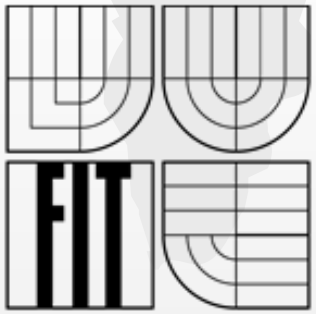
ZRE - Kódování řeči II.

# CELP

Vladimír Malenovský, ÚPGM FIT VUT Brno
malenov@fit.vutbr.cz



# Úvod

- CELP je nejpoužívanější technologie na kódování řeči v současnosti
- CELP dosahuje výrazně lepší kvality řeči při stejném bitovém toku jako jiné technologie, např. ADPCM, LPC nebo RELP
- Akronym CELP použili poprvé Manfred Schroeder a Bishnu Atal v roce 1985 v článku

Schroeder,M.R. and B. S. Atal (1985). ‘‘Code-Excited Linear Prediction (CELP): High-Quality Speech at Very Low Bit Rates,’’ IEEE ICASSP, pp. 2511–2514.

# Úvod

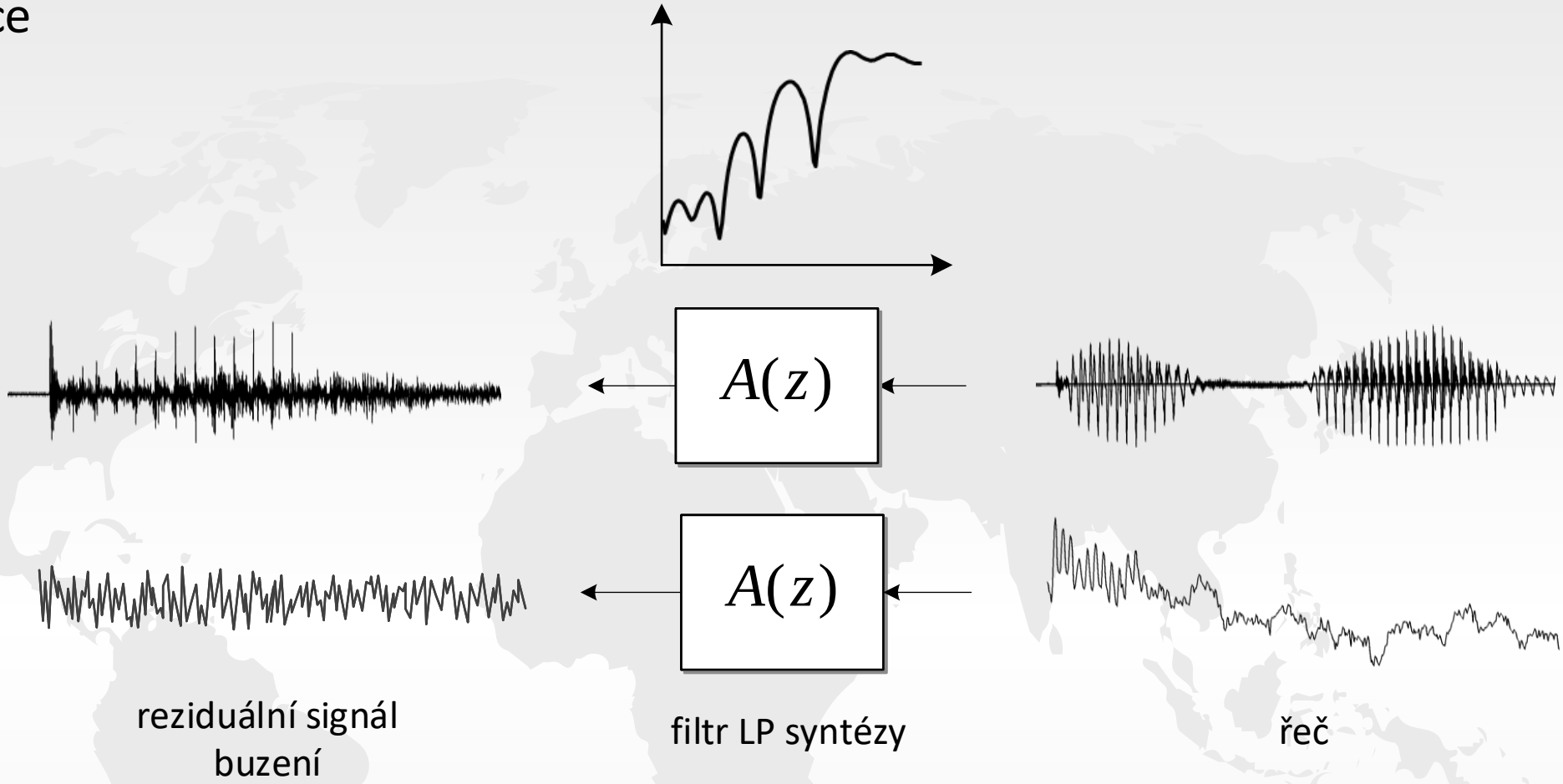
- nedostatky “tehdy nejlepšího” kodeku LPC-10
  - příliš zjednodušené kódování buzení - pouze 2.4 kbps
  - striktní rozlišení znělé a neznělé řeči - robotický hlas
  - fáze vstupního signálu není zachována

- hlavní „nové“ myšlenky CELPu:
  - dlouhodobý prediktor (LTP) –
    znělou řeč lze krásně předpovědět i ze vzdálené minulosti
  - kódování obou složek buzení naráz, znělé i neznělé –
    lidé přece neříkají buď „a“ nebo „s“
  - koncept analýzy syntézou (analysis-by-synthesis approach) –
    učení se z vlastní chyby
  - perceptuální váhování – co ucho neslyší, to kodek nekóduje

- nové kodeky:
  FS 1016 (CELP)  GSM-FR (RPE LTP)  GSM-HR (VSELP)

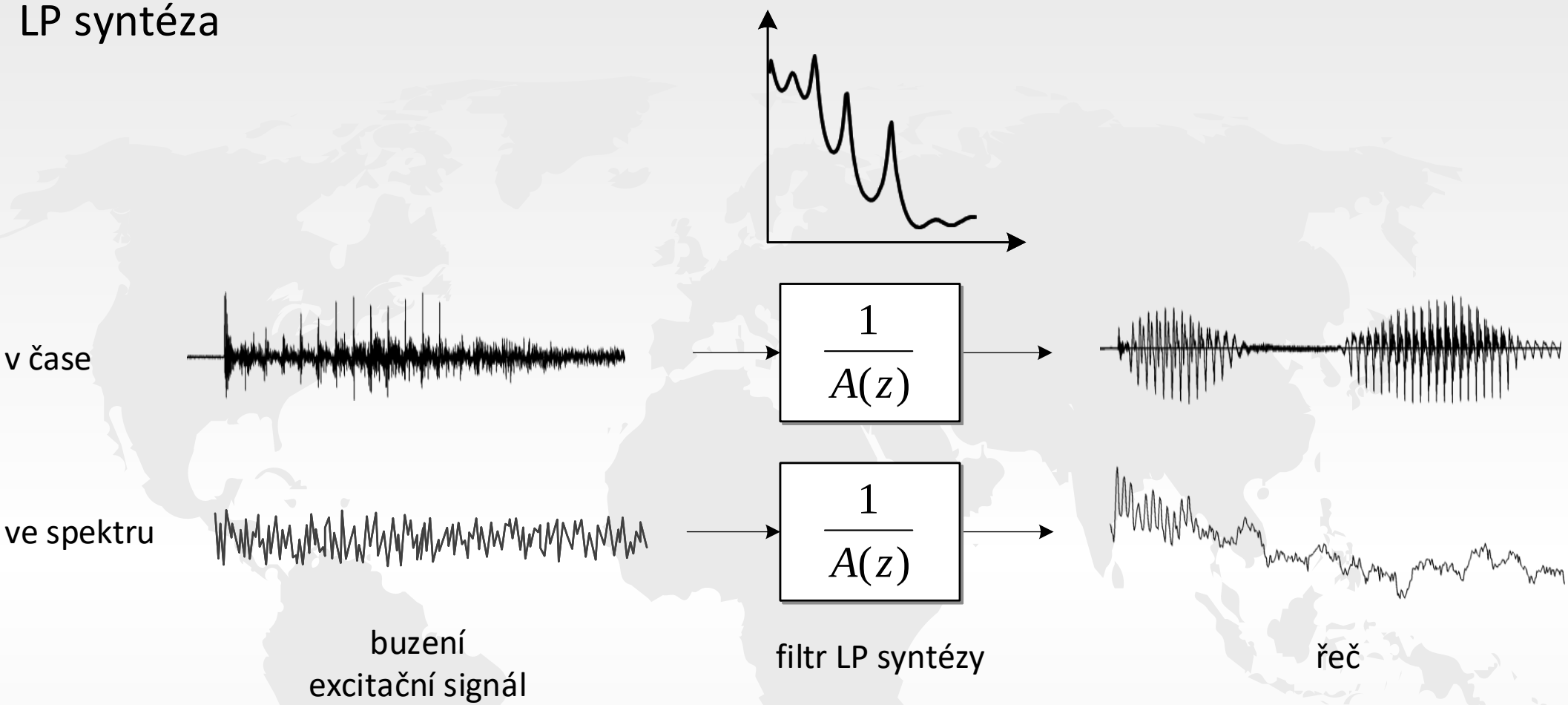
# LP model tvorby řeči

# LP filtrace



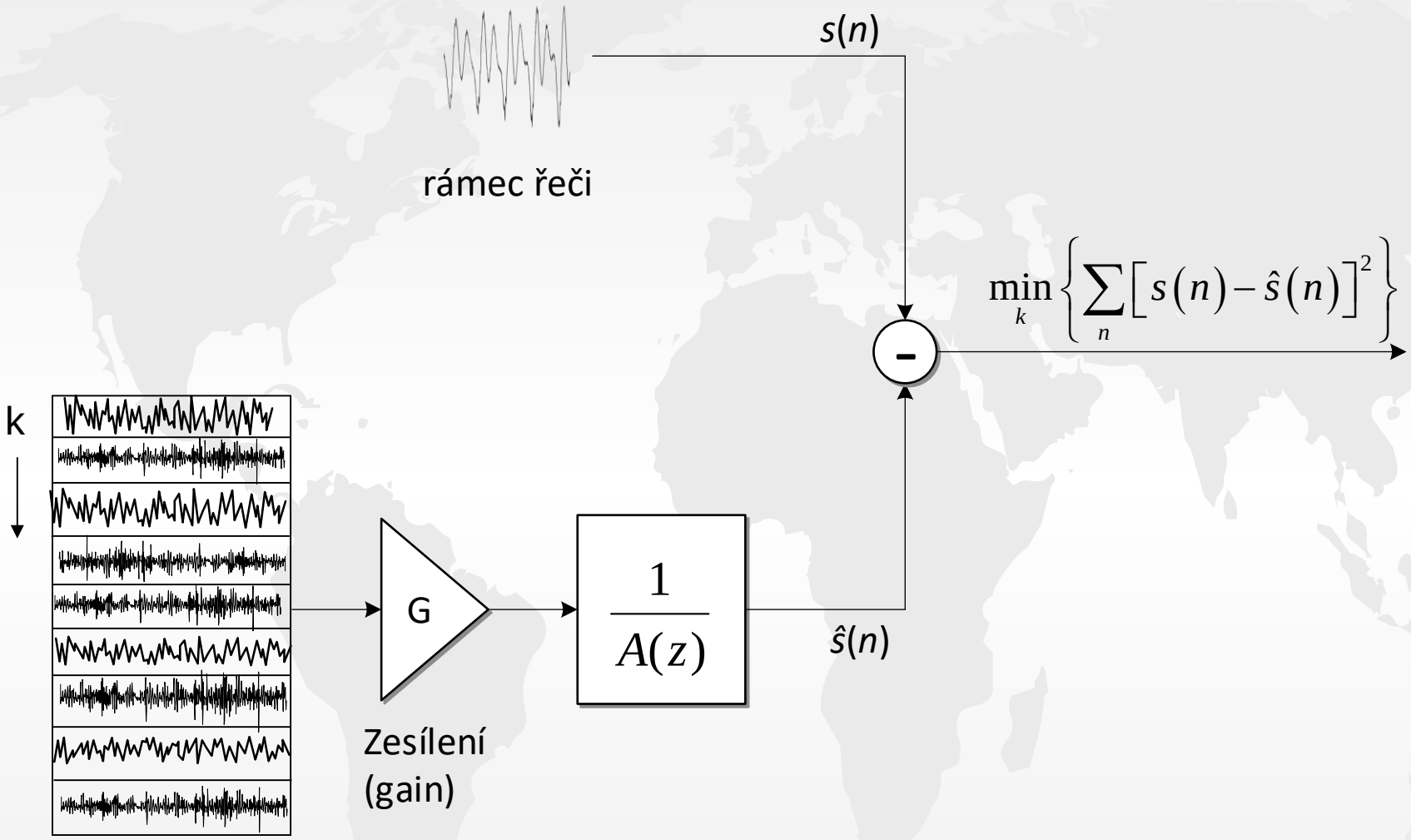

- Filtrace vstupního signálu filterm A(z) odstraňuje z řečového signálu jeho vlastní „obálku“ a tím zbavuje spektrum vlastních formantů
- Reziduální signál (buzení) je to co po této operaci zbyde a to se dále kóduje

# LP syntéza




- téměř všechny řečové kodeky pro veřejné komunikace v dnešní době používají LP model
- filtr LP syntézy je filterm typu IIR, takže má vlastní paměť, která odpovídá několika posledním vzorkům z minulosti řeči
- filtr LP syntézy má pouze póly, t.j. umí modelovat pouze spektrální „špičky“, nikoliv „zářezy“
- vzhledem k charakteru filtru může dojít k jeho nestabilitě a „explozi“ syntézy

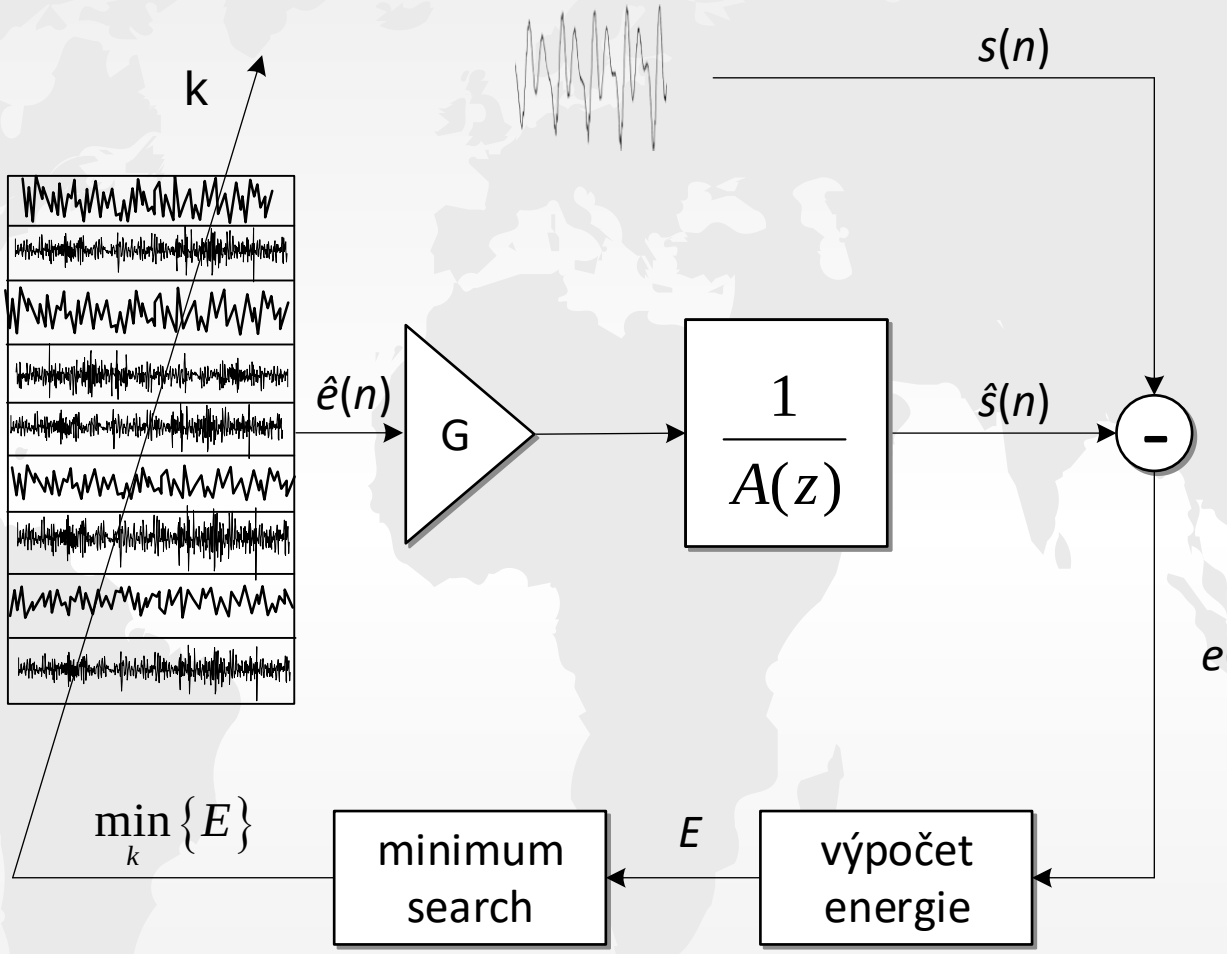
# CELP – princip kodéru



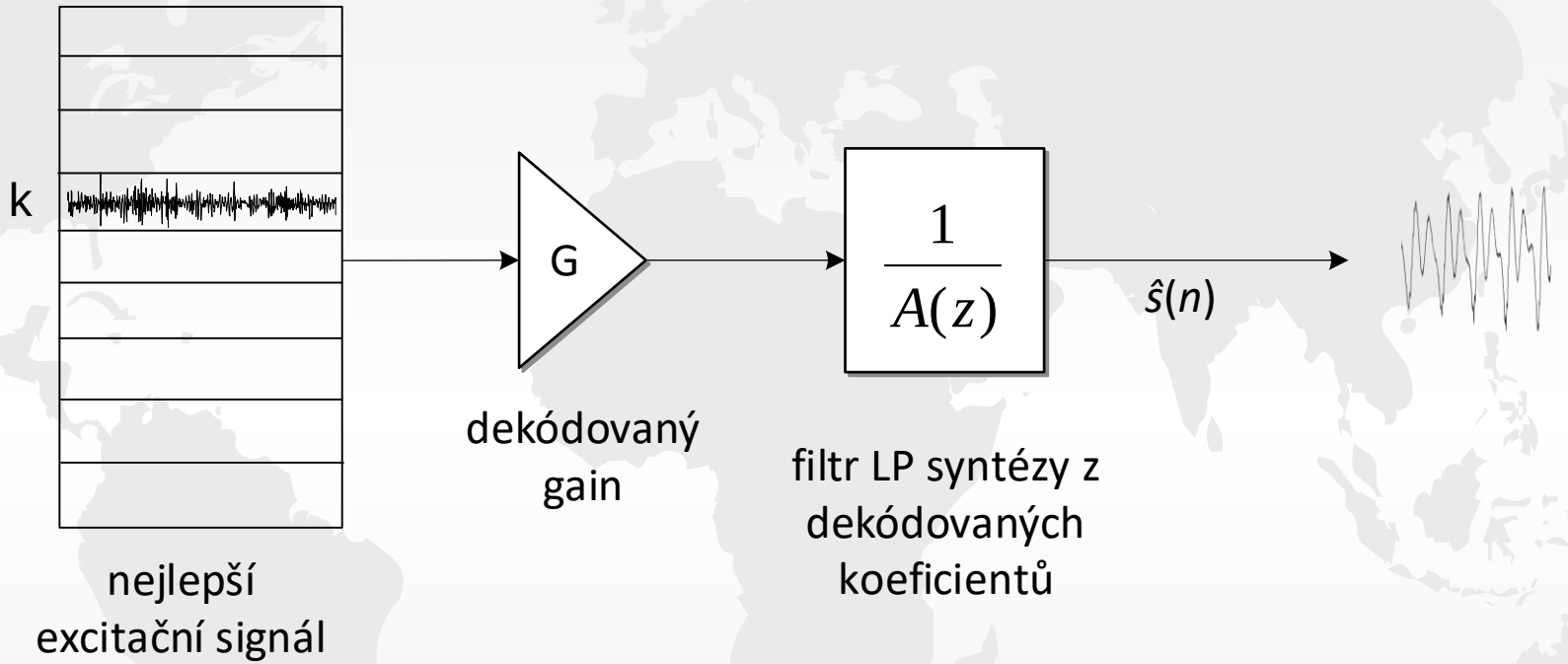

- Snažíme se minimalizovat chybu výběrem vhodného excitačního signálu a zesílení

# Koncept analýzy syntézou

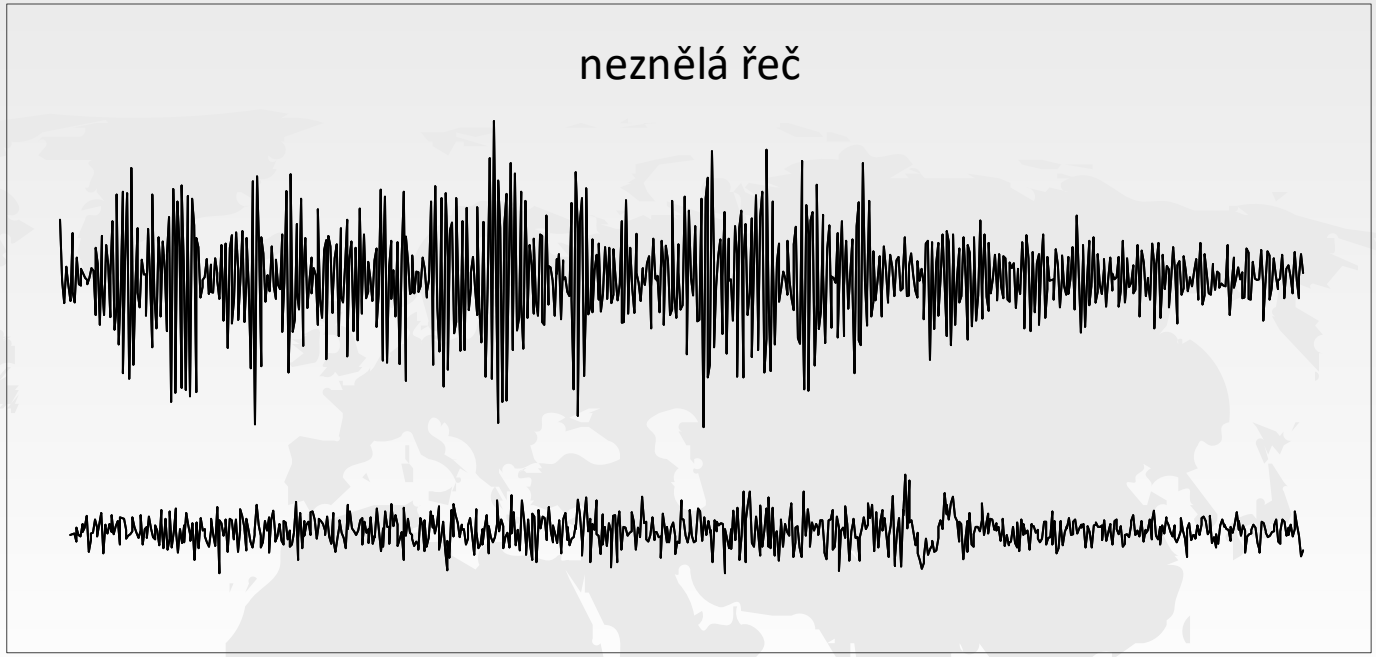
překreslené schéma

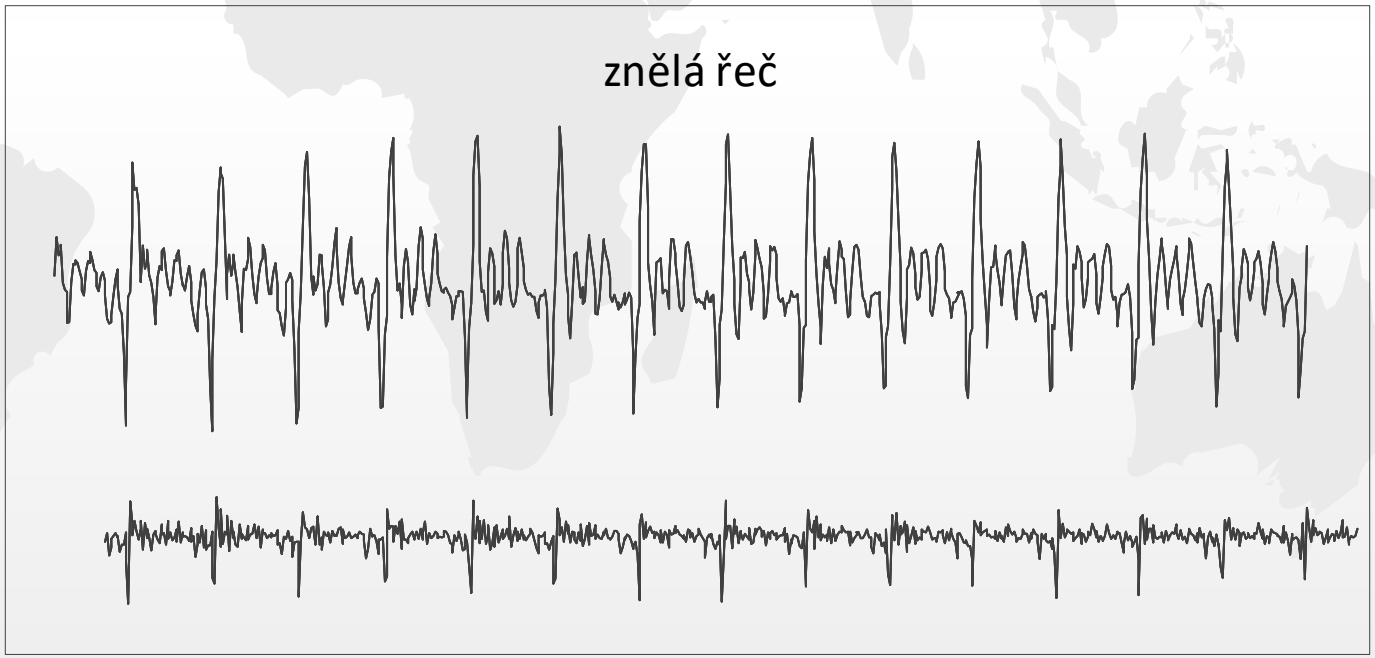


# CELP – princip dekodéru



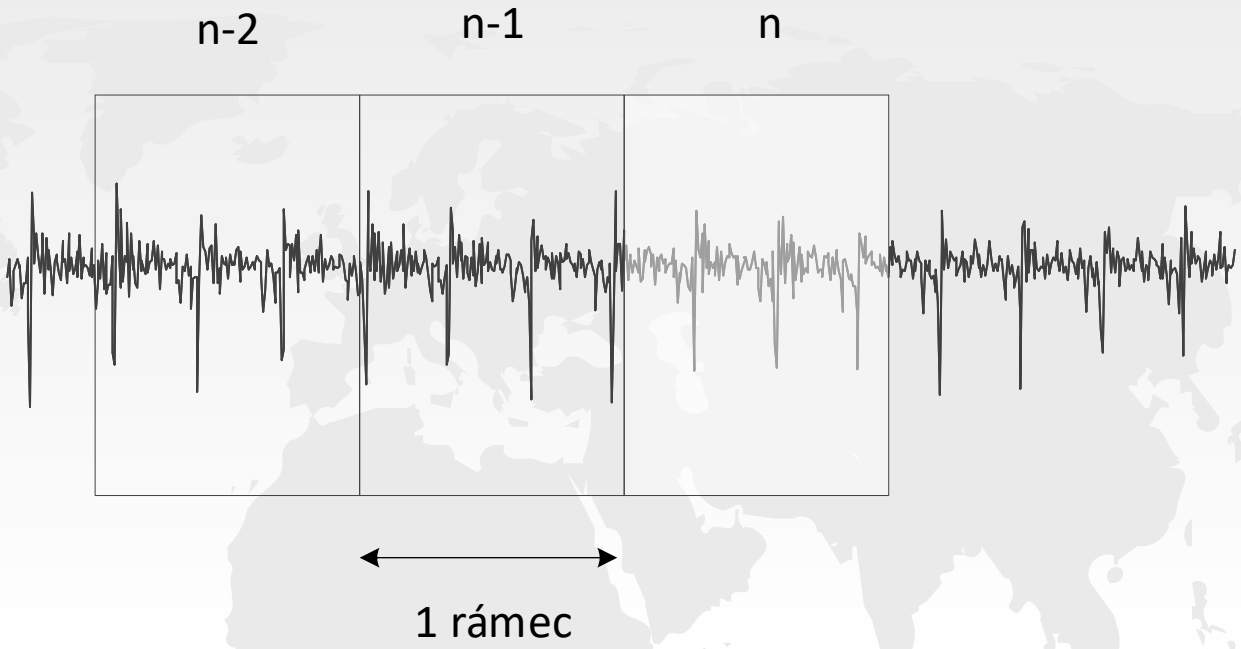
# Reziduální signál




periodicita v LP reziduu

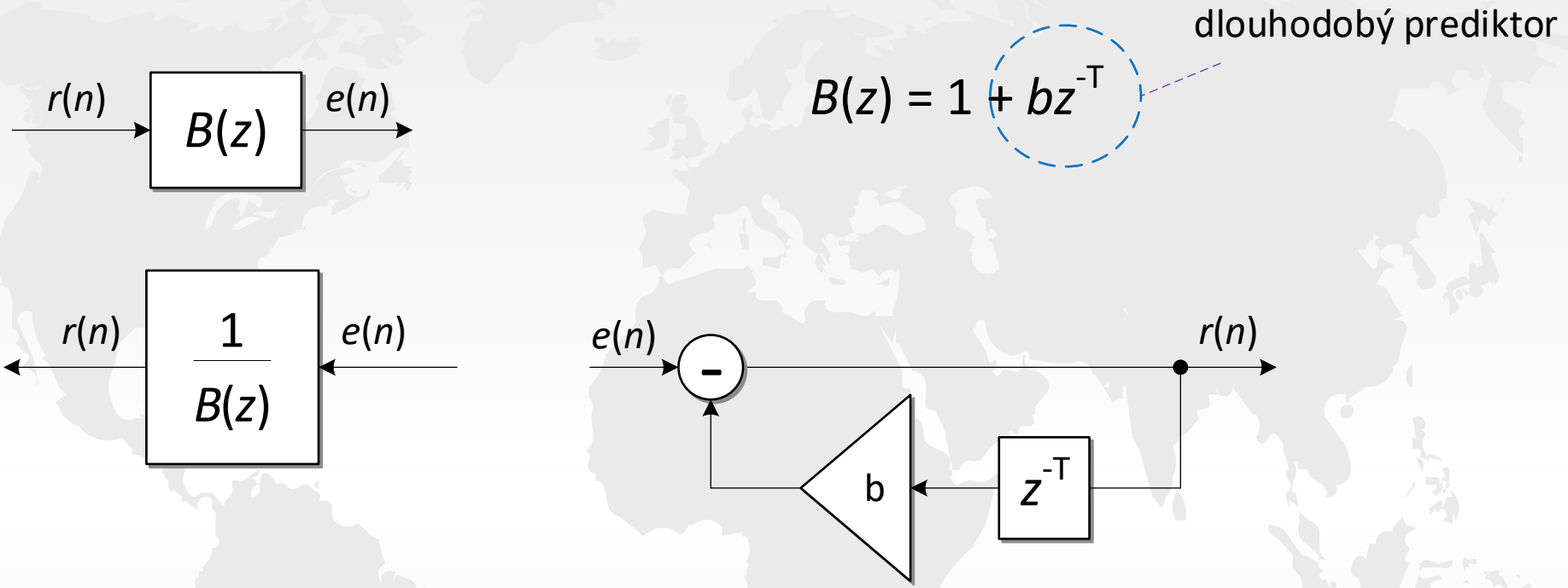
# Reziduální signál




- reziduální signál v současném rámci lze predikovat z minulosti
- u znělého signálu se úseky „opakují“ v periodách odpovídajících délce základního tónu
- můžeme toho využít a kódovat pouze rozdíl mezi současným a predikovaným signálem
- ale pozor, k predikci musíme použít již zakódovaný (přenesený) rez. signál a ne originál, protože jinak by enkodér a dekodér nepracovali se stejnými signály

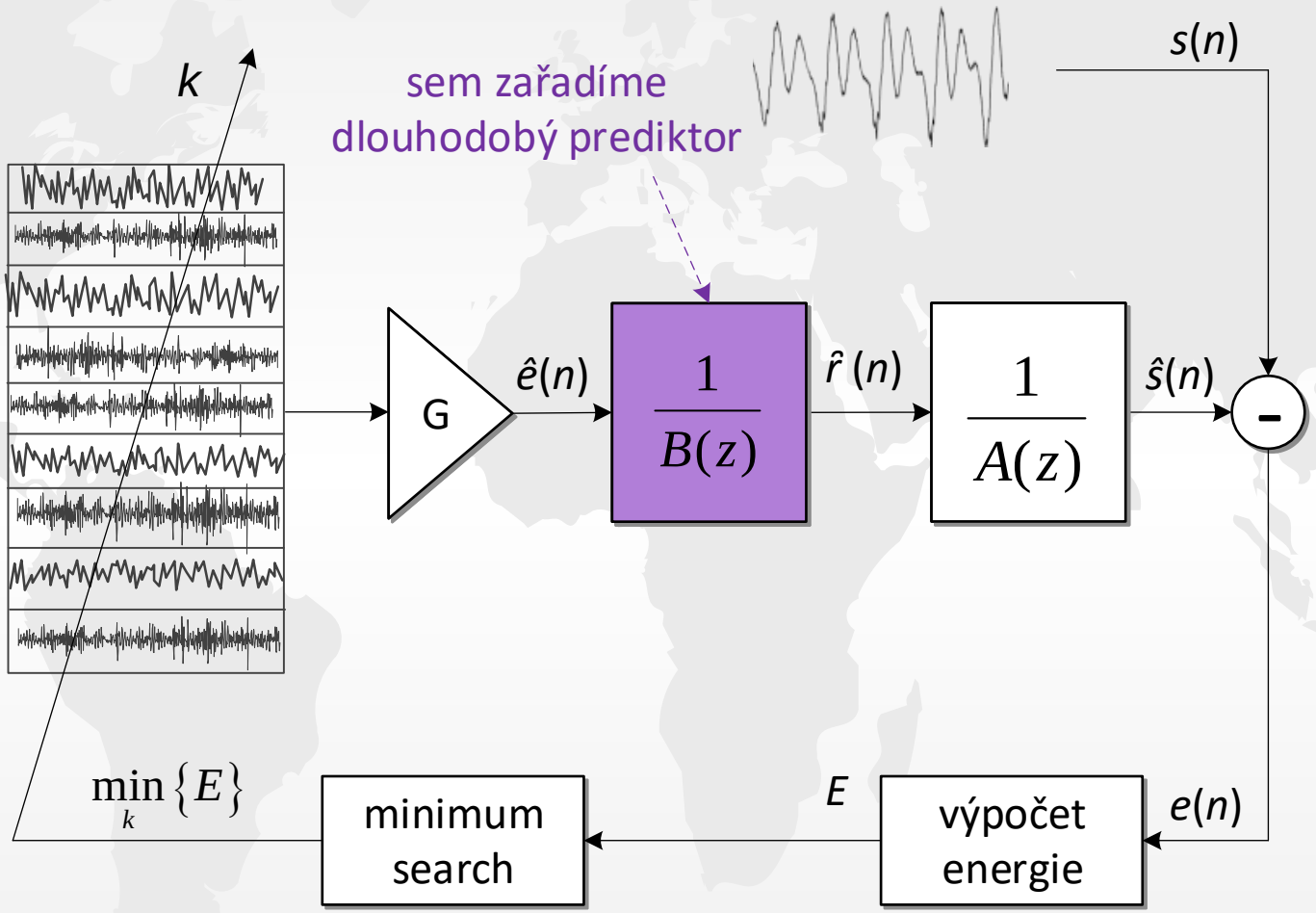
# Dlouhodobý prediktor (LTP)

# Dlouhodobý prediktor (Long-Term Predictor)



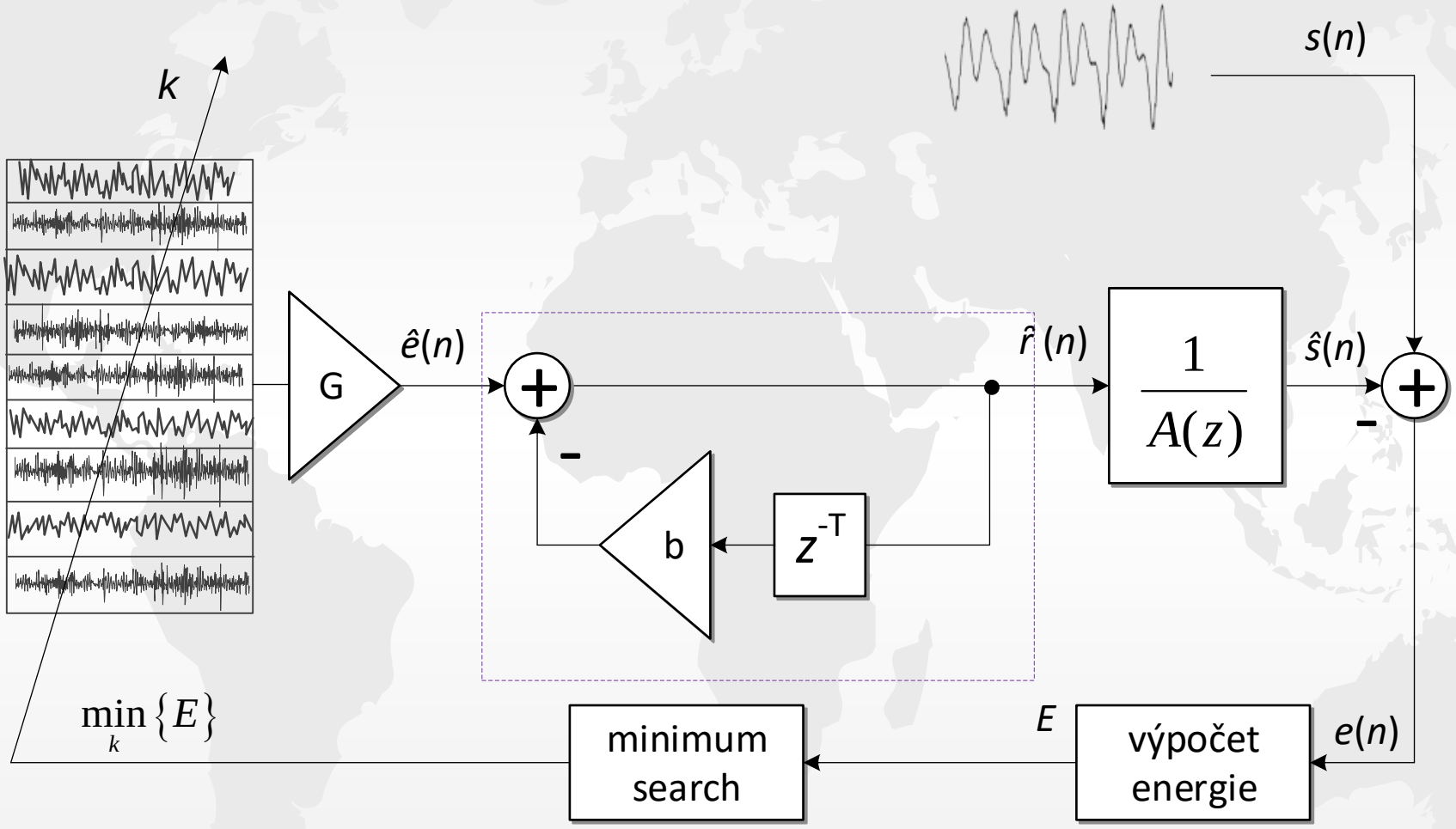

- dlouhodobý prediktor předpovídá současný vzorek signálu $r(n)$ ze vzorku vzdáleného o $T$ v minulosti, t.j. $r(n-T)$
- chybový signál: $e(n) = r(n) + br(n-T)$
- zpětná operace: $r(n) = e(n) - br(n-T)$
- říká se mu „dlouhodobý“, protože prediktuje ze vzorků vzdálených až 20ms, zatímco krátkodobý prediktor (LP) predikuje ze vzorků vzdálených ~2ms

# Dlouhodobý prediktor (LTP)

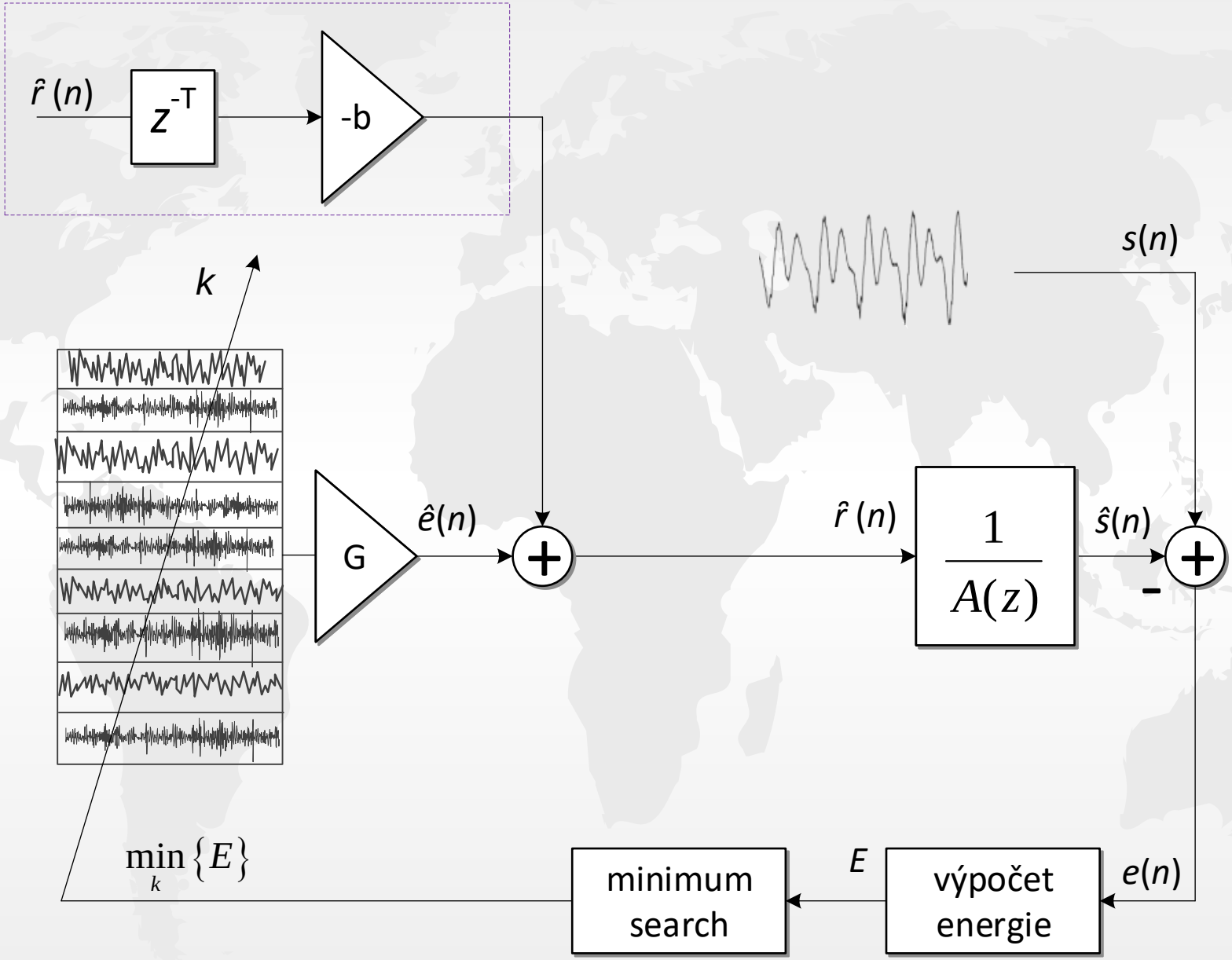


# Dlouhodobý prediktor (LTP)

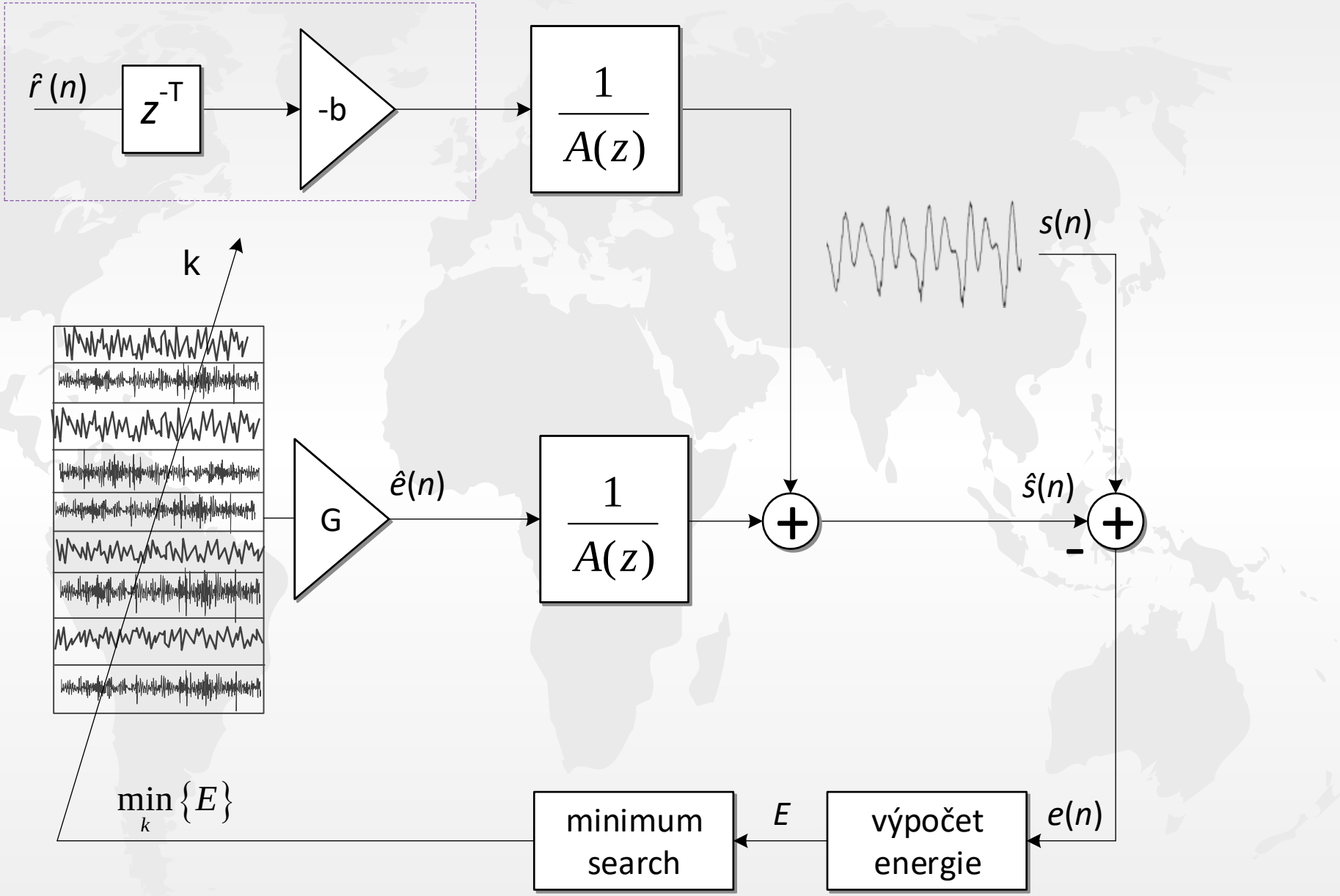
překreslené schéma



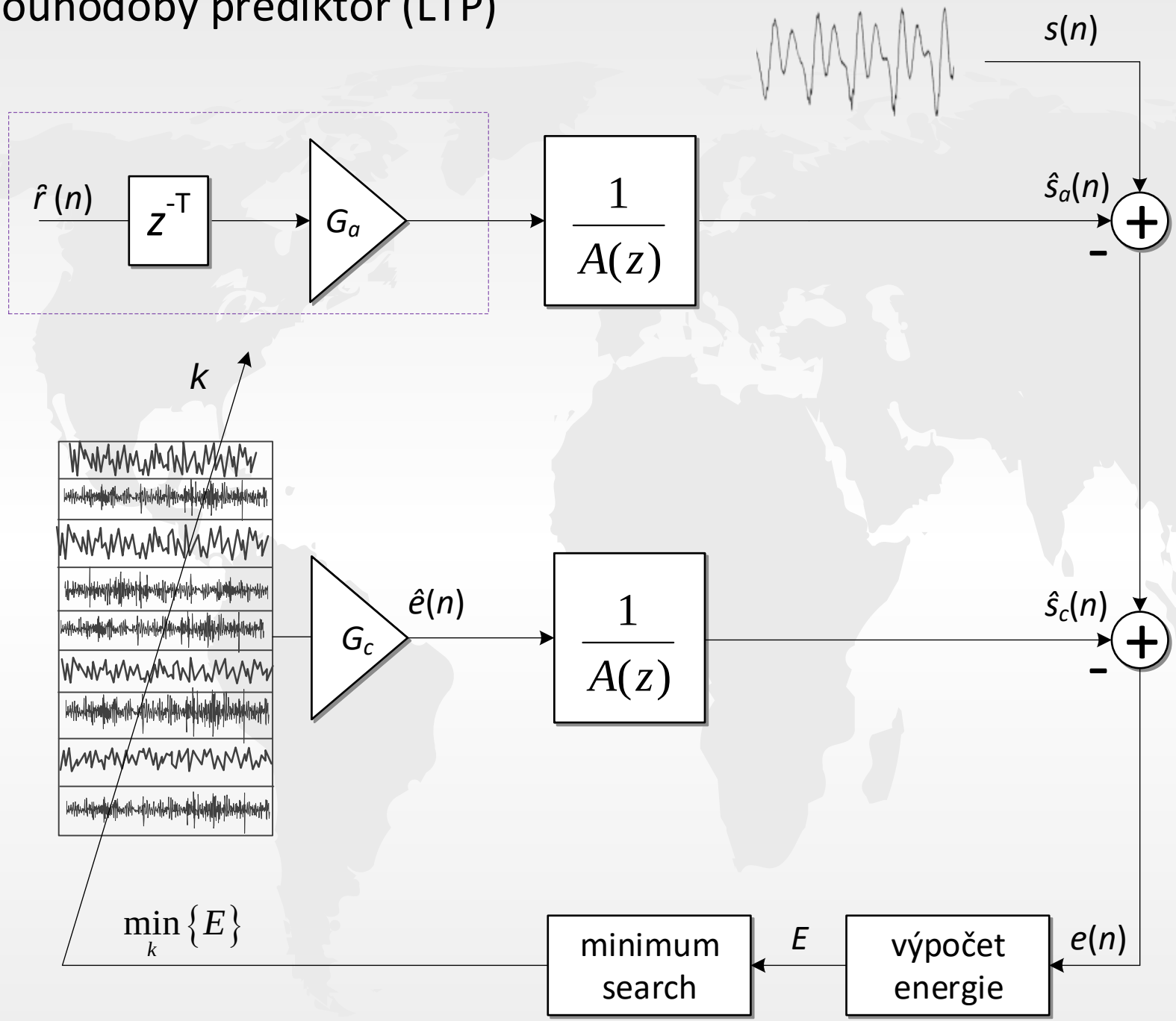
# Dlouhodobý prediktor (LTP)



# Dlouhodobý prediktor (LTP)

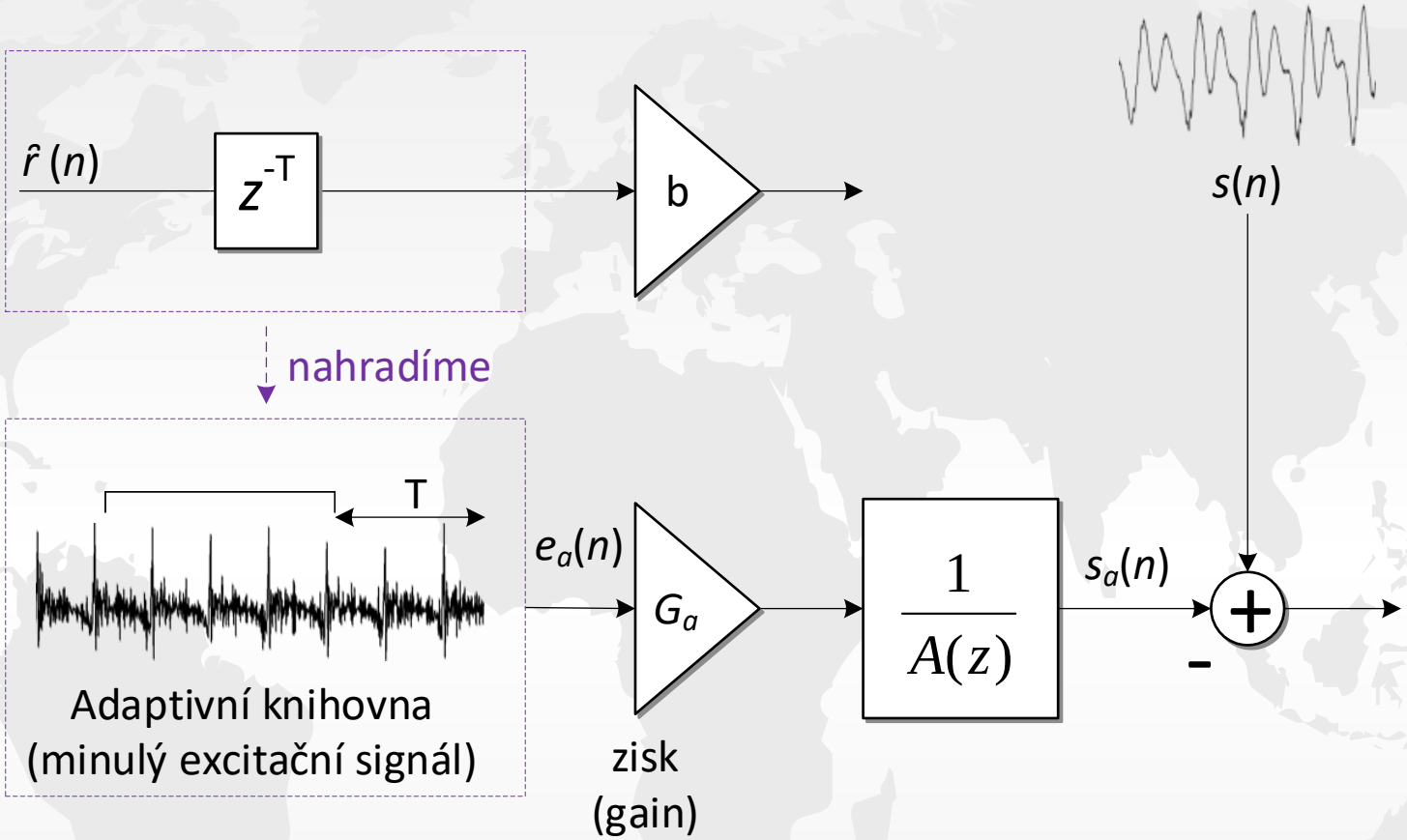


# Dlouhodobý prediktor (LTP)



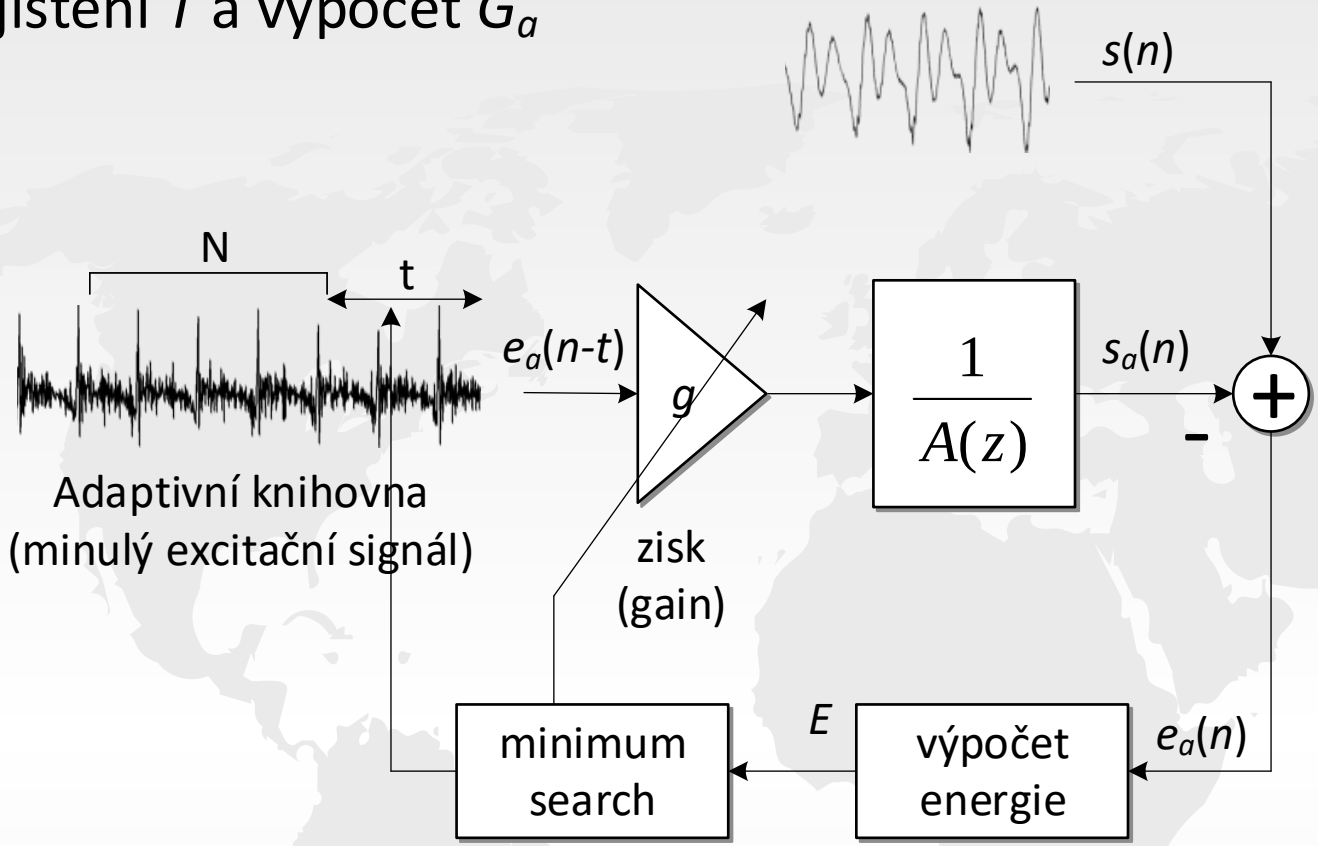

- přejmenování
  *-b* -> $G_a$
  G -> $G_c$
  $\hat{s}(n)$ -> $\hat{s}_c(n)$

# Zavedení adaptivní knihovny




- dlouhodobý prediktor lze nahradit tzv. adaptivní knihovnou, což je v podstatě minulý excitační signál (pozn. $r(n)$ je LP reziduum, $\hat{r}(n)$ je excitace)

# Zjištění $T$ a výpočet $G_a$




Iterativní postup:

- zvolíme $t$ a vybereme vektor $e_a(n-t)$
- vypočítáme $g$

$$g = \frac{-\sum_{n=1}^{N} r(n)e_a(n-t)}{\sum_{n=1}^{N} e_a^2(n-t)}$$

- vypočítáme $s_a(n)$

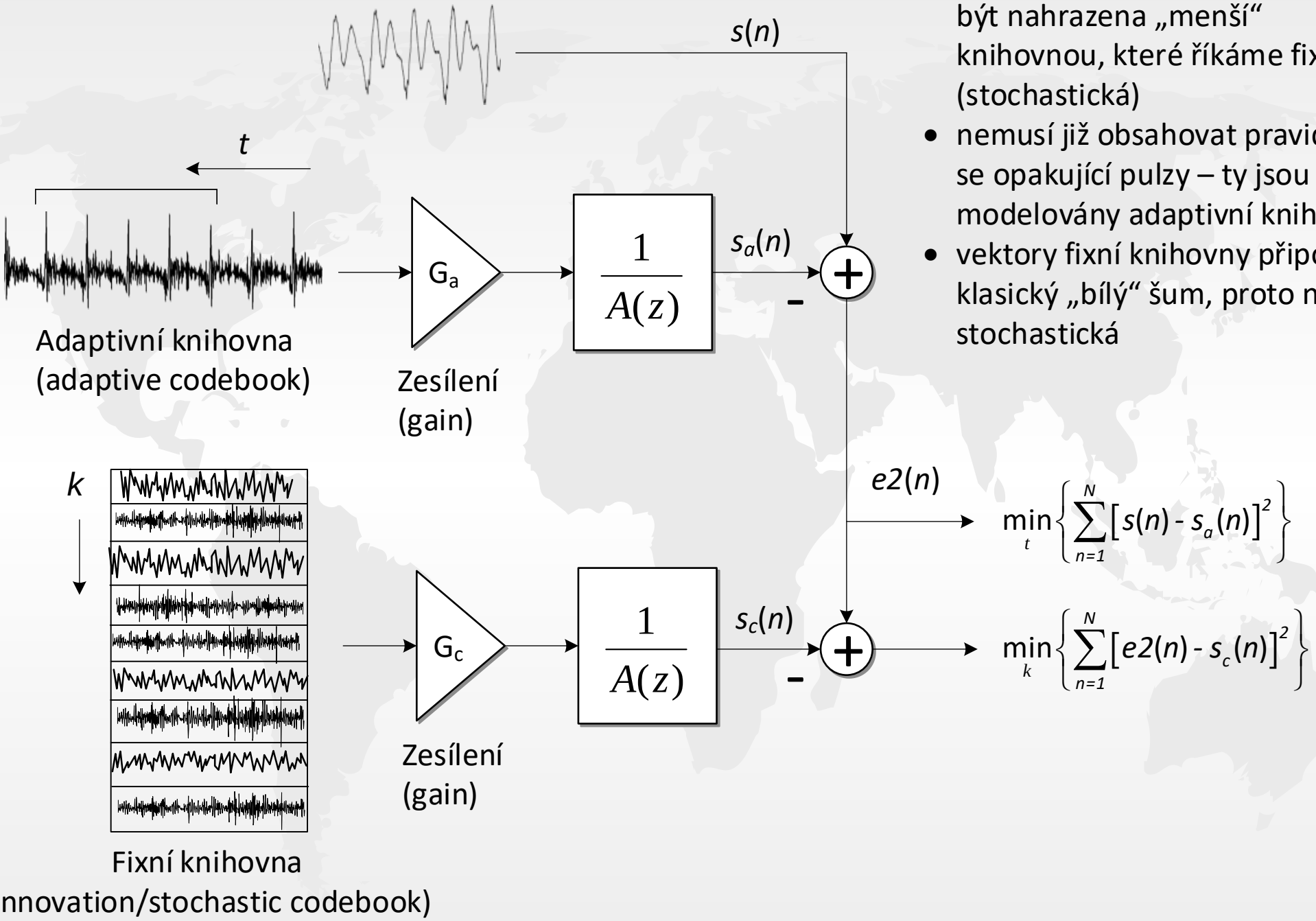
$$s_a(n) = ge_a(n-t) - \sum_{m=1}^{M} a_m s_a(n-m)$$

- vypočítáme $E$

$$E = \sum_{n=1}^{N} \left[ s(n) - s_a(n) \right]^2$$

- parametry $T$ a $G_a$ najdeme tak, abychom minimalizovali chybu mezi $s(n)$ a $s_a(n)$
- iterativně „prohledáváme“ adaptivní knihovnu a testujeme $N$-dlouhé úseky minulé excitace různě vzdálené od počátku ($n$=0)
- abychom ušetřili počet nutných iterací, prohledáváme pouze v okolí základního tónu $T_0$, který jsme zjistili předem
- pro každé testované $t$ dopočítáme analyticky optimální hodnotu gainu $g$
- gain $g$ se prozatím nekóduje, později ho totiž budeme přepočítávat

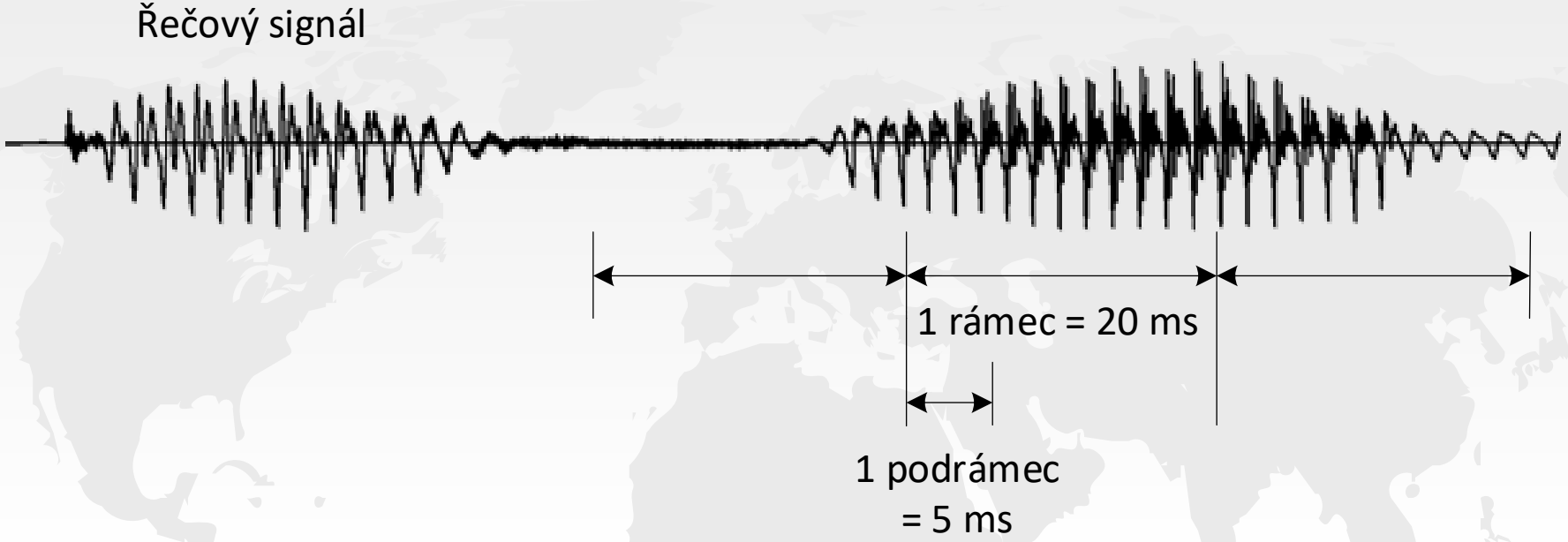
# Fixní (stochastická) knihovna




- původní „obří“ knihovna může být nahrazena „menší“ knihovnou, které říkáme fixní (stochastická)
- nemusí již obsahovat pravidelně se opakující pulzy – ty jsou modelovány adaptivní knihovnou
- vektory fixní knihovny připomínají klasický „bílý“ šum, proto název stochastická

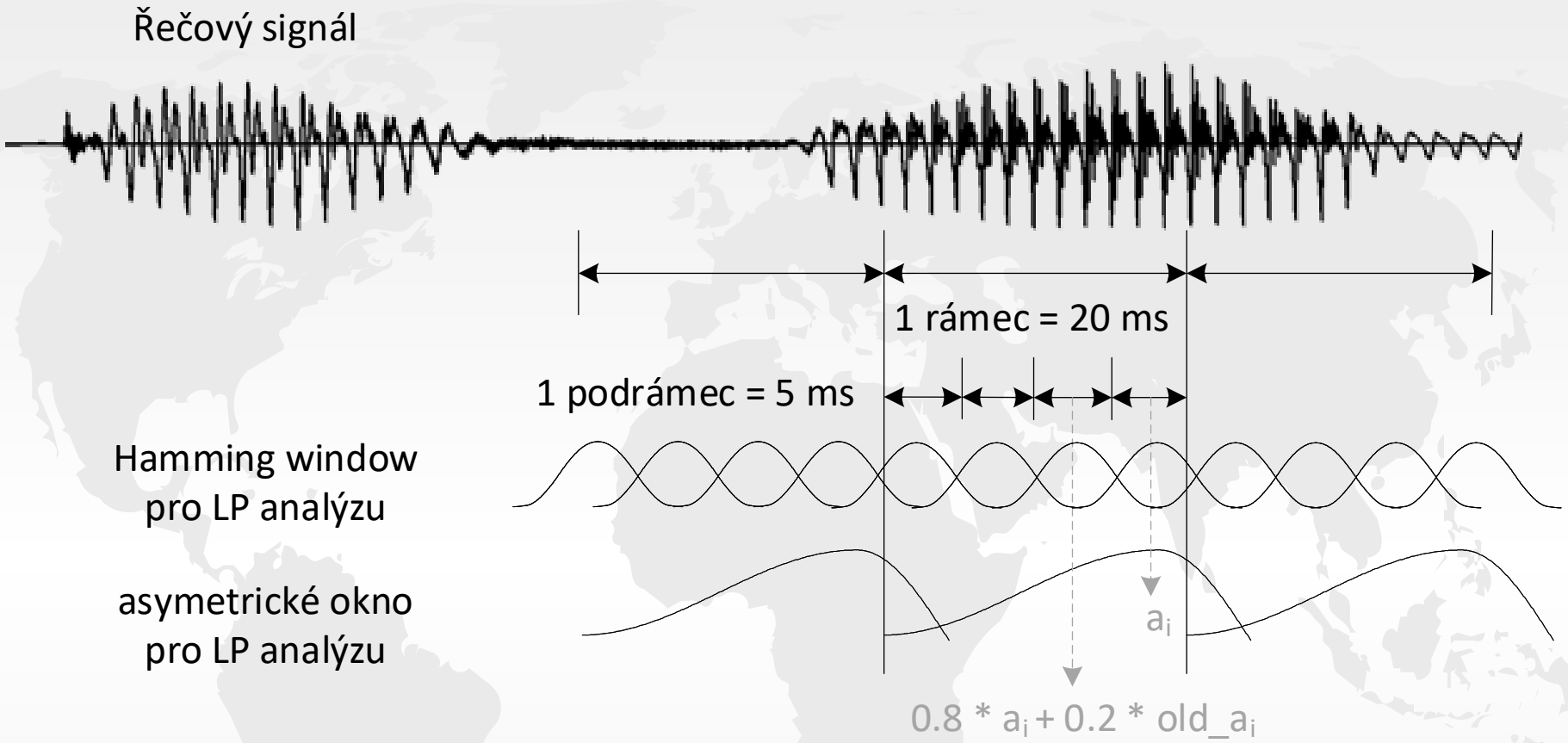
# Rámce a podrámce

# Rámce a podrámce




- klasické rámce o délce 15-25ms jsou pro některé parametry (LP koeficienty, $T_0$) příliš dlouhé
- lidé totiž mění polohu vokálního traktu nebo základní tón někdy mnohem rychleji
- řešením je tedy zkrácení rámce např. na 5ms (podrámec) pouze pro výpočet těchto parametrů
- ale pozor, např. LP analýza je docela komplexní a dělat ji 4x v jednom rámci stojí hodně MIPS

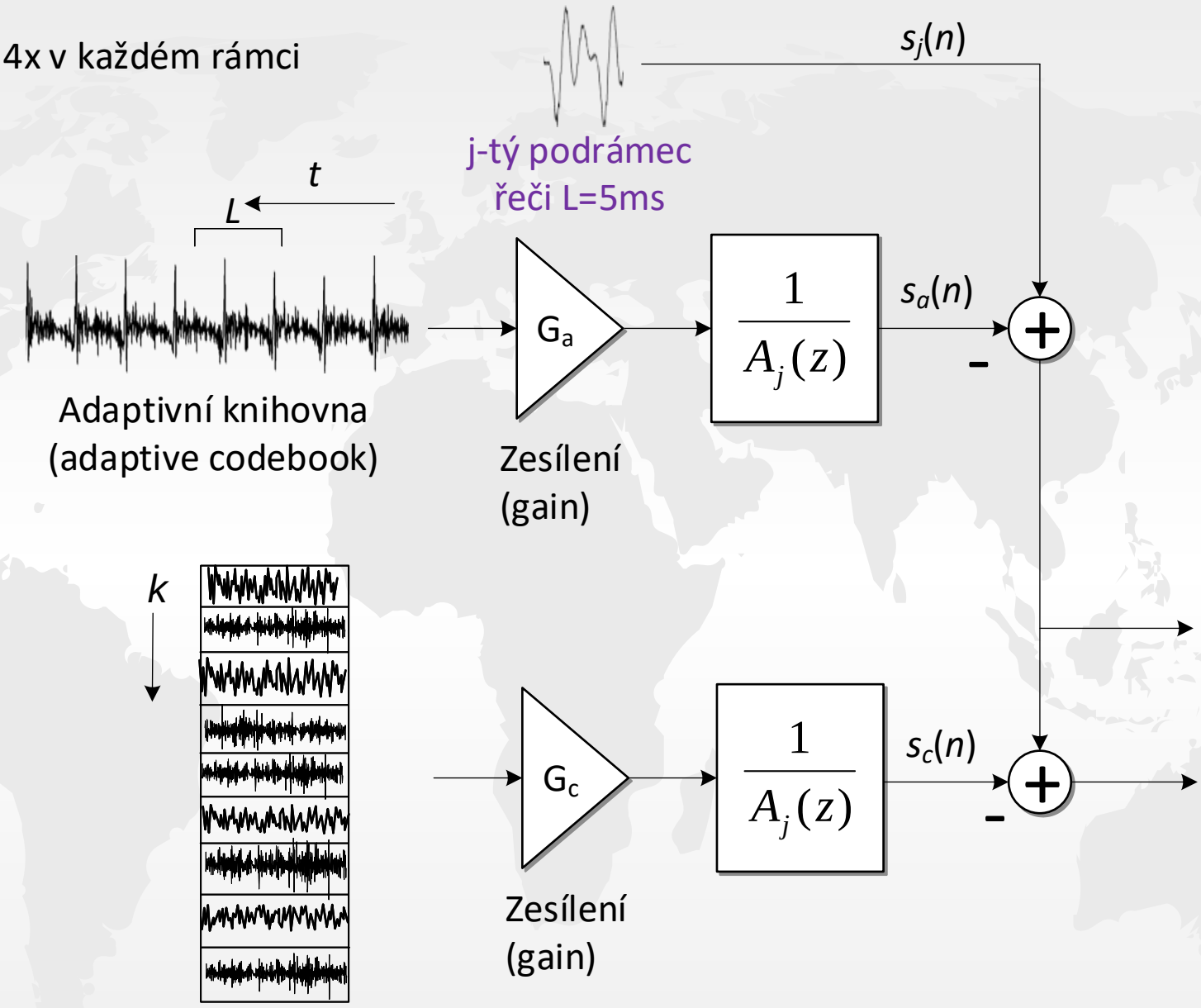
# interpolace LP koeficientů v podrámcích




- lepší nápad – budeme dělat LP analýzu jen jednou, např. na konci rámce a 3x interpolovat mezi starými a novými koeficienty LP filtru
- k tomu ale potřebujeme jiný tvar okna -> asymetrické okno
- asymetrické okno musí co nejméně zasahovat do „budoucího“ rámce, protože kodek musí na tyhle vzorky „čekat“
- vzorky „budoucího“ rámce nazýváme LOOKAHEAD (způsobuje zpoždění kodeku, tzv. algorithmic delay)

# CELP v podrámcích

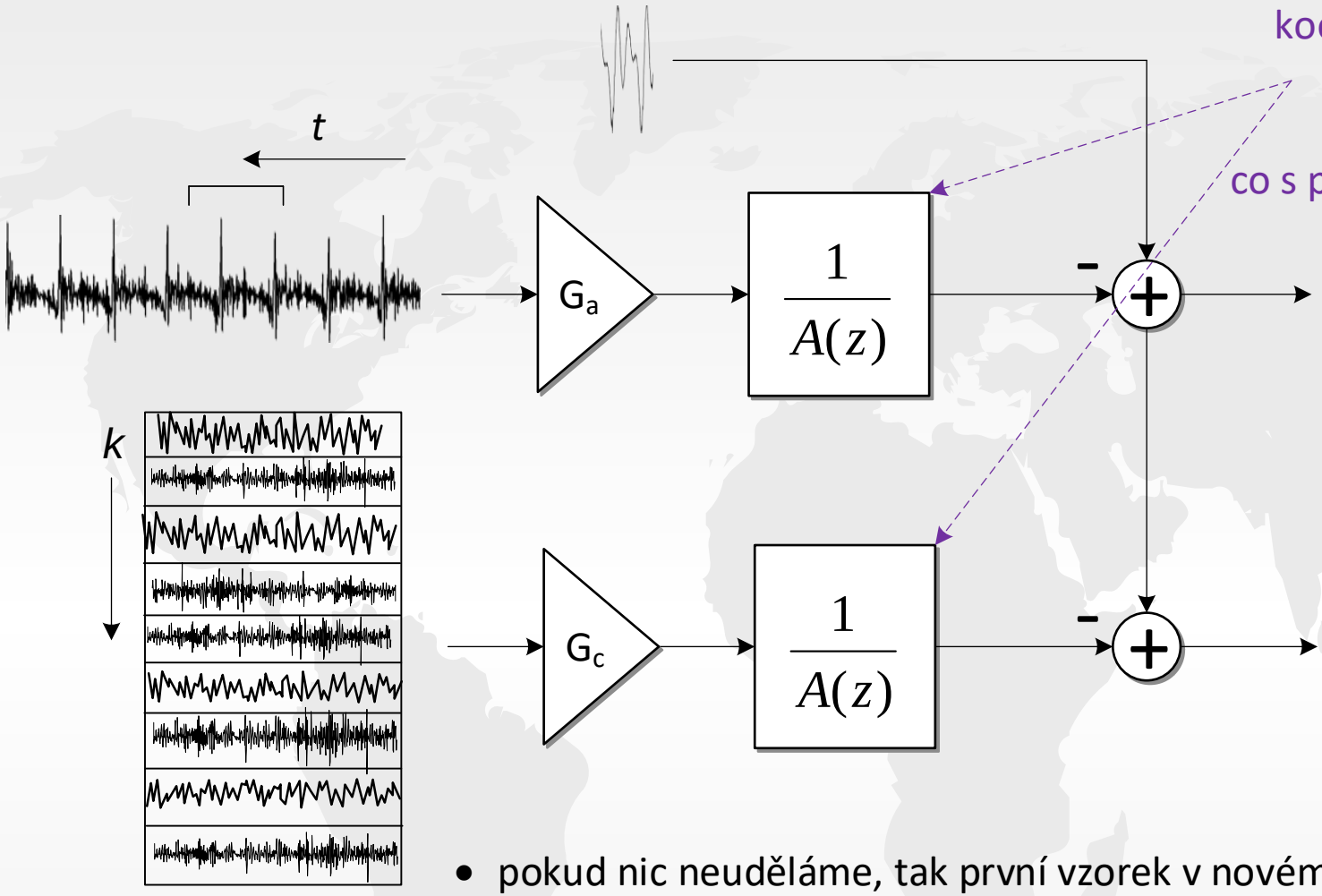
tohle musíme dělat 4x v každém rámci




!! v dalším výkladu však index *j* pro přehlednost vynecháme !!

# Zero-Input Response
# Zero-State Response

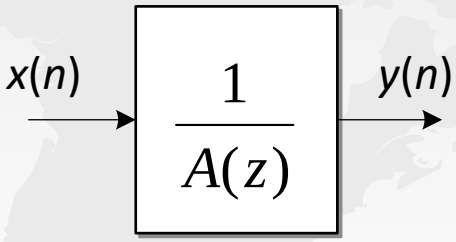
# Problém s přechody mezi podrámci

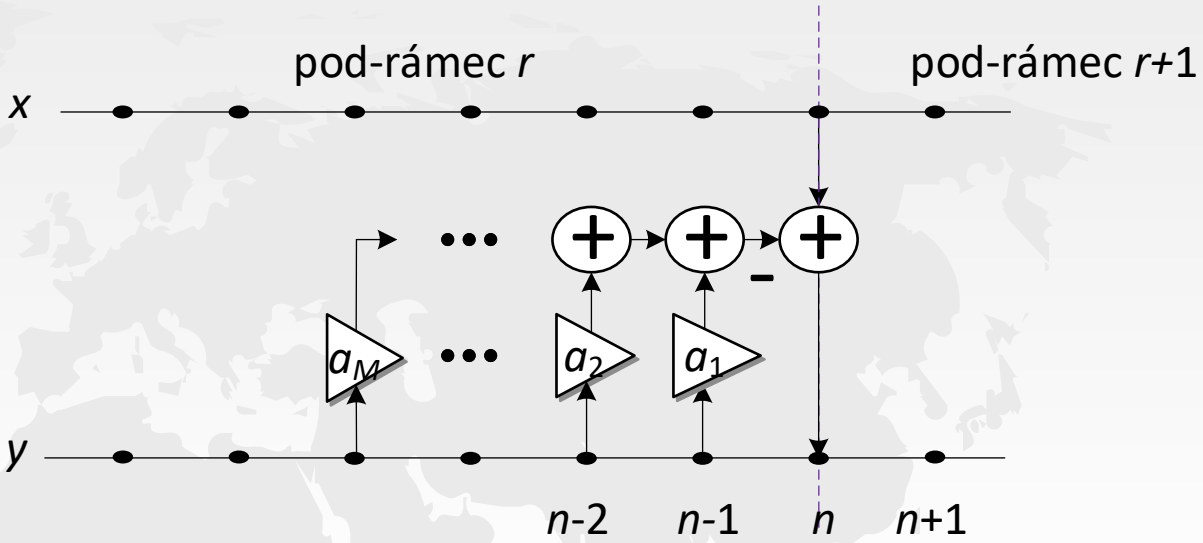



koeficienty filtru 1/A(z) se mění v každém podrámci
=>
co s pamětmi filtru na začátku nového podrámce?

- pokud nic neuděláme, tak první vzorek v novém podrámci bude filtrován nesprávně v důsledku špatných pamětí a v syntetickém signálu vznikne skok (discontinuity)
- musíme rozdělit odezvu filtru na dvě části – jedna, která bude odpovídat pouze na minulý výstupní signál (uložen v pamětech filtru) a druhá, která bude odpovídat pouze na vstupní signál
- první část pak jednoduše odečteme

# Zero-Input Response (ZIR) a Zero-State Response (ZSR)



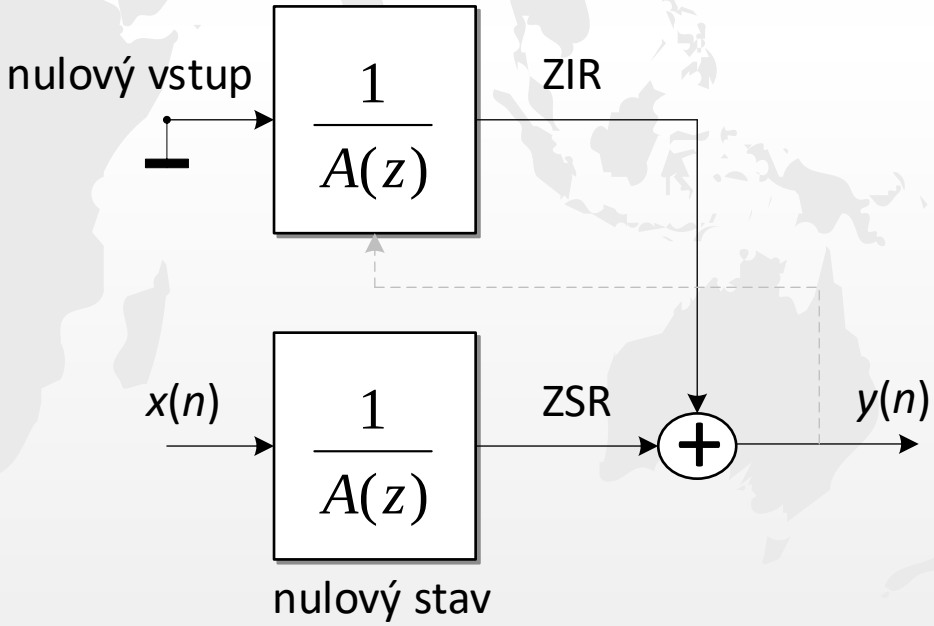

$$y(n) = x(n) - \sum_{i=1}^{M} a_i y(n-i) \qquad 0 \le n < N$$

pro M=1 máme:

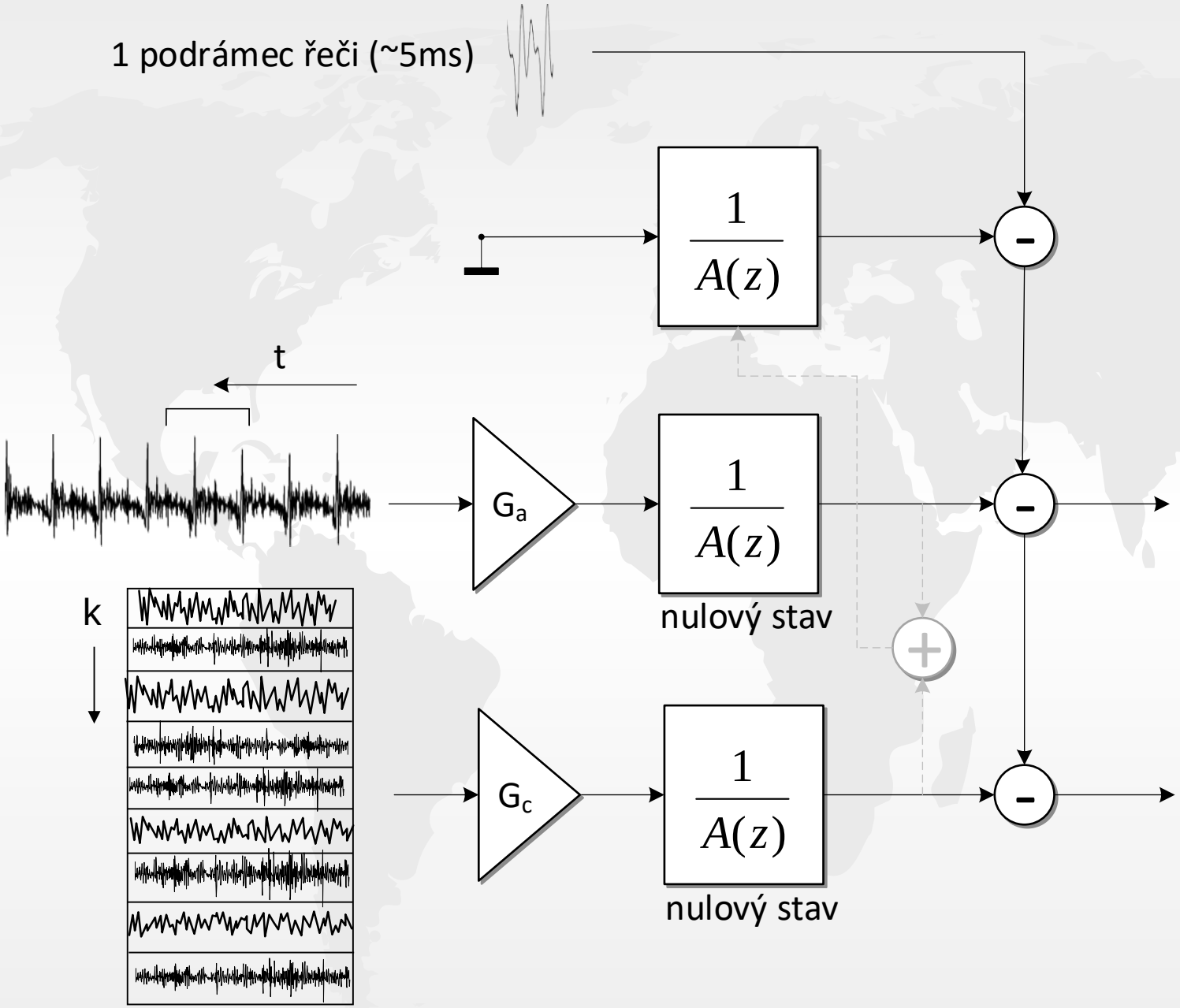
$$y(n) = x(n) - a_1 y(n-1)$$

$$y(n+1) = x(n+1) - a_1 x(n) + a_1^2 y(n-1)$$

$$y(n+2) = \underbrace{x(n+2) - a_1 x(n+1) + a_1^2 x(n)}_{\text{ZSR}} \underbrace{- a_1^3 y(n-1)}_{\text{ZIR}}$$

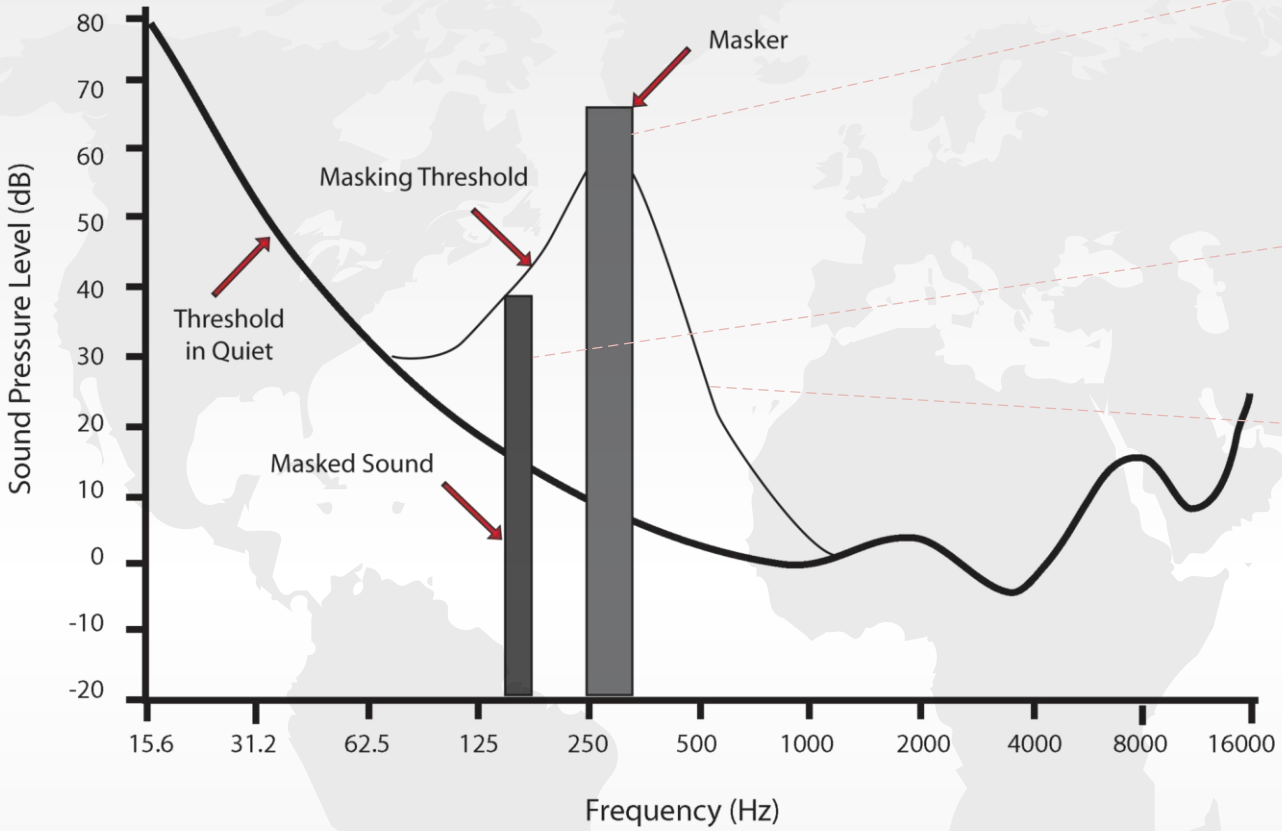


# Odečtení ZIR



# Perceptuální váhování

# Maskování tónů




maskovací šum o centralní frekvenci 300Hz a šírce pásma 100Hz a úrovni 65 dB

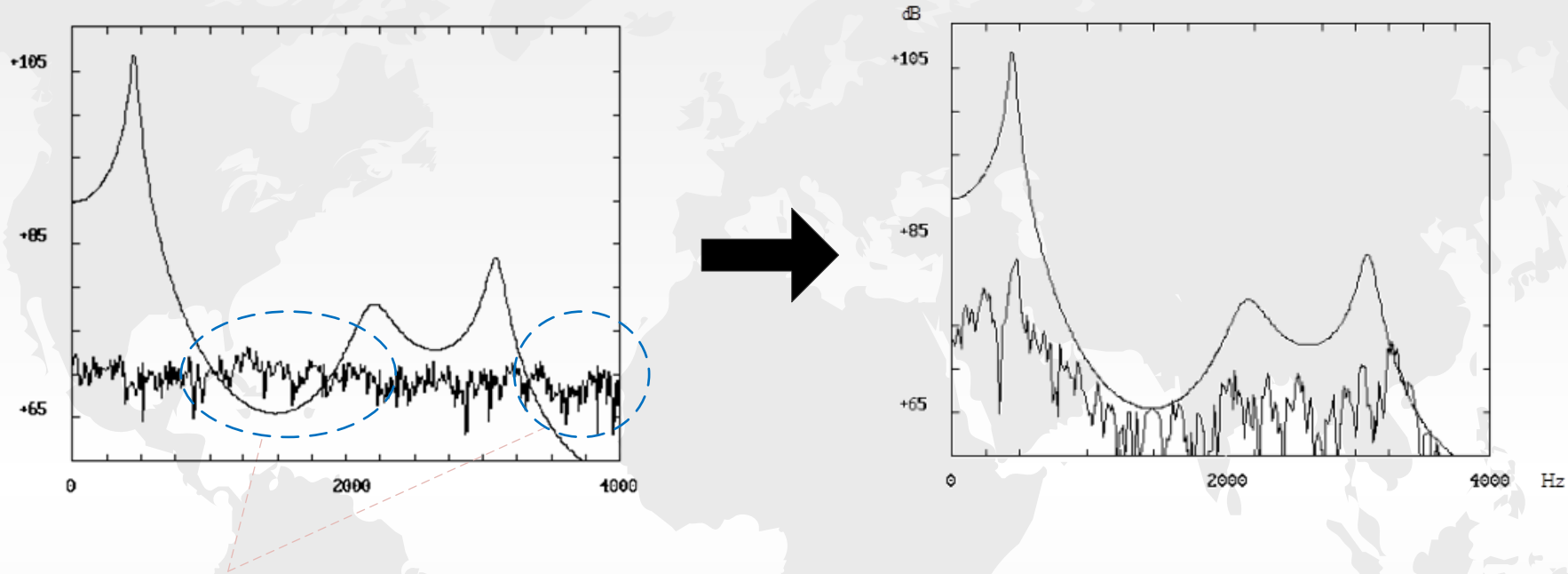
maskovaný signál o centralní frekvenci 200Hz a šírce pásma 50Hz a úrovni 38 dB

práh slyšitelnosti

- tóny v blízkosti silného tónu jsou maskovány
- spektrální komponenty s úrovní pod prahem slyšitelnosti není třeba kódovat
- lze tolerovat vyšší úroveň kvantizačního šumu v blízkosti silných tónů, např. formantů

- demo na https://www.youtube.com/watch?v=k6DVywW5NR4

# Maskování kvantizačního šumu



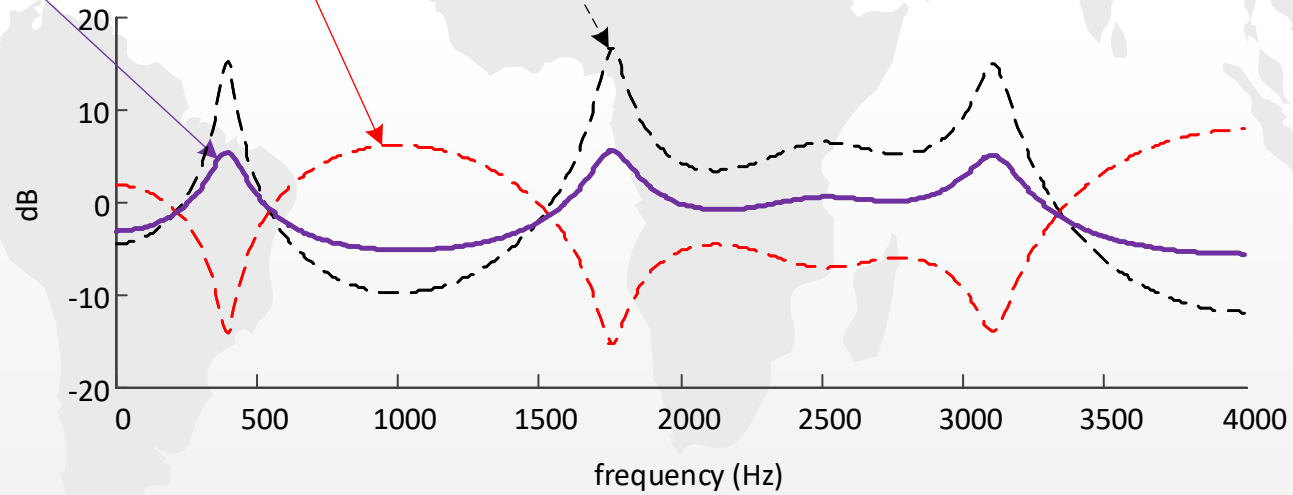

tohle bude slyšet

chtěli bychom, aby kvantizační šum vypadal nějak takhle

- Lidské ucho nedokáže registrovat zvuky, které jsou maskovány silnějším signálem
- i když najdeme adaptivní a fixní část excitace jak nejlépe umíme, pořád zbyde nějaký chybový signál, tzv. kvantizační šum
- části kvantizačního šumu v „údolích“ mohou být slyšet
- potřebovali bychom tenhle šum „zdeformovat“ tak, aby byl potlačen v „údolích“ a na oplátku si můžeme dovolit, aby byl zesílen v oblasti formantů

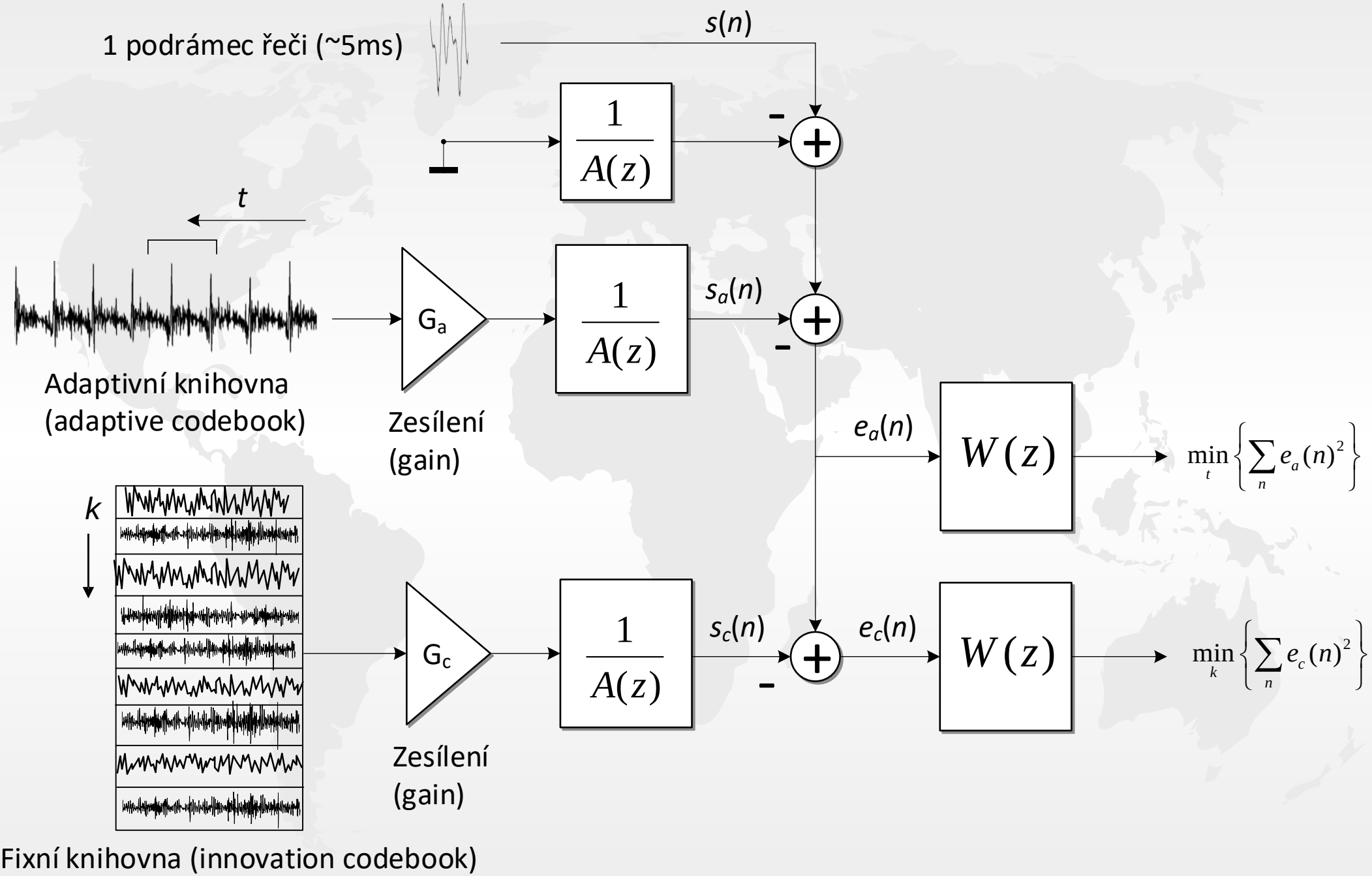
# Perceptuální váhování

$$W(z) = \frac{A(z)}{A(z/\gamma)} = \frac{1 - \sum_{i=1}^{M} a_i z^{-i}}{1 - \sum_{i=1}^{M} \gamma^i a_i z^{-i}}$$




- musíme najít takový tvar perceptuálního (váhovacícho) filtru, který bude dávat velkou váhu „údolím“ a malou váhu „formantům“
- celkový zisk filtru musí být 1, jinak by docházelo k postupnému zesilování/ zeslabování signálu
- $\gamma = 0.5 – 0.6$

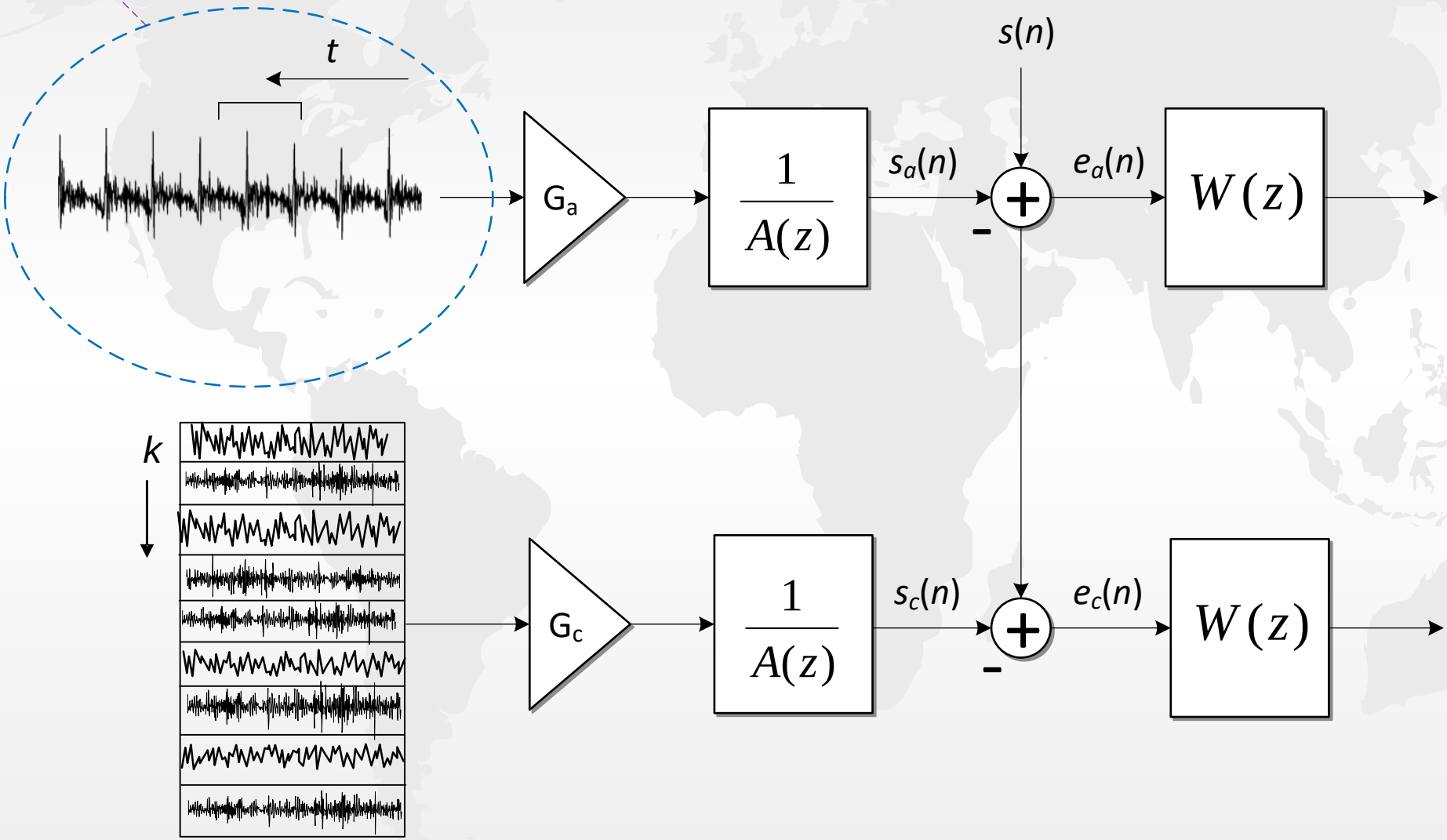
# Zavedení perceptuálního filtru



# Problém s výpočetní náročností

## Adaptivní knihovna: (7-9 bitů, t.j. 128 – 512 vektorů)

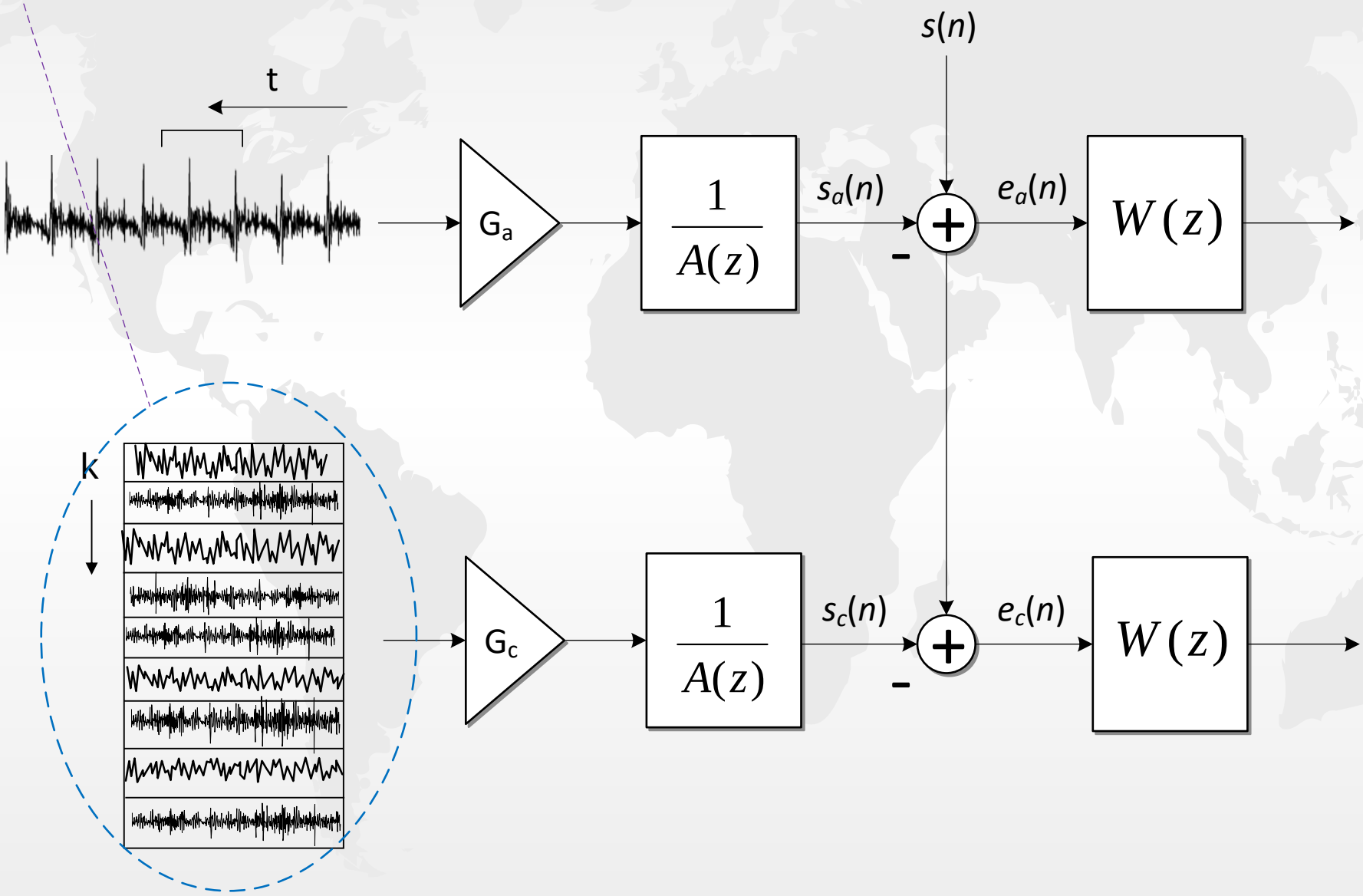
(prohledáváme knihovnu pouze kolem základního tónu)



# Problém s výpočetní náročností

## Fixní knihovna: (10-88 bitů, t.j. 1024 – $3.10^{26}$ vektorů)

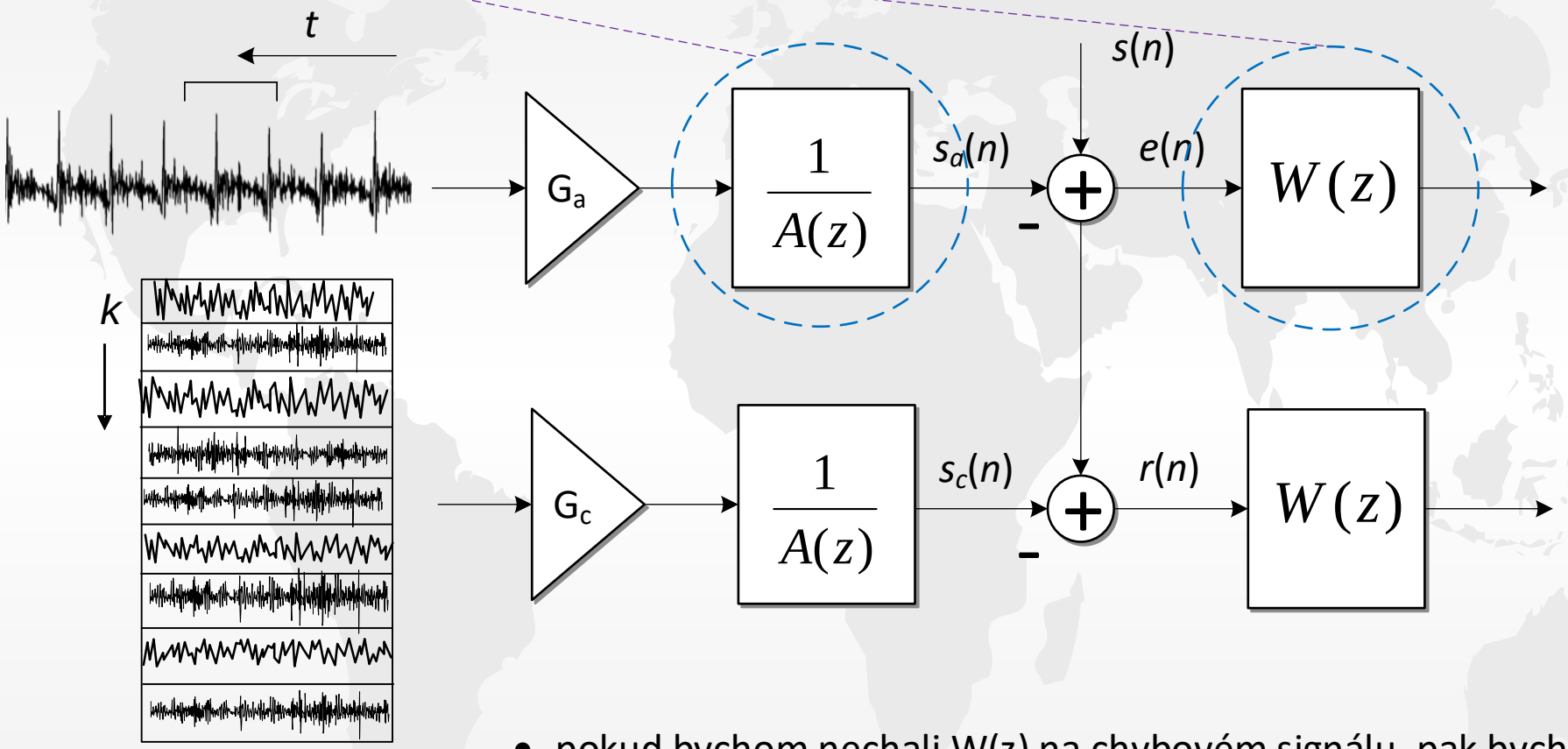
(vnucení jednoduchých struktur – pouze několik pulzů na stopu, omezený počet pozic pulzů, znaménka)



# Problém s výpočetní náročností

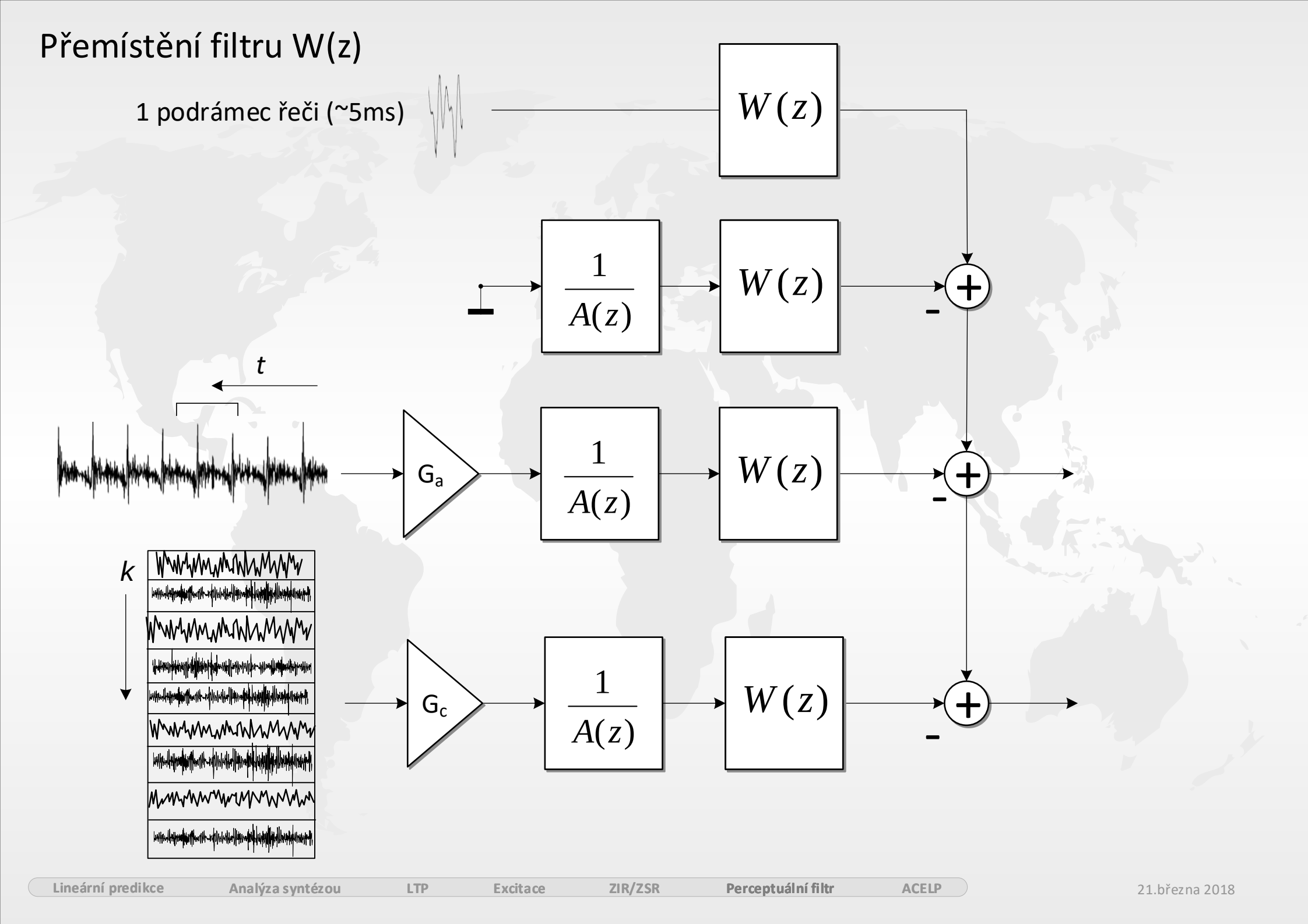
Filtrace: (pro každý vektor z knihovny je nutno provést konvoluci s filtry 16.řádu)

(nahrazení filtrů 1/A(z) a W(z) jejich impulzní odezvou) – viz dále

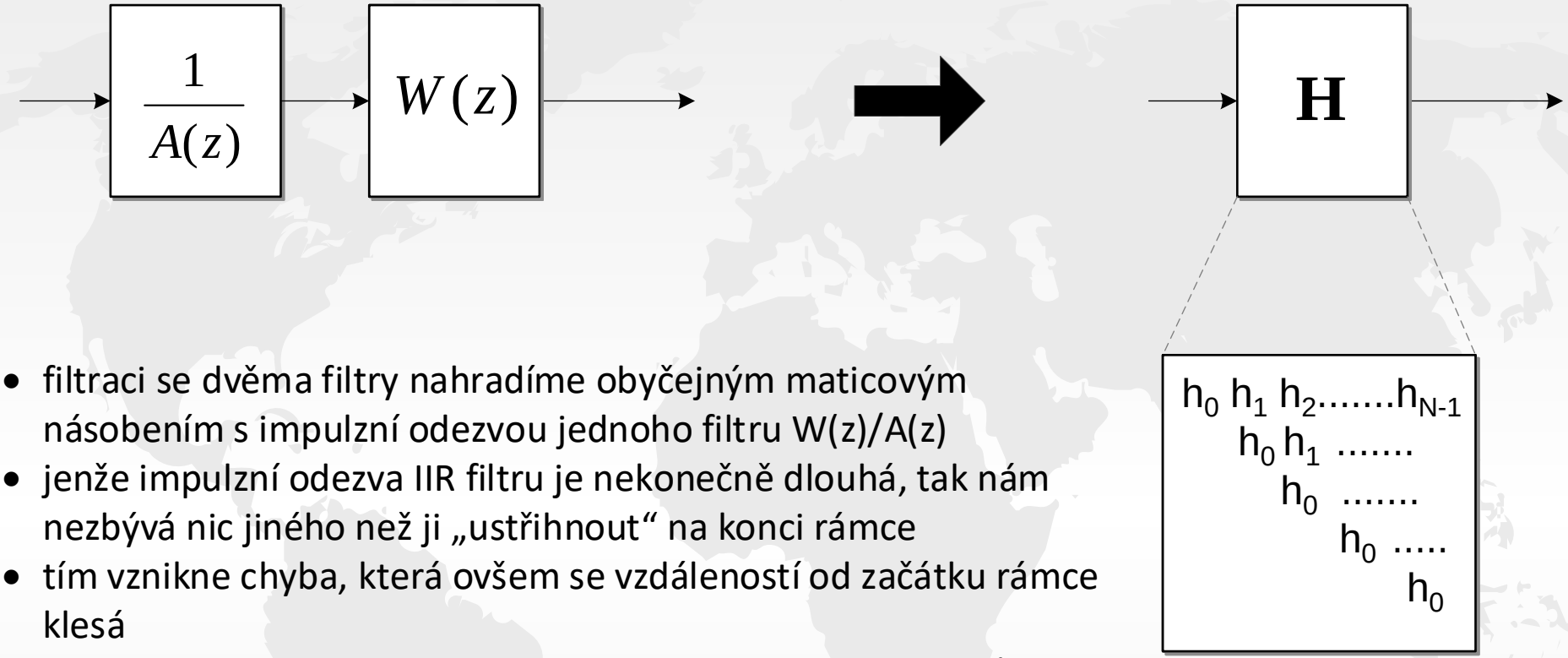



- pokud bychom nechali W(z) na chybovém signálu, pak bychom museli počítat konvoluci pro každý testovaný vektor z knihoven -> to je moc výpočetně náročné
- lepší nápad je přesunout filtr W(z) do obou větví excitace a na vstup

# Přemístění filtru W(z)

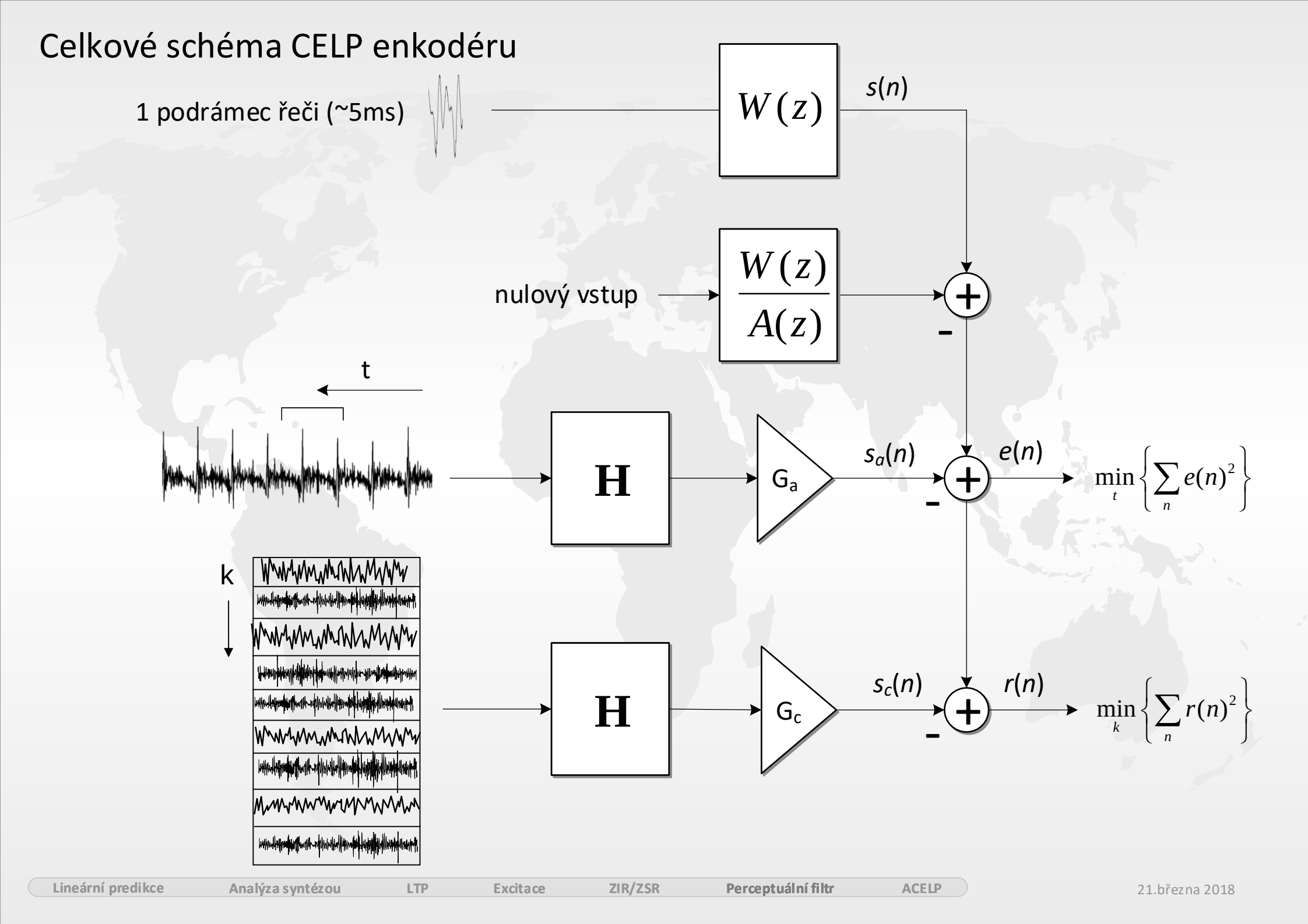


# Nahrazení filtrů impulzní odezvou

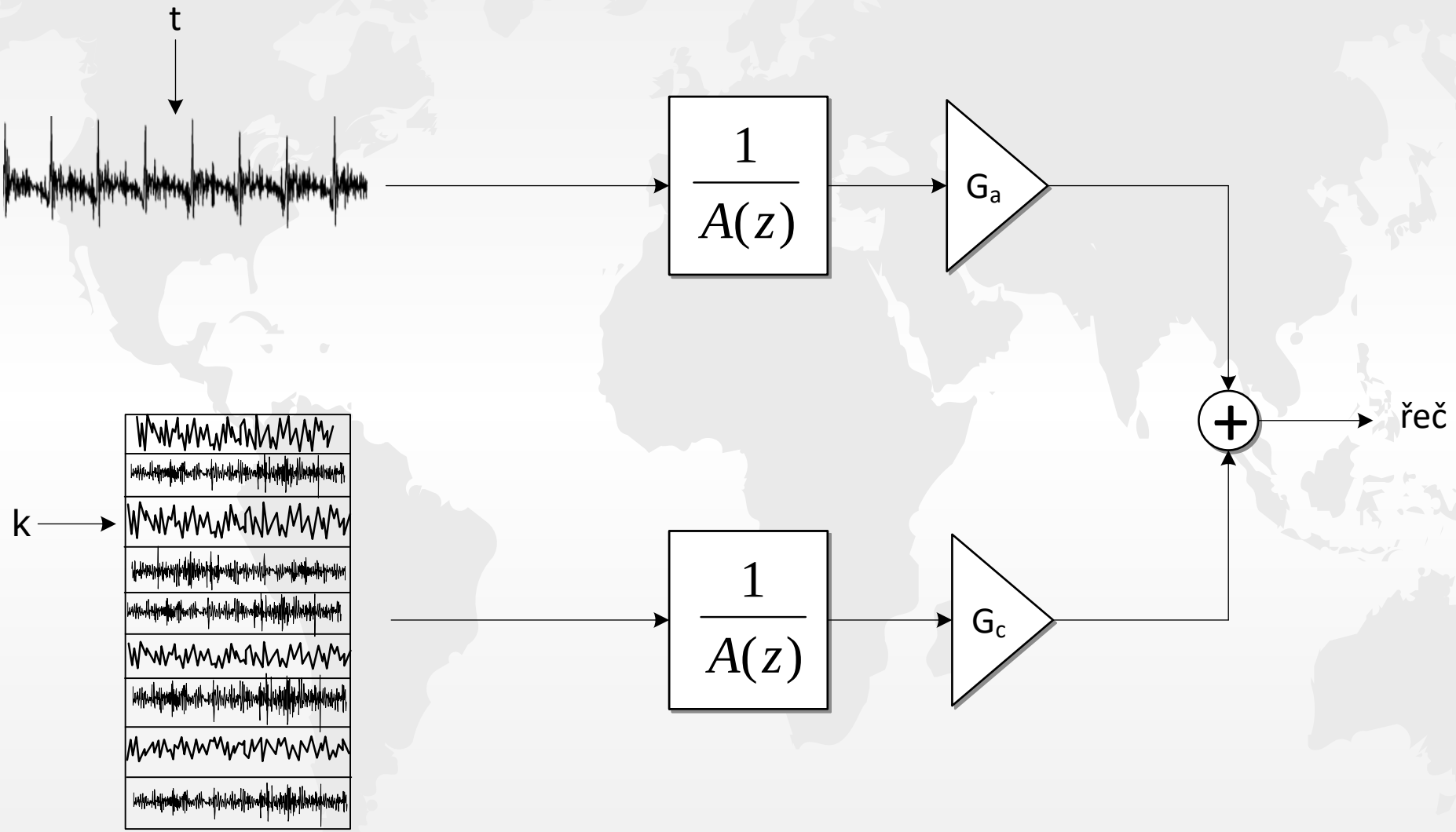



- filtraci se dvěma filtry nahradíme obyčejným maticovým násobením s impulzní odezvou jednoho filtru W(z)/A(z)
- jenže impulzní odezva IIR filtru je nekonečně dlouhá, tak nám nezbývá nic jiného než ji „ustřihnout“ na konci rámce
- tím vznikne chyba, která ovšem se vzdáleností od začátku rámce klesá
- koeficienty $h_0$, $h_1$, ..., $h_{N-1}$ tvoří impulzní odezvu filtru W(z)/A(z)
- vzhledem k předpokladu nulového stavu pamětí má matice H triangulární tvar

# Celkové schéma CELP enkodéru



# Celkové schéma CELP dekodéru



# Kodeky používající CELP

**GSM 6.10 Full-Rate (1987)**

13 kbps, 18.5 WMOPS, ETSI standard
technologie RPE-LTP (Regular Pulse-Excited Long-Term Prediction)
buzení podvzorkováno faktorem 3 a je kvantována pouze poloha prvního impulzu
další impulzy jsou kvantovány pomocí APCM

**IS-54 (1989), GSM 6.20 Half-Rate (1995)**

7.95 kbps, 20 MIPS, TIA standard
5.6 kbps, 30 MIPS, ETSI standard
technologie VSELP (Vector Sum-Excited Linear Prediction)
několik vektorů, filtrovaných předem, tvoří bázi
výsledné vektory se pak vytvoří jejich lineární kombinací

**FS-1016 (1991)**

4.8 kbps, 19 MIPS, U.S. DoD (Dept. of Defence) standard
technologie CELP
lin. knihovna s hodnotami 0, 1, -1
jednotlivé vektory se liší pouze ve 2 vzorcích

# ACELP

# ACELP

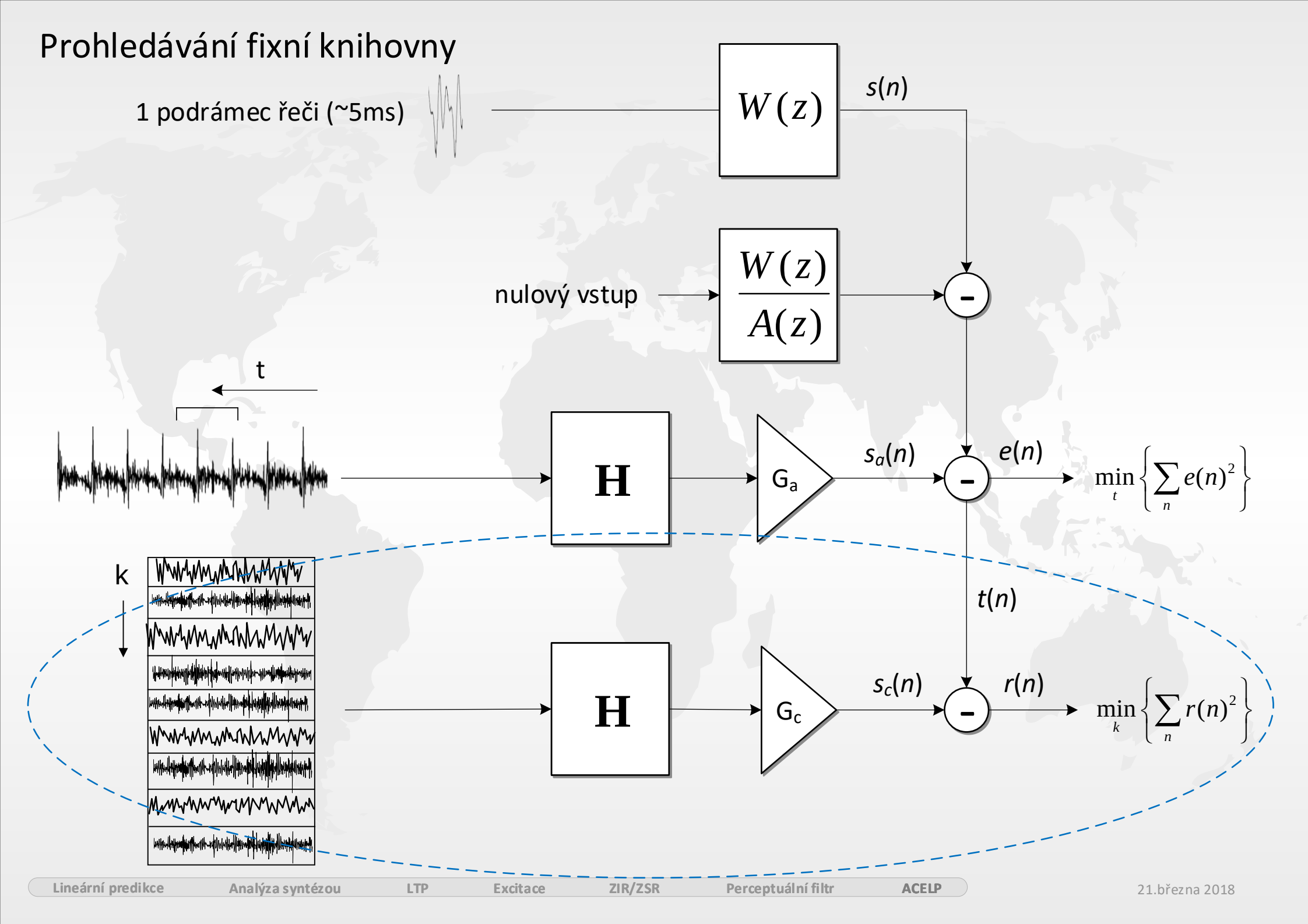
- ACELP® je patentovanou technologií VoiceAge Corp. a Université de Sherbrooke, CANADA
- vyvinuto výzk. skupinou Jean-Pierre Adoula v roce 1987
- poprvé publikováno zde: Adoul, J-P. and C. Lamblin (1987). ‘‘A Comparison of Some Algebraic Structures for CELP Coding of Speech,’’ IEEE ICASSP, pp. 1953–1956.

UNIVERSITÉ DE SHERBROOKE

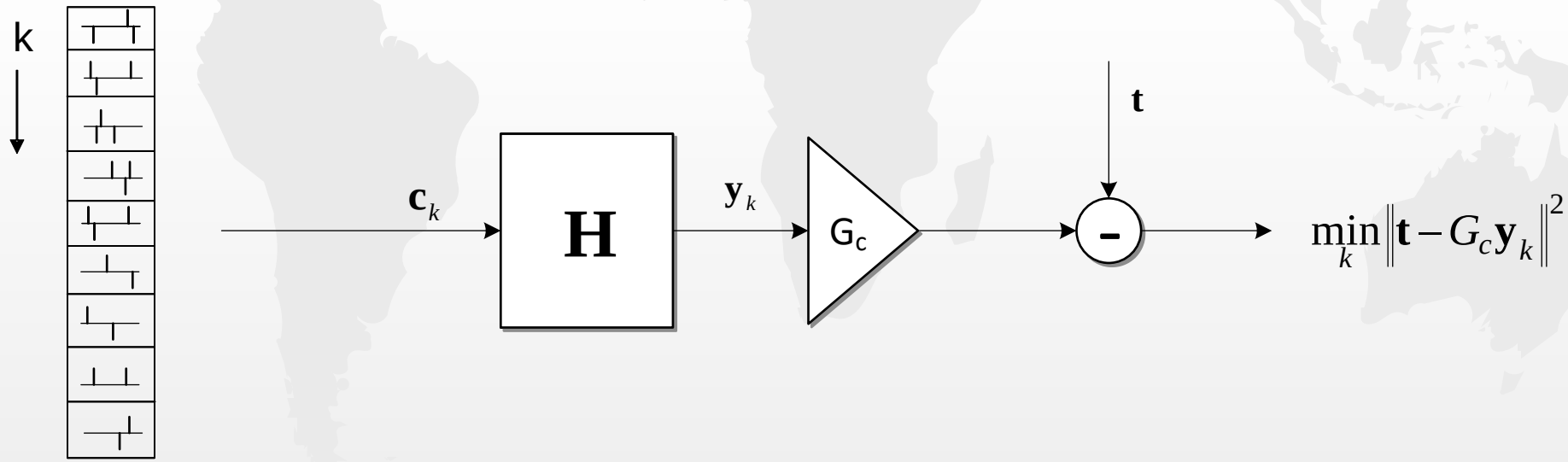
VoiceAge®
www.voiceage.com
The World's Premier Supplier of Speech and Audio Codecs

- kouzlo ACELPu spočívá v tom, že dokáže nahradit „obří“ fixní knihovnu signálů jednoduchou knihovnou s algebraickou strukturou, kde je jen několik málo pulzů v přesně definovaných pozicích, a tím zredukovat paměťovou a výpočetní náročnost
- technologii ACELP využívá cca
  - 2,4 miliard uživatelů mobilních telefonů na celém světě
  - 35 milionů uživatelů přehrávačů MP3
  - 500 milionů uživatelů internetových přehrávačů RealPlayer nebo MediaPlayer

# Prohledávání fixní knihovny



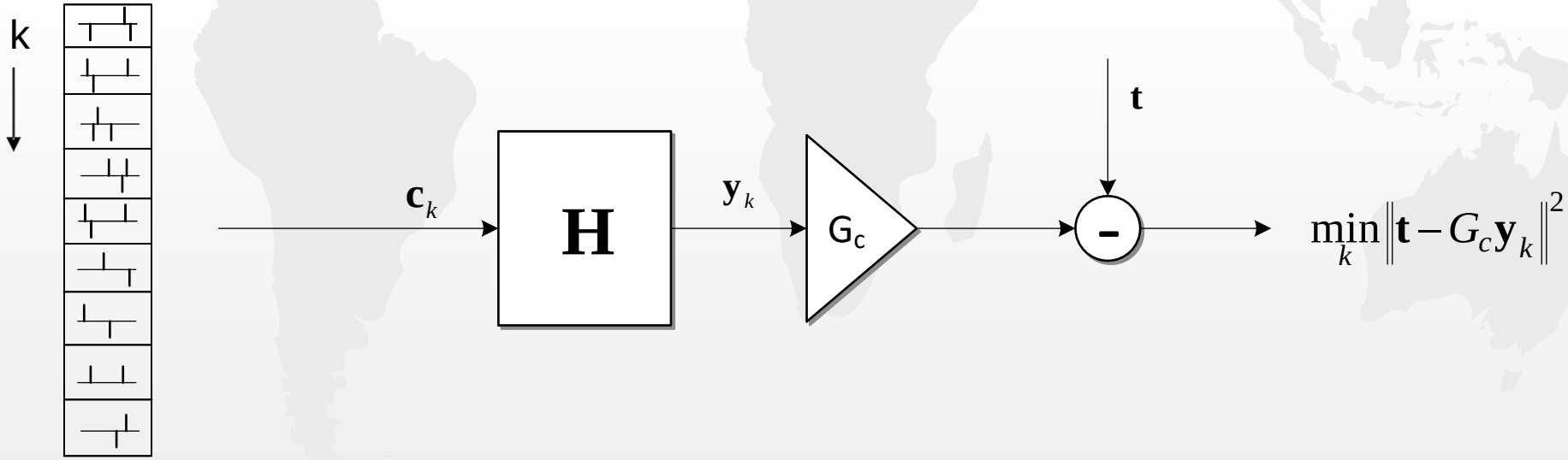
# Zavedení algebraické knihovny




algebraická knihovna (až 80 bitů)

# Prohledávání fixní knihovny

$$\min_k \left\| \mathbf{t} - G_c \mathbf{y}_k \right\|^2$$




algebraická knihovna (až 80 bitů)

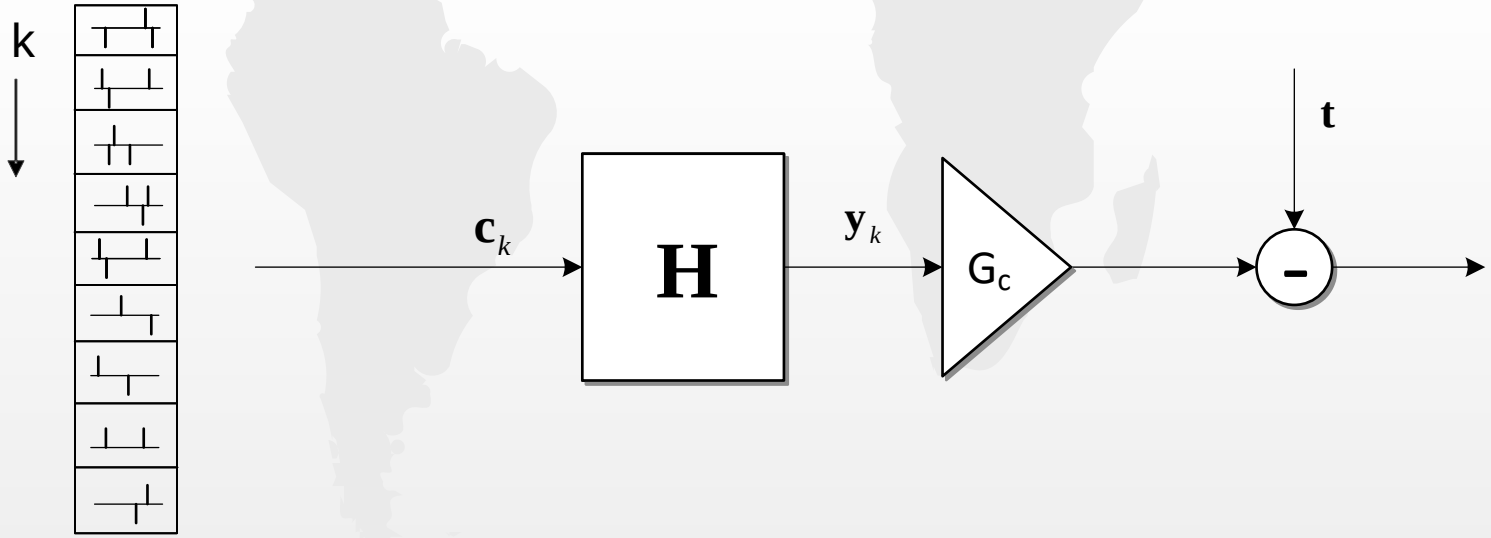
# Prohledávání algebraické knihovny

$$\min_k \left\| \mathbf{t} - G_c \mathbf{y}_k \right\|^2$$

$$\max_k \frac{\mathbf{t}^T . \mathbf{y}_k}{\mathbf{y}_k^T . \mathbf{y}_k}$$

korelace mezi cílovým (target) vektorem a testovaným vektorem

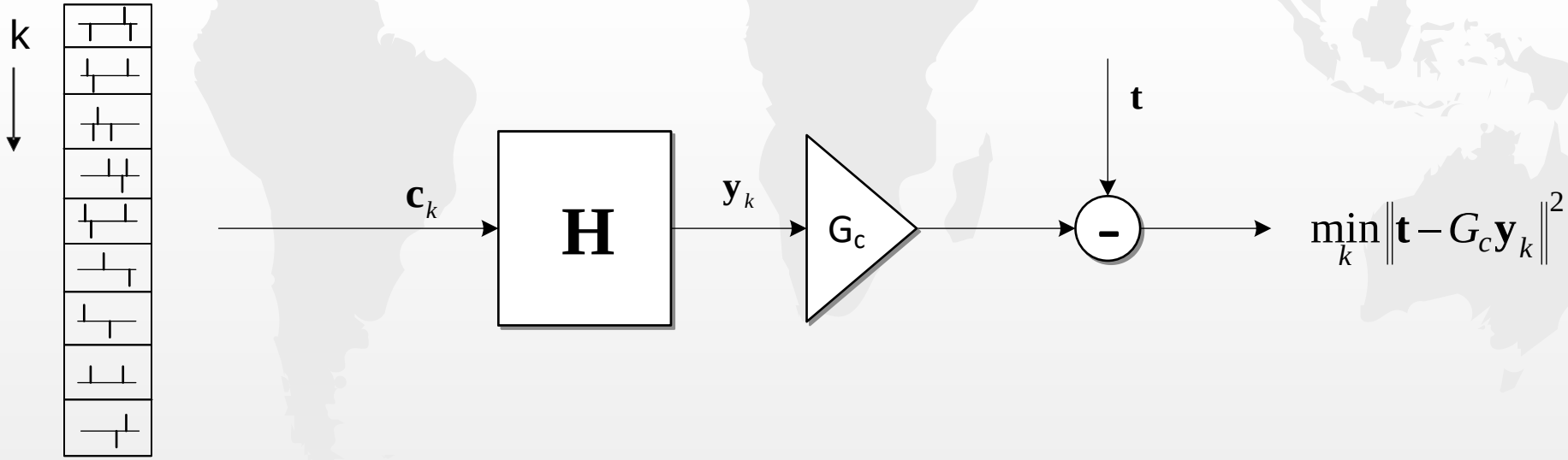
energie testovaného vektoru




algebraická knihovna (až 80 bitů)

# Prohledávání algebraické knihovny

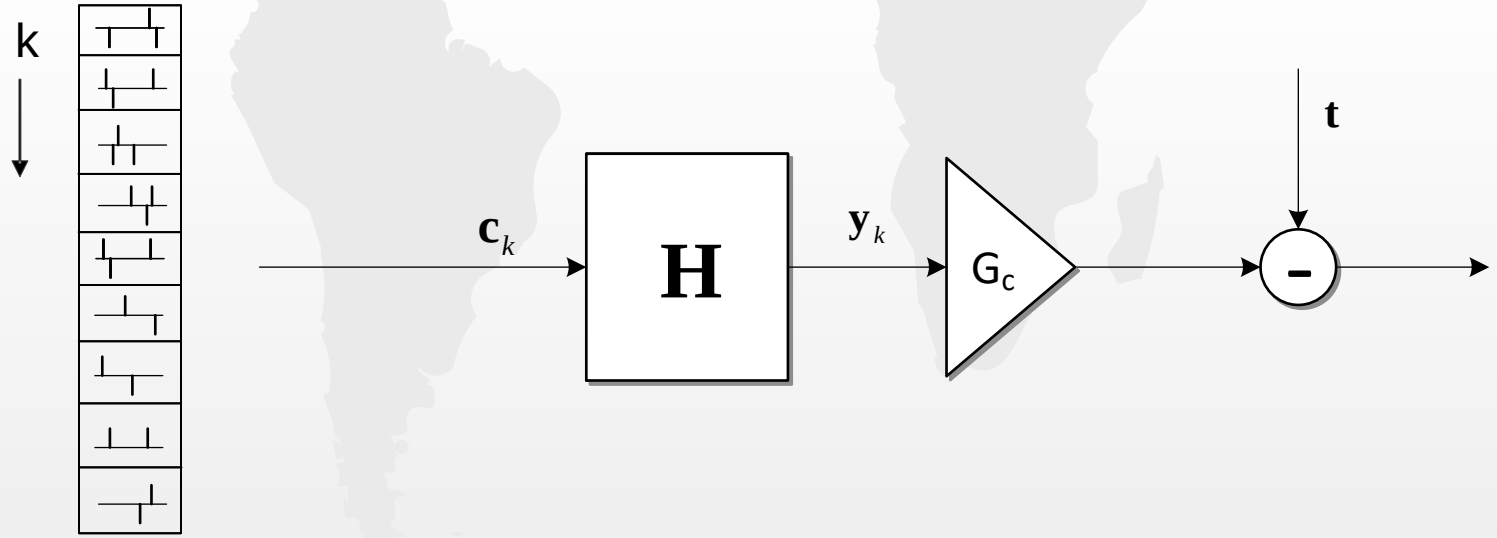
$$\min_k \left\| \mathbf{t} - G_c \mathbf{y}_k \right\|^2 \rightarrow \max_k \frac{\left( \mathbf{t}^T . \mathbf{y}_k \right)^2}{\mathbf{y}_k^T . \mathbf{y}_k} \rightarrow \max_k \frac{\left( \mathbf{t}^T . \mathbf{H} . \mathbf{c}_k \right)^2}{\mathbf{c}_k^T . \mathbf{H}^T . \mathbf{H} . \mathbf{c}_k}$$




algebraická knihovna (až 80 bitů)

# Prohledávání algebraické knihovny

$$\min_k \left\| \mathbf{t} - G_c \mathbf{y}_k \right\|^2 \rightarrow \max_k \frac{\left( \mathbf{t}^T . \mathbf{y}_k \right)^2}{\mathbf{y}_k^T . \mathbf{y}_k} \rightarrow \max_k \frac{\left( \mathbf{t}^T . \mathbf{H} . \mathbf{c}_k \right)^2}{\mathbf{c}_k^T . \mathbf{H}^T . \mathbf{H} . \mathbf{c}_k} \rightarrow \max_k \frac{\left( \mathbf{d}^T . \mathbf{c}_k \right)^2}{\mathbf{c}_k^T . \mathbf{\Phi} . \mathbf{c}_k}$$



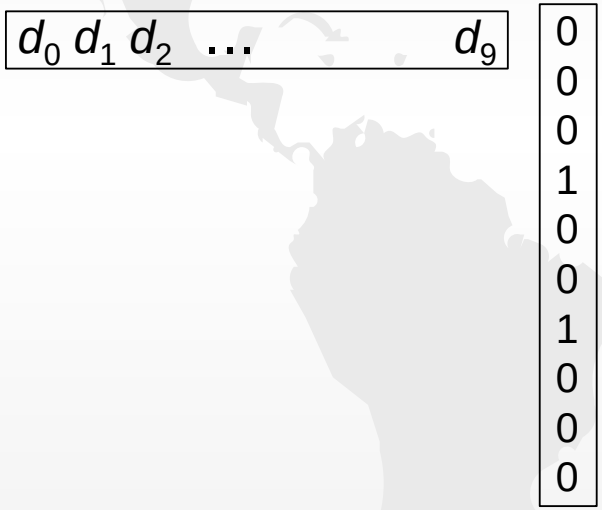

algebraická knihovna (až 80 bitů)

# Prohledávání algebraické knihovny

$$\max_k \frac{\left(\mathbf{d}^T.\mathbf{c}_k\right)^2}{\mathbf{c}_k^T.\mathbf{\Phi}.\mathbf{c}_k}$$

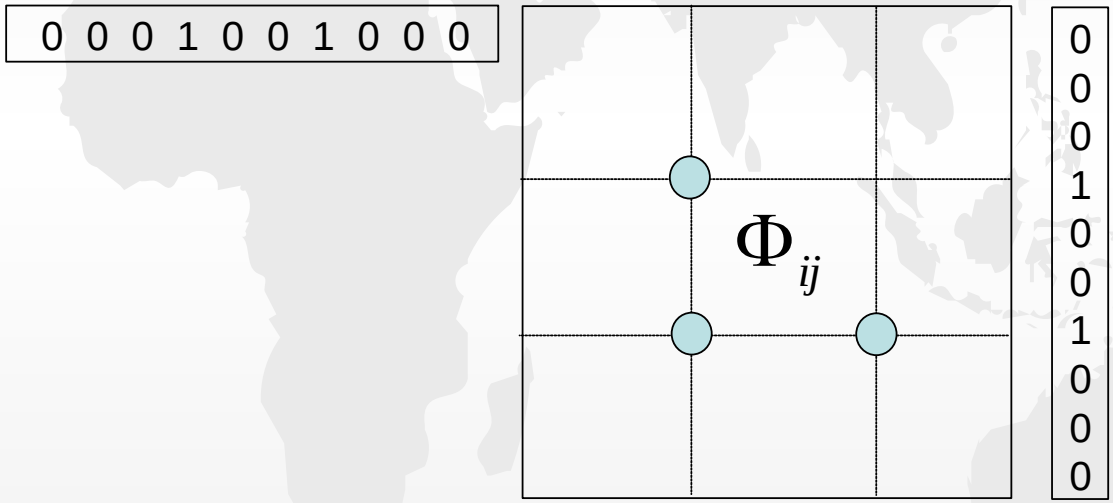
Lze prohledávat rychle, pokud $\mathbf{c}_k$ obsahuje jen velmi málo nenulových prvků s hodnotami +1 nebo -1

$\mathbf{d}^T.\mathbf{c}_k$




$= d_3 + d_6$

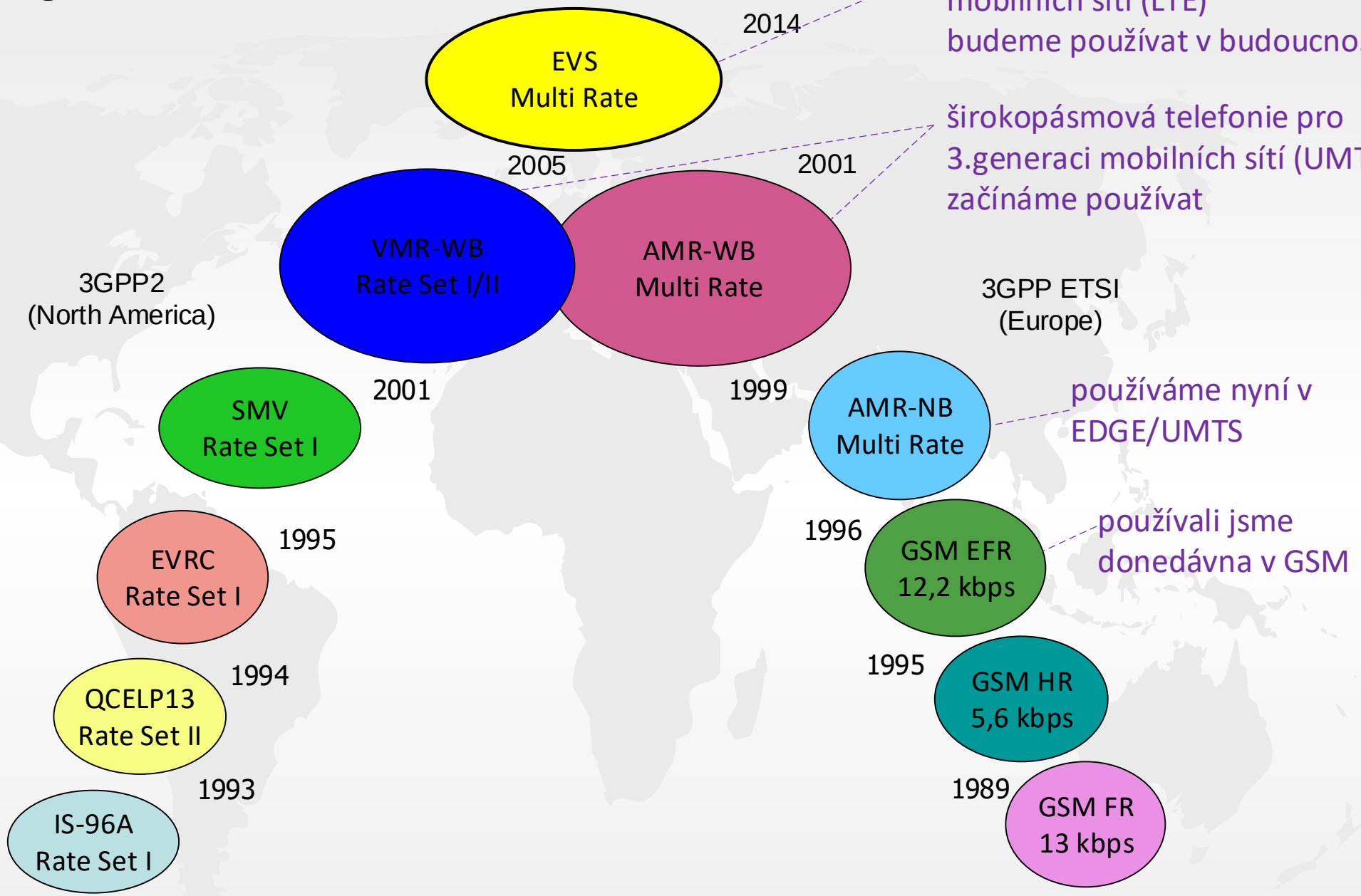
$\mathbf{c}_k^T.\mathbf{\Phi}.\mathbf{c}_k$



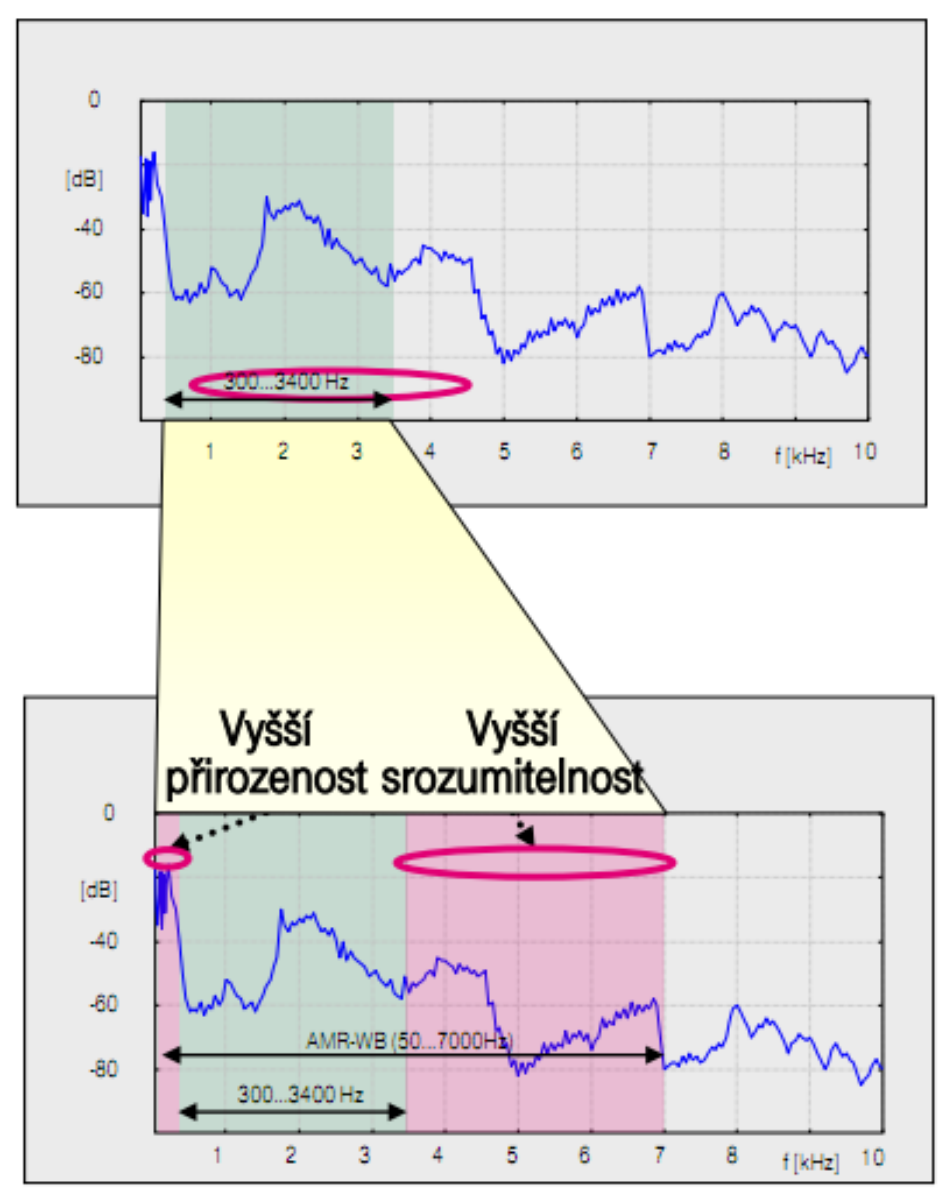

$= \Phi_{3,3} + \Phi_{6,6} + 2\Phi_{3,6}$


ACELP ve světě

# Technologie ACELP v mezinárodních standardech



# Od AMR-NB k AMR-WB (HD VOICE)

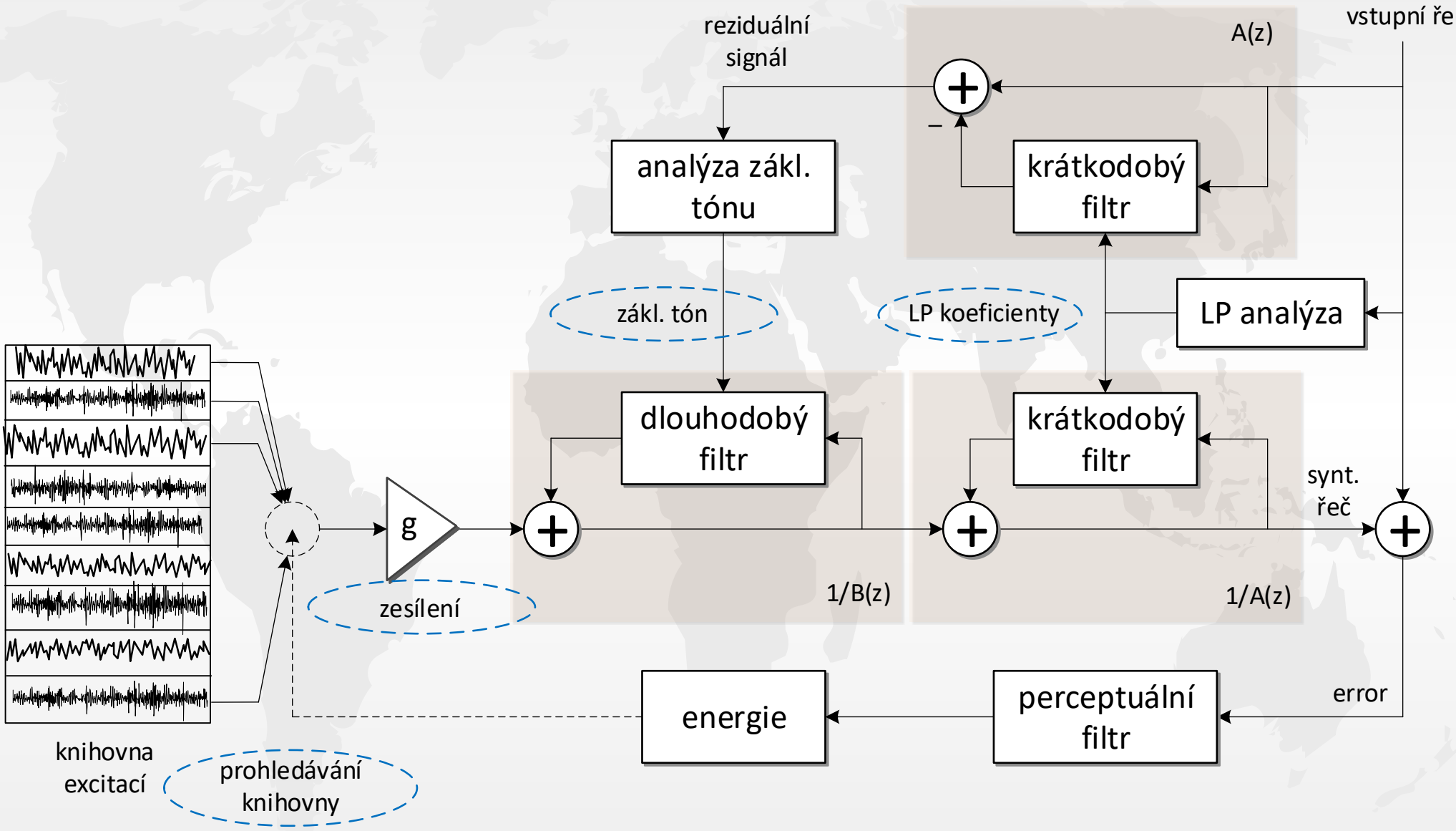




- HD voice demo na https://www.youtube.com/watch?v=Y4bb3b9PiRg


Konec

# Základní schéma kodeku



# Základní schéma kodeku