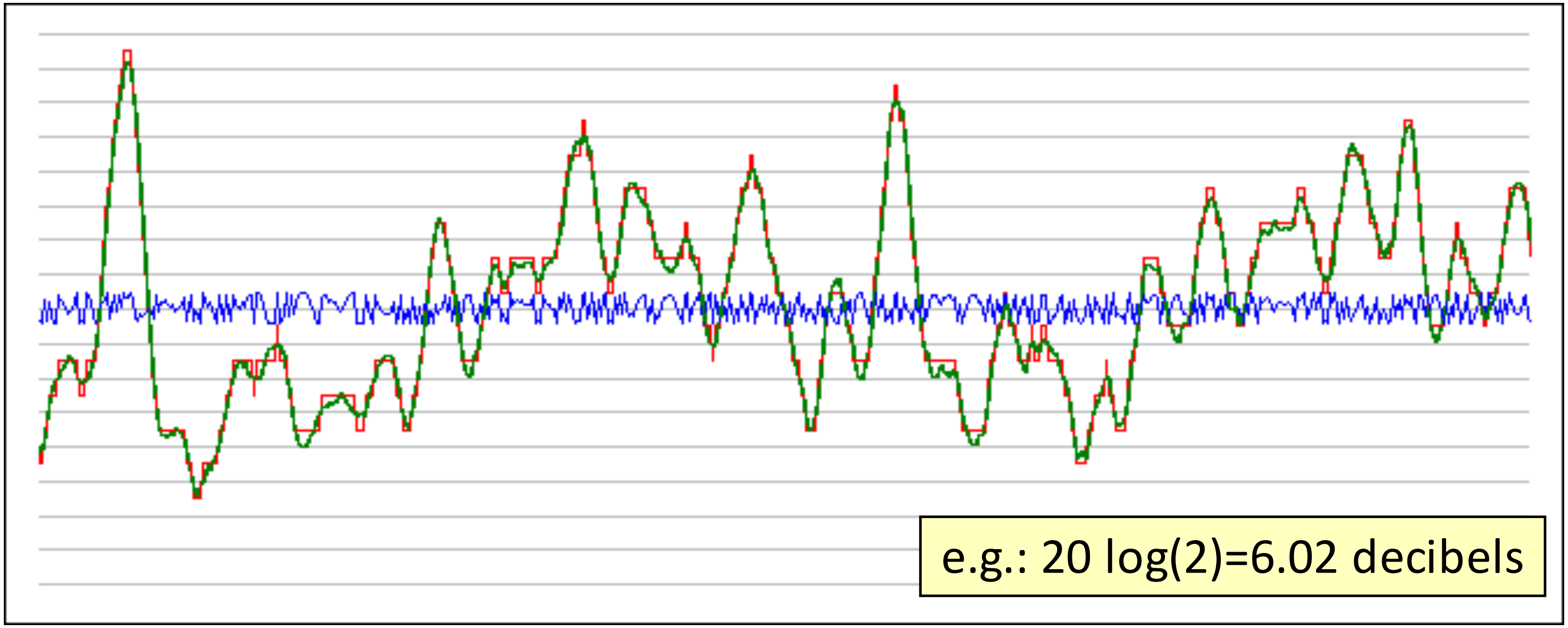


Quantization Noise (2)

- Assuming “rounding“ (truncating) towards 0



Quantization Noise for Audio Signal



$$\text{signal to noise ratio (SNR) [db]} = 20 \log \left(\frac{\text{effective signal voltage}}{\text{effective noise voltage}} \right)$$

- Signal to noise for ideal *n*-bit converter: *n* * 6.02 + 1.76 [dB]
  - *e.g.* 98.1 db for 16-bit converter, ~160 db for 24-bit converter
- Additional noise for converters that aren't ideal (all of them?)

Source: [http://www.beis.de/Elektronik/DeltaSigma/DeltaSigma.html]

© 2016 Marwedel, Meyer

Summary

- Embedded systems employ hardware in a loop
- Sensors
- Discretization
  - Definition of signals
  - Sample-and-hold circuits
    - Aliasing (and how to avoid it)
    - Nyquist criterion
  - A/D-converters
    - Flash-based
    - Successive approximation
    - Quantization noise

© 2016 Marwedel, Meyer

Next Time

- Embedded Information Processing
  - Chapter 3.3