

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

FAKULTA ELEKTROTECHNICKÁ

KATEDRA RADIOELEKTRONIKY



**Dvoudimenzionální iterativní MAP detekce:  
Principy a aplikace v oblasti rekonstrukce obrazu**

**2D Iterative MAP Detection:  
Principles and Applications in Image Restoration**

**Disertační práce**

Doktorský studijní program: P2612 Elektrotechnika a Informatika

Studijní obor: 2601V013 Telekomunikační technika

Školitel: prof. Ing. Miloš Klíma, CSc.

**Daniel Kekrt**

**Praha, červen 2014**

**Abstrakt:** Disertační práce je věnována problematice *iterativní detekce* na bázi *kritéria maximalizujícího a posteriori pravděpodobnost (MAP)*. V této oblasti již bylo publikováno mnoho vědeckých prací. Iterativní detekce se hojně využívá v radioelektronice a telekomunikační technice. Konkrétně v moderních přenosových systémech založených na kanálovém kódování pomocí *zřetězených konvolučních kódů*, známých pod názvem *Turbokódy (TC)*. K detekci takových kódů se využívají metody vycházející z koncepce *Viterbiho algoritmu (VA)*. Jedná se o například o *Dopředně-Zpětně rekurentní algoritmus (FBA, BCJR)* nebo také *Viterbiho algoritmus s měkkým výstupem (SOVA)* v případě detekce kódů s binárními vstupními daty. Všechny tyto systémy lze označit a chápat jako jednodimenzionální, kde nezávislou proměnnou představuje čas. Aplikace uvedených (jednodimenzionálních) algoritmů je v současnosti poměrně běžná a dobře známá, včetně implementačních aspektů. Práce navazuje na tuto problematiku a zabývá tzv. *dvoudimenzionální iterativní MAP detekcí*. Tato oblast je nová, méně známá a v praxi doposud minoritně aplikovaná. Zde jsou vstupní data uvažována jako dvoudimenzionální a jsou kódována *prostorovým kernelem*, který zavádí do signálu obecnou *mezisymbolovou interferenci (ISI)*. Prostorový ISI kanál je v podstatě *dvoudimenzionálním konečným stavovým automatem (2D FSM)* a je dekomponovatelný na horizontálně a vertikálně zřetězenou síť jednoduchých kombinačních elementů (buněk). Možných uspořádání takové *kódovací sítě (EN)* je velmi mnoho a mohou být závislé i nezávislé na tvaru kernelu. Jejich topologie pak představují předlohu pro konstrukci *2D Iterativní dekódovací sítě (IDN)*, kde každá buňka této struktury je tvořena obecným *kombinačně-marginalizačním statistickým elementem (SISO modulem)*. Snahou zobecněného dvoudimenzionálního iterativní detektoru je rekonstrukce původních (vstupních) dat, podobně jako u jednodimenzionálních variant. Tedy potlačení mezisymbolové interference a aditivních šumů. Tento *iterativní detekční proces* je založen na určité sekvenční *aktivaci* příslušných elementů dekódovací sítě a následné výměně měkké informace (diskrétních hustot pravděpodobností) mezi nimi. Obecnému popisu dvoudimenzionální MAP detekce (2D MAP kritériu) je věnován úvod práce. Následně výklad přechází k otázkám konstrukce a implementace SISO modulů a topologií detekčních sítí, včetně jejich *aktivačních schémat*. Závěr práce je věnován ukázkám aplikace detekčních sítí v oblasti rekonstrukce černobílých snímků získaných ze CCD kamery. Tato rekonstrukce (binarizace) zahrnuje zaostření, potlačení šumů a interpolaci (de-mosaicing). Bude ukázáno, že předkládané řešení je schopno doplnit chybějící informace v Bayerově masce snímače kamery a poskytnout uspokojivý výstup i při obtížných podmínkách, kdy vstup detektoru tvoří decimovaný signál. Jednotlivé představené topologie budou vzájemně porovnány na této rekonstrukční úloze a to z hlediska výkonnosti, pomocí analýzy chybovosti v rekonstruovaných obrazech, a také z hlediska implementační náročnosti.

**Abstract:** The thesis provides a theoretical framework for the *iterative detection* based on *maximum a posteriori probability criterion* (MAP). Many thesis and papers have been published in this scientific area. The Iterative detection is mainly and widely used in the radio and telecommunication applications. Namely, in the modern transmission systems using channel coding by *concatenated convolutional encoders* also known as *TurboCodes* (TC). Such codes can be decoded by techniques and methods based on *Viterbi algorithm* (VA). We understand these systems as one-dimensional, where independent variable is time. Application of (one-dimensional) iterative decoding algorithms is currently well known and ordinary just as implementation aspects and issues. The thesis extends this research area and deals with *two-dimensional iterative MAP detection* issue that presents new and relatively unknown topic without significant practical usage. This generalization is based on the concept of *Forward-Backward detection algorithm* (FBA, BCJR) and *Soft-output Viterbi algorithm* (SOVA), i.e. the classical (one-dimensional) iterative detectors used in telecommunication applications. We generalize the one-dimensional detection problem considering the spatial *inter-symbol interference* (ISI) kernel as a *two-dimensional finite state machine* (2D FSM) representing a *encoding network* (EN) of the spatially concatenated elements. The cellular structure topology defines the design of the 2D *Iterative decoding network* (IDN), where each cell is a general *combination-marginalization statistical element* (SISO module) exchanging discrete probability density functions (information metrics) with neighboring cells during the process known as *activation*. The thesis introduction is dedicated to description of general principles of mentioned generalized detection technique. Consequently, we focus on implementation of SISO modules, IDN topologies and their *activation schedules*. In the end will be presented statistical (performance) analysis of various topologies with respect to their application in the field of image restoration. The iterative detection algorithm was applied on the task of binarization of images taken from a CCD camera. The reconstruction includes suppression of the defocus caused by the lens, CCD sensor noise suppression and interpolation (de-mosaicing). The simulations prove that the algorithm provides satisfactory results even in the case of an input image that is under-sampled due to the Bayer mask.

**Klíčová slova:** Kritérium maximální a posteriori pravděpodobnosti, Iterativní detekce, Dvoudimenzionální iterativní dekódovací síť, Dvoudimenzionální kanál (kernel) s mezisymbolovou interferencí, Dvoudimenzionální konečný stavový automat, Aditivní Gaussův šum, Dopředně-Zpětně rekurentní algoritmus, Viterbiho algoritmus s měkkým výstupem, Měkký inverzní modul, Demodulátor s měkkým výstupem, Turbokódy

**Keywords:** Maximum a posteriori probability criterion, Iterative detection, 2D Iterative decoding networks, 2D Inter-symbol interference channel, 2D Finite state machine, Additive Gaussian noise, Forward-Backward algorithm, Soft-Output Viterbi algorithm, Soft-In soft-Out module, Soft-Output demodulator, Turbocodes

### **Bibliografická citace**

KEKRT, D., *Dvoudimenzionální iterativní MAP detekce: Principy a aplikace v oblasti rekonstrukce obrazu*. Praha: České vysoké učení technické v Praze, Fakulta elektrotechnická, 2014. 95 s. Vedoucí disertační práce prof. Ing. Miloš Klíma, CSc.

Výtisk této práce je uložen:

Katedra radioelektroniky  
Fakulta elektrotechnická  
České vysoké učení technické v Praze  
Technická 2, 166 27 Praha 6, Česká Republika

Tel: +420 224 352 205  
Fax: +420 233 339 801

**Poděkování:** Rád bych poděkoval vedoucímu mé disertační práce, panu profesorovi Miloši Klímovi, nejen za profesionální, ale především lidský přístup. Za to, že se mne ujal v dobách pro mne nelehkých. Nebýt jeho, tato práce by nikdy nevznikla.

Tuto práci bych rád věnoval svému zesnulému nevlastnímu otci, který vždy byl a stále je mým nejlepším přítelem.



*To my best friend*

PETR SLABYHOUDEK  
\*2.12.1955 †10.3.2012

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Obecný model systému</b>	<b>4</b>
2.1	Deterministický kanál s mezisymbolovou interferencí . . . . .	5
2.2	Náhodný nekorelovaný kanál bez paměti . . . . .	6
<b>3</b>	<b>Kódovací sítě — buňkové modely kanálů s mezisymbolovou interferencí</b>	<b>7</b>
3.1	Elementy kódovacích sítí . . . . .	7
3.2	Topologie kódovacích sítí . . . . .	8
3.2.1	Pevné topologie . . . . .	8
3.2.2	Variabilní topologie . . . . .	10
3.2.3	Vrstvové sítě pro kanály s dekomponovatelným kernelem . . . . .	11
<b>4</b>	<b>2D MAP detekce</b>	<b>16</b>
4.1	Jednostupňová (optimální) MAP detekce . . . . .	16
4.2	Iterativní (suboptimální) MAP detekce . . . . .	17
<b>5</b>	<b>Iterativní dekódovací sítě</b>	<b>18</b>
5.1	Elementy iterativních dekódovacích sítí — SISO moduly . . . . .	18
5.2	Implementační aspekty . . . . .	19
5.2.1	Kombinace a marginalizace se stromovou strukturou . . . . .	19
5.2.2	Ošetření aritmetiky . . . . .	21
5.3	Topologie iterativních dekódovacích sítí . . . . .	23
5.3.1	Pevné topologie . . . . .	23
5.3.2	Variabilní topologie . . . . .	31
5.3.3	Zjednodušená variabilní topologie . . . . .	35
5.3.4	Vrstvové sítě pro kanály s dekomponovatelným kernelem . . . . .	37
5.4	Front-end iterativní dekódovací sítě . . . . .	43
5.5	Rozhodovací blok a tvrdý odhad . . . . .	43
<b>6</b>	<b>Aplikace v oblasti rekonstrukce černobílých snímků</b>	<b>45</b>
6.1	Model systému . . . . .	45
6.1.1	Rozostření snímku a rozmazání snímku pohybem . . . . .	46
6.1.2	Šumový model snímacího senzoru . . . . .	46
6.2	Potlačení rozostření a šumu senzoru . . . . .	47
6.2.1	Kernel kanálu modelující rozostření . . . . .	48
6.2.2	Příklady rekonstrukcí černobílých snímků iterativními sítěmi při znalosti kernelu kanálu . . . . .	49
6.2.3	Analýzy výkonnosti . . . . .	58
6.3	Potlačení rozmazání pohybem a šumu senzoru . . . . .	58
6.3.1	Kernel kanálu modelující rozmazání pohybem . . . . .	65
6.3.2	Příklady rekonstrukcí černobílých snímků iterativními sítěmi při znalosti kernelu kanálu . . . . .	66
<b>7</b>	<b>Závěr</b>	<b>76</b>

## Seznam konvencí a zkratk

### Seznam symbolů

	$a$	Skalár
$\mathbf{A} = [a[k, l]]_{k, l}, \mathbf{A} = [a_{k, l}]_{k, l}$		Matice
$a[k, l] = [\mathbf{A}]_{k, l}, a_{k, l} = [\mathbf{A}]_{k, l}$		Prvky matice
$\mathcal{A} = \{a[\ell]\}_{\ell}, \mathcal{A} = \{a_k\}_k$		Množiny
$\mathcal{A} = \{a[k, l]\}_{k, l}, \mathcal{A} = \{a_{k, l}\}_{k, l}$		
$a[\ell] = \{\mathcal{A}\}_{\ell}, a_k = \{\mathcal{A}\}_k$		Prvky množin
$a[k, l] = \{\mathcal{A}\}_{k, l}, a_{k, l} = \{\mathcal{A}\}_{k, l}$		
$\mathcal{A}_a = \{a^{(i)}\}_i$		Abecedy

### Seznam operátorů

$\odot, \odot$	Kombinační operátor
$\odot^{-1}$	Inverzní kombinační operátor
$\textcircled{M}, \textcircled{M}$	Marginalizační operátor

### Seznam indexů

$k, l$	Prostorové indexy
$i$	Index prvků abecedy

### Seznam označení klíčových proměnných

$a, \mathbf{A}$	Skutečná realizace náhodné proměnné
$\hat{a}, \hat{\mathbf{A}}$	Možná realizace náhodné proměnné (testovací estimátor)
$\hat{a}, \hat{\mathbf{A}}$	Odhad náhodné proměnné $a$ nebo $\mathbf{A}$
$p(\xi)$	Hustota pravděpodobnosti
$p_a(\xi), p_{\mathbf{A}}(\Xi)$	Hustota pravděpodobnosti náhodné proměnné $a$ nebo $\mathbf{A}$
$\Pr_a[n]$	Pravděpodobnost, že celočíselná náhodná proměnná $a$ je rovna $n$
$\mu$	Střední hodnota náhodné proměnné
$\sigma$	Směrodatná odchylka náhodné proměnné
$\mathcal{S}_I(\cdot), \mathcal{S}_O(\cdot), \mathcal{S}(\cdot)$	Vstupní, výstupní a sdružená měkká metrika (informace)
$\mathcal{S}_F(\cdot), \mathcal{S}_B(\cdot)$	Dopředná a zpětná měkká metrika (informace)
$\mathcal{P}_I(\cdot), \mathcal{P}_O(\cdot), \mathcal{P}(\cdot)$	Vstupní, výstupní a sdružená pravděpodobnost
$\mathcal{P}_F(\cdot), \mathcal{P}_B(\cdot)$	Dopředná a zpětná pravděpodobnost
$\mathcal{M}_I(\cdot), \mathcal{M}_O(\cdot), \mathcal{M}(\cdot)$	Vstupní, výstupní a sdružená logaritmická metrika
$\mathcal{M}_F(\cdot), \mathcal{M}_B(\cdot)$	Dopředná a zpětná logaritmická metrika
$M$	Mohutnost abecedy
$I$	Iterace
$\mathcal{A} : \mathcal{B}$	Sjednocení (Množina) všech množin $\mathcal{A}$ obsahujících podmnožinu $\mathcal{B}$
$\mathcal{A} : a$	Sjednocení (Množina) všech množin $\mathcal{A}$ obsahujících prvek $a$
$\mathbf{A} : a$	Množina všech matic $\mathbf{A}$ obsahujících prvek $a$
$f(\cdot), f[\cdot], f(\cdot)[\cdot]$	Funkce reálného argumentu, celočíselného argumentu a kombinovaného argumentu
{prvky : podmínky}	Definice množiny

## Seznam funkcí

card(.)	Kardinální číslo
ln(.)	Přirozený logaritmus
[.]	Zaokrouhlení na nejbližší celé číslo
[.]	Zaokrouhlení směrem dolů
efr(.)	Chybová funkce
efrc(.)	Komplementární chybová funkce

## Seznam zkratek

AWGN	Additive White Gaussian Noise
BCJR	Bahl, Cocke, Jelinek and Raviv
BER	Bit Error Ratio
BOM	Blurring due to Object Moving
CCD	Charge-Coupled Device
CFA	Color Filter Array
CSI	Channel State Information
DEC	Decision block
EFT	Extended Fixed Topology
EM	Expectation-Maximization
EN	Encoding Network
FBA	Forward-Backward Algorithm
FSM	Finite State Machine
FT	Fixed Topology
FWC	Full Well Capacity
GDCH	Gaussian Defocusing Channel
GPE	General Processing Element
IDN	Iterative Decoding Network
IECS	Independent Eliminated Channel States
ISI	Inter-Symbol Interference
LSTV	Layered Simplified Variable Topology
LT	Layered Topology
LVT	Layered Variable Topology
MAP	Maximum A posteriori Probability
Md	Metric Domain
ML	Memory-Less, Maximum Likelihood
Pd	Probability Domain
PDF	Probability Density Function
PgD	Page Detection
SGDCH	Simplified Gaussian Defocusing Channel
SISO	Soft-In Soft-Out Module
SODEM	Soft-Output Demodulator
SOVA	Soft-Output Viterbi Algorithm
SVT	Simplified Variable Topology
SyD	Symbol Detection
TCM	Trellis Coded Modulations
TC	Turbocodes
VA	Viterbi Algorithm
VT	Variable Topology



## Seznam obrázků

- Obr. 1 Blokové schéma obecného dvoudimenzionálního přenosového systému.
- Obr. 2 Obecný kombinační element (GPE) a jeho měkká inverze (SISO modul).
- Obr. 3 Kódovací síť se základní fixní topologií.
- Obr. 4 Tvary stavových proměnných  $\mathcal{R}[k, l]$  a  $\mathcal{C}[k, l]$  pro obecný ISI kernel  $\mathcal{G}_{3 \times 3}$ .
- Obr. 5 Kódovací síť s rozšířenou fixní topologií.
- Obr. 6 Tvary stavových proměnných  $\mathcal{R}[k, l]$ ,  $\mathcal{C}[k, l]$  a  $\mathcal{B}[k, l]$  pro obecný ISI kernel  $\mathcal{G}_{3 \times 3}$ .
- Obr. 7 Kódovací síť s variabilní topologií pro obecný ISI kernel  $\mathcal{G}_{3 \times 3}$ .
- Obr. 8 Kódovací síť s fixní vrstvou topologií.
- Obr. 9 Kódovací síť s variabilní vrstvou topologií pro obecný dekomponovatelný ISI kernel  $\mathcal{G}_{3 \times 3}^{(HV)}$ .
- Obr. 10 Obecná vnitřní struktura Soft-In Soft-Out modulu.
- Obr. 11 Princip kombinace a marginalizace se stromovou strukturou.
- Obr. 12 Vnitřní struktura jednoduchého ukázkového SISO modulu založená na kombinování a marginalizaci se stromovou strukturou.
- Obr. 13 Iterativní dekódovací síť marginalizující na úrovni bloku symbolů se základní topologií.
- Obr. 14 Tvary stavových proměnných  $\mathcal{R}[k, l]$  a  $\mathcal{C}[k, l]$  pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .
- Obr. 15 Iterativní dekódovací síť marginalizující na úrovni bloku symbolů s rozšířenou topologií.
- Obr. 16 Tvary stavových proměnných  $\mathcal{R}[k, l]$ ,  $\mathcal{C}[k, l]$  a  $\mathcal{B}[k, l]$  pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .
- Obr. 17 Iterativní dekódovací síť marginalizující na úrovni symbolů pro obecný ISI kernel  $\mathcal{G}_{3 \times 3}$ .
- Obr. 18 Iterativní dekódovací síť marginalizující na úrovni symbolů pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .
- Obr. 19 Zjednodušená iterativní dekódovací síť marginalizující na úrovni symbolů pro obecný ISI kernel  $\mathcal{G}_{3 \times 3}$  s dominantním koeficientem.
- Obr. 20 Vrstvová iterativní dekódovací síť marginalizující na úrovni bloku symbolů.
- Obr. 21 Vrstvová iterativní dekódovací síť marginalizující na úrovni symbolů pro obecný dekomponovatelný ISI kernel  $\mathcal{G}_{3 \times 3}^{(HV)}$ .
- Obr. 22 Zjednodušená vrstvová iterativní dekódovací síť marginalizující na úrovni symbolů pro obecný dekomponovatelný ISI kernel  $\mathcal{G}_{3 \times 3}^{(HV)}$  s dominantním koeficientem.
- Obr. 23 Blokové schéma CCD kamery napojené na rekonstrukční 2D iterativní detektor.
- Obr. 24 Příklady Gaussova defokusačního kernelu společně s výchozí PSF.
- Obr. 25 Příklady transformačních funkcí front-endu detektoru (gateway). Parametry modelu objektivu: Gaussovský kernel  $3 \times 3$ ,  $\Delta = 1.13$ . Úroveň šumu  $\hat{\sigma} = 10^3$  [e] a  $\hat{\sigma} = 316$  [e].
- Obr. 26 Příklady transformačních funkcí front-endu detektoru (gateway). Parametry modelu objektivu: Gaussovský kernel  $5 \times 5$ ,  $\Delta = 0.705$ . Úroveň šumu  $\hat{\sigma} = 70$  [e] a  $\hat{\sigma} = 10$  [e].
- Obr. 27 Ukázky realizací vstupních obrazů detektoru pro různé úrovně šumu a rozostření.
- Obr. 28 Ukázky binarizovaných obrazů QR kódu na výstupu různých druhů IDN využívajících Pd-SyD techniku. Vstupní šedotónový obraz  $\mathbf{R}$  byl simulován s plným rozlišením (monochromatická kamera). Parametry modelu objektivu: Gaussovský kernel  $3 \times 3$ ,  $\Delta = 1.13$ . Úroveň šumu  $\hat{\sigma} = 10^3$  [e].
- Obr. 29 Ukázky binarizovaných obrazů QR kódu  $\hat{\mathbf{D}}$  na výstupu různých druhů IDN využívajících Pd-PgD techniku. Vstupní šedotónový obraz  $\mathbf{R}$  byl simulován s plným rozlišením (monochromatická kamera). Parametry modelu objektivu: Gaussovský kernel  $3 \times 3$ ,  $\Delta = 1.13$ . Úroveň šumu  $\hat{\sigma} = 10^3$  [e].
- Obr. 30 Ukázky binarizovaných obrazů QR kódu  $\hat{\mathbf{D}}$  na výstupu různých druhů IDN využívajících Pd-SyD techniku. Vstupní šedotónový obraz  $\mathbf{R}$  byl simulován s polovičním rozlišením (barevná kamera). Parametry modelu objektivu: Gaussovský kernel  $3 \times 3$ ,  $\Delta = 1.13$ . Úroveň šumu  $\hat{\sigma} = 316$  [e].

- Obr. 31 Ukázky binarizovaných obrazů QR kódu  $\hat{\mathbf{D}}$  na výstupu různých druhů IDN využívajících Pd-PgD techniku. Vstupní šedotónový obraz  $\mathbf{R}$  byl simulován s polovičním rozlišením (barevná kamera). Parametry modelu objektivu: Gaussovský kernel  $3 \times 3$ ,  $\Delta = 1.13$ . Úroveň šumu  $\hat{\sigma} = 316$  [e].
- Obr. 32 Ukázky binarizovaných obrazů QR kódu  $\hat{\mathbf{D}}$  na výstupu různých druhů IDN marginalizujících na úrovni symbolů. Vstupní šedotónový obraz  $\mathbf{R}$  byl simulován s plným rozlišením (monochromatická kamera). Parametry modelu objektivu: Gaussovský kernel  $5 \times 5$ ,  $\Delta = 0.705$ . Úroveň šumu  $\hat{\sigma} = 70$  [e].
- Obr. 33 Ukázky binarizovaných obrazů QR kódu  $\hat{\mathbf{D}}$  na výstupu různých druhů IDN marginalizujících na úrovni symbolů. Vstupní šedotónový obraz  $\mathbf{R}$  byl simulován s polovičním rozlišením (barevná kamera). Parametry modelu objektivu: Gaussovský kernel  $5 \times 5$ ,  $\Delta = 0.705$ . Úroveň šumu  $\hat{\sigma} = 10$  [e].
- Obr. 34 Tvary stavových proměnných  $\mathcal{R}[k, l]$ ,  $\mathcal{C}[k, l]$  a  $\mathcal{B}[k, l]$  pro Gaussovský kernel  $\mathcal{G}_{3 \times 3, \Delta}^{(\text{Gauss})}$ .
- Obr. 35 Analýzy chybovosti rekonstrukce náhodných realizací binárních obrazů pomocí různých druhů IDN využívajících Pd-SyD techniku. Vstupní (šedotónový) obraz byl simulován s plným rozlišením (monochromatická kamera).
- Obr. 36 Analýzy chybovosti rekonstrukce náhodných realizací binárních obrazů pomocí různých druhů IDN využívajících Pd-PgD techniku. Vstupní (šedotónový) obraz byl simulován s plným rozlišením (monochromatická kamera).
- Obr. 37 Analýzy chybovosti rekonstrukce náhodných realizací binárních obrazů pomocí různých druhů Vrstvových IDN využívajících Pd-SyD techniku. Vstupní (šedotónový) obraz byl simulován s plným rozlišením (monochromatická kamera).
- Obr. 38 Analýzy chybovosti rekonstrukce náhodných realizací binárních obrazů pomocí různých druhů Vrstvových IDN využívajících Pd-PgD techniku. Vstupní (šedotónový) obraz byl simulován s plným rozlišením (monochromatická kamera).
- Obr. 39 Analýzy chybovosti rekonstrukce náhodných realizací binárních obrazů pomocí různých druhů IDN využívajících Pd-SyD techniku. Vstupní (šedotónový) obraz byl simulován s polovičním rozlišením (barevná kamera).
- Obr. 40 Analýzy chybovosti rekonstrukce náhodných realizací binárních obrazů pomocí různých druhů IDN využívajících Pd-PgD techniku. Vstupní (šedotónový) obraz byl simulován s polovičním rozlišením (barevná kamera).
- Obr. 41 Geneze rozmazání snímku díky pohybu snímaného objektu či kamery.
- Obr. 42 Tvary stavových proměnných  $\mathcal{R}[k, l]$  a  $\mathcal{C}[k, l]$  pro ukázkový ISI kernel  $\mathcal{G}_{3 \times 3, 45^\circ}^{(\text{Blur})}$ .
- Obr. 43 Iterativní dekodovací síť marginalizující na úrovni symbolů pro ukázkový ISI kernel  $\mathcal{G}_{3 \times 3, 45^\circ}^{(\text{Blur})}$ .
- Obr. 44 Tvar stavové proměnné  $\mathcal{C}[k, l]$  pro ukázkový ISI kernel  $\mathcal{G}_{7 \times 1, 0^\circ}^{(\text{Blur})}$ .
- Obr. 45 Iterativní dekodovací síť marginalizující na úrovni symbolů pro ukázkový ISI kernel  $\mathcal{G}_{7 \times 1, 0^\circ}^{(\text{Blur})}$ .
- Obr. 46 Příklady transformačních funkcí front-endu detektoru (gateway). Model rozmazání pohybem:  $3 \times 3$ ,  $\varphi = 45^\circ$ . Úroveň šumu  $\hat{\sigma} = 10^3$  [e].
- Obr. 47 Příklady transformačních funkcí front-endu detektoru (gateway). Model rozmazání pohybem:  $7 \times 1$ ,  $\varphi = 0^\circ$ . Úroveň šumu  $\hat{\sigma} = 316$  [e].
- Obr. 48 Ukázky realizací vstupních obrazů detektoru  $\mathbf{R}$  pro různé úrovně šumu a rozmazání pohybem.
- Obr. 49 Ukázky binarizovaných obrazů textu  $\hat{\mathbf{D}}$  na výstupu IDN marginalizující na úrovni bloku symbolů se základní topologií. Vstupní (šedotónový) obraz  $\mathbf{R}$  byl simulován s plným rozlišením (monochromatická kamera). Model rozmazání pohybem:  $3 \times 3$ ,  $\varphi = 45^\circ$ . Úroveň šumu  $\hat{\sigma} = 10^3$  [e].

- Obr. 50 Ukázky binarizovaných obrazů textu  $\hat{\mathbf{D}}$  na výstupu IDN marginalizující na úrovni symbolů. Vstupní (šedotónový) obraz  $\mathbf{R}$  byl simulován s plným rozlišením (monochromatická kamera). Model rozmazání pohybem:  $7 \times 1$ ,  $\varphi = 0^\circ$ . Úroveň šumu  $\hat{\sigma} = 10^3$  [e].
- Obr. 51 Ukázky binarizovaných obrazů textu  $\hat{\mathbf{D}}$  na výstupu IDN marginalizující na úrovni bloku symbolů se základní topologií. Vstupní (šedotónový) obraz  $\mathbf{R}$  byl simulován s polovičním rozlišením (barevná kamera). Model rozmazání pohybem:  $3 \times 3$ ,  $\varphi = 45^\circ$ . Úroveň šumu  $\hat{\sigma} = 316$  [e].
- Obr. 52 Ukázky binarizovaných obrazů textu  $\hat{\mathbf{D}}$  na výstupu IDN marginalizující na úrovni symbolů. Vstupní (šedotónový) obraz  $\mathbf{R}$  byl simulován s polovičním rozlišením (barevná kamera). Model rozmazání pohybem:  $7 \times 1$ ,  $\varphi = 0^\circ$ . Úroveň šumu  $\hat{\sigma} = 316$  [e].

## Seznam tabulek

- Tab. 1 Přehled kombinačních a marginalizačních operátorů.
- Tab. 2 Tabulka kombinací  $IO$  (vstupů a výstupů) definující strukturu jednoduchého ukázkového SISO modulu a abecedy  $IO$  proměnných.
- Tab. 3 Tabulka kombinací  $IO$  (vstupů, výstupů a stavů) definující strukturu SISO modulů v IDN se základní fixní topologií pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .
- Tab. 4 Abecedy  $IO$  proměnných v IDN se základní fixní topologií pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .
- Tab. 5 Tabulka kombinací  $IO$  (vstupů, výstupů a stavů) definující strukturu SISO modulů v IDN s rozšířenou fixní topologií pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .
- Tab. 6 Abecedy stavových proměnných v IDN s rozšířenou fixní topologií pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .
- Tab. 7 Tabulka kombinací  $IO$  (vstupů a výstupů) definující strukturu měkké inverze binárního broadcasteru.
- Tab. 8 Tabulka kombinací  $IO$  (vstupů a výstupů) definující strukturu SISO modulů v IDN s variabilní topologií pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .
- Tab. 9 Tabulka kombinací  $IO$  (vstupů, výstupů a stavů) definující strukturu SISO modulů ve spodní vrstvě IDN s vrstvou fixní topologií pro obecný dekomponovatelný ISI kernel  $\mathcal{G}_{3 \times 3}^{(HV)}$ .
- Tab. 10 Přehled vlastností iterativních dekódovacích sítí pro ISI kernel  $\mathcal{G}_{3 \times 3}^{(Gauss)}$ . Horní meze počtů kombinačních a marginalizačních operací (numerická složitost SISO modulů).
- Tab. 11 Přehled vlastností iterativních dekódovacích sítí pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(Gauss,R)}$ . Horní meze počtů kombinačních a marginalizačních operací (numerická složitost SISO modulů).

# 1 Úvod

Statistická detekce dat založená na MAP (Maximum A posteriori Probability) kritériu je poměrně známou a účinnou metodou. Své uplatnění má především v oblasti rádiové komunikace, ale i v jiných oblastech. Obecně řečeno, aplikací kritéria získáváme přenášená data z observace přijatého signálu na vstupu přijímací strany přenosového systému. Tento proces, označovaný jako estimace, využívá informace o statistických vlastnostech jak rušivých vlivů v přenosovém kanálu tak znalosti statistických vlastností samotných přenášených dat. Vstupní informace MAP detektoru, z hlediska terminologie, dělíme na dva typy. Informace apriorní a aposteriorní. Apriorní informační metrika je znalost předem daná, kterou detektor vnímá jako dogma a nezkontroluje její správnost. V našem případě tato informace specifikuje statistické rozdělení zdroje dat. Naopak aposteriorní metrika kvantifikuje znalost vycházející z určité prodělané zkušenosti. Zde tedy reprezentuje informaci získanou z realizace přijatého signálu. Oba typy informací detektor zkombinuje, vyhodnotí a vydá odhad dat.

Mezi klíčové aplikace MAP kritéria, v oblasti komunikací, patří bezesporu detekce modulací třídy TCM (Trellis Coded Modulations). Tyto tzv. mřížkové modulace [27, 28, 29] byly objeveny v 70. letech minulého století Gottfriedem Ungerboeckem a používají se hojně dodnes. Zakládají se na protekci přenášených dat pomocí mřížkového (konvolučního) kodéru. Tento jednoduchý stavový automat FSM (Finite State Machine) obohacuje užitečnou informaci o redundanci v podobě jistého popisu změn této informace v čase. Redundantní složka, nesoucí výstup této konvoluční operace, následně pomáhá opravnému detekčnímu procesu vyslaných dat z degradovaného signálu na straně příjmu. Sloučením vstupních dat a redundance tak vznikne tzv. systematický mřížkový kód, který lze považovat za základ kanálového (ochranného) kódování. Často se toto kódování navazuje na digitální modulátor s lineární expanzní částí, kde způsob mapování kódového toku na kanálový tok se řídí tzv. Ungerboeckovými pravidly. Detekce těchto kódů se řeší aplikací MAP kritéria. Efektivní výpočet kritéria zde provádí tzv. Viterbiho algoritmus [25], který byl objeven Andrewem Viterbim v roce 1967. Výpočetní mechanizmus algoritmu je založen na periodickém opakování dvojice úkonů během posunu algoritmu po přijatém signále v čase. První krok spočívá v kombinaci nashromážděných aposteriorních informací algoritmem z historie s aposteriorní informací o právě přichozím přírůstku přijatého signálu. Vznikne tak množina více či méně pravděpodobných scénářů událostí, které se odehrály či se mohly odehrát v konvolučním kodéru na straně vysílání. V následující fázi algoritmus provede marginalizaci těchto informací, kdy redukuje možné události o ty více nepravděpodobné. Tak si množina metrik zachovává konstantní kardinalitu a neobtěžuje s každým dalším krokem algoritmu vpřed. Výsledkem je soubor tzv. "Viterbiho cest" o mohutnosti shodné s počtem stavů nebo také konvoluční délkou použitého mřížkového kodéru, přičemž každá z těchto cest končí v jiném koncovém stavu. Viterbiho cesty lze tedy interpretovat jako nejpravděpodobnější možné scénáře průchodu možných vysílaných dat skrz mřížku kodéru. Po proběhnutí požadovaného počtu kroků je vybrána cesta s minimální celkovou Euklidovskou metrikou a data, která tuto cestu generují, jsou označena jako hledaný odhad.

Uvedený způsob kanálového kódování a detekce se také označuje jako jednostupňový a je velmi žádoucí i v současnosti díky jednoduchosti a výkonnosti. Vývoj samoopravných kódů šel však dále kupředu a další významný pokrok v této oblasti zaznamenali pánové Claud Berrou, Alain Glavieux a Punya Thitimajshima, když v roce 1993 vynalezli Turbokódy [20, 21]. Tato technika se stala skutečnou revolucí v kanálovém kódování a je založena na poměrně jednoduché, ovšem geniální myšlence. Objevitelé optimalizovali jednostupňové kódování rozkladem složitějšího stavového automatu na množinu jednodušších automatů, vzájemně separovaných pomocí prokladačů. Ty zajišťovali permutaci kódu, čímž výrazně zvýšili minimální volnou vzdálenost mezi signály na výstupu takové struktury. Uspořádání těchto "kódovacích sítí" může být velmi variabilní. Základní topologie je sériová nebo paralelní a při dostatečné délce vysílaných paketů je možné se přiblížit teoretické mezi informační propustnosti kanálu s aditivním Gaussovským šumem, která byla odvozena Claudem Shannonem [26] v roce 1949.

Detekce Turbokódů se provádí pomocí iterativních dekódovacích sítí, s měkkým rozhodováním, které svou topologií kopírují uspořádání Turbokodéru. Každý z funkčních bloků v kódovací síti na straně vysílání má pak svůj protějšek v dekódovací síti na straně detekce. Tyto komplementy nazýváme jako měkké (statistické) inverze, nebo také jako SISO (Soft-In Soft-Out) moduly. V průběhu dekódovacího procesu jsou jednotlivé moduly aktivovány pomocí tzv. aktivčního schématu. Pod tímto pojmem si můžeme představit situaci, kdy SISO modul obdrží apriorní nebo aposteriorní měkké informace od svých sousedů a na jejich základě vyprodukuje

aktualizované informace, které odešle zpět do okolních modulů. Podstatou iterativní detekce je tedy zpřesňování informace v jistých uzavřených smyčkách uvnitř detekční sítě. Klíčovým prvkem iterativní dekodovací sítě pro detekci Turbokódu je měkká inverze konečného stavového automatu. Takový SISO modul realizuje tzv. BCJR (Bahl, Cocke, Jelinek and Raviv) algoritmus [22], který byl objeven pány Lalitem R. Bahlem, Johnem Cockem, Frederickem Jelinkem, Josefem Ravivem v roce 1974. BCJR je detekční algoritmus podobný Viterbiho algoritmu. Rozdíl je pouze v tom, že zpracovává úsek signálu oběma směry a na svém výstupu neprodukuje tvrdý odhad, ale měkkou informaci. Díky tomu, že vstupní signál je současně nebo postupně (podle charakteru implementace) zpracováván od konce i začátku příslušného úseku (paketu) je BCJR algoritmus také označován jako dopředný-zpětný algoritmus FBA (Forward-Backward Algorithm). BCJR algoritmus má i svou redukovanou variantu pro detekci kódů s binárními vstupními daty. Toto zjednodušení je označováno jako Viterbiho algoritmus s měkkým výstupem SOVA (Soft-Output Viterbi Algorithm) [23].

Oblast jednodimenzionální iterativní detekce Turbokódů je v současné době dobře známá a existují o ní obsáhlé publikace, například [12, 14, 13, 15, 16]. Disertační práce navazuje na tyto poznatky a věnuje se méně známé oblasti tzv. dvoudimenzionální iterativní detekce. Dvoudimenzionální iterativní detektory tvoří zobecněné struktury klasické iterativní detekce. Zde byly předmětem detekce jednodimenzionální signály, jejichž nezávislou proměnnou byl čas. Výstupem dvoudimenzionálního iterativního detektoru jsou dvourozměrná data závislé na dvou nezávislých proměnných. V této práci budeme tyto proměnné chápat jako souřadnice v planárním prostoru. Základní idea dvoudimenzionální detekce dat byla prezentována Xiaopengem Chenem v publikaci [11] z roku 1998. Na tyto poznatky navazují práce Phunsaka Thiennviboona věnované problematice využití iterativní detekce ke generování pultónových obrázků [9, 10]. Jinou oblast využití iterativních dekodovacích sítí zkoumá předkládaná práce a sumarizuje výsledky využití této technologie v oblasti rekonstrukce rozostřených [4, 6], či pohybem rozmazaných [5] černobílých obrázků. Kromě výsledků těchto experimentů, obsahují citované publikace také nové varianty uspořádání topologie dvoudimenzionálních iterativních detektorů.

Hlavním cílem disertační práce je poskytnutí teoretického pozadí dvoudimenzionální formy iterativní detekce se zaměřením na aplikace v oblasti zpracování obrazu. Nově představené topologie detektoru budou porovnány s těmi převzatými z dřívějších pramenů a u všech variant bude zkoumána jejich aplikovatelnost na úlohu rekonstrukce obrazu. Vstupní dvoudimenzionální data budou tedy představovat obrazovou informaci, která je konvoluována deterministickým kanálem s prostorovým kernelem. Takový kanál zavádí do vstupních dat obecnou mezisymbolovou interferenci ISI (Inter-Symbol Interference). Prostorový ISI kanál můžeme chápat jako dvoudimenzionální konečný stavový automat (2D FSM) dekomponovatelný do planární kódovací sítě EN (Encoding Network) tvořené jednoduchými buňkami — kombinačními logikami GPE (General Processing Element) s obecným propojením v horizontálním a vertikálním směru. Možností propojení kombinačních logik v kódovací síti existuje celá řada. Topologie kódovací sítě dále představuje vzorovou strukturu pro uspořádání iterativní dekodovací sítě, kde každá buňka struktury je tvořena určitým SISO modulem, neboli kombinačně-marginalizačním statistickým elementem.

Úkolem dvoudimenzionálního iterativního detektoru je rekonstruovat vstupní obraz do původní podoby, podobně jako u systémů jednodimenzionálních, kde byla rekonstruována sekvence uživatelských dat. Taková úloha obsahuje v základní variantě potlačení mezisymbolové interference a aditivních náhodných fluktuací (šumů). Samotný rekonstrukční (detekční) proces ve dvoudimenzionální iterativní dekodovací síti je rovněž analogický k jednodimenzionálním variantám iterativních detektorů. Opět probíhá sekvenční a řízená aktivace jednotlivých buněk (SISO modulů) v zapojení detektoru prostřednictvím aktivačního schématu a dochází k výměně aposteriorních hustot pravděpodobnosti mezi sousedícími, vzájemně propojenými, SISO moduly.

V teoretické části disertační práce bude popsána řada možných topologií iterativní dekodovacích sítí a jejich aktivačních schémat. Na teoretickou část předkládané práce navazuje část experimentální, kde budou jednotlivé představené topologie aplikovány v úloze rekonstrukce dichromatických obrázků získaných z matematického modelu kamery se CCD senzorem. Základní varianta rekonstrukční úlohy experimentální části bude zahrnovat potlačení rozostření objektivem kamery a eliminaci šumů CCD senzoru. Rozšířená varianta zahrnuje navíc ještě de-mosaicing. Výkonnost jednotlivých topologií iterativních dekodovacích sítí bude zjištěna objektivním empirickým testováním pomocí metody Monte Carlo, kdy je měřena průměrná chybovost vstupního náhodného dichromatického obrazu při různé úrovni jeho rozostření a při různé úrovni kontaminace aditivními šумы. V závěru předkládané práce je provedena diskuze nad naměřenou výkonností jednotlivých představených struktur a jejich výpočetní náročností.

### Sumarizace cílů disertační práce:

1. Sumarizace současných poznatků o dvoudimenzionální iterativní MAP detekci. Rešerše známých topologií dvoudimenzionálních iterativních dekódovacích sítí a oblastí jejich aplikace.
2. Rozšíření známých implementací o nové topologie a uspořádání dvoudimenzionálního iterativního MAP detektoru (Fixní topologie s diagonálními propojkami, vrstevné topologie, variabilní topologie s neredukovanými detekčními buňkami).
3. Vytvoření komplexního objektově orientovaného toolboxu v programovém prostředí MATLAB, který simuluje dvoudimenzionální iterativní detektor s libovolnou topologií uvedenou v této disertační práci. Toolbox obsahuje 20 tříd o celkovém počtu 115 m-skriptů a je zastřešen 3 spouštěcími m-skripty. Nastavení detektoru (konstrukce topologie iterativní dekódovací sítě) je plně automatické a to podle hodnot koeficientů zadaného kernelu libovolného tvaru.
4. Ověření správnosti a účinnosti předkládaných řešení dvoudimenzionálních iterativních detektorů na vybrané aplikaci, která provádí binarizaci digitálních fotografií získaných pomocí kamery se CCD senzorem.
5. Porovnání výkonnosti a vlastností všech obsažených topologií dekódovacích sítí mezi sebou a to v oblasti dichromatické rekonstrukce obrazu (binarizační úloha zahrnující potlačení rozostření, potlačení šumů a demosaicing). Výkonnost jednotlivých řešení je stanovena na základě empirického měření chybovosti binarizovaných obrazů metodou Monte Carlo v programovém prostředí MATLAB pomocí vytvořeného simulačního toolboxu a matematického modelu kamery se CCD senzorem.
6. Diskuse nad vlastnostmi jednotlivých předložených detektorů. Především z hlediska výkonnosti a flexibility versus implementační náročnost.

## 2 Obecný model systému

Předpokládejme obecný dvoudimenzionální přenosový systém na Obr. 1. Vstup systému tvoří matice (zdrojových) dat  $\mathbf{D}$ , jejíž prvky jsou diskrétní v hodnotách. V systémovém řetězci jsou zdrojová data nejprve postoupena do obecného deterministického ISI (Inter-symbol interference) kanálu. Působením takového kanálu (enkodéru) vznikne v datech, obecně dvoudimenzionální, mezisymbolová interference. Konvoluční produkt na výstupu kanálu označme maticí  $\mathbf{Q}$ . Elementy matice budou rovněž diskrétní v hodnotách a nazývávejme je kódovými symboly. Hodnoty matice  $\mathbf{Q}$  budou tedy záviset na datech  $\mathbf{D}$  a dvoudimenzionální impulsové odezvě ISI kanálu. Tuto odezvu označme pomocí množiny  $\mathcal{G}$ .

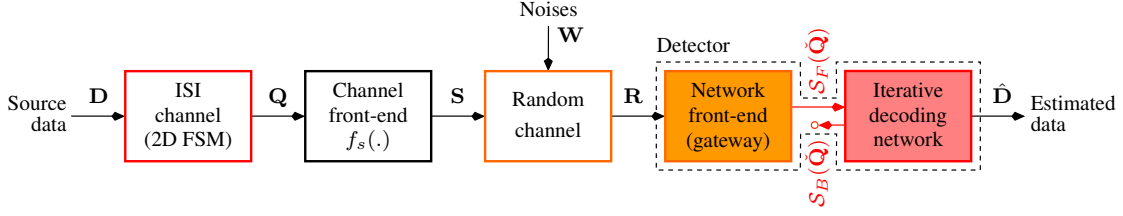
Matice kódu je následně modifikována kanálovým front-endem. Tento blok je definován prostřednictvím funkce  $f_s(\cdot)$  provádějící jednoznačné zobrazení kódu  $\mathbf{Q}$  na signál  $\mathbf{S} = f_s(\mathbf{Q})$ , který vyslaný do přenosového kanálu. Transformační funkce front-endu bude bez paměti a často lineární. Front-end tedy chápeme jako jednoduchý obvod sloužící k normování signálu  $\mathbf{Q}$  do určitého (požadovaného) intervalu hodnot.

V přenosovém kanále je vyslaný deterministický signál  $\mathbf{S}$  kontaminován a znáhodněn rušivými vlivy  $\mathbf{W}$ . Tyto stochastické procesy budeme uvažovat jako aditivní nekorelované šумы. Ve většině případů budeme předpokládat pouze jeden zdroj šumu a to Gaussovský bílý šum AWGN (Additive White Gaussian Noise).

Degradovaný signál průchodem skrze náhodný kanál označíme maticí  $\mathbf{R}$ . Tato observace tvoří vstup iterativního detektoru, který je na Obr. 1 lemován čárkovanou čarou. Samotný detektor se skládá z kaskády dvou klíčových bloků. První blok je dvoudimenzionální front-end detektoru (Gateway) a ten následuje iterativní dekódovací síť IDN (Iterative decoding network). Dekódovací síť tvoří jádro celého iterativního detektoru. Jedná se o ryze statistický systém, který nepracuje s izolovanými hodnotami, ale s celými hustotami pravděpodobností PDF (Probability Density Function). Proto úkolem front-endu detektoru je příprava těchto vstupních aposteriorních metrik a to na základě získané realizace  $\mathbf{R}$  a apriorní znalosti distribucí šumů v kanále. Hodnotu metriky (velikost konkrétního prvku v určité diskrétní hustotě) budeme značit obecným symbolem  $\mathcal{S}(\cdot)$  s tím, že se může jednat přímo o pravděpodobnost (veličinu s oborem hodnot od 0 do 1) nebo její transformaci přes určité jednoznačné zobrazení. Fakt, zda-li se jedná o apriorní nebo aposteriorní měkkou metriku bude určovat její index. Metriky označené indexem  $F$  budeme chápat jako metriky aposteriorní nebo také “dopředné”. Naproti tomu index  $B$  bude specifikovat metriky apriorní nebo také “zpětné”. Pokud metriky budou uvedené bez indexu, budeme předpokládat, že se jedná o metriky “sdružené”. Argument každé měkké metriky tvoří tzv. testovací estimátor  $\xi$ , což je nějaká, hypoteticky možná, hodnota z oboru hodnot té či oné diskrétní náhodné veličiny  $\xi \in \{\xi^{(i)}\}_i$ . Pomocí testovacího estimátoru  $\xi$  tedy budeme přistupovat (adresovat) ke konkrétní metrice  $\mathcal{S}(\xi)$ , v určité diskrétní hustotě  $\{\mathcal{S}(\xi^{(i)})\}_i$ , jež nás zrovna zajímá.

V okamžiku, kdy front-end detektoru transformuje přijatou realizaci  $\mathbf{R}$  na odpovídající množství dopředných hustot  $\{\mathcal{S}_F(\mathbf{Q})\}$  začíná samotná iterativní detekce uvnitř jádra detektoru. Cílem tohoto procesu je měkká dekonvoluce ISI kanálu a potlačení aditivního šumu  $\mathbf{W}$ , který byl na užitečný signál  $\mathbf{S}$  superponován v přenosovém kanále. Faktickým výstupem procesu je výpočet výstupních sdružených hustot vyslaných dat  $\mathbf{D}$ . Měkké výstupy iterativní dekódovací sítě jsou opatřeny rozhodovacími bloky DEC (Decision block). Předpokládejme, že tyto bloky tvoří nedílnou součást IDN a proto nejsou na Obr. 1 explicitně zakresleny. Jejich úkolem je prahování finálních výstupních sdružených hustot do tvrdého odhadu dat  $\hat{\mathbf{D}}$  po skončení iterativního detekčního procesu v jádře detektoru. Tuto prahovací operaci chápeme jako nalezení testovacího estimátoru s maximální věrohodností (estimátor navázaný na maximální resp. minimální hodnotu metriky ve sdružené hustotě), který je následně označen jako hledaný odhad.

V následujícím textu navážeme na tento počáteční a stručný popis přenosového systému. Zaměříme se na jeho jednotlivé části, které zevrubně rozebereme. Dále se zaměříme na možnosti použití tohoto konceptu v určitých reálných detekčních aplikacích.



Obr. 1. Blokové schéma obecného dvoudimenzionálního přenosového systému.

## 2.1 Deterministický kanál s mezisymbolovou interferencí

V předešlém odstavci 2 byla stručně popsána podstata ISI kanálu, který uvozuje celý řetězec na Obr. 1. Bylo řečeno, že provádí určitou dvoudimenzionální konvoluci nad vstupními daty  $\mathbf{D}$ . Z tohoto úhlu pohledu vykonává stejnou operaci jako například klasický mřížkový kódér u modulací třídy TCM o nichž jsme v krátkosti hovořili v úvodní kapitole. Rozdíl je pouze v dimensionalitě obou úloh. Proto ISI kanál můžeme rovněž chápat jako dvoudimenzionální konečný stavový automat (2D FSM), protože za předpokladu vstupu  $\mathbf{D}$ , který je diskretní v hodnotách, jím skutečně je. Z toho důvodu byl v předešlém textu výstup kanálu  $\mathbf{Q}$  pojmenován též jako kód a ISI kanál samotný jako enkodér.

Přenosové vlastnosti ISI kanálu závisí na jeho konvolučním jádře, tzv. “kernelu”. Kernel označme pomocí obecné množiny izolovaných hodnot  $\mathcal{G} = \{g[k', l']\}_{k', l' \in \mathcal{L}(\mathcal{G})}$  ve dvoudimenzionálním prostoru. Množství těchto koeficientů nazýváme jako mohutnost kernelu  $\mathcal{L}(\mathcal{G})$ . Rozmístění koeficientů kernelu pak definuje konvoluční region  $\mathcal{N}[k, l] = \{d[k + k', l + l']\}_{k', l' \in \mathcal{L}(\mathcal{G})}$ , tedy jistou podmnožinu vstupních dat  $\mathbf{D} = [d[k, l]]_{k, l}$ . Na základě těchto definic můžeme vyjádřit 2D konvoluci, prováděnou ISI kanálem, pomocí vztahu

$$\begin{aligned} q[k, l] &= f(\mathcal{G}, \mathcal{N}[k, l]) \\ &= \sum_{k', l' \in \mathcal{L}(\mathcal{G})} g[k', l'] d[k + k', l + l'], \end{aligned} \quad (1)$$

kde výstupem je kódový symbol na příslušné pozici  $[k, l]$ . Obor hodnot kódových symbolů, v matici kódu  $\mathbf{Q} = [q[k, l]]_{k, l}$ , označme množinou (abecedou)

$$\begin{aligned} q[k, l] &\in \mathcal{A}_q \\ &= \{q^{(i)}\}_{i < M_q}, \end{aligned} \quad (2)$$

kde  $M_q$  je mohutnost abecedy a nerovnost  $i < M_q$  značí sekvenci indexů  $i = 0, 1, \dots, M_q - 1$ . Hodnoty prvků abecedy  $\mathcal{A}_q$  i jejich počet bude záviset jednak na kernelu  $\mathcal{G}$  a také na abecedě vstupních dat

$$\begin{aligned} d[k, l] &\in \mathcal{A}_d \\ &= \{d^{(i)}\}_{i < M_d}. \end{aligned} \quad (3)$$

Tím bychom měli uzavřený základní matematický popis ISI kanálu a definice jednotlivých zúčastněných veličin. V souvislosti s ISI kanálem se budeme dále zabývat problematikou výpočtu konvoluce (1). V kapitole 3 ukážeme jak lze faktorizovat vztah (1) do zřetěžené struktury tvořené určitými elementárními funkčními bloky (GPE) společně s popisem možností a variant jejich propojení.



## 2.2 Náhodný nekorelovaný kanál bez paměti

Dalším blokem v řetězci na Obr. 1 je náhodný kanál bez paměti (ML) s nezávislými eliminovanými stavy (IECS). Při průchodu signálu  $\mathbf{S}$  tímto kanálem

$$\mathbf{R} = \mathbf{S} + \mathbf{W} \quad (4)$$

dochází pouze k jeho znáhodnění (kontaminaci) aditivním šumem  $\mathbf{W}$  (kanál je ML). Hodnoty šumu předpokládáme jako vzájemně nekorelované v prostoru  $[k, l]$  (kanál je IECS). Proto můžeme vstupně-výstupní vztah kanálu dekomponovat na úroveň jednotlivých symbolů a zapsat jej ve skalární formě

$$r[k, l] = s[k, l] + w[k, l], \quad (5)$$

kde každá hodnota šumu  $w[k, l]$  má stejné statistické rozdělení  $p_w(\xi)$ . Toto rozdělení předpokládáme jako Gaussovské s obecně nenulovou střední hodnotou.

Náhodný kanál s aditivním šumem (4) lze statisticky vyjádřit podmíněnou hustotou

$$p_{\mathbf{R}|\mathbf{Q}, \mathbf{W}}(\Xi|\check{\mathbf{Q}}, \Xi') = \delta(\Xi - (f_s(\check{\mathbf{Q}}) + \Xi')), \quad (6)$$

kde  $\delta(\cdot)$  značí Diracův puls. V uvedené funkci vystupuje možná realizace šumu  $\mathbf{W}$  prostřednictvím proměnné  $\Xi'$ . Jelikož je kanál IECS, pak statistické rozdělení matice  $\mathbf{W}$  definuje sdružená hustota

$$p_{\mathbf{W}}(\Xi) = \prod_{k,l} p_w(\xi), \quad (7)$$

která je produktem marginálních hustot  $p_w(\xi)$ . Eliminací (vyprůměrováním)

$$p_{\mathbf{R}}(\Xi|\check{\mathbf{Q}}) = \int p_{\mathbf{R}|\mathbf{Q}, \mathbf{W}}(\Xi|\check{\mathbf{Q}}, \Xi') p_{\mathbf{W}}(\Xi') d\Xi' \quad (8)$$

sdružené hustoty (7) z podmíněné hustoty (6) získáme věrohodnostní funkci kanálu s aditivním šumem

$$\begin{aligned} p_{\mathbf{R}}(\Xi|\check{\mathbf{Q}}) &= \int \delta(\Xi - (f_s(\check{\mathbf{Q}}) + \Xi')) p_{\mathbf{W}}(\Xi') d\Xi' \\ &= p_{\mathbf{W}}(\Xi - f_s(\check{\mathbf{Q}})). \end{aligned} \quad (9)$$

Za předpokladu, že funkce kanálového front-endu  $f_s(\cdot)$  je lineární (jednoznačné zobrazení mezi  $s[k, l]$  a  $q[k, l]$ ) a s přihlédnutím ke skutečnosti, že kanál je typu ML-IECS můžeme věrohodnostní funkci zapsat jako produkt

$$\begin{aligned} p_{\mathbf{R}}(\Xi|\check{\mathbf{Q}}) &= \prod_{k,l} p_w(\xi - f_s(\check{q}[k, l])) \\ &= \prod_{k,l} p_r(\xi|\check{q}[k, l]) \end{aligned} \quad (10)$$

marginálních příspěvků  $p_r(\xi|\check{q}[k, l])$ . Tato vlastnost věrohodnostní funkce je zcela klíčová. Zajišťuje faktorizovatelnost celé detekční úlohy a umožňuje přechod od jednostupňové MAP detekce k iterativní MAP detekci, která je prakticky realizovatelná.

### 3 Kódovací sítě — buňkové modely kanálů s mezisymbolovou interferencí

V této kapitole navážeme na matematickou definici ISI kanálu z odstavce 2.1 a budeme se zabývat otázkou implementace konvoluce (1). Bylo řečeno, že ISI kanál je diskretním systémem díky vstupům, které jsou diskretní v hodnotách. Proto lze konvoluci (1) realizovat pomocí dvoudimenzionální sítě vzájemně propojených elementů (GPE) s diskretním definičním oborem a diskretním oborem hodnot. Taková kódovací síť tvoří, jako celek, dvoudimenzionální konečný stavový automat (2D FSM) a její struktura slouží jako předloha pro konstrukci iterativní dekódovací sítě.

#### 3.1 Elementy kódovacích sítí

Elementy kódovací sítě schematicky označme dle Obr. 2a a předpokládejme, že mají obecný soubor vstupů a výstupů

$$\mathcal{V}_{IO}[\ell] \in \{\mathcal{V}_{IO}^{(i)}[\ell]\}_{i \in \mathcal{L}(\mathcal{V}_{IO}[\ell])} \quad (11)$$

s diskretním oborem hodnot  $\mathcal{V}_{IO}^{(i)}[\ell]$  o mohutnosti  $\mathcal{L}(\mathcal{V}_{IO}[\ell])$ . Hodnoty proměnných na všech portech elementu uspořádáme do množiny

$$\mathcal{N}_{IO} = \bigcup_{\ell} \mathcal{V}_{IO}[\ell] \quad (12)$$

a hodnoty pouze vstupních proměnných elementu do množiny

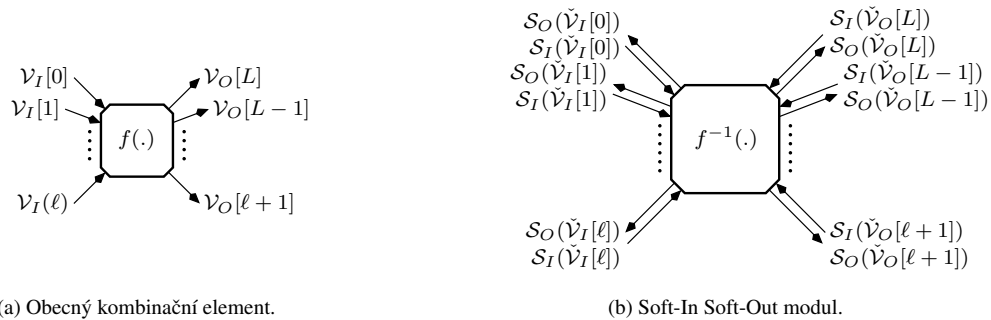
$$\mathcal{N} = \bigcup_{\ell} \mathcal{V}_I[\ell] \quad (13)$$

Potom hodnoty na výstupu elementu

$$\{\mathcal{V}_O[\ell]\}_{\ell} = f(\mathcal{G}, \mathcal{N}) \quad (14)$$

jsou funkcí kernelu  $\mathcal{G}$  a vstupů  $\mathcal{N}$ .

Elementy budeme primárně chápat jako sduženou kombinační logiku tvořící automat Mealyho typu. Proto značka na Obr. 2a obsahuje více výstupních portů, přestože výstup ISI kanálu (1) je pouze jeden. Tyto výstupy “navíc” budeme vnímat jako určité stavové proměnné. Konkrétně výstupní (koncové) stavy, které reprezentují vstupní (počáteční) stavy v sousedních elementech. Sekundárně budou elementy tvořit klasickou kombinační logiku, kdy nejsou výstupní stavové proměnné přítomny. Element bude mít pouze jeden výstup a proto mohutnost  $\text{card}(\mathcal{N}_{IO}) = \text{card}(\mathcal{N}) + 1$ .



Obr. 2. Obecný kombinační element (GPE) a jeho měkká inverze (SISO modul).

## 3.2 Topologie kódovacích sítí

Vzájemné propojení elementů v kódovací síti můžeme rozdělit do tří základních variant. Na fixní topologie (FT), variabilní topologie (VT) a vrstvené topologie (LT). Jednotlivé topologie budou popsány v následujících odstavcích 3.2.1, 3.2.2 a 3.2.3.

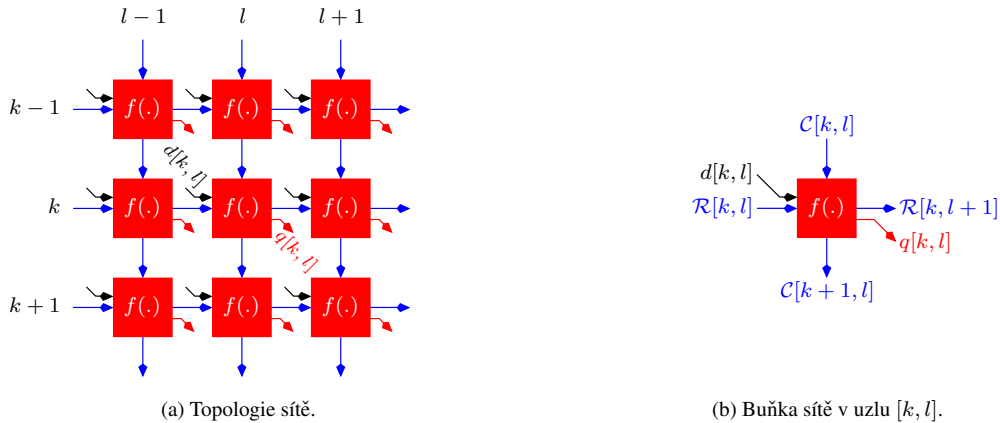
### 3.2.1 Pevné topologie

Fixní topologie (FT) představuje nejjednodušší možné uspořádání kódovací sítě. Spočívá v propojení kombinačních elementů v horizontálním a vertikálním směru [12]. Kódovací síť s fixní topologií znázorňuje Obr. 3a, kde každý její uzel tvoří jediný funkční blok na Obr. 3b s fixním počtem vstupů a výstupů. Tyto porty tvoří  $IO$  (vstupně-výstupní) množinu

$$\mathcal{N}_{IO}[k, l] = \{\mathcal{R}[k, l], \mathcal{C}[k, l], d[k, l], \mathcal{R}[k, l + 1], \mathcal{C}[k + 1, l], q[k, l]\}, \quad (15)$$

kde  $\mathcal{R}[k, l]$  a  $\mathcal{C}[k, l]$  jsou pomocné stavové proměnné obsahující více datových symbolů dohromady. Tvary těchto pomocných proměnných vyplývají z podmínky kauzality sítě

$$\begin{aligned} \mathcal{N}[k, l] &\subseteq \mathcal{R}[k, l] \cup \mathcal{C}[k, l] \cup d[k, l] \\ &\supseteq \mathcal{R}[k, l + 1], \mathcal{C}[k + 1, l]. \end{aligned} \quad (16)$$



Obr. 3. Kódovací síť se základní fixní topologií.

Pro názornost uved' me jednoduchý příklad FT kódovací sítě. Předpokládejme ISI kanál s obecným čtvercovým posunutým kernelem  $3 \times 3$

$$\mathcal{G}_{3 \times 3} = \{g[k', l']\}_{-2 \leq k', l' \leq 0}. \quad (17)$$

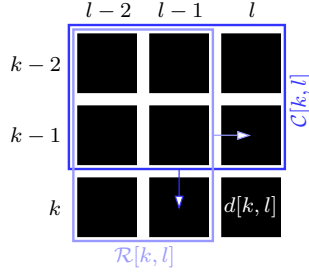
Optimální dekompozice konvolučního regionu  $\mathcal{N}[k, l] = \mathcal{R}[k, l] \cup \mathcal{C}[k, l] \cup d[k, l]$  je možná pouze za situace, kdy  $g(0, 0) \neq 0$ . V tomto případě, a s přihlédnutím k podmínce (16), jsou pomocné stavové proměnné

$$\mathcal{R}[k, l] = \left\{ \begin{array}{cc} d[k-2, l-2] & d[k-2, l-1] \\ d[k-1, l-2] & d[k-1, l-1] \\ d[k, l-2] & d[k, l-1] \end{array} \right\} \quad (18)$$

a

$$\mathcal{C}[k, l] = \left\{ \begin{array}{ccc} d[k-2, l-2] & d[k-2, l-1] & d[k-2, l] \\ d[k-1, l-2] & d[k-1, l-1] & d[k-1, l] \end{array} \right\}. \quad (19)$$

Jejich tvary znázorňuje Obr. 4.



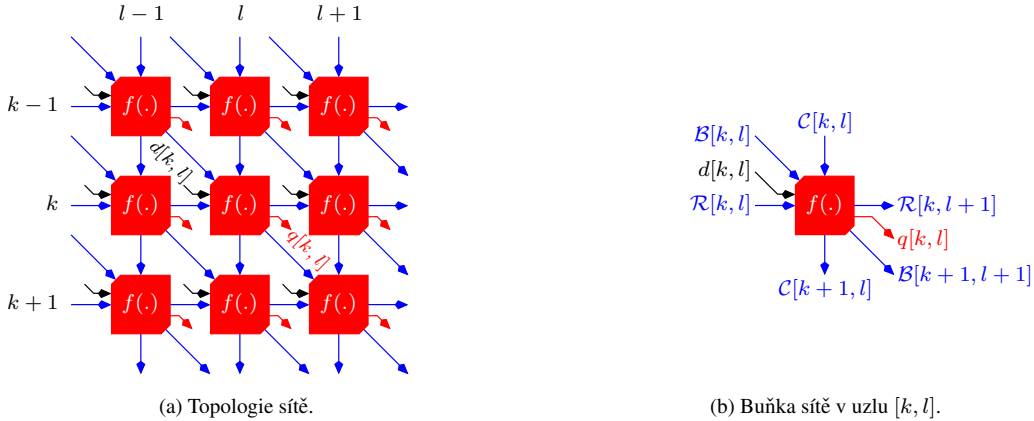
Obr. 4. Tvary stavových proměnných  $\mathcal{R}[k, l]$  a  $\mathcal{C}[k, l]$  pro obecný ISI kernel  $\mathcal{G}_{3 \times 3}$ .

Rozšířená varianta fixní topologie (EFT) obsahuje také propojení diagonální [4]. Kódovací síť s rozšířenou fixní topologií znázorňuje Obr. 5a. Zde každý uzel tvoří rozšířený element na Obr. 3b. Porty elementu tvoří IO množinu

$$\mathcal{N}_{IO}[k, l] = \{\mathcal{R}[k, l], \mathcal{C}[k, l], \mathcal{B}[k, l], d[k, l], \mathcal{R}[k, l+1], \mathcal{C}[k+1, l], \mathcal{B}[k+1, l+1], q[k, l]\}. \quad (20)$$

Oproti předcházející FT variantě přibyla další pomocná stavová proměnná  $\mathcal{B}[k, l]$ , která je uvnitř struktury předávána na diagonálních spojnících. Tvary trojice stavových proměnných vyplývají z rozšířené podmínky kauzality sítě

$$\begin{aligned} \mathcal{N}[k, l] &\subseteq \mathcal{R}[k, l] \cup \mathcal{C}[k, l] \cup \mathcal{B}[k, l] \cup d[k, l] \\ &\supset \mathcal{R}[k, l+1], \mathcal{C}[k+1, l], \mathcal{B}[k+1, l+1]. \end{aligned} \quad (21)$$



Obr. 5. Kódovací síť s rozšířenou fixní topologií.

Podmínka (21) není tak svazující jako u FT kódovací sítě (16) a dovoluje větší volnost a variabilitu volby tvarů stavových proměnných. Přibližme tuto skutečnost opět na předešlém příkladě s posunutým kernelem  $\mathcal{G}_{3 \times 3}$  a to prostřednictvím dvou různých variant. V prvním příkladě je kontura všech stavových proměnných

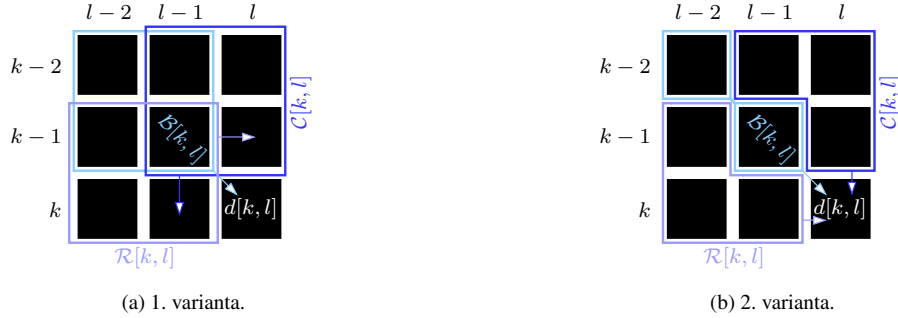
$$\begin{aligned} \mathcal{R}[k, l] &= \left\{ \begin{array}{cc} d[k-1, l-2] & d[k-1, l-1] \\ d[k, l-2] & d[k, l-1] \end{array} \right\}, \quad \mathcal{C}[k, l] = \left\{ \begin{array}{cc} d[k-2, l-1] & d[k-2, l] \\ d[k-1, l-1] & d[k-1, l] \end{array} \right\} \\ \text{a } \mathcal{B}[k, l] &= \left\{ \begin{array}{cc} d[k-2, l-2] & d[k-2, l-1] \\ d[k-1, l-2] & d[k-1, l-1] \end{array} \right\} \end{aligned} \quad (22)$$

stejná a čtvercová. Stavby sdružují čtveřice datových symbolů a vykazují významný vzájemný překryv. V druhém příkladě se kontury stavů

$$\mathcal{R}[k, l] = \left\{ \begin{array}{cc} d[k-1, l-2] & \\ d[k, l-2] & d[k, l-1] \end{array} \right\}, \quad \mathcal{C}[k, l] = \left\{ \begin{array}{cc} d[k-2, l-1] & d[k-2, l] \\ & d[k-1, l] \end{array} \right\} \quad (23)$$

$$\text{a } \mathcal{B}[k, l] = \left\{ \begin{array}{c} d[k-2, l-2] \\ d[k-1, l-1] \end{array} \right\}$$

liší. Mají minimální možnou mohutnost a jsou vzájemně disjunktní. Oba uvedené příklady tvarů stavových proměnných znázorňuje Obr. 6.



Obr. 6. Tvary stavových proměnných  $\mathcal{R}[k, l]$ ,  $\mathcal{C}[k, l]$  a  $\mathcal{B}[k, l]$  pro obecný ISI kernel  $\mathcal{G}_{3 \times 3}$ .

Závěrem proved' me stručnou sumarizaci vlastností FT resp. EFT kódovacích sítí. V obou případech je topologie nezávislá na tvaru kernelu  $\mathcal{G}$ . Elementy si předávají stavové proměnné  $\mathcal{R}$  a  $\mathcal{C}$  nebo také  $\mathcal{B}$  ve vertikálním a horizontálním nebo také diagonálním směru. Stavové proměnné obsahují fragment příslušného konvolučního regionu  $\mathcal{N}[k, l]$  a mohou být vzájemně disjunktní, ale také nemusí. Počáteční stavby  $\mathcal{R}[k, l]$ ,  $\mathcal{C}[k, l]$  a  $\mathcal{B}[k, l]$  společně s daty  $d[k, l]$  tvoří celý konvoluční region  $\mathcal{N}[k, l]$  a vstupní data splývají s jeho pravým spodním rohem. Koncové stavby  $\mathcal{R}[k, l+1]$ ,  $\mathcal{C}[k+1, l]$ ,  $\mathcal{B}[k+1, l+1] \subset \mathcal{N}[k, l]$ , přenesené do sousedních elementů, následně vzniknou posunutím kontury stavových proměnných v příslušném směru. Tvary stavových proměnných musí být zvoleny tak, aby neodporovaly podmínce kauzality (16) resp. (21).

### 3.2.2 Variabilní topologie

Odlíšný přístup k výpočtu (1) představuje variabilní topologie (VT). Nazýváme ji tak z toho důvodu, že je závislá na tvaru kernelu  $\mathcal{G}$ . Sít' obsahuje dva druhy kombinačních elementů a obecně vyšší množství propojek v různých směrech. Zpracování probíhá ve dvou úrovních. Na spodní úrovni dochází ke zkopírování vstupních dat  $d[k, l]$  do stavových veličin  $c[k, l, \ell]$  prostřednictvím tzv. broadcasterů. Tyto elementy, s IO množinou  $\mathcal{N}_{IO}^{(B)}[k, l] = d[k, l] \cup \{c[k, l, \ell]\}$ , vytvoří veškeré, vzájemně překryté, konvoluční regiony  $\mathcal{N}[k, l] = \{c[k+k', l+l', \ell]\}$ . Vzniklé regiony jsou následně zkombinovány do výstupů  $q[k, l]$  pomocí kombinačních elementů na horní úrovni. Proměnné na portech těchto elementů sdružuje množina

$$\mathcal{N}_{IO}[k, l] = \mathcal{N}[k, l] \cup q[k, l]. \quad (24)$$

Kódovací sít' VT nelze obecně graficky znázornit, protože se různí podle tvaru ISI kanálu. Proto ji přiblížíme na konkrétním případu kernelu. Předpokládejme ISI kanál s obecným čtvercovým centrováním kernelu  $3 \times 3$

$$\mathcal{G}_{3 \times 3} = \{g[k', l']\}_{-1 \leq k', l' \leq 1}. \quad (25)$$

V tomto případě je IO množina broadcasterů na spodní úrovni struktury ve tvaru

$$\mathcal{N}_{IO}^{(B)}[k, l] = d[k, l] \cup \{c[k, l, \ell]\}_{\ell < 9} \quad (26)$$

a jednotlivé broadcastery kopírují a distribuují hodnotu  $d[k, l]$  do osmi nejbližších buněk prostřednictvím proměnných  $\{c[k, l, \ell]\}_{\ell < 8}$ . Vzniklé stavové proměnné jsou následně postoupeny do kombinačních elementů na horní úrovni. Zde společně vytvoří jednotlivé konvoluční regiony

$$\mathcal{N}[k, l] = \{c[k + \{\mathcal{I}_k\}_\ell, l + \{\mathcal{I}_l\}_\ell, \ell]\}_{\ell < 9}, \quad (27)$$

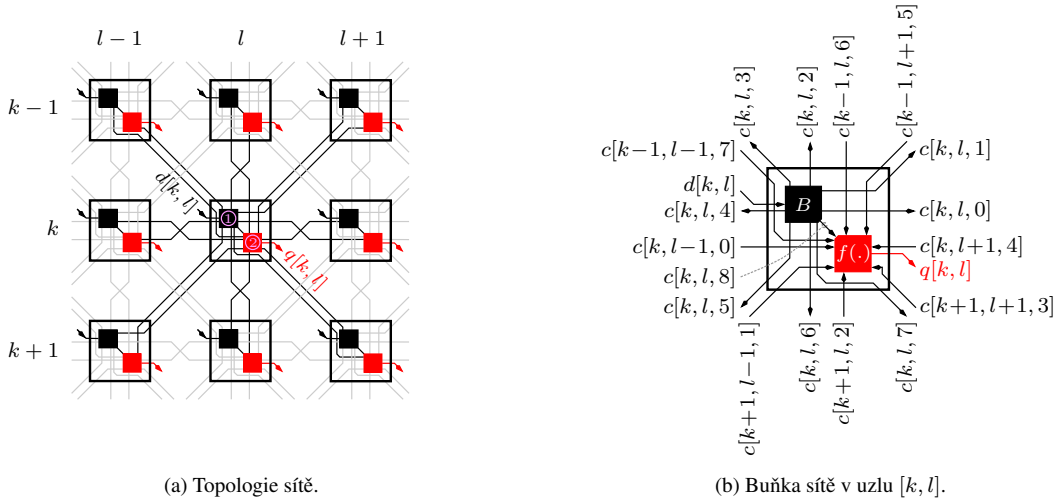
kde

$$\mathcal{I}_k = \{0, +1, +1, +1, 0, -1, -1 - 1, 0\} \quad (28)$$

a

$$\mathcal{I}_l = \{-1, -1, 0, +1, +1, +1, 0, -1, 0\} \quad (29)$$

jsou množiny indexů definující propojení mezi oběma úrovněmi ve struktuře s ohledem na tvar předpokládaného kernelu. Uvedený matematický popis reflektuje zapojení na Obr. 7a, kde jednotlivé uzly tvoří dvojice elementů na Obr. 7b. Pro jiný tvar kernelu je situace analogická.



(a) Topologie sítě.

(b) Buňka sítě v uzlu  $[k, l]$ .

Obr. 7. Kódovací síť s variabilní topologií pro obecný ISI kernel  $\mathcal{G}_{3 \times 3}$ .

Mezi přednosti variabilní topologie patří minimální mohutnost stavových proměnných  $c[k, l, \ell]$  a maximální flexibilita umožňující snadné modelování ISI kanálů s nestandardním tvarem kernelu. Nevýhoda spočívá ve složitější struktuře, s velkým množstvím propojek, a ve skutečnosti, že každý uzel tvoří dva elementy namísto jednoho jak tomu bylo v případě FT resp. EFT kódovacích sítí.

### 3.2.3 Vrstvové sítě pro kanály s dekomponovatelným kernelem

Především kódovací sítě FT resp. EFT tak VT jsou schopny modelovat kernel libovolného tvaru. Třetí typ topologie, který označujeme jako vrstvený, takový není. Tento typ můžeme aplikovat pouze na určité druhy kernelů, kdy je konvoluce (1) separovatelná do vertikálního a horizontálního směru resp. do dvou ortogonálních směrů. Pro takové kernely platí, že

$$\mathcal{G}^{(HV)} = \mathcal{G}^{(V)} \times \mathcal{G}^{(H)}, \quad (30)$$

kde  $\times$  značí kartézský součin. Separovatelný kernel má vždy čtvercový nebo obdélníkový tvar a konvoluci (1) je možné dekomponovat na dvojici následných jednodimenzionálních konvolucí

$$\begin{aligned} c[k, l] &= f_H(\mathcal{G}^{(H)}, \mathcal{N}^{(H)}[k, l]) \\ &= \sum_{l' \in \mathcal{L}(\mathcal{G}^{(H)})} g_H[l'] d[k, l + l'] \end{aligned} \quad (31)$$

a

$$\begin{aligned} q[k, l] &= f_V(\mathcal{G}^{(V)}, \mathcal{N}^{(V)}[k, l]) \\ &= \sum_{k' \in \mathcal{L}(\mathcal{G}^{(V)})} g_V[k'] c[k + k', l], \end{aligned} \quad (32)$$

kde  $\mathcal{G}^{(H)} = \{g_H[l']\}_{l' \in \mathcal{L}(\mathcal{G}^{(H)})}$  resp.  $\mathcal{G}^{(V)} = \{g_V[k']\}_{k' \in \mathcal{L}(\mathcal{G}^{(V)})}$  je horizontální resp. vertikální jednodimenzionální kernel na spodní resp. vrchní vrstvě a  $\mathcal{N}^{(H)}[k, l] = \{d[k, l + l']\}_{l' \in \mathcal{L}(\mathcal{G}^{(H)})}$  resp.  $\mathcal{N}^{(V)}[k, l] = \{c[k + k', l]\}_{k' \in \mathcal{L}(\mathcal{G}^{(V)})}$  je horizontální resp. vertikální jednodimenzionální konvoluční region na spodní resp. vrchní vrstvě.

Obě vrstvy kódovací sítě jsou vzájemně nezávislé a mohou být implementovány kterýmkoliv z předcházejících způsobů uspořádání i jejich kombinacemi. Pro názornost uvedme dvě konkrétní varianty uspořádání vrstvené sítě. Předpokládejme, že obě implementace budou na obou vrstvách používat tentýž druh topologie a budou realizovat obecný čtvercový dekomponovatelný kernel  $3 \times 3$

$$\begin{aligned} \mathcal{G}_{3 \times 3}^{(HV)} &= \begin{Bmatrix} g'_V \\ g_V \\ g'_V \end{Bmatrix} \times \{ g'_H \quad g_H \quad g'_H \} \\ &= \begin{Bmatrix} g'_V g'_H & g'_V g_H & g'_V g'_H \\ g_V g'_H & g_V g_H & g_V g'_H \\ g'_V g'_H & g'_V g_H & g'_V g'_H \end{Bmatrix}. \end{aligned} \quad (33)$$

První ukázka se týká vrstvené kódovací sítě (LT), která využívá na obou vrstvách kódovací podsítě s fixní topologií. Podsít' FT na spodní vrstvě má kombinační elementy propojené v horizontálním směru. Porty elementů tvoří IO množinu

$$\mathcal{N}_{IO}^{(H)}[k, l] = \{ \mathcal{R}[k, l], d[k, l], \mathcal{R}[k, l + 1], c[k, l] \}, \quad (34)$$

kde

$$\mathcal{R}[k, l] = \{ d[k, l - 2] \quad d[k, l - 1] \} \quad (35)$$

je horizontální stavová proměnná respektující podmínku kauzality

$$\begin{aligned} \mathcal{N}^{(H)}[k, l] &= \mathcal{R}[k, l] \cup d[k, l] \\ &\supset \mathcal{R}[k, l + 1]. \end{aligned} \quad (36)$$

Podsít' FT na vrchní vrstvě má kombinační elementy propojené ve vertikálním směru. Porty elementů tvoří IO množinu

$$\mathcal{N}_{IO}^{(V)}[k, l] = \{ \mathcal{C}[k, l], c[k, l], \mathcal{C}[k + 1, l], q[k, l] \}, \quad (37)$$

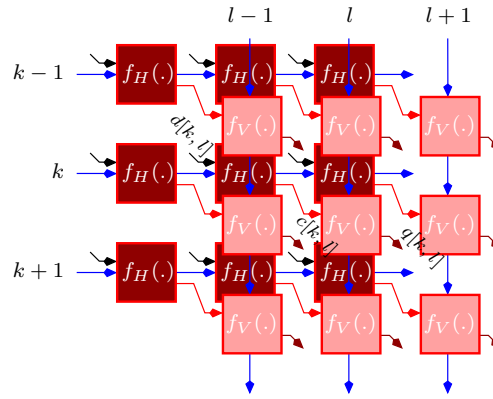
kde

$$\mathcal{C}[k, l] = \left\{ \begin{array}{l} d[k - 2, l] \\ d[k - 1, l] \end{array} \right\} \quad (38)$$

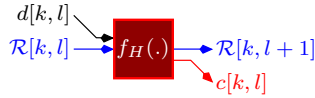
je vertikální stavová proměnná, která splňuje analogickou podmínku

$$\begin{aligned} \mathcal{N}^{(V)}[k, l] &= \mathcal{C}[k, l] \cup c[k, l] \\ &\supset \mathcal{C}[k+1, l]. \end{aligned} \quad (39)$$

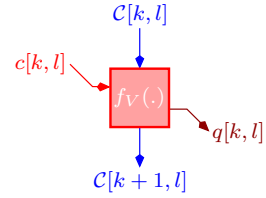
Celou výše popsanou strukturu znázorňuje Obr. 8a, kde jednotlivé uzly na spodní resp. vrchní vrstvě tvoří jediný funkční blok na Obr. 8b resp. Obr. 8c. Proces kódování probíhá ve dvou krocích. Nejprve je realizován paralelní výpočet řádkových konvolucí (31) na spodní vrstvě. Následně jsou mezi-vrstvové stavové proměnné  $c[k, l]$  předány do vrchní vrstvy a je spuštěn paralelní výpočet všech sloupcových konvolucí (32) na vrchní vrstvě. Uvedená topologie na Obr. 8a je obecně platná a zůstává beze změny i pro jiný (libovolně veliký) dekomponovatelný kernel. Přesto, že je v tomto výkladu asociována ke kernelu  $\mathcal{G}_{3 \times 3}^{(HV)}$  konkrétní velikosti  $3 \times 3$ .



(a) Topologie vrstevné sítě sestavené ze dvou podsítí s fixní topologií.



(b) Buňka sítě v uzlu  $[k, l]$  spodní vrstvy.



(c) Buňka sítě v uzlu  $[k, l]$  vrchní vrstvy.

Obr. 8. Kódovací síť s fixní vrstevnou topologií.

Druhý příklad se týká vrstevné sítě (LVT), která je složena ze dvou podsítí s variabilní topologií. Proces zpracování vstupních dat  $d[k, l]$  začíná na spodní úrovni podsítě VT ve spodní vrstvě struktury. Data vstupují do broadcasterů, kde dochází k jejich klonování, prostřednictvím stavových veličin  $c[k, l, \ell]$ , a odeslání do nejbližších uzlů po levé i pravé straně každého broadcasteru v jednotlivých horizontálách. Vstupy a výstupy broadcasterů ve spodní podsíti tvoří  $IO$  množinu

$$\mathcal{N}_{IO}^{(BH)}[k, l] = d[k, l] \cup \{c[k, l, \ell]\}_{\ell < 3}. \quad (40)$$

Vzniklé horizontální konvoluční regiony

$$\mathcal{N}^{(H)}[k, l] = \{c[k, l + \{\mathcal{I}_\ell^{(H)}\}_{\ell < 3}]\}_{\ell < 3}, \quad (41)$$

tvoří vstupy kombinačních elementů v horní úrovni spodní podsítě zapojení. Zde množina indexů



$$\mathcal{I}_l^{(H)} = \{-1, +1, 0\} \quad (42)$$

definuje propojení mezi broadcastery a kombinačními elementy ve spodní podsíti. Porty kombinačních elementů sdružuje  $IO$  množina

$$\mathcal{N}_{IO}^{(H)}[k, l] = \{c[k, l + \{\mathcal{I}_l^{(H)}\}_{\ell, \ell}]\}_{\ell < 3} \cup c[k, l], \quad (43)$$

kde  $c[k, l]$  jsou mezi-vrstvové symboly na výstupech spodní podsítě. V okamžiku, kdy jsou všechny mezi-vrstvové symboly vypočtené přechází proces kódování do vrchní vrstvy kódovací sítě. Vypočtené symboly  $c[k, l]$  zde působí na vstupech broadcasteřů vrchní podsítě jejichž porty sdružuje  $IO$  množina

$$\mathcal{N}_{IO}^{(BV)}[k, l] = c[k, l] \cup \{q[k, l, \ell]\}_{\ell < 3}. \quad (44)$$

Broadcastery kopírují mezi-vrstvové symboly, prostřednictvím stavových veličin  $q[k, l, \ell]$ , do nejbližších uzlů nad nimi a pod nimi v jednotlivých vertikálách. Vzniknou tak vertikální konvoluční regiony

$$\mathcal{N}^{(V)}[k, l] = \{q[k + \{\mathcal{I}_k^{(V)}\}_{\ell, \ell}]\}_{\ell < 3}, \quad (45)$$

kde množina indexů

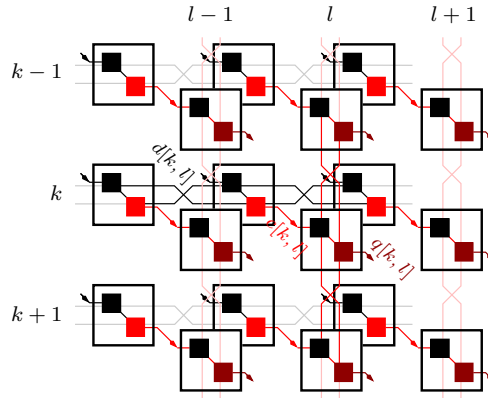
$$\mathcal{I}_k^{(V)} = \{+1, -1, 0\} \quad (46)$$

definuje propojení mezi broadcastery a kombinačními elementy ve vrchní podsíti. Přípravené vertikální konvoluční regiony vstupují do kombinačních elementů vrchní podsítě. Porty těchto elementů definuje  $IO$  množina

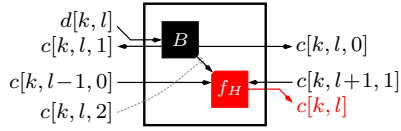
$$\mathcal{N}_{IO}^{(V)}[k, l] = \mathcal{N}[k, l] \cup q[k, l] \quad (47)$$

a jejich výstup tvoří finální kódové symboly  $q[k, l]$ . Uvedený matematický aparát koresponduje se zapojením na Obr. 9a, kde jednotlivé uzly na spodní resp. vrchní vrstvě tvoří kompozitní funkční blok na Obr. 9b resp. Obr. 9c.

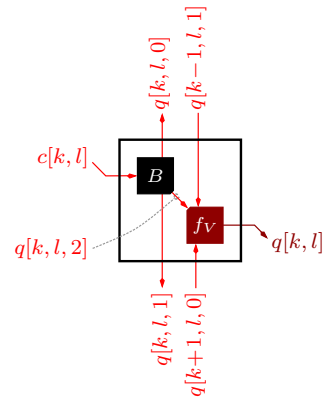
Z výše uvedeného je zřejmé, že aplikovatelnost a flexibilita vrstevové topologie je dosti omezená a ze strukturálního hlediska je poměrně složitá. Přesto má určité významné přednosti, které budou zevrubně popsány v následujícím textu a které, stručně řečeno, zajišťují redukci složitosti detekčních buněk korespondující iterativní dekodovací sítě.



(a) Topologie vrstvé sítě sestavené ze dvou podsítí s variabilní topologií.



(b) Buňka sítě v uzlu  $[k, l]$  spodní vrstvy.



(c) Buňka sítě v uzlu  $[k, l]$  vrchní vrstvy.

Obr. 9. Kódovací síť s variabilní vrstvou topologií pro obecný dekomponovatelný ISI kernel  $\mathcal{G}_{3 \times 3}^{(HV)}$ .

## 4 2D MAP detekce

V této kapitole se přesuneme na stranu detekce a budeme se zabývat teoretickými aspekty MAP detektoru a výchozími matematickými formulacemi. Dvoudimenzionální MAP detekce je z hlediska principu stejná jako ta klasická jednodimenzionální. Další dimenze navíc ovšem celou problematiku značně rozšiřuje na celou škálu rozmanitých řešení a různých uspořádání takového zobecněného iterativního detektoru. Rozsáhlé teoretické pozadí k oblasti klasické MAP detekce je uvedeno v [12, 14, 13, 15], včetně odvození MAP kritéria. Zde vyjdeme z dvoudimenzionální formy MAP kritéria [12] a budeme se zabývat otázkou jak lze kritérium “dekomponovat” a tak dospět od tzv. jednostupňové (optimální) MAP detekce k tzv. iterativní (suboptimální) MAP detekci, která je prakticky realizovatelná. Bude ukázáno, že tato dekompozice nebo také faktorizace výchozí formule MAP kritéria je možná, pokud platí podmínka (10).

### 4.1 Jednostupňová (optimální) MAP detekce

Optimální 2D MAP detektor je založen na kritériu

$$\hat{d}[k, l] = \arg \underset{\hat{d}[k, l]}{\mathbb{M}} \left( \underset{\check{\mathbf{D}}: \hat{d}[k, l]}{\mathbb{M}} \mathcal{S}(\mathbf{R}, \check{\mathbf{D}}) \right), \quad (48)$$

kde  $\hat{d}[k, l]$  je výstupní odhad dat v příslušném místě,  $\check{d}[k, l]$  značí testovací estimátor v příslušném místě (hypoteticky možná hodnota dat),  $\check{\mathbf{D}} : \check{d}[k, l]$  značí množinu všech možných realizací dat, která obsahují v daném místě hodnotu  $\check{d}[k, l]$  a  $\mathbb{M}$  společně s  $\mathbb{M}$  představují dvojici obecných marginalizačních operátorů. Proces marginalizace (postupné vylučování) probíhá nad tzv. sdruženými měkkými metrikami detektoru  $\mathcal{S}(\mathbf{R}, \check{\mathbf{D}})$ . Tyto metriky nám kvantifikují míru věrohodnosti tvrzení, že  $\check{\mathbf{D}} = \mathbf{D}$  pokud jsme přijali realizaci  $\mathbf{R}$ . Je zřejmé, že v této podobě není kritérium implementovatelné, protože množství metrik  $\mathcal{S}(\mathbf{R}, \check{\mathbf{D}})$  roste exponenciálně s velikostí  $\check{\mathbf{D}}$  a základem tohoto růstu je mohutnost abecedy dat. Abychom mohli detektor realizovat je nezbytně nutné kritérium nejprve faktorizovat na menší části. Tato operace je možná díky předpokladu nekorelovanosti hodnot šumu  $w[k, l]$  v matici  $\mathbf{W}$ . Platí, že  $p_{\mathbf{W}}(\Xi) = \prod_{k,l} p_w(\xi)$  a proto můžeme metriku detektoru dekomponovat do tvaru

$$\begin{aligned} \mathcal{S}(\mathbf{R}, \check{\mathbf{D}}) &= \mathcal{S}_F(\mathbf{R}|\check{\mathbf{D}}) \odot \mathcal{S}_B(\check{\mathbf{D}}) \\ &= \left( \underset{k,l}{\odot} \mathcal{S}_F(r[k, l]|\check{\mathcal{N}}[k, l]) \right) \odot \left( \underset{k,l}{\odot} \mathcal{S}_B(\check{d}[k, l]) \right), \end{aligned} \quad (49)$$

neboli na úroveň jednotlivých, vzájemně překrytých, konvolučních regionů  $\check{\mathcal{N}}[k, l]$ . Zde

$$\begin{aligned} \mathcal{S}_F(\mathbf{R}|\check{\mathbf{D}}) &= \underset{k,l}{\odot} \mathcal{S}_F(r[k, l]|\check{\mathbf{D}}) \\ &= \underset{k,l}{\odot} \mathcal{S}_F(r[k, l]|\check{\mathcal{N}}[k, l]) \end{aligned} \quad (50)$$

jsou aposteriorní metriky a

$$\mathcal{S}_B(\check{\mathbf{D}}) = \underset{k,l}{\odot} \mathcal{S}_B(\check{d}[k, l]). \quad (51)$$

apriorní metriky dat. Data předpokládáme jako nekorelovaná v prostoru  $[k, l]$ . Pokud jsou data i statisticky uniformně rozdělená, lze člen  $\mathcal{S}_B(\check{\mathbf{D}})$  ze vztahu (49) vyřadit a MAP kritérium se nám změní na ML (Maximum likelihood) kritérium. Všechny faktorizované metriky ve vztahu (49) jsou kombinovány do sdružené metriky detektoru pomocí obecného kombinačního operátoru  $\odot$ . Kombinační i marginalizační operátory dělíme podle domény implementace detektoru a také podle detekční techniky. Tato technika může být symbolová (SyD) nebo sekvenční (PgD). Symbolová technika se snaží minimalizovat chybu detekce daného symbolu  $d[k, l]$ , bez ohledu na ostatní data v matici  $\mathbf{D}$ . Cílem sekvenční nebo také stránkové techniky je minimalizace chyby detekce veškerých dat  $\mathbf{D}$  jako celku. Přehled operátorů uvádí Tab. 1, kde  $\mathcal{M}(\cdot) = -\ln \mathcal{P}(\cdot)$  a  $\min^*(x, y) = \min(x, y) - \ln(1 + e^{-|x-y|})$ . Implementačně nejvýhodnější je varianta Md-PgD, která má, z hlediska výpočetní náročnosti, nejjednodušší operátory.

## 4.2 Iterativní (suboptimální) MAP detekce

Bylo řečeno, že přímý výpočet odhadu  $\hat{\mathbf{D}}$  z MAP kriteria (48) není prakticky realizovatelný, protože vyžaduje postupné dosažení všech potenciálních realizací dat  $\hat{\mathbf{D}}$ . Díky náhodnému kanálu typu ML-IECS však můžeme MAP kritérium faktorizovat (49) až na úroveň jednotlivých, vzájemně překrytých, konvolučních regionů  $\mathcal{N}[k, l]$ . Tím se dostáváme k samotné iterativní detekci, kde redukuje se složitost jednoduššího MAP kriteria (48) z velikosti vstupních dat  $\mathbf{D}$  na velikost kernelu  $\mathcal{G}$ . Iterativní detektor (dekódovací síť) představuje systém vzájemně propojených kombinačně-marginalizačních elementů (SISO modulů) v topologiích, které přesně odpovídají výchozí struktuře v podobě kódovací sítě. Každý element, nebo skupina elementů, této dekodovací sítě se snaží dekodovat příslušný konvoluční region. Minimální počet elementů sítě je tedy totožný s rozměrem dat  $\mathbf{D}$ . Kombinačně-marginalizační elementy obecně znázorňuje Obr. 2b a představují měkkou statistickou inverzi obecných GPE elementů na Obr. 2a, vystupujících v použitém celulárním modelu ISI kanálu. Každý port SISO modulu je obousměrný a nepřenáší pouze jednu hodnotu příslušné proměnné, jak tomu bylo v modelu ISI kanálu, ale celou diskretní hustotu měkkých metrik (hustotu pravděpodobnosti) kvantifikující věrohodnosti všech možných hodnot této proměnné. V průběhu detekčního procesu každý SISO modul nejprve převezme vstupní metriky  $\mathcal{S}_I(\check{\mathcal{V}}_{I|O}[\ell]) \in \{\mathcal{S}_I(\mathcal{V}_{I|O}^{(i)}[\ell])\}_{i \in \mathcal{L}(\mathcal{V}_{I|O}[\ell])}$  od svých sousedů, kde  $\mathcal{V}_{I|O}[\ell]$  značí obecnou proměnnou na portu buněk výchozí struktury kódovací sítě. Vstupní metriky modul zkombinuje do dílčích sdružených a posteriori metrik  $\mathcal{S}(\check{\mathcal{N}}_{IO})$ , které udávají věrohodnosti všech možných uspořádání (realizací) celého příslušného konvolučního regionu  $\check{\mathcal{N}}$ . Následně modul marginalizuje množinu sdružených metrik  $\{\mathcal{S}(\mathcal{N}_{IO}^{(i)})\}_i$  do výstupních metrik  $\mathcal{S}_O(\check{\mathcal{V}}_{I|O}[\ell]) \in \{\mathcal{S}_O(\mathcal{V}_{I|O}^{(i)}[\ell])\}_{i \in \mathcal{L}(\mathcal{V}_{I|O}[\ell])}$ . Tuto operaci označujeme jako aktivace SISO modulu. Aktivace jednotlivých modulů se děje podle tzv. aktivačního schématu. Tento rozvrh musí mít svoji logiku, která by měla být v souladu s tokem informace stavových a pomocných proměnných uvnitř modelu ISI kanálu. Pokud je aktivační schéma zvoleno nevhodně (proti směru toku stavových proměnných), tak se doba konvergence systému ke správnému řešení, udávaná v počtu iterací  $N_I$ , prodlužuje. Pod pojmem iterace systému budeme tedy chápat časový interval, kdy proběhne celé aktivační schéma a každý z modulů je alespoň jednou aktivován. Jednotlivé iterace uzavírá výměna metrik

$$\begin{aligned} \mathcal{S}_O(\check{\mathcal{V}}_{I|O}[\ell]) &\rightarrow \mathcal{S}_I(\check{\mathcal{V}}_{I|O}[\ell]) \\ \mathcal{S}_I(\check{\mathcal{V}}_{I|O}[\ell]) &\leftarrow \mathcal{S}_O(\check{\mathcal{V}}_{I|O}[\ell]) \end{aligned} \quad (52)$$

mezi moduly.

Na počátku detekčního procesu nastavíme vstupní metriky  $\mathcal{S}_I$  v souladu s našimi apriorními znalostmi o proměnných, které tyto metriky reprezentují. Pokud žádné apriorní znalosti nemáme, nebo zdroj generuje statisticky uniformně rozdělená data, nastavíme tyto metriky do uniformního tvaru. Za takové situace bude odpovídat každé možné hodnotě testovacího estimátoru dané proměnné tatáž metrika. Na počátku detekčního procesu tak nebudou uniformní pouze dopředné (vstupní) metriky  $\mathcal{S}_F$  z front-endu iterativního detektoru.

Iterativní detekce je detekcí suboptimální, protože probíhá iterativně a odhad je možno označit za optimální po proběhnutí nekonečně velkého počtu iterací. Ve většině případů je konvergence sítě ke správnému odhadu však velmi rychlá, řádově v jednotkách iterací  $I$ . Systém dospěje do stavu, kdy se vyměňované hustoty mezi SISO moduly ustálí a s dalšími iteracemi již nemění. V tomto stacionárním stavu “uzamčení” metrik ve zpětných vazbách (iteračních smyčkách) lze proces detekce ukončit a provést závěrečné tvrdé rozhodnutí v rozhodovacích blocích (DEC)

$$\hat{\mathcal{V}}_{I|O}[\ell] = \arg \underset{\check{\mathcal{V}}_{I|O}[\ell]}{\text{M}} (\mathcal{S}_I(\check{\mathcal{V}}_{I|O}[\ell]) \odot \mathcal{S}_O(\check{\mathcal{V}}_{I|O}[\ell])) \quad (53)$$

kombinací vstupních a výstupních metrik na příslušném portu SISO modulu. Pokud v průběhu detekce dochází ke kmitavému charakteru, kdy se metriky nedostávají do stavu uzamčení, ukončíme proces detekce po proběhnutí maximálního zvoleného počtu iterací  $N_I$ .

## 5 Iterativní dekódovací sítě

V kapitole 4 byla stručně popsána podstata iterativní detekce, celulární struktura dvoudimenzionálního iterativního MAP detektoru a funkční bloky, které tuto strukturu tvoří. Také byl přiblížen pojem aktivace SISO modulu. Tento mechanismus podrobíme zevrubnému popisu z obecného hlediska, protože princip činnosti modulů v partikulárních topologiích je analogický. Dále se budeme zabývat konkrétními variantami uspořádání SISO modulů v jednotlivých výchozích topologiích z kapitoly 3 a popisem jejich portů.

### 5.1 Elementy iterativních dekódovacích sítí — SISO moduly

V předcházejícím textu bylo naznačeno, že aktivace SISO modulu se skládá ze dvou po sobě jdoucích kroků. Kombinace a marginalizace. Pod pojmem kombinace budeme rozumět sloučení vstupních (apriorních nebo aposteriorních) měkkých metrik  $\{\mathcal{S}_I(\check{\mathcal{V}}_{IO}[\ell])\}_\ell$  jednotlivých vstupních nebo výstupních veličin  $\mathcal{V}_{IO}[\ell]$  do sdružené (aposteriorní) metriky  $\mathcal{S}(\check{\mathcal{N}}_{IO})$  a to pomocí příslušného kombinačního operátoru  $\odot$ . Proces kombinování můžeme matematicky formulovat pomocí vztahu

$$\mathcal{S}(\check{\mathcal{N}}_{IO}) = \bigodot_{\check{\mathcal{V}}_{IO}[\ell] \in \check{\mathcal{N}}_{IO}} \mathcal{S}_I(\check{\mathcal{V}}_{IO}[\ell]), \quad (54)$$

kde podmínka  $\check{\mathcal{V}}_{IO}[\ell] \in \check{\mathcal{N}}_{IO}$  vyplývá z tabulky kombinací SISO modulu. Aposteriorní metriky  $\mathcal{S}(\check{\mathcal{N}}_{IO}) \in \{\mathcal{S}(\mathcal{N}_{IO}^{(i)})\}_i$  reflektují věrohodnosti jednotlivých realizací množiny vstupů a výstupů  $\mathcal{N}_{IO}^{(i)}$  a tedy i konvolučního regionu  $\mathcal{N}^{(i)}$ . Mezi oběma množinami totiž panuje jednoznačná relace. Abecedu realizací množiny  $\mathcal{N}_{IO}$  označme jako  $\mathcal{A}_{IO} = \{\mathcal{N}_{IO}^{(i)}\}_i$  a například pro ISI kernel  $3 \times 3$  s binárním vstupem bude její mohutnost  $\text{card}(\mathcal{A}_{IO}) = 512$ . Na výstupu kombinační části SISO modulu tak získáme, v tomto konkrétním příkladě, sadu celkem 512 sdružených metrik  $\mathcal{S}(\check{\mathcal{N}}_{IO})$ .

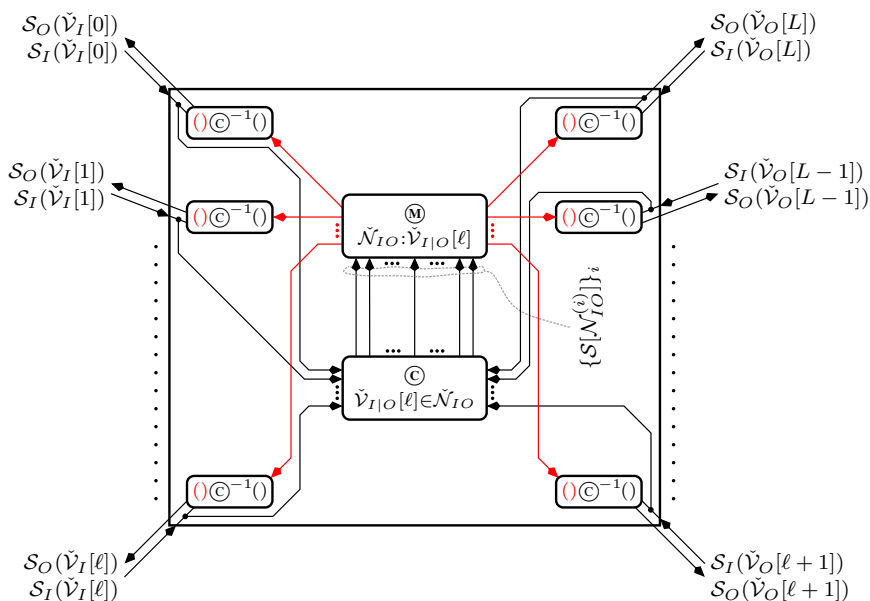
Na proces kombinování navazuje proces marginalizace, nebo také vylučování, kdy jsou sdružené metriky  $\{\mathcal{S}(\mathcal{N}_{IO}^{(i)})\}_i$  marginalizovány do výstupních (aposteriorních) metrik  $\{\mathcal{S}_O(\check{\mathcal{V}}_{IO}[\ell])\}_\ell$  prostřednictvím marginalizačního operátoru  $\otimes$ . Druhý krok aktivace je definován vztahem

$$\mathcal{S}_O(\check{\mathcal{V}}_{IO}[\ell]) = \left( \bigotimes_{\check{\mathcal{N}}_{IO} \in \check{\mathcal{N}}_{IO}[\ell]} \mathcal{S}(\check{\mathcal{N}}_{IO}) \right) \odot^{-1} \mathcal{S}_I(\check{\mathcal{V}}_{IO}[\ell]). \quad (55)$$

kde  $\check{\mathcal{N}}_{IO} : \check{\mathcal{V}}_{IO}[\ell] \subset \mathcal{A}_{IO}$  označuje množinu všech realizací množin vstupů a výstupů, které obsahují testovací estimátor  $\check{\mathcal{V}}_{IO}[\ell]$ . SISO modul následně odešle vypočtené výstupní metriky do sousedících modulů, se kterými je propojen. Celý výše popsaný mechanismus aktivace schématicky znázorňuje Obr. 10, kde je zakreslena vnitřní struktura obecné statistické měkké inverze.

Doména	$\mathcal{S}(\cdot)$	Detekce	$\otimes$	$\oplus$	$\odot$	$\oplus$	$\odot^{-1}$
Pravděpodobnostní (Pd)	$\mathcal{P}(\cdot)$	Sekvenční - stránková (PgD)	max	max	$\Pi$	$\times$	$\div$
Pravděpodobnostní (Pd)	$\mathcal{P}(\cdot)$	Symbolová (SyD)	max	$\Sigma$	$\Pi$	$\times$	$\div$
Metrická (Md)	$\mathcal{M}(\cdot)$	Sekvenční - stránková (PgD)	min	min	$\Sigma$	+	−
Metrická (Md)	$\mathcal{M}(\cdot)$	Symbol (SyD)	min	min*	$\Sigma$	+	−

Tab. 1. Přehled kombinačních a marginalizačních operátorů.



Obr. 10. Obecná vnitřní struktura Soft-In Soft-Out modulu.

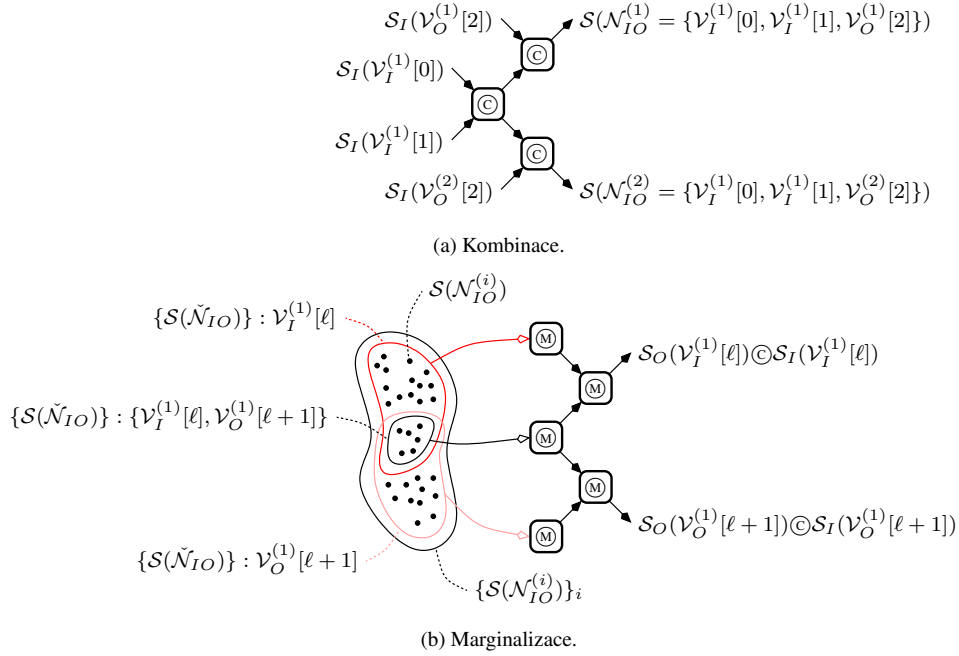
## 5.2 Implementační aspekty

Ačkoli chování SISO modulu popisuje dvojice jednoduchých vztahů (54) a (55) jedná se ve skutečnosti o numericky velmi náročný proces. Kombinační tabulka, definující strukturu SISO modulu, může obsahovat obrovské množství různých relací a proto je vhodné provádět výpočet vztahů (54) a (55) maximálně efektivně. Dále je zapotřebí hustoty pravděpodobností či metrik na výstupu SISO modulu normovat a aritmeticky ošetřit tak, aby nedocházelo k jejich “přetékání” či “podtékání” v průběhu procesu iterativní detekce. Oběma těmito problémům se budeme věnovat v této kapitole a správnou implementaci SISO modulu budeme demonstrovat na jednoduchém příkladu.

### 5.2.1 Kombinace a marginalizace se stromovou strukturou

Kombinování i marginalizaci lze efektivně implementovat pomocí stromové struktury. Tento princip je vcelku prostý a je přiblížen na Obr. 11. Oba procesy zpracování jsou zde rozděleny do řetězce následných sekcí, neboli určité struktury s “průtokovým” zpracováním, kdy se maximálně využívá pomocných mezivýsledků z předcházejícího úseku. Ukázka na Obr. 11a znázorňuje část kombinační sekce SISO modulu se třemi porty. Zde vstupní hustoty reprezentují tři binární náhodné veličiny  $\mathcal{V}_I[1]$ ,  $\mathcal{V}_I[2]$  a  $\mathcal{V}_I[3]$ . Celkový počet možných kombinací  $\mathcal{S}(\tilde{\mathcal{N}}_{IO})$  je v tomto případě 8 s tím, že uvedený výňatek na Obr. 11a určuje dvě z nich a zbylých šest bychom získali analogickým způsobem. Ukázka na Obr. 11b se týká marginalizačního procesu. Znárodněna je opět pouze část struktury o celkovém počtu dvou následných sekcí. Předkládáme, že množina všech aposteriorních sdružených metrik  $\{\mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(i)})\}_i$  je zde rozdělena na tři části. První část  $\{\mathcal{S}(\tilde{\mathcal{N}}_{IO})\} : \mathcal{V}_I^{(1)}[\ell]$ , lemovaná červenou čarou, necht’ obsahuje ty metriky  $\mathcal{S}(\tilde{\mathcal{N}}_{IO})$ , které zahrnují realizaci  $\mathcal{V}_I^{(1)}[\ell]$  vstupní náhodné veličiny  $\mathcal{V}_I[\ell]$ . Druhá část  $\{\mathcal{S}(\tilde{\mathcal{N}}_{IO})\} : \mathcal{V}_O^{(1)}[\ell+1]$ , lemovaná světle červenou čarou, necht’ obsahuje ty metriky  $\mathcal{S}(\tilde{\mathcal{N}}_{IO})$ , které zahrnují realizaci  $\mathcal{V}_O^{(1)}[\ell+1]$  výstupní náhodné veličiny  $\mathcal{V}_O[\ell+1]$ . Třetí část  $\{\mathcal{S}(\tilde{\mathcal{N}}_{IO})\} : \{\mathcal{V}_I^{(1)}[\ell], \mathcal{V}_O^{(1)}[\ell+1]\}$ , lemovaná černou čarou, pak představuje průnik obou množin a obsahuje ty metriky  $\mathcal{S}(\tilde{\mathcal{N}}_{IO})$ , které zahrnují jak realizaci  $\mathcal{V}_I^{(1)}[\ell]$  vstupní náhodné veličiny  $\mathcal{V}_I[\ell]$  tak realizaci  $\mathcal{V}_O^{(1)}[\ell+1]$  výstupní náhodné veličiny  $\mathcal{V}_O[\ell+1]$ . Stromovou marginalizací takto definovaných množin na Obr. 11b pak získáme dvě konkrétní metriky  $\mathcal{S}(\mathcal{V}_I^{(1)}[\ell]) = \mathcal{S}_O(\mathcal{V}_I^{(1)}[\ell]) \circledast \mathcal{S}_I(\mathcal{V}_I^{(1)}[\ell])$  a  $\mathcal{S}(\mathcal{V}_O^{(1)}[\ell+1]) = \mathcal{S}_O(\mathcal{V}_O^{(1)}[\ell+1]) \circledast \mathcal{S}_I(\mathcal{V}_O^{(1)}[\ell+1])$  ze sdružených

hustot  $\{\mathcal{S}(\mathcal{V}_I^{(i)}[\ell])\}_i$  a  $\{\mathcal{S}(\mathcal{V}_O^{(i)}[\ell + 1])\}_i$  a tyto metriky určují zpřesněnou (výstupní) věrohodnost hypotéz, že náhodné proměnné  $\mathcal{V}_I[\ell]$  a  $\mathcal{V}_O[\ell + 1]$  jsou realizovány hodnotami  $\mathcal{V}_I^{(1)}[\ell]$  a  $\mathcal{V}_O^{(1)}[\ell + 1]$ . Analogickým postupem bychom získali ostatní hodnoty  $\mathcal{S}(\check{\mathcal{V}}_I[\ell])$  a  $\mathcal{S}(\check{\mathcal{V}}_O[\ell + 1])$  ze sdružených výstupních hustot  $\{\mathcal{S}(\mathcal{V}_I^{(i)}[\ell])\}_i$  a  $\{\mathcal{S}(\mathcal{V}_O^{(i)}[\ell + 1])\}_i$ , potažmo i sdružené výstupní hustoty dalších proměnných  $\mathcal{V}_{I|O}$  na ostatních portech SISO modulu.



Obr. 11. Princip kombinace a marginalizace se stromovou strukturou.

Uvedená fakta více přiblížíme a ozřejmíme na konkrétním příkladu zapojení SISO modulu, který se ještě dále vyskytne v jedné z předkládaných realizací iterativního detektoru v následujícím textu. Předpokládejme jednoduchý jednodimenzionální kernel  $\mathcal{G}^{(H)} = \{ g' \ g \ g' \}$  na straně vysílání. Výstup takového ISI kanálu označme

$$q[k, l] = g'c[k, l - 1, 0] + gc[k, l, 2] + g'c[k, l + 1, 1], \quad (56)$$

kde  $c[k, l - 1, 0]$ ,  $c[k, l, 1]$  a  $c[k, l + 1, 2]$  jsou binární vstupy. Uvedené proměnné definují testovací estimátor  $IO$  množiny SISO modulu

$$\check{\mathcal{N}}_{IO} = \{ \check{c}[0], \check{c}[2], \check{c}[1], \check{q} \}. \quad (57)$$

Hodnoty testovacích estimátorů v estimátoru  $IO$  množiny jsou vzájemně svázané kombinacemi uvedenými v Tab. 2. Každý z osmi řádků tabulky představuje jednu realizaci  $\mathcal{N}_{IO}^{(i)}$  množiny  $\check{\mathcal{N}}_{IO}$ . Příslušné kombinace vstupů a výstupů ISI kanálu jsou zde vyznačeny prostřednictvím mapovacích ukazatelů  $i_{c[\ell]}$  a  $i_q$  do abeced  $\mathcal{A}_c$  a  $\mathcal{A}_q$  jednotlivých proměnných. Vstup kombinační sekce SISO modulu tvoří tedy čtyři hustoty. Tři dvouprvkové  $\{\mathcal{S}_I(c^{(i_{c[\ell]})})\}_{i_{c[\ell]}}$ , obsahující vstupní metriky  $\mathcal{S}_I(\check{c}[\ell])[k, l + \{\mathcal{I}_l\}_\ell, \ell]$  binárních vstupů  $c$ , a jedna šesti-prvková  $\{\mathcal{S}_{IO}(q^{(i_q)})\}_{i_q}$ , obsahující vstupní metriky  $\mathcal{S}_I(\check{q})[k, l]$  hexálního výstupu  $q$ . Čísla  $c^{(i)}$  resp.  $q^{(i)}$  zde představují jednotlivé prvky  $\{\mathcal{A}_c\}_i$  resp.  $\{\mathcal{A}_q\}_i$  abeced  $\mathcal{A}_c$  resp.  $\mathcal{A}_q$  o mohutnostech  $\text{card}(\mathcal{A}_c) = 2$  resp.  $\text{card}(\mathcal{A}_q) = 6$  a množina  $\mathcal{I}_l = \{-1, +1, 0\}$  definuje propojení sousedních SISO modulů ve struktuře jednoho z iterativních detektorů z následující kapitoly. Zkombinováním vstupních metrik získáme osm sdružených metrik  $\mathcal{S}(\check{\mathcal{N}}_{IO})$ .

Následuje marginalizace těchto hodnot do čtyř výstupních hustot. Tři dvouprvkových  $\{\mathcal{S}_O(c^{(i_{c[\ell]})})\}_{i_{c[\ell]}}$ , obsahující výstupní metriky  $\mathcal{S}_O(\tilde{c}[\ell])[k, l + \{\mathcal{L}_l\}_\ell, \ell]$  binárních vstupů  $c$ , a jedné šesti-prvkové  $\{\mathcal{S}_{IO}(q^{(i_q)})\}_{i_q}$ , obsahující výstupní metriky  $\mathcal{S}_O(\tilde{q})[k, l]$  hexálního výstupu  $q$ . Vnitřní zapojení SISO modulu, provádějící měkkou inverzi konvoluční operace (56) a využívající kombinování a marginalizace se stromovou strukturou znázorňuje Obr. 12.

No.	$i_{c[0]}$	$i_{c[2]}$	$i_{c[1]}$	$i_q$	
1	0	0	0	0	
2	1	0	0	1	
3	0	1	0	3	
4	1	1	0	4	
5	0	0	1	1	
6	1	0	1	2	
7	0	1	1	4	
8	1	1	1	5	

$i$	0	1	2	3	4	5
$\{\mathcal{A}_c\}_i$	0	1				
$\{\mathcal{A}_q\}_i$	0	$g'$	$2g'$	$g$	$g + g'$	$g + 2g'$

Tab. 2. Tabulka kombinací IO (vstupů a výstupů) definující strukturu jednoduchého ukázkového SISO modulu a abecedy IO proměnných.

Z uvedeného příkladu je patrné, že GPE modelující poměrně jednoduchý jednodimenzionální kernel (56) má dosti složitou měkkou inverzi, přestože je realizovaná optimálně. V případě větších kernelů struktura značně narůstá a optimální implementace je obtížně určitelná. Obzvlášť, pokud nejsou veličiny na jednotlivých vstupních portech GPE vzájemně zcela disjunktní.

### 5.2.2 Ošetření aritmetiky

Kromě samotných výpočtů kombinace a marginalizace musí SISO modul provádět ještě normování metrik. Bez tohoto opatření může dojít k posunu veličin mimo požadovaný dynamický rozsah. Z tohoto hlediska je citlivý především detektor realizovaný v pravděpodobnostní doméně. Veličiny je zapotřebí držet v intervalu  $\langle 0, 1 \rangle$  pomocí normování

$$\mathcal{P}_O(\mathcal{V}^{(i)}) = \frac{\mathcal{P}_O(\mathcal{V}^{(i)})}{\sum_i \mathcal{P}_O(\mathcal{V}^{(i)})}. \quad (58)$$

Také je vhodné zavést prahování pravděpodobností, aby nedocházelo k poklesu metrik pod určitou minimální mez. Toto opatření je nezbytné díky dělení  $\odot^{-1}$ , které může způsobit přetečení.

Detektor realizovaný v metrické doméně se z aritmetického hlediska chová mnohem rozumněji. Metriky mohou nabývat kladných i záporných hodnot a normují se pomocí vztahu

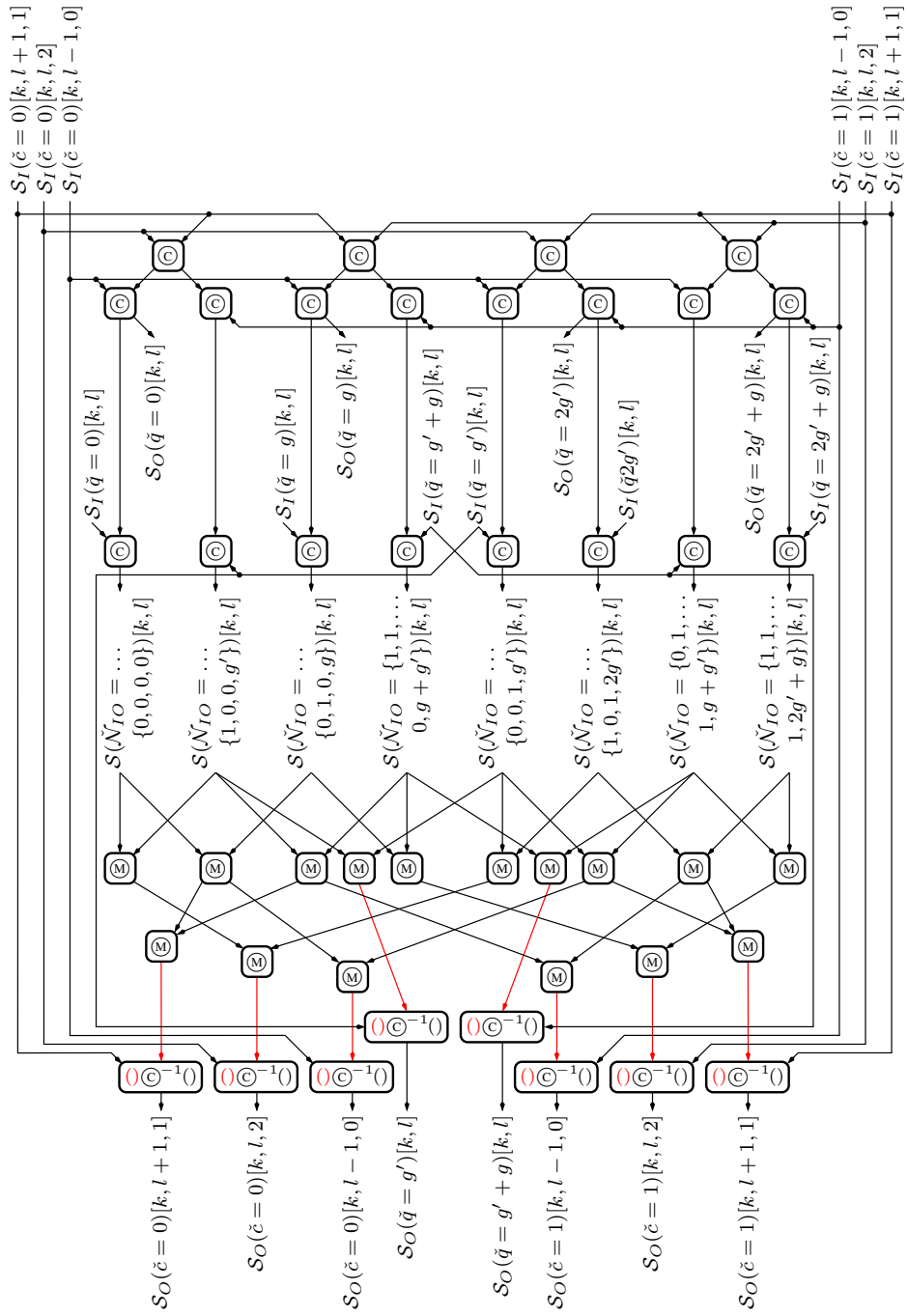
$$\mathcal{M}_O(\mathcal{V}^{(i)}) = \mathcal{M}_O(\mathcal{V}^{(i)}) - \min_i \mathcal{M}_O(\mathcal{V}^{(i)}) \quad (59)$$

nebo

$$\mathcal{M}_O(\mathcal{V}^{(i)}) = \mathcal{M}_O(\mathcal{V}^{(i)}) - \mathcal{M}_O(\mathcal{V}^{(1)}). \quad (60)$$

První druh normování drží metriky v intervalu  $\langle 0, \infty \rangle$ . Druhý připouští i záporné hodnoty metrik z intervalu  $(-\infty, \infty)$  s tím, že první metrika v sadě je vždy nulová a proto ji nemusíme skladovat a poklesnou paměťové nároky. Pro skutečnou implementaci detektoru je bezpochyby výhodnější metrická doména a to v kombinaci s detekční technikou PgD. Přetečení nebo podtečení aritmetiky zde prakticky nehrozí. Jako zcela dostatečné je kvantování metrik pomocí 6ti až 12ti bitových čísel v pevné řádové čárce se znaménkem (60) nebo bez znaménka (59).





Obr. 12. Vnitřní struktura jednoduchého ukázkového SISO modulu založená na kombinování a marginalizaci se stromovou strukturou.

### 5.3 Topologie iterativních dekódovacích sítí

V předcházejícím textu byly sumarizovány principy MAP detekce a podmínky vytvoření dvoudimenzionálního iterativního detektoru faktorizací výchozího kritéria na úroveň konvolučních regionů. Dále jsme podrobili popisu SISO moduly, které tvoří stavební prvky iterativní dekódovací sítě. Implementace SISO modulu byla demonstrována na jednoduchém příkladě. Nyní přejdeme k problematice topologie iterativní dekódovací sítě, neboli vzájemného propojení SISO modulů uvnitř této struktury. Z tohoto hlediska dělíme topologie IDN do tří základních skupin. Na IDN marginalizující na úrovni bloku symbolů, IDN marginalizující na úrovni symbolů a vrstevné IDN marginalizující na úrovni bloku symbolů nebo na úrovni symbolů. Každá topologie má své specifické uspořádání a vnitřní implementaci SISO modulu nebo dvojice SISO modulů, tvořících příslušnou kompozitní buňku ve struktuře detektoru. Samotné propojení buněk v jednotlivých uzlech zapojení je analogické k topologiím kódovací sítě z kapitoly 3.2 s tím rozdílem, že každý GPE je zaměněn za odpovídající SISO modul a ve většině případů dochází k přenosu měkké informace v obou směrech. Jednotlivé topologie budou popsány v následujících odstavcích 5.3.1, 5.3.2, 5.3.3 a 5.3.4.

#### 5.3.1 Pevné topologie

Sítě s fixní topologií též označujeme jako sítě marginalizující na úrovni bloků symbolů. Konvoluční region je v referenčním modelu separován do bloků. Základní zřetězení modulů je v horizontálním a vertikálním směru [12]. Takovou iterativní dekódovací síť s pevnou topologií znázorňuje Obr. 13a. Topologie sítě není závislá na tvaru kernelu a tvoří ji SISO moduly na Obr. 13b. Vstupní a výstupní metriky stavových veličin budeme označovat jako  $\mathcal{S}_I$  a  $\mathcal{S}_O$ . Vstupní a výstupní metriky dat a výstupu kanálu budeme označovat jako  $\mathcal{S}_F$  a  $\mathcal{S}_B$  respektive  $\mathcal{S}_B$  a  $\mathcal{S}_F$ . Rozdílná notace byla zavedena z toho důvodu, že sítě mohou být i vrstvené. Metriky  $\mathcal{S}_I$  a  $\mathcal{S}_O$  tak budeme obecně chápat jako spojnice mezi moduly na dané vrstvě sítě a  $\mathcal{S}_F$  a  $\mathcal{S}_B$  respektive  $\mathcal{S}_B$  a  $\mathcal{S}_F$  budou představovat propojení modulů napříč vrstvami.

Aktivační schéma detektoru je vyznačeno na Obr. 13a fialovou čárkovanou čarou. Kromě aktivace “po řádcích” můžeme použít také schéma překlopené přes diagonálu struktury, neboli “po sloupcích”. Další možná varianta je aktivace “zig-zag” z levého horního rohu do pravého spodního rohu. Všechny tyto varianty postupují ve směru toku stavových veličin v referenčním modelu ISI kanálu na Obr. 3.

Testovací estimátor  $IO$  množiny SISO modulu je u základní pevné topologie ve tvaru

$$\tilde{\mathcal{N}}_{IO} = \{\tilde{\mathcal{R}}[0], \check{\mathcal{C}}[0], \check{d}, \tilde{\mathcal{R}}[1], \check{\mathcal{C}}[1], \check{q}\}, \quad (61)$$

kde  $\tilde{\mathcal{R}}[\ell]$  a  $\check{\mathcal{C}}[\ell]$  jsou testovací estimátory pomocných stavových proměnných. V případě ukázkového obecného posunutého kernelu (17) z odstavce 3.2.1, tvoří estimátory

$$\tilde{\mathcal{R}}[\ell] = \left\{ \begin{array}{cc} \check{d}[-2, \ell - 2] & \check{d}[-2, \ell - 1] \\ \check{d}[-1, \ell - 2] & \check{d}[-1, \ell - 1] \\ \check{d}[0, \ell - 2] & \check{d}[0, \ell - 1] \end{array} \right\} \quad (62)$$

a

$$\check{\mathcal{C}}[\ell] = \left\{ \begin{array}{ccc} \check{d}[\ell - 2, -2] & \check{d}[\ell - 2, -1] & \check{d}[\ell - 2, 0] \\ \check{d}[\ell - 1, -2] & \check{d}[\ell - 1, -1] & \check{d}[\ell - 1, 0] \end{array} \right\} \quad (63)$$

šestice estimátorů dat  $\check{d}$ . Aktivace SISO modulu v uzlu  $[k, l]$  se skládá z kombinování

$$\begin{aligned} \mathcal{S}(\tilde{\mathcal{N}}_{IO})[k, l] &= \left( \bigcirc_{\tilde{\mathcal{R}}[\ell] \in \tilde{\mathcal{N}}_{IO}} \mathcal{S}_I(\tilde{\mathcal{R}}[\ell])[k, l + \ell] \right) \bigcirc \left( \bigcirc_{\check{\mathcal{C}}[\ell] \in \tilde{\mathcal{N}}_{IO}} \mathcal{S}_I(\check{\mathcal{C}}[\ell])[k + \ell, l] \right) \\ &\quad \bigcirc \mathcal{S}_B(\check{d} \in \tilde{\mathcal{N}}_{IO})[k, l] \bigcirc \mathcal{S}_F(\check{q} \in \tilde{\mathcal{N}}_{IO})[k, l], \end{aligned} \quad (64)$$

kde  $\ell = 0, 1$ . Získaná sdružená hustota  $\{\mathcal{S}(\mathcal{N}_{IO}^{(i)})\}_i$  je následně marginalizována prostřednictvím pěti marginalizačních procesů



kdy hustota  $\{\mathcal{S}_B(q^{(i)})[k, l]\}_i$  slouží jako měkká podpora různých synchronizátorů iterativní dekodovací sítě, které korigují vstupní signál  $r[k, l]$  před vstupem do front-endu iterativního detektoru. Tato synchronizace není předmětem předkládané práce a proto je smyčka hustot  $\{\mathcal{S}_B(q^{(i)})[k, l]\}_i$  je na Obr. 1 otevřená a výpočet (68) se neprovádí. Žádný synchronizátor není přítomen a my předpokládáme síť perfektně synchronizovanou.

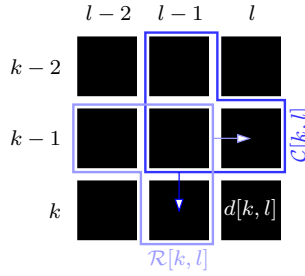
Před první iterací dekodovací sítě jsou metriky

$$\mathcal{S}_I(\check{\mathcal{R}})[k, l] = \bigoplus_{\check{d}[k', l'] \in \check{\mathcal{R}}} \mathcal{S}_B(\check{d}[k', l'])[k + k', l + l'] \quad (69)$$

a

$$\mathcal{S}_I(\check{\mathcal{C}}[k, l]) = \bigoplus_{\check{d}[k', l'] \in \check{\mathcal{C}}} \mathcal{S}_B(\check{d}[k', l'])[k + k', l + l'], \quad (70)$$

nastaveny pomocí kombinace apriorní měkké informace  $\mathcal{S}_B(\check{d}[k', l'])$  jednotlivých estimatorů vstupních dat  $\check{d}[k', l']$ , které spadají do blokových estimatorů  $\check{\mathcal{R}}$  a  $\check{\mathcal{C}}$ . V případě, že jsou vstupní data uniformně rozdělená, pak nutně  $\mathcal{S}_I(\check{\mathcal{R}})[k, l] = 1/\text{card}(\mathcal{A}_{\check{\mathcal{R}}})$  a  $\mathcal{S}_I(\check{\mathcal{C}})[k, l] = 1/\text{card}(\mathcal{A}_{\check{\mathcal{C}}})$  pro všechna  $[k, l]$  pokud je detektor realizován v pravděpodobnostní doméně. Pokud je realizován v metrické doméně nastavíme  $\mathcal{S}_I(\check{\mathcal{R}})[k, l] = 0$  a  $\mathcal{S}_I(\check{\mathcal{C}})[k, l] = 0$ .



Obr. 14. Tvary stavových proměnných  $\mathcal{R}[k, l]$  a  $\mathcal{C}[k, l]$  pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .

Kombinační tabulka SISO modulu z předcházejícího příkladu obsahuje 512 relací, které vedou na tentýž počet sdružených metrik  $\{\mathcal{S}(\mathcal{N}_{IO}^{(i)})\}_i$  a díky své značné délce zde není obsažena. Proto uved' me další příklad uspořádání iterativní dekodovací sítě se základní fixní topologií. Tentokrát se bude jednat o detektor pro účely dekonvoluce jednoduššího hvězdicového ISI kanálu. Předpokládáme ISI kanál s obecným redukovaným posunutým kernelem  $3 \times 3$

$$\mathcal{G}_{3 \times 3}^{(R)} = \begin{Bmatrix} & g' & \\ g' & g & g' \\ & g' & \end{Bmatrix} \quad (71)$$

a binárními vstupními daty. Pomocné stavové proměnné vzorové kódovací sítě na Obr. 3, s přihlédnutím k podmínce kauzality (16), budou mít tvar

$$\mathcal{R}[k, l] = \left\{ \begin{array}{cc} d[k-1, l-2] & d[k-1, l-1] \\ & d[k, l-1] \end{array} \right\} \text{ a } \mathcal{C}[k, l] = \left\{ \begin{array}{cc} d[k-2, l-1] & \\ d[k-1, l-1] & d[k-1, l] \end{array} \right\}, \quad (72)$$

který graficky znázorňuje Obr. 14. Těmto stavovým proměnným odpovídají na straně detekce testovací estimatorů

$$\check{\mathcal{R}}[\ell] = \left\{ \begin{array}{cc} \check{d}[-1, \ell-2] & \check{d}[-1, \ell-1] \\ & \check{d}[0, \ell-1] \end{array} \right\} \text{ a } \check{\mathcal{C}}[\ell] = \left\{ \begin{array}{cc} \check{d}[\ell-2, -1] & \\ \check{d}[\ell-1, -1] & \check{d}[\ell-1, 0] \end{array} \right\} \quad (73)$$

No.	$i_{\mathcal{R}[0]}$	$i_{\mathcal{C}[0]}$	$i_d$	$i_{\mathcal{R}[1]}$	$i_{\mathcal{C}[1]}$	$i_q$	No.	$i_{\mathcal{R}[0]}$	$i_{\mathcal{C}[0]}$	$i_d$	$i_{\mathcal{R}[1]}$	$i_{\mathcal{C}[1]}$	$i_q$
1	0	0	0	0	0	0	33	0	1	0	0	0	1
2	0	0	1	4	4	0	34	0	1	1	4	4	1
3	4	0	0	0	2	1	35	4	1	0	0	2	2
4	4	0	1	4	6	1	36	4	1	1	4	6	2
5	0	4	0	2	0	1	37	0	5	0	2	0	2
6	0	4	1	6	4	1	38	0	5	1	6	4	2
7	4	4	0	2	2	2	39	4	5	0	2	2	3
8	4	4	1	6	6	2	40	4	5	1	6	6	3
9	2	2	0	1	1	5	41	2	3	0	1	1	6
10	2	2	1	5	5	5	42	2	3	1	5	5	6
11	6	2	0	1	3	6	43	6	3	0	1	3	7
12	6	2	1	5	7	6	44	6	3	1	5	7	7
13	2	6	0	3	1	6	45	2	7	0	3	1	7
14	2	6	1	7	5	6	46	2	7	1	7	5	7
15	6	6	0	3	2	7	47	6	7	0	3	2	8
16	6	6	1	7	7	7	48	6	7	1	7	7	8
17	1	0	0	0	0	1	49	1	1	0	0	0	2
18	1	0	1	4	4	1	50	1	1	1	4	4	2
19	5	0	0	0	2	2	51	5	1	0	0	2	3
20	5	0	1	4	6	2	52	5	1	1	4	6	3
21	1	4	0	2	0	2	53	1	5	0	2	0	3
22	1	4	1	6	4	2	54	1	5	1	6	4	3
23	5	4	0	2	2	3	55	5	5	0	2	2	4
24	5	4	1	6	6	3	56	5	5	1	6	6	4
25	3	2	0	1	1	6	57	3	3	0	1	1	7
26	3	2	1	5	5	6	58	3	3	1	5	5	7
27	7	2	0	1	3	7	59	7	3	0	1	3	8
28	7	2	1	5	7	7	60	7	3	1	5	7	8
29	3	6	0	3	1	7	61	3	7	0	3	1	8
30	3	6	1	7	5	7	62	3	7	1	7	5	8
31	7	6	0	3	2	8	63	7	7	0	3	2	9
32	7	6	1	7	7	8	64	7	7	1	7	7	9

Tab. 3. Tabulka kombinací  $IO$  (vstupů, výstupů a stavů) definující strukturu SISO modulů v IDN se základní fixní topologií pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .

složené z trojic estimátorů dat  $\check{d}$ . Hodnoty testovacích estimátorů v estimátoru  $IO$  množiny (61) tvoří jednotlivé kombinační relace v Tab. 3. Celkový počet realizací  $\mathcal{N}_{IO}^{(i)}$  množiny  $\check{\mathcal{N}}_{IO}$  je roven mohutnosti  $\text{card}(\mathcal{A}_{IO}) = 64$  abecedy  $IO$  množiny. Příslušné kombinace vstupu, výstupu a stavů jsou zde definovány prostřednictvím mapovacích ukazatelů  $i_{\mathcal{R}[\ell]}$ ,  $i_{\mathcal{C}[\ell]}$ ,  $i_d$  a  $i_q$  do abeced  $\mathcal{A}_{\mathcal{R}}$ ,  $\mathcal{A}_{\mathcal{C}}$ ,  $\mathcal{A}_d$  a  $\mathcal{A}_q$  o mohutnostech  $\text{card}(\mathcal{A}_{\mathcal{R}}) = \text{card}(\mathcal{A}_{\mathcal{C}}) = 8$ ,  $\text{card}(\mathcal{A}_d) = 2$  a  $\text{card}(\mathcal{A}_q) = 10$ . Všechny abecedy jsou uvedeny v Tab. 4. Partikulární metriky  $\mathcal{S}_{IO}(\check{\mathcal{R}}[\ell])[k, l + \ell]$ ,  $\mathcal{S}_{IO}(\check{\mathcal{C}}[\ell])[k + \ell, l]$ ,  $\mathcal{S}_{FB}(\check{d})[k, l]$  a  $\mathcal{S}_{FB}(\check{q})[k, l]$  tvoří na portech SISO modulu vstupní resp. výstupní hustoty

$\{\mathcal{S}_{I|O}(\mathcal{R}^{(i_{\mathcal{R}[l]})})\}_{i_{\mathcal{R}[l]}}$  a  $\{\mathcal{S}_{I|O}(\mathcal{C}^{(i_{\mathcal{C}[l]})})\}_{i_{\mathcal{C}[l]}}$  a dopředné resp. zpětné hustoty  $\{\mathcal{S}_{F|B}(d^{(i_d)})\}_{i_d}$  a  $\{\mathcal{S}_{F|B}(q^{(i_q)})\}_{i_q}$ , kde  $\mathcal{R}^{(i_{\mathcal{R}})} = \{\mathcal{A}_{\mathcal{R}}\}_{i_{\mathcal{R}}}$ ,  $\mathcal{C}^{(i_{\mathcal{C}})} = \{\mathcal{A}_{\mathcal{C}}\}_{i_{\mathcal{C}}}$ ,  $d^{(i)} = \{\mathcal{A}_d\}_i$  a  $q^{(i)} = \{\mathcal{A}_q\}_i$  jsou prvky odpovídajících abeced  $\mathcal{A}_{\mathcal{R}}$ ,  $\mathcal{A}_{\mathcal{C}}$ ,  $\mathcal{A}_d$  a  $\mathcal{A}_q$ .

$i$	0	1	2	3	4	5	6	7	8	9
$\{\mathcal{A}_d\}_i$	0	1								
$\{\mathcal{A}_q\}_i$	0	$g'$	$2g'$	$3g'$	$4g'$	$g$	$g + g'$	$g + 2g'$	$g + 3g'$	$g + 4g'$

(a) Abecedy vstupních a výstupních proměnných.

$i$	0	1	2	3	4	5	6	7
$\{\mathcal{A}_{\mathcal{R}}\}_i$	0 0	1 0	0 1	1 1	0 0	1 0	0 1	1 1
$\{\mathcal{A}_{\mathcal{C}}\}_i$	0 0	1 0	0 1	1 0	0 1	1 1	0 1	1 1

(b) Abecedy stavových proměnných.

Tab. 4. Abecedy IO proměnných v IDN se základní fixní topologií pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .

Rozšířená varianta iterativní dekodovací sítě s pevnou topologií zahrnuje také zřetězení diagonální [4]. Iterativní dekodovací síť s rozšířenou pevnou topologií znázorňuje Obr. 15a a tvoří ji SISO moduly na Obr. 15b. Testovací estimátor IO množiny SISO modulu je u rozšířené pevné topologie ve tvaru

$$\tilde{\mathcal{N}}_{IO} = \{\tilde{\mathcal{R}}[0], \tilde{\mathcal{C}}[0], \tilde{\mathcal{B}}[0], \tilde{d}, \tilde{\mathcal{R}}[1], \tilde{\mathcal{C}}[1], \tilde{\mathcal{B}}[1], \tilde{q}\}, \quad (74)$$

kde  $\tilde{\mathcal{R}}[\ell]$ ,  $\tilde{\mathcal{C}}[\ell]$  a  $\tilde{\mathcal{B}}[\ell]$  jsou testovací estimátory pomocných stavových proměnných. V případě ukázkového obecného posunutého kernelu (17) z odstavce 3.2.1, tvoří estimátory

$$\begin{aligned} \tilde{\mathcal{R}}[\ell] &= \left\{ \begin{array}{cc} d[-1, \ell - 2] & d[-1, \ell - 1] \\ d[0, \ell - 2] & d[0, \ell - 1] \end{array} \right\}, \quad \tilde{\mathcal{C}}[\ell] = \left\{ \begin{array}{cc} d[\ell - 2, -1] & d[\ell - 2, 0] \\ d[\ell - 1, -1] & d[\ell - 1, 0] \end{array} \right\} \\ \text{a } \tilde{\mathcal{B}}[\ell] &= \left\{ \begin{array}{cc} d[\ell - 2, \ell - 2] & d[\ell - 2, \ell - 1] \\ d[\ell - 1, \ell - 2] & d[\ell - 1, \ell - 1] \end{array} \right\} \end{aligned} \quad (75)$$

čtveřice estimátorů dat  $\tilde{d}$ . Síť tedy bude obsahovat více propojek přenášejících měkké informace o nové stavové proměnné, ale sníží se mohutnosti všech hustot stavových proměnných a to není na škodu. Aktivace SISO modulu v uzlu  $[k, l]$  se skládá z kombinování

$$\begin{aligned} \mathcal{S}(\tilde{\mathcal{N}}_{IO})[k, l] &= \left( \bigcirc_{\tilde{\mathcal{R}}[\ell] \in \tilde{\mathcal{N}}_{IO}} \mathcal{S}_I(\tilde{\mathcal{R}}[\ell])[k, l + \ell] \right) \bigcirc \left( \bigcirc_{\tilde{\mathcal{C}}[\ell] \in \tilde{\mathcal{N}}_{IO}} \mathcal{S}_I(\tilde{\mathcal{C}}[\ell])[k + \ell, l] \right) \\ &\quad \bigcirc \left( \bigcirc_{\tilde{\mathcal{B}}[\ell] \in \tilde{\mathcal{N}}_{IO}} \mathcal{S}_I(\tilde{\mathcal{B}}[\ell])[k + \ell, l + \ell] \right) \bigcirc \mathcal{S}_B(\tilde{d} \in \tilde{\mathcal{N}}_{IO})[k, l] \bigcirc \mathcal{S}_F(\tilde{q} \in \tilde{\mathcal{N}}_{IO})[k, l], \end{aligned} \quad (76)$$

kde  $\ell = 0, 1$ . Získaná sdružená hustota  $\{\mathcal{S}(\mathcal{N}_{IO}^{(i)})\}_i$  je následně marginalizována prostřednictvím sedmi marginalizačních procesů (65), (66), (67) a

$$\mathcal{S}_O(\tilde{\mathcal{B}}[\ell])[k + \ell, l + \ell] = \left( \bigcirc_{\tilde{\mathcal{N}}_{IO}: \tilde{\mathcal{B}}[\ell]} \mathcal{S}(\tilde{\mathcal{N}}_{IO})[k, l] \right) \bigcirc^{-1} \mathcal{S}_I(\tilde{\mathcal{B}}[\ell])[k + \ell, l + \ell], \quad (77)$$

kde  $\ell = 0, 1$  a dále prostřednictvím doplňkového procesu (68).

Před první iterací dekódovací sítě jsou metriky hustot  $\{\mathcal{S}_I(\mathcal{R}^{(i)})[k, l]\}_i$  a  $\{\mathcal{S}_I(\mathcal{C}^{(i)})[k, l]\}_i$  nastaveny dle vztahů (69) a (70) a metriky

$$\mathcal{S}_I(\tilde{\mathcal{B}})[k, l] = \bigoplus_{\tilde{d}[k', l'] \in \tilde{\mathcal{B}}} \mathcal{S}_B(\tilde{d}[k', l'])[k + k', l + l'], \quad (78)$$

analogickým způsobem. Zde  $\mathcal{S}_B(\tilde{d}[k', l'])$  představují apriorní měkké informace jednotlivých estimátorů vstupních dat  $\tilde{d}[k', l']$ , které spadají do blokového estimátoru  $\tilde{\mathcal{B}}$ .

Také rozšířenou fixní topologii, podobně jako tu základní, ještě dále přiblížme na jednoduchém příkladu hvězdicového ISI kanálu (71) s binárními vstupními daty, abychom mohli obě varianty realizací vzájemně porovnat. Pomocné proměnné vzorové kódovací sítě na Obr. 5 budou nyní tři a jejich tvary

$$\mathcal{R}[k, l] = \left\{ \begin{array}{c} d[k-1, l-2] \\ d[k, l-1] \end{array} \right\}, \quad \mathcal{C}[k, l] = \left\{ \begin{array}{c} d[k-2, l-1] \\ d[k-1, l] \end{array} \right\} \quad (79)$$

a  $\mathcal{B}[k, l] = d[k-1, l-1]$

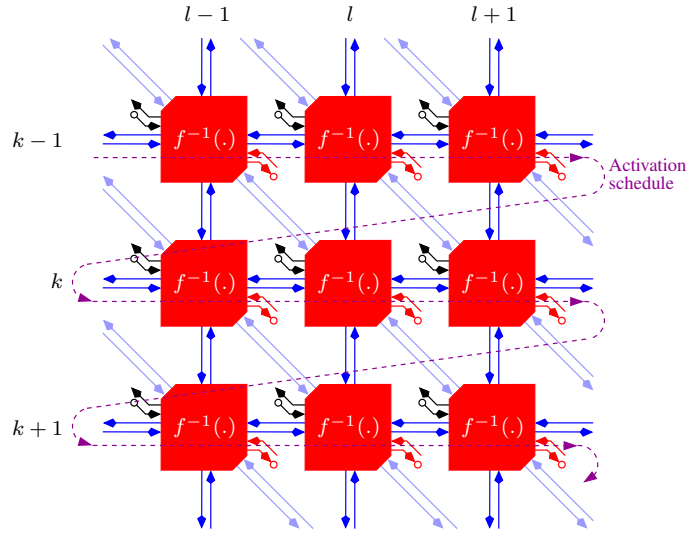
respektují rozšířenou podmínku kauzality (21). Pomocné stavové proměnné graficky znázorňuje Obr. 16, včetně směru jejich posunu nad vstupními daty. Na straně detekce reprezentují tyto stavové veličiny odpovídající testovací estimátory

$$\tilde{\mathcal{R}}[\ell] = \left\{ \begin{array}{c} \tilde{d}[-1, \ell-2] \\ \tilde{d}[0, \ell-1] \end{array} \right\}, \quad \tilde{\mathcal{C}}[\ell] = \left\{ \begin{array}{c} \tilde{d}[\ell-2, -1] \\ \tilde{d}[\ell-1, 0] \end{array} \right\} \quad (80)$$

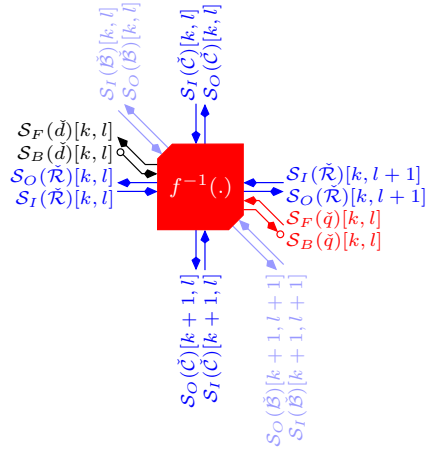
a  $\tilde{\mathcal{B}}[\ell] = \tilde{d}[\ell-1, \ell-1]$ .

Dekódovací síť obsahuje SISO moduly, které jsou definovány kombinacemi vstupních, výstupních a stavových veličin v Tab. 5. Jednotlivé relace jsou opět zaneseny prostřednictvím mapovacích ukazatelů  $i_{\mathcal{R}[\ell]}$ ,  $i_{\mathcal{C}[\ell]}$ ,  $i_d$  a  $i_q$  a dvou nových ukazatelů  $i_{\mathcal{B}[\ell]}$  do abecedy  $\mathcal{A}_{\mathcal{B}}$ . Celkový počet relací  $\text{card}(\mathcal{A}_{IO}) = 64$  je stejný jako v předcházejícím příkladě. Abecedy  $\mathcal{A}_d$  a  $\mathcal{A}_q$  jsou také stejné a lze je nalézt v Tab. 4a. Abecedy  $\mathcal{A}_{\mathcal{R}}$  a  $\mathcal{A}_{\mathcal{C}}$  jsou jiné a společně s abecedou  $\mathcal{A}_{\mathcal{B}}$  je uvádí Tab. 6. Porty SISO modulů se rozšíří o vstupní resp. výstupní hustoty  $\{\mathcal{S}_{IO}(\mathcal{B}^{(i_{\mathcal{B}[\ell]})})\}_{i_{\mathcal{B}[\ell]}}$  na diagonálních spojnicích obsahující partikulární metriky  $\mathcal{S}_{IO}(\tilde{\mathcal{B}}[\ell])[k + \ell, l + \ell]$ , kde  $\mathcal{B}^{(i)} = \{\mathcal{A}_{\mathcal{B}}\}_i$  jsou prvky binární abecedy  $\mathcal{A}_{\mathcal{B}}$ . Při porovnání obou realizací vidíme, že mohutnosti abeced  $\mathcal{A}_{\mathcal{R}}$  a  $\mathcal{A}_{\mathcal{C}}$  se redukovali na  $\text{card}(\mathcal{A}_{\mathcal{R}}) = \text{card}(\mathcal{A}_{\mathcal{C}}) = 4$ . Mohutnost abecedy  $\mathcal{A}_{\mathcal{B}}$  je  $\text{card}(\mathcal{A}_{\mathcal{B}}) = 2$  a proto paměťové nároky klesly celkem o 40%, což je významná hodnota. Je však pravdou, že tato signifikantní úspora paměti je částečně kompenzována další proměnnou zvyšující výpočetní nároky na kombinační i marginalizační část SISO modulů v celé síti.

Závěrem proved' me stručnou sumarizaci iterativních dekódovacích sítí s fixní topologií. Mezi základní přednosti patří především vysoká výkonnost díky mohutným stavovým veličinám, které mnohdy mají i významný překryv. Nevýhodou je, že mezi porty SISO modulů musíme přenášet a skladovat velké množství dat v podobě mohutných hustot. Pro některé druhy kernelů (hvězdicové kernely apod.) je problematické provést optimální separaci konvolučního regionu do jednotlivých stavových proměnných. U některých typů kernelů to není možné vůbec. V takovém případě mají stavové veličiny zbytečně vyšší mohutnost a síť je numericky složitější a paměťové nároky vyšší. Tyto problémy se snaží zmírnit rozšířená varianta s diagonálním zřetěžením. Stavová veličina "navíc" umožňuje vyšší variabilitu v dekompozici konvolučního regionu. Stavové proměnné jsou tři a proto mohou být méně rozlehlé a paměťové nároky se sníží. Také je snazší navrhnout tvary stavových proměnných tak, aby byly vzájemně zcela disjunktní. Potom je snazší nalézt optimální implementaci SISO modulu podle odstavce 5.2.1. Uvedené přednosti však kompenzuje určité zhoršení výkonnosti detektoru a často vyšší výpočetní nároky na kombinačně-marginalizační proces v SISO modulech.

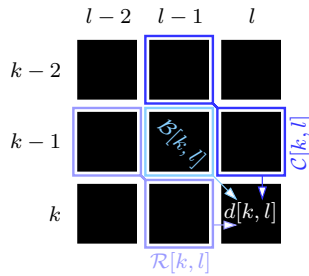


(a) Topologie IDN s vyznačeným aktivačním schématem.



(b) Buňka IDN v uzlu  $[k, l]$ .

Obr. 15. Iterativní dekódovací síť marginalizující na úrovni bloku symbolů s rozšířenou topologií.



Obr. 16. Tvary stavových proměnných  $\mathcal{R}[k, l]$ ,  $\mathcal{C}[k, l]$  a  $\mathcal{B}[k, l]$  pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .



No.	$i_{\mathcal{R}[0]}$	$i_{\mathcal{C}[0]}$	$i_{\mathcal{B}[0]}$	$i_d$	$i_{\mathcal{R}[1]}$	$i_{\mathcal{C}[1]}$	$i_{\mathcal{B}[1]}$	$i_q$	No.	$i_{\mathcal{R}[0]}$	$i_{\mathcal{C}[0]}$	$i_{\mathcal{B}[0]}$	$i_d$	$i_{\mathcal{R}[1]}$	$i_{\mathcal{C}[1]}$	$i_{\mathcal{B}[1]}$	$i_q$
1	0	0	0	0	0	0	0	0	33	0	1	0	0	0	0	0	1
2	0	0	0	1	2	2	1	0	34	0	1	0	1	2	2	1	1
3	2	0	0	0	0	0	0	1	35	2	1	0	0	0	0	0	2
4	2	0	0	1	2	2	1	1	36	2	1	0	1	2	2	1	2
5	0	2	0	0	0	0	0	1	37	0	3	0	0	0	0	0	2
6	0	2	0	1	2	2	1	1	38	0	3	0	1	2	2	1	2
7	2	2	0	0	0	0	0	2	39	2	3	0	0	0	0	0	3
8	2	2	0	1	2	2	1	2	40	2	3	0	1	2	2	1	3
9	0	0	1	0	1	1	0	5	41	0	1	1	0	1	1	0	6
10	0	0	1	1	3	3	1	5	42	0	1	1	1	3	3	1	6
11	2	0	1	0	1	1	0	6	43	2	1	1	0	1	1	0	7
12	2	0	1	1	3	3	1	6	44	2	1	1	1	3	3	1	7
13	0	2	1	0	1	1	0	6	45	0	3	1	0	1	1	0	7
14	0	2	1	1	3	3	1	6	46	0	3	1	1	3	3	1	7
15	2	2	1	0	1	1	0	7	47	2	3	1	0	1	1	0	8
16	2	2	1	1	3	3	1	7	48	2	3	1	1	3	3	1	8
17	1	0	0	0	0	0	0	1	49	1	1	0	0	0	0	0	2
18	1	0	0	1	2	2	1	1	50	1	1	0	1	2	2	1	2
19	3	0	0	0	0	0	0	2	51	3	1	0	0	0	0	0	3
20	3	0	0	1	2	2	1	2	52	3	1	0	1	2	2	1	3
21	1	2	0	0	0	0	0	2	53	1	3	0	0	0	0	0	3
22	1	2	0	1	2	2	1	2	54	1	3	0	1	2	2	1	3
23	3	2	0	0	0	0	0	3	55	3	3	0	0	0	0	0	4
24	3	2	0	1	2	2	1	3	56	3	3	0	1	2	2	1	4
25	1	0	1	0	1	1	0	6	57	1	1	1	0	1	1	0	7
26	1	0	1	1	3	3	1	6	58	1	1	1	1	3	3	1	7
27	3	0	1	0	1	1	0	7	59	3	1	1	0	1	1	0	8
28	3	0	1	1	3	3	1	7	60	3	1	1	1	3	3	1	8
29	1	2	1	0	1	1	0	7	61	1	3	1	0	1	1	0	8
30	1	2	1	1	3	3	1	7	62	1	3	1	1	3	3	1	8
31	3	2	1	0	1	1	0	8	63	3	3	1	0	1	1	0	9
32	3	2	1	1	3	3	1	8	64	3	3	1	1	3	3	1	9

Tab. 5. Tabulka kombinací IO (vstupů, výstupů a stavů) definující strukturu SISO modulů v IDN s rozšířenou fixní topologií pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .

$i$	0	1	2	3
$\{\mathcal{A}_{\mathcal{R}}\}_i, \{\mathcal{A}_{\mathcal{C}}\}_i$	0	1	0	1
$\{\mathcal{A}_{\mathcal{B}}\}_i$	0	1	0	1

Tab. 6. Abecedy stavových proměnných v IDN s rozšířenou fixní topologií pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .

### 5.3.2 Variabilní topologie

Dekódovací sítě s variabilní topologií mají zřetězení závislé na tvaru kernelu. Topologii nelze znázornit z obecného hlediska. Abychom získali představu o zřetězení, uvedeme konkrétní příklad pro obecný kernel  $\mathcal{G}_{3 \times 3}$  se vstupními daty  $d[k, l]$  ve dvojkové aritmetice. Strukturu iterativní dekodovací sítě s variabilní topologií přizpůsobenou pro úlohu dekonvoluce kernelu o velikosti  $3 \times 3$  znázorňuje Obr. 17a. Každý element uvedeného zapojení obsahuje dva SISO moduly [6] na Obr. 17b. SISO moduly znázorněné červeně provádějí měkkou inverzi samotné konvoluce z kompletního konvolučního regionu  $\tilde{\mathcal{N}}[k, l]$ . Na ně navazují měkké inverze broadcasterů, vyznačené černě. Broadcastery v referenčním modelu kanálu  $\mathcal{G}_{3 \times 3}$  na Obr. 7 zodpovídají za větvení vstupních dat a přípravu jednotlivých konvolučních regionů  $\mathcal{N}[k, l]$ . Tuto operaci větvení popisuje jednoduchá kombinační tabulka Tab. 7 definující vnitřní uspořádání měkké inverze broadcasterů v dekodovací síti.

$i \mapsto \mathcal{A}_{IO}^{(B)}$	$i_{c[0]} \mapsto \mathcal{A}_d$	$\dots$	$i_{c[8]} \mapsto \mathcal{A}_d$	$i_d \mapsto \mathcal{A}_d$
0	0	$\dots$	0	0
1	1	$\dots$	1	1

Tab. 7. Tabulka kombinací IO (vstupů a výstupů) definující strukturu měkké inverze binárního broadcasteru.

Díky separaci konvolučního regionu na nejnižší možnou úroveň (úroveň datových symbolů) je tento druh sítě také označován jako síť marginalizující na úrovni symbolů. Flexibilita je opravdu vysoká a síť můžeme snadno a optimálně přizpůsobit i tvarově nestandardním kernelům. Nevýhodou je nižší výkonnost způsobená maximální dekompozicí regionu a velké množství propojek, které není možné obecně předpřipravit. Paměťové nároky jsou ovšem nízké, protože mohutnost metrik stavových veličin je nejmenší možná.

Aktivační schéma je paralelní a znázorněné na Obr. 17a fialovou čárkovanou čarou. Probíhá ve všech elementech naráz a skládá se ze dvou kroků. Nejprve jsou aktivovány měkké inverze kombinačních logik prostřednictvím metrik shromážděných z okolních elementů. Dále proběhnou aktivace měkkých inverzí broadcasterů následované distribucí výstupních metrik do sousedních elementů. V další iteraci  $I$  se proces opakuje.

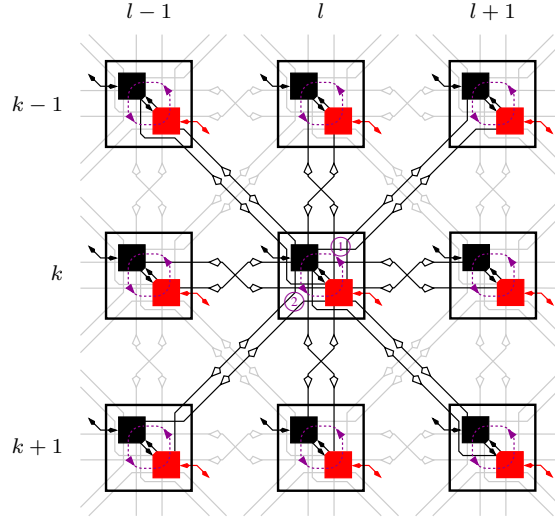
Testovací estimátor IO množiny měkké inverze kombinační logiky je v případě příkladu variabilní topologie na Obr. 17a ve tvaru

$$\tilde{\mathcal{N}}_{IO} = \left\{ \begin{array}{ccc} \check{c}[7] & \check{c}[6] & \check{c}[5] \\ \check{c}[0] & \check{c}[8] & \check{c}[4] \\ \check{c}[1] & \check{c}[2] & \check{c}[3] \end{array} \right\} \cup \check{q}, \quad (81)$$

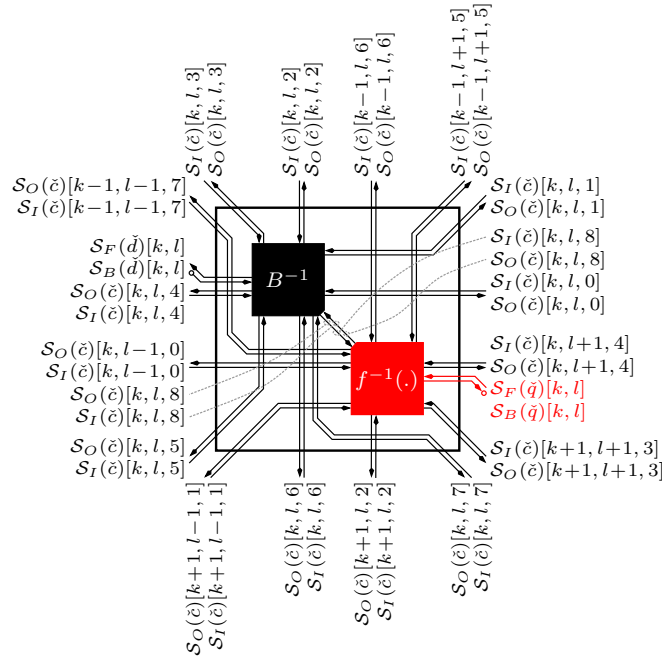
kde  $\check{c}[\ell]$  jsou testovací estimátory výstupů sousedících měkkých inverzí broadcasterů. Aktivace měkké inverze kombinační logiky v uzlu  $[k, l]$  se skládá z kombinování

$$\mathcal{S}(\tilde{\mathcal{N}}_{IO})[k, l] = \left( \bigodot_{\check{c}[\ell] \in \tilde{\mathcal{N}}_{IO}} \mathcal{S}_I(\check{c}[\ell])[k + \{\mathcal{I}_k\}_\ell, l + \{\mathcal{I}_l\}_\ell, \ell] \right) \odot \mathcal{S}_F(\check{q} \in \tilde{\mathcal{N}}_{IO})[k, l], \quad (82)$$

kde  $\ell = 0, 1, \dots, 8$  a  $\mathcal{I}_k$  společně s  $\mathcal{I}_l$  jsou množiny indexů (28) a (29) definující propojení mezi měkkými inverzemi broadcasterů a kombinačních logik. Získaná sružená hustota  $\{\mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(i)})\}_i$  je následně marginalizována prostřednictvím devíti marginalizačních procesů



(a) Topologie IDN s vyznačeným aktivačním schématem.



(b) Buňka IDN v uzlu  $[k, l]$ .

Obr. 17. Iterativní dekodovací síť marginalizující na úrovni symbolů pro obecný ISI kernel  $\mathcal{G}_{3 \times 3}$ .

$$S_O(\check{e}[\ell])[k + \{\mathcal{I}_k\}_\ell, l + \{\mathcal{I}_l\}_\ell] = \left( \bigotimes_{\check{N}_{IO:\check{e}[\ell]}} S(\check{N}_{IO})[k, l] \right) \odot^{-1} S_I(\check{e}[\ell])[k + \{\mathcal{I}_k\}_\ell, l + \{\mathcal{I}_l\}_\ell], \quad (83)$$

kde  $\ell = 0, 1, \dots, 8$  a dále prostřednictvím doplňkového procesu (68).

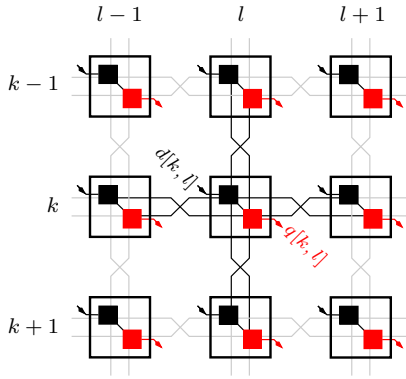
Následně se aktivační proces sítě přesune na úroveň měkkých inverzí binárních broadcasterů. Aktivaci těchto elementů tvoří pouze dvojice kombinačních procesů

$$\mathcal{S}_F(\check{d})[k, l] = \bigodot_{\check{c}[l]=\check{d}} \mathcal{S}_I(\check{c}[l])[k, l, \ell] \quad (84)$$

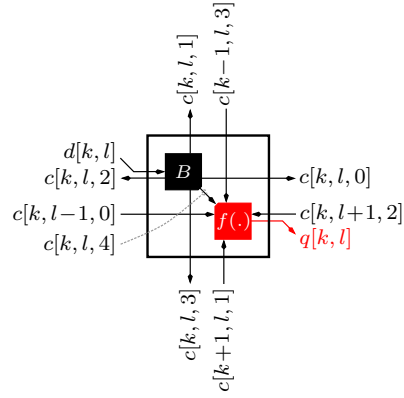
a

$$\mathcal{S}_O(\check{c}[l])[k, l, \ell] = \left( \bigodot_{\check{c}[\ell']=\check{c}[l], \ell' \neq \ell} \mathcal{S}_I(\check{c}[\ell'])[k, l, \ell'] \right) \bigodot \mathcal{S}_B(\check{d} = \check{c}[l])[k, l], \quad (85)$$

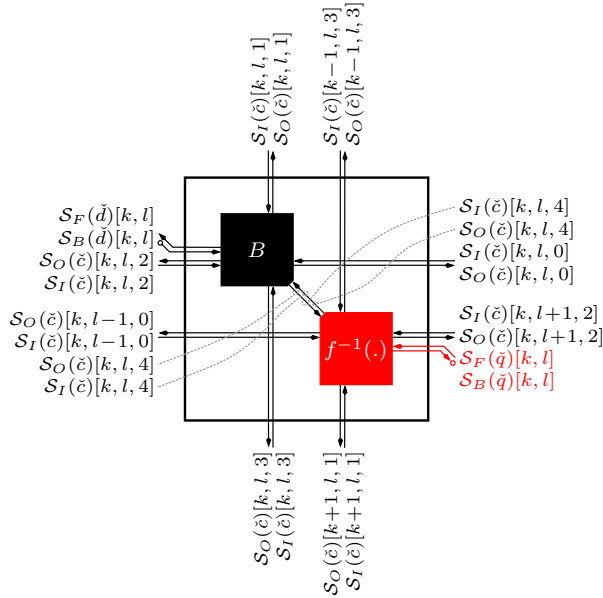
kteře přímo vyplývají z Tab. 7.



(a) Topologie EN/IDN.



(b) Buňka EN v uzlu  $[k, l]$ .



(c) Buňka IDN v uzlu  $[k, l]$ .

Obr. 18. Iterativní dekódovací síť marginalizující na úrovni symbolů pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .

I tuto topologii ještě dále přiblížíme na jednoduchém příkladu hvězdicového ISI kanálu (71) s binárními vstupními daty. Oproti předcházejícím příkladům s fixními topologiemi v odstavci 5.3.1 je zde drobný rozdíl v tom, že kernel nebude posunutý a vstup  $d[k, l]$  je ztotožněn se středem kernelu, neboli s koeficientem  $g$ . Topologii vzorové kódovací sítě uvádí Obr. 18a. Každá buňka této struktury tvoří kompozitní element na Obr. 18b. Proměnné na vstupu kombinačních logik jednotlivých elementů sdružuje množina

$$\mathcal{N}_{IO}[k, l] = \{c[k + \{\mathcal{I}_k\}_\ell, l + \{\mathcal{I}_l\}_\ell, \ell]\}_{\ell < 5} \cup q[k, l], \quad (86)$$

kde

$$\mathcal{I}_k = \{0, +1, 0, -1, 0\} \quad (87)$$

a

$$\mathcal{I}_l = \{-1, 0, +1, 0, 0\} \quad (88)$$

jsou množiny indexů definující horizontální a vertikální propojení kombinačních logik s broadcastery v sousedních elementech kódovací sítě. Na straně detekce budou v předložené topologii nahrazeny kódovací elementy na Obr. 18b odpovídajícími SISO moduly na Obr. 18c. Testovací estimátor  $IO$  množiny měkké inverze kombinační logiky bude ve tvaru

$$\check{\mathcal{N}}_{IO} = \left\{ \begin{array}{ccc} & \check{c}[3] & \\ \check{c}[0] & \check{c}[4] & \check{c}[2] \\ & \check{c}[1] & \end{array} \right\} \cup \check{q}. \quad (89)$$

Jednotlivé testovací estimátory binárních dat  $\check{c}[\ell]$  pak tvoří kombinace v Tab. 8. Uvedené relace jsou zde definovány mapovacími ukazateli  $i_{c[\ell]}$  resp.  $i_q$  do příslušných abeced  $\mathcal{A}_d$  resp.  $\mathcal{A}_q$ , které lze nalézt v Tab. 4a. Mohutnost kombinační tabulky je  $\text{card}(\mathcal{A}_{IO}) = 32$ , tedy optimální. Partikulární metriky  $\mathcal{S}_{IO}(\check{c}[\ell])[k + \{\mathcal{I}_k\}_\ell, l + \{\mathcal{I}_l\}_\ell, \ell]$ ,  $\mathcal{S}_{F|B}(\check{d})[k, l]$  a  $\mathcal{S}_{F|B}(\check{q})[k, l]$  tvoří na portech SISO modulu vstupní resp. výstupní hustoty  $\{\mathcal{S}_{IO}(c^{(i_{c[\ell]})})\}_{i_{c[\ell]}}$  a dopředné resp. zpětné hustoty  $\{\mathcal{S}_{F|B}(d^{(i_d)})\}_{i_d}$  a  $\{\mathcal{S}_{F|B}(q^{(i_q)})\}_{i_q}$ , kde  $c^{(i_{c[\ell]})} = \{\mathcal{A}_d\}_{i_{c[\ell]}}$ ,  $d^{(i)} = \{\mathcal{A}_d\}_i$  a  $q^{(i)} = \{\mathcal{A}_q\}_i$  jsou prvky odpovídajících abeced  $\mathcal{A}_d$  a  $\mathcal{A}_q$ .

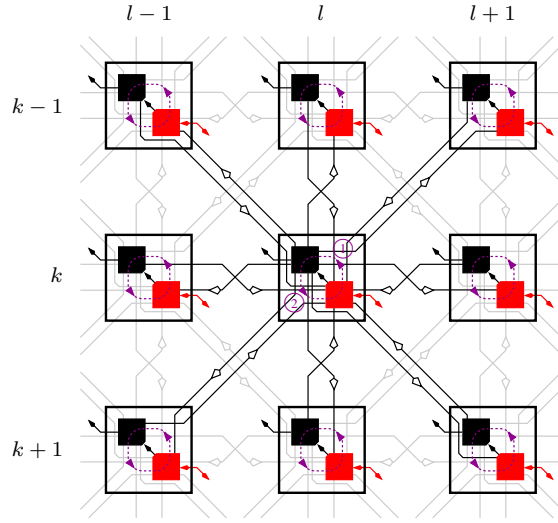
Z výše uvedených skutečností je jasné patrné, že pro tentýž hvězdicový kernel (71) došlo u variabilní topologie, v porovnání s fixní topologií, k redukci kombinací v Tab. 8 na polovinu. Tím dojde ke snížení výpočetních nároků jednotlivých SISO modulů na polovinu. Tuto redukci způsobuje flexibilita variabilní topologie a její snadná aplikovatelnost na libovolný kernel. Vstupní proměnné  $d$  jednotlivých buněk vzorové kódovací sítě můžeme vztáhnout k jakémukoliv koeficientu v tomto kernelu. Ve výše uvedeném příkladě to byl střed kernelu  $\mathcal{G}_{3 \times 3}^{(R)}$ . U fixní topologie toto není možné. Zde je vstup vztážen vždy k levému spodnímu rohu, aby vzniklá struktura mohla být kauzální a splnila podmínku (16) resp. (21). Pokud je přenos kernelu v tomto místě nulový, pak dochází u fixních topologií k neoptimální situaci, která zbytečně zvyšuje mohutnost  $\text{card}(\mathcal{A}_{IO})$  nad minimální hranici. Kromě optimální mohutnosti  $\text{card}(\mathcal{A}_{IO})$  má variabilní topologie také minimální mohutnosti pomocných proměnných  $c[\ell]$ , což vede na nízké paměťové nároky. Pomocné proměnné  $c[\ell]$  jsou navíc zcela disjunktní. Proto lze snadno nalézt optimální implementaci SISO modulu prostřednictvím stromové kombinace a marginalizace. Závěr je takový, že pro nestandardní nebo hvězdicové kernely je variabilní topologie nejlepší variantou. Na rozdíl od fixních topologií, které se na tyto kernely špatně aplikují a SISO moduly sítě mají zbytečně vysokou složitost. Všechny výhody detektorů s variabilní topologií však kompenzuje jejich výkonnost, která je nižší v porovnání s výkonností detektorů se základní fixní topologií nebo rozšířenou fixní topologií.

No.	$i_{c[0]}$	$i_{c[3]}$	$i_{c[4]}$	$i_{c[1]}$	$i_{c[2]}$	$i_q$	No.	$i_{c[0]}$	$i_{c[3]}$	$i_{c[4]}$	$i_{c[1]}$	$i_{c[2]}$	$i_q$
1	0	0	0	0	0	0	17	1	0	0	0	0	1
2	0	0	0	0	1	1	18	1	0	0	0	1	2
3	0	0	0	1	0	1	19	1	0	0	1	0	2
4	0	0	0	1	1	2	20	1	0	0	1	1	3
5	0	0	1	0	0	5	21	1	0	1	0	0	6
6	0	0	1	0	1	6	22	1	0	1	0	1	7
7	0	0	1	1	0	6	23	1	0	1	1	0	7
8	0	0	1	1	1	7	24	1	0	1	1	1	8
9	0	1	0	0	0	1	25	1	1	0	0	0	2
10	0	1	0	0	1	2	26	1	1	0	0	1	3
11	0	1	0	1	0	2	27	1	1	0	1	0	3
12	0	1	0	1	1	3	28	1	1	0	1	1	4
13	0	1	1	0	0	6	29	1	1	1	0	0	7
14	0	1	1	0	1	7	30	1	1	1	0	1	8
15	0	1	1	1	0	7	31	1	1	1	1	0	8
16	0	1	1	1	1	8	32	1	1	1	1	1	9

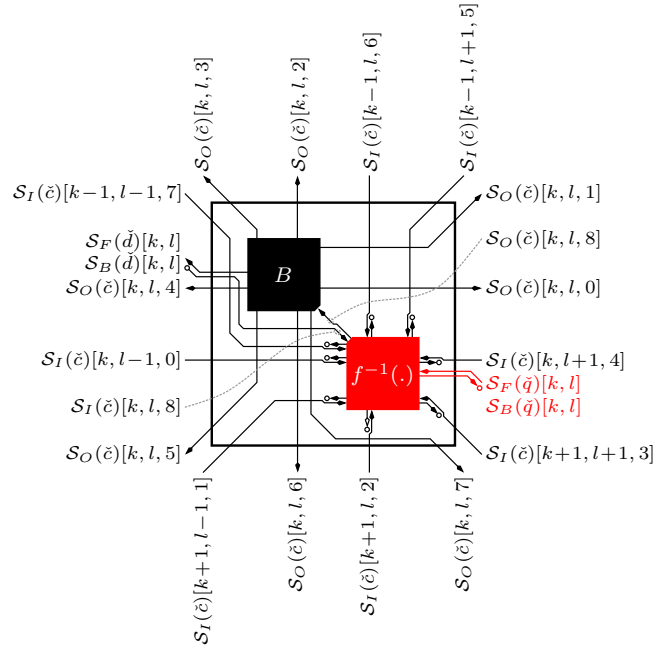
Tab. 8. Tabulka kombinací IO (vstupů a výstupů) definující strukturu SISO modulů v IDN s variabilní topologií pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(R)}$ .

### 5.3.3 Zjednodušená variabilní topologie

Sít s variabilní topologií lze také implementovat ve zjednodušené variantě s další významnou úsporou výpočetních požadavků. Zjednodušený element [12] takového detektoru je uveden na Obr. 19. Implementace modulu je založena na předpokladu, že střed kernelu tvoří dominantní koeficient. Díky tomu je výstup referenčního modelu ISI kanálu  $\mathcal{G}_{3 \times 3}$  majoritně ovlivňován středem konvolučního regionu a minoritně všemi osmi sousedícími hodnotami. Proto element obsahuje pouze jeden SISO modul, který navíc provádí pouze jednu marginalizaci a to stavové proměnné ve středu regionu. Výstupem je jedna klíčová hustota  $\{\mathcal{S}_O(\tilde{c})[k, l, 8]\}$  a ta je distribuována do sousedních bloků klasickým broadcasterem. Spojnice v topologii jsou tedy stejné, ale na rozdíl od uspořádání na Obr. 17a probíhá distribuce metrik jen jedním směrem (od broadcasteru). Místo devíti marginalizací, v neredukované struktuře, se provádí pouze jedna, což přináší obrovský pokles výpočetní náročnosti. Navíc není zapotřebí měkká inverze broadcasteru. Množství propojek je také redukováno a to na polovinu, protože tok informací je jen jednosměrný. Z toho důvodu i paměťové nároky klesnou na polovinu. Všechny příznivé vlastnosti se ovšem promítají do nižší výkonnosti a naprosté nepoužitelnosti struktury na složitější úlohy. Například na detekci proděravěného vstupního signálu  $\mathbf{R}$ .



(a) Topologie IDN s vyznačeným aktivačním schématem.

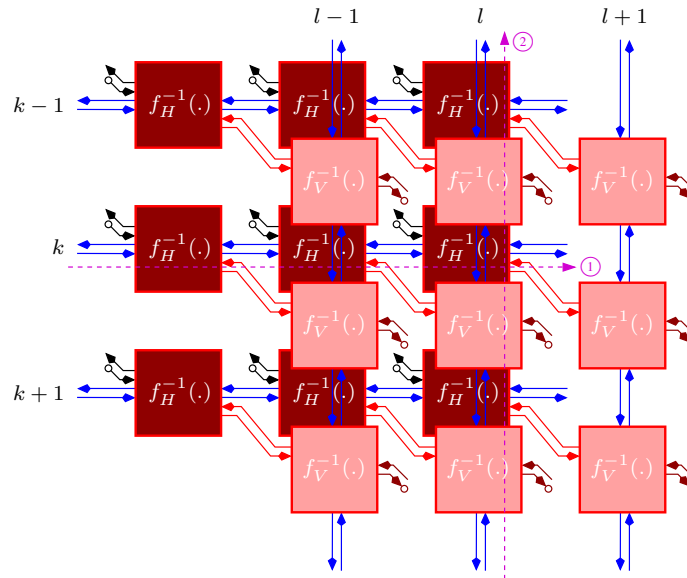


(b) Buňka IDN v uzlu  $[k, l]$ .

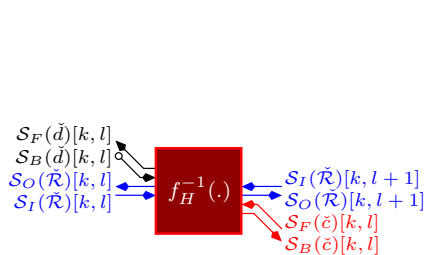
Obr. 19. Zjednodušená iterativní dekódovací síť marginalizující na úrovni symbolů pro obecný ISI kernel  $\mathcal{G}_{3 \times 3}$  s dominantním koeficientem.

### 5.3.4 Vrstvové sítě pro kanály s dekomponovatelným kernelem

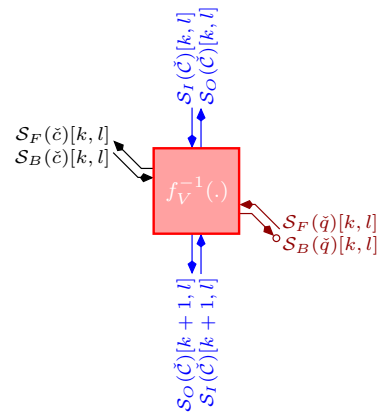
Kombinaci obou předcházejících přístupů představují vrstvené topologie. Mají sice omezenou aplikovatelnost, ale také určité pozitivní vlastnosti. Vrstvenou topologii můžeme použít pouze za předpokladu, že kernel je dekomponovatelný do dvou kolmých směrů tak, jak bylo řečeno v závěru odstavce 3.2.3. Pak dvoudimenzionální konvoluci lze plnohodnotně nahradit dvěma následnými jednodimenzionálními konvolucemi, kdy referenční model kanálu má dvě vrstvy. Proto i iterativní dekódovací síť má dvě vrstvy a na každé z těchto vrstev můžeme použít fixní nebo variabilní topologii, potažmo kombinaci obou topologií.



(a) Topologie vrstevné IDN sestavené ze dvou podsítí s fixní topologií a s vyznačeným akivačním schématem.



(b) Buňka IDN v uzlu  $[k, l]$  spodní vrstvy.



(c) Buňka IDN v uzlu  $[k, l]$  vrchní vrstvy.

Obr. 20. Vrstvová iterativní dekódovací síť marginalizující na úrovni bloku symbolů.

Pokud použijeme fixní topologii vznikne detektor na Obr. 20a, kde spodní vrstva je zřetězená horizontálně a vrchní vertikálně. SISO moduly na spodní resp. horní vrstvě přibližuje Obr. 20b resp. Obr. 20c. Iterace systému se skládá z paralelní aktivace “po sloupcích” vrchní vrstvy, kterou následuje paralelní aktivace “po řádcích” spodní vrstvy. Testovací estimátor IO množiny SISO modulu na spodní resp. horní vrstvě je ve tvaru



$$\tilde{\mathcal{N}}_{IO}^{(H)} = \{\tilde{\mathcal{R}}[0], \check{d}, \tilde{\mathcal{R}}[1], \check{c}\} \quad (90)$$

resp.

$$\tilde{\mathcal{N}}_{IO}^{(V)} = \{\check{\mathcal{C}}[0], \check{c}, \check{\mathcal{C}}[1], \check{q}\}, \quad (91)$$

kde  $\tilde{\mathcal{R}}[\ell]$  a  $\check{\mathcal{C}}[\ell]$  jsou testovací estimátory pomocných stavových proměnných. V případě ukázkového dekomponovatelného kernelu (33) z odstavce 3.2.3, tvoří estimátory

$$\tilde{\mathcal{R}}[\ell] = \{ \check{d}[0, \ell - 2] \quad \check{d}[0, \ell - 1] \} \quad (92)$$

resp.

$$\check{\mathcal{C}}[\ell] = \left\{ \begin{array}{l} \check{c}[\ell - 2, 0] \\ \check{c}[\ell - 1, 0] \end{array} \right\} \quad (93)$$

dvojice estimátorů vstupních dat  $\check{d}$  resp. kódových symbolů  $\check{c}$  na výstupu spodní vrstvy. Aktivace SISO modulu v uzlu  $[k, l]$  na vrchní vrstvě tvoří kombinační operace

$$\begin{aligned} \mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(V)})[k, l] &= \left( \bigoplus_{\check{c}[\ell] \in \tilde{\mathcal{N}}_{IO}^{(V)}} \mathcal{S}_I(\check{\mathcal{C}}[\ell])[k + \ell, l] \right) \\ &\quad \odot \mathcal{S}_B(\check{c} \in \tilde{\mathcal{N}}_{IO}^{(V)})[k, l] \odot \mathcal{S}_F(\check{q} \in \tilde{\mathcal{N}}_{IO}^{(V)})[k, l], \end{aligned} \quad (94)$$

kterou následují marginalizace získané sdružené hustoty  $\{\mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(V)(i)})\}_i$ . Marginalizační proces tvoří jednak dvojice marginalizací

$$\mathcal{S}_O(\check{\mathcal{C}}[\ell])[k + \ell, l] = \left( \bigotimes_{\tilde{\mathcal{N}}_{IO}^{(V)}: \check{c}[\ell]} \mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(V)})[k, l] \right) \odot^{-1} \mathcal{S}_I(\check{\mathcal{C}}[\ell])[k + \ell, l] \quad (95)$$

v rámci vrchní vrstvy, kde  $\ell = 0, 1$ . Dále výpočet metrik

$$\mathcal{S}_F(\check{c})[k, l] = \left( \bigotimes_{\tilde{\mathcal{N}}_{IO}^{(V)}: \check{c}} \mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(V)})[k, l] \right) \odot^{-1} \mathcal{S}_B(\check{c})[k, l], \quad (96)$$

vstupních hustot  $\{\mathcal{S}_F(\check{c})\}$  spodní vrstvy. Tyto tři klíčové marginalizační operace může ještě rozšířit doplňková marginalizace

$$\mathcal{S}_B(\check{q})[k, l] = \left( \bigotimes_{\tilde{\mathcal{N}}_{IO}^{(V)}: \check{q}} \mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(V)})[k, l] \right) \odot^{-1} \mathcal{S}_F(\check{q})[k, l]. \quad (97)$$

Po provedení aktivací všech SISO modulů na vrchní vrstvě přejde aktivační proces na spodní vrstvu struktury. Aktivace SISO modulu v uzlu  $[k, l]$  na spodní vrstvě tvoří analogická kombinační operace

$$\begin{aligned} \mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(H)})[k, l] &= \left( \bigoplus_{\tilde{\mathcal{R}}[\ell] \in \tilde{\mathcal{N}}_{IO}^{(H)}} \mathcal{S}_I(\tilde{\mathcal{R}}[\ell])[k, l + \ell] \right) \\ &\quad \odot \mathcal{S}_B(\check{d} \in \tilde{\mathcal{N}}_{IO}^{(H)})[k, l] \odot \mathcal{S}_F(\check{c} \in \tilde{\mathcal{N}}_{IO}^{(H)})[k, l] \end{aligned} \quad (98)$$

s výstupem v podobě sdružené hustoty  $\{\mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(H)(i)})\}_i$ . Následuje výpočet čtveřice marginalizací této hustoty

$$\mathcal{S}_O(\tilde{\mathcal{R}}[\ell])[k, l + \ell] = \left( \bigotimes_{\tilde{\mathcal{N}}_{IO}^{(H)}: \tilde{\mathcal{R}}[\ell]} \mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(H)})[k, l] \right) \odot^{-1} \mathcal{S}_I(\tilde{\mathcal{R}}[\ell])[k, l + \ell], \quad (99)$$

$$\mathcal{S}_F(\check{d})[k, l] = \left( \bigoplus_{\check{\mathcal{N}}_{IO}^{(H)}:\check{d}} \mathcal{S}(\check{\mathcal{N}}_{IO}^{(H)})[k, l] \right) \odot^{-1} \mathcal{S}_B(\check{d})[k, l] \quad (100)$$

a

$$\mathcal{S}_B(\check{c})[k, l] = \left( \bigoplus_{\check{\mathcal{N}}_{IO}^{(H)}:\check{c}} \mathcal{S}(\check{\mathcal{N}}_{IO}^{(H)})[k, l] \right) \odot^{-1} \mathcal{S}_F(\check{c})[k, l], \quad (101)$$

kde  $\ell = 0, 1$ . Marginalizace (101) je analogická k doplňkové marginalizaci (97) na vrchní vrstvě. Zásadní rozdíl je v tom, že marginalizace (101) není volitelná, ale klíčová. Prostřednictvím přepočtených hustot  $\{\mathcal{S}_B(\check{c})\}$  se totiž uzavírají všechny smyčky zpětných vazeb mezi spodní a vrchní vrstvou sítě. V průběhu iteračního procesu se tedy zpřesňují nejen stavové metriky  $\mathcal{S}_{I|O}(\check{\mathcal{R}})$  a  $\mathcal{S}_{I|O}(\check{\mathcal{C}})$  na jednotlivých vrstvách, ale také “mezi-vrstvové” metriky  $\mathcal{S}_{F|B}(\check{c})$ .

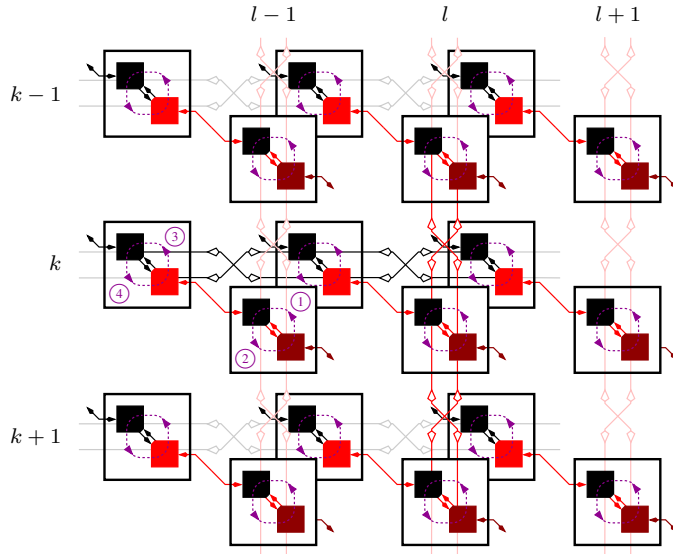
Pokud iterativní dekódovací síť na Obr. 20a realizuje detekci vstupních dat ve dvojkové aritmetice po jejich průchodu kernelem (33), pak vnitřní uspořádání SISO modulů na spodní vrstvě topologie je definováno relacemi v Tab. 9a. Celkový počet realizací  $\mathcal{N}_{IO}^{(H)(i)}$  množiny  $\check{\mathcal{N}}_{IO}^{(H)}$  je roven  $\text{card}(\mathcal{A}_{IO}^{(H)}) = 8$ . Příslušné kombinace vstupu, výstupu a stavů jsou zde definovány prostřednictvím mapovacích ukazatelů  $i_{\mathcal{R}[\ell]}$ ,  $i_d$  a  $i_c$  do abeced  $\mathcal{A}_{\mathcal{R}}$ ,  $\mathcal{A}_d$  a  $\mathcal{A}_c$  o mohutnostech  $\text{card}(\mathcal{A}_{\mathcal{R}}) = 4$ ,  $\text{card}(\mathcal{A}_d) = 2$  a  $\text{card}(\mathcal{A}_c) = 6$ . Jednotlivé abecedy jsou uvedeny v Tab. 9b a Tab. 9c. U vrchní vrstvy je celkový počet realizací  $\mathcal{N}_{IO}^{(V)(i)}$  množiny  $\check{\mathcal{N}}_{IO}^{(V)}$  roven  $\text{card}(\mathcal{A}_{IO}^{(V)}) = 216$ . Mohutnost stavové proměnné činí  $\text{card}(\mathcal{A}_{\mathcal{C}}) = 36$  a výstupní proměnné  $\text{card}(\mathcal{A}_q) = 50$ . Tabulka kombinací definující SISO moduly na vrchní vrstvě není uvedena z důvodu velkého počtu relací.

No.	$i_{\mathcal{R}[0]}$	$i_d$	$i_{\mathcal{R}[1]}$	$i_c$	$i$	0	1	2	3	4	5
1	0	0	0	0	$\{\mathcal{A}_d\}_i$	0	1				
2	0	1	1	1	$\{\mathcal{A}_c\}_i$	0	$g'_H$	$2g'_H$	$g_H$	$g_H + g'_H$	$g_H + 2g'_H$
3	1	0	2	3	(b) Abecedy vstupních a výstupních proměnných.						
4	1	1	3	4							
5	2	0	0	1							
6	2	1	1	2	$i$	0	1	2	3		
7	3	0	2	4	$\{\mathcal{A}_{\mathcal{R}}\}_i$	0	0	0	1	1	0
8	3	1	3	5							1

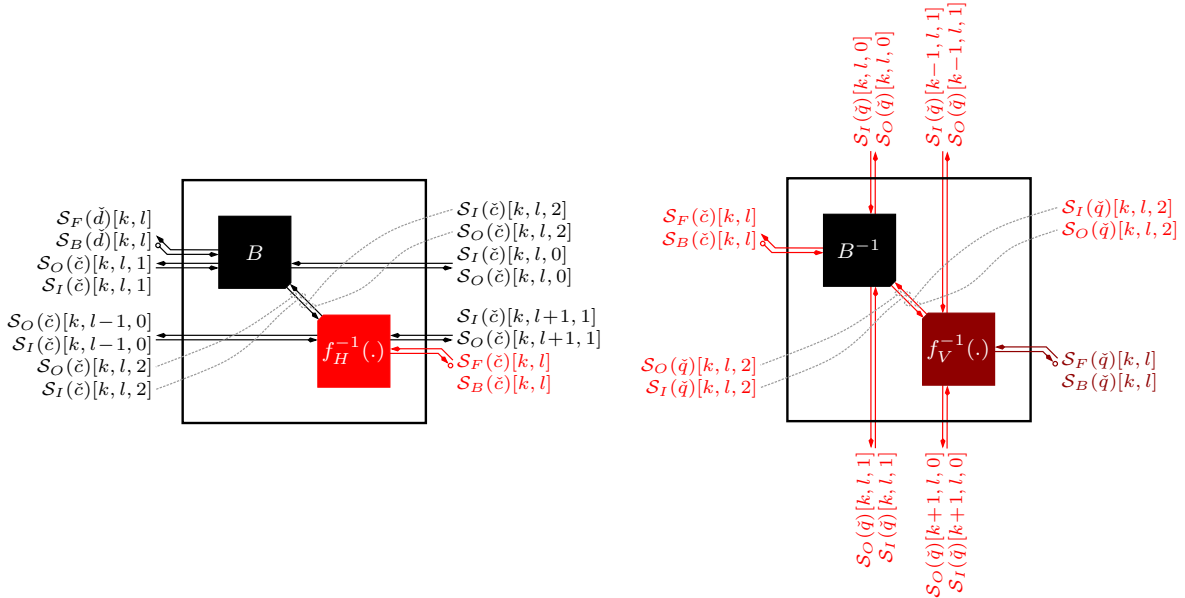
(a) Tabulka kombinací.

(c) Abeceda stavové proměnné.

Tab. 9. Tabulka kombinací IO (vstupů, výstupů a stavů) definující strukturu SISO modulů ve spodní vrstvě IDN s vrstvou fixní topologií pro obecný dekomponovatelný ISI kernel  $\mathcal{G}_{3 \times 3}^{(HV)}$ .



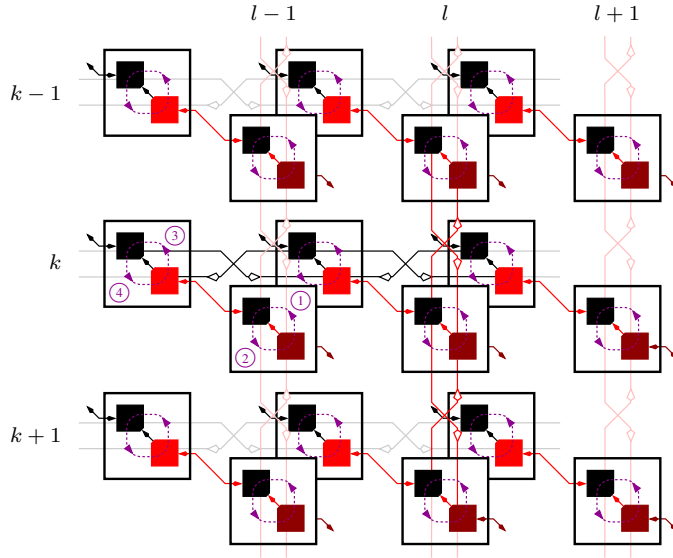
(a) Topologie vrstevné IDN sestavené ze dvou podsítí s variabilní topologií a s vyznačeným aktivačním schématem.



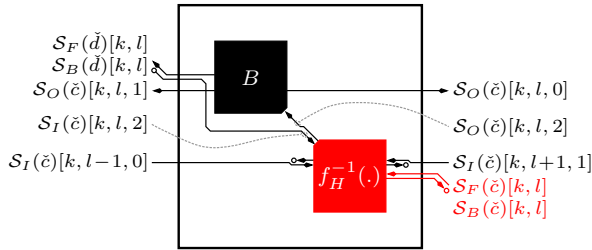
(b) Buňka IDN v uzlu  $[k, l]$  spodní vrstvy.

(c) Buňka IDN v uzlu  $[k, l]$  vrchní vrstvy.

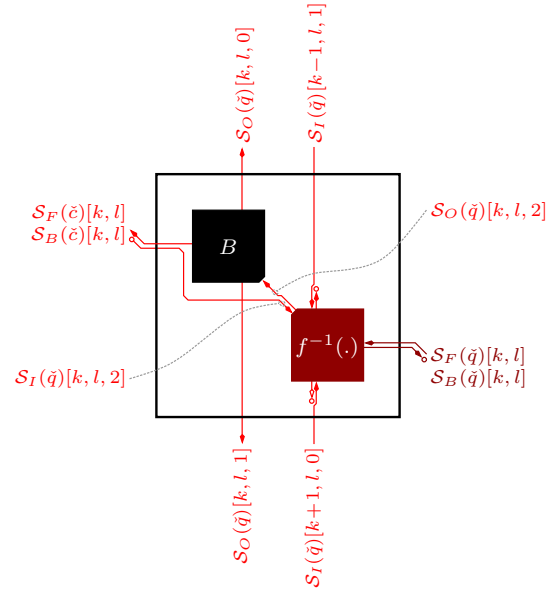
Obr. 21. Vrstvová iterativní dekodovací síť marginalizující na úrovni symbolů pro obecný dekomponovatelný ISI kernel  $\mathcal{G}_{3 \times 3}^{(HV)}$ .



(a) Topologie vrstevné IDN sestavené ze dvou zjednodušených podsítí s variabilní topologií a s vyznačeným aktivačním schématem.



(b) Buňka IDN v uzlu  $[k, l]$  spodní vrstvy.



(c) Buňka IDN v uzlu  $[k, l]$  vrchní vrstvy.

Obr. 22. Zjednodušená vrstevná iterativní dekodovací síť marginalizující na úrovni symbolů pro obecný dekomponovatelný ISI kernel  $\mathcal{G}_{3 \times 3}^{(HV)}$  s dominantním koeficientem.

V případě použití variabilní topologie vznikne struktura na Obr. 21. Toto konkrétní zapojení by řešilo dekonvoluci libovolného dekomponovatelného ISI kanálu o velikosti  $3 \times 3$ . Aktivace opět začíná na vrchní vrstvě a je následována aktivací spodní vrstvy. SISO moduly na spodní resp. horní vrstvě znázorňuje Obr. 21b resp. Obr. 21c. Testovací estimátor  $IO$  množiny měkké inverze kombinační logiky spodní resp. vrchní vrstvy je ve tvaru

$$\tilde{\mathcal{N}}_{IO}^{(H)} = \{ \check{c}[0] \quad \check{c}[2] \quad \check{c}[1] \} \cup \check{c} \quad (102)$$

resp.

$$\tilde{\mathcal{N}}_{IO}^{(V)} = \{ \check{q}[0] \quad \check{q}[2] \quad \check{q}[1] \} \cup \check{q}, \quad (103)$$

kde  $\check{c}[\ell]$  resp.  $\check{q}[\ell]$  jsou testovací estimátory výstupů sousedících měkkých inverzí broadcasterů na spodní resp. vrchní vrstvě sítě. Aktivace měkké inverze kombinační logiky v uzlu  $[k, l]$  na vrchní vrstvě tvoří kombinační operace

$$\mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(V)})[k, l] = \left( \bigcirc_{\check{q}[\ell] \in \tilde{\mathcal{N}}_{IO}^{(V)}} \mathcal{S}_I(\check{q}[\ell])[k + \{\mathcal{I}_k^{(V)}\}_{\ell}, l, \ell] \right) \bigcirc_{\mathcal{S}_F(\check{q} \in \tilde{\mathcal{N}}_{IO}^{(V)})} [k, l], \quad (104)$$

kde  $\ell = 0, 1, 2$  a  $\mathcal{I}_k^{(V)}$  je množina indexů (46) definující propojení mezi měkkými inverzemi broadcasterů a kombinačních logik vrchní vrstvy. Výstupem kombinační operace (104) je sdružená hustota  $\{\mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(V)(i)})\}_i$ , která je následně marginalizována prostřednictvím třech marginalizačních procesů

$$\mathcal{S}_O(\check{q}[\ell])[k + \{\mathcal{I}_k^{(V)}\}_{\ell}, l, \ell] = \left( \bigotimes_{\tilde{\mathcal{N}}_{IO}^{(V)}:\check{q}[\ell]} \mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(V)})[k, l] \right) \bigcirc^{-1} \mathcal{S}_I(\check{q}[\ell])[k + \{\mathcal{I}_k^{(V)}\}_{\ell}, l, \ell] \quad (105)$$

kde  $\ell = 0, 1, 2$  a popřípadě také doplňkovým procesem

$$\mathcal{S}_B(\check{q})[k, l] = \left( \bigotimes_{\tilde{\mathcal{N}}_{IO}^{(V)}:\check{q}} \mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(V)})[k, l] \right) \bigcirc^{-1} \mathcal{S}_F(\check{q})[k, l]. \quad (106)$$

Dále se aktivační proces přesune na úroveň měkkých inverzí broadcasterů vrchní vrstvy sítě. V těchto elementech se provede pouze dvojice kombinačních procesů

$$\mathcal{S}_F(\check{c})[k, l] = \bigcirc_{\check{q}[\ell]=\check{c}} \mathcal{S}_I(\check{q}[\ell])[k, l, \ell] \quad (107)$$

a

$$\mathcal{S}_O(\check{q}[\ell])[k, l, \ell] = \left( \bigcirc_{\check{q}[\ell']=\check{q}[\ell], \ell' \neq \ell} \mathcal{S}_I(\check{q}[\ell'])[k, l, \ell'] \right) \bigcirc_{\mathcal{S}_B(\check{c} = \check{q}[\ell])} [k, l]. \quad (108)$$

Klíčovými výstupy aktivace jsou hustoty  $\{\mathcal{S}_F(\check{c})\}$  předané na mezivrstevové vstupní porty spodní vrstvy. Po provedení aktivací všech kompozitních SISO modulů na vrchní vrstvě přejde aktivační proces na úroveň měkkých kombinačních logik ve spodní vrstvě sítě. Aktivace měkké inverze kombinační logiky v uzlu  $[k, l]$  na spodní vrstvě tvoří kombinační operace

$$\mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(H)})[k, l] = \left( \bigcirc_{\check{c}[\ell] \in \tilde{\mathcal{N}}_{IO}^{(H)}} \mathcal{S}_I(\check{c}[\ell])[k, l + \{\mathcal{I}_l^{(H)}\}_{\ell}, \ell] \right) \bigcirc_{\mathcal{S}_F(\check{c} \in \tilde{\mathcal{N}}_{IO}^{(H)})} [k, l], \quad (109)$$

kde  $\ell = 0, 1, 2$  a  $\mathcal{I}_k^{(H)}$  je množina indexů (42) definující propojení mezi měkkými inverzemi broadcasterů a kombinačních logik spodní vrstvy. Výstupem kombinační operace (109) je sdružená hustota  $\{\mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(H)(i)})\}_i$ . Získaná hustota je následně marginalizována prostřednictvím čtyřech marginalizačních procesů

$$\mathcal{S}_O(\check{c}[\ell])[k, l + \{\mathcal{I}_l^{(H)}\}_{\ell}, \ell] = \left( \bigotimes_{\tilde{\mathcal{N}}_{IO}^{(H)}:\check{c}[\ell]} \mathcal{S}(\tilde{\mathcal{N}}_{IO}^{(H)})[k, l] \right) \bigcirc^{-1} \mathcal{S}_I(\check{c}[\ell])[k, l + \{\mathcal{I}_l^{(H)}\}_{\ell}, \ell] \quad (110)$$

a

$$\mathcal{S}_B(\check{c})[k, l] = \left( \bigotimes_{\mathcal{N}_{IO}^{(H)}: \check{c}} \mathcal{S}(\mathcal{N}_{IO}^{(H)})[k, l] \right) \odot^{-1} \mathcal{S}_F(\check{c})[k, l], \quad (111)$$

kde  $\ell = 0, 1, 2$ . Iteraci systému uzavírá aktivace měkkých inverzí broadcasterů spodní vrstvy sítě. Tato část se skládá z dvojice kombinačních procesů (84) a (85). Aktualizované hustoty  $\{\mathcal{S}_B(\check{c})\}$  jsou následně zavedeny zpět na mezivrstevové porty vrchní vrstvy a struktura je připravena k vykonání další iterace ze stanoveného počtu  $N_I$ .

Za předpokladu, že iterativní dekodovací síť na Obr. 21a realizuje detekci vstupních dat ve dvojkové aritmetice po jejich průchodu kernelem (33), pak vnitřní uspořádání SISO modulů na spodní vrstvě je zcela totožné s příkladem na Obr. 12 z odstavce 5.2.1. Rozdíl je pouze v označení výstupní proměnné.

Pokud dekomponovatelná odezva ISI kanálu (30) obsahuje ve svých komponentách  $\mathcal{G}^{(V)}$  resp.  $\mathcal{G}^{(H)}$  dominantní koeficient, můžeme provést zjednodušení vrstevové sítě s variabilní topologií v souladu se závěry z odstavce 5.3.3. Získáme tak strukturu s topologií na Obr. 22a. Kompozitní SISO moduly na spodní resp. vrchní vrstvě takového detektoru jsou znázorněné na Obr. 22b resp. Obr. 22c. Aktivační proces je stejný jako u výše uvedeného zapojení s plnou složitostí.

Výhodou vrstvených detektorů je především redukce výpočetní náročnosti. Aplikací standardní (nezjednodušené) variabilní topologie nebo fixní topologie na ukázkový dekomponovatelný kernel  $\mathcal{G}_{3 \times 3}^{(HV)}$  bychom dospěli ke kombinační tabulce SISO modulu o mohutnosti  $\text{card}(\mathcal{A}_{IO}) = 2^9 = 512$  za předpokladu, že vstupní data jsou binární. U vrstvené topologie je složitost na spodní vrstvě  $\text{card}(\mathcal{A}_{IO}^{(H)}) = 2^3 = 8$  a na vrchní vrstvě  $\text{card}(\mathcal{A}_{IO}^{(V)}) = 6^3 = 216$ . V součtu tedy výrazně méně. Odlehčení výpočetní náročnosti má opět vliv na nižší výkonnost. Zjednodušeně řečeno, výkonnost struktury klesá s množstvím marginalizovaných veličin. U vrstvené struktury se vyskytuje jedna marginalizace navíc, která produkuje “mezi-vrstevové” metriky. Další nevýhodou je nízká flexibilita a možná aplikovatelnost pouze na specifické kernely. Také musíme počítat s vyšší mohutností stavových proměnných ve vrchní vrstvě detektoru a s tím spojené vyšší paměťové nároky na ukládání jejich měkkých metrik. I tak se jedná o velmi zajímavé uspořádání výrazně redukující výpočetní náročnost s mírným snížením výkonnosti při porovnání s klasickými jednovrstevovými variabilními nebo fixními topologiemi.

## 5.4 Front-end iterativní dekodovací síť

Front-end iterativní dekodovací síť provádí prosté zobrazení

$$r[k, l] \mapsto \{\mathcal{S}_F(q^{(i)})[k, l]\}_i \quad (112)$$

vstupní observace detektoru do množiny hustot. Charakter tohoto zobrazení závisí především na distribuci šumů  $w[k, l]$  v náhodném ML-IECS kanále. Dále na převodní funkci  $f_s(\cdot)$  kanálového front-endu a doméně implementace samotného iterativního detektoru. Lepší představu o tomto zobrazení můžeme získat prostřednictvím příkladů na Obr. 25. Zde jsou vyobrazeny transformační funkce front-endů některých detektorů, které budou představeny a testovány v experimentální části práce. Zobrazení jsou uvedena pro obě možné domény implementace a každá modrá křivka reprezentuje konturu určité hustoty  $\{\mathcal{S}_F(q^{(i)})[k, l]\}_i$ , která je výstupu front-endu vygenerována na základě určitého buzení  $r[k, l]$ .

## 5.5 Rozhodovací blok a tvrdý odhad

Iterativní dekodovací síť pracuje pouze měkkými informacemi v podobě vstupních  $\{\mathcal{S}_I\}$  nebo výstupních  $\{\mathcal{S}_O\}$  resp. dopředných  $\{\mathcal{S}_F\}$  nebo zpětných  $\{\mathcal{S}_B\}$  hustot. Pokud chceme stanovit hodnotu tvrdého odhadu  $\hat{\mathcal{V}}_{IO}[\ell]$  některé náhodné veličiny  $\mathcal{V}_{IO}[\ell]$ , vyskytující se ve struktuře iterativního detektoru, provedeme nejprve kombinování  $\odot$  příslušné dvojice hustot do sdružené hustoty  $\{\mathcal{S}\}$ . Vzniklou sdruženou hustotu následně prahujeme podle vztahu (53). Uvedenou operaci provádí tzv. rozhodovací bloky detektoru. U většiny obsažených proměnných však není zapotřebí znát hodnoty jejich tvrdých odhadů. Především se jedná o pomocné a stavové proměnné. Jediné veličiny u nichž je nezbytné stanovit jejich tvrdé odhady jsou data  $\hat{d}[k, l]$  na výstupu iterativního detektoru a to prostřednictvím vztahu

$$\hat{d}[k, l] = \arg \underset{\check{d}}{\mathbb{M}} (\mathcal{S}_F(\check{d})[k, l] \odot \mathcal{S}_B(\check{d})[k, l]). \quad (113)$$

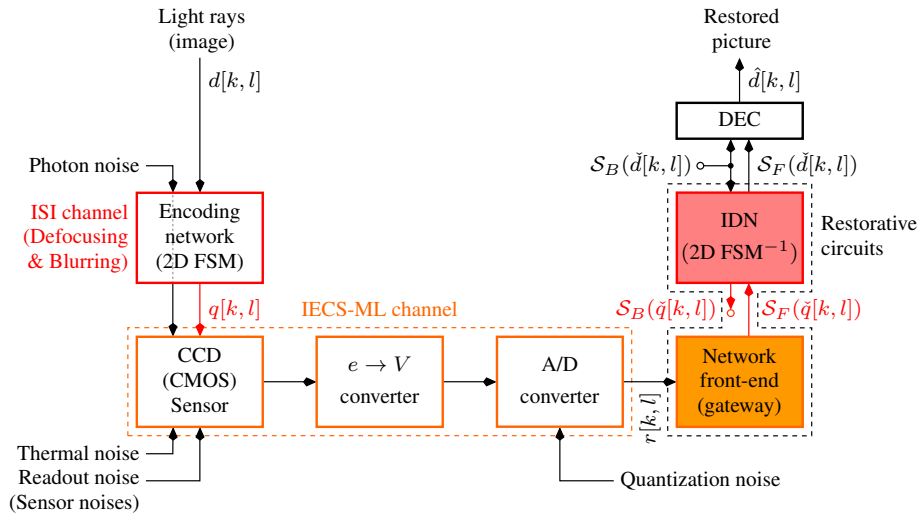
Rozhodovací operace (113) je provedena pro veškerá data nejpozději po proběhnutí maximálního zvoleného počtu iterací  $N_I$  a nebo dříve za předpokladu, že iterativní detektor dokonvergoval do stabilního stavu a hustoty  $\{\mathcal{S}_F(\check{d})[k, l]\}$  se přestaly měnit.

## 6 Aplikace v oblasti rekonstrukce černobílých snímků

Účinnost některých topologií iterativních detektorů budeme demonstrovat na aplikaci detekce binárních obrazů získaných z monochromatické nebo barevné kamery se CCD senzorem. Tato úloha, označovaná jako binarizace, spočívá v rekonstrukci obrazů, které jsou ve své podstatě binární, zpět do této dichromatické formy po jejich průchodu optickým systémem kamery. Jedná se o první a důležitý krok v analýze obrazu [30], protože binární formy obrazu využívá řada aplikací. Toto “prahování” můžeme chápat jako obnovu obrazu do ideální stavu, kterého je schopna iterativní dekodovací síť docílit díky svým schopnostem dekonvoluce a potlačování šumů. Algoritmus provádí rozhodnutí, zda-li každý jednotlivý pixel vstupního obrazu byl původně bílý nebo černý a to na základě prahování aposteriorních hustot na výstupu sítě. Dále je proveden experiment o schopnosti sítě dopočítat chybějící informaci při podvzorkování obrazu, který otevírá možnost budoucí aplikace iterativních dekodovacích sítí také v oblasti interpolace obrazu. Iterativní dekodovací síť by tak tvořila jediný a úplný systém, určený k celkové rekonstrukci obrazu.

### 6.1 Model systému

Pro účely demonstrace binarizační úlohy je použit matematický model CCD kamery na Obr. 23, který simuluje náhodné realizace šumů CCD senzoru a optickou část aproximuje pomocí ISI kanálu. Uvedený model plně vychází z obecného referenčního modelu na Obr. 2. Vstup modelu tvoří určitý binární obraz, reprezentovaný daty  $d[k, l]$  ve dvojkové aritmetice. Následuje blok simulující degradaci vstupního signálu při jeho průchodu optickou částí systému. Tato část je realizována prostřednictvím libovolné kódovací sítě z kapitoly 3. Další bloky z kaskády přidávají do konvoluovaného signálu specifické šумы CCD senzoru a kvantizační šum A/D převodníku kamery. Matematický popis šumového modelu je odvozen v odstavci 6.1.2. Výstupem modelu je náhodná realizace šedotónového obrazu  $r[k, l]$ , která je předložena iterativnímu detektoru aby z ní extrahoval odhad původního binární obrazu  $\hat{d}[k, l]$ .



Obr. 23. Blokové schéma CCD kamery napojené na rekonstrukční 2D iterativní detektor.



### 6.1.1 Rozostření snímku a rozmazání snímku pohybem

Experimentální část pracuje se dvěma základními modely degradace vstupního obrazu po průchodu ISI kanálem. Jednak s modelem rozostření a dále s modelem rozmazání obrazu. Rozostření vzniká při průchodu obrazu reálným objektivem kamery. K rozmazání obrazu dochází vlivem pohybu kamery, či samotného snímaného objektu. O matematických modelech těchto rušivých vlivů pojednávají odstavce 6.2.1 a 6.3.1. Výsledky eliminace obou těchto druhů degradace pomocí aplikace dvoudimenzionálních iterativních detektorů uvádí odstavce 6.2.2 a 6.3.2.

### 6.1.2 Šumový model snímacího senzoru

Náhodné rušivé vlivy, v podobě aditivních šumů, vnáší do signálu sensor kamery. Základní šumový model senzoru zahrnuje tři signifikantní zdroje šumu. Jedná se termický šum (Thermal noise)  $\mathbf{W}_T$ , čtecí šum (Readout noise)  $\mathbf{W}_R$  a kvantizační šum (Quantization noise)  $\mathbf{W}_C$  v A/D převodníku na výstupu senzoru [19]. Termický šum vzniká spontánní generací elektronů v potenciálové jámě snímací buňky senzoru vlivem teploty a statisticky ho popisuje Poissonovo rozdělení. Čtecí šum má rozdělení Normální s nulovou střední hodnotou a vzniká v průběhu vyčítání nábojových balíků, prostřednictvím pohybujícího se potenciálového reliéfu, do sběrného registru po uzavření clony objektivu. Čím je tato rychlost vyšší tím je vyšší směrodatná odchylka čtecího šumu. Signál na výstupu senzoru tedy získáme jako superpozici

$$\begin{aligned} \mathbf{R}_C &= \frac{N_e^{(ET)} \mathbf{Q} + \mathbf{W}_T + \mathbf{W}_R}{\Delta_C} \\ &= \mathbf{S} + \frac{\mathbf{W}_T + \mathbf{W}_R}{\Delta_C}, \end{aligned} \quad (114)$$

kde

$$\Delta_C = \left\lfloor \frac{N_e^{(FWC)}}{2^{N_B}} \right\rfloor \quad (115)$$

značí počet elektronů na jednu kvantovací úroveň převodníku a  $N_e^{(ET)}$  je počet elektronů nagenovaných v potenciálové jámě při maximálním ozáření senzoru ( $\mathbf{Q} = \mathbf{1}$ ) v průběhu zvolené expoziční doby  $T_E$ . Expoziční dobu předpokládáme takovou, která nepřivede sensor do saturace a platí, že  $N_e^{(ET)} < N_e^{(FWC)}$ . Zde  $N_e^{(FWC)}$  určuje maximální počet elektronů, který se může v buňce nagenovat (Full Well Capacity). Signál na výstupu senzoru je dále kvantován

$$\begin{aligned} \mathbf{R} &= \mathbf{R}_C + \mathbf{W}_C \\ &= \lceil \mathbf{R}_C \rceil, \end{aligned} \quad (116)$$

pomocí zaokrouhlení  $\lceil \cdot \rceil$  na nejbližší celočíselnou úroveň.

Front-end detektoru zpracovává signál  $\mathbf{R}$  na základně znalosti statistických vlastností šumu. Distribuce jednotlivých zdrojů jsou známé, včetně jejich parametrů. Poissonovské rozdělení

$$\Pr_w^{(T)}[n](\mu_T) = \begin{cases} 0, & \mu_T < 40 \text{ and } n \geq 70 \\ e^{-\mu_T} \frac{\mu_T^n}{n!}, & \mu_T < 40 \text{ and } n < 70 \\ \Pr_w^{(Cut)}[n, N_e^{(FWC)}](\mu_T, \sqrt{\mu_T}), & \mu_T \geq 40 \end{cases} \quad (117)$$

termického šumu  $w_T[k, l] \sim \Pr_w^{(T)}(\mu_T)[n]$  můžeme za normálních okolností aproximovat Normálním rozdělením se střední hodnotou  $\mu_T$  a směrodatnou odchylkou  $\sqrt{\mu_T}$ . Střední hodnotu temného proudu  $\mu_T$  budeme uvažovat v desítkách elektronů, což odpovídá běžným podmínkám (teplotám) provozu senzoru. Aproximace je

optimální pro hodnoty  $\mu_T \geq 40$ . V následujících experimentech použijeme jako referenční hodnotu průměrného temného proudu 60 [e]. Čtecí šum má střední hodnotu nulovou a směrodatnou odchylku  $\sigma_R$  lze určit z rychlosti potenciálového reliéfu. Součet obou zdrojů tedy vytvoří Gaussovský aditivní šum s hustotou

$$p_w(\xi, \mu_w, \sigma_w) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\xi - \mu_w)^2}{2\sigma_w^2}\right). \quad (118)$$

Toto spojité rozdělení budeme integrovat přes jednotlivé kvantizační kroky od 0 až do kapacity potenciálové jámy. Získáme tak “useknuté” diskrétní rozdělení

$$\Pr_w^{(\text{Cut})}(\mu, \sigma)[n, N] = \begin{cases} \int_{-\infty}^{\frac{1}{2}} p_w(\xi, \mu, \sigma) d\xi, & n = 0 \\ \int_{n-\frac{1}{2}}^{n+\frac{1}{2}} p_w(\xi, \mu, \sigma) d\xi, & 0 < n < N \\ \int_{N-\frac{1}{2}}^{\infty} p_w(\xi, \mu, \sigma) d\xi, & n = N \end{cases} \quad (119)$$

Vypočtením obsažených integrálů získáme finální verzi useknutého diskrétní rozdělení

$$\Pr_w^{(\text{Cut})}(\mu, \sigma)[n, N] = \begin{cases} \frac{1}{2} \operatorname{erfc}\left(\frac{2\mu-1}{\sqrt{8\sigma}}\right), & n = 0 \\ \frac{1}{2} \left( \operatorname{erf}\left(\frac{2(n-\mu)+1}{\sqrt{8\sigma}}\right) - \operatorname{erf}\left(\frac{2(n-\mu)-1}{\sqrt{8\sigma}}\right) \right), & 0 < n < N \\ \frac{1}{2} \operatorname{erfc}\left(\frac{2(N-\mu)-1}{\sqrt{8\sigma}}\right), & n = N \end{cases} \quad (120)$$

Uvedený vztah představuje diskrétní rozdělení čtecího šumu  $\Pr_w^{(\text{R})}(\mu, \sigma_R)[n] = \Pr_w^{(\text{Cut})}(\mu, \sigma_R)[n, N_e^{(\text{FWC})}]$  v počtech elektronů a to na intervalu od 0 do kapacity buněk senzoru  $N_e^{(\text{FWC})}$ . Pokud by byl přítomen pouze čtecí šum, pocházel by kompozitní signál na výstupu senzoru  $N_e^{(\text{ET})}q[k, l] + w_R[k, l]$  z diskrétního rozdělení  $\Pr_w^{(\text{R})}(N_e^{(\text{ET})}q[k, l], \sigma_R)[n]$ . Střední hodnotu uvedeného rozdělení tvoří užitečná část signálu v počtu elektronů  $N_e^{(\text{ET})}q[k, l]$  a tato část se nagenovala v  $[k, l]$ -té buňce senzoru společně se čtecím šumem  $w_R[k, l]$ . Jelikož je přítomen v senzoru ještě termální šum, je nezbytné provést jeho začlenění. Výsledné rozdělení součtu obou rušivých vlivů udává obecná konvoluce

$$\Pr_w(\mu, \mu_T, \sigma_R)[n] = \sum_{n'=0}^{N_e^{(\text{FWC})}} \Pr_w^{(\text{T})}[n'](\mu_T) \Pr_w^{(\text{R})}[n - n'](\mu, \sigma_R). \quad (121)$$

Dosažením střední hodnoty, v podobě užitečného signálu, a odhadů střední hodnoty termálního šumu  $\hat{\mu}_T$  resp. směrodatné odchylky čtecího šumu  $\hat{\sigma}_R$  získáme finální obecný tvar transformační funkce front-endu iterativní dekodovací sítě

$$\{\mathcal{P}_F(q^{(i)})[k, l]\}_i = \left\{ \sum_{|n\Delta_c^{-1} - r[k, l]| \leq \frac{1}{2}} \Pr_w[n](N_e^{(\text{ET})}q^{(i)}, \hat{\mu}_T, \hat{\sigma}_R) \right\}_i, \quad (122)$$

kde odhady parametrů  $\hat{\mu}_T$  a  $\hat{\sigma}_R$  získáme z teploty senzoru a rychlosti vyčítání nahromaděného náboje.

## 6.2 Potlačení rozostření a šumu senzoru

V této kapitole jsou prezentovány výsledky binarizace obrazu za předpokladu, že obraz je rozostřen po průchodu optickou částí kamery a následně kontaminován šumy CCD senzoru a kvantizačním šumem v A/D převodníku na výstupu senzoru. Bude ukázáno několik příkladů rekonstrukce binárního obrazu prostřednictvím vybraných topologií iterativní dekodovací sítě v odstavci 6.2.2. Následně budou všechny topologie podrobeny statistické analýze výkonnosti, jejíž výsledky uvádí odstavce 6.2.3.

### 6.2.1 Kernel kanálu modelující rozostření

V této ukázkové aplikaci je ISI kanálem objektiv kamery. Reálný objektiv má nenulovou aperturu s reálnou čočkou a tedy omezenou hloubku ostrosti. V apertuře objektivu dochází k difrakci světla a následné interferenci dopadajících paprsků v rovině senzoru kamery. V případě systémů s ideální kruhovou čočkou popisuje interferenci tzv. Airyho disk [31]. U běžné kamery s reálnou čočkou je vhodnější a jednodušší modelovat rozostření v rovině snímáče pomocí Gaussovské PSF (Point Spread Function) [33]

$$\text{PSF}_{\Delta}(x, y) = \frac{\Delta}{\pi} e^{-\Delta(x^2+y^2)}, \quad (123)$$

kde parametr  $\Delta$  určuje šířku laloku. Model zanedbává ztráty v čočce a proto

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \text{PSF}_{\Delta}(x, y) dx dy = 1. \quad (124)$$

Snímací buňky senzoru budeme předpokládat jako čtvercové, bez mezer mezi nimi a s normovanou délkou 1. Dílčí množství dopadajícího záření na buňky senzoru získáme integrací PSF přes příslušné čtvercové oblasti jednotkového obsahu. Tak obdržíme Gaussův kernel

$$\mathcal{G}_{L \times L, \Delta}^{(\text{Gauss})} = \begin{pmatrix} \ddots & & \vdots & & \ddots \\ & g'' & g' & g'' & \\ & g' & g & g' & \\ & g'' & g' & g'' & \\ \ddots & & \vdots & & \ddots \end{pmatrix} \quad (125)$$

s koeficienty

$$\begin{aligned} \{\mathcal{G}_{L \times L, \Delta}^{(\text{Gauss})}\}_{k,l} &= \int_l^{l+1} \int_k^{k+1} \text{PSF}_{\Delta} \left( x - \frac{1}{2}, y - \frac{1}{2} \right) dx dy \\ &= \frac{1}{4} \left( \text{erf} \left( \left( \frac{1}{2} - k \right) \sqrt{\Delta} \right) + \text{erf} \left( \left( \frac{1}{2} + k \right) \sqrt{\Delta} \right) \right) \\ &\quad \times \left( \text{erf} \left( \left( \frac{1}{2} - l \right) \sqrt{\Delta} \right) + \text{erf} \left( \left( \frac{1}{2} + l \right) \sqrt{\Delta} \right) \right), \end{aligned} \quad (126)$$

kde  $L$  značí jeho šířku. Šířka kernelu tedy určuje počet sousedících pixelů, které se rozostřením vzájemně ovlivňují. Mezi potlačením hlavního svazku a šířkou laloku PSF platí relace

$$\{\mathcal{G}_{L \times L, \Delta}^{(\text{Gauss})}\}_{0,0} = \text{erf} \left( \frac{\sqrt{\Delta}}{2} \right)^2. \quad (127)$$

Šířky kernelu  $L$  a laloku  $\Delta$  společně reflektují kvalitu zaostření a také kvalitu samotného objektivu.

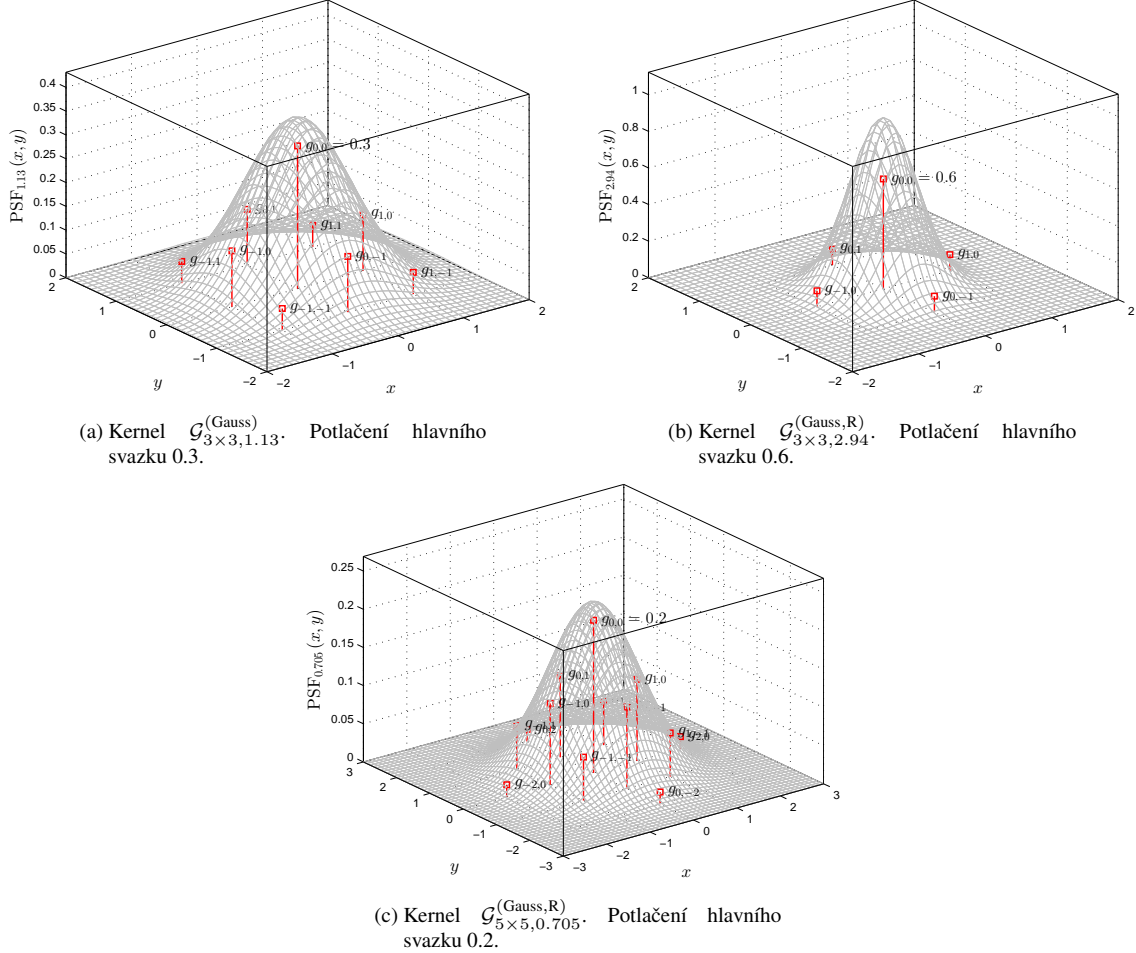
Tvary Gaussova kernelu demonstrujeme na třech konkrétních příkladech s šířkami laloku 1.13, 2.94, a 0.705, které korespondují s potlačením hlavního svazku 0.3, 0.6 a 0.2. Při těchto výchozích parametrech vzniknou konvoluční jádra

$$\mathcal{G}_{3 \times 3, 1.13}^{(\text{Gauss})} = \begin{pmatrix} 0.0458 & 0.1172 & 0.0458 \\ 0.1172 & 0.3000 & 0.1172 \\ 0.0458 & 0.1172 & 0.0458 \end{pmatrix}, \quad \mathcal{G}_{3 \times 3, 2.94}^{(\text{Gauss,R})} = \begin{pmatrix} & 0.0872 & \\ 0.0872 & 0.6000 & 0.0872 \\ & 0.0872 & \end{pmatrix} \quad (128)$$

a

$$\mathcal{G}_{5 \times 5, 0.705}^{(\text{Gauss}, \text{R})} = \left\{ \begin{array}{ccccc} & & 0.0161 & & \\ & 0.0571 & 0.1068 & 0.0571 & \\ 0.0161 & 0.1068 & 0.2000 & 0.1068 & 0.0161 \\ & 0.0571 & 0.1068 & 0.0571 & \\ & & 0.0161 & & \end{array} \right\}. \quad (129)$$

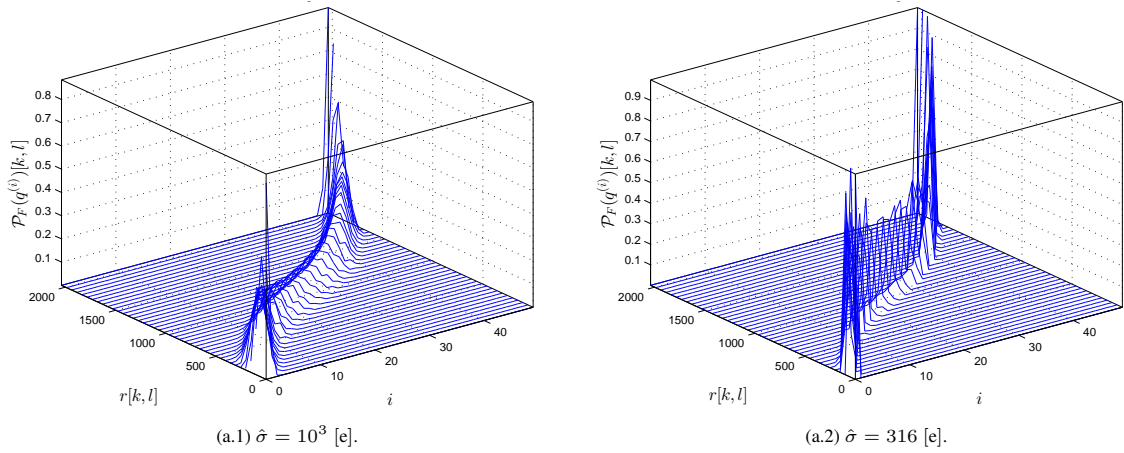
První množina reprezentuje 9-ti paprskový čtvercový kernel. Druhá resp. třetí množina představuje 5-ti resp. 13-ti paprskový kernel s potlačenými postranními (nevýznamnými) paprsky. Všechny uvedené příklady znázorňuje Obr. 24 společně s výchozími PSF.



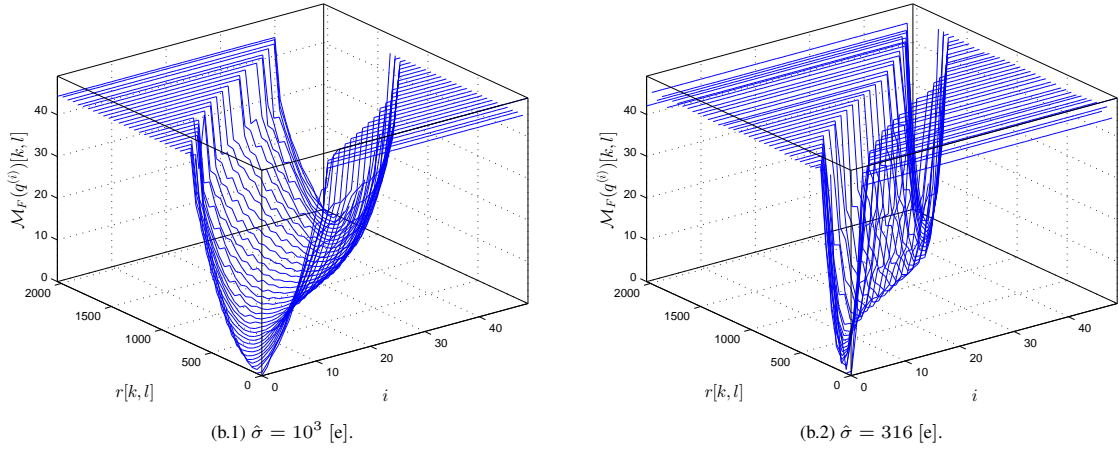
Obr. 24. Příklady Gaussova defokusačního kernelu společně s výchozí PSF.

## 6.2.2 Příklady rekonstrukcí černobílých snímků iterativními sítěmi při znalosti kernelu kanálu

Ukázky binarizace obrazu provedeme na dvojici modelů rozostření. Jednak na úplném defokusačním kernelu  $\mathcal{G}_{3 \times 3, 1.13}^{(\text{Gauss})}$  o velikosti  $3 \times 3$  s přenosem hlavního svazku 0.3 a také na redukovaném kernelu  $\mathcal{G}_{5 \times 5, 0.705}^{(\text{Gauss}, \text{R})}$  o velikosti  $5 \times 5$  s potlačenými postranními svazky a s přenosem hlavního svazku 0.2. Předpokládáme, že vlastnosti optické části systému jsou známy na straně detekce a detektor pracuje s perfektní informací CSI (Channel State Information) o uspořádání kernelu. Parametry senzoru budeme ve všech případech uvažovat  $N_e^{(\text{FWC})} = 30 \times 10^3$  [e],  $N_e^{(\text{ET})} = 29 \times 10^3$  [e],  $\Delta_C = 14$  [e],  $N_B = 11$  a  $\hat{\mu}_T = 60$  [e]. Tyto parametry, kromě úrovní šumů, byly převzaty



(a) Pravděpodobnostní doména.



(b) Metrická doména.

Obr. 25. Příklady transformačních funkcí front-endu detektoru (gateway). Parametry modelu objektivu: Gaussovský kernel  $3 \times 3$ ,  $\Delta = 1.13$ . Úrovně šumu  $\hat{\sigma} = 10^3$  [e] a  $\hat{\sigma} = 316$  [e].

ze senzoru iXon3 885 od firmy Andor Technology a jsou tedy parametry reálnými. Pro zvolené  $N_e^{(ET)}$  je senzor vybuděn na 96.5 %. Jako nezávislou proměnnou ponecháme  $\hat{\sigma}_R$ , která determinuje směrodatnou odchylku  $\hat{\sigma} = \sqrt{\hat{\mu}_T} + \hat{\sigma}_R$  výsledného Gaussova šumu. Přenosové funkce front-endu

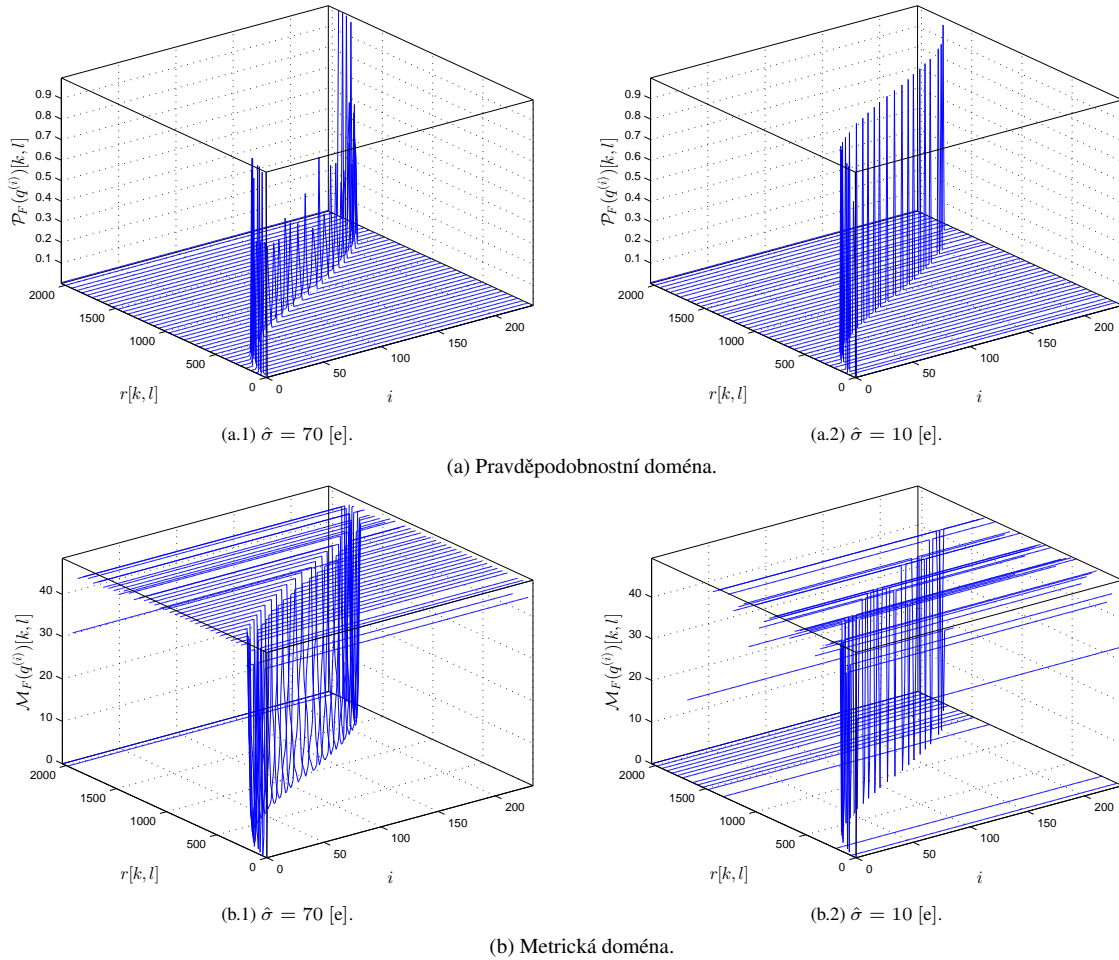
$$\{\mathcal{P}_F(q^{(i)})[k, l]\}_i = \left\{ \Pr_w^{(\text{Cut})} \left( \frac{\hat{\mu}_T + N_e^{(ET)} q^{(i)}}{\Delta_C}, \frac{\sqrt{\hat{\mu}_T} + \hat{\sigma}_R}{\Delta_C} \right) [r[k, l], 2^{N_B} - 1] \right\}_i \quad (130)$$

resp.

$$\{\mathcal{M}_F(q^{(i)})[k, l]\}_i = \left\{ -\ln \left( \Pr_w^{(\text{Cut})} \left( \frac{\hat{\mu}_T + N_e^{(ET)} q^{(i)}}{\Delta_C}, \frac{\sqrt{\hat{\mu}_T} + \hat{\sigma}_R}{\Delta_C} \right) [r[k, l], 2^{N_B} - 1] \right) \right\}_i \quad (131)$$

vycházejí z předpokladu, že  $\hat{\mu}_T > 40$  a Poissonovo rozdělení lze úspěšně aproximovat Gaussovským diskretním rozdělením. Výpočet konvoluce (121) se tak redukuje na výše uvedený tvar (130) resp. (131), kdy sečteme střední hodnoty a směrodatné odchylky dvou nezávislých diskretních veličin s normálním rozdělením. Šumový model se zjednoduší do tvaru  $\mathbf{R} = [\mathbf{R}_C + \mathbf{W}]$ , kde  $\mathbf{W}$  je realizace kompozitního Gaussova šumu se střední hodnotou  $\hat{\mu}_T \Delta_C^{-1}$  a směrodatnou odchylkou  $(\sqrt{\hat{\mu}_T} + \hat{\sigma}_R) \Delta_C^{-1}$ . Tvary funkcí (130) a (131) při výše zvolených parametrech senzoru a míry jeho vybudění v kombinaci s prvním modelem kernelu (128) znázorňuje Obr. 25.

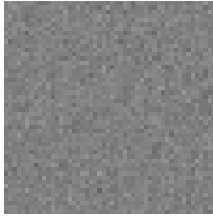
Pro druhý model (129) znázorňuje přenosové funkce Obr. 26. Mezi oběma příklady přenosových funkcí panuje značná podobnost z hlediska typického prohnutí ve tvaru písmene S, které vyplývá z tvaru Gaussova defokusačního kernelu.



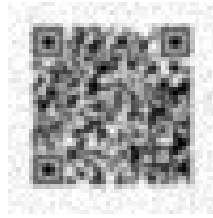
Obr. 26. Příklady transformačních funkcí front-endu detektoru (gateway). Parametry modelu objektivu: Gaussovský kernel  $5 \times 5$ ,  $\Delta = 0.705$ . Úroveň šumu  $\hat{\sigma} = 70$  [e] a  $\hat{\sigma} = 10$  [e].



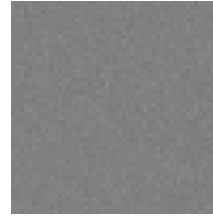
(a) Skutečný binární obraz.  $\mathbf{D}$ .



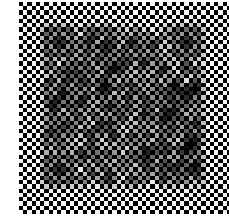
(b.1) Realizace aditivního šumu  $\mathbf{W}$ .



(b.2) Vstupní šedotónový obraz  $\mathbf{R}$ .



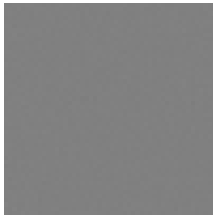
(c.1) Realizace aditivního šumu  $\mathbf{W}$ .



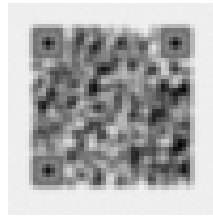
(c.2) Vstupní šedotónový obraz  $\mathbf{R}$ .

(b) Monochromatická kamera. Model objektivu: Gaussovský kernel  $3 \times 3$ ,  $\Delta = 1.13$ . Model sensoru:  $\hat{\sigma} = 10^3$  [e].

(c) Barevná kamera s Bayerovským CFA. Model objektivu: Gaussovský kernel  $3 \times 3$ ,  $\Delta = 1.13$ . Model sensoru:  $\hat{\sigma} = 316$  [e].



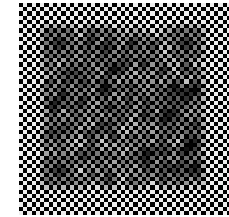
(d.1) Realizace aditivního šumu  $\mathbf{W}$ .



(d.2) Vstupní šedotónový obraz  $\mathbf{R}$ .



(e.1) Realizace aditivního šumu  $\mathbf{W}$ .



(e.2) Vstupní šedotónový obraz  $\mathbf{R}$ .

(d) Monochromatická kamera. Model objektivu: Gaussovský kernel  $5 \times 5$ ,  $\Delta = 0.705$ . Model sensoru:  $\hat{\sigma} = 70$  [e].

(e) Barevná kamera s Bayerovským CFA. Model objektivu: Gaussovský kernel  $5 \times 5$ ,  $\Delta = 0.705$ . Model sensoru:  $\hat{\sigma} = 10$  [e].

Obr. 27. Ukázky realizací vstupních obrazů detektoru  $\mathbf{R}$  pro různé úrovně šumu a rozostření.

Na Obr. 27 můžeme vidět realizace ukázkového obrazu (QR kódu) na výstupu A/D převodníku kamery pro zvolená rozostření a parametry sensoru. Ve dvou případech se jedná o výstup monochromatické kamery a další dva výstupy jsou získané z modelu kamery s Bayerovským polem. Dále Obr. 27 znázorňuje realizace aditivního šumu CCD sensoru na neutrálním šedém pozadí. Na Obr. 28 a Obr. 29 jsou rekonstruované (binarizované) QR kódy na výstupu detektoru po proběhnutí 1., 3. a 5. iterace společně s příslušnými chybovostmi detekce a to za předpokladu, že vstupní obraz  $\mathbf{R}$  byl získán v plném rozlišení z modelu monochromatické kamery při použití kernelu  $3 \times 3$ . Na Obr. 30 a Obr. 31 je znázorněn binarizovaný výstup detektoru za stejných podmínek pouze s tím rozdílem, že vstupní obraz  $\mathbf{R}$  byl získán v polovičním rozlišení (zelený kanál) pomocí simulace barevné kamery s Bayerovským CFA a binarizační úloha proto zahrnuje i de-mosaicing. V případě redukovaného kernelu  $5 \times 5$  jsme použili dekodovací síť s marginalizací na úrovni pixelů. Z uvedených výsledků na Obr. 32 a Obr. 33 je patrné, že rekonstruované obrazy jsou prakticky bez chyby už po 5. iteraci. Pro větší kernely musí být úroveň šumu podstatně nižší, jinak detekce začne selhávat. Toto způsobuje relativně nízká bitová šíře A/D převodníku. Kvantovací krok  $\Delta_C$  je v tomto případě příliš hrubý a dekonvoluční úloha se stane nejednoznačnou díky větší mohutnosti abecedy  $\mathcal{A}_q$ . Pokud jsou šumové podmínky, u většího kernelu, horší je nutné použít převodník s jemnějším kvantovacím rastrem, například 16-ti bitový. Totéž platí pokud je detekční úloha rozšířena o de-mosaicing. Pak detektor nemá k dispozici každou druhou vstupní metriku  $\mathcal{P}_F$ . Chybějící metriky jsou nahrazeny uniformními metrikami. Tím detektoru sdělíme, že v daném místě nemáme žádnou informaci o signále  $r[k, l]$ .

Příslušný SISO modul je tak plně odkázán na informační pomoc svých sousedů, což činí detekci extrémně komplikovanou a vysoce nejednoznačnou. Přesto je detektor schopen úspěšně obraz rekonstruovat, nicméně úroveň šumu musí být opět výrazně nižší než u snímání v plném rastru. Pokud to není splněno zaznamenáme v průběhu iterací oscilační charakter, kdy sousedící moduly nejsou schopny se dohodnout na podobě odhadu.



(a.1)  $I = 1$ , BER = 0.348



(a.2)  $I = 3$ , BER = 0.014



(a.3)  $I = 5$ , BER = 0.006



(a.4)  $I = 7$ , BER = 0.006

(a) Iterativní dekodovací síť marginalizující na úrovni bloku symbolů s rozšířenou topologií.



(b.1)  $I = 1$ , BER = 0.152



(b.2)  $I = 3$ , BER = 0.137



(b.3)  $I = 5$ , BER = 0.105



(b.4)  $I = 7$ , BER = 0.050

(b) Iterativní dekodovací síť marginalizující na úrovni symbolů.



(c.1)  $I = 1$ , BER = 0.388



(c.2)  $I = 3$ , BER = 0.036



(c.3)  $I = 5$ , BER = 0.001



(c.4)  $I = 7$ , BER = 0

(c) Vrstvová iterativní dekodovací síť marginalizující na úrovni bloku symbolů.

Obr. 28. Ukázky binarizovaných obrazů QR kódu  $\hat{\mathbf{D}}$  na výstupu různých druhů IDN využívajících Pd-SyD techniku. Vstupní šedotónový obraz  $\mathbf{R}$  byl simulován s plným rozlišením (monochromatická kamera). Parametry modelu objektivu: Gaussovský kernel  $3 \times 3$ ,  $\Delta = 1.13$ . Úroveň šumu  $\hat{\sigma} = 10^3$  [e].





(a.1)  $I = 1$ , BER = 0.397



(a.2)  $I = 3$ , BER = 0.005



(a.3)  $I = 5$ , BER = 0.002



(a.4)  $I = 7$ , BER = 0.002

(a) Iterativní dekodovací síť marginalizující na úrovni bloku symbolů s rozšířenou topologií.



(b.1)  $I = 1$ , BER = 0.216



(b.2)  $I = 3$ , BER = 0.081



(b.3)  $I = 5$ , BER = 0.075



(b.4)  $I = 7$ , BER = 0.056

(b) Iterativní dekodovací síť marginalizující na úrovni symbolů.



(c.1)  $I = 1$ , BER = 0.403



(c.2)  $I = 3$ , BER = 0.057



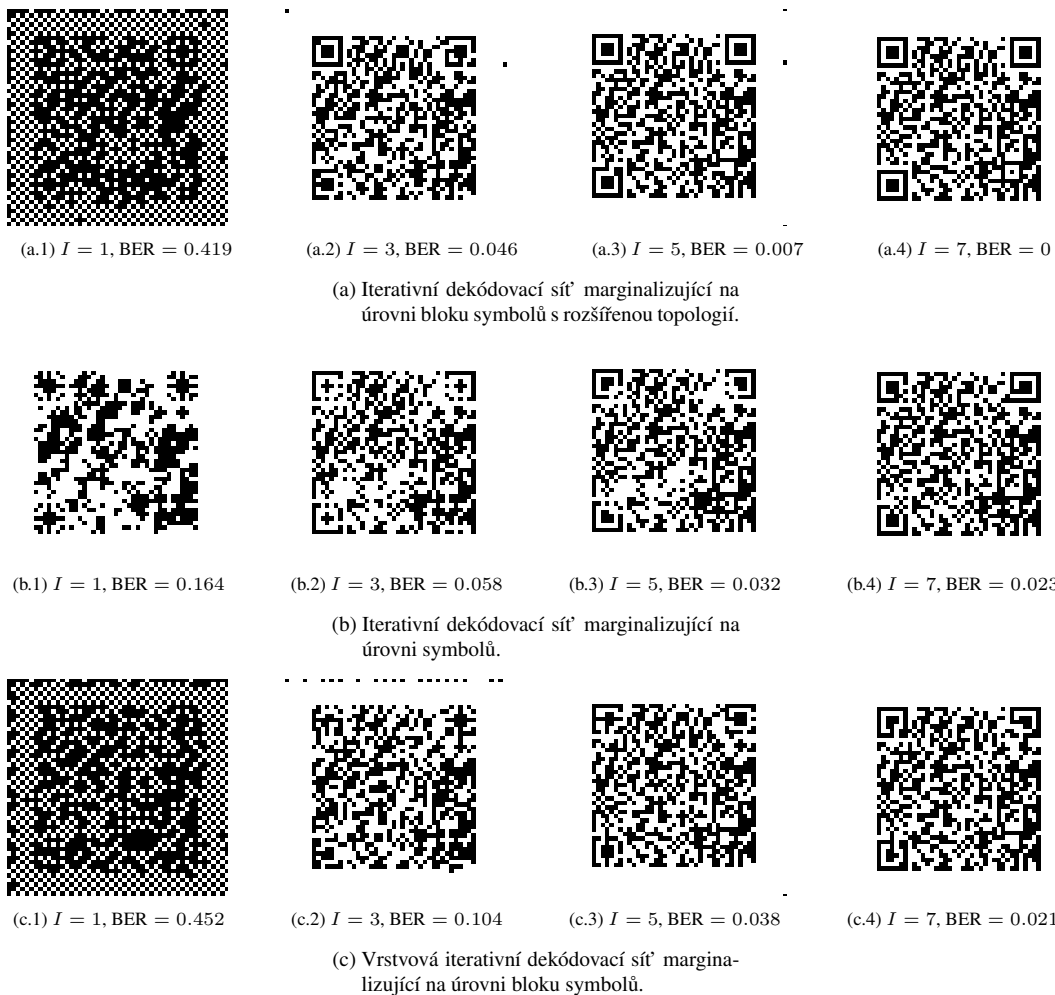
(c.3)  $I = 5$ , BER = 0.009



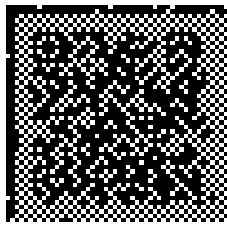
(c.4)  $I = 7$ , BER = 0.006

(c) Vrstvová iterativní dekodovací síť marginalizující na úrovni bloku symbolů.

Obr. 29. Ukázky binarizovaných obrazů QR kódu  $\hat{D}$  na výstupu různých druhů IDN využívajících Pd-PgD techniku. Vstupní šedotónový obraz  $R$  byl simulován s plným rozlišením (monochromatická kamera). Parametry modelu objektivu: Gaussovský kernel  $3 \times 3$ ,  $\Delta = 1.13$ . Úroveň šumu  $\hat{\sigma} = 10^3$  [e].



*Obr. 30.* Ukázky binarizovaných obrazů QR kódu  $\hat{D}$  na výstupu různých druhů IDN využívajících Pd-SyD techniku. Vstupní šedotónový obraz  $R$  byl simulován s polovičním rozlišením (barevná kamera). Parametry modelu objektivu: Gaussovský kernel  $3 \times 3$ ,  $\Delta = 1.13$ . Úroveň šumu  $\hat{\sigma} = 316$  [e].



(a.1)  $I = 1$ , BER = 0.432



(a.2)  $I = 3$ , BER = 0.079



(a.3)  $I = 5$ , BER = 0.014



(a.4)  $I = 7$ , BER = 0.002

(a) Iterativní dekodovací síť marginalizující na úrovni bloku symbolů s rozšířenou topologií.



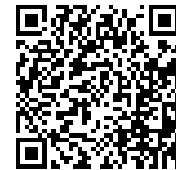
(b.1)  $I = 1$ , BER = 0.216



(b.2)  $I = 3$ , BER = 0.068

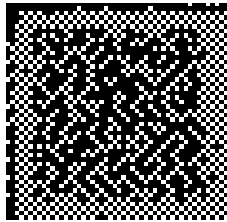


(b.3)  $I = 5$ , BER = 0.028



(b.4)  $I = 7$ , BER = 0.011

(b) Iterativní dekodovací síť marginalizující na úrovni symbolů.



(c.1)  $I = 1$ , BER = 0.475



(c.2)  $I = 3$ , BER = 0.162



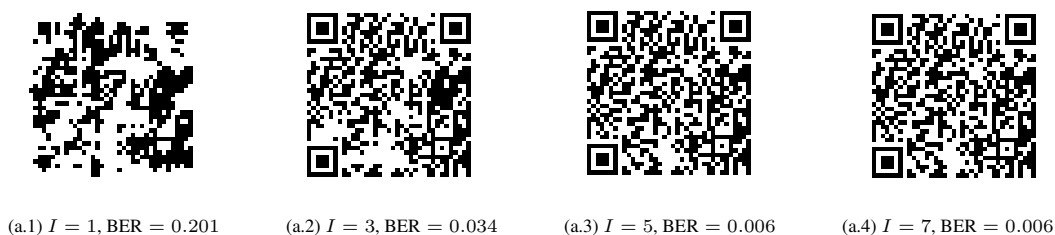
(c.3)  $I = 5$ , BER = 0.061



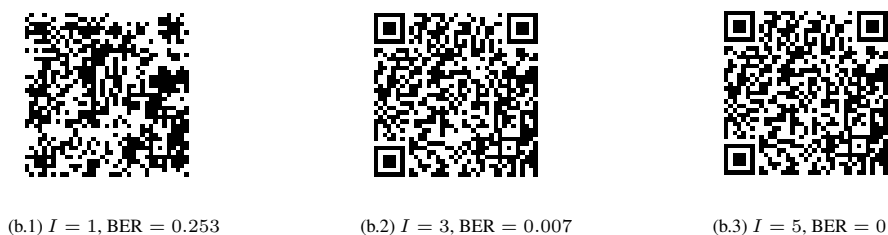
(c.4)  $I = 7$ , BER = 0.027

(c) Vrstvová iterativní dekodovací síť marginalizující na úrovni bloku symbolů.

Obr. 31. Ukázky binarizovaných obrazů QR kódu  $\hat{D}$  na výstupu různých druhů IDN využívajících Pd-PgD techniku. Vstupní šedotónový obraz  $R$  byl simulován s polovičním rozlišením (barevná kamera). Parametry modelu objektivu: Gaussovský kernel  $3 \times 3$ ,  $\Delta = 1.13$ . Úroveň šumu  $\hat{\sigma} = 316$  [e].



(a) Iterativní dekodovací síť využívající Pd-SyD techniku.

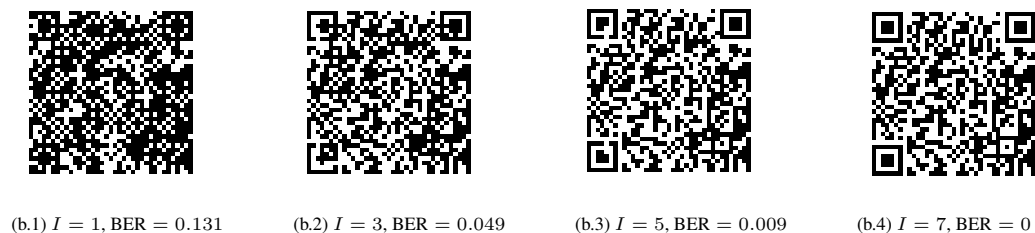


(b) Iterativní dekodovací síť využívající Pd-PgD techniku.

*Obr. 32.* Ukázky binarizovaných obrazů QR kódu  $\hat{\mathbf{D}}$  na výstupu různých druhů IDN marginalizujících na úrovni symbolů. Vstupní šedotónový obraz  $\mathbf{R}$  byl simulován s plným rozlišením (monochromatická kamera). Parametry modelu objektivu: Gaussovský kernel  $5 \times 5$ ,  $\Delta = 0.705$ . Úroveň šumu  $\hat{\sigma} = 70$  [e].



(a) Iterativní dekodovací síť využívající Pd-SyD techniku.



(b) Iterativní dekodovací síť využívající Pd-PgD techniku.

*Obr. 33.* Ukázky binarizovaných obrazů QR kódu  $\hat{\mathbf{D}}$  na výstupu různých druhů IDN marginalizujících na úrovni symbolů. Vstupní (šedotónový) obraz  $\mathbf{R}$  byl simulován s polovičním rozlišením (barevná kamera). Parametry modelu objektivu: Gaussovský kernel  $5 \times 5$ ,  $\Delta = 0.705$ . Úroveň šumu  $\hat{\sigma} = 10$  [e].

### 6.2.3 Analýzy výkonnosti

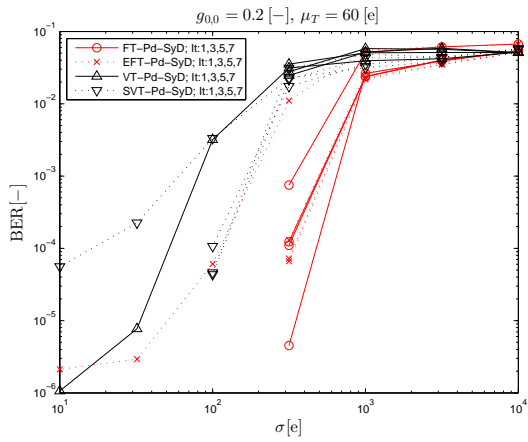
Abychom mohli objektivně vyhodnotit výkonnost jednotlivých topologií iterativních dekódovacích sítí, provedeme měření chybovosti detekovaných dat  $\hat{\mathbf{D}}$  na jejich výstupu a to pomocí metody Monte Carlo s náhodnými vstupními daty  $\mathbf{D}$  o velikosti  $64 \times 64$ . Analýzy provedeme při rozostření Gaussovským kanálem  $3 \times 3$  ve třech různých úrovních potlačení hlavního svazku  $g_{0,0}$  a to 0.2, 0.3 a 0.5. Parametry modelu senzoru jsou  $N_e^{(\text{FWC})} = 30 \times 10^3$  [e],  $N_e^{(\text{ET})} = 29 \times 10^3$  [e],  $\Delta_C = 14$  [e],  $N_B = 11$ ,  $\hat{\mu}_T = 60$  [e] a s proměnnou úrovní Gaussova šumu. Logaritmická škála směrodatných odchylek Gaussova šumu  $\hat{\sigma}$ , v množství elektronů, byla zvolena jako 10000, 3162, 1000, 316, 100, 32, 10. Každá průměrná hodnota chybovosti, pro jedno nastavení systému, byla získána průměrem ze 150 realizací. U fixních topologií byl použit rozklad konvolučního regionu na stavové proměnné, které znázorňuje Obr. 34. Výsledky analýz, v podobě BER křivek, jsou uvedeny na Obr. 35, Obr. 36, Obr. 37 a Obr. 38 pro monochromatickou kameru a na Obr. 39 a Obr. 40 pro barevnou kameru s Bayerovským polem.



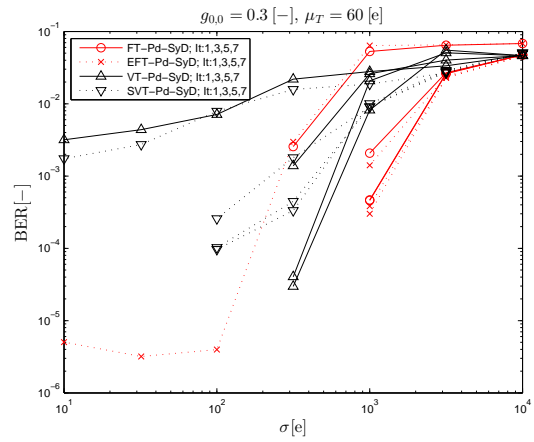
Obr. 34. Tvary stavových proměnných  $\mathcal{R}[k, l]$ ,  $\mathcal{C}[k, l]$  a  $\mathcal{B}[k, l]$  pro Gaussovský kernel  $\mathcal{G}_{3 \times 3, \Delta}^{(\text{Gauss})}$ .

### 6.3 Potlačení rozmazání pohybem a šumu senzoru

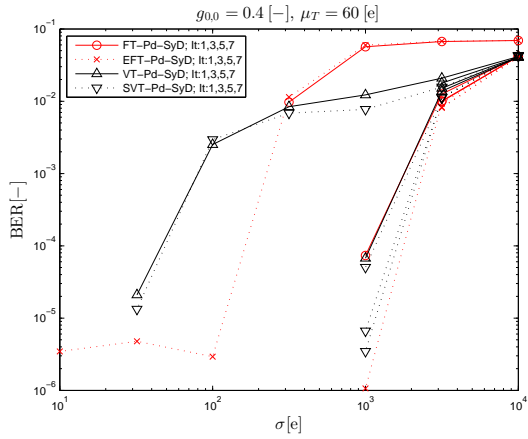
V této kapitole budeme demonstrovat binarizaci obrazu při jeho rozmazání pohybem. Optickou část kamery budeme předpokládat jako ideální, tedy s nekonečně velkou hloubkou ostrosti. Jediná degradace vstupu bude tedy způsobena samotným pohybem snímaného předmětu nebo kamery. Takové poškození lze modelovat konvolučním ISI kanálem s atypickým kernelem, který bude odvozen v odstavci 6.3.1. Podobně jako u demonstrací v kapitole 6.2.1, bude rozmazaný obraz kontaminován šumem CCD kamery generovaným na základě modelu v odstavci 6.1.2. Výsledky takové rekonstrukční úlohy, pomocí vybraných topologií iterativní dekódovacích sítí, jsou uvedeny v odstavci 6.3.2 při rozmazání obrazu vlivem dvou různých pohybů. První příklad se bude týkat rozmazání pohybem pod úhlem  $45^\circ$  a druhý příklad při rozmazání pohybem ve vertikálním směru.



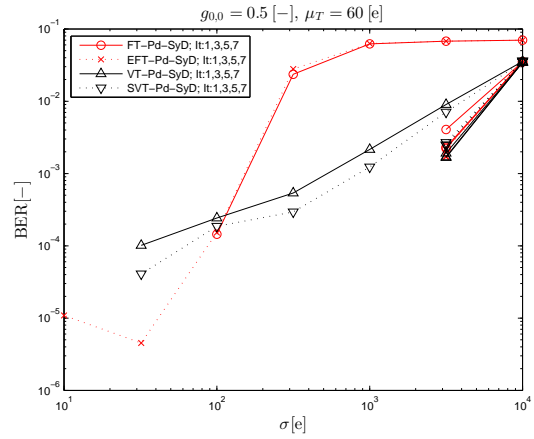
(a) Potlačení hlavního svazku 0.2 ( $\Delta = 0.705$ ).



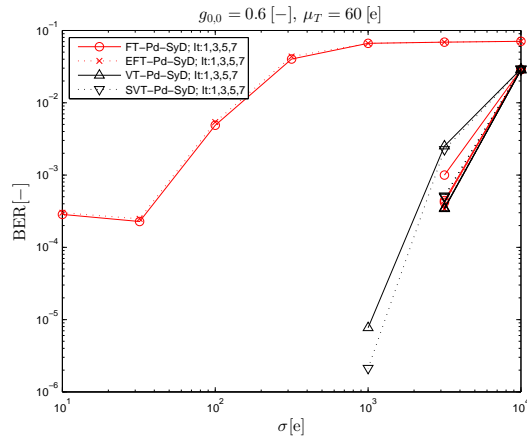
(b) Potlačení hlavního svazku 0.3 ( $\Delta = 1.13$ ).



(c) Potlačení hlavního svazku 0.4 ( $\Delta = 1.624$ ).

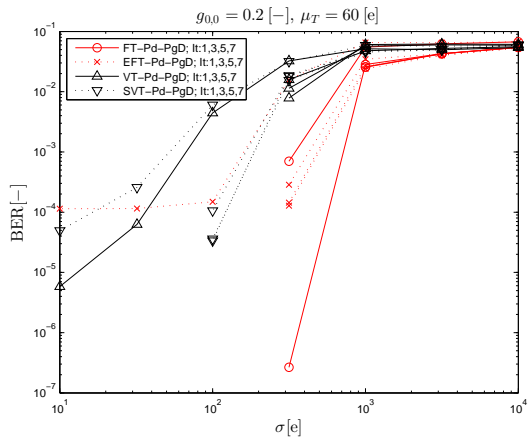


(d) Potlačení hlavního svazku 0.5 ( $\Delta = 2.213$ ).

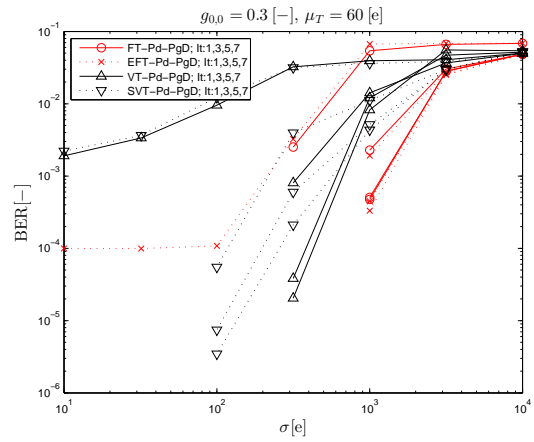


(e) Potlačení hlavního svazku 0.6 ( $\Delta = 2.939$ ).

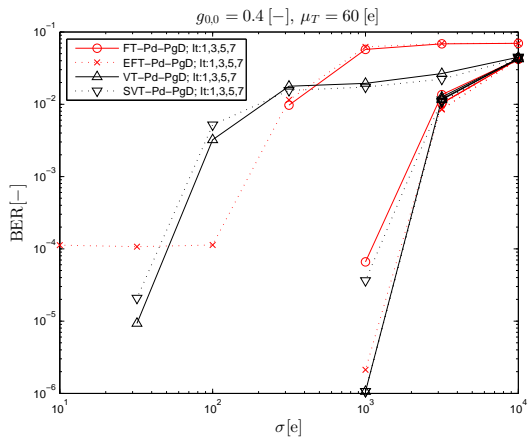
Obr. 35. Analýzy chybovosti rekonstrukce náhodných realizací binárních obrazů pomocí různých druhů IDN využívajících Pd-SyD techniku. Vstupní (šedotónový) obraz byl simulován s plným rozlišením (monochromatická kamera).



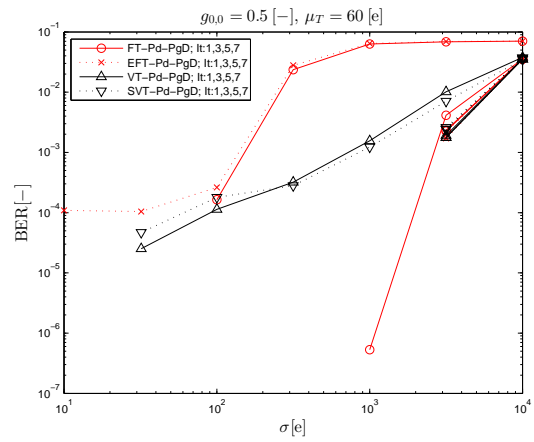
(a) Potlačení hlavního svazku 0.2 ( $\Delta = 0.705$ ).



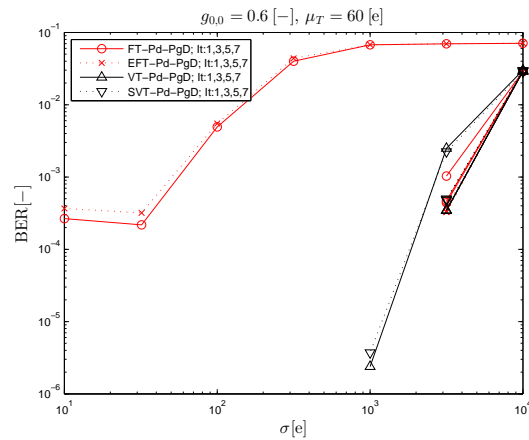
(b) Potlačení hlavního svazku 0.3 ( $\Delta = 1.13$ ).



(c) Potlačení hlavního svazku 0.4 ( $\Delta = 1.624$ ).

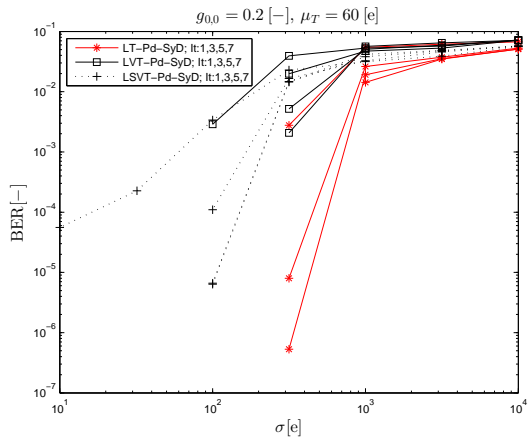


(d) Potlačení hlavního svazku 0.5 ( $\Delta = 2.213$ ).

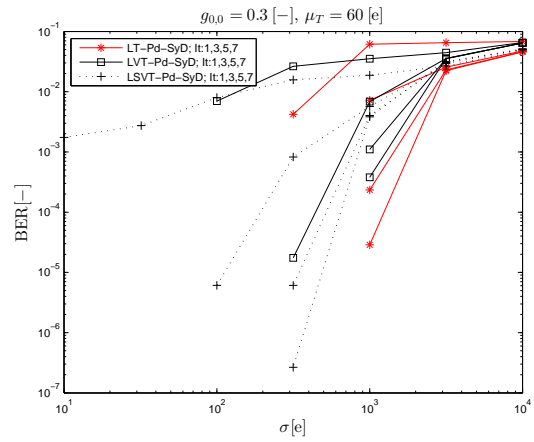


(e) Potlačení hlavního svazku 0.6 ( $\Delta = 2.939$ ).

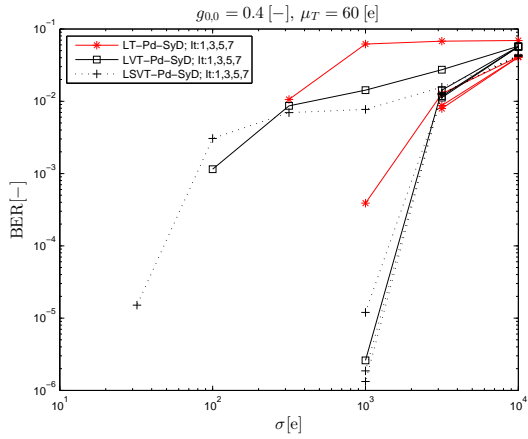
Obr. 36. Analýzy chybovosti rekonstrukce náhodných realizací binárních obrazů pomocí různých druhů IDN využívajících Pd-PgD techniku. Vstupní (šedotónový) obraz byl simulován s plným rozlišením (monochromatická kamera).



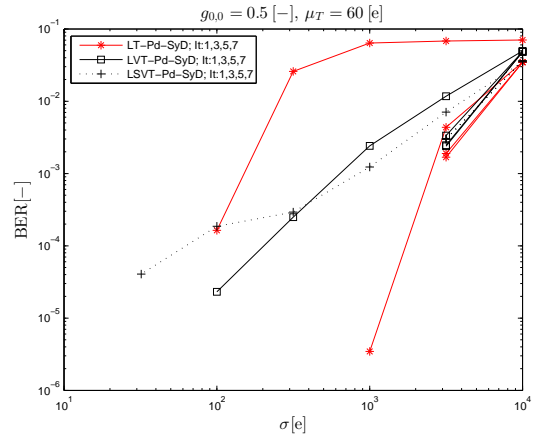
(a) Potlačení hlavního svazku 0.2 ( $\Delta = 0.705$ ).



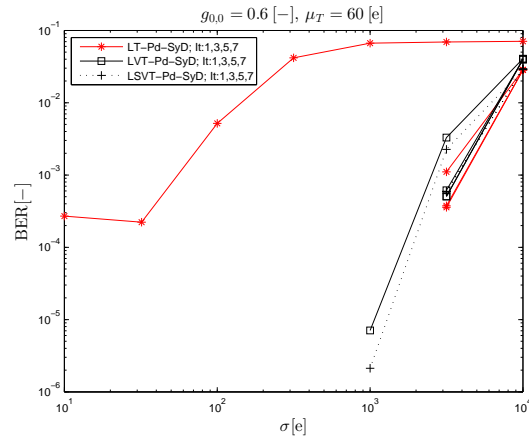
(b) Potlačení hlavního svazku 0.3 ( $\Delta = 1.13$ ).



(c) Potlačení hlavního svazku 0.4 ( $\Delta = 1.624$ ).



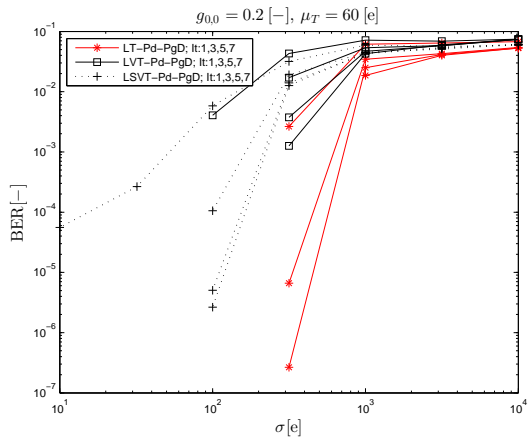
(d) Potlačení hlavního svazku 0.5 ( $\Delta = 2.213$ ).



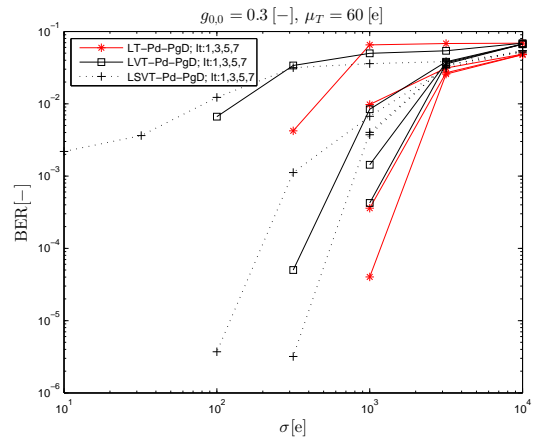
(e) Potlačení hlavního svazku 0.6 ( $\Delta = 2.939$ ).

Obr. 37. Analýzy chybovosti rekonstrukce náhodných realizací binárních obrazů pomocí různých druhů Vrstvových IDN využívajících Pd-SyD techniku. Vstupní (šedotónový) obraz byl simulován s plným rozlišením (monochromatická kamera).

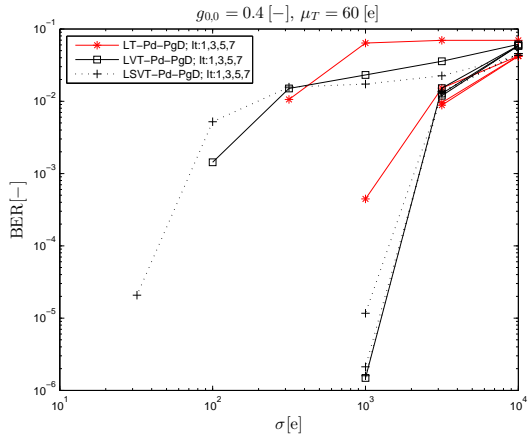




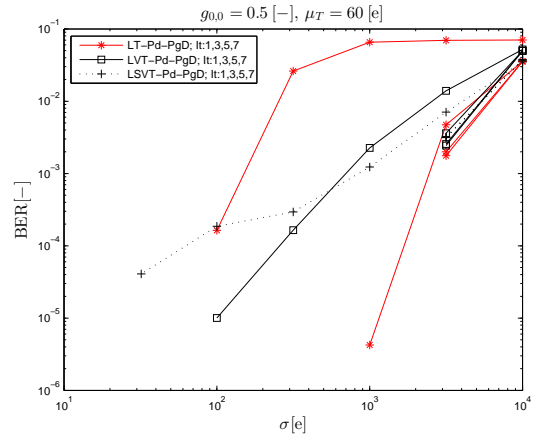
(a) Potlačení hlavního svazku 0.2 ( $\Delta = 0.705$ ).



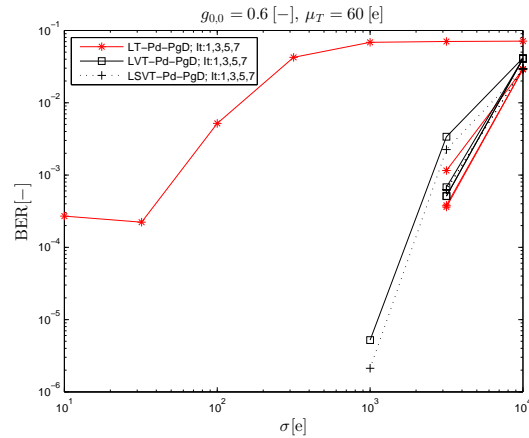
(b) Potlačení hlavního svazku 0.3 ( $\Delta = 1.13$ ).



(c) Potlačení hlavního svazku 0.4 ( $\Delta = 1.624$ ).

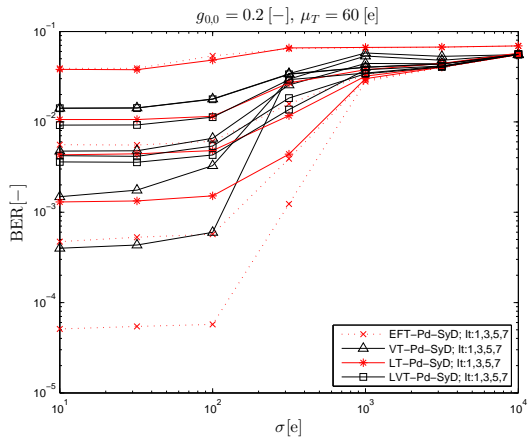


(d) Potlačení hlavního svazku 0.5 ( $\Delta = 2.213$ ).

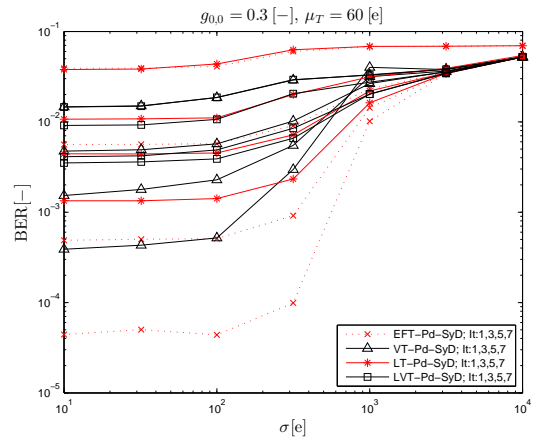


(e) Potlačení hlavního svazku 0.6 ( $\Delta = 2.939$ ).

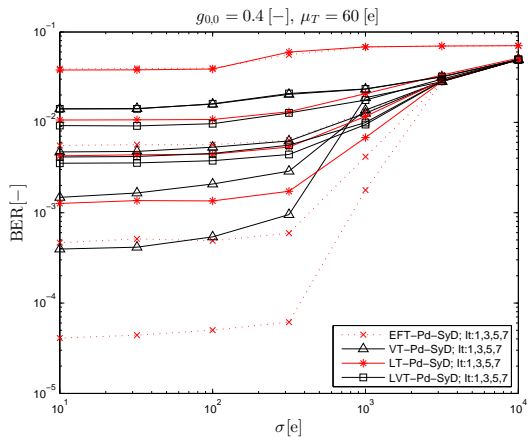
Obr. 38. Analýzy chybovosti rekonstrukce náhodných realizací binárních obrazů pomocí různých druhů Vrstvových IDN využívajících Pd-PgD techniku. Vstupní (šedotónový) obraz byl simulován s plným rozlišením (monochromatická kamera).



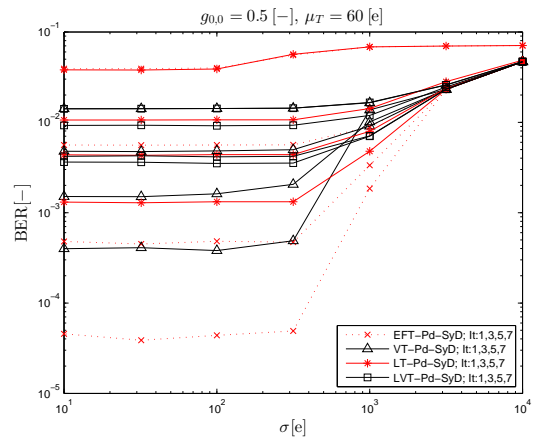
(a) Potlačení hlavního svazku 0.2 ( $\Delta = 0.705$ ).



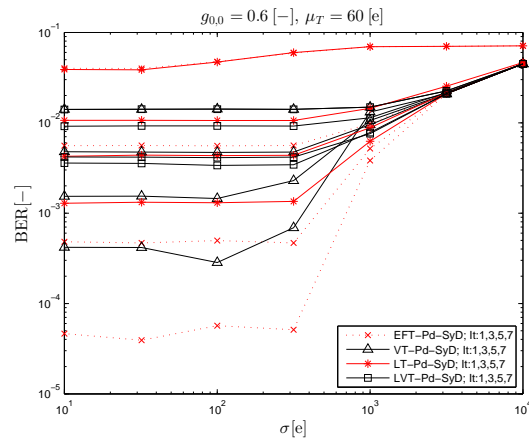
(b) Potlačení hlavního svazku 0.3 ( $\Delta = 1.13$ ).



(c) Potlačení hlavního svazku 0.4 ( $\Delta = 1.624$ ).

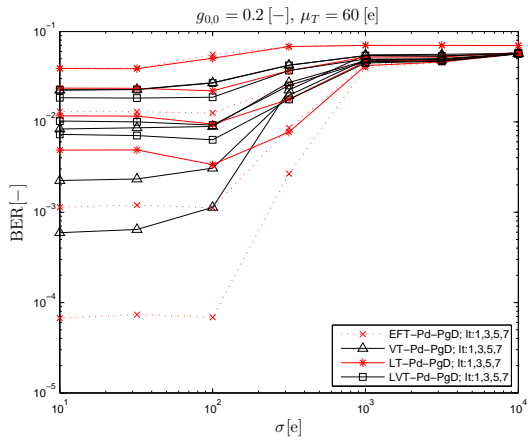


(d) Potlačení hlavního svazku 0.5 ( $\Delta = 2.213$ ).

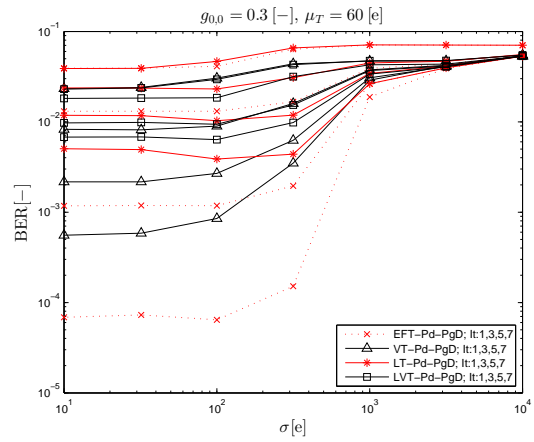


(e) Potlačení hlavního svazku ( $\Delta = 2.939$ ).

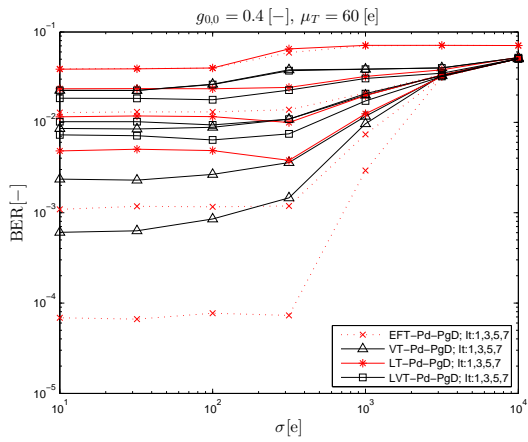
Obr. 39. Analýzy chybovosti rekonstrukce náhodných realizací binárních obrazů pomocí různých druhů IDN využívajících Pd-SyD techniku. Vstupní (šedotónový) obraz byl simulován s polovičním rozlišením (barevná kamera).



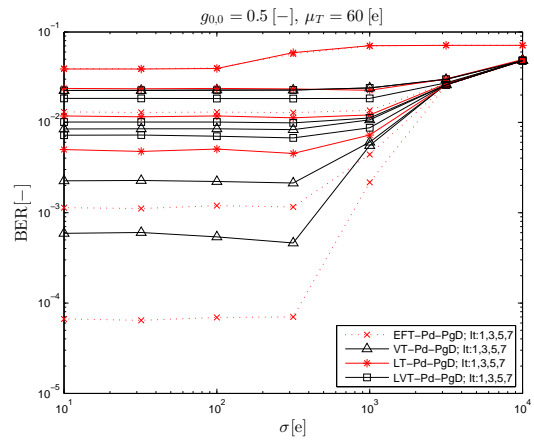
(a) Potlačení hlavního svazku 0.2 ( $\Delta = 0.705$ ).



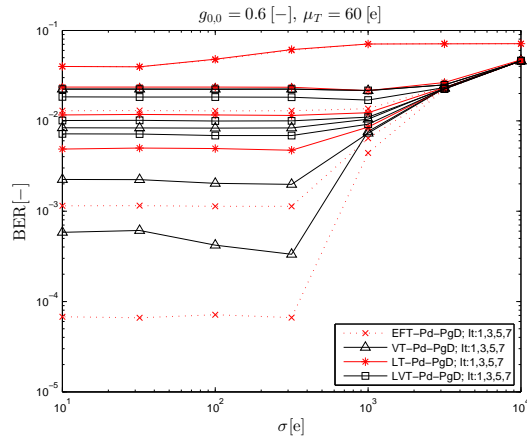
(b) Potlačení hlavního svazku 0.3 ( $\Delta = 1.13$ ).



(c) Potlačení hlavního svazku 0.4 ( $\Delta = 1.624$ ).



(d) Potlačení hlavního svazku 0.5 ( $\Delta = 2.213$ ).



(e) Potlačení hlavního svazku 0.6 ( $\Delta = 2.939$ ).

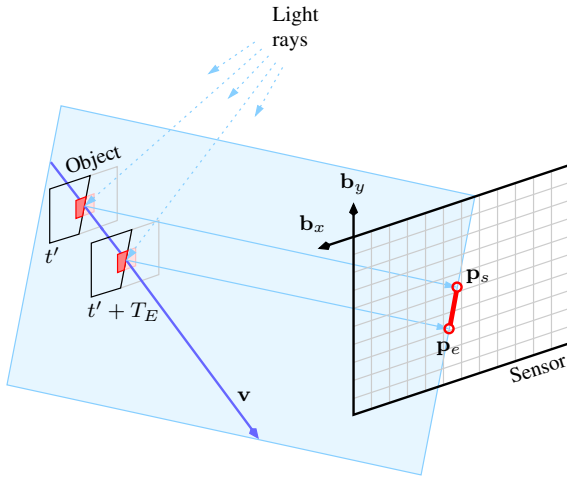
Obr. 40. Analýzy chybovosti rekonstrukce náhodných realizací binárních obrazů pomocí různých druhů IDN využívajících Pd-PgD techniku. Vstupní (šedotónový) obraz byl simulován s polovičním rozlišením (barevná kamera).

### 6.3.1 Kernel kanálu modelující rozmazání pohybem

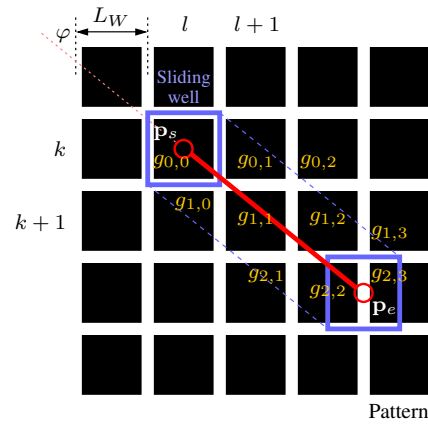
Z fyzikálního hlediska je geneze rozmazání snímaného obrazu vlivem pohybu velice prostá. Předpokládejme snímaný předmět na Obr. 41a, pohybující se rovnoměrně přímočaře po trajektorii specifikované vektorem rychlosti  $\mathbf{v}$ . Plošný element objektu se tak promítne v rovině senzoru  $\mathbf{b}_x \times \mathbf{b}_y$  na úsečku

$$\mathbf{p}(t) = \begin{bmatrix} p_x(t) \\ p_y(t) \end{bmatrix} \quad (132)$$

s počátečním bodem  $\mathbf{p}_s = \mathbf{p}(t')$  a koncovým bodem  $\mathbf{p}_e = \mathbf{p}(t' + T_E)$ , kde  $T_E$  je expoziční doba. Pokud je pohyb jiného charakteru nastává analogická situace, kdy průmět plošky objektu v rovině senzoru vytvoří obecnou křivku  $\mathbf{p}(t)$ .



(a) Projekce objektu v pohybu do roviny senzoru.



(b) Příklad ekvivalentní trajektorie pohybu snímací buňky.

Obr. 41. Geneze rozmazání snímku díky pohybu snímaného objektu či kamery.

Dále učiníme zjednodušující předpoklad, že snímací buňky senzoru jsou čtvercové, bez mezer, a délka jejich strany činí  $L_W$ . Homogenní snímací oblast  $[k, l]$ -té buňky senzoru pak popisuje funkce

$$f_W(x, y)[k, l] = \begin{cases} 1, & |2(x - kL_W)| \leq L_W \text{ and } |2(y - lL_W)| \leq L_W \\ 0, & \text{otherwise} \end{cases} \quad (133)$$

Svazek odražených paprsků ze čtvercové plošky snímaného předmětu se prostřednictvím optické části kamery zaostří do roviny senzoru na plošku o obsahu  $L_W \times L_W$  (Sliding well). Tato čtvercová oblast homogenního odraženého záření se postupně posouvá nad rovinou senzoru a generuje náboj v jednotlivých zasažených buňkách ve směru svého pohybu po trajektorii  $\mathbf{p}(t)$  s tím, že zanedbáme radiální průmět vektoru  $\mathbf{v}$  do roviny senzoru  $\mathbf{b}_x \times \mathbf{b}_y$ . Neboli, že axiální průmět je dominantní a v průběhu expoziční doby bude mít svazek dopadajícího záření stále stejnou velikost  $L_W \times L_W$ . Celou situaci znázorňuje Obr. 41b. Koeficienty kernelu ISI kanálu modelujícího takové rozmazání pak vyplývají z jednoduché integrace

$$\{\mathcal{G}^{(\text{Blur})}\}_{k,l} = \int_{t'}^{t'+T_E} \iint_{S_W(\mathbf{p}(t))} f_W(x, y)[k, l] dx dy dt, \quad (134)$$

kde  $t'$  je okamžik otevření clony a

$$S_W(\mathbf{p}) = \mathbf{p} + \left\langle -\frac{1}{2}L_W, \frac{1}{2}L_W \right\rangle \times \left\langle -\frac{1}{2}L_W, \frac{1}{2}L_W \right\rangle \quad (135)$$

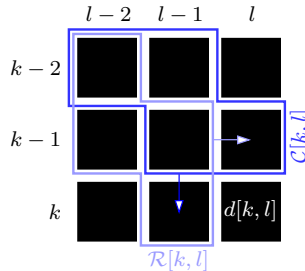
je integrační ploška v místě  $\mathbf{p}$  na rovině senzoru, která svým tvarem a velikostí kopíruje konturu svazku dopadajícího záření odraženého z čtvercové plošky snímaného objektu do roviny senzoru.

### 6.3.2 Příklady rekonstrukcí černobílých snímků iterativním sítěmi při znalosti kernelu kanálu

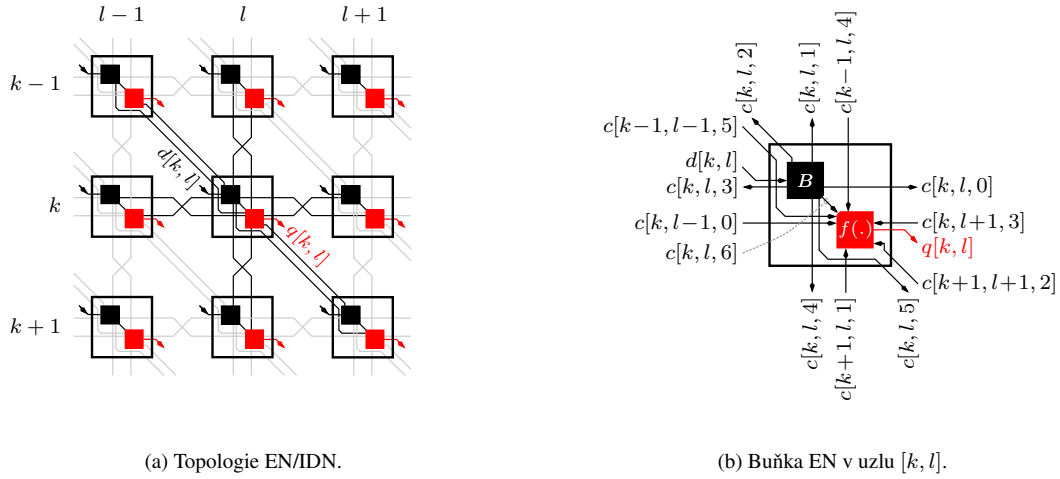
Demonstraci binarizace potlačující rozmazání pohybem provedeme na dvojici kernelů

$$\mathcal{G}_{3 \times 3, 45^\circ}^{(\text{Blur})} = \left\{ \begin{array}{cc} \frac{1}{6} & \frac{1}{12} \\ \frac{1}{12} & \frac{1}{3} & \frac{1}{12} \\ & \frac{1}{12} & \frac{1}{6} \end{array} \right\} \text{ a } \mathcal{G}_{7 \times 1, 0^\circ}^{(\text{Blur})} = \left\{ \begin{array}{c} \frac{1}{12} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{12} \end{array} \right\}, \quad (136)$$

kteřé jsou výsledkem integrace (134) a reprezentují axiální projekci pohybu snímaného předmětu do roviny snímače ve směru  $45^\circ$  a  $0^\circ$ . Nejprve rozeberme možnosti uspořádání klíčových topologií iterativního detektoru eliminující vliv prvního druhu konvolučního kernelu  $\mathcal{G}_{3 \times 3, 45^\circ}^{(\text{Blur})}$ . Na Obr. 42 jsou znázorněné tvary stavových veličin  $\mathcal{R}[k, l]$  a  $\mathcal{C}[k, l]$  vzorové kódovací sítě se základní fixní topologií. Tyto proměnné tvoří předlohu blokových estimatorů  $\mathcal{R}$  a  $\mathcal{C}$  korespondující iterativní dekodovací sítě marginalizující na úrovni bloku symbolů z Obr. 13. Při použití iterativní dekodovací sítě marginalizující na úrovni symbolů bychom získali topologii na Obr. 43a. V tomto uspořádání se jednotlivé uzly vzorové kódovací sítě skládají z kompozitních elementů na Obr. 43b. Na straně detekce by těmto elementům odpovídali měkké inverze na Obr. 43c. V případě druhého typu rozmazání  $\mathcal{G}_{7 \times 1, 0^\circ}^{(\text{Blur})}$  získáme jiná uspořádání. U sítě se základní fixní topologií pozbývá smyslu horizontální stavová proměnná  $\mathcal{R}[k, l]$ . Tvoří prázdnou množinu a síť zahrnuje pouze vertikální stavovou veličinu na Obr. 44. Analogická situace nastává také u variabilní topologie na Obr. 45a. V této struktuře jsou všechna zřetězení také pouze ve vertikálním směru a jednotlivé uzly vzorové kódovací sítě tvoří kompozitní elementy na Obr. 45b. Měkká inverze elementu na Obr. 45b je uvedena na Obr. 45c.

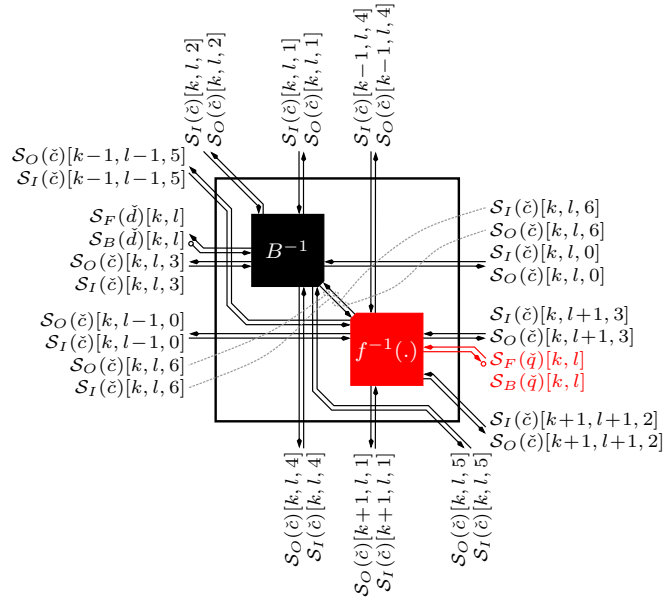


Obr. 42. Tvary stavových proměnných  $\mathcal{R}[k, l]$  a  $\mathcal{C}[k, l]$  pro ukázkový ISI kernel  $\mathcal{G}_{3 \times 3, 45^\circ}^{(\text{Blur})}$ .



(a) Topologie EN/IDN.

(b) Buňka EN v uzlu  $[k, l]$ .

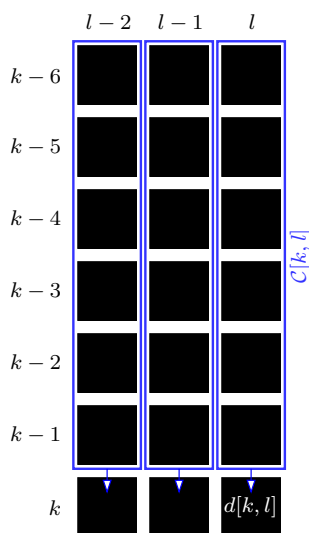


(c) Buňka IDN v uzlu  $[k, l]$ .

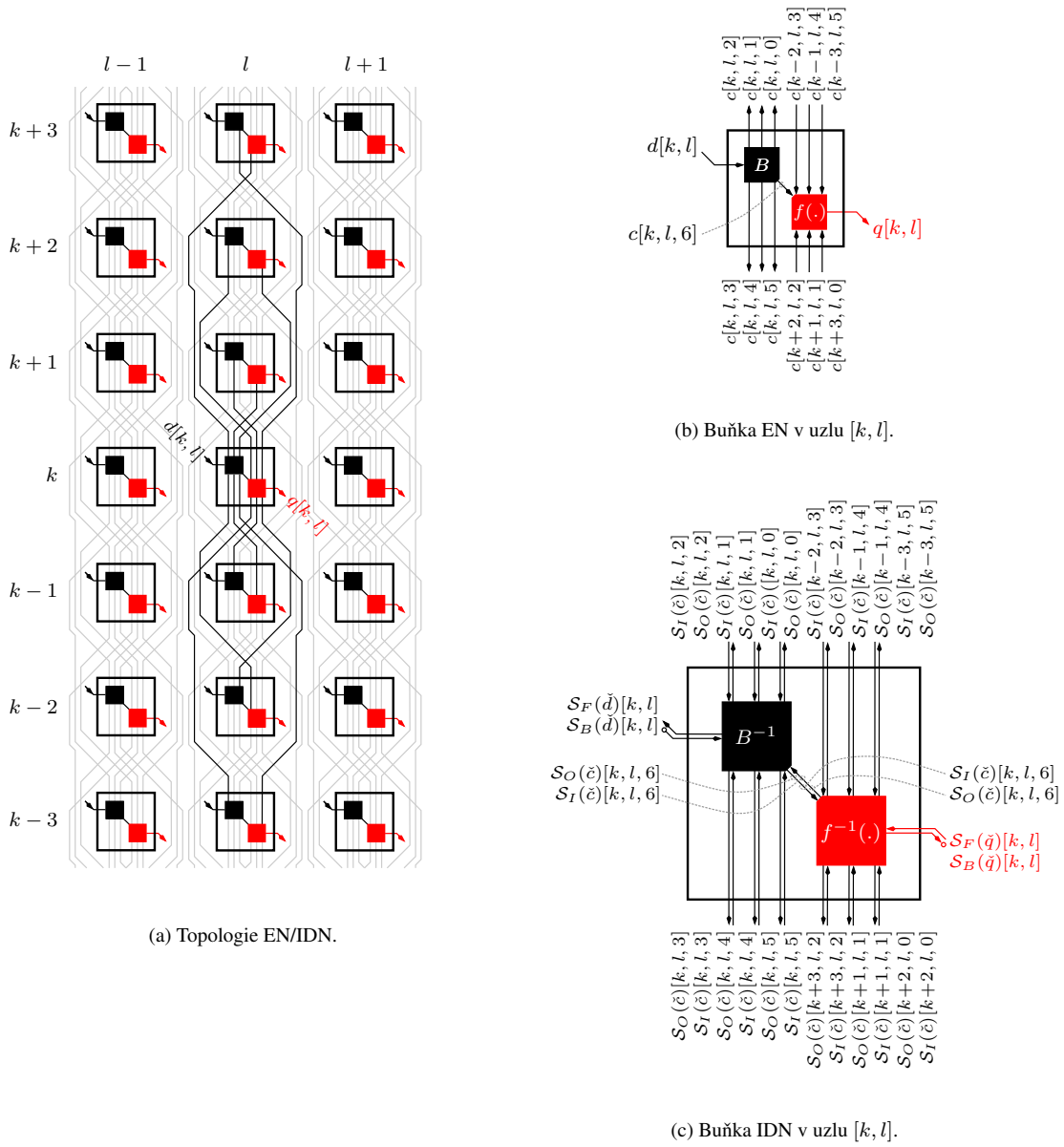
Obr. 43. Iterativní dekodovací síť marginalizující na úrovni symbolů pro ukázkový ISI kernel  $\mathcal{G}_{3 \times 3, 45^\circ}^{(\text{Blur})}$ .

Stejně jako u demonstrací z odstavce 6.2.2 i zde budeme předpokládat perfektní znalost konvolučního kernelu na straně detekce. Také parametry šumového modelu senzoru zachováme a budeme uvažovat  $N_e^{(\text{FWC})} = 30 \times 10^3$  [e],  $N_e^{(\text{ET})} = 29 \times 10^3$  [e],  $\Delta_C = 14$  [e],  $N_B = 11$  a  $\hat{\mu}_T = 60$  [e]. Zvolené parametry vytvoří transformační funkce front-endu detektoru (130) a (131) znázorněné na Obr. 46 resp. Obr. 47 v případě kernelu  $\mathcal{G}_{3 \times 3, 45^\circ}^{(\text{Blur})}$  resp.  $\mathcal{G}_{7 \times 1, 0^\circ}^{(\text{Blur})}$ . Na Obr. 48 jsou uvedeny realizace ukázkového binárního obrazu textu na výstupu modelu CCD senzoru pro obě zvolená rozmazání pohybem. Simulace byla opět provedena jak pro plný vzorkovací rastr (výstup monochromatické kamery) tak pro proděravěný rastr s polovičním rozlišením (výstup barevné kamery). Na Obr. 49 a Obr. 50 jsou vyobrazené rekonstruované texty na výstupu detektoru po proběhnutí 1, 3 a 5 iterace společně s příslušnými chybavostmi detekce a to za předpokladu, že vstupní obraz  $\mathbf{R}$  byl získán v plném rozlišení z modelu monochromatické kamery. První rekonstrukce byla provedena pomocí iterativní dekodovací

sítě marginalizující na úrovni bloku symbolů se základní fixní topologií a při použití stavových proměnných na Obr. 42. Druhou rekonstrukci realizovala síť s variabilní topologií na Obr. 45. Na Obr. 51 a Obr. 52 je znázorněn binarizovaný výstup detektoru za stejných podmínek pouze s tím rozdílem, že vstupní obraz  $\mathbf{R}$  byl získán v polovičním rozlišení (zelený kanál) pomocí simulace barevné kamery s Bayerovským CFA a binarizační úloha proto zahrnuje i de-mosaicing. Obě tyto rekonstrukce zajišťovaly stejné typy sítě jako v předchozích dvou případech. Z výsledků je patrné, že všechny výstupy jsou do 7. iterace detektoru bez chyby, kromě demonstrace na Obr. 50. V této ukázce je šumová kontaminace příliš vysoká vzhledem ke charakteru kernelu. Detekční úloha se stala nejednoznačnou a použitá síť začala divergovat. Uvedené rozpoznávání textu, založené na aplikaci iterativních dekódovacích sítí, by z praktického hlediska mohlo najít uplatnění například v úloze snímání a detekce poznávacích značek jedoucích vozidel.

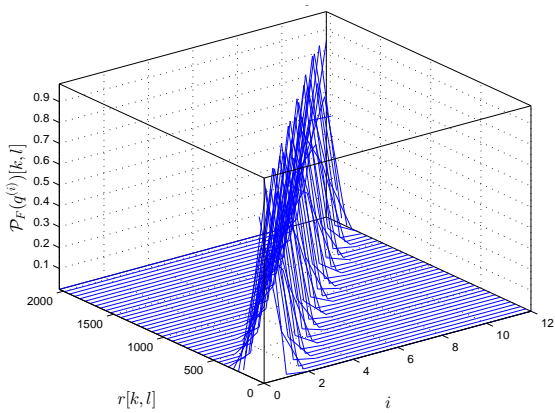


Obr. 44. Tvar stavové proměnné  $\mathcal{C}[k, l]$  pro ukázkový ISI kernel  $\mathcal{G}_{7 \times 1, 0}^{(\text{Blur})}$ .

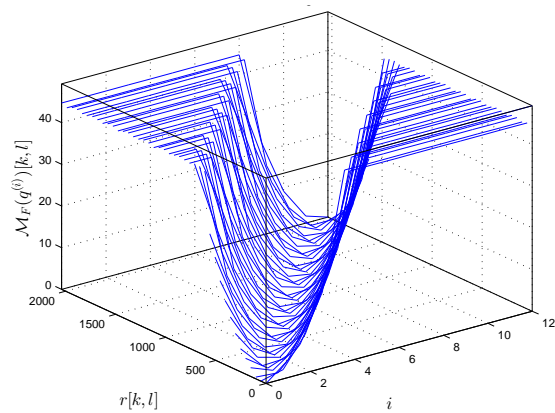


Obr. 45. Iterativní dekodovací síť marginalizující na úrovni symbolů pro ukázkový ISI kernel  $\mathcal{G}_{7 \times 1, 0}^{(\text{Blur})}$ .



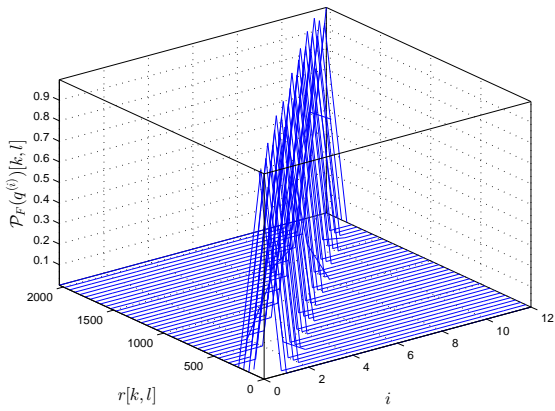


(a) Pravděpodobnostní doména.

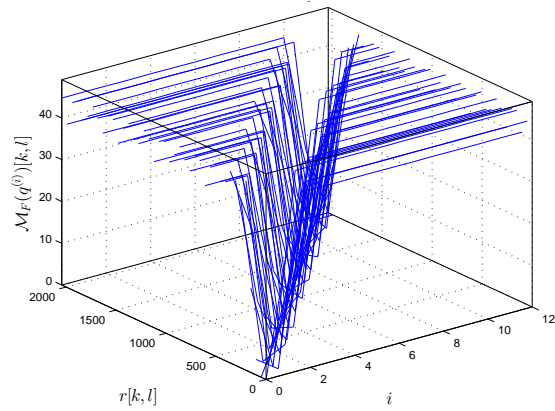


(b) Metrická doména.

Obr. 46. Příklady transformačních funkcí front-endu detektoru (gateway). Model rozmazání pohybem:  $3 \times 3$ ,  $\varphi = 45^\circ$ . Úroveň šumu  $\hat{\sigma} = 10^3$  [e].



(a) Pravděpodobnostní doména.

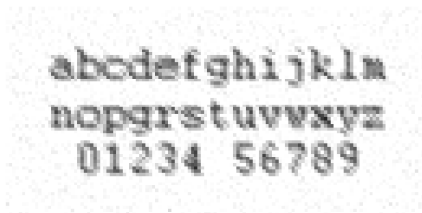


(b) Metrická doména.

Obr. 47. Příklady transformačních funkcí front-endu detektoru (gateway). Model rozmazání pohybem:  $7 \times 1$ ,  $\varphi = 0^\circ$ . Úroveň šumu  $\hat{\sigma} = 316$  [e].

abcdefghijklm  
nopqrstuvwxyz  
01234 56789

(a) Skutečný binární obraz. D.



(b) Monochromatická kamera. Model rozmazání pohybem:  $3 \times 3$ ,  $\varphi = 45^\circ$ . Model sensoru:  $\hat{\sigma} = 10^3$  [e].



(d) Monochromatická kamera. Model rozmazání pohybem:  $7 \times 1$ ,  $\varphi = 0^\circ$ . Model sensoru:  $\hat{\sigma} = 10^3$  [e].

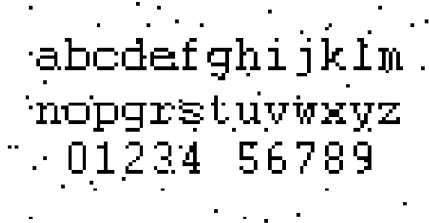


(c) Barevná kamera s Bayerovským CFA. Model rozmazání pohybem:  $3 \times 3$ ,  $\varphi = 45^\circ$ . Model sensoru:  $\hat{\sigma} = 316$  [e].



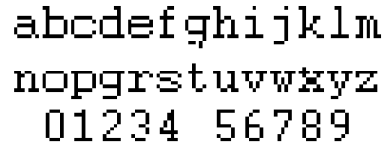
(e) Barevná kamera s Bayerovským CFA. Model rozmazání pohybem:  $7 \times 1$ ,  $\varphi = 0^\circ$ . Model sensoru:  $\hat{\sigma} = 316$  [e].

Obr. 48. Ukázky realizací vstupních obrazů detektoru  $\mathbf{R}$  pro různé úrovně šumu a rozmazání pohybem.



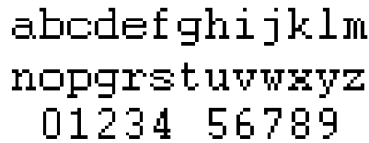
abcdefghijklm  
nopqrstuvwxyz  
01234 56789

(a.1)  $I = 1$ , BER = 0.009



abcdefghijklm  
nopqrstuvwxyz  
01234 56789

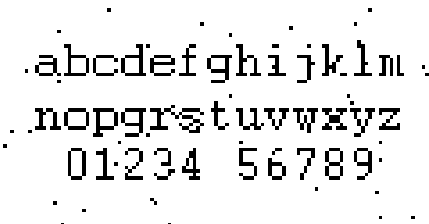
(a.2)  $I = 3$ , BER = 0.002



abcdefghijklm  
nopqrstuvwxyz  
01234 56789

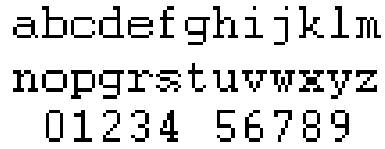
(a.3)  $I = 5$ , BER = 0

(a) Iterativní dekódovací síť využívající Pd-SyD techniku.



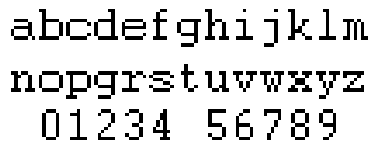
abcdefghijklm  
nopqrstuvwxyz  
01234 56789

(b.1)  $I = 1$ , BER = 0.008



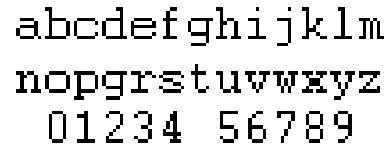
abcdefghijklm  
nopqrstuvwxyz  
01234 56789

(b.2)  $I = 3$ , BER = 0.002



abcdefghijklm  
nopqrstuvwxyz  
01234 56789

(b.3)  $I = 5$ , BER = 0.0006



abcdefghijklm  
nopqrstuvwxyz  
01234 56789

(b.4)  $I = 7$ , BER = 0

(b) Iterativní dekódovací síť využívající Pd-PgD techniku.

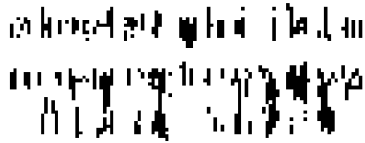
Obr. 49. Ukázky binarizovaných obrazů textu  $\hat{D}$  na výstupu IDN marginalizující na úrovni bloku symbolů se základní topologií. Vstupní (šedotónový) obraz  $R$  byl simulován s plným rozlišením (monochromatická kamera). Model rozmazání pohybem:  $3 \times 3$ ,  $\varphi = 45^\circ$ . Úroveň šumu  $\hat{\sigma} = 10^3$  [e].



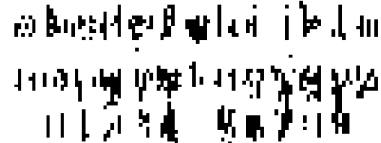
(a.1)  $I = 1$ , BER = 0.068



(a.2)  $I = 3$ , BER = 0.068



(a.3)  $I = 5$ , BER = 0.089

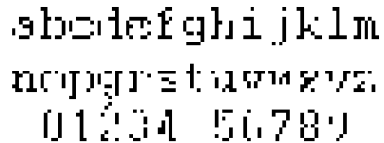


(a.4)  $I = 7$ , BER = 0.080

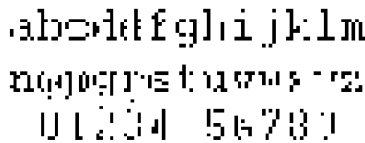
(a) Iterativní dekodovací síť využívající Pd-SyD techniku.



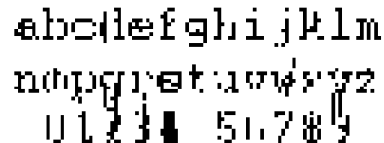
(b.1)  $I = 1$ , BER = 0.077



(b.2)  $I = 3$ , BER = 0.026



(b.3)  $I = 5$ , BER = 0.033



(b.4)  $I = 7$ , BER = 0.035

(b) Iterativní dekodovací síť využívající Pd-PgD techniku.

Obr. 50. Ukázky binarizovaných obrazů textu  $\hat{D}$  na výstupu IDN marginalizující na úrovni symbolů. Vstupní (šedotónový) obraz  $\mathbf{R}$  byl simulován s plným rozlišením (monochromatická kamera). Model rozmazání pohybem:  $7 \times 1$ ,  $\varphi = 0^\circ$ . Úroveň šumu  $\hat{\sigma} = 10^3$  [e].

abdefghijklm  
nopqrstuwxvz  
01234 56789

(a.1)  $I = 1$ , BER = 0.059

abcdefghijklm  
nopgrstuvwxyz  
01234 56789

(a.2)  $I = 3$ , BER = 0.005

abcdefghijklm  
nopgrstuvwxyz  
01234 56789

(a.3)  $I = 5$ , BER = 0

(a) Iterativní dekódovací síť využívající Pd-SyD  
techniku.

abdefghijklm  
nopqrstuwxvz  
01234 56789

(b.1)  $I = 1$ , BER = 0.069

abcdefghijklm  
nopgrstuvwxyz  
01234 56789

(b.2)  $I = 3$ , BER = 0.007

abcdefghijklm  
nopgrstuvwxyz  
01234 56789

(b.3)  $I = 5$ , BER = 0

(b) Iterativní dekódovací síť využívající Pd-PgD  
techniku.

*Obr. 51.* Ukázky binarizovaných obrazů textu  $\hat{\mathbf{D}}$  na výstupu IDN marginalizující na úrovni bloku symbolů se základní topologií. Vstupní (šedotónový) obraz  $\mathbf{R}$  byl simulován s polovičním rozlišením (barevná kamera). Model rozmazání pohybem:  $3 \times 3$ ,  $\varphi = 45^\circ$ . Úroveň šumu  $\hat{\sigma} = 316$  [e].

(a.1)  $I = 1$ , BER = 0.067

(a.2)  $I = 3$ , BER = 0.005

(a.3)  $I = 5$ , BER = 0.002

(a.4)  $I = 7$ , BER = 0

(a) Iterativní dekódovací síť využívající Pd-SyD techniku.

(b.1)  $I = 1$ , BER = 0.079

(b.2)  $I = 3$ , BER = 0.004

(b.3)  $I = 5$ , BER = 0.0002

(b.4)  $I = 7$ , BER = 0

(b) Iterativní dekódovací síť využívající Pd-PgD techniku.

*Obr. 52.* Ukázky binarizovaných obrazů textu  $\hat{D}$  na výstupu IDN marginalizující na úrovni symbolů. Vstupní (šedotónový) obraz  $R$  byl simulován s polovičním rozlišením (barevná kamera). Model rozmazání pohybem:  $7 \times 1$ ,  $\varphi = 0^\circ$ . Úroveň šumu  $\hat{\sigma} = 316$  [e].

## 7 Závěr

Použitím iterativního principu detekce docílíme zjednodušení MAP kriteria na úroveň jednotlivých, vzájemně překrytých, konvolučních regionů. Přesto jsou SISO moduly příliš složité pro větší kernely a vstupní data ve vyšší aritmetice, než té binární. Tato složitost nás omezuje a nedovoluje aplikaci dvoudimenzionálních iterativních detektorů na dekonvoluci větších kernelů a při vstupu dat ve vyšší aritmetice (šedotónové obrazy). Doposud byly tyto algoritmy úspěšně aplikovány pouze v oblasti tvorby půltónových obrazů [9, 10] nebo v úloze binarizace obrazu uvedené v této práci a v publikacích [4, 6]. Zde však dosahují iterativní dekodovací sítě velmi dobrých výsledků, mnohdy zcela bezchybných. Platí to i v případě rozšíření úlohy o de-mosaicing, kdy polovina vstupní informace není přítomna vůbec a ta druhá je rozostřena a kontaminována šumy. Přesto je síť schopna původní data obnovit a doplnit do plného rozlišení. Jako klíčová oblast praktického využití současných implementací se jeví mikroskopie. Konkrétně binarizace obrazů mikroorganismů a buněk získaných z mikroskopů se CCD senzorem [2] za účelem zjištění jejich koncentrace (počtu) v předmětném vzorku. V této publikaci [2] byla aplikována uvedená technologie již na reálných obrazech. Výstupy byly porovnány s klíčovými binarizačními algoritmy [37] a [38]. Objektívni testy ukázali, že iterativní dekodovací sítě poskytují mnohem lepší výsledek. Obzvláště při velmi krátkých expozičních dobách s velkou šumovou kontaminací. V těchto podmínkách, nutných pro snímání biologických vzorků senzitivních na světlo, klasické metody binarizace dosti selhávají.

Z hlediska výkonnosti jsou nejlepší variantou iterativní dekodovací sítě se základním fixním zřetěžením. Pracují s obsáhlými stavovými veličinami, které nemusí být a mnohdy nejsou disjunktní. Tento fakt podporuje výkonnost těchto struktur. Jejich nevýhoda spočívá ve velkých paměťových nárocích a obtížné aplikaci na tvarově nestandardní kernely. Průniky stavových veličin také komplikují použití a návrh optimální implementace SISO modulu pomocí stromové kombinace-marginalizace, která výrazně snižuje výpočetní náročnost. Negativní vlastnosti jsou částečně eliminovány v rozšířené fixní topologii s diagonální stavovou veličinou. Tato varianta však nedosahuje takové výkonnosti. Pokud požadujeme nízké paměťové nároky a vysokou aplikační flexibilitu jsou nejlepší variantou variabilní topologie. Dobře se aplikují na kernely hvězdicového tvaru, kdy fixní topologie většinou nelze použít a nalézt v optimální složitosti. U fixní topologie je totiž datový vstup  $d[k, l]$  vždy ztotožněn s pravým spodním rohem konvolučního regionu. Pokud v tomto místě má kanál nulový přenos, což například hvězdicový kernel  $\mathcal{G}_{3 \times 3}^{(R)}$  má, tak vstupní proměnná v referenčním modelu neovlivňuje výstup  $q[k, l]$ , ale pouze výstupní stavy  $\mathcal{R}[k, l + 1]$ ,  $\mathcal{C}[k + 1, l]$  a  $\mathcal{B}[k + 1, l + 1]$ . Výstup zde závisí jen na vstupních stavech  $\mathcal{R}[k, l]$ ,  $\mathcal{C}[k, l]$  a  $\mathcal{B}[k, l]$ . Proto mohutnost kombinační tabulky SISO modulů bude  $\text{card}(\mathcal{A}_{IO}) = 2^6 = 64$ , tedy dvakrát více než pro tentýž kernel detekovaný pomocí variabilní topologie, kde mohutnost tabulky je  $\text{card}(\mathcal{A}_{IO}) = 2^5 = 32$  díky optimálnímu pokrytí regionu. Za předpokladu, že klíčovým parametrem, který chceme optimalizovat, je výpočetní náročnost použijeme vrstvenou architekturu. Díky “mezi-vrstvové” marginalizaci dochází k redukci mohutnosti tabulky SISO modulů na vrchní vrstvě architektury. Zároveň dojde ke snížení výkonnosti. Nicméně toto zhoršení je pouze mírné v porovnání s úsporou výpočetních operací. Všechny uvedené varianty lze dobře aplikovat jak na plný snímací rastr, tak na proděravěný rastr s Bayerovským polem. To neplatí u zjednodušené vrstvené nebo zjednodušené variabilní topologie. Zde není přítomna měkká inverze broadcasteru a proto tato struktura vyžaduje vstupní informaci v plném rozlišení. Jinak není schopná detekce. V úloze s plným vzorkováním má však velmi uspokojivou výkonnost vůči naprosto zanedbatelným výpočetním nárokům na marginalizaci, která se provádí pouze jedna a to v centru regionu. Veškeré klíčové vlastnosti všech představených topologií iterativních dekodovacích sítí jsou sumarizovány v Tab. 10 pro kernel  $\mathcal{G}_{3 \times 3}^{(\text{Gauss})}$  a v Tab. 11 pro redukovaný kernel  $\mathcal{G}_{3 \times 3}^{(\text{Gauss}, R)}$ . Zde symbol  $\mathcal{V}_S \subset \mathcal{N}_{IO}$  značí množinu všech stavových a pomocných proměnných v dané topologii a symbol  $\mathcal{V}_M \subset \mathcal{N}_{IO}$  značí množinu všech (nutně) marginalizovaných proměnných v dané topologii. Výpočetní náročnost SISO modulů je zanesena prostřednictvím maximálních počtů kombinačních a marginalizačních operací. Tyto hodnoty tedy korespondují se strukturou bez optimalizace v podobě kombinace a marginalizace se stromovou strukturou.

Topologie	Pevná	Rozšířená pevná	Vrstvová pevná	Variabilní	Zjednodušená variabilní	Vrstvová variabilní	Zj. vrstvová variabilní
Složítost: $\text{card}(\mathcal{A}_{IO})$	512	512	8+216	512	512	8+216	8+216
$\mathcal{V}_S$	$\mathcal{R}[0], \mathcal{R}[1]$ $\mathcal{C}[0], \mathcal{C}[1]$	$\mathcal{R}[0], \mathcal{R}[1]$ $\mathcal{C}[0], \mathcal{C}[1]$ $\mathcal{B}[0], \mathcal{B}[1]$	$\mathcal{R}[0], \mathcal{R}[1]$ $\mathcal{C}[0], \mathcal{C}[1]$ $\mathcal{C}$	$\mathcal{C}[0], \dots, \mathcal{C}[8]$	$\mathcal{C}[0], \dots, \mathcal{C}[8]$	$\mathcal{C}[0], \dots, \mathcal{C}[2]$ , $\mathcal{C}(\mathbf{B}), \mathcal{C}(\mathbf{T})$ , $\mathcal{Q}[0], \dots, \mathcal{Q}[2]$	$\mathcal{C}[0], \dots, \mathcal{C}[2]$ , $\mathcal{C}(\mathbf{B}), \mathcal{C}(\mathbf{T})$ , $\mathcal{Q}[0], \dots, \mathcal{Q}[2]$
$\text{card}(\mathcal{V}_S)$	4	6	5	9	9	7	7
Počet $\odot$ na buňku	$512 \times 6$ (3072)	$512 \times 8$ (4096)	$8 \times 4$ $+216 \times 4$ (896)	$512 \times 10$ $+2 \times 10$ (5140)	$512 \times 10$ (5120)	$(216 + 6) \times 4$ $+(8 + 2) \times 4$ (928)	$8 \times 4$ $+216 \times 4$ (896)
$\mathcal{V}_M$	$\mathcal{V}_S, \mathcal{A}$	$\mathcal{V}_S, \mathcal{A}$	$\mathcal{V}_S, \mathcal{A}$	$\mathcal{V}_S, \mathcal{A}$	$\mathcal{C}[8], \mathcal{A}$	$\mathcal{V}_S, \mathcal{A}$	$\mathcal{C}[2], \mathcal{C}(\mathbf{B})$ , $\mathcal{Q}[2], \mathcal{A}$
$\text{card}(\mathcal{V}_M)$	4+1	6+1	6+1	9+1	1+1	8+1	3+1
$\text{card}(\{\mathcal{V}_M\}_\ell)$	64, 64, 64, 64, 2	8, 8, 8, 8, 4, 4, 2	4, 4, 6, 6, 36, 36, 2	2, 2, 2, 2, 2, 2, 2, 2, 2	2, 2	2, 2, 2, 6, 6, 6, 6, 6, 2	2, 6, 6, 2
Počet $\oplus$ na buňku	$4 \times 448$ $+510$ (2302)	$4 \times 504$ $+2 \times 508$ $+510$ (3542)	$2 \times 4 + 6$ $+2 \times 180$ $+210$ (584)	$9 \times 510$ (4590)	510 (510)	$3 \times 6 + 2$ $+3 \times 210$ (650)	$6 + 2$ $+210$ (218)
Počet propojek na buňku	4	6	2 (B) $+1 (\mathbf{B} \rightleftharpoons \mathbf{T})$ $+2 (\mathbf{T})$	8 (■) $+8 (\mathbf{■})$	8 (■)	2 × 2 (■) $+1 (\mathbf{B} \rightleftharpoons \mathbf{T})$ $+2 \times 2 (\mathbf{■})$	2 (■) $+1 (\mathbf{B} \rightleftharpoons \mathbf{T})$ $+2 (\mathbf{■})$
Počet registrů na buňku	$4 \times 64$ (256)	$4 \times 8$ $+2 \times 4$ (40)	$2 \times 4 + 6$ $+2 \times 36$ (86)	$8 \times 2$ $+8 \times 2$ (32)	$8 \times 2$ (16)	$4 \times 2 + 6$ $+4 \times 6$ (38)	$2 \times 2 + 6$ $+2 \times 6$ (22)

Tab. 10. Přehled vlastností iterativních dekodovacích sítí pro ISI kernel  $\mathcal{G}_{3 \times 3}^{(\text{Gauss})}$ . Horní meze počtů kombináčích a marginalizačních operací (numerická složitost SISO modulů).



Topologie	Pevná	Rozšířená pevná	Variabilní	Zjednodušená variabilní
Složitost: $\text{card}(\mathcal{A}_{IO})$	64	64	32	32
$\mathcal{V}_S$	$\mathcal{R}[0], \mathcal{R}[1]$ $\mathcal{C}[0], \mathcal{C}[1]$	$\mathcal{R}[0], \mathcal{R}[1], \mathcal{C}[0],$ $\mathcal{C}[1], \mathcal{B}[0], \mathcal{B}[1]$	$\mathcal{C}[0], \dots, \mathcal{C}[4]$	$\mathcal{C}[0], \dots, \mathcal{C}[4]$
$\text{card}(\mathcal{V}_S)$	4	6	5	5
Počet $\odot$ na buňku	$64 \times 6$ (384)	$64 \times 8$ (512)	$32 \times 6 + 2 \times 6$ (204)	$32 \times 6$ (192)
$\mathcal{V}_M$	$\mathcal{V}_S, \mathcal{d}$	$\mathcal{V}_S, \mathcal{d}$	$\mathcal{V}_S, \mathcal{d}$	$\mathcal{C}[4], \mathcal{d}$
$\text{card}(\mathcal{V}_M)$	4+1	6+1	5+1	1+1
$\text{card}(\{\mathcal{V}_M\}_\ell)$	8, 8, 8, 8, 2	4, 4, 4, 4, 2, 2, 2	2, 2, 2, 2, 2	2, 2
Počet $\otimes$ na buňku	$4 \times 56 + 62$ (286)	$4 \times 60 + 3 \times 62$ (426)	$5 \times 30$ (150)	30 (30)
Počet propojek na buňku	4	6	4 (■) + 4 (■)	4 (■)
Počet registrů na buňku	$4 \times 8$ (32)	$4 \times 4 + 2 \times 2$ (20)	$4 \times 2 + 4 \times 2$ (16)	$4 \times 2$ (8)

Tab. 11. Přehled vlastností iterativních dekódovacích sítí pro redukovaný ISI kernel  $\mathcal{G}_{3 \times 3}^{(\text{Gauss}, \text{R})}$ . Horní meze počtů kombinačních a marginalizačních operací (numerická složitost SISO modulů).

Otevřených problémů je oblastí dvoudimenzionální iterativní MAP detekce několik. Především redukce složitosti SISO modulů. Řešením může spočívat v provedení redukce kombinačních tabulek o méně pravděpodobné či zcela nepravděpodobné přechody. Například z Obr. 26 je jasné patrné, že v řezu transformační funkce ve směru indexu  $i$  (histogram  $\mathcal{P}_F$  pro vstupní  $r[k, l]$ ) je většina hodnot nulových. Všechny aposteriorní sdružené pravděpodobnosti, které obsahují tyto nulové hodnoty jsou proto automaticky také nulové a lze je vyřadit z procesu kombinace-marginalizace bez újmy na výsledek detekce. Tabulku tedy adaptivně přizpůsobíme příslušné hustotě  $\mathcal{P}_F(\tilde{q})$  tak, že v ní ponecháme pouze ty přechody v nichž jsou hodnoty hustoty nenulové. Tím se tabulka značně redukuje, přičemž množství vyřazených přechodů bude tím vyšší, čím bude směrodatná odchylka šumu nižší.

Další oblast výzkumu se týká problematiky synchronizace sítě. V reálné aplikaci není kernel detailně známý, tedy CSI na straně detekce není perfektní. Je pouze změřitelný kalibračním procesem s omezenou přesností. Je nezbytné tuto míru neurčitosti účinně synchronizovat, aby síť podala optimální výkon. K vyřešení této přidružené synchronizační úlohy bychom patrně mohli použít partikulární formy EM kritéria (Expectation-Maximization) nebo obdobného algoritmu.

# Přílohy

## A. Obsah přiloženého CD-ROMu

1. Adresář **Thesis** obsahuje zdrojové soubory disertační práce a obrázky v ní obsažené
2. Adresář **Simulation** obsahuje:
  - (a) Spouštěcí m-skripty simulace
    - i. `ENsSimulator.m` (simulace kódovací sítě FT, EFT, VT a LT)
    - ii. `DNsSimulator.m` (simulace iterativní dekódovací sítě FT, EFT, VT, SVT, LT, LVT nebo LSVT včetně zbytku přenosového řetězce)
    - iii. `DNsPerformanceAnalyzer.m` (analýza výkonnosti iterativní dekódovací sítě FT, EFT, VT, SVT, LT, LVT nebo LSVT)
  - (b) Vytvořený toolbox tříd v podadresáři **+Classes** pro simulaci obecného 2D iterativního detektoru a kódovací sítě
    - i. `@BlurringChannel` (Obecný 2D ISI kernel)
      - A. `BlurringChannel.m` (konstruktor)
      - B. `Plotting.m` (vykreslení koeficientů kernelu)
    - ii. `@EFTDecodingNetwork` (Iterativní dekódovací síť marginalizující na úrovni bloku symbolů s rozšířenou topologií)
      - A. `CreateNSets.m` (definice tvarů a poloh stavových proměnných nad kernelem a definice polohy vstupních dat nad kernelem)
      - B. `EFTDecodingNetwork.m` (konstruktor)
      - C. `NextInput.m` (čtení hustot z registrů na propojkách sítě a jejich zápis na vstupy SISO modulu)
      - D. `NextOutput.m` (čtení hustot z výstupů SISO modulu a jejich zápis do registrů na propojkách sítě)
      - E. `NextOutputExtended.m` (rozšířené čtení hustot z výstupů SISO modulu a jejich zápis do registrů na propojkách sítě)
      - F. `Reset.m` (nastavení stavových hustot do uniformního tvaru)
      - G. `SetMode.m` (nastavení selektoru vstupů a výstupů SISO modulu)
      - H. `SetTOC.m` (nastavení tabulky kombinací SISO modulu na základě zadaného ISI kernelu)
    - iii. `@EncodingNetwork` (Kódovací síť)
      - A. `EFTNetwork.m` (kódovací síť s rozšířenou fixní topologií)
      - B. `EncodingNetwork.m` (konstruktor)
      - C. `FTNetwork.m` (kódovací síť s fixní topologií)
      - D. `LTNetwork.m` (kódovací síť s vrstvou topologií)
      - E. `SetStates.m` (vytvoření stavových proměnných)
      - F. `SetTopology.m` (nastavení topologie kódovací sítě)
      - G. `VTNetwork.m` (kódovací síť s variabilní topologií)
    - iv. `@ENProperties` (Vlastnosti kódovací sítě)
      - A. `CreateWNSets.m` (pomocná funkce výpočtu tabulky kombinací a abecedy  $\mathcal{A}_q$  ze zadaného ISI kernelu)
      - B. `ENProperties.m` (konstruktor)
      - C. `SetWeights.m` (pomocná funkce výpočtu tabulky kombinací a abecedy  $\mathcal{A}_q$  ze zadaného ISI kernelu)
      - D. `StabilizeTOCandOA.m` (stabilizace tabulky kombinací a abecedy  $\mathcal{A}_q$  výstupů kódovací sítě)

- v. @FastSISO (Akcelerovaný SISO modul)
  - A. FastSISO.m (konstruktor)
  - B. MarginalizeCorrMinSM.m (marginalizace sdružených hustot Md-SyD)
  - C. MarginalizeMaxSM.m (marginalizace sdružených hustot Pd-PgD)
  - D. MarginalizeMinSM.m (marginalizace sdružených hustot Md-PgD)
  - E. MarginalizeSumSM.m (marginalizace sdružených hustot Pd-SyD)
  - F. PrepareSM.m (příprava vstupních hustot před kombinováním)
- vi. @FastSISOB (Akcelerovaná měkká inverze broadcasteru)
  - A. FastSISOB.m (konstruktor)
  - B. NextInput.m (čtení ze vstupů měkké inverze)
  - C. NextOutput.m (zápis na výstupy měkké inverze)
- vii. @FileWriter (Zápis do souboru)
  - A. FileWriter.m (konstruktor)
  - B. WriteMatrix3D.m (zápis hodnot 3D pole do souboru)
- viii. @FTDecodingNetwork (Iterativní dekódovací síť marginalizující na úrovni bloku symbolů se základní topologií)
  - A. All.m (vykonání všech iterací dekódovací sítě ze zadaného počtu)
  - B. CreateNSets.m (definice tvarů a poloh stavových proměnných nad kernelem a definice polohy vstupních dat nad kernelem)
  - C. Decode.m (vykonání 1 iterace dekódovací sítě)
  - D. DecodeExtended.m (vykonání 1 iterace dekódovací sítě včetně výpočtu  $\{\mathcal{S}_B(\check{\mathbf{Q}})\}$ )
  - E. DecodeNormal.m (vykonání 1 iterace dekódovací sítě bez výpočtu  $\{\mathcal{S}_B(\check{\mathbf{Q}})\}$ )
  - F. FTDecodingNetwork.m (konstruktor)
  - G. HardDecisionMD.m (prahování výstupních sdružených hustot Md - tvrdý odhad dat)
  - H. HardDecisionPD.m (prahování výstupních sdružených hustot Pd - tvrdý odhad dat)
  - I. NextInput.m (čtení hustot z registrů na propojkách sítě a jejich zápis na vstupy SISO modulu)
  - J. NextOutput.m (čtení hustot z výstupů SISO modulu a jejich zápis do registrů na propojkách sítě)
  - K. NextOutputExtended.m (rozšířené čtení hustot z výstupů SISO modulu a jejich zápis do registrů na propojkách sítě)
  - L. Reset.m (nastavení stavových hustot do uniformního tvaru)
  - M. SetMode.m (nastavení selektoru vstupů a výstupů SISO modulu)
  - N. SetTOC.m (nastavení tabulky kombinací SISO modulu na základě zadaného ISI kernelu)
- ix. @Gateway (Front-end iterativního detektoru)
  - A. CreateHistograms.m (výpočet transformačních tabulek front-endu ze zadaných hodnot Gaussova šumu)
  - B. Gateway.m (konstruktor)
  - C. GetAddresses.m (výpočet adresy do transformační tabulky front-endu ze vzorku vstupního signálu  $r[k, l]$ )
  - D. GetSM.m (čtení příslušné dopředné hustoty z transformační tabulky front-endu prostřednictvím vstupní adresy)
  - E. PlottingHistograms.m (vykreslení 3D grafu transformačních tabulek front-endu)
  - F. PlottingSM.m (vykreslení 3D grafu hustot na výstupu funkcí front-endu)
- x. @GaussianDefocusingChannel (Gaussův defokusační 2D ISI kernel)
  - A. GaussianDefocusingChannel.m (konstruktor)
  - B. Plotting.m (vykreslení koeficientů kernelu)

- xi. @LTDecodingNetwork (Iterativní dekódovací síť marginalizující na úrovni bloku symbolů vrstvou topologií)
  - A. All.m (vykonání všech iterací dekódovací sítě ze zadaného počtu)
  - B. DecodeBL.m (vykonání 1 iterace spodní vrstvy dekódovací sítě)
  - C. DecodeExtended.m (vykonání 1 iterace dekódovací sítě včetně výpočtu  $\{\mathcal{S}_B(\check{\mathbf{Q}})\}$ )
  - D. DecodeNormal.m (vykonání 1 iterace dekódovací sítě bez výpočtu  $\{\mathcal{S}_B(\check{\mathbf{Q}})\}$ )
  - E. LTDecodingNetwork.m (konstruktor)
  - F. NextInput.m (čtení hustot z registrů na propojkách spodní vrstvy sítě a jejich zápis na vstupy SISO modulu)
  - G. NextInputTL.m (čtení hustot z registrů na propojkách vrchní vrstvy sítě a jejich zápis na vstupy SISO modulu)
  - H. NextOutput.m (čtení hustot z výstupů SISO modulu a jejich zápis do registrů na propojkách spodní vrstvy sítě)
  - I. NextOutputExtendedTL.m (rozšířené čtení hustot z výstupů SISO modulu a jejich zápis do registrů na propojkách vrchní vrstvy sítě)
  - J. NextOutputTL.m (čtení hustot z výstupů SISO modulu a jejich zápis do registrů na propojkách vrchní vrstvy sítě)
  - K. Reset.m (nastavení stavových hustot do uniformního tvaru)
  - L. SetMode.m (nastavení selektoru vstupů a výstupů SISO modulu na spodní vrstvě)
  - M. SetModeTL.m (nastavení selektoru vstupů a výstupů SISO modulu na vrchní vrstvě)
  - N. SetTOC.m (nastavení tabulky kombinací SISO modulu na základě zadaného ISI kernelu)
- xii. @LVTDecodingNetwork (Iterativní dekódovací síť marginalizující na úrovni symbolů vrstvou topologií)
  - A. All.m (vykonání všech iterací dekódovací sítě ze zadaného počtu)
  - B. Decode.m (vykonání 1 iterace dekódovací sítě)
  - C. LVTDecodingNetwork.m (konstruktor)
  - D. Reset.m (nastavení stavových hustot do uniformního tvaru)
- xiii. @PlotSTATS (Vykreslení výsledků analýzy výkonnosti)
  - A. PlotSTATS.m (konstruktor)
  - B. Plotting.m (vykreslení BER křivek z naměřených hodnot chybovosti v jednotlivých iteracích)
- xiv. @ReadSTATS (Čtení výsledků analýzy výkonnosti ze souboru)
  - A. ReadHeader.m (čtení hlavičky měření chybovosti)
  - B. ReadMeasureSTATS.m (čtení výsledků měření chybovosti v jednotlivých iteracích)
  - C. ReadSTATS.m (konstruktor)
- xv. @SISO (SISO modul)
  - A. CombineSM.m (kombinace vstupních hustot)
  - B. InitializeSM.m (inicializace modulu)
  - C. MarginalizeSM.m (marginalizace sdružených hustot)
  - D. SISO.m (konstruktor)
- xvi. @SISOB (Měkká inverze broadcasteru)
  - A. NextInput.m (čtení ze vstupů měkké inverze)
  - B. NextOutput.m (zápis na výstupy měkké inverze)
  - C. SetOperators.m (nastavení kombinačních a marginalizačních operátorů)
  - D. SISOB.m (konstruktor)
- xvii. @SISOProperties (Vlastnosti SISO modulu)

- A. SetMOperator.m (nastavení hlavního marginalizačního operátoru)
  - B. SetOperators.m (nastavení kombinačních a marginalizačních operátorů)
  - C. SISOProperties.m (konstruktor)
- xviii. @SVTDecodingNetwork (Zjednodušená iterativní dekodovací síť marginalizující na úrovni symbolů)
- A. CreateNSets.m (definice tvarů a poloh stavových proměnných nad kernelem a definice polohy vstupních dat nad kernelem)
  - B. DecodeBL.m (vykonání 1 iterace spodní vrstvy dekodovací sítě)
  - C. DecodeExtended.m (vykonání 1 iterace dekodovací sítě včetně výpočtu  $\{\mathcal{S}_B(\check{\mathbf{Q}})\}$ )
  - D. DecodeNormal.m (vykonání 1 iterace dekodovací sítě bez výpočtu  $\{\mathcal{S}_B(\check{\mathbf{Q}})\}$ )
  - E. NextInputTL.m (čtení hustot z registrů na propojkách horní úrovně sítě a jejich zápis na vstupy měkké inverze kombinační logiky)
  - F. NextOutput.m (čtení hustot z výstupů měkké inverze broadcasteru a jejich zápis do registrů na propojkách dolní úrovně sítě)
  - G. NextOutputExtendedTL.m (rozšířené čtení hustot z výstupů měkké inverze kombinační logiky a jejich zápis do registrů na propojkách horní úrovně sítě)
  - H. NextOutputTL.m (čtení hustot z výstupů měkké inverze kombinační logiky a jejich zápis do registrů na propojkách horní úrovně sítě)
  - I. Reset.m (nastavení stavových hustot do uniformního tvaru)
  - J. SetMode.m (nastavení selektoru vstupů a výstupů SISO modulu)
  - K. SetTOC.m (nastavení tabulky kombinací SISO modulu na základě zadaného ISI kernelu)
  - L. SVTDecodingNetwork.m (konstruktor)
- xix. @VTDecodingNetwork (Iterativní dekodovací síť marginalizující na úrovni symbolů)
- A. CreateNSets.m (definice poloh stavových proměnných nad kernelem a definice polohy vstupních dat nad kernelem)
  - B. DecodeBL.m (vykonání 1 iterace spodní vrstvy dekodovací sítě)
  - C. NextInput.m (čtení hustot z registrů na propojkách dolní úrovně sítě a jejich zápis na vstupy měkké inverze broadcasteru)
  - D. NextOutput.m (čtení hustot z výstupů měkké inverze broadcasteru a jejich zápis do registrů na propojkách dolní úrovně sítě)
  - E. NextOutputExtendedTL.m (rozšířené čtení hustot z výstupů měkké inverze kombinační logiky a jejich zápis do registrů na propojkách horní úrovně sítě)
  - F. NextOutputTL.m (čtení hustot z výstupů měkké inverze kombinační logiky a jejich zápis do registrů na propojkách horní úrovně sítě)
  - G. Reset.m (nastavení stavových hustot do uniformního tvaru)
  - H. SetTOC.m (nastavení tabulky kombinací SISO modulu na základě zadaného ISI kernelu)
  - I. VTDecodingNetwork.m (konstruktor)
- xx. @WriteSTATS (Zápis výsledků analýzy výkonnosti do souboru)
- A. WriteHeader.m (zápis hlavičky souboru)
  - B. WriteMeasureHeader.m (zápis hlavičky měření chybovosti)
  - C. WriteMeasureSTATS.m (zápis výsledků měření chybovosti v jednotlivých iteracích)
  - D. WriteNSets.m (zápis tvaru použitých stavových a pomocných proměnných)
  - E. WriteSTATS.m (konstruktor)
3. Adresář **Measurement** obsahuje výsledky měření chybovosti jednotlivých iterativních detekčních sítí pro následné analýzy výkonnosti

## B. Vypis m-skriptů klíčových tříd toolboxu

### M-skript 1 SISO modul.

```
1 classdef SISO < handle
2 % Soft-In Soft-Out module
3 %
4 % Constructor: SISO = SISO(Properties,'PropertyName',PropertyValue,...)
5 %
6 % Properties <- Object-handle (SISO properties)
7 %
8 % PropertyName,   PropertyValue (default value "")
9 % -----
10 % InputSelector, Row boolean vector -> {0,1} (On/off switch of input
11 %                variables in the SISO.Properties.TOC - example: [1 1 0]
12 %                means that 1st and 2nd variable (column in the TOC) will
13 %                be combined), "ones(1,SISO.Properties.TOCSize(2))"
14 % OutputSelector, Row boolean vector -> {0,1} (On/off switch of output
15 %                variables in the SISO.Properties.TOC - example: [1 0 1]
16 %                means that 1st and 3rd variable (column in the TOC) will
17 %                be marginalized), "ones(1,SISO.Properties.TOCSize(2))"
18 % -----
19 %
20 % Protected Methods:
21 %
22 % 1) SISO.Set
23 %    Setting of the SISO; automatically triggered
24 %    by SISO.Properties
25 %
26 % 2) SISO.SetIP
27 %    Setting of the SISO - indexing of active inputs;
28 %    automatically triggered by SISO.InputSelector
29 %
30 % 3) SISO.SetOP
31 %    Setting of the SISO - indexing of active outputs;
32 %    automatically triggered by SISO.OutputSelector
33 %
34 % 4) SISO.Next: SISO.Output <- SISO.Input
35 %    Next step of the SISO; automatically triggered by SISO.Input
36 %
37 %    SISO.Input, SISO.Output <- Cell array,
38 %                                Length = SISO.Properties.TOCSize(2)
39 %                                (Soft-Out measures - cells: densities)
40 %
41 % 5) SISO.InitializeSM: SISO.ASM
42 %    Initialization of the a posteriori soft measures
43 %
44 % 6) SISO.CombineSM(TOC): SISO.ASM <- SISO.Input
45 %    Combining of the input soft measures to the a posteriori soft
46 %    measures
47 %
48 %    TOC <- Matrix, Integers >= 1
49 %            (Table of combinations incremented by 1)
50 %
51 % 7) SISO.MarginalizeSM(TOC): SISO.Output <- SISO.ASM
52 %    Marginalization of the a posteriori soft measures to the output
53 %    soft measures
54 %
55 %    TOC <- Matrix, Integers >= 1
56 %            (Table of combinations incremented by 1)
57 %
58 %
59 % ©Daniel Kekrt, Dept. of Radio Engineering,
60 % Faculty of Electrical Engineering, Czech technical university in Prague
61 % rev. 13.10.2013
```

```

62
63 properties (SetAccess = private)
64     IPIs % Current indexes of input pointers to the SISO TOC
65     OPIs % Current indexes of output pointers to the SISO TOC
66
67     InputLength % Current input length
68     OutputLength % Current output length
69
70     DS          % Domain selector 'probability'=-1/'metric'=1
71     ASM         % A posteriori soft measures
72 end
73
74 properties (SetObservable, AbortSet)
75     Properties % SISO module properties list
76             % (Object of SISOProperties class)
77
78     InputSelector % Current pointers to inputs in the SISO TOC
79     OutputSelector % Current pointers to outputs in the SISO TOC
80 end
81
82 properties (SetObservable)
83     Input % Current input soft measures
84 end
85
86 properties
87     Output % Current output soft measures
88 end
89
90 % PUBLIC METHODS -----
91 methods
92     function SISO = SISO(Properties, varargin)
93         addlistener(SISO, 'Input', 'PostSet', @SISO.Next);
94         addlistener(SISO, 'Properties', 'PostSet', @SISO.Set);
95         addlistener(SISO, 'InputSelector', 'PostSet', @SISO.SetIP);
96         addlistener(SISO, 'OutputSelector', 'PostSet', @SISO.SetOP);
97
98         SISO.Properties = Properties;
99         SISO.InputSelector = ones(1, SISO.Properties.TOCSIZE(2));
100        SISO.OutputSelector = zeros(1, SISO.Properties.TOCSIZE(2));
101
102        for LoopInput = 1:2:length(varargin)
103            switch varargin{LoopInput}
104                case 'InputSelector'
105                    SISO.InputSelector = varargin{LoopInput+1};
106                case 'OutputSelector'
107                    SISO.OutputSelector = varargin{LoopInput+1};
108            end
109        end
110    end
111
112    % functions in the separated files -----
113    % -----
114 end
115 % -----
116
117
118 % PROTECTED (PRIVATE) METHODS -----
119 methods (Access = protected)
120     function Set(SISO, ~, ~)
121         switch SISO.Properties.Domain
122             case 'probability'
123                 SISO.DS = 0;
124                 SISO.ASM = ones(SISO.Properties.TOCSIZE(1), 1);
125             case 'metric'
126                 SISO.DS = 1;

```

```

127         SISO.ASM = zeros(SISO.Properties.TOCSize(1),1);
128     end
129 end
130
131 function SetIP(SISO,~,~)
132     SISO.IPIs = find(SISO.InputSelector);
133     SISO.InputLength = length(SISO.IPIs);
134 end
135
136 function SetOP(SISO,~,~)
137     SISO.OPIs = find(SISO.OutputSelector);
138     SISO.OutputLength = length(SISO.OPIs);
139 end
140
141 function Next(SISO,~,~)
142     % Initialization
143     SISO.InitializeSM;
144
145     % TOC Initialization
146     TOC = SISO.Properties.TOC+1;
147
148     % Combination
149     SISO.CombineSM(TOC);
150
151     % Marginalization
152     SISO.MarginalizeSM(TOC);
153 end
154
155 % functions in the separated files -----
156 InitializeSM(SISO)
157
158 CombineSM(SISO,TOC)
159
160 MarginalizeSM(SISO,TOC)
161 % -----
162 end
163 % -----
164 end

```

---

#### M-skript 2 SISO modul: Inicializace modulu.

---

```

1 % ©Daniel Kekrt, Dept. of Radio Engineering,
2 % Faculty of Electrical Engineering, Czech technical university in Prague
3 % rev. 11.8.2013
4
5 function InitializeSM(SISO)
6     SISO.ASM = zeros(SISO.Properties.TOCSize(1),1).^SISO.DS;
7
8     SISO.Output = cell(1,SISO.Properties.TOCSize(2));
9     for LoopIOs = 1:SISO.Properties.TOCSize(2)
10        SISO.Output{LoopIOs} = Inf(size(SISO.Input{LoopIOs})).^SISO.DS;
11    end
12 end

```

---



---

### M-skript 3 SISO modul: Kombinace vstupních hustot.

---

```
1 % ©Daniel Kekrt, Dept. of Radio Engineering,
2 % Faculty of Electrical Engineering, Czech technical university in Prague
3 % rev. 11.8.2013
4
5 function CombineSM(SISO,TOC)
6     % Loop over particular I/O variables
7     for LoopIOs = 1:SISO.InputLength
8
9         % Current input I/O variable index
10        IPI = SISO.IPIs(LoopIOs);
11
12        SISO.ASM = SISO.Properties.C(SISO.ASM,SISO.Input{IPI}(TOC(:,IPI)));
13    end
14 end
```

---

---

### M-skript 4 SISO modul: Marginalizace sdružených hustot.

---

```
1 % ©Daniel Kekrt, Dept. of Radio Engineering,
2 % Faculty of Electrical Engineering, Czech technical university in Prague
3 % rev. 11.1.2014
4
5 function MarginalizeSM(SISO,TOC)
6     % Loop over particular I/O variables
7     for LoopIOs = 1:SISO.OutputLength
8
9         % Current output I/O variable index
10        OPI = SISO.OPIs(LoopIOs);
11
12        % Loop over particular values of the current variable
13        for LoopIOVs = unique(TOC(:,OPI))'
14
15            % Current output I/O variable value indexes
16            OPVIs = find(TOC(:,OPI) == LoopIOVs);
17            SISO.Output{OPI}(LoopIOVs) = SISO.ASM(OPVIs(1));
18
19            % Loop over current value of the current variable
20            for LoopIOV = 2:length(OPVIs)
21
22                SISO.Output{OPI}(LoopIOVs) = ...
23                SISO.Properties.M(SISO.Output{OPI}(LoopIOVs), ...
24                SISO.ASM(OPVIs(LoopIOV)));
25            end
26
27            % Input soft measure arithmetic stabilization
28            ISM = SISO.Input{OPI}(LoopIOVs);
29            ISM = SISO.Properties.AS(ISM,SISO.Properties.SMT);
30
31            % Eliminating of the input soft measures
32            SISO.Output{OPI}(LoopIOVs) = ...
33            SISO.Properties.C2(SISO.Output{OPI}(LoopIOVs),ISM);
34        end
35
36        % Scaling
37        SISO.Output{OPI} = SISO.Properties.C2(SISO.Output{OPI}, ...
38        SISO.Properties.M2(SISO.Output{OPI}));
39    end
40 end
```

---

## M-skript 5 Vlastnosti SISO modulu.

```

1 classdef SISOProperties < handle
2 % Soft-In Soft-Out module properties list
3 %
4 % Constructor: SISOP = SISOProperties(LOC, 'PropertyName', PropertyValue, ...)
5 %
6 % LOC <- Matrix, Integers >= 0
7 % (Table of combinations - columns: I/O variables;
8 % rows: possible combinations of I/O variables)
9 %
10 % PropertyName, PropertyValues (default value "")
11 % -----
12 % Domain, String {'probability'/'metric'}
13 % (Computational domain of the SISO)
14 % Detection, String {'symbol'/'sequence' = 'page'}
15 % (Detection technique of the SISO)
16 % SMT, Scalar (Required maximum/minimum of the input soft
17 % measures in the 'metric'/'probability' domain for
18 % provision of the arithmetic stability), "1e4/1e-4"
19 % -----
20 %
21 % Protected Methods:
22 %
23 % 1) SISOP.Set
24 % Setting of the SISOP.LOCSize; automatically triggered by SISOP.LOC
25 %
26 % 2) SISOP.SetOperators
27 % Setting of the anonymous functions (operators) SISOP.C, SISOP.C2,
28 % SISOP.M2 and SISOP.AS and default value of the SISOP.SMT
29 %
30 % 3) SISOP.SetMOperator
31 % Setting of the anonymous function (operator) SISOP.M
32
33
34 % ©Daniel Kekrt, Dept. of Radio Engineering,
35 % Faculty of Electrical Engineering, Czech technical university in Prague
36 % rev. 5.1.2014
37
38 properties (SetAccess = private)
39     Domain = 'metric' % 'probability'/'metric'
40     Detection = 'sequence' % 'symbol'/'sequence'
41
42     % Combination operators
43     C
44     C2
45
46     % Marginalization operators
47     M
48     M2
49
50     AS % Aritmetic stabilization operator
51
52     SMT % Soft measure threshold for AS operator
53
54     LOCSize % Table of combinations size
55 end
56
57 properties (SetObservable, AbortSet)
58     LOC % Table of combinations
59 end
60
61 % PUBLIC METHODS -----
62 methods
63     function SISOP = SISOProperties(LOC, varargin)
64         addlistener(SISOP, 'LOC', 'PostSet', @SISOP.Set);

```

```

65
66     for LoopInput = 1:2:length(varargin)
67         switch varargin{LoopInput}
68             case 'Domain'
69                 SISOP.Domain = varargin{LoopInput+1};
70             case 'Detection'
71                 SISOP.Detection = varargin{LoopInput+1};
72             end
73         end
74
75         SISOP.TOC = TOC;
76
77         % Setting of the C, C2, M2, AS operators and default SMT
78         SISOP.SetOperators;
79
80         % Setting of the M operator
81         SISOP.SetMOperator;
82
83         for LoopInput = 1:2:length(varargin)
84             switch varargin{LoopInput}
85                 case 'SMT'
86                     SISOP.SMT = varargin{LoopInput+1};
87             end
88         end
89     end
90
91     % functions in the separated files -----
92     % -----
93 end
94 % -----
95
96
97 % PROTECTED (PRIVATE) METHODS -----
98 methods (Access = protected)
99     function Set(SISOP,~,~)
100         SISOP.TOCSize = size(SISOP.TOC);
101     end
102
103     % functions in the separated files -----
104     SetOperators(SISOP)
105
106     SetMOperator(SISOP)
107     % -----
108 end
109 % -----
110 end

```

---

#### M-skript 6 Vlastnosti SISO modulu: Nastavení kombinačních a marginalizačních operátorů.

```

1 % ©Daniel Kekrt, Dept. of Radio Engineering,
2 % Faculty of Electrical Engineering, Czech technical university in Prague
3 % rev. 11.1.2014
4
5 function SetOperators(SISOP)
6     switch SISOP.Domain
7         case 'probability'
8             SISOP.C = @(x,y) x.*y;
9             SISOP.C2 = @(x,y) x./y;
10            SISOP.M2 = @sum;
11            SISOP.AS = @(x,y) x+(x<y)*(y-x);
12            SISOP.SMT = 1e-4;
13
14        case 'metric'
15            SISOP.C = @(x,y) x+y;

```

```

16         SISOP.C2 = @(x,y) x-y;
17         SISOP.M2 = @min;
18         SISOP.AS = @(x,y) x+(x>y)*(y-x);
19         SISOP.SMT = 1e4;
20     end
21 end

```

---

#### M-skript 7 Vlastnosti SISO modulu: Nastavení hlavního marginalizačního operátoru.

---

```

1  % ©Daniel Kekrt, Dept. of Radio Engineering,
2  % Faculty of Electrical Engineering, Czech technical university in Prague
3  % rev. 5.1.2014
4
5  function SetMOperator(SISOP)
6      switch strcat(SISOP.Domain,SISOP.Detection)
7          case 'probabilitysymbol'
8              SISOP.M = @(x,y) sum([x y]);
9
10         case {'probabilitysequence','probabilitypage'}
11             SISOP.M = @(x,y) max([x y]);
12
13         case 'metricsymbol'
14             SISOP.M = @(x,y) min(x,y)-log(1+exp(-abs(x-y)));
15
16         case {'metricsequence','metricpage'}
17             SISOP.M = @(x,y) min([x y]);
18     end
19 end

```

---

#### M-skript 8 Měkká inverze broadcasteru.

---

```

1  classdef SISOB < handle
2  % Soft-In Soft-Out broadcaster
3  %
4  % Constructor: SISOB = SISOB('PropertyName',PropertyValue,...)
5  %
6  % Properties <- Object-handle (SISO properties)
7  %
8  % PropertyName, PropertyValues (default value "")
9  %
10 % -----
11 % Domain,          String {'probability'/'metric'}
12 % SMT,             Scalar (Required maximum/minimum of the input soft
13 %                  measures in the 'metric'/'probability' domain for
14 %                  provision of the arithmetic stability), "1e4/1e-4"
15 % -----
16 % Protected Methods:
17 %
18 % 1) SISOB.SetOperators
19 %    Setting of the anonymous functions (operators) SISOB.C, SISOB.C2,
20 %    SISOB.SC and SISOB.AS and default value of the SISOB.SMT
21 %
22 % 2) [ISM] = SISOB.NextInput
23 %    Transformation of the SISOB.Input to the matrix form iff SISOB.Input
24 %    is in the cell array form; automatically triggered by SISOB.Input
25 %
26 %    ISM <- Matrix (Soft-In measures - columns: densities)
27 %
28 % 3) SISOB.NextOutput
29 %    Transformation of the SISOB.Output to the cell array form iff
30 %    SISOB.Input is the cell array form
31 %

```

```

32 % 4) SISOB.Next: SISOB.Output <- SISOB.Input
33 % Next steps of the SISOB; automatically triggered by SISOB.Input
34 %
35 % SISOB.Input <- Cell array or matrix (Soft-In measures - cells or
36 % columns: densities)
37 % SISOB.Output <- Cell array or matrix (Soft-Out measures - cells or
38 % columns: densities)
39 %
40 %
41 % ©Daniel Kekrt, Dept. of Radio Engineering,
42 % Faculty of Electrical Engineering, Czech technical university in Prague
43 % rev. 5.1.2014
44 %
45 properties (SetAccess = private)
46     Domain = 'metric' % Detection domain 'metric'/'probability'
47
48     InputSize % Current input size
49
50     C % Combination operator
51     C2 % Combination operator
52     SC % Scaling operator
53     AS % Arithmetic stabilization operator
54     SMT % Soft measure threshold for AS operator
55
56     ASM % A posteriori soft measures
57 end
58
59 properties (SetObservable)
60     Input % Current input soft measures
61 end
62
63 properties
64     Output % Current output soft measures
65 end
66
67 % PUBLIC METHODS -----
68 methods
69     function SISOB = SISOB(varargin)
70         addlistener(SISOB,'Input','PostSet',@SISOB.Next);
71
72         for LoopInput = 1:2:length(varargin)
73             switch varargin{LoopInput}
74                 case 'Domain'
75                     SISOB.Domain = varargin{LoopInput+1};
76             end
77         end
78
79         % Setting of the C, C2, SC, AS operators and default SMT
80         SISOB.SetOperators;
81
82         for LoopInput = 1:2:length(varargin)
83             switch varargin{LoopInput}
84                 case 'SMT'
85                     SISOB.SMT = varargin{LoopInput+1};
86             end
87         end
88     end
89
90     % functions in the separated files -----
91     % -----
92 end
93 % -----
94
95 % PROTECTED (PRIVATE) METHODS -----
96

```

```

97     methods (Access = protected)
98         function Next(SISOB,~,~)
99             % Initialization
100            ISM = SISOB.NextInput;
101            SISOB.InputSize = size(ISM);
102
103            % Combination
104            SISOB.ASM = SISOB.C(ISM);
105
106            % Aritmetic stabilization of the input soft measures
107            ISM = SISOB.AS(ISM,SISOB.SMT*ones(SISOB.InputSize));
108
109            % Eliminating of the input soft measures
110            OSM = SISOB.C2(ISM);
111
112            % Scaling
113            SISOB.Output = SISOB.SC(OSM);
114            SISOB.NextOutput;
115        end
116
117        % functions in the separated files -----
118        SetOperators(SISOB)
119
120        [ISM] = NextInput(SISOB)
121
122        NextOutput(SISOB)
123        % -----
124    end
125    % -----
126 end

```

---

#### M-skript 9 Měkká inverze broadcasteru: Nastavení operátorů.

```

1 % ©Daniel Kekrt, Dept. of Radio Engineering,
2 % Faculty of Electrical Engineering, Czech technical university in Prague
3 % rev. 3.1.2014
4
5 function SetOperators(SISOB)
6     switch SISOB.Domain
7         case 'probability'
8             SISOB.C = @(x) prod(x,2);
9             SISOB.C2 = @(x) ...
10                repmat(SISOB.ASM,1,SISOB.InputSize(2))./x;
11             SISOB.SC = @(x) x./repmat(sum(x),SISOB.InputSize(1),1);
12
13             SISOB.AS = @(x,y) x+(x<y).* (y-x);
14             SISOB.SMT = 1e-4;
15
16         otherwise
17             SISOB.C = @(x) sum(x,2);
18             SISOB.C2 = @(x) ...
19                repmat(SISOB.ASM,1,SISOB.InputSize(2))-x;
20             SISOB.SC = @(x) x-repmat(min(x),ISOB.InputSize(1),1);
21
22             SISOB.AS = @(x,y) x+(x>y).* (y-x);
23             SISOB.SMT = 1e4;
24         end
25 end

```

---

### M-skript 10 Měkká inverze broadcasteru: Čtení ze vstupů.

---

```
1 % ©Daniel Kekrt, Dept. of Radio Engineering,  
2 % Faculty of Electrical Engineering, Czech technical university in Prague  
3 % rev. 3.1.2014  
4  
5 function [ISM] = NextInput(SISOB)  
6     if iscell(SISOB.Input)  
7         ISM = cell2mat(SISOB.Input);  
8     else  
9         ISM = SISOB.Input;  
10    end  
11 end
```

---

### M-skript 11 Měkká inverze broadcasteru: Zápis na výstupy.

---

```
1 % ©Daniel Kekrt, Dept. of Radio Engineering,  
2 % Faculty of Electrical Engineering, Czech technical university in Prague  
3 % rev. 3.1.2014  
4  
5 function NextOutput(SISOB)  
6     if iscell(SISOB.Output)  
7         SISOB.Output = num2cell(SISOB.Output,1);  
8     end  
9 end
```

---

## Reference

- [1] KEKRT, D., LUKEŠ, T., KLÍMA, M., FLIEGEL, K., 2D Iterative MAP Detection: Principles and Applications in Image Restoration. *Radioengineering*, vol. 23, no. 2, 2014, p. 618-631.
- [2] LUKEŠ, T., KEKRT, D., KLÍMA, M., FLIEGEL, K., Binarization of Noisy Microscopy Images through Signal Reconstruction Using Iterative Detection Network. In *Proceedings of the International Conference on Image Processing (ICIP)*. Paris (France), 2014.
- [3] KEKRT, D., KLÍMA, M., 2D Iterative Detection Network Based Image Restoration: Principles, Applications and Performance Analysis. In *Image Restoration - Recent Advances and Applications*, ISBN 978-953-51-0388-2, InTech, 2012, chapter 14, p. 315-350.
- [4] KEKRT, D., KLÍMA, M., FLIEGEL, K., Restoration of Dichromatic Images Gained from CCD/CMOS Camera by Iterative Detection Networks with fragmented marginalization at the symbol block level. In *Proceedings of SPIE 7444, Mathematics for Signal and Information Processing, 74441A*. San Diego (USA), 2009, p. 74441A-1-74441A-12.
- [5] KEKRT, D., KLÍMA, M., FLIEGEL, K., The Iterative Detection Network Based Suppression of the Thermal Noise and Blurring due to Object Moving in Black and White Pictures Shot by a Camera with CCD/CMOS sensor. In *Proceedings of SPIE 7076, Optics and Photonics*. San Diego (USA), 2008, p. 70760M-1-70760M-9.
- [6] KEKRT, D., KLÍMA, M., PODGORNÝ, R., The Iterative Detection Network Suppression of Defocusing and Thermal Noise in Black and White Pictures Shot By a Camera with CCD/CMOS Sensor. In *Photonics Prague 2008, The 6th International Conference on Photonics, Devices and Systems*, 71381Z. Prague (Czech Republic), 2008, p. 71381Z-1-71381Z-7.
- [7] KEKRT, D., KLÍMA, M., The Advanced Noise Model for an IDN Based Restoration of Black and White Pictures Captured by a Camera with CCD/CMOS Sensor. In *Proceeding of the 42nd Annual IEEE International Carnahan Conference on Security Technology, ICCST 2008*. Prague (Czech Republic), 2008, p. 126-130.
- [8] KEKRT, D., KLÍMA, M., A Black & White Picture Reconstruction by Iterative Detection Network in the Image Capturing System with CCD (CMOS) Sensor. In *Proceedings of 50th International Symposium ELMAR 2008*. Zadar (Croatia), 2008, p. 125-128.
- [9] THIENNVIBOON, P., ORTEGA, A., CHUGG, K., Simplified Grid Message-passing Algorithm with Application to Digital Image Halftoning. In *Proceedings of the International Conference on Image Processing, 2001*. Thessaloniki (Greece), 2001, p. 1061-1064 volume 2.
- [10] THIENNVIBOON, P., CHUGG, K., Model-based Digital Image Halftoning Using Iterative Reduced-complexity Grid Message-passing Algorithm. In *Proceedings of SPIE 5008, Color Imaging: Processing, Hardcopy and Applications VIII*. Santa Clara (USA), 2003, p. 419-430.
- [11] CHEN, X., CHUGG, K., Near-optimal Page Detection for Two-dimensional ISI/AWGN Channels Using Concatenated Modeling and Iterative Detection. In *Proceedings of the International Conference on Communications*. Atlanta (USA), 1998. p. 952-956 volume 2.
- [12] CHUGG, K., ANASTASOPOULOS, A., CHEN, X. *Iterative detection: Adaptivity, Complexity reduction and Applications*. ISBN 0-729-37277-8, Kluwer Academic Publishers, 2001.
- [13] VUCETIC, B., JUAN, J. *Space-Time coding*. ISBN 0-470-84757-3, Kluwer Academic Publishers, 2003.
- [14] VUCETIC, B., JUAN, J. *Turbo codes: Principles and Applications*. ISBN 0-792-37868-7, Kluwer Academic Publishers, 2000.



- [15] SCHLEGEL, CH., PEREZ L., *Trellis and Turbo Coding*. ISBN 978-0-471-22755-7, Wiley-IEEE Press, 2004.
- [16] ABBASFAR, A., *Turbo-like Codes: Design for High Speed Decoding*. ISBN 978-1-402-06390-9, Springer, 2007.
- [17] PROAKIS, J., SALEHI, M., *Digital Communications*. ISBN 978-0-072-95716-7 , McGraw-Hill Science/Engineering/Math, 5th edition, 2007.
- [18] GOLDSMITH, A., *Wireless Communications*. ISBN 978-0-521-83716-3, Cambridge University Press, 2005.
- [19] VLIET, L. J., BODDEKE, F. R., SUDAR, D., YOUNG, I. T., Image Detectors for Digital Image Microscopy. In *Digital Image Analysis of Microbes; Imaging, Morphometry, Fluorometry and Motility Techniques and Applications, Modern Microbiological Methods*. Chichester (UK), 1998, p. 37-64.
- [20] BERROU, C., GLAVIEUX, A., THITMAJSHIMA, P., Near Shannon Limit Errorcorrecting Coding and Decoding: Turbo-codes. In *Proceedings of the International Conference on Communications*. Geneva (Switzerland), 1993, p. 1064-1070.
- [21] BERROU, C., GLAVIEUX, A., Near Optimal Error-correcting Coding and Decoding: Turbo codes. *IEEE Transactions on Communication*, vol. COM-44, no. 10, 1996, p. 1261-1271.
- [22] BAHL, L. R., COCKE., J., JELINEK, F., RAVIV, J., Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate. *IEEE Transactions on Information Theory*, vol. IT-20., 1974, p. 284-287.
- [23] HAGENAUER, J., HOEHER, P., A Viterbi Algorithm with Soft-decision Outputs and Its Applications. In *Proceedings of IEEE Global Telecommunications Conference 1989*. Dallas (USA), 1989, p. 1680-1686.
- [24] HAGENAUER, J., Rate Compatible Punctured Convolutional Codes (RCPC codes) and their Application. *IEEE Transactions on Communication*, vol. COM-36, 1988, p. 389-400.
- [25] FORNEY, J. D., The Viterbi Algorithm. *Proceedings of the IEEE*, vol. 61, no. 3, 1973, p. 268-278.
- [26] SHANNON, C., E., Communication in the Presence of Noise. In *Proceedings of the IEEE*, vol. 86, issue 2, 1998, p. 447 - 457 .
- [27] UNGERBOECK, G., Channel Coding with Multilevel/Phase Signals. *IEEE Transactions on Information Theory*, vol. IT-28, no. 1, 1982, p. 55-67.
- [28] UNGERBOECK, G., Trellis-coded Modulation with Redundant Signal Sets part I: Introduction. *IEEE Communication Mag.*, vol. 25, no.2, 1987, p. 5-11.
- [29] UNGERBOECK, G., Trellis-coded Modulation with Redundant Signal Sets part II: State of the art. *IEEE Communication Mag.*, vol. 25, no.2, 1987, p. 12-21.
- [30] SANKUR, B., SEZGIN, M., Survey Over Image Thresholding Techniques and Quantitative Performance Evaluation. *Journal of Electronic Imaging*, 2004, p. 146-165.
- [31] GOODMAN, J. W., *Introduction to Fourier Optics*. ISBN 0-974-70772-4, New York: McGraw-Hill Int. Ed., 1996.
- [32] GONZALEZ, R. C., WOODS R. E., *Digital Image Processing*. ISBN 978-0-131-68728-8, Prentice Hall, 3th edition, 2007.

- [33] STALLINGA, S., RIEGER B., Accuracy of the Gaussian Point Spread Function Model in 2D Localization Microscopy. *Opt. Express* 18 (24), 2010, p. 24461-24476.
- [34] KLÍMA, M., KEKRT, D., PODGORNÝ, R., Aspects of Image Quality Enhancement in Security Technology. In *Proceeding of the IEEE International Carnahan Conference on Security Technology 2008*. Prague (Czech Republic), 2008, p. 142-149.
- [35] KLÍMA, M., FLIEGEL, K., KEKRT, D., DOSTÁL, P., PODGORNÝ, R., Image Quality and QoE in Multimedia Systems. In *Proceeding of the 19th International Conference Radioelektronika 2009*. Bratislava (Slovak Republic), 2009, p. 3-10.
- [36] KEKRT, D., Layered Iterative Detection Network Based Text Recognition in the Snapshot Gained by Camera with CCD (CMOS) Sensor. In *Proceedings of the 12th International Student Conference on Electrical Engineering POSTER 2008*. [CD-ROM], Prague (Czech Republic), 2008.
- [37] OTSU, N., A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, 1979, p. 62-66.
- [38] NIBLACK, W., *An Introduction to Digital Image Processing*. Prentice-Hall, 1986, p. 115-116.