Digital Speech and Audio Processing 2016-2017

Project 3: Real-time LPC vocoder

Vrije Universiteit Brussel

Realtime Audio processing: input-output handling strategies

Vrije Universiteit Brussel

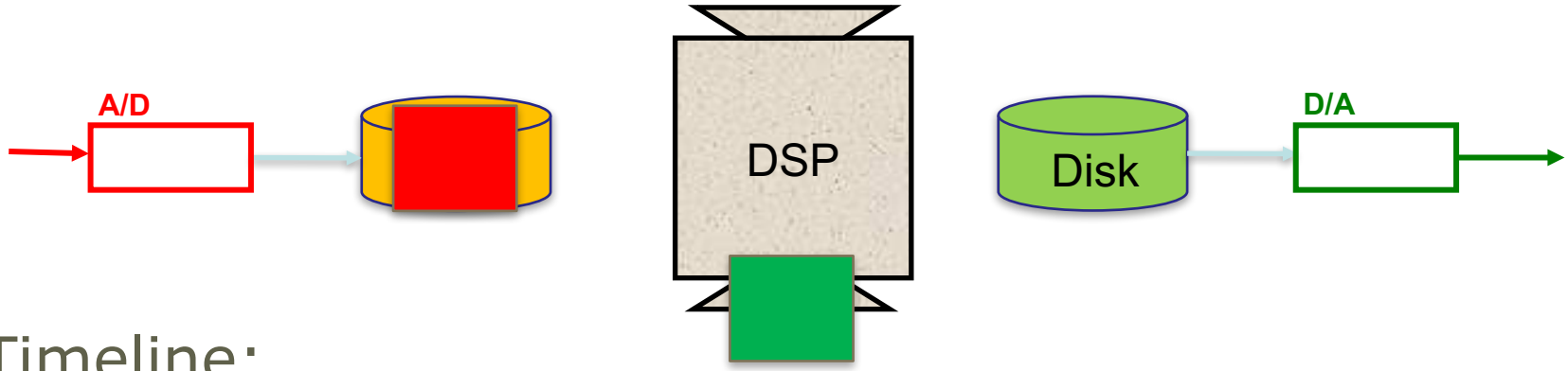
Realtime systems

1. File-based processing (typical Matlab use):

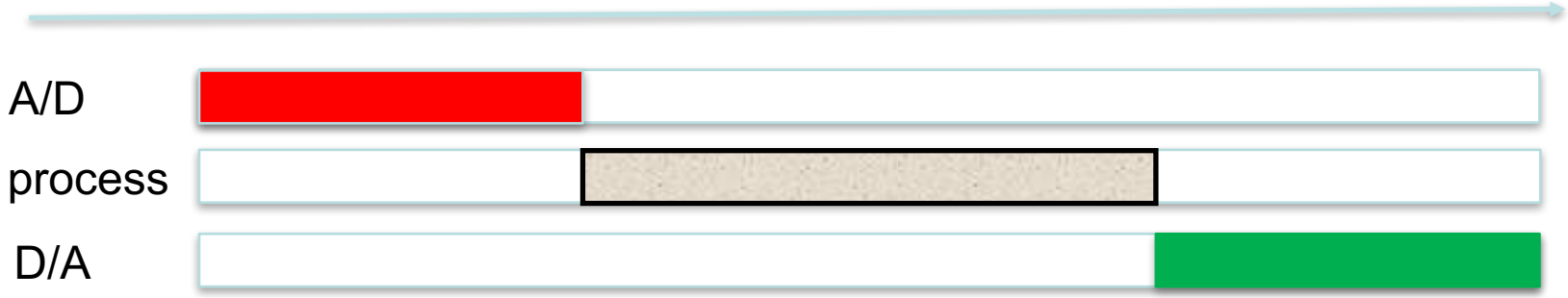
→ pre-recorded audio, full signal is known at processing time

Realtime systems

1. File-based processing (typical Matlab use):



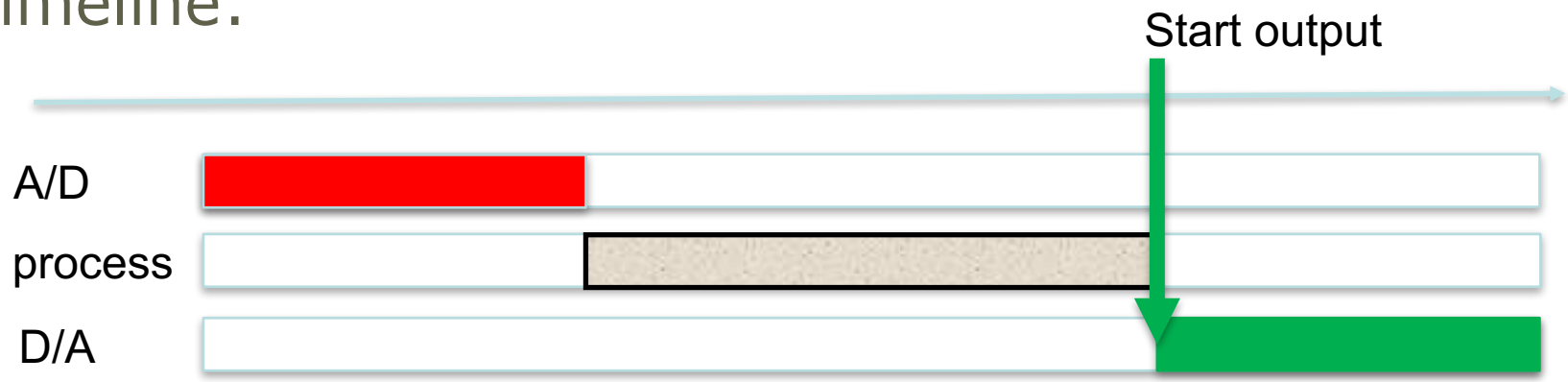
Timeline:



Realtime systems

1. File-based processing (typical Matlab use):

Timeline:



Realtime systems

1. File-based processing (typical Matlab use):

Pro

- All input data is available at processing time
- No time restriction on processing

Con

- Not realtime! What about cellphone filters? Echo cancellation?
- Very high latency (long delay) for all types of processing
- Memory requirements

Realtime systems

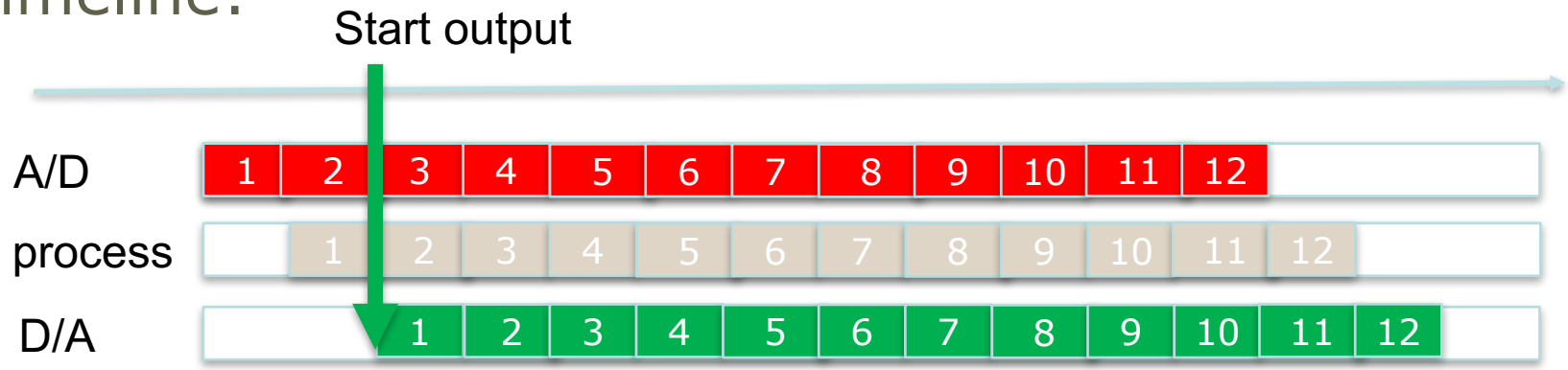
Sample-based processing:

→ Samples are immediately processed as they come in

Realtime systems

Sample-based processing:

Timeline:



Realtime systems

Sample-based processing:

Pro

- Low(est) I/O latency, realtime processing possible
- I/O operations require minimal amount of memory

Con

- Limited amount of processing time available (1 sample)
- Restricted to purely causal processing (no look-ahead)
- Some algorithms are only updated once every N samples, but this update only has 1 sample processing time.
- May lead to non-uniform processor load with short and high peaks ➔ inefficient use of processor

Realtime systems

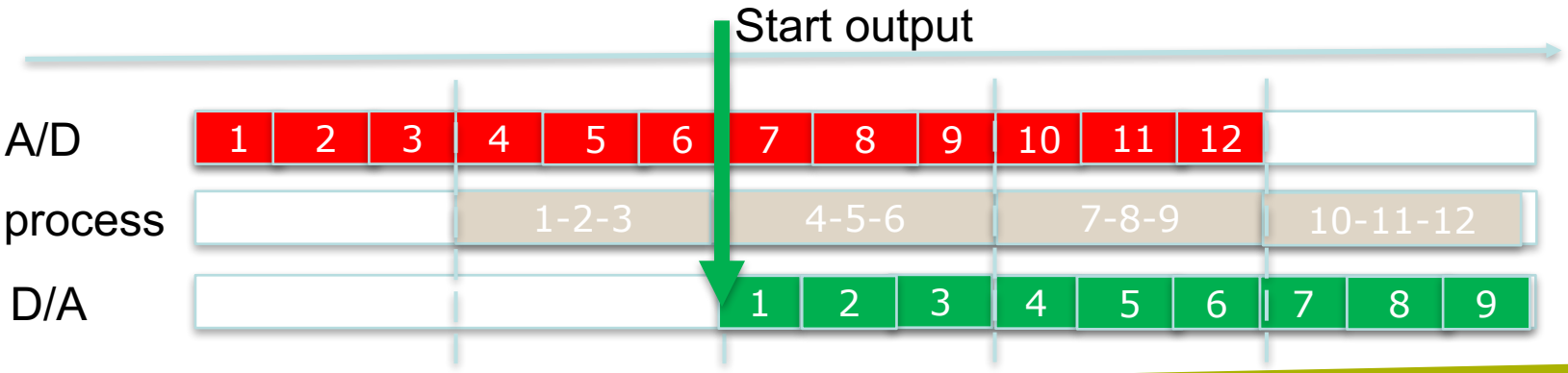
Block-based processing:

Samples are buffered and processed in blocks

Realtime systems

Block-based processing:

Timeline:



Realtime systems

Block-based processing:

Pro

- Low/scalable I/O latency, realtime processing possible
- I/O memory requirements are scalable
- Processing can be spread over multiple samples, non-uniform peak loads are decreased

Con

- I/O latency and memory requirements higher than for sample-based systems
- Limited amount of processing time available

Tir

A

p

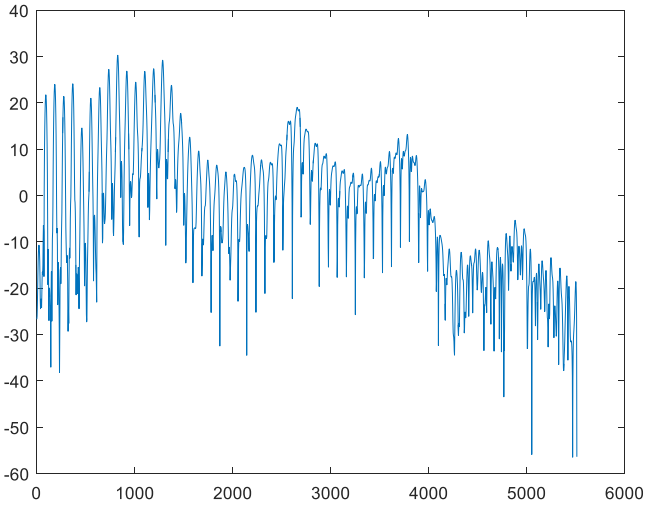
D/A

LPC-based vocoder

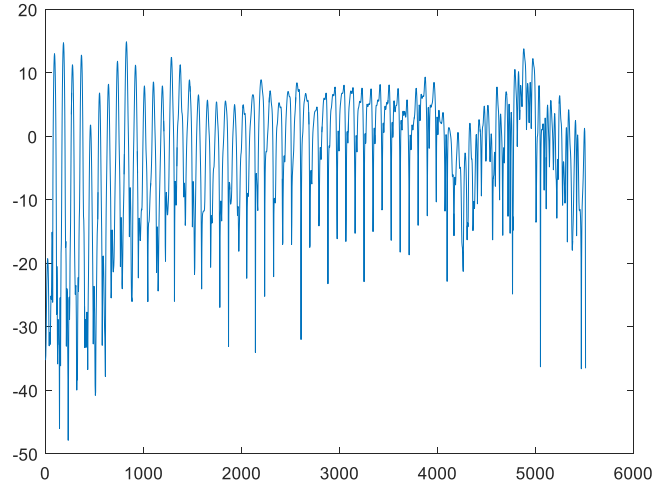
Vrije Universiteit Brussel

Linear Predictive Coding

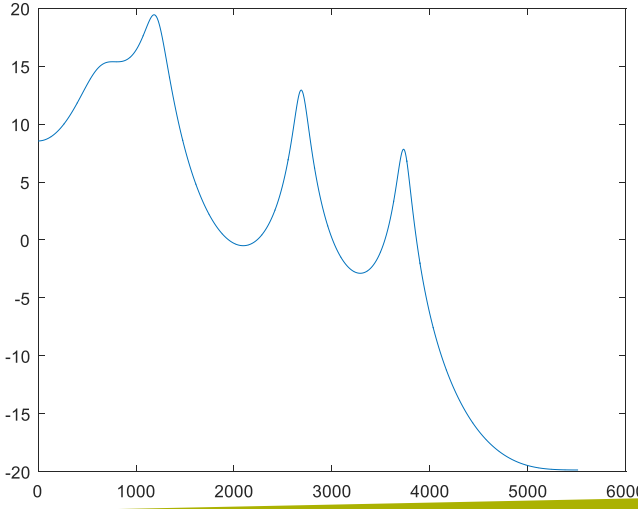
$S(f) - signal\ spectrum$



$E(f) - Source$



$H(f) - \text{filter / envelope}$



Linear Predictive Coding

- Signal $S(z) = E(z)\,H(z)$ (filtered version of source)
- Filter $H(z)$
  - $H(z) = \frac{1}{1+\sum_{i=1}^{O} a_i\, z^{-i}}$ -> $s(n) = \sum_{i=1}^{O} a_i\, s(n-i) + e(n)$
  - Determines spectral envelope (phoneme)
- Source $E(z)$
  - $e(n)$ = part of $s(n)$ that cannot be predicted with LP
  - Theoretically a periodic impulse train or white noise (more complex in reality)
  - Determines F0 (pitch)

LPC vocoder in music

- LPC analysis on speech
  - $H_{speech}(z)$, $E_{speech}(z)$
  - $g_{speech} = \sqrt{\sum e^2_{speech}(n)}$
- LPC analysis on instrument
  - $H_{instr}(z)$, $E_{instr}(z)$
  - $g_{instr} = \sqrt{\sum e^2_{instr}(n)}$
- $S_{out}(z) = E_{instr}(z) \frac{g_{speech}}{g_{instr}} H_{speech}(z)$
  - Apply spectral envelope of speech (phoneme) on instrument
  - Apply amplitude dynamics of speech on instrument
  - Keep pitch of instrument
  - Makes instrument "talk"

LPC vocoder in music

- Examples
  - Daft Punk – Instant Crush
    https://www.youtube.com/watch?v=a5uQMwRMHcs
  - Kavinsky – Nightcall
    https://www.youtube.com/watch?v=MV_3Dpw-BRY
  - Mogwai – Hunted by a freak
    https://www.youtube.com/watch?v=Pbx6J3kw_PI
  - Imogen Heap - Hide And Seek
    https://www.youtube.com/watch?v=UYIAfiVGluk
  - Neil Young - Transformer Man
    https://www.youtube.com/watch?v=eblFQppJfyg

Assignment

- Make matlab function

  $out = vocodeLPC(in_{speech}, in_{instr}, O_{LPC})$

  - out = output frame
  - in_{speech} = input speech frame (windowed)
  - in_{inst} = input instrument frame (windowed)
  - O_{LPC} = LPC order
- Use built-in matlab function lpc()
- Test offline using testVocoder.m
- Test real-time using rtVocoder.m
  - Matlab, directSound -> Latency!!!

LPC vocoder

- Example

Modulator	Carrier	Output
		

M83 - Starwaves