

UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação



Dissertação

**Controlador Dinâmico de Energia para o Codificador HEVC Baseado em
Frentes de Pareto**

Wagner Ishizaka Penny

Pelotas, 2016

Wagner Ishizaka Penny

**CONTROLADOR DINÂMICO DE ENERGIA PARA O CODIFICADOR HEVC
BASEADO EM FRETES DE PARETO**

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal de Pelotas, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Bruno Zatt

Co-orientador: Prof. Dr. Marcelo Schiavon Porto

Co-orientador: Prof. Dr. Luciano Volcan Agostini

Pelotas, 2016.

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

P416c Penny, Wagner Ishizaka

Controlador dinâmico de energia para o codificador HEVC baseado em frentes de pareto / Wagner Ishizaka Penny ; Bruno Zatt, orientador ; Marcelo Schiavon Porto, Luciano Volcan Agostini, coorientadores. — Pelotas, 2016.

90 f. : il.

Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2016.

1. HEVC. 2. RAPL. 3. Análise do consumo de energia. 4. Codificação de vídeo. 5. Controle de energia. I. Zatt, Bruno, orient. II. Porto, Marcelo Schiavon, coorient. III. Agostini, Luciano Volcan, coorient. IV. Título.

CDD : 005

Wagner Ishizaka Penny

**Controlador Dinâmico de Energia para o Codificador HEVC Baseado em
Frentes de Pareto**

Dissertação aprovada, como requisito parcial, para obtenção do grau de Mestre em
Ciência da Computação, Programa de Pós-Graduação em Computação, Centro de
Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

Data da Defesa: 25/02/2016

Banca examinadora:

.....
Prof. Dr. Bruno Zatt (Orientador)
Doutor em Microeletrônica pela Universidade Federal do Rio Grande do Sul

.....
Prof. Dr. Marcelo Schiavon Porto (Co-orientador)
Doutor em Computação pela Universidade Federal do Rio Grande do Sul

.....
Prof. Dr. Luciano Volcan Agostini (Co-orientador)
Doutor em Computação pela Universidade Federal do Rio Grande do Sul

.....
Prof. Dr. Cláudio Machado Diniz
Doutor em Computação pela Universidade Federal do Rio Grande do Sul

.....
Prof. Dr. Felipe de Souza Marques
Doutor em Computação pela Universidade Federal do Rio Grande do Sul

.....
Prof. Dr. Júlio Carlos Balzano de Mattos
Doutor em Computação pela Universidade Federal do Rio Grande do Sul

Dedico este trabalho ao meu filho Thiago, a pessoa mais importante da minha vida, um dia tu entenderás o quanto o papai te ama.

Agradecimentos

Agradeço em primeiro lugar a Deus, por ter me dado forças para realizar este trabalho.

À Universidade Federal de Pelotas pelos ótimos programas de pós-graduação ofertados, oferecendo a possibilidade de profissionais graduados seguirem no caminho da pesquisa.

Aos professores Bruno Zatt, Marcelo Porto e Luciano Agostini pela orientação neste trabalho, foram de fato orientadores, presentes e incansáveis ao longo deste processo, com certeza este trabalho não teria sido possível sem eles. Agradeço também ao professor Guilherme Corrêa, que, apesar de não ser meu orientador formal, atuou de igual maneira aos demais.

Ao professor Vladimir Afonso por me apresentar ao PPGC e me incentivar nessa mudança de ares, com certeza sem seu incentivo eu não teria tomado a iniciativa de entrar para o mestrado.

Aos professores André Arthur Perleberg Lerm e Adão Antônio de Souza Júnior pelas cartas de recomendação para ingresso no programa de mestrado.

Agradeço também a todos os demais professores do PPGC pelos seus ensinamentos, extremamente valiosos na minha formação.

Aos colegas do GACI pela parceria, especialmente aos colegas Guilherme Paim e Ítalo Machado, com certeza a ajuda e parceria de vocês foi fundamental e extremamente valiosa para que eu alcançasse os objetivos deste trabalho. Ao colega Mateus Grellert por sua ajuda nas etapas iniciais deste trabalho.

Agradeço minha esposa Janice pelo amor, compreensão, apoio e companheirismo em todas as etapas deste trabalho (e da minha vida). Ao meu filho Thiago por alegrar todos os meus dias. A minha filha emprestada Antonella, por seu amor, carinho e admiração. Ao meu pai Solmar e minha mãe Rosa por todo amor, carinho e dedicação ao longo desses anos. A minha irmã Amanda por seu apoio e carinho. Ao meu sogro Jorge e minha tia Sílvia pelo carinho e incentivo.

Por fim, agradeço a todos que, de uma forma ou de outra, estiveram presentes na minha vida e contribuíram para que este trabalho pudesse ter sido realizado.

*“Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar.
Mas o mar seria menor se lhe faltasse uma gota.”*

(Madre Teresa de Calcutá)

RESUMO

PENNY, Wagner Ishizaka. **Controlador Dinâmico de Energia para o Codificador HEVC Baseado em Frentes de Pareto**. 2016. 90f. Dissertação (Mestrado) - Mestrado em Ciência da Computação. Universidade Federal de Pelotas, Pelotas.

Atualmente, a reprodução de vídeos digitais representa um importante tema de pesquisa. Devido ao rápido crescimento da demanda por maiores resoluções e desempenho, foi desenvolvido o padrão de codificação de vídeo da próxima geração, o *High Efficiency Video Coding* (HEVC). Este padrão se tornou o novo estado da arte em codificação de vídeo, trazendo muitas inovações como forma de aumentar a eficiência de codificação. No entanto, o aumento de desempenho também acarretou no aumento do esforço computacional (complexidade) e do consumo de energia. Na maior parte das vezes os vídeos devem ser manipulados em sistemas embarcados, os quais são sistemas dedicados que possuem uma funcionalidade restrita para atender uma tarefa específica, além de apresentarem como característica, na maioria das vezes, a mobilidade e o uso de pilhas e baterias como fonte de energia. São exemplos de sistemas embarcados os computadores de bordo, câmeras digitais, drones, etc. Dessa forma, é necessário que exista o equilíbrio entre uso de memória, qualidade de vídeo, desempenho e consumo de energia. De nada adiantaria a obtenção de um alto desempenho de codificação se a energia disponível tivesse de ser utilizada de maneira exagerada. Nesse contexto, técnicas de controle de energia são altamente desejáveis, uma vez que tal controle possibilita a adaptação do consumo de energia conforme disponibilidade do sistema e natureza dos dados. Este trabalho apresenta o desenvolvimento de um controlador dinâmico de energia para o codificador de vídeo HEVC baseado em Frentes de Pareto e suportado por uma análise de sensibilidade à variação de parâmetros de codificação em relação ao consumo energético, utilizando a ferramenta RAPL (*Running Average Power Limit*). O controlador desenvolvido permite a adaptação do codificador para cenários de disponibilidade de energia de até 20% em relação à original, e, para uma redução média de 30% no consumo de energia, apresenta em torno de 5% de aumento na taxa de bits.

Palavras-chave: HEVC; RAPL; Análise do Consumo de Energia; Codificação de Vídeo; Controle de Energia.

ABSTRACT

PENNY, Wagner Ishizaka. **Controlador Dinâmico de Energia para o Codificador HEVC Baseado em Frentes de Pareto**. 2016. 90f. Dissertação (Mestrado) - Mestrado em Ciência da Computação. Universidade Federal de Pelotas, Pelotas.

Nowadays, the manipulation of digital videos represents an important research challenge. The fast increase on the demand for higher resolutions and performance leads to the development of the next-generation video coding standard, the High Efficiency Video Coding (HEVC). The HEVC became the new state of the art and brings many innovations as a way to improve the coding efficiency. However, the improvement on performance also brought higher computational effort (complexity) and energy consumption. Commonly, the videos are handled by embedded systems, which are dedicated systems that have a restricted functionality to meet a specific task, and are, typically, portable and battery powered. Onboard computers, digital cameras and drones are examples of embedded systems. This way, the balance between memory usage, video quality, performance and energy consumption is strongly desirable. Thereby, the usage of energy control techniques is mandatory, once such control allows the adaption of the energy consumption according to the system availability and data behavior. This work presents the development of a dynamic energy controller for the HEVC video coding standard based on Pareto Fronts and supported by an analysis that relates energy consumption to coding parameters variation, using the RAPL (Running Average Power Limit) tool. The developed controller allows adapting the encoder to 20% of its original energy consumption, and, for a 30% energy consumption reduction, presents around 5% increase on bit-rate.

Keywords: HEVC; RAPL; Energy Consumption Analysis; Video Coding; Energy Control.

LISTA DE FIGURAS

Figura 1 – Sequência de quadros temporalmente vizinhos.....	21
Figura 2 – Diagrama de blocos do codificador HEVC	23
Figura 3 – Aplicação da transformada e da quantização em um bloco B arbitrário...	24
Figura 4 – Divisão de uma CTU em CUs	26
Figura 5 – Geometria de PUs possíveis.....	26
Figura 6 – Divisão de uma CU em TUs.....	27
Figura 7 – Exemplo de curva R-D	29
Figura 8 – Curvas de (a) BD-BR e (b) BD-PSNR para uma dada aplicação	32
Figura 9 – Representação gráfica da configuração <i>low-delay</i>	34
Figura 10 – Exemplo de uma solução ótima de Pareto.....	35
Figura 11 – Potência VS Energia	37
Figura 12 – Pseudocódigo de leitura de energia do processo de codificação.....	44
Figura 13 – Metodologia paralela de medição de energia.....	45
Figura 14 - Metodologia sequencial de medição de energia	47
Figura 15 – Economia de energia média com relação à c_0	52
Figura 16 – Energia normalizada para cada PS.....	56
Figura 17 – BD-BR para cada PS	56
Figura 18 – BD-PSNR para cada PS	57
Figura 19 – Sistema de controle de energia integrado com o codificador HEVC	58
Figura 20 – Frente de Pareto no espaço D-E.....	60
Figura 21 – Frente de Pareto no espaço R-E.....	61
Figura 22 – Controlador dinâmico de energia do HEVC proposto.....	63
Figura 23 – Pseudocódigo do controlador dinâmico de energia.....	66
Figura 24 – Resultado do controlador para a sequência <i>PeopleonStreet</i> (QP 22)....	69
Figura 25 – Resultado do controlador para a sequência <i>PeopleonStreet</i> (QP 27)....	69
Figura 26 – Resultado do controlador para a sequência <i>BasketballDrill</i> (QP22).....	70
Figura 27 – Resultado do controlador para a sequência <i>Kimono</i> (QP 22)	70
Figura 28 – Resultado do controlador para a sequência <i>ChinaSpeed</i> (QP 27).....	70
Figura 29 – Resultado do controlador para a sequência <i>BlowingBubbles</i> (QP37)....	72
Figura 30 – Número de bits por quadro para a sequência <i>PeopleonStreet</i> (QP 22) .	73
Figura 31 – PSNR por quadro para a sequência <i>PeopleonStreet</i> (QP 22)	73
Figura 32 – Mudança no SP para a sequência <i>PeopleonStreet</i> (QP22)	74

Figura 33 – Mudança no SP para a sequência <i>BasketballDrill</i> (QP22)	74
Figura 34 – Controlador de QP para a sequência <i>PeopleonStreet</i> para SP de 30%	77

LISTA DE TABELAS

Tabela 1 – Configuração para as condições de teste	50
Tabela 2 – Configurações utilizadas na análise de sensibilidade.....	51
Tabela 3 – Energia média consumida e desvio padrão em cada configuração	52
Tabela 4 – Valores médios de BD-BR e BD-PSNR para cada configuração	53
Tabela 5 – Conjunto de parâmetros (PS) propostos	54
Tabela 6 – Resultados para cada PS.....	55
Tabela 7 – Pontos de Controle (PCs) definidos para o controlador	62
Tabela 8 – Configuração de codificação utilizada em cada PC utilizado no controlador	62
Tabela 9 – Tabela de atualização do controlador.....	64
Tabela 10 – Resultados do controlador para SP configurado em 30%	68
Tabela 11 – Tempos de estabilização médios do controlador	75
Tabela 12 – Comparação entre controladores com <i>set point</i> fixo em 30%	76
Tabela 13 – Tempo de estabilização médio para o controlador de QP fixo em 30%	78

LISTA DE ABREVIATURAS E SIGLAS

AMP	<i>Asymmetric Motion Partition</i>
AVC	<i>Advanced Video Coding</i>
AVS	<i>Audio Video coding Standard</i>
BD	<i>Bjøntegaard</i>
BD-BR	<i>Bjøntegaard Bit-rate</i>
BD-PSNR	<i>Bjøntegaard Peak Signal-to-Noise Ratio</i>
CABAC	<i>Context-Adaptive Binary Arithmetic Coding</i>
CPU	<i>Central Processing Unit</i>
CTC	<i>Common Test Conditions</i>
CTU	<i>Coding Tree Unit</i>
CU	<i>Coding Unit</i>
DBF	<i>Deblocking Filter</i>
DE	<i>Distortion Energy</i>
DVFS	<i>Dynamic Voltage and Frequency Scaling</i>
FME	<i>Fractional Motion Estimation</i>
FSM	<i>Finite State Machine</i>
GOP	<i>Group of Pictures</i>
GPB	<i>Generalized P and B Picture</i>
HT	<i>Hadamard Transform</i>
HadME	<i>Hadamard Motion Estimation</i>
HEVC	<i>High Efficiency Video Coding</i>
HM	<i>HEVC Test Model</i>
IDR	<i>Instantaneous Decoding Refresh</i>
ITU-T	<i>International Telecommunication Union – Telecommunication Standardization Sector</i>
JCT-VC	<i>Joint Collaborative Team on Video Coding</i>
MAX	<i>Máximo</i>
MAXCUd	<i>Max Partition Depth</i>
McPAT	<i>Multicore Power Area Timing</i>
ME	<i>Motion Estimation</i>
MPEG	<i>Moving Picture Experts Group</i>
MSE	<i>Mean Squared Error</i>

MSR	<i>Model Specific Register</i>
PC	<i>Ponto de Controle</i>
PS	<i>Parameter Set</i>
PSNR	<i>Peak Signal-to-Noise Ratio</i>
PU	<i>Prediction Unit</i>
QP	<i>Quantization Parameter</i>
RAPL	<i>Running Average Power Limit</i>
RD	<i>Rate Distortion</i>
RDE	<i>Rate Distortion Energy</i>
RDEO	<i>Rate Distortion Energy Optimization</i>
RDO	<i>Rate Distortion Optimization</i>
RE	<i>Rate Energy</i>
RQT	<i>Residual Quad Tree</i>
SAD	<i>Sum of Absolute Differences</i>
SAO	<i>Sample Adaptive Offset</i>
SATD	<i>Sum of Absolute Transformed Differences</i>
SI	<i>Sistema Internacional de Unidades</i>
SO	<i>Sistema Operacional</i>
SoC	<i>System on a Chip</i>
SP	<i>Set Point</i>
SR	<i>Search Range</i>
TU	<i>Transform Unit</i>

LISTA DE SÍMBOLOS

Y	<i>Luminância</i>
C _b	<i>Crominância Azul</i>
C _r	<i>Crominância Vermelha</i>
D	<i>Distorção</i>
R	<i>Taxa de bits</i>
J	<i>Função custo</i>
λ	<i>Multiplicador de Lagrange</i>
P	<i>Potência</i>
V	<i>Tensão Elétrica</i>
ΔE	<i>Varição de Energia</i>
Δt	<i>Varição de Tempo</i>
M	<i>Fator de Redução de Consumo de Energia</i>
E _D	<i>Energia Desejada</i>
E _N	<i>Energia Normalizada</i>
E _M	<i>Energia Medida</i>

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Motivação	18
1.2	Objetivos.....	19
1.3	Organização do Trabalho	20
2	FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS	21
2.1	A Codificação de Vídeo e o Codificador HEVC	21
2.1.1	<i>Fluxo de Codificação no HEVC</i>	22
2.1.2	<i>Estruturas de Particionamento no HEVC</i>	25
2.1.3	<i>Otimização da Taxa-Distorção (Rate-Distortion Optimization - RDO)</i> ..	27
2.1.4	<i>Métricas de Comparação</i>	29
2.1.5	<i>Condições Comuns de Teste</i>	33
2.2	Frentes de Pareto.....	34
2.3	Consumo de Energia em um Sistema Microprocessado	36
2.3.1	<i>Energia e Potência em Circuitos Digitais</i>	36
2.3.2	<i>Ferramentas para Análise do Consumo de Energia</i>	37
2.4	Trabalhos Relacionados.....	40
3	METODOLOGIA UTILIZADA PARA MEDIÇÃO DE ENERGIA NO	
	PROCESSO DE CODIFICAÇÃO DO HEVC	43
3.1	Utilização do RAPL	44
3.2	Integração com o HM.....	46
4	ANÁLISE DE SENSIBILIDADE DO CONSUMO DE ENERGIA DO	
	CODIFICADOR HEVC	48
4.1	Parâmetros de Codificação do Codificador HEVC.....	48
4.2	Resultados da Análise de Sensibilidade.....	49
5	CONTROLADOR DINÂMICO DE ENERGIA APLICADO AO	
	CODIFICADOR HEVC	58
5.1	Determinação dos pontos de operação do codificador	58
5.2	Sistema de Controle de Energia Proposto	63
6	RESULTADOS E DISCUSSÕES	67
6.1	Comportamento do controlador para cada sequência	68
6.2	Resultados para adaptação à mudança de SP	74
6.3	Comparação com outros controladores	76
7	CONCLUSÃO E TRABALHOS FUTUROS	79

7.1	Trabalhos Futuros.....	80
	REFERÊNCIAS.....	82
	APÊNDICES	87
	ANEXOS	89

1 INTRODUÇÃO

Nos últimos tempos os dispositivos digitais atingiram elevado grau de desenvolvimento e complexidade. Cada vez mais a busca por redução de consumo de energia e aumento de desempenho são pontos chave no desenvolvimento de qualquer sistema. Neste cenário podem-se enquadrar os sistemas embarcados e todos os dispositivos que se utilizam desse tipo de sistema.

Segundo Wolf (2005), sistemas embarcados são sistemas com caráter dedicado e possuem funcionalidade específica dentro de um sistema mais complexo. Já para Marwedel (2006) são sistemas dedicados que possuem uma funcionalidade restrita para atender uma tarefa específica em sistemas maiores nos quais estão inseridos. Ainda, de acordo com Hennessy e Patterson (2007), sistemas embarcados são sistemas embutidos em sistemas maiores e sua presença nem sempre é óbvia; representando a parcela de maior crescimento do mercado de computadores dos últimos tempos. Dessa forma, diferentemente de sistemas computacionais de propósito geral, sistemas embarcados são de aplicação específica e devem atender restrições rígidas de desempenho, dentre as quais se destaca o consumo de energia. Embora possuam essas características, as soluções de análises realizadas em sistemas computacionais de propósito geral podem ser estendidas para sistemas embarcados (CARRO, 2003).

Aplicações com vídeos digitais são exemplos típicos desse cenário. Segundo Grellert (2014), a antiga tendência de compartilhar informações através de texto em uma *webpage* vem sendo substituída pela reprodução de vídeos localizados em bancos de dados de fácil acesso, como, por exemplo, o *YouTube*. Além disso, videoconferências vêm sendo amplamente utilizadas com a disponibilidade de maiores larguras de banda para a transmissão de dados. De acordo com CISCO (2015), o volume do tráfego de dados relacionado à transmissão de vídeos na internet irá aumentar de 64% em 2014 para 80% em 2019.

O processo de codificação de vídeo visa à diminuição das redundâncias presentes na representação computacional de imagens ou vídeos (AGOSTINI, 2007). Afirma-se que um dado é redundante quando o mesmo não contribui com novas informações para a representação da imagem. Existem três tipos de redundância: redundância espacial (redundância intra-quadro, ou seja, semelhanças dentro do próprio quadro), redundância temporal (redundância inter-quadro, ou seja,

semelhanças entre quadros) e redundância entrópica (redução estatística na representação dos símbolos). O correto processo de codificação/decodificação torna possível a transmissão e o armazenamento de vídeos digitais.

Padrões de codificação de vídeo estão presentes no nosso dia-a-dia e a não utilização dos mesmos para a manipulação de vídeos digitais é impraticável, visto que a utilização de um vídeo digital sem qualquer tipo de compressão (codificação) requer a manipulação uma enorme quantidade de dados (GONZALEZ, 2003).

Diversos padrões de codificação de vídeo foram então desenvolvidos ao longo dos anos para sanar as necessidades correlatas a esta área. MPEG-2 (HASKELL, 2002), MPEG-4 e H.264/AVC (SULLIVAN, 2006) são exemplos de padrões já desenvolvidos e que ainda continuam sendo utilizados. Para atender às demandas mais recentes em relação a vídeos digitais, o *Joint Collaborative Team on Video Coding* (JCT-VC) desenvolveu o *High Efficiency Video Coding* (HEVC), um dos padrões mais atuais em codificação de vídeo (ITU-T, 2015).

Atualmente, muitos sistemas embarcados possuem suporte à manipulação de vídeo, onde a utilização de codificadores e decodificadores de vídeo é mandatória. Neste universo, questões relacionadas à eficiência de codificação (relação entre qualidade e taxa de bits), complexidade e consumo de energia são extremamente importantes. Tais características são alvo de estudos e pesquisa pela indústria e pela academia dada a importância que possuem.

1.1 Motivação

A maior parte dos sistemas que utilizam vídeos digitais é embarcada, ou seja, possui um sistema microprocessado no qual o computador é completamente dedicado a este sistema e, na maioria das vezes, possuem como característica a mobilidade, funcionando basicamente com pilhas e baterias como fonte de energia. Segundo o Youtube (2016), mais da metade das visualizações de vídeos ocorrem em dispositivos móveis. São exemplos de sistemas embarcados com estas características os computadores de bordo, câmeras digitais, drones, etc. Dessa forma, é necessário que exista o equilíbrio entre consumo de memória, eficiência de codificação, desempenho e consumo de energia. De nada adiantaria a obtenção de um alto desempenho de codificação em detrimento do uso exagerado da energia disponível.

A maioria dos trabalhos de pesquisa envolvendo a reprodução de vídeos digitais preocupa-se, principalmente, com redução de esforço computacional (complexidade). Tendo em vista que muitas vezes a relação entre complexidade e energia não é linear, principalmente em sistemas heterogêneos, analisar somente o esforço computacional não basta, é necessária também a análise do consumo energético. No entanto, a questão do consumo de energia em muitos trabalhos simplesmente não é mencionada. Os poucos trabalhos que consideram energia no HEVC, como Monteiro (2015) e Monteiro (2014), não propõem controle da energia consumida. Ainda, é possível controlar energia através da variação dos QPs (*Quantization Parameters*), parâmetro o qual determina como se dá a remoção de informações visuais menos perceptíveis ao olho humano; como é feito para o controle de taxa de bits, pois há relação entre o volume de dados, o qual é parametrizado através do QP, e o consumo de energia do codificador. No entanto, conforme será demonstrado, essas soluções são ineficientes, dessa forma, a principal motivação para o desenvolvimento deste trabalho consiste na necessidade do estudo e execução de técnicas de controle aplicáveis à redução da energia consumida pelo codificador de vídeo sem ônus em relação à eficiência de codificação (ou com o menor ônus possível). Este tema é bastante atual e caracteriza-se como importante tema de pesquisa.

1.2 Objetivos

O codificador/decodificador (codec) de vídeo é um dos principais responsáveis pela demanda de desempenho e consumo de energia em sistemas embarcados atuais (CORRÊA, 2016). O processo de codificação/decodificação envolve diversas etapas as quais são custosas computacionalmente e energeticamente. Dessa forma, o consumo de energia do codec assume especial importância em sistemas embarcados que utilizam bateria, como em dispositivos móveis e portáteis.

O principal objetivo deste trabalho é o desenvolvimento de um controlador que, em tempo de execução, atue no processo de codificação de modo a efetuar o controle de energia consumida pelo sistema, tomando decisões que impliquem em menores perdas em eficiência de codificação. Dentre os objetivos específicos ainda pode-se destacar a realização de uma análise de sensibilidade em relação ao

consumo de energia do codificador HEVC, e a obtenção de pontos ótimos de operação a partir da aplicação de técnicas de otimização.

1.3 Organização do Trabalho

Este trabalho está dividido em sete capítulos. No segundo capítulo é realizada uma revisão sobre codificação de vídeo e sobre o padrão HEVC, um estudo a respeito do consumo de energia em sistemas microprocessados e sobre ferramentas de medição de energia e, por fim, apresenta também uma análise sobre trabalhos relacionados.

O terceiro capítulo apresenta a metodologia utilizada neste trabalho para efetuar a medição de energia do codificador HEVC, abordando um estudo sobre a utilização do RAPL bem como a integração desta ferramenta com o *software* de referência do HEVC.

O quarto capítulo apresenta a análise de sensibilidade do consumo de energia do codificador HEVC em relação à eficiência de codificação quando se variam parâmetros de codificação.

O quinto capítulo apresenta o controlador de energia proposto, o qual atua em tempo de codificação de forma dinâmica aplicado ao *software* de referência do HEVC. O sexto capítulo apresenta os resultados obtidos e uma breve discussão. E, por fim, o sétimo capítulo apresenta as conclusões deste trabalho e sugestões de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

Este capítulo apresenta uma fundamentação teórica sobre o tema abordado nesta dissertação, abrangendo um breve estudo sobre o codificador HEVC, suas estruturas de particionamento e o fluxo de codificação, uma análise a respeito das métricas de comparação utilizadas, um estudo sobre o consumo de energia em um sistema microprocessado e algumas das ferramentas de medição de energia existentes, bem como uma breve explanação a respeito dos trabalhos relacionados.

2.1 A Codificação de Vídeo e o Codificador HEVC

De acordo com Richardson (2003), um vídeo digital consiste em uma sequência de imagens independentes, captadas a uma determinada taxa de amostragem. Estas imagens, chamadas de quadros ou *frames*, formam uma sequência de vídeo. Os quadros são compostos por pixels, os quais são os menores pontos de informação visual que formam a imagem. Os pixels são representados através de três amostras, correspondentes a informações de brilho ou cor, conforme o sistema de representação de cores utilizado, no caso do HEVC é aplicado o sistema YCbCr (Luminância, Crominância Azul e Crominância Vermelha) (RICHARDSON, 2002). Cada quadro pode ser subdividido em blocos pelo codificador e, dependendo do padrão de codificação utilizado, os blocos podem possuir diferentes tamanhos. Na Figura 1 pode ser observada uma sequência de quadros temporalmente vizinhos e um quadro dividido em blocos.

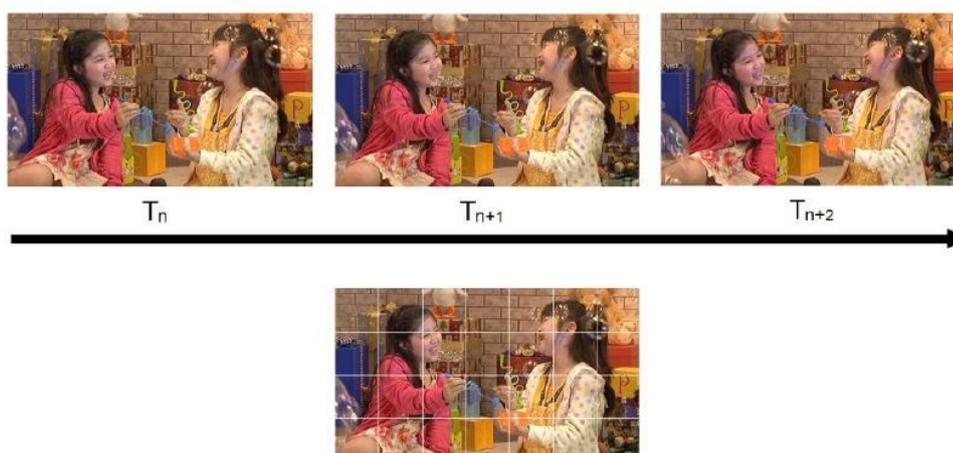


Figura 1 – Sequência de quadros temporalmente vizinhos
(Fonte: AFONSO, 2013)

A codificação de vídeos busca diminuir a quantidade de dados considerados redundantes na representação das informações da imagem ou vídeo. Considera-se redundante aquele dado que não contribui com novas informações visuais relevantes para a representação da imagem. Basicamente, existem três tipos de redundância: redundância espacial, redundância temporal e redundância entrópica (GHANBARI, 2003).

Em relação à amostragem espacial, existe uma matriz de pontos chamada de resolução do vídeo, quanto maior a resolução melhor será a qualidade da imagem (assumindo a mesma qualidade de captação), uma vez que maior será o detalhamento da mesma em virtude da maior utilização de pixels para representá-la. A amostragem temporal está diretamente relacionada ao tempo transcorrido entre a captação da imagem atual e a imagem seguinte. Para que a sensação de movimento real seja obtida, altas taxas de amostragem são desejáveis. Aplicações que necessitem de sensação de tempo real demandam no mínimo de 24 a 30 imagens a cada segundo (GHANBARI, 2003).

2.1.1 Fluxo de Codificação no HEVC

O principal objetivo durante a criação do HEVC era dobrar a taxa de compressão alcançada pelo padrão H.264/AVC, mantendo a mesma qualidade subjetiva do vídeo codificado (GROIS et. al., 2013). Este objetivo, em média, foi alcançado, e o HEVC é o estado da arte em codificação de vídeo (SULLIVAN et. al., 2012). O *software* de referência utilizado pelo HEVC é o HM – HEVC *Test Model* (BOYCE, 2014), o qual é periodicamente atualizado.

Na Figura 2 está representado o diagrama de blocos de um codificador de vídeo, mais precisamente este diagrama representa as etapas da codificação do padrão HEVC, no qual não são representados os fluxos dos sinais de controle.

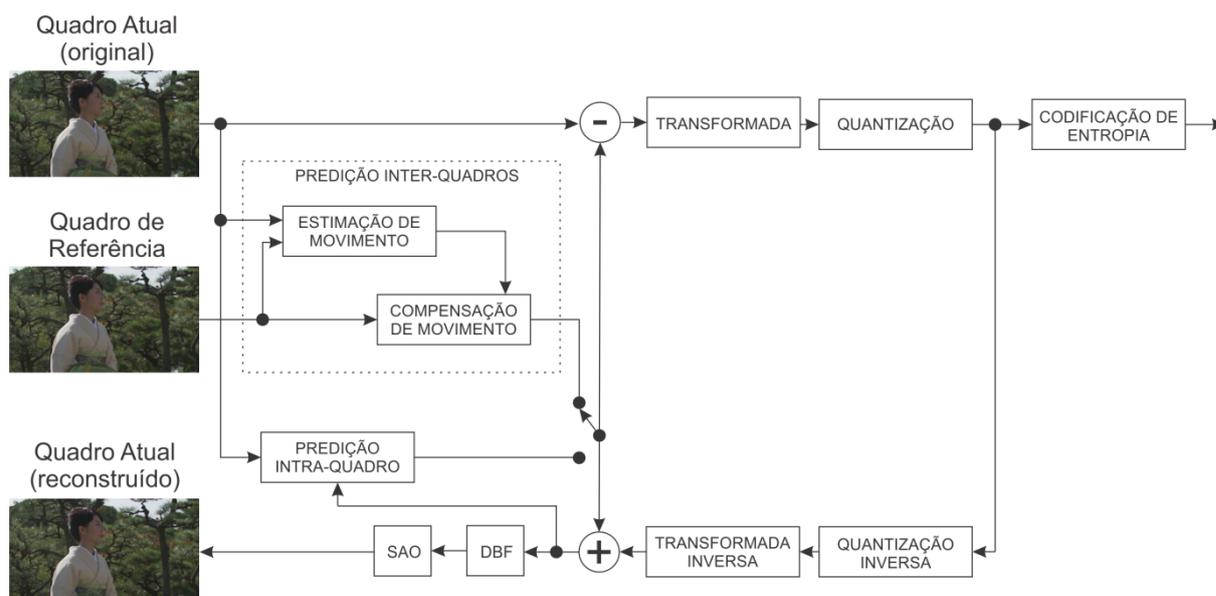


Figura 2 – Diagrama de blocos do codificador HEVC
(Fonte: Adaptado de AFONSO, 2013)

Codificadores modernos aplicam codificação residual, ou seja, trabalham com a diferença de valores entre um quadro atual e um anteriormente processado, a qual é chamada de resíduo (RICHARDSON, 2003). Para codificar um vídeo, o HEVC realiza codificação residual e utiliza dois tipos de previsão: a previsão intra-quadro e a previsão inter-quadros.

Para codificar o primeiro quadro da sequência, no início do processamento, é utilizada a previsão intra-quadro, visto que nenhum *frame* foi codificado ainda, e é necessário gerar o primeiro quadro de referência. Após o segundo quadro do vídeo, o codificador utiliza, simultaneamente, dois quadros do vídeo, um que está sendo comprimido (atual) e outro que foi anteriormente processado (referência). O codificador faz a seleção do tipo de previsão que será utilizada, previsão intra-quadro ou previsão inter-quadros. Essa tomada de decisão é representada pela chave seletora presente no diagrama de blocos do codificador.

Após a etapa de previsão os resíduos são então enviados para as etapas de transformada e quantização. Primeiramente ocorre a operação da transformada, responsável por transformar informações do domínio espacial para o domínio das frequências, tornando mais evidente as informações desnecessárias à percepção visual humana, as quais serão posteriormente descartadas. Após esta etapa é aplicada a quantização, representando os resultados das transformadas como um conjunto reduzido de níveis. Dessa forma, algumas informações, desnecessárias do

ponto de vista da percepção visual humana, são atenuadas ou até mesmo eliminadas (GONZALEZ, 2003). A Figura 3 apresenta um exemplo da aplicação da transformada e da quantização em um bloco B arbitrário. O resultado após a quantização é uma matriz esparsa, que potencializa a etapa de codificação de entropia, etapa final do codificador de vídeo, a qual aplica um algoritmo de compressão chamado *Context-Adaptive Binary Arithmetic Coding* (CABAC) (SZE et al., 2014), com o objetivo de reduzir a redundância entrópica, relacionada à forma como os dados são codificados. Finalmente tem-se o *bitstream* final, o qual é a informação que será enviada ou armazenada.

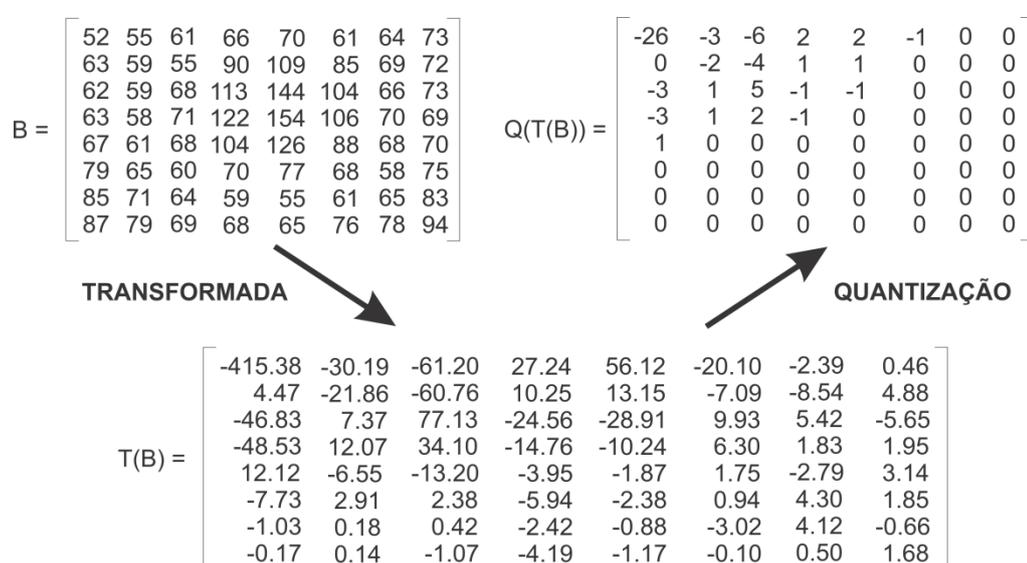


Figura 3 – Aplicação da transformada e da quantização em um bloco B arbitrário
(Fonte: Adaptado de GRELLERT, 2014)

Como a etapa de quantização gera perdas irreversíveis de informação, o quadro codificado será diferente do quadro original após a reconstrução (RICHARDSON, 2003). Logo, a referência utilizada pelo codificador e pelo decodificador deve ser a mesma. O codificador descarta o quadro original após ser processado e armazena o quadro reconstruído. O quadro reconstruído é obtido através da presença de etapas referentes à decodificação, como a quantização, a transformada inversa, o DBF (*Deblocking Filter*), o SAO (*Sample Adaptive Offset*) e a compensação de movimento. Dessa forma o quadro codificado se torna a referência para a codificação e decodificação.

2.1.2 Estruturas de Particionamento no HEVC

De acordo com ITU-T (2015), em contraste com esquemas apresentados por padrões de codificação de vídeo anteriores, o HEVC emprega um esquema de compressão de vídeo baseado no particionamento dos blocos codificados em uma hierarquia altamente flexível, a qual permite o uso de blocos grandes e múltiplos níveis para blocos de predição e transformadas, além de novas ferramentas de codificação. Todos estes aperfeiçoamentos fazem do HEVC um codificador de vídeo muito eficiente.

O esquema de compressão de vídeo se baseia numa hierarquia flexível, a qual inclui três conceitos de bloco: Unidade de Codificação (CU – *Coding Unit*), Unidade de Predição (PU – *Prediction Unit*) e Unidade de Transformada (TU – *Transform Unit*) (ROSEWARNE, et.al., 2015). Esta separação em diferentes conceitos permite a cada um ser otimizado da melhor forma possível.

A imagem é particionada em CTUs (*Coding Tree Unit*). Cada CTU é composta por um bloco de amostras de luminância juntamente com outros dois blocos de crominância. Os tamanhos dos blocos de crominância irão depender da amostragem de cores que foi adotada; geralmente as amostras de crominância são subamostradas (RICHARDSON, 2003). O tamanho máximo de uma CTU é 64x64 amostras de luminância, que também corresponde ao tamanho máximo de uma CU.

A CTU pode ser composta por uma ou mais CUs. Estas são utilizadas em ambos os tipos de predição e são sempre quadradas, com tamanho $2N \times 2N$, onde N pode ser 4, 8, 16 ou 32. Logo, a faixa de valores de uma CU é de no mínimo 8x8 amostras de luminância até o tamanho da CTU. A divisão recursiva da CTU em CUs permite desde tamanhos pequenos até tamanhos grandes, formando uma árvore quadrática (*quadtree*) formada por blocos de CU (SULLIVAN et. al., 2012). A Figura 4 apresenta a ilustração da divisão de uma CTU em CUs. Pode-se perceber que na profundidade 1 tem-se um quadro 128x128, particionado em quatro blocos 64x64 (I, II, III e IV). Este é o primeiro nível da CTU, no entanto, cada bloco 64x64 pode ainda ser subdividido em quatro blocos 32x32, perfazendo o nível de profundidade 2. Dessa forma, recursivamente, cada bloco pode continuar sendo subdividido até a profundidade 4, na qual os blocos terão tamanho 8x8.

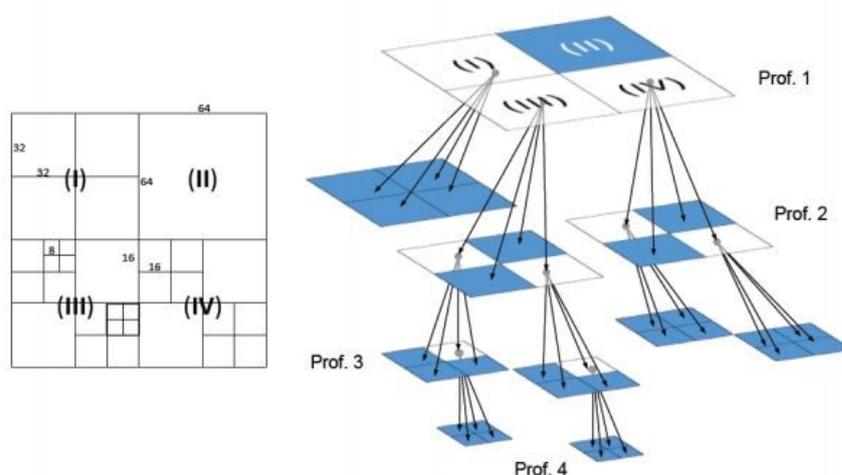


Figura 4 – Divisão de uma CTU em CUs
(Fonte: SANCHEZ, 2013)

CUs podem ser particionadas em PUs, que possuem informações relacionadas ao processo, como modo de predição (intra ou inter), vetores de movimento, índice dos quadros de referência, etc (SZE et. al, 2014). São definidas quatro partições simétricas e quatro partições assimétricas, como pode ser ilustrado na Figura 5. O menor tamanho de uma PU é 4x4.

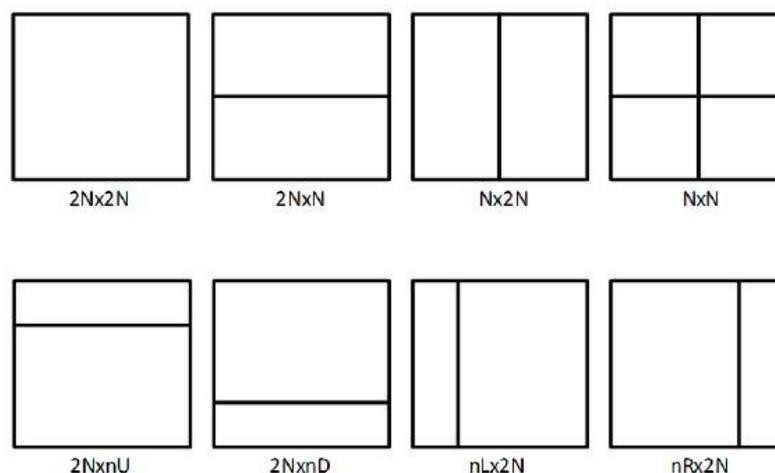


Figura 5 – Geometria de PUs possíveis
(Fonte: AFONSO, 2013)

A TU é a unidade básica para os processos de transformada e quantização. A TU apresenta formato sempre quadrado, de tamanhos 4x4 até 32x32 amostras. Cada CU pode conter uma ou mais TUs, onde várias TUs podem estar dispostas em

uma estrutura de árvore quadrática (SULLIVAN et. al., 2012), como pode ser observado na Figura 6.

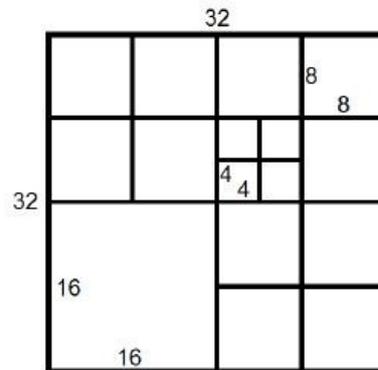


Figura 6 – Divisão de uma CU em TUs
(Fonte: AFONSO, 2013)

2.1.3 Otimização da Taxa-Distorção (*Rate-Distortion Optimization - RDO*)

Baseado no que foi explicado até o momento e, conforme Grellert (2014), pode-se perceber que existem inúmeras combinações possíveis para se codificar uma simples CTU: diferentes modos de predição, diferentes modos de particionamento, tamanho das transformadas, parâmetros de quantização, etc. Para decidir qual a melhor escolha o codificador utiliza a chamada Otimização da Taxa-Distorção (aqui neste trabalho será utilizada a nomenclatura em inglês, amplamente encontrada na literatura RDO – *Rate Distortion Optimazation*). De maneira simplificada, pode-se interpretar que, dado um valor de *bit-rate* limite, o codificador deve selecionar o modo de codificação que retorne a mínima distorção possível da imagem. Matematicamente este problema pode ser exposto como segue (CORRÊA, 2014):

Se S representa todos os modos de codificação possíveis e i é um elemento de S , ou seja, $i \in S$. Então,

$$i^* = \arg \min_{i \in S} D(i) \quad (1)$$

sujeito à $R(i) \leq R_T$

onde i^* é o modo ótimo que minimiza a distorção, $D(i)$ é a distorção obtida com o modo i , $R(i)$ é a taxa de bits do modo i e R_T é a taxa de bits limite.

Este problema de otimização pode ser resolvido através do método dos multiplicadores de Lagrange, onde a distorção é normalizada em conjunto com a taxa de bits. A minimização de Lagrange é representada em (8), onde $J(i)$ é o custo R-D, $D(i)$ e $R(i)$ são respectivamente a distorção e a taxa de bits do modo i , e λ é o multiplicador de Lagrange (WIEGAND, 1998). O modo de codificação i^* que retorna o custo mínimo $J(i)$ é selecionado como a solução do problema.

$$i^* = \arg \min_{i \in S} J(i) \quad (2)$$

onde $J(i) = D(i) + \lambda \cdot R(i)$

O multiplicador de Lagrange λ é determinado no codificador HEVC através da equação apresentada em (3) (ROSEWARNE, 2015).

$$\lambda = \alpha \cdot W_k \cdot 2^{((QP-12)/3)} \quad (3)$$

Onde W_k representa um fator de normalização dependente da configuração e do nível de hierarquia do *offset* dos QPs do quadro atual em um GOP, e α representa um fator de multiplicação que associa os tipos de quadro (referenciados ou não referenciados) (ROSEWARNE, 2015). Note que existe uma forte correlação entre o QP utilizado na codificação e o respectivo custo R-D.

De acordo com Corrêa (2014), como não existem modelos simples para descrever a relação entre os modos de codificação e o custo R-D (J), o processo RDO implementado pelos codificadores de vídeo atuais testa alguns dos modos de codificação possíveis (determinados a partir de heurísticas) e seleciona o que apresentar menor custo J . Na Figura 7 pode ser observado um exemplo de curva R-D. Nesta figura, cada ponto corresponde a um diferente modo de codificação testado. Os pontos que apresentam a menor distorção D para uma taxa de bits R especificada compõem a curva R-D ótima, representada na figura pela linha tracejada.

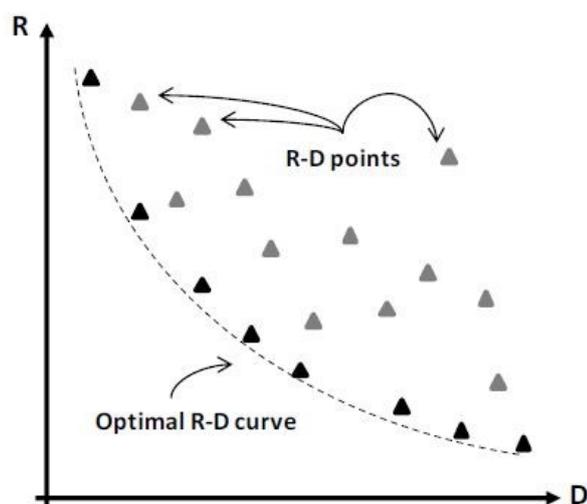


Figura 7 – Exemplo de curva R-D
(Fonte: CORRÊA, 2014)

2.1.4 Métricas de Comparação

No processo de codificação de um vídeo digital é muito importante a utilização de critérios que possibilitem a comparação entre o vídeo original e o vídeo comprimido em termos da eficiência da compressão, isto é, a relação entre a redução da quantidade de dados a serem transmitidos ou armazenados e os efeitos provocados na qualidade da imagem (RICHARDSON, 2003).

Um dos critérios mais simples de comparação é a taxa de compressão, que é definida como sendo a redução no volume de dados obtidos por um determinado algoritmo de compressão (SANBORN, 2005). A medição do *bit-rate* (taxa de bits) sem a utilização de compressão, e outra medição, com a utilização de alguma técnica de compressão, permite visualizar esta redução (GHANBARI, 2003).

De acordo com Richardson (2003), a avaliação da qualidade de vídeos digitais pode ser feita utilizando métricas objetivas e subjetivas. As métricas objetivas permitem verificar a qualidade do vídeo analiticamente, comparando o vídeo original com o vídeo codificado, já métricas subjetivas estão relacionadas à percepção visual humana e são difíceis de mensurar.

De acordo com Ghanbari (2003), um dos critérios objetivos mais utilizados é o PSNR (*Peak Signal-to-Noise Ratio*). O PSNR pode ser aplicado a um bloco, um quadro ou até mesmo ao vídeo todo, e o valor é fornecido em decibéis (dB). O

cálculo do PSNR utiliza o critério de similaridade do erro médio quadrático (MSE – *Mean Squared Error*), como pode ser observado nas equações (4) e (5).

$$PSNR = 20 \cdot \log_{10} \frac{MAX}{\sqrt{MSE}} \quad (4)$$

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (R_{i,j} - O_{i,j})^2 \quad (5)$$

Em (4) e (5) MSE representa o erro médio quadrático das amostras de um quadro, ou seja, a diferença média quadrática entre os valores de cada amostra que compõem o quadro, R representa o quadro reconstruído, O representa o quadro original, m e n são as dimensões do quadro e MAX é o valor máximo das amostras. Geralmente, quanto maior for o valor resultante do PSNR maior será a qualidade da imagem. Entretanto, o PSNR possui uma série de limitações, sendo que, em alguns casos, uma imagem que possui uma qualidade visual (subjéctiva) superior pode ter um valor de PSNR inferior à outra com resultado visual inferior (RICHARDSON, 2003). Isto ocorre porque o PSNR atribui a mesma importância para todas as amostras, enquanto que, do ponto de vista subjéctivo determinadas partes da imagem podem ser mais importantes do que outras, quando observadas por um ser humano (RICHARDSON, 2003).

Outro critério de similaridade bastante utilizado é a SAD (*Sum of Absolute Differences*) (KHUN, 1999). O critério da soma das diferenças absolutas, em relação ao MSE, apresenta a vantagem de possuir implementação em hardware mais simples, pois utiliza apenas operações de soma e subtração, como pode ser observado na equação (6). Novamente m e n são as dimensões, R e O os quadros reconstruído e original, respectivamente.

$$SAD = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |R_{i,j} - O_{i,j}| \quad (6)$$

O *software* de referência do HEVC emprega também o uso de uma versão alterada da SAD, a chamada SATD (*Sum of Absolute Transformed Differences*), que envolve o uso da transformada *Hadamard* e agrega maior esforço computacional ao processo. Apesar de ser uma métrica que atinge melhores resultados quando

comparada com a SAD, a SATD emprega quatro vezes mais operações aritméticas que a SAD (GRELLERT, 2014). Em (7) pode ser observada a equação da SATD.

$$SATD = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |HT(i,j)| \quad (7)$$

$HT(i,j)$ em (7) é o coeficiente (i,j) de um bloco que foi obtido através da aplicação da Transformada *Hadamard* (ZHU, 2013).

Além das métricas *bit-rate* e PSNR, são utilizadas também as métricas *BD-bit-rate* (BD-BR), ou simplesmente *BD-Rate*, e *BD Peak Signal-to-Noise Ratio* (BD-PSNR), as quais são baseadas na *Bjontegaard Difference* (BD) (BJØNTEGAARD, 2008). Como existe um compromisso entre *bit-rate* e qualidade, não é conclusivo comparar apenas um destes critérios de modo isolado, logo, uma métrica que relacione estes dois valores é interessante. O *BD-Rate* pode ser interpretado como a variação percentual do *bit-rate* entre uma configuração de referência (A) e uma configuração teste (B), considerando vídeos com a mesma qualidade objetiva após a codificação. Assim, um *BD-Rate* positivo significa perda de desempenho de codificação de B em relação a A, um *BD-Rate* negativo indica ganho, ou seja, uma codificação que resultou em um vídeo de menor *bit-rate*, mas com a mesma qualidade objetiva. Já o *BD-PSNR* pode ser interpretado como a variação em decibéis do PSNR entre a referência e a condição de teste, considerando vídeos que apresentam a mesma taxa de bits. Dessa forma, um valor de *BD-PSNR* negativo indica perda de qualidade enquanto que um valor positivo representa melhora na qualidade do vídeo, sendo que ambos possuem o mesmo *bit-rate*. Para obter estas métricas as configurações A e B são codificados para quatro diferentes *Quantization Parameters* (QP), produzindo oito pares (PSNR, *bit-rate*). Estes pares são utilizados para interpolar duas curvas através de uma função de interpolação de terceira ordem, gerando duas curvas *Rate-Distortion* (RD) (BJØNTEGAARD, 2008). A área entre as curvas é integrada utilizando o eixo X como referência para o *BD-Rate* e o eixo Y como referência para o *BD-PSNR*. A equação do cálculo do *BD-Rate* e do *BD-PSNR* são dadas a seguir, cujos dados estão em concordância com o exemplo apresentado na Figura 8.

$$BD_{Rate} = \frac{\int_{38}^{40} (REF_{PSNR}(x) - TESTE_{PSNR}(x)) dx}{40 - 38} \quad (8)$$

$$BD_{PSNR} = \frac{\int_{3000}^{9000} (REF_{bitrate}(y) - TESTE_{bitrate}(y)) dy}{9000 - 3000} \quad (9)$$

Em (8) e (9), $REF(x$ ou $y)$ e $TESTE(x$ ou $y)$ representam, respectivamente, os valores de PSNR e $bit-rate$ para a curva de referência (A) e para a curva de teste (B). Como pode ser observado na Figura 8, são traçadas duas curvas relacionando PSNR e $bit-rate$, chamadas curvas R-D (*Rate – Distortion*). As curvas são traçadas a partir de um método de interpolação de terceira ordem e a área entre ambas, dividida pelo intervalo de integração, representa o BD-BR, na Figura 8-(a) e o BD-PSNR, na Figura 8-(b). A única diferença é o eixo de integração, para se calcular BD-BR se faz a integração em relação ao eixo x, para se calcular BD-PSNR se faz a integração em relação ao eixo y (BJØNTEGAARD, 2008).

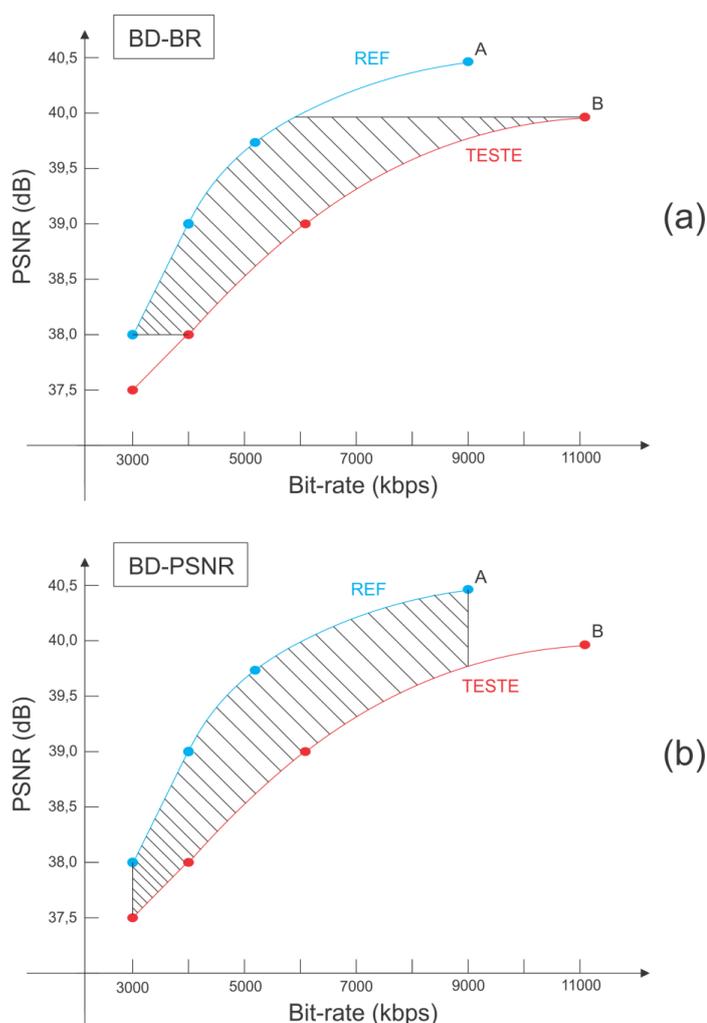


Figura 8 – Curvas de (a) BD-BR e (b) BD-PSNR para uma dada aplicação
(Fonte: Adaptado de GRELLERT, 2014)

2.1.5 Condições Comuns de Teste

Uma vez que codificadores de vídeo possuem inúmeras aplicações, desde *broadcasting* até videoconferências em *smartphones*, padrões de codificação de vídeo devem proporcionar uma larga gama de configurações de modo para suportar essas inúmeras aplicações. Justamente este vasto número de aplicações se torna um problema quando é necessário efetuar comparações entre diferentes implementações, uma vez que para uma comparação justa as configurações de ambas devem ser as mesmas (GRELLERT, 2014).

Devido a esta situação, o JCT-VC estipulou um documento que estabelece Condições Comuns de Teste (*Common Test Conditions – CTC*) para o HM (BOSSSEN, 2013). O documento das CTCs define seis classes de sequências de vídeos (de A até F) que devem ser utilizadas quando são executados testes com o HEVC e, também, define que as simulações devem utilizar quatro níveis de QP diferentes: 22, 27, 32 e 37, ou seja, devem ser realizadas simulações que envolvam menor degradação à imagem (QPs mais baixos) e simulações que causem maior degradação (QPs mais altos), uma vez que o valor do QP está diretamente relacionado à quantidade de informação que será descartada durante o processo de codificação.

Além disso, este documento estabelece as estruturas de predição temporal, as quais são divididas em três tipos: *intra-only*, *low-delay* e *random-access*, as quais estabelecem a maneira pela qual os quadros se distribuem e se referenciam em um GOP (*Group of Pictures*). Neste trabalho é utilizada a estrutura *low-delay*.

Existem dois tipos de configuração *low-delay*, tipo P (predição unidirecional) ou tipo B (predição unidirecional ou bidirecional). Em ambas as condições apenas o primeiro quadro da sequência é codificado como quadro *intra*. Na configuração *low-delay* tipo B todos os quadros sucessivos são codificados como quadros tipo GPB (*Generalized P and B Picture*). O quadro GPB é capaz de usar somente quadros de referência temporalmente anteriores com relação ao quadro atual, quadros B podem usar quadros de referência de duas listas distintas, compostas pelos quadros de referência anteriormente armazenados, e, quadros P, usam somente quadros de referência de uma lista. Na configuração *low-delay* tipo P todos os quadros do tipo *inter* são codificados como tipo P (ROSEWARNE, 2015). A Figura 9 apresenta a representação gráfica de uma configuração *low-delay*. Os índices acima de cada

quadro indicam a ordem de codificação. Na configuração *low-delay* a ordem de codificação se dá na mesma ordem de exibição dos quadros.

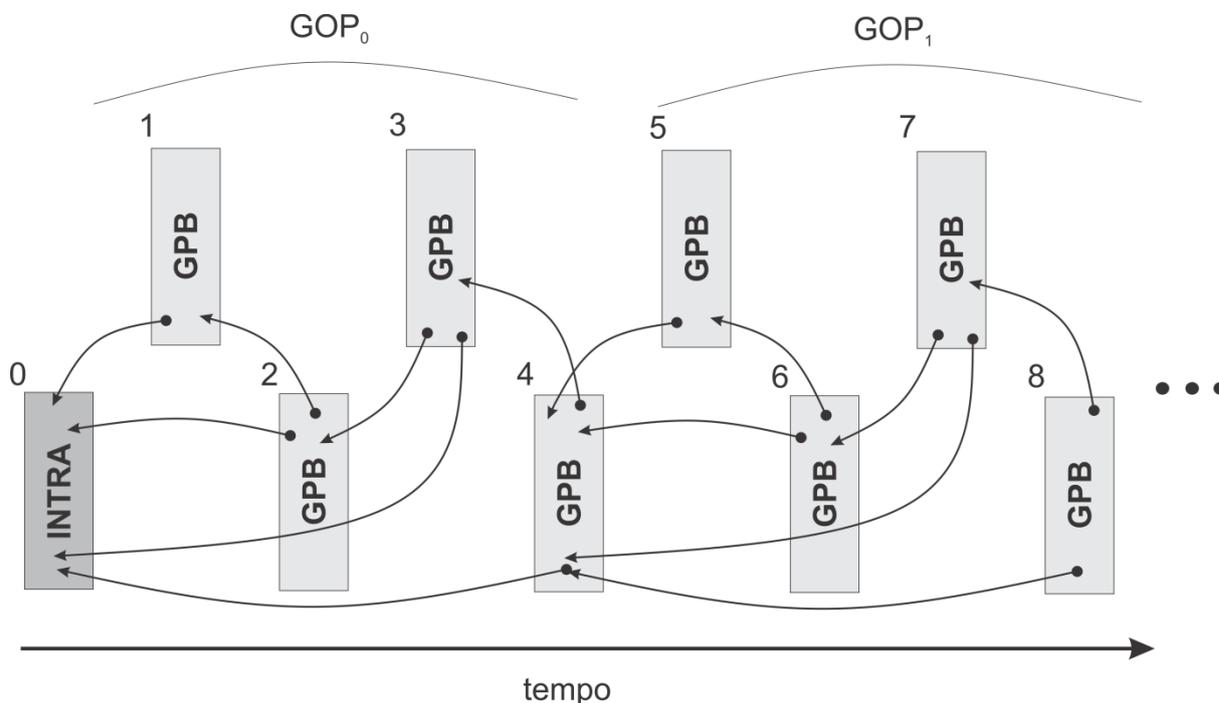


Figura 9 – Representação gráfica da configuração *low-delay*
(Fonte: Adaptado de ROSEWARNE, 2015)

2.2 Frentes de Pareto

No cálculo do custo R-D são levadas em consideração duas variáveis independentes, taxa de bits e distorção, o que representa um problema bidimensional. Se o consumo de energia for considerado na seleção de parâmetros de codificação, uma nova dimensão deve ser adicionada ao problema RDO, resultando numa otimização R-D-E (RDEO – *Rate Distortion Energy Optimization*). Entretanto, a RDEO é uma tarefa muito difícil de ser executada em tempo real de codificação, principalmente porque, diferentemente das variáveis taxa de bits e da distorção, que podem ser reduzidas com a alteração dos parâmetros de codificação, a energia consumida durante o processo de codificação não pode ser recuperada pelo codificador. A partir dos resultados demonstrados no Capítulo 4, pode-se determinar, de maneira *off-line*, os pontos ótimos de operação do codificador, a partir da utilização de frentes de Pareto, uma vez que este problema necessita de uma otimização multi-objetivo.

De acordo com Ghosh (2004), uma frente de Pareto é composta por pontos que sejam eficientes à Pareto, isto é, pontos ótimos de operação em um problema multi-objetivo. Tal método de otimização multi-objetivo foi proposto em 1896 pelo economista Vilfredo Pareto e é conhecido como Eficiência à Pareto. O conceito consiste no fato de que a solução de um problema de otimização multi-objetivo não é normalmente um único valor, mas um conjunto de valores denominado frente de Pareto. O “ótimo de Pareto” ocorrerá quando existe uma situação (A) onde, ao se sair dela, para que “um ganhe”, pelo menos “um perde”, necessariamente. Desta forma, uma dada situação é ótima no sentido de Pareto se não for possível melhorar a situação de um agente, sem degradar a situação de qualquer outro agente.

Na Figura 10 é ilustrado o exemplo de uma solução ótima de Pareto, onde são analisados o tempo utilizado e o espaço ocupado por determinado algoritmo. Neste problema, ambas as variáveis, tempo e espaço, devem ser minimizadas.

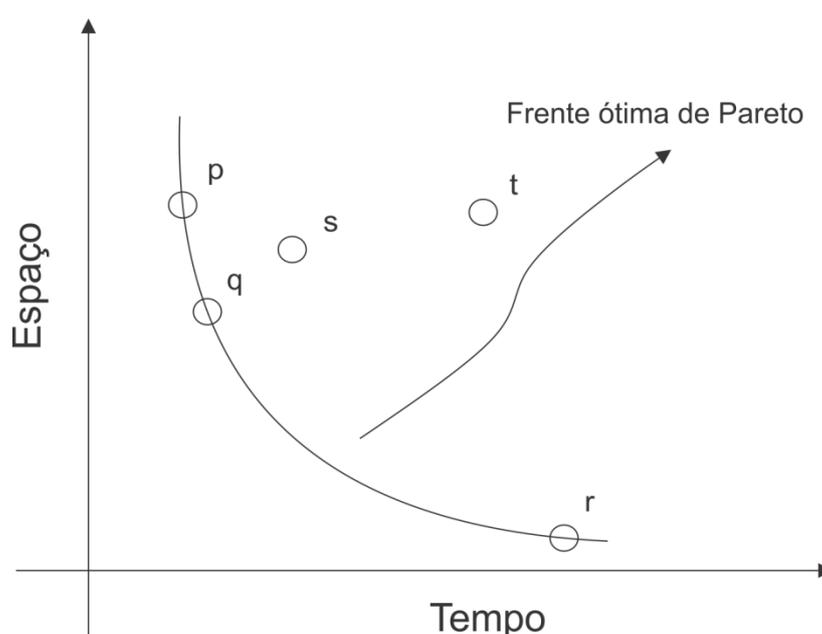


Figura 10 – Exemplo de uma solução ótima de Pareto
(Fonte: Adaptado de GHOSH, 2004)

O ponto 'p' representa a solução que apresenta menor tempo possível, mas o espaço ocupado é grande. No entanto, o ponto 'r' representa uma solução com menor espaço possível, porém, é a que mais tempo leva para ser executada. Considerando-se ambos objetivos, nenhuma das soluções é ótima, neste caso, não se pode afirmar que 'p' é melhor que 'r' ou vice-versa. Além disso, ainda existem

soluções como a solução ‘q’ por exemplo, que representa uma situação intermediária. Dessa forma, afirma-se que existem soluções que fazem parte da frente de Pareto, na qual um ponto ou outro será melhor ou pior, dependendo da métrica adotada. Também fica claro que existem soluções que não pertencem à frente de Pareto, como os pontos ‘s’ e ‘t’ por exemplo. A razão pela qual estes pontos não fazem parte da frente de Pareto é porque, se compararmos ‘t’ com ‘q’ por exemplo, pode-se observar que ‘t’ não é melhor que ‘q’ sob nenhum dos aspectos. Neste caso pode-se dizer que o ponto ‘t’ é dominado pelo ponto ‘q’.

2.3 Consumo de Energia em um Sistema Microprocessado

Nesta seção será realizada uma breve revisão sobre energia e potência em circuitos digitais bem como serão estudadas as ferramentas que possibilitam a medição da energia consumida pelo codificador num sistema microprocessado.

2.3.1 Energia e Potência em Circuitos Digitais

Energia é uma grandeza escalar que diz respeito à quantidade de trabalho que um sistema é capaz de realizar. Já potência é uma grandeza, também escalar, que diz a quantidade de energia que um sistema gasta em determinado intervalo de tempo. Em (10) pode ser observada a equação da potência média P de uma dada aplicação, a qual é a razão entre a variação de energia $\Delta(E)$ e a variação de tempo $\Delta(t)$.

$$P = \frac{\Delta(E)}{\Delta(t)} \quad (10)$$

A unidade de potência no Sistema Internacional de Unidades (SI) é o Watt (W), da energia é o Joule (J) e do tempo é o segundo (s).

A Figura 11 demonstra a diferença entre estas duas grandezas, enquanto a potência é representada pela curva do gráfico (eixo das ordenadas), a energia é representada pela área abaixo desta curva. A energia pode ser então obtida através da integral da potência, presente na equação (11).

$$Energia = \int P . dt \quad (11)$$

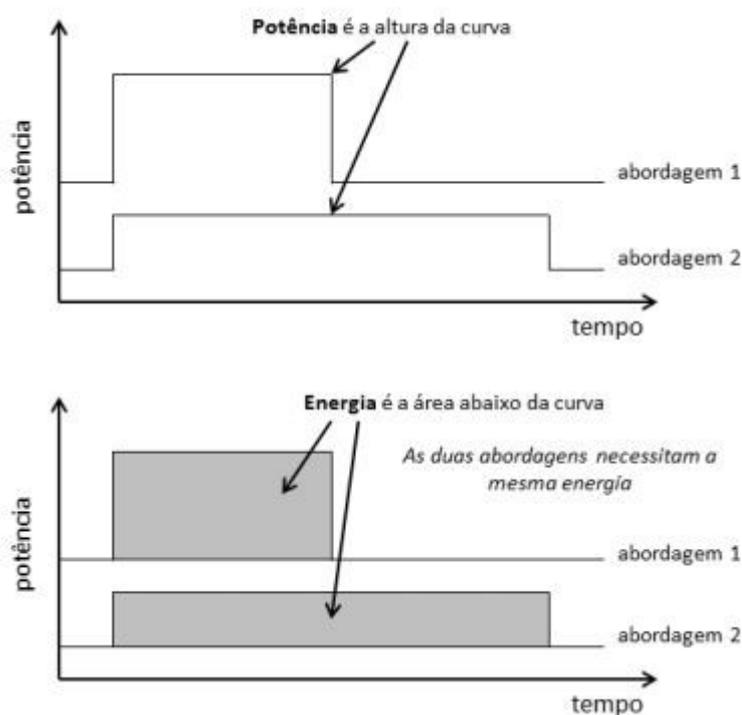


Figura 11 – Potência VS Energia
(Fonte: FISCHER, 2012)

2.3.2 Ferramentas para Análise do Consumo de Energia

Segundo Fischer (2012), existe três características que são levadas em consideração no projeto de ferramentas de avaliação de consumo energético: precisão, granularidade e *overhead*. Ferramentas de baixa precisão não fornecem resultados satisfatórios, sendo a precisão uma expressão do grau de concordância entre repetidas medições. Granularidade diz respeito à quantidade de detalhes e nível de abstração da análise. Ferramentas flexíveis permitem que o usuário ajuste o nível de granularidade de acordo com suas necessidades. A aliança entre precisão e granularidade norteia o desenvolvimento de ferramentas de análise de consumo de energia.

Outra questão importante a ser analisada é o *overhead* inserido pela ferramenta, isto é, a quantidade de recursos do sistema consumidos para a ferramenta de consumo energético funcionar. Dadas essas três características, é necessário encontrar uma solução que apresente equilíbrio entre precisão, granularidade e *overhead*. Ferramentas com elevado *overhead* afetam o sistema em termos de desempenho e consumo de energia de forma indesejada, alocando

recursos para processos que deveriam apenas analisar o funcionamento do sistema sem, no entanto, afetá-lo.

Os resultados da medição de energia podem ocorrer de maneira *online*, ou seja, resultados obtidos em tempo de execução; ou de maneira *off-line*, com resultados obtidos após a execução de todo o processo. Como em codificação de vídeo o foco principal é a atuação do sistema de controle durante a codificação, o controle de energia alvo neste trabalho deve ser *online*.

As ferramentas de análise do consumo energético podem ser desenvolvidas com base em dois tipos de técnicas: baseadas em modelos e baseadas em medição. Segundo Fischer (2012), técnicas baseadas em modelos utilizam como base para a avaliação do consumo energético um modelo de potência do sistema, relacionando amostras de potência com blocos funcionais, instruções ou funções do sistema, calculando as estimativas de consumo através deste modelo. Como demonstrado em Fischer (2012), diferentes tipos de abordagens baseadas em modelos têm sido proposta, dentre elas:

- **Simuladores** – ferramentas capazes de reproduzir o funcionamento de outro sistema. Podem ser em nível de ciclos de *clock*, arquiteturas, instruções ou do sistema completo.
- **Contagem de eventos** – ferramentas que realizam a contagem de eventos, utilizando dados reais gerados por ferramentas de análise de desempenho como entrada para o modelo. Os contadores efetuam a contagem de eventos significativos em termos de consumo de energia, interpretando este resultado para estimar a energia consumida.
- **Baseadas em estados** – ferramentas que fazem uso de FSM (*Finite State Machine*) e instrumentam o sistema operacional, de forma que o *software* possa rastrear o estado da CPU e dos periféricos. Cada estado tem a potência calibrada e, de acordo com o tempo gasto em cada estado, a energia é determinada.

Técnicas baseadas em medição utilizam ferramentas de instrumentação para a medição dos valores de potência consumida. Um *software* de *profile* deve coletar as informações do sistema alvo e o sistema instrumentado adquire amostras

do consumo de energia do sistema. Essas informações são associadas gerando o perfil do consumo energético do sistema.

Existem algumas ferramentas, como o Gem5 (GEM5, 2014) e o McPAT (LI et.al., 2009), que são essencialmente simuladores. O simulador Gem5 é uma plataforma modular para pesquisa em arquitetura de computadores, englobando a arquitetura em nível de sistema e, também, em nível de microarquitetura do processador (GEM5, 2014). O Gem5 oferece inúmeras ferramentas, uma delas é a modelagem de potência e energia em sistemas computacionais. Os objetos são arranjados no domínio da potência e *clock*, habilitando uma gama de experimentos em termos de eficiência energética.

O McPAT é um sistema de modelagem que integra potência, área e tempo, suportando exploração do espaço de projeto para vários processadores de múltiplos *cores*, com configuração para tecnologias de 90nm a 22nm. Em nível microarquitetural, o McPAT inclui modelos para os componentes fundamentais do chip do multiprocessador, incluindo os núcleos, redes *on-chip*, caches compartilhadas, controladores integrados de memória e múltiplos domínios de *clock*. Em nível de circuitos e tecnologia, o McPAT suporta os caminhos críticos na modelagem, modelagem de área, e modelagem de potência de curto-circuito, potência dinâmica e potência de *leakage* (LI et.al., 2009).

Além do GEM5 e do McPAT, outra ferramenta disponível para medição de energia é o RAPL (*Running Average Power Limit*), a qual é uma ferramenta livre e leva em consideração não somente os modelos, mas também valores *onboard* (PAN, 2015). O RAPL é uma interface de configuração e leitura do consumo de energia dos processadores e memória em um SoC (*System on a Chip*), baseada em registradores MSR (*Model Specific Registers*), desenvolvida para microarquiteturas Intel *Sandy Bridge* e *Ivy Bridge*. Este software é a ferramenta utilizada neste trabalho para medição de energia e terá seu uso detalhado no Capítulo 3. Embora a análise seja toda ela realizada em dispositivos de propósito geral, a maior contribuição deste trabalho reside na metodologia adotada, a qual pode ser estendida para as mais diversas aplicações, inclusive sistemas embarcados.

2.4 Trabalhos Relacionados

Alguns trabalhos na literatura, como Ma (2011), Li (2011) e Lee (2012), preocupam-se com a análise do consumo de energia em codificadores/decodificadores de vídeo. Em Ma (2011), é empregado um esquema do tipo DVFS (*Dynamic Voltage Frequency Scaling*), num processador ARM Cortex-A8. A informação sobre a potência instantânea dissipada por um decodificador H.264/AVC é obtida através desse sistema. Já o trabalho desenvolvido por Li (2011), utiliza medições de complexidade de um codificador AVS (*Audio Video coding Standard*) para estimar o consumo de energia. Por outro lado, Lee (2012) utiliza um simulador baseado em *software* para estimar o consumo de energia de um codificador H.264/AVC. Observe que, embora esses trabalhos se preocupem com o consumo de energia em sistemas de codificação de vídeo, focam sua pesquisa em codificadores de vídeo da geração passada, os quais, ainda que muito utilizados atualmente, possuem a tendência natural de terem sua aplicação cada vez mais reduzida, sendo gradativamente substituídos por codificadores mais atuais, como o HEVC.

Focando em sistemas de codificação atuais, outros trabalhos desenvolvem perfis de consumo de complexidade do codificador HEVC. Em Corrêa (2012) é realizado um estudo da eficiência de codificação relacionado à complexidade computacional do codificador HEVC, identificando as ferramentas que mais influenciam a eficiência de codificação e a complexidade computacional. Já Bossen (2012), realiza um *profile* da complexidade tanto do codificador quanto do decodificador HEVC, a partir de simulações realizadas em um processador Intel Xeon. Embora tenham realizado estudos interessantes, esses trabalhos não consideram consumo de energia nem, tampouco, propõem sistemas de controle (nem de energia nem de complexidade).

Na literatura existem muitos trabalhos que se preocupam com a redução e controle da complexidade do codificador HEVC. Grellert (2014) investiga técnicas de controle de complexidade para codificadores HEVC, a partir de uma análise do esforço computacional associado a parâmetros-chave da codificação, utilizando algumas métricas de comparação, bem como propõe um sistema de controle de complexidade para codificadores HEVC que se adapta dinamicamente às restrições computacionais da plataforma. Corrêa (2014) realiza uma avaliação de desempenho

e complexidade do codificador HEVC, propondo algoritmos de redução e métodos de controle de complexidade, levando em consideração o compromisso com a manutenção de taxas de compressão razoáveis. Esses trabalhos, além de proporem estudos de complexidade, propõem também sistemas de controle de complexidade, embora deixem de lado a questão do consumo de energia. Ainda que estes trabalhos tenham suas maiores contribuições no que diz respeito à complexidade do codificador, pode-se estender suas análises com relação ao consumo de energia, uma vez que a energia consumida e complexidade da codificação estão intimamente ligadas. Cabe salientar que, apesar de tempo e energia estarem relacionados, uma análise sobre energia desvinculada do tempo é muito importante, uma vez que em sistemas heterogêneos a relação entre tempo e energia não ocorre de forma linear.

Considerando as ferramentas de medição de energia apresentadas, foram encontrados trabalhos como o apresentado por Hahnel (2012), o qual realiza uma comparação demonstrando o comportamento do RAPL em face a uma medição real, bem como o que é discutido por Venkatesh (2013), no qual o RAPL é utilizado para medição de energia de um sistema de transmissão de mensagens.

Mais especificamente, foram encontrados apenas dois trabalhos que utilizam a ferramenta RAPL em conjunto com o codificador HEVC. Monteiro (2014) realiza uma avaliação do consumo energético do codificador HEVC variando parâmetros ligados à área de busca, níveis de quantização e resolução. Neste trabalho o RAPL é utilizado como uma aplicação paralela que realiza a medição de energia de um processo específico (codificação dos vídeos). Dessa forma é traçado um perfil de consumo de energia para o HEVC frente essa variação de parâmetros. Esse trabalho também propõe algumas métricas de comparação relacionando a variável energia. Uma extensão desse trabalho, presente em Monteiro (2015), estende essas mesmas avaliações para o padrão de codificação de vídeos anterior ao HEVC, o H.264/AVC.

No entanto, nenhum dos trabalhos relacionados propõe um controlador para otimizar o consumo de energia do codificador HEVC. Assim, dada a inexistência de trabalhos relacionados, e a relevância do tema, este trabalho desenvolve um controlador dinâmico de energia, aliando técnicas já conhecidas de redução de complexidade à redução do consumo de energia, bem como integrando a ferramenta RAPL para análise do consumo de energia.

Existem na literatura diversos trabalhos que realizam controle de taxa de bits através da variação do QP para o codificador HEVC. Xu (2008) propõe um algoritmo para adaptação da taxa de bits a um determinado valor através da variação do QP. Sun (2014) também propõe um esquema de controle adaptativo para o controle da taxa de bits utilizando técnica semelhante. Além disso, Choi (2013) realiza um controle multi-nível de taxa de bits alterando também valores de QP. Uma vez que a variação do QP também afeta o consumo energético é possível o desenvolvimento de um controlador de energia simples, baseado nessa técnica. Neste trabalho é desenvolvido também um controlador de energia baseado em QP com a finalidade de proporcionar uma comparação justa com o controlador dinâmico desenvolvido, reiterando mais uma vez a inexistência de trabalhos relacionados.

3 METODOLOGIA UTILIZADA PARA MEDIÇÃO DE ENERGIA NO PROCESSO DE CODIFICAÇÃO DO HEVC

O *Running Average Power Limit* (RAPL) é uma ferramenta para medição do consumo de energia, disponível para processadores Intel *Sandy Bridge* e *Ivy Bridge*, se comportando como um *driver* de uma arquitetura *x86* que se liga em conjunto com componentes da CPU.

Consiste basicamente em uma forma de acessar registradores específicos do processador que têm por finalidade medir e registrar valores de potência e energia. Basta que, através de um código em *C++*, escreva-se um *script* que realize acesso a estes registradores específicos os quais, por padrão, estão desabilitados. Através da compilação deste código gera-se um arquivo executável, o qual atua no processo de medição de potência e energia habilitando os registradores MSR.

O RAPL retorna valores fixos relacionados a limites especificados nos registradores, como, por exemplo, valores de potência termal, energia e tempo das unidades e, também, valores que se alteram conforme a utilização dos recursos da CPU, como potência e energia consumidas. Os componentes que podem ter seus consumos energéticos mensurados são: *(i) rapl_pkg*: todo o *package* da CPU; *(ii) rapl_pp0*: somente os núcleos dos processadores; *(iii) rapl_pp1*: dispositivos específicos *uncore* (funções do processador que não estão no núcleo, geralmente unidades de processamento gráfico); *(iv) rapl_dram*: controlador de memória. Isto permite traçar o perfil de energia de uma dada aplicação (MONTEIRO et. al., 2014).

O RAPL pode ser configurado alterando-se os códigos do mesmo, de modo que possa haver a interação deste programa com o *software* de referência do HEVC, o HM, tornando possível a medição do consumo de energia.

Dessa forma, essa ferramenta pode ser utilizada para efetuar medições de energia do sistema. As medições podem ser efetuadas de forma paralela, com o RAPL funcionando separadamente do codificador, ou de modo sequencial, atuando em tempo de processo, integrado ao codificador. Neste trabalho, ambas análises serão necessárias, a primeira com foco na análise de sensibilidade à variação de parâmetros e a segunda com foco na utilização em conjunto com o codificador.

3.1 Utilização do RAPL

Analisando o código do RAPL, pode ser verificado que ele é composto por um *loop* infinito, o qual é finalizado por uma função chamada SIGTERM. Esta função, além de finalizar o programa, é também responsável por chamar outra função para imprimir os resultados antes da finalização do processo. Para ativar a função SIGTERM e obter o resultado desejado, é necessário finalizar o processo do RAPL utilizando um comando *kill*. Dessa forma, a maneira mais prática de obter o consumo de energia de um processo de codificação de um vídeo inteiro é realizar um *script* de leitura de energia que execute o código do RAPL, salve a identificação do processo numa variável, inicie o processo de codificação, meça periodicamente e, finalmente, finalize o processo através do comando *kill*. O pseudocódigo do *script* de leitura de energia é apresentado na Figura 12.

```

Ler_energia(video)
01 executar rapl // no core 0
02 codificar video com AMP, FME, HadME, MaxCUd,SR, MaxTUd // no core 1
03 kill process

rapl()
01 signal(SIGTERM, Imprime)
02 Leitura_anterior ← Energia_consumida
03 enquanto(1) // permanece no laço até receber o comando kill
04   Espera (1 segundo)
05   Leitura_atual ← Energia_consumida
06   se (Leitura_atual > Leitura_anterior)
07     Consumo_total = Leitura_atual - Leitura_Anterior
08   Leitura_anterior ← Leitura_atual
09   imprime()
10   Imprimir Consumo_total
11 fim enquanto

```

Figura 12 – Pseudocódigo de leitura de energia do processo de codificação

A informação do consumo de energia de cada *core* não podem ser obtidas separadamente, uma vez que o domínio do *core* é único, assim, o RAPL fornece informações somente a respeito do consumo de energia de todos os processos que

estiverem sendo executados pelo processador. Este fato faz com que alguns aspectos devam ser levados em consideração.

Um dos aspectos mais críticos é ter certeza que a CPU execute a menor quantidade de processos possível, de modo que o consumo de tais programas interfira o mínimo possível na análise energética. É aconselhável também a execução individual do RAPL para realizar um *profile* de energia do computador no qual será executado o *software* do HEVC. Dessa forma, é possível saber o quanto de energia o Sistema Operacional (SO) está consumindo, bem como programas que estejam sendo executados paralelamente (especialmente em *background*) ao processo de codificação. Isto possibilita isolar o consumo de energia do processo de codificação. Entretanto, este processo deve ser executado em cada plataforma que se deseja realizar a codificação de vídeo, uma vez que o consumo energético de CPUs diferentes será também diferente, em virtude de configurações e programas sendo executados em paralelo.

Dessa forma, uma boa prática é isolar o processo do RAPL em um *core* e o *software* do HEVC em outro *core*. Isto garante que os processos não serão comutados de um *core* para outro pelo sistema operacional e, também, que outros programas não interfiram no processo de codificação. Além disso, a frequência de operação da CPU também deve ser configurada para um valor fixo, de modo que o SO não altere as frequências de cada *core*, evitando picos de processamento, proporcionando uma simulação e leituras mais lineares. A Figura 13 apresenta a metodologia utilizada, representada em diagrama de blocos, de medição de energia consumida pelo codificador HEVC em conjunto com o RAPL.

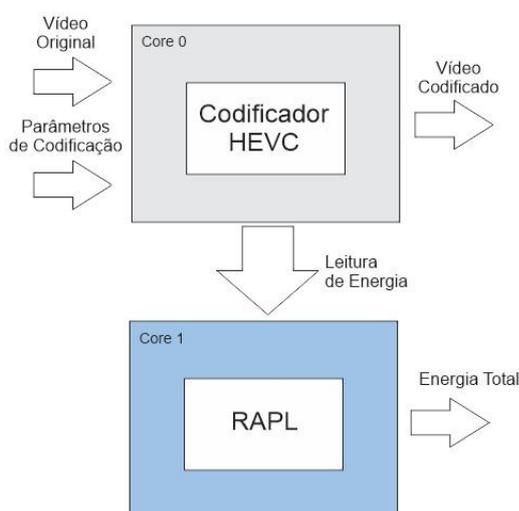


Figura 13 – Metodologia paralela de medição de energia

O método descrito na Figura 13 consiste numa análise paralela de medição de energia, ou seja, o RAPL e o HM trabalham simultaneamente em cores diferentes. Apesar de útil para traçar o perfil de consumo energético do codificador HEVC, esta metodologia possui um entrave para trabalhar em conjunto com o controlador. Embora os dados possam ser coletados, não existe a possibilidade de usá-los uma vez que estes dados existem em um processo separado. Para que as informações colhidas pelo RAPL possam ser utilizadas no codificador HEVC, ambas as aplicações devem coexistir num mesmo processo, assim, é necessária a integração entre o RAPL e o HM.

3.2 Integração com o HM

Para permitir que os resultados de consumo de energia do codificador HEVC sejam utilizados como entrada do controlador proposto neste trabalho, o RAPL deve ser integrado ao *software* de referência do codificador HEVC.

O primeiro procedimento para efetuar a integração é transcrever o código do RAPL como uma função que retorne o valor do consumo de energia a cada intervalo de tempo (qualquer múltiplo ou submúltiplo de unidade de tempo), através de uma amostragem periódica. Dessa forma, pode-se criar um objeto RAPL dentro do codificador, permitindo acesso aos valores das leituras de energia efetuadas pelo *software* de medição de energia.

Como o valor de energia é um valor acumulado, o resultado final do processo de medição será o valor lido na última etapa do processo de codificação. Para se determinar o valor de energia consumida em um determinado intervalo deve-se realizar a diferença entre o valor indicado por uma leitura atual e uma temporalmente pregressa. Com a integração do RAPL com o HM é possível utilizar, de modo sequencial, os valores de energia lidos como dados de entrada do controlador. A Figura 14 apresenta a metodologia de leitura de energia sequencial, com a integração entre o *software* de medição de energia e o de codificação de vídeo.

Ambas as metodologias apresentadas (paralela e sequencial) serão utilizadas nas etapas de análise de sensibilidade e no desenvolvimento do controlador, respectivamente, conforme será apresentado nos Capítulos 4 e 5.

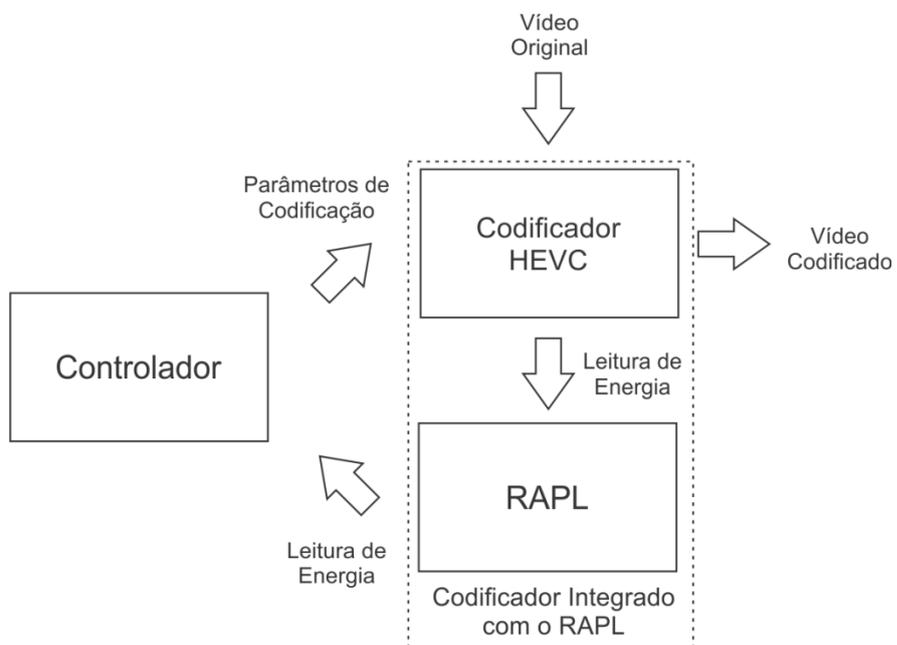


Figura 14 - Metodologia sequencial de medição de energia

4 ANÁLISE DE SENSIBILIDADE DO CONSUMO DE ENERGIA DO CODIFICADOR HEVC

Este capítulo apresenta, primeiramente, alguns aspectos importantes sobre o codificador HEVC, enumerando e explicando os parâmetros de codificação que mais impactam no consumo de energia do codificador. A análise de sensibilidade consiste em avaliar o comportamento do codificador, do ponto de vista do consumo de energia, utilizando uma metodologia de medição paralela, frente à variação de parâmetros de codificação. Os resultados desse processo são importantes para o desenvolvimento do controlador, apresentado no Capítulo 5, o qual é a principal contribuição deste trabalho.

4.1 Parâmetros de Codificação do Codificador HEVC

O HEVC possui diversos parâmetros de codificação, sendo que valores diferentes nos parâmetros contribuem para diferentes valores de consumo de energia. Grellert (2014) define alguns parâmetros que são mais relevantes em termos de complexidade computacional no codificador HEVC. Como complexidade envolve a variável tempo e , sendo a energia função do tempo, parâmetros que afetam a complexidade também afetam a energia. Portanto, tais parâmetros são fortes candidatos como principais componentes do consumo energético do codificador HEVC.

- **Max Partition Depth (MaxCUd):** este parâmetro define a profundidade máxima da CTU. Cada novo nível de profundidade representa uma nova subdivisão das CUs. Por padrão este parâmetro é definido com valor 4, começando por uma tamanho 64x64. As CUs suportam até 3 subdivisões, chegando a um tamanho 8x8. O MaxCUd impacta diretamente no esforço computacional e consumo de energia, tanto nas etapas de predição quanto na codificação residual.
- **Search Range (SR):** parâmetro utilizado na determinação da área de busca na Estimção de Movimento (*Motion Estimation* – ME). Este parâmetro está relacionado ao número de blocos candidatos avaliados na ME, reduzindo ou aumentando o consumo na etapa de predição

inter-quadros e também no acesso à memória externa.

- ***Asymmetric Motion Partition (AMP)***: parâmetro utilizado para habilitar/desabilitar a avaliação das partições assimétricas para as CUs na predição inter-quadros. Embora versões mais recentes do HM apresentem heurísticas para a redução do número de partições assimétricas avaliadas, o custo RD (*Rate Distortion*) das AMPs ainda é calculado muitas vezes para cada quadro.
- ***Hadamard ME (HadME)***: parâmetro que habilita o cálculo de SATD durante a ME fracionária. Este parâmetro afeta principalmente na etapa de predição inter-quadros, visto que o custo computacional e energético da SATD é maior que o custo da SAD.
- ***Max TU Depth (MaxTUd)***: parâmetro que define a profundidade máxima da árvore das TUs. Isto afeta o esforço computacional e o consumo energético dos módulos de transformadas, quantização e entropia, visto que a quantidade das operações realizadas nestes módulos varia com o MaxTUd.
- ***Fractional Motion Estimation (FME)***: não é implementada como um parâmetro no HM, ou seja, não é configurável por arquivos externos, entretanto, pode ser modificada alterando o código fonte do HM. Consiste em habilitar ou desabilitar a estimação de movimento fracionária no processo de codificação.

4.2 Resultados da Análise de Sensibilidade

Com o objetivo de mensurar a importância de cada parâmetro para o consumo de energia e eficiência da codificação, uma análise de sensibilidade deve ser realizada. Este tipo de análise consiste em traçar uma curva de como uma variável específica (energia, por exemplo) varia com a mudança dos parâmetros com relação a um modelo de referência. Neste trabalho, o modelo de referência é a configuração padrão do HM, incluindo modos de decisão rápida, habilitados no próprio codificador.

Os resultados discutidos neste capítulo foram obtidos a partir de um vasto conjunto de experimentos, realizados em uma plataforma *Intel Core i5-3330, 3.00GHz, 4 cores, 4 GB Memória RAM*, em concordância com as CTCs. O principal

foco da análise de sensibilidade é verificar o impacto individual de cada parâmetro no consumo de energia do codificador. Para realizar este processo, somente um subconjunto de todas as sequências recomendadas pelas CTCs foi utilizado, em virtude da quantidade proibitiva de simulações que seriam necessárias para codificar as 24 sequências das condições de teste. Para se ter ideia, em média um teste completo para uma sequência leva 600h. Caso fosse utilizada a CCT completa o tempo total seria equivalente a 14400h (600 dias).

A Tabela 1 apresenta a configuração das condições de teste utilizadas. Foram escolhidas uma sequência de cada classe em virtude das classes possuírem resoluções distintas e variadas taxa de quadros. Além disso, vídeos da classe F apresentam como característica comum muitas mudanças de cena e iluminação. Não foi utilizado nenhum vídeo da classe E, pois os vídeos dessa classe apresentam grande similaridade entre quadros da sequência, uma vez que apresentam planos de fundo sem qualquer modificação, o que faz com que sequências da classe E não sejam interessantes para traçar um perfil de energia em virtude dessa natureza. Uma vez que os quadros são similares, o custo energético, principalmente nas etapas de predição, é significativamente menor.

Tabela 1 – Configuração para as condições de teste

Classe	Resolução	Sequências	Taxa de Quadros (fps)	Número de Quadros
A	2560x1600	<i>PeopleOnStreet</i>	30	64
B	1920x1080	<i>Kimono</i>	24	64
C	832x480	<i>BasketballDrill</i>	50	64
D	416x240	<i>BlowingBubbles</i>	50	64
F	1024x768	<i>ChinaSpeed</i>	30	64

As configurações utilizadas na análise de sensibilidade do consumo de energia do codificador estão presentes na Tabela 2 e seguem o modelo apresentado por Grellert (2014). A configuração padrão *c0*, utilizada como ponto de partida, serviu como modelo de comparação para as demais, e as configurações restantes foram definidas pela variação individual de um único parâmetro (de *c1* a *c10*), bem como a

combinação de alguns deles (*c11*). Com base nisso, foram elaboradas 12 configurações, conforme pode ser observado na própria Tabela 2. Os parâmetros foram modificados com o objetivo de reduzir o consumo energético do codificador, quando comparados com a configuração padrão, o qual é o principal objetivo deste trabalho.

Tabela 2 – Configurações utilizadas na análise de sensibilidade

	<i>c0</i>	<i>c1</i>	<i>c2</i>	<i>c3</i>	<i>c4</i>	<i>c5</i>	<i>c6</i>	<i>c7</i>	<i>c8</i>	<i>c9</i>	<i>c10</i>	<i>c11</i>
AMP	1	0	1	1	1	1	1	1	1	1	1	1
FME	1	1	0	1	1	1	1	1	1	1	1	1
HadME	1	1	1	0	1	1	1	1	1	1	1	1
MaxCUd	4	4	4	4	1	2	3	4	4	4	4	3
SR	64	64	64	64	64	64	64	16	32	8	64	64
MaxTUd	3	3	3	3	3	3	3	3	3	3	1	2

A configuração *c0* corresponde às condições de teste inicialmente encontradas nos arquivos de configuração do HM. As demais configurações foram desenvolvidas com o objetivo de codificar os vídeos com um consumo de energia inferior ao consumo de *c0*. Dessa forma, pode-se observar quão menor é o consumo de energia para cada configuração. Uma vez que existem 12 configurações, cada uma delas foi usada para codificar as cinco sequências informadas na Tabela 1, para os quatro QPs recomendados pelas CTCs, na configuração temporal *Low-Delay*, e repetidas cinco vezes para uma melhor análise estatística, logo, foram realizadas 1200 simulações para traçar a análise de sensibilidade, totalizando aproximadamente 3000 horas de simulação.

O objetivo dessa análise é verificar a redução do consumo de energia proporcionada por cada configuração bem como as perdas em PSNR e aumento de *bit-rate* associados a cada uma, permitindo uma análise da eficiência de codificação. A Tabela 3 apresenta dados médios do consumo de energia e desvio padrão de cada configuração para cada QP avaliado. Pode-se observar que o consumo varia com a alteração dos QPs, em virtude do nível de quantização adotado. Para QPs mais altos, o consumo de energia diminui, uma vez que também diminui a qualidade

da imagem. Observa-se também que os valores dos desvios padrão (σ) são consideravelmente baixos, o que indica um comportamento uniforme de consumo de energia no processo de codificação.

Tabela 3 – Energia média consumida e desvio padrão em cada configuração

Config.	QP 22		QP 27		QP 32		QP 37		Valores Médios	
	Energia (J)	σ (%)	Energia Média (J)	σ médio (%)						
<i>c0</i>	77021	1,40	57321	0,20	50256	0,29	44443	0,20	57260	0,52
<i>c1</i>	64582	0,30	48018	0,14	43238	0,31	39494	0,28	48833	0,26
<i>c2</i>	45619	0,44	35584	0,22	29051	0,60	24889	0,29	33786	0,39
<i>c3</i>	69730	0,20	50307	0,40	43465	0,24	38114	0,23	50404	0,27
<i>c4</i>	15749	0,21	12716	0,23	11824	0,28	11402	0,25	12923	0,24
<i>c5</i>	31282	0,26	24417	0,27	22455	0,37	19967	0,21	24530	0,28
<i>c6</i>	57086	0,23	43712	0,21	38881	0,24	35025	0,16	43676	0,21
<i>c7</i>	73453	0,25	53273	0,25	46506	0,23	41348	0,35	53645	0,27
<i>c8</i>	64920	0,24	54488	0,20	47387	0,26	42402	0,19	52299	0,22
<i>c9</i>	62887	0,27	52581	0,20	45799	0,32	40978	0,27	50561	0,26
<i>c10</i>	53277	0,27	46434	0,24	41971	0,25	37469	0,25	44788	0,25
<i>c11</i>	42060	0,28	37330	0,26	34192	0,28	30942	0,39	36131	0,30

Na Figura 15 é apresentado o percentual de economia de energia para cada estado com relação à *c0*. Observa-se que alguns estados proporcionam maior economia de energia com relação a outros. O estado *c4*, por exemplo, é o que proporciona maiores valores percentuais, atingindo quase 80% de economia de energia. Outros estados apresentam valores medianos, como *c2*, *c5*, *c6*, *c10* e *c11*, os demais apresentam resultados significativamente mais baixos de economia de energia.

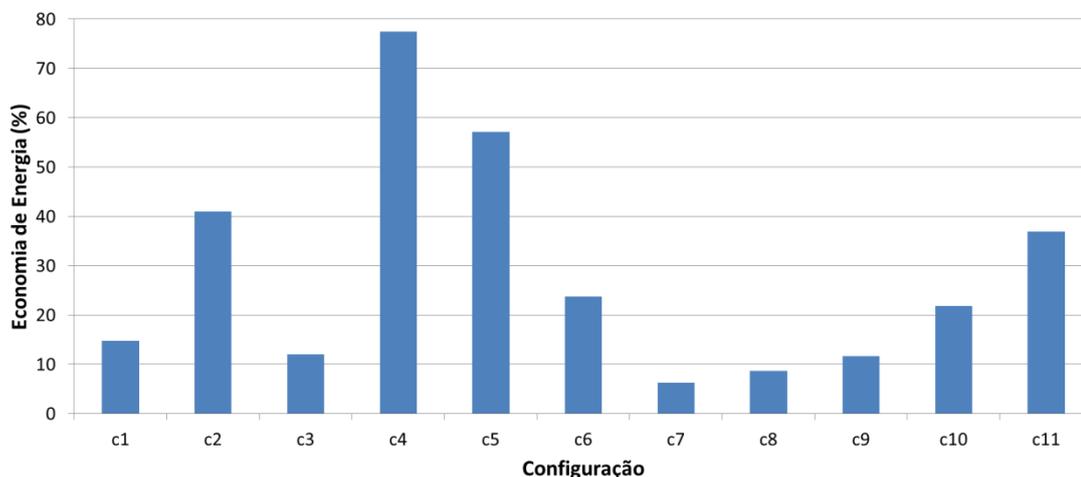


Figura 15 – Economia de energia média com relação à *c0*

No entanto, uma simples análise do consumo de energia não basta para definir se determinada configuração é adequada ou não para ser utilizada pelo processo de codificação. Uma análise mais criteriosa é necessária, a qual deve levar em consideração outras métricas de comparação, como BD-BR e BD-PSNR. Na Tabela 4 são apresentados os valores médios de economia de energia, BD-BR e BD-PSNR para cada configuração proposta.

Conforme pode ser observado, embora a configuração *c4* seja a que apresenta maior economia de energia, ela também é a configuração que apresenta maior aumento de BD-BR, se for analisado um cenário com qualidade (PSNR) constante, bem como a configuração que mais deteriora a imagem (BD-PSNR), se analisado um cenário com *bit-rate* constante. A análise da eficiência de codificação pode levar em consideração apenas um dos parâmetros (BD-BR ou BD-PSNR), uma vez que ambos fornecem a mesma informação, vista por ângulos distintos. Este resultado demonstra que esta configuração, apesar de economizar energia, apresenta um impacto significativo na eficiência de codificação, podendo não ser interessante em alguns cenários onde a qualidade do vídeo codificado seja prioridade. Logo, um estudo mais aprofundado é necessário para encontrar as configurações com um melhor comprometimento entre economia de energia, taxa de bits e qualidade.

Tabela 4 – Valores médios de BD-BR e BD-PSNR para cada configuração

Configuração	Economia de Energia (%)	BD-BR (%)	BD-PSNR (dB)
<i>c1</i>	14,72	0,8517	-0,0354
<i>c2</i>	41,00	6,9591	-0,2489
<i>c3</i>	11,97	1,3011	-0,0526
<i>c4</i>	77,43	68,1340	-2,0422
<i>c5</i>	57,16	27,6012	-1,0280
<i>c6</i>	23,72	6,0237	-0,2663
<i>c7</i>	6,31	0,2308	-0,0114
<i>c8</i>	8,66	0,0111	-0,0009
<i>c9</i>	11,70	35,2452	-0,3268
<i>c10</i>	21,78	1,7769	-0,0822
<i>c11</i>	36,90	11,4484	-0,4758

Após a análise sensibilidade inicial (baseada em uma abordagem clássica da teoria de controle, a qual consiste na avaliação individual de cada variável) tem-se, portanto, uma estimativa da quantidade de energia que cada configuração utiliza

individualmente. Pode-se então realizar combinações diversas entre parâmetros de codificação através de análise combinatória, potencializando os efeitos da economia de energia, de modo que diversos valores sejam atingidos. Um conjunto de parâmetros foi definido, chamados neste trabalho de *Parameter Set* (PS), os quais combinam diversas configurações a fim de obter níveis específicos de economia de energia. A Tabela 5 apresenta os 32 PS propostos por este trabalho (não foram utilizadas todas as possibilidades em virtude do tempo de execução das mesmas e a inter-relação entre parâmetros de codificação), dispostos em ordem decrescente de consumo de energia.

Tabela 5 – Conjunto de parâmetros (PS) propostos

<i>Parameter Set</i>	Configurações	Parâmetros					
		AMP	FME	HadME	MaxCud	SR	MaxTud
PS0	c0	1	1	1	4	64	3
PS1	c8	1	1	1	4	32	3
PS2	c7	1	1	1	4	16	3
PS3	c9	1	1	1	4	8	3
PS4	c1	0	1	1	4	64	3
PS5	c3	1	1	0	4	64	3
PS6	c10	1	1	1	4	64	1
PS7	c1, c7	0	1	1	4	16	3
PS8	c6	1	1	1	3	64	3
PS9	c3, c7	1	1	0	4	16	3
PS10	c1, c3	0	1	0	4	64	3
PS11	c2, c8	1	0	1	4	32	3
PS12	c1, c3, c7	0	1	0	4	16	3
PS13	c1, c3, c8	0	1	0	4	32	3
PS14	c1, c10	0	1	1	4	64	1
PS15	c6,c7	1	1	1	3	16	3
PS16	c1, c2	0	0	1	4	64	3
PS17	c11	1	1	1	3	64	2
PS18	c1, c7, c10	0	1	1	4	16	1
PS19	c3, c10	1	1	0	4	64	1
PS20	c3, c7, c10	1	1	0	4	16	1
PS21	c2	1	0	1	4	64	3
PS22	c2, c7	1	0	1	4	16	3
PS23	c1, c6, c10	0	1	1	3	64	1
PS24	c1, c3, c6	0	1	0	3	64	3
PS25	c1, c2, c10	0	0	1	4	64	1
PS26	c5	1	1	1	2	64	3
PS27	c2, c10	1	0	1	4	64	1
PS28	c2, c6, c8	1	0	1	3	32	3
PS29	c1, c2, c7, c10	0	0	1	4	16	1
PS30	c2, c6, c10	1	0	1	3	64	1
PS31	c4	1	1	1	1	64	3

O parâmetro PS0 (c_0) representa a referência para as demais, uma vez que utiliza as configurações padrões do HM. Foram analisados os resultados de consumo de energia, BD-BR e BD-PSNR de cada PS. Os resultados são apresentados na Tabela 6 e representados de maneira gráfica nas Figuras 16, 17 e 18, contemplando os resultados de energia, BD-BR e BD-PSNR, respectivamente.

Tabela 6 – Resultados para cada PS

Parameter Set	Energia Normalizada	BD-BR (%)	BD-PSNR (dB)
PS0	1,0000	0,0000	0,0000
PS1	0,9284	0,0111	-0,0009
PS2	0,9151	0,2308	-0,0114
PS3	0,8943	35,2452	-0,3268
PS4	0,8472	0,8517	-0,0354
PS5	0,8323	1,3011	-0,0526
PS6	0,7986	1,7769	-0,0822
PS7	0,7849	1,1618	-0,0501
PS8	0,7501	6,0237	-0,2663
PS9	0,7206	1,5075	-0,0381
PS10	0,7136	1,6821	-0,0610
PS11	0,6940	7,8664	-0,2840
PS12	0,6659	1,8257	-0,0685
PS13	0,6634	1,7288	-0,0636
PS14	0,6612	2,3553	-0,1062
PS15	0,6576	6,3891	-0,2820
PS16	0,6459	8,7740	-0,3176
PS17	0,6431	11,4484	-0,4758
PS18	0,6196	2,6007	-0,1175
PS19	0,6132	2,1930	-0,0870
PS20	0,6067	2,4694	-0,0993
PS21	0,6017	6,9591	-0,2489
PS22	0,5188	8,0283	-0,2916
PS23	0,5152	14,7206	-0,5900
PS24	0,5143	6,8404	-0,2827
PS25	0,4572	10,5237	-0,3869
PS26	0,4158	27,6012	-1,0280
PS27	0,4066	9,6874	-0,3565
PS28	0,3901	13,4395	-0,5092
PS29	0,3840	11,0182	-0,4087
PS30	0,2981	25,0067	-0,9301
PS31	0,2146	68,1340	-2,0422

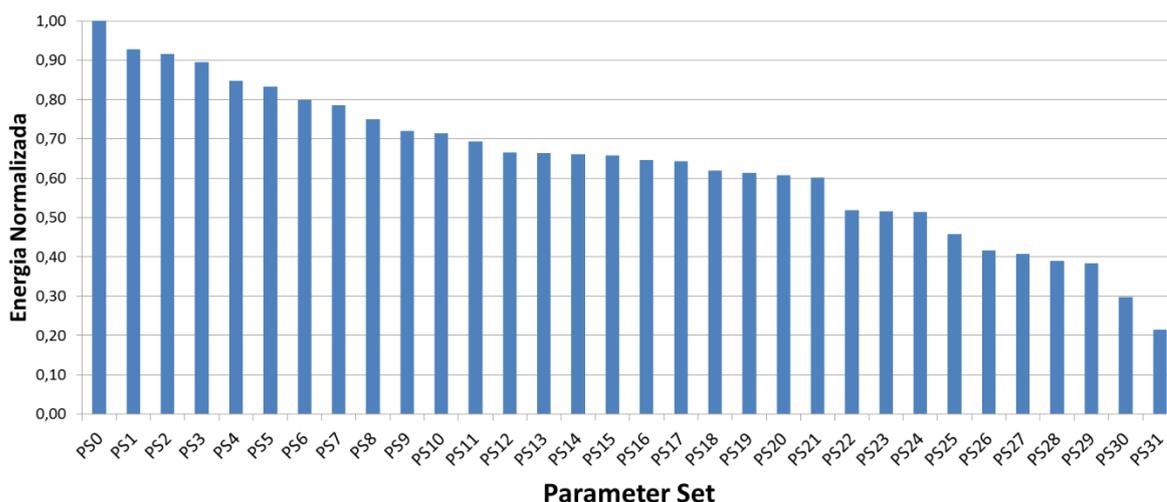


Figura 16 – Energia normalizada para cada PS

Conforme pode ser observado, conforme se aumenta o índice, cada PS proporciona gradativamente maior economia de energia. Através do conjunto de parâmetros proposto foi atingida uma energia normalizada de até 0,2146, obtida com o PS 31, o que representa uma economia significativa, de até 78,54% no consumo de energia do codificador.

No entanto, conforme o que já foi exposto por este trabalho, a análise de consumo energético deve ser acompanhada da avaliação de impacto causado na eficiência de codificação. A Tabela 6 também apresenta valores de BD-BR e BD-PSNR para cada PS avaliado, graficamente apresentados nas Figuras 17 e 18, respectivamente.

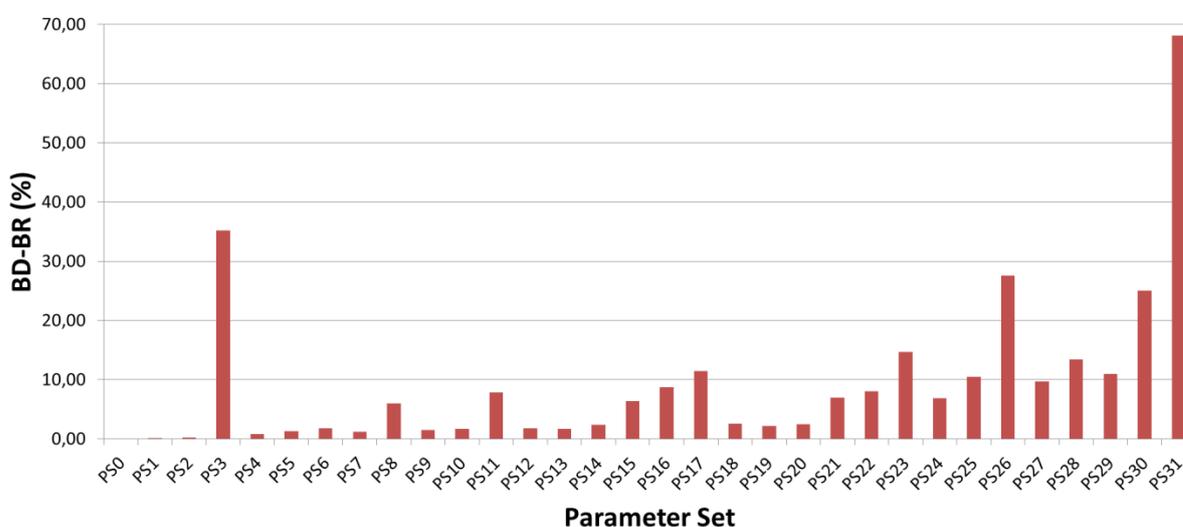


Figura 17 – BD-BR para cada PS

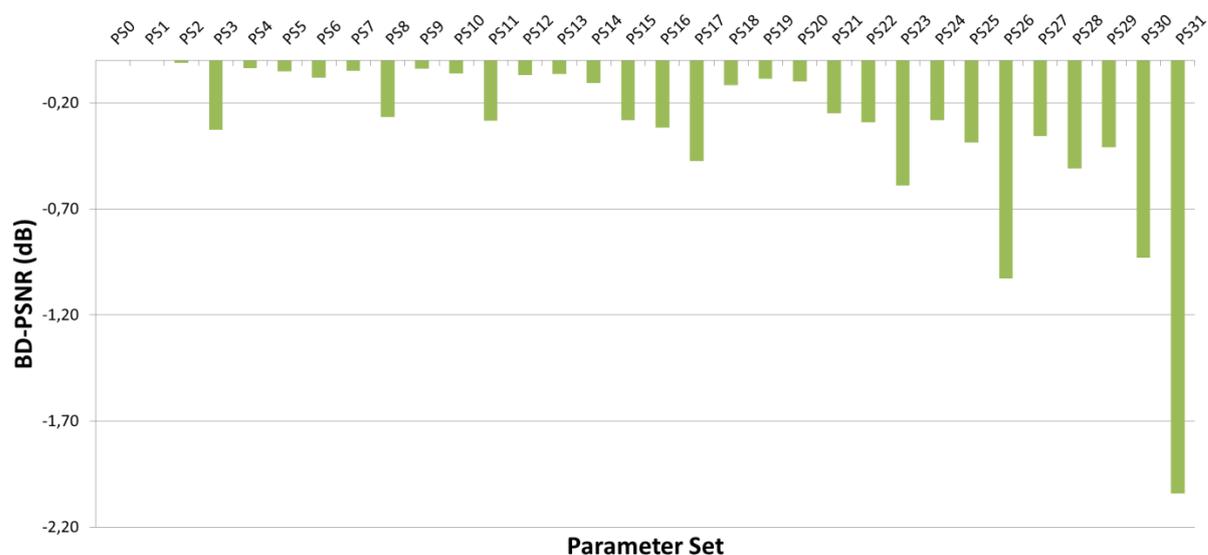


Figura 18 – BD-PSNR para cada PS

Pode-se observar, de maneira geral, que quanto maior é a economia de energia menor é a eficiência de codificação, no entanto, isso não é regra geral. Os parâmetros PS3, PS15, PS16 e PS17, por exemplo, apresentam tanto eficiência de codificação quanto redução no consumo de energia inferiores a de parâmetros como PS4 e PS19, por exemplo.

A utilização de todos os PS como pontos de operação é ineficiente, uma vez que algumas condições são melhores que outras (levando em consideração a relação entre consumo de energia e eficiência de codificação - BD-BR / BD-PSNR). Portanto, uma seleção criteriosa sobre esses pontos é necessária, conforme será visto no Capítulo 5.

5 CONTROLADOR DINÂMICO DE ENERGIA APLICADO AO CODIFICADOR HEVC

Neste capítulo será apresentada toda a metodologia utilizada para o desenvolvimento do controlador dinâmico de energia para o codificador HEVC.

A Figura 19 apresenta um diagrama em alto nível do sistema de controle de energia proposto. As entradas do sistema de controle são a energia desejada (*set point* – SP) por GOP (*Group of Pictures*), o qual é definido no próprio controlador (conforme será visto a seguir), e a leitura da energia consumida pelo codificador HEVC, utilizando o RAPL de forma sequencial.

Nessa representação simplificada, o codificador HEVC recebe como entrada o vídeo original e os parâmetros de codificação. As saídas do codificador são o próprio vídeo codificado e o valor de energia consumida por GOP, através da utilização do RAPL. A cada GOP o RAPL envia informações do consumo energético para o controlador, que define os parâmetros de codificação do próximo GOP através do cálculo do fator de redução do consumo de energia e da determinação de novos pontos de operação do codificador.

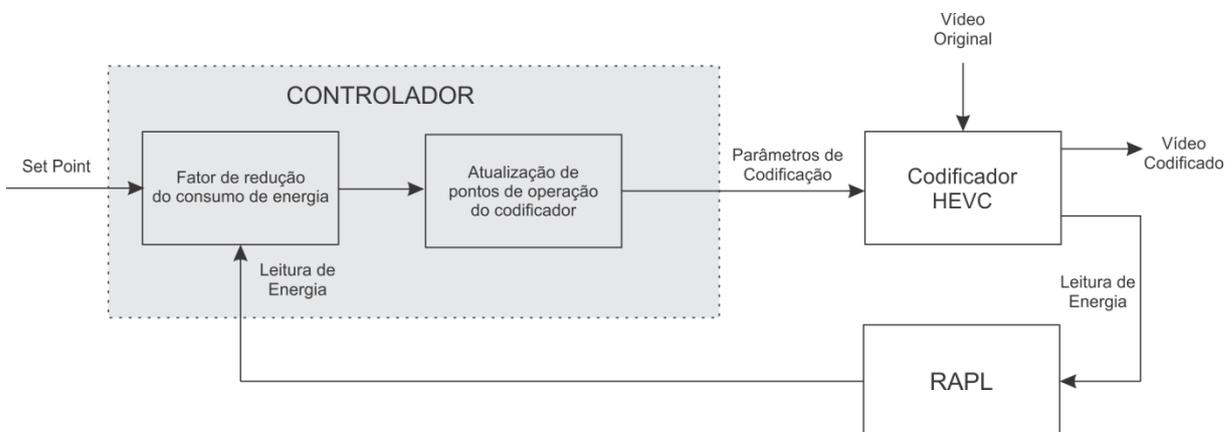


Figura 19 – Sistema de controle de energia integrado com o codificador HEVC

5.1 Determinação dos pontos de operação do codificador

O objetivo principal deste capítulo é encontrar os pontos ótimos de operação (que proporcionam maior economia de energia com a menor distorção e o menor incremento na taxa de bits possíveis) através da utilização do conceito de frentes de Pareto.

O projeto do sistema de controle do controlador de energia inicia pela definição de um conjunto de pontos de operação, em que cada ponto possui uma

configuração distinta, combinando diversas ferramentas do codificador (baseado no que foi definido no Capítulo 4). Dessa forma, cada configuração é testada num processo de treinamento *off-line*, a partir do qual é possível realizar a definição da melhor configuração do ponto de vista do custo R-D, bem como do consumo de energia. O algoritmo de controle de energia implementado neste trabalho usa essas configurações para ajustar o valor de energia consumida pelo codificador HEVC para processar cada GOP (cada GOP possui quatro quadros, conforme estrutura de predição temporal selecionada).

Cada uma das 32 configurações propostas no capítulo anterior foi utilizada para codificar as cinco sequências de vídeo apresentadas anteriormente, com os QPs 22, 27, 32 e 37, na configuração temporal *LowDelay*, na mesma plataforma utilizada na análise de sensibilidade. Os valores de consumo de energia, *bit-rate* e PSNR foram medidos e utilizados no cálculo do valor médio de BD-BR e BD-PSNR para cada configuração, conforme apresentado no Capítulo 4.

Numa implementação com um codificador real são utilizadas análises com *bit-rate* constante (com qualidade variável) ou com qualidade constante (com *bit-rate* variável), dessa forma, nem todos os pontos em uma análise R-D-E devem ser considerados simultaneamente na análise para encontrar a configuração com melhor desempenho. Uma análise R-D-E, que ocorreria num espaço bidimensional com qualidade ou taxa de bits normalizados, pode ser simplificada pela eliminação da taxa de bits ou do PSNR da análise, através da comparação de resultados BD em duas análises independentes. A primeira análise considera um cenário no qual a taxa de bits é constante, fornecendo um estudo no espaço *Distortion-Energy* (D-E), enquanto a segunda análise considera um cenário no qual a qualidade é constante, caracterizando um estudo no espaço *Rate-Energy* (R-E).

A Figura 20 mostra a frente de Pareto envolvendo os 32 parâmetros, os quais são plotados em pontos azuis no espaço D-E. O parâmetro PS0 foi usado como referência tanto para o BD-PSNR quanto para a energia normalizada, e aparece na região superior direita do gráfico. Este parâmetro é o que apresenta melhor qualidade de imagem e maior consumo de energia. O parâmetro PS31 em contrapartida é o que apresenta menor consumo energético, mas também, é o que mais deteriora a imagem. A linha verde representa uma curva convexa superior, a qual passa pelos pontos (circulados em verde) que compõem a frente de Pareto no espaço D-E.

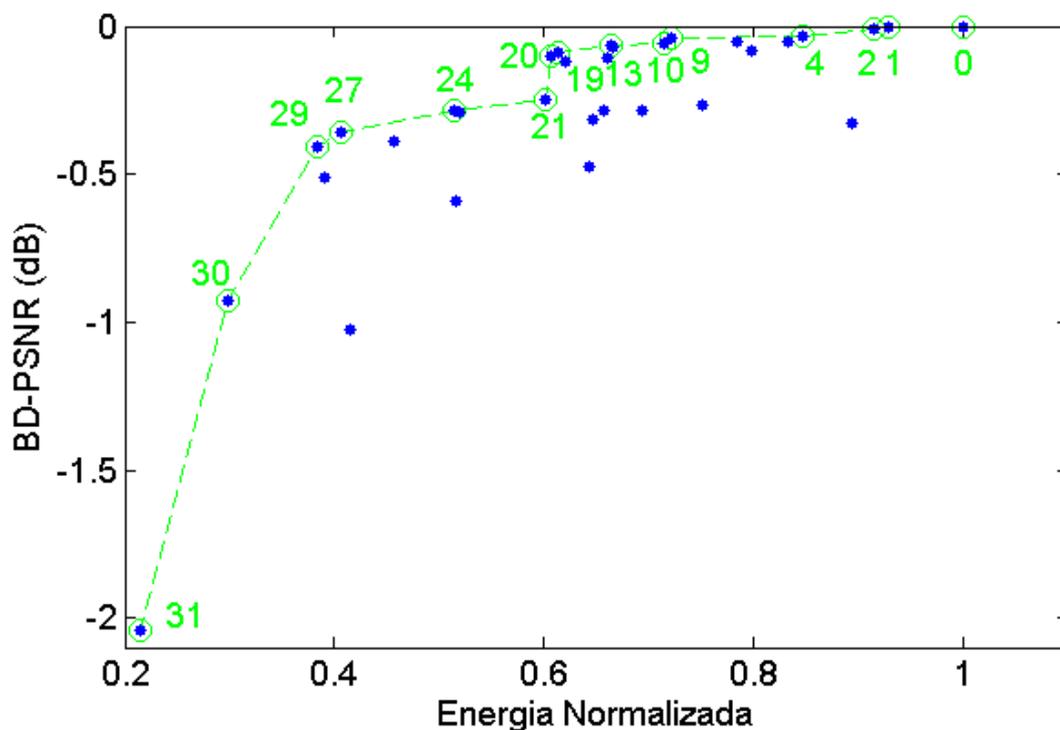


Figura 20 – Frente de Pareto no espaço D-E

A Figura 21 mostra uma análise similar a anterior, envolvendo os mesmos 32 parâmetros, igualmente plotadas em pontos azuis, mas, desta vez, no espaço R-E. Novamente, o parâmetro PS0 foi usado como referência tanto para o BD-BR quanto para a energia normalizada, porém, neste caso aparece na região inferior direita do gráfico. Este parâmetro é o que apresenta menor taxa de bits e maior consumo de energia. O parâmetro PS31 em contrapartida é o que apresenta menor consumo energético, mas também, é o que apresenta maior taxa de bits. A linha vermelha representa uma curva convexa inferior, a qual passa pelos pontos (circulados em vermelho) que compõem a frente de Pareto no espaço R-E.

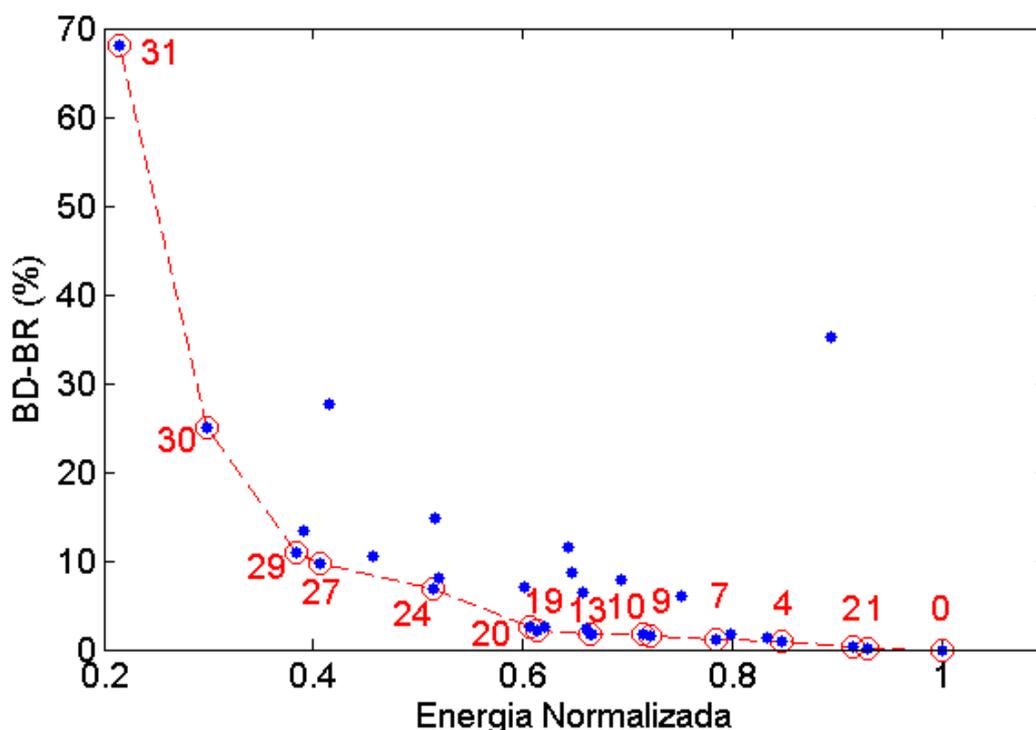


Figura 21 – Frente de Pareto no espaço R-E

Pode-se verificar que os pontos que compõem a Frente de Pareto em ambas as análises são praticamente os mesmos porque, de forma geral, BD-BR e BD-PSNR representam a mesma informação, uma vez que a única diferença entre ambos é o eixo de integração utilizado na determinação dos mesmos.

É possível perceber que conforme se percorre a frente de Pareto, incrementando a configuração na curva convexa, a energia normalizada diminui, bem como a eficiência da codificação (em termos de taxa de bits e qualidade). Isto pode ser utilizado para permitir com que a energia consumida pelo codificador possa ser controlada, num sistema que admita perdas na eficiência R-D, selecionando dinamicamente a configuração do codificador entre os pontos que pertençam à frente de Pareto.

Os PSs que pertencem à frente estão circulados em verde no gráfico D-E e em vermelho no gráfico R-E. Os PSs que geraram pontos pertencentes, simultaneamente, tanto à frente no espaço D-E quanto no espaço R-E, foram escolhidos para serem utilizados no desenvolvimento do controlador. No entanto, conforme será exposto a seguir, alguns PSs, embora pertençam à ambas frentes de Pareto, não foram utilizados no desenvolvimento do controlador em virtude de

possuírem valores de energia normalizada muito próximos, como, por exemplo, o caso dos PSs 9 (energia normalizada de 0,7206) e 10 (energia normalizada de 0,7136), nesta situação, apenas o PS9 foi incluído no controlador dinâmico de energia.

Com isso, os parâmetros PS0, PS1, PS2, PS4, PS9, PS10, PS13, PS19, PS20, PS24, PS27, PS29, PS30 e PS31 são os pontos comuns nas duas frentes de Pareto.

Na Tabela 7 são mostrados os 11 pontos ótimos de operação escolhidos, considerando o cenário desenvolvido, denominados Pontos de Controle (PC), os quais apresentam as configurações que pertencem às frentes de Pareto indicadas nas Figuras 20 e 21. A Tabela 7 também apresenta o *Parameter Set* que caracteriza cada PC, bem como as configurações que compõem o parâmetro (apresentadas na Tabela 2), a energia normalizada, valores de BD-BR e BD-PSNR. Na Tabela 8 são apresentados quais parâmetros de codificação são alterados em cada PC.

Tabela 7 – Pontos de Controle (PCs) definidos para o controlador

Pontos de Controle	Parameter Set	Configurações	Energia Medida (J)	Energia Normalizada	BD-BR (%)	BD-PSNR (dB)
PC0	PS0	c0	44037,83	1,0000	0,0000	0,0000
PC1	PS1	c8	40883,39	0,9284	0,0111	-0,0009
PC2	PS4	c1	37308,18	0,8472	0,8517	-0,0354
PC3	PS9	c3, c7	31732,59	0,7206	1,5075	-0,0381
PC4	PS13	c1, c3, c8	29214,46	0,6634	1,7288	-0,0636
PC5	PS19	c3, c10	27004,82	0,6132	2,1930	-0,0870
PC6	PS24	c1, c3, c6	22649,72	0,5143	6,8404	-0,2827
PC7	PS27	c2, c10	17907,39	0,4066	9,6874	-0,3565
PC8	PS29	c1, c2, c7, c10	16908,78	0,3840	11,0182	-0,4087
PC9	PS30	c2, c6, c10	13128,40	0,2981	25,0067	-0,9301
PC10	PS31	c4	9449,68	0,2146	68,1340	-2,0422

Tabela 8 – Configuração de codificação utilizada em cada PC utilizado no controlador

Parâmetro	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
AMP	1	1	0	1	0	1	0	1	0	1	1
FME	1	1	1	1	1	1	1	0	0	0	1
HadME	1	1	1	0	0	0	0	1	1	1	1
MaxCUd	4	4	4	4	4	4	3	4	4	3	1
SR	64	32	64	16	32	64	64	64	16	64	64
MAXTUd	3	3	3	3	3	1	3	1	1	1	3

5.2 Sistema de Controle de Energia Proposto

Nesta seção é apresentado o sistema de controle dinâmico de energia proposto por este trabalho, o qual atua ajustando a operação do codificador através de configurações que proporcionam menor consumo de energia.

A Figura 22 apresenta um diagrama detalhado do controlador dinâmico de energia proposto para o codificador HEVC. Todos os blocos do controlador serão explicados nessa seção utilizando a Figura 24 como referência.

O processo de codificação inicia com consumo de energia máximo e o primeiro GOP é codificado com o PC0 ($PC_i = 0$, onde i é índice do GOP atual). Os três primeiros GOPs são codificados dessa forma para obtenção do consumo de energia médio por GOP. Dessa forma, o consumo de energia do sistema sem o uso do controlador é determinado, permitindo o cálculo da energia desejada, a partir de informações fornecidas ao sistema pelo próprio usuário (valor percentual de economia desejada).

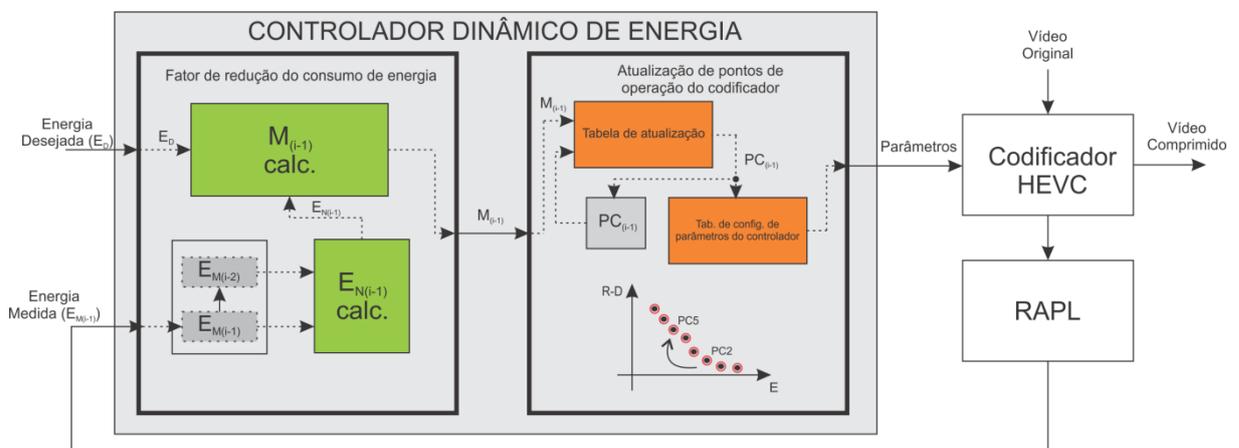


Figura 22 – Controlador dinâmico de energia do HEVC proposto

Se existir a necessidade de redução do consumo de energia (a energia medida é maior que a energia desejada) o controlador irá atuar e buscar a próxima configuração. Essa busca é realizada na estrutura presente na Tabela 9, a qual é denominada tabela de atualização do controlador. Esta tabela apresenta a variação no consumo médio de energia normalizada com a alteração entre os PCs.

Tabela 9 – Tabela de atualização do controlador

Config.		Anterior										
		PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
Nova	PC0	1	1,08	1,18	1,39	1,51	1,63	1,94	2,46	2,60	3,35	4,66
	PC1	0,93	1	1,09	1,29	1,40	1,51	1,80	2,28	2,41	3,11	4,33
	PC2	0,85	0,91	1	1,18	1,28	1,38	1,65	2,08	2,20	2,84	3,95
	PC3	0,72	0,78	0,85	1	1,09	1,17	1,40	1,77	1,87	2,41	3,36
	PC4	0,66	0,71	0,78	0,92	1	1,08	1,29	1,63	1,73	2,22	3,09
	PC5	0,61	0,66	0,72	0,85	0,92	1	1,19	1,51	1,59	2,06	2,86
	PC6	0,51	0,55	0,61	0,71	0,77	0,84	1	1,26	1,34	1,72	2,39
	PC7	0,41	0,44	0,48	0,56	0,61	0,66	0,79	1	1,06	1,36	1,89
	PC8	0,38	0,41	0,45	0,53	0,58	0,63	0,74	0,94	1	1,29	1,79
	PC9	0,30	0,32	0,35	0,41	0,45	0,49	0,58	0,73	0,78	1	1,39
	PC10	0,21	0,23	0,25	0,30	0,32	0,35	0,42	0,53	0,56	0,72	1

De acordo com o valor de consumo de energia desejado, a tabela de atualização será percorrida, definindo o índice da próxima configuração. Essa nova configuração será utilizada na codificação do próximo GOP, por exemplo, estavam sendo consumidos 10000 joules na configuração PC0, e é desejado um consumo de 7000 joules (economia de 30%), a nova configuração passa a ser então a configuração PC3. Não existe limitação na mudança de configuração, o controlador pode migrar de uma determinada configuração para qualquer outra. A energia consumida pelo próximo GOP também é medida, e a tabela de atualização é novamente percorrida e, dessa forma, sucessivamente, de modo a atingir o consumo de energia desejado. O bloco chamado de “Fator de redução do consumo de energia”, na Figura 22, é a etapa na qual o fator de redução é calculado, o que permitirá com que se procure uma nova configuração na tabela de atualização.

O ponto de operação a ser utilizado na codificação do próximo GOP é calculado no bloco denominado “Atualização de pontos de operação do codificador”, baseado no cálculo da relação $M_{(i-1)}$ entre o consumo de energia desejado (E_D) e o consumo de energia normalizado ($E_{N(i-1)}$), o qual é uma média ponderada entre o consumo de energia dos últimos dois GOPs. Em (11) é ilustrado o cálculo dessa relação. Em (12) é mostrada a equação que calcula o consumo de energia normalizado ($E_{N(i-1)}$), onde ($E_{M(i-1)}$) e ($E_{M(i-2)}$) são as energias consumidas pelos GOPs $i - 1$ e $i - 2$ (considerando que i é o índice do GOP atual).

$$M_{(i-1)} = \frac{E_D}{E_{N(i-1)}} \quad (11)$$

$$E_{N(i-1)} = \frac{2.E_{M(i-1)} + 1.E_{M(i-2)}}{3} \quad (12)$$

As caixas em linha tracejada, preenchidas com cinza escuro, correspondentes a $(E_{M(i-1)})$ e $(E_{M(i-2)})$ na Figura 22, são elementos de memória que gravam os consumos de energia dos dois GOPs mais recentes. Os pesos 2 e 1 aplicados a $(E_{M(i-1)})$ e $(E_{M(i-2)})$ no cálculo da média ponderada foram escolhidos por experimentação manual e baseados no que é considerado em Corrêa (2014), o qual avalia individualmente a complexidade de cada GOP para cada sequência, constando experimentalmente a relação entre a complexidade do GOP atual e dos temporalmente pregressos. Verificou-se, através da análise de sensibilidade demonstrada no Capítulo 4, que o valor de energia consumida por determinado GOP é diretamente ligado ao comportamento do consumo de energia dos GOPs temporalmente anteriores, com maior relevância do GOP imediatamente anterior. Isto vai ao encontro do que é realizado por Corrêa (2014) no desenvolvimento de seu controlador de complexidade, no qual é realizada uma análise de sensibilidade semelhante e se considera o maior peso para a complexidade do GOP temporalmente progresso mais próximo, no cálculo da média ponderada. Se o valor calculado de $M_{(i-1)}$ é maior que 1, a energia consumida é menor que a desejada, então mais recursos podem ser utilizados no processo de codificação do GOP atual, permitindo o aumento da energia consumida. Já se $M_{(i-1)}$ é menor que 1, o consumo de energia deve ser diminuído na codificação do GOP atual como forma de atingir o consumo de energia desejado.

Uma vez que $M_{(i-1)}$ é calculado a tabela de atualização de parâmetros é percorrida até que o valor mais próximo a $M_{(i-1)}$ seja encontrado, determinando qual configuração deve ser utilizada para ajustar o valor do consumo de energia. O índice correspondente a esse valor de energia normalizada será o índice da nova configuração. Por exemplo, se a configuração anterior é a PC2 e, uma economia de 40% é requerida, significa que a relação $M_{(i-1)}$ será 0,6, logo a configuração correspondente que fornece um valor mais próximo a 0,6 é a configuração PC6. Logo, o GOP atual será codificado baseado na configuração PC6. Na Tabela 9, cada

coluna representa a configuração anterior, de acordo com o valor de $M_{(i-1)}$, as linhas são percorridas uma a uma, determinando a configuração atual.

A Figura 23 apresenta o pseudocódigo para o algoritmo do controlador, implementado no HM para permitir o controle dinâmico de energia. Nas linhas 01 até 13 são codificados os três primeiros GOPs e as energias consumidas são armazenadas. A energia desejada é determinada por outro processo em paralelo. Após, são calculadas a energia normalizada $E_{N(i-1)}$ e a relação $M_{(i-1)}$. Estes cálculos permitem que um novo PC seja encontrado, determinando um novo ponto de operação. Com o novo ponto de operação determinado, o próximo GOP é codificado com a nova configuração. O algoritmo possui um laço de repetição, assim, o algoritmo refaz o processo para os GOPs seguintes, até que todos os GOPs do vídeo sejam codificados.

```

codificar(video, ED)
01  INICIO
02   $i \leftarrow 0, PC_{(i)} \leftarrow 0$ 
03  codificar GOP(i) com AMPPC(i), FMEPC(i), HadMEPC(i),
04      MaxCUDPC(i), SRPC(i), MaxTUDPC(i)
05  EM(i-1)  $\leftarrow$  energia medida na codificacao do GOP(i)
06   $i \leftarrow i + 1$ 
07  codificar GOP(i) com AMPPC(i), FMEPC(i), HadMEPC(i),
08      MaxCUDPC(i), SRPC(i), MaxTUDPC(i)
09  EM(i-2)  $\leftarrow$  EM(i-1)
10  EM(i-1)  $\leftarrow$  energia medida na codificacao do GOP(i)
11   $i \leftarrow i + 1$ 
12  codificar GOP(i) com AMPPC(i), FMEPC(i), HadMEPC(i),
13      MaxCUDPC(i), SRPC(i), MaxTUDPC(i)
14  enquanto  $i <$  numero de GOPs faça
15      EM(i-2)  $\leftarrow$  EM(i-1)
16      EM(i-1)  $\leftarrow$  energia medida na codificacao do GOP(i)
17      EN(i-1) =  $(2 \cdot E_{M(i-1)} + 1 \cdot E_{M(i-2)}) / 3$ 
18      M(i-1) = ED / EN(i-1)
19      PC(i)  $\leftarrow$  receber a configuração PC indexada por M(i-1)
20       $i \leftarrow i + 1$ 
21      codificar GOP(i) com AMPPC(i), FMEPC(i), HadMEPC(i),
22          MaxCUDPC(i), SRPC(i), MaxTUDPC(i)
23  fim enquanto

```

Figura 23 – Pseudocódigo do controlador dinâmico de energia

6 RESULTADOS E DISCUSSÕES

Neste capítulo serão apresentados os resultados obtidos com a utilização do controlador dinâmico de energia do HEVC desenvolvido. Foram realizadas simulações na mesma plataforma utilizada para a análise de sensibilidade, para as cinco sequências anteriormente mencionadas, nos QPs 22, 27, 32 e 37. Todos os vídeos foram utilizados para avaliar o controlador com *set point* fixado para fornecer 30% de economia de energia, simulações com mudança de *set point* e, também, simulações com um controlador de energia baseado na variação dos QPs, para permitir uma análise da eficiência do controlador dinâmico em comparação com outro tipo de controlador. As medições de energia consideram o *overhead* do sistema e a energia consumida pelo próprio controlador.

Na Tabela 10 são apresentados os valores médios de consumo para as simulações sem o uso do controlador, com controlador e *set point* fixo em 30% do consumo original, o valor do *set point* para cada vídeo e QP, o valor do erro médio quadrático (*Mean Squared Error* – MSE), a economia de energia proporcionada e os valores de BD-BR e BD-PSNR.

O controlador dinâmico é capaz de fornecer uma economia de energia média de 25,6%, com um BD-BR médio de 5,09% e BD-PSNR médio de -0,188dB. O erro médio quadrático é utilizado como forma de avaliação dos resultados obtidos com o uso do controlador. O MSE médio equivale a 77,79 J e representa um valor consideravelmente baixo comparado aos valores médios de energia consumida, o que significa que os valores desejados estão muito próximos ao SP.

Note que o desvio padrão do BD-BR é igual a 3,2%, o que representa um valor significativo e implica em grande variação. Isto acontece devido à natureza distinta das sequências utilizadas e, também, às diferentes resoluções. Determinadas decisões do controlador, como a alteração de determinado parâmetro, podem contribuir para isso, uma vez que a alteração de um parâmetro de codificação pode afetar sequências distintas em maior ou menor grau, dependendo de sua função.

Pode-se perceber que, em termos de valores médios, o valor de consumo real fica muito próximo dos valores desejados e o erro médio quadrático apresenta valores baixos. A economia de energia desejada ficou próxima ao valor alvo, sendo que as diferenças apresentadas acontecem principalmente em virtude das decisões

baseadas em dados coletados *off-line*, do tempo de convergência e oscilações do controlador a cada GOP.

Tabela 10 – Resultados do controlador para SP configurado em 30%

Sequência	QP	Consumo Médio S/Controlador (J)	Consumo Médio C/Controlador (J)	Set Point (J)	MSE (J)	Economia de energia (%)	BD-BR (%)	BD-PSNR (dB)
<i>People on Street</i>	22	9659,70	6897,70	6740,10	215,75	27,66	6,09	-0,2593
	27	8039,08	6035,70	5689,28	254,40	23,91		
	32	7164,03	5546,44	5059,20	273,16	20,37		
	37	6578,09	4899,80	4675,12	247,91	25,19		
<i>Kimono</i>	22	4001,16	3080,59	3077,80	84,05	29,91	1,62	-0,0483
	27	3424,56	2805,49	2650,23	97,33	24,14		
	32	3018,57	2360,96	2280,53	93,16	26,47		
	37	2733,83	2096,02	2061,32	83,26	28,32		
<i>China Speed</i>	22	1675,36	1203,93	1142,51	28,87	24,62	0,58	-0,0292
	27	1467,38	1033,65	992,21	28,93	25,82		
	32	1277,00	924,55	876,96	34,90	24,57		
	37	1124,70	818,58	766,98	35,90	23,27		
<i>Blowing Bubbles</i>	22	177,34	147,12	143,88	4,25	27,75	10,11	-0,3528
	27	137,69	102,51	94,71	3,18	21,76		
	32	114,90	85,22	79,61	3,16	22,94		
	37	99,64	73,74	69,72	3,16	24,24		
<i>Basketball Drill</i>	22	759,88	567,42	552,58	16,36	27,32	7,06	-0,2506
	27	648,84	486,24	451,65	18,91	22,34		
	32	565,32	382,66	394,03	12,43	32,89		
	37	510,44	356,15	350,84	16,82	28,49		
Valores médios		2658,88	1995,22	1907,46	77,79	25,60	5,09	-0,188

6.1 Comportamento do controlador para cada sequência

As Figuras 24, 25, 26, 27 e 28 apresentam, respectivamente, os resultados do controlador para um *set point* fixo em 30%, para as sequências *People on Street* (QPs 22 e 27), *Basketball Drill* (QP 22), *Kimono* (QP 22) e *China Speed* (QP 27). Nas figuras, são apresentados os valores de consumo energético com atuação do controlador (controlada), média do consumo com controlador (controlada média), o *set point*, o consumo de energia sem o controlador (original) e, também, o consumo médio de energia sem o controlador (original média).

Nesses exemplos é informada a energia média consumida por GOP, a qual é representada por uma linha de tendência, uma vez que os dados são discretos.

Valores de consumo GOP a GOP são representados por linhas contínuas, enquanto que valores médios estão representados por linhas tracejadas ou pontilhadas.

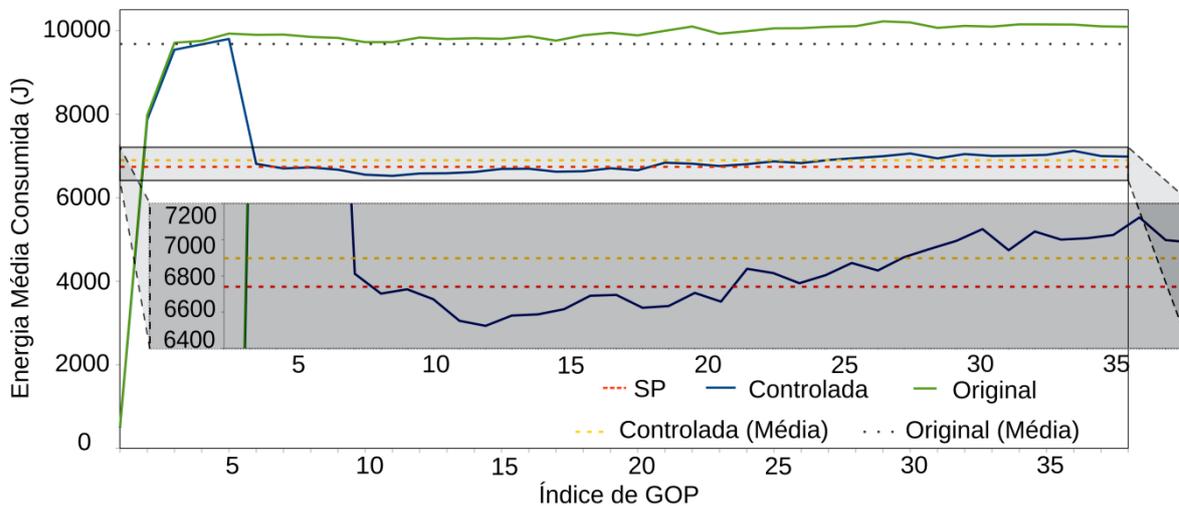


Figura 24 – Resultado do controlador para a sequência *PeopleonStreet* (QP 22)

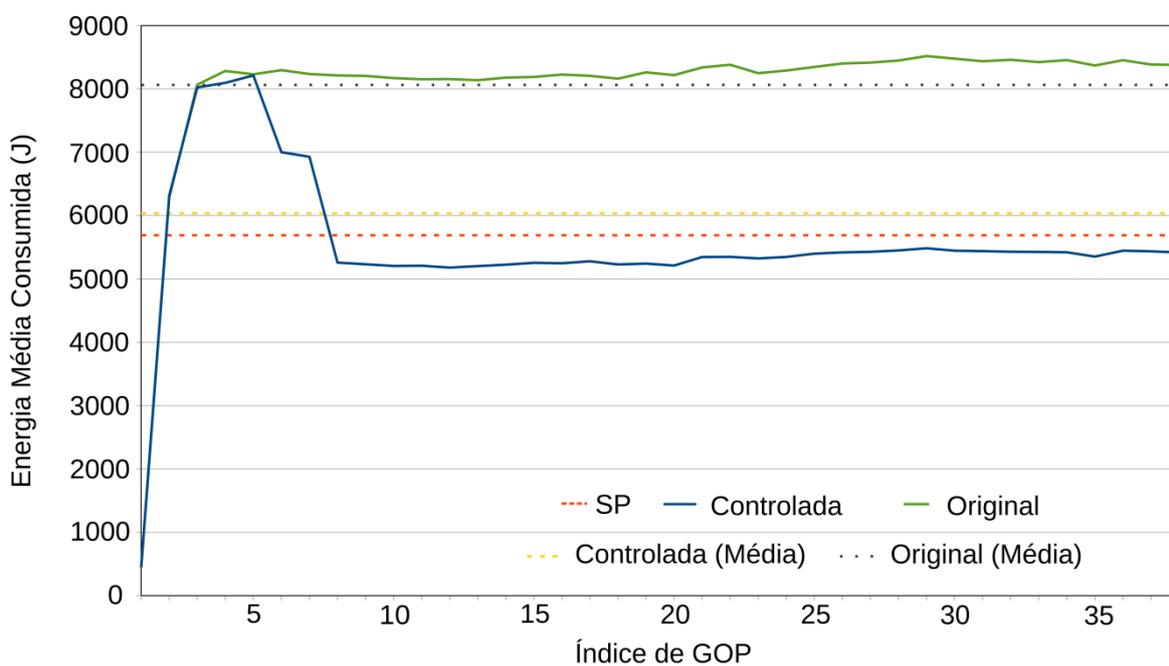


Figura 25 – Resultado do controlador para a sequência *PeopleonStreet* (QP 27)

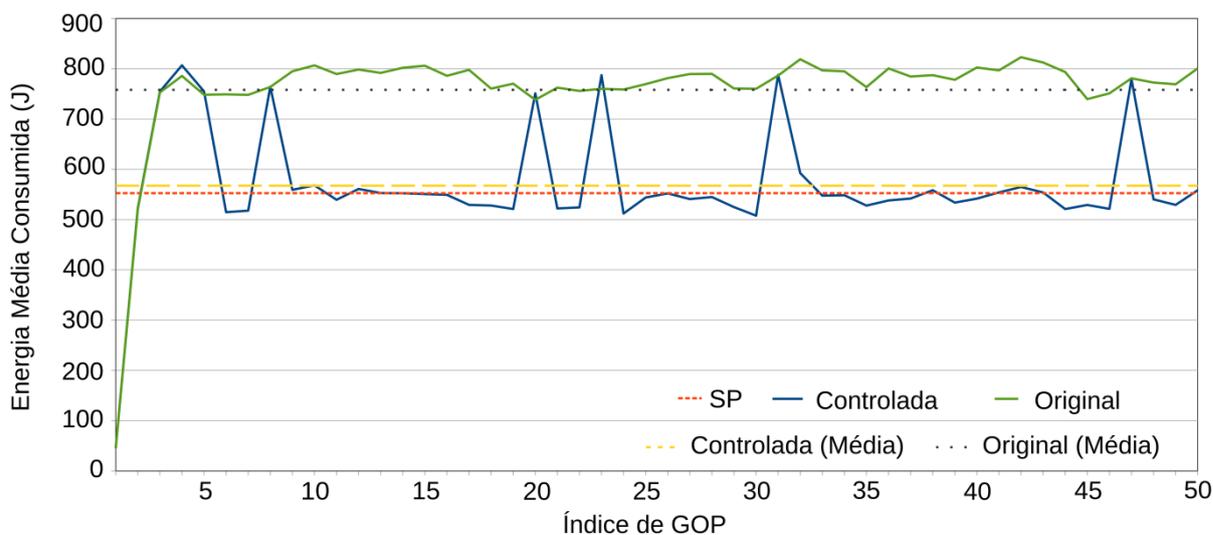


Figura 26 – Resultado do controlador para a sequência *BasketballDrill* (QP22)

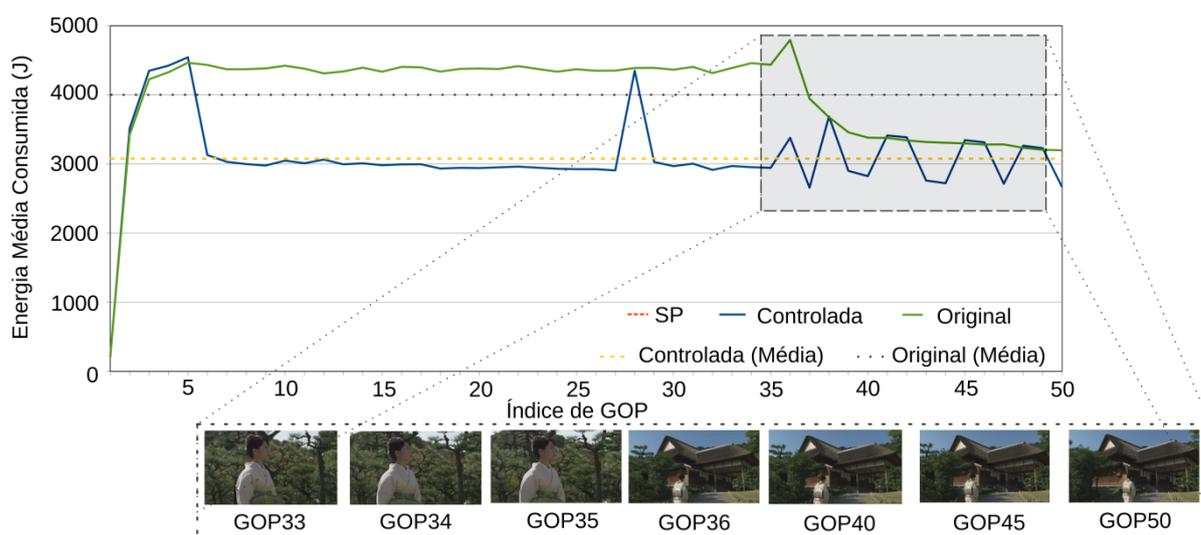


Figura 27 – Resultado do controlador para a sequência *Kimono* (QP 22)

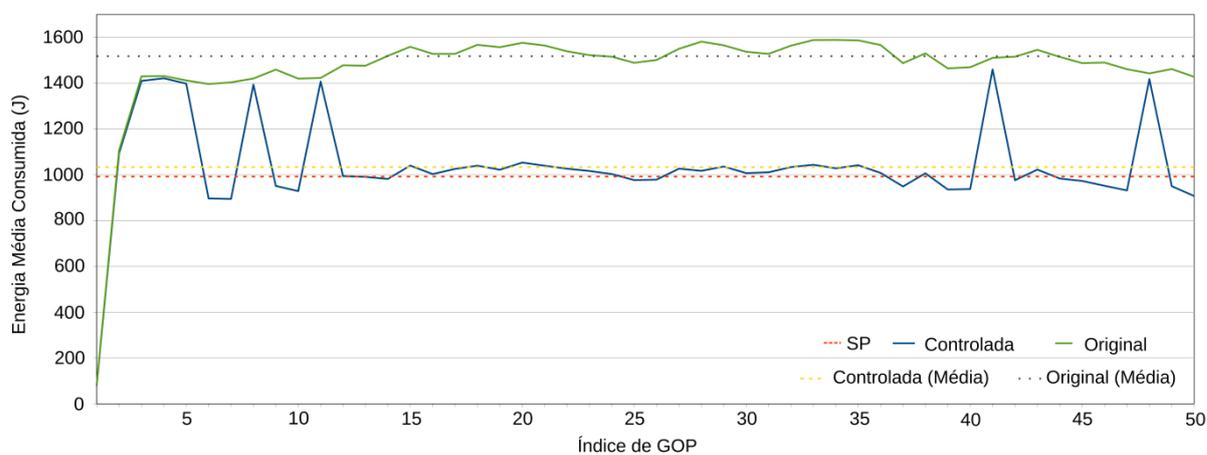


Figura 28 – Resultado do controlador para a sequência *ChinaSpeed* (QP 27)

Pode-se perceber que, nos primeiros GOPs, o controlador busca o *set point*, oscilando em torno deste ponto com grande amplitude (observe o detalhamento na ampliação presente na Figura 24) e, após determinado tempo, o consumo com atuação do controle passa a oscilar em torno do valor desejado com baixa amplitude, apresentando pequeno erro, o qual para as sequências testadas, e para o *set point* adotado equivale, a 4,69%, com BD-BR médio de 5,09% e BD-PSNR médio de -0,188dB.

Em vídeos de maior resolução, como *PeopleonStreet* e *Kimono*, o comportamento do processo de controle é mais estável, em virtude justamente do tamanho do vídeo, pois nesses casos a quantidade de informação codificada é maior e, portanto, o impacto sobre os resíduos acaba sendo menos significativo. Outra característica que cabe destacar é a quantidade de movimento presente nos vídeos, quer seja pela variação na posição dos componentes, quer seja na mudança de cenários e texturas. A sequência *BasketballDrill*, cujo comportamento de consumo energético é representado na Figura 26, representa um jogo de basquetebol, no qual há intensa movimentação da bola e dos jogadores, dessa forma, o perfil de consumo de energia é caracterizado por picos de consumo, justamente nos momentos de mais intensa movimentação. Já na sequência *Kimono*, cujo consumo energético é representado na Figura 27, o consumo é linear nos GOPs iniciais, mas note que, a partir do GOP 35, há uma mudança no cenário (veja na Figura 27 a representação de alguns quadros da sequência), o qual parte de um cenário altamente texturizado (vegetação) para um cenário com uma casa e parte de um céu azul no fundo. O que acontece é um decaimento no consumo de energia justamente porque o plano de fundo varia muito pouco, dessa forma há gasto menor de energia na etapa de estimação de movimento.

Na Figura 29 é apresentado o comportamento do controlador para a sequência *BlowingBubbles* (QP37), evidenciando maior instabilidade no controle, embora o valor médio final do consumo de energia fique próximo do *set point* (vide Tabela 10). Nessa análise, são amostrados alguns quadros do vídeo (primeiro frame de cada GOP), para que seja possível realizar uma correlação entre o consumo energético do codificador e a cena que se passa em determinado instante. Como a sequência é de baixa resolução, o consumo de energia é menor se comparado a outras sequências de maior resolução. Isso faz com que pequenas variações no consumo de energia tenham uma representatividade maior no perfil de consumo

energético. Observe que uma característica marcante é a oscilação do consumo, o que se deve ao fato do ambiente ser tomado pelas bolhas de sabão, quando há a passagem da bolha em frente às câmeras há mudança na textura da cena. Perceba ainda que a iluminação do ambiente também se altera no GOP 15 e no GOP 35, refletindo diretamente no consumo de energia do vídeo. Para maior detalhamento vide o Anexo A.

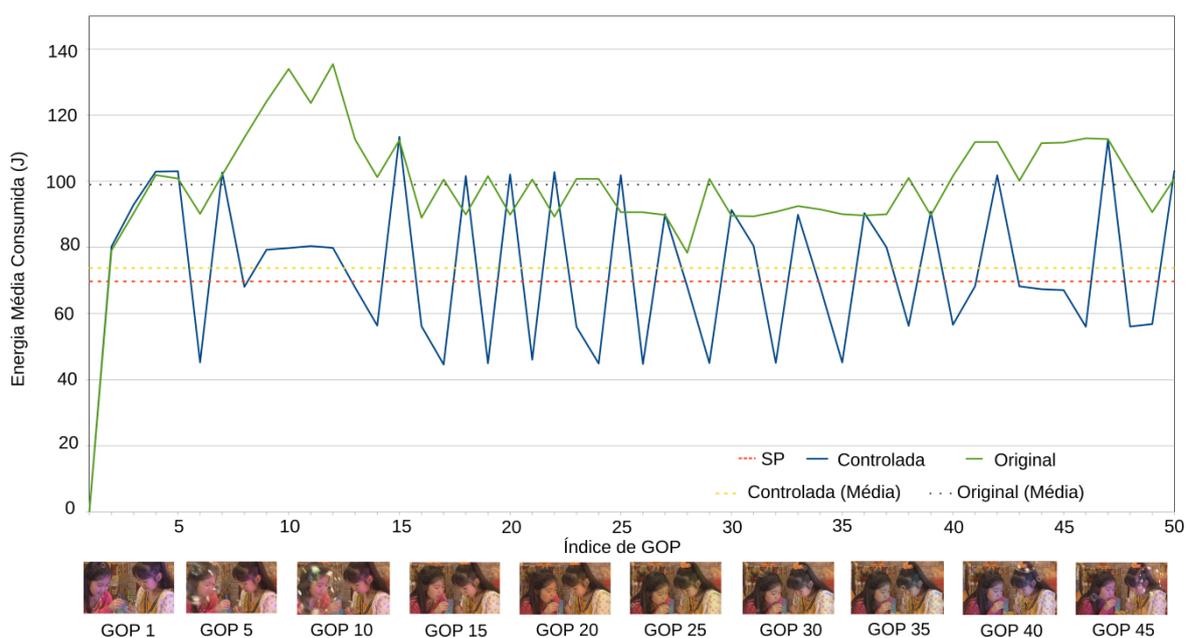


Figura 29 – Resultado do controlador para a sequência *BlowingBubbles* (QP37)

Durante a codificação de uma sequência, cada GOP processado possui certa quantidade de bits para ser representado e determinada qualidade objetiva (PSNR). Para aprofundar a discussão dos resultados obtidos, na Figura 30 são apresentados os resultados do número de bits utilizados por GOP, para a sequência de vídeo *PeopleonStreet*, com QP fixado em 22, com e sem atuação do controlador (vide Figura 24). Na Figura 31 são apresentados os resultados de PSNR por GOP, para a mesma sequência, com e sem atuação do controlador.

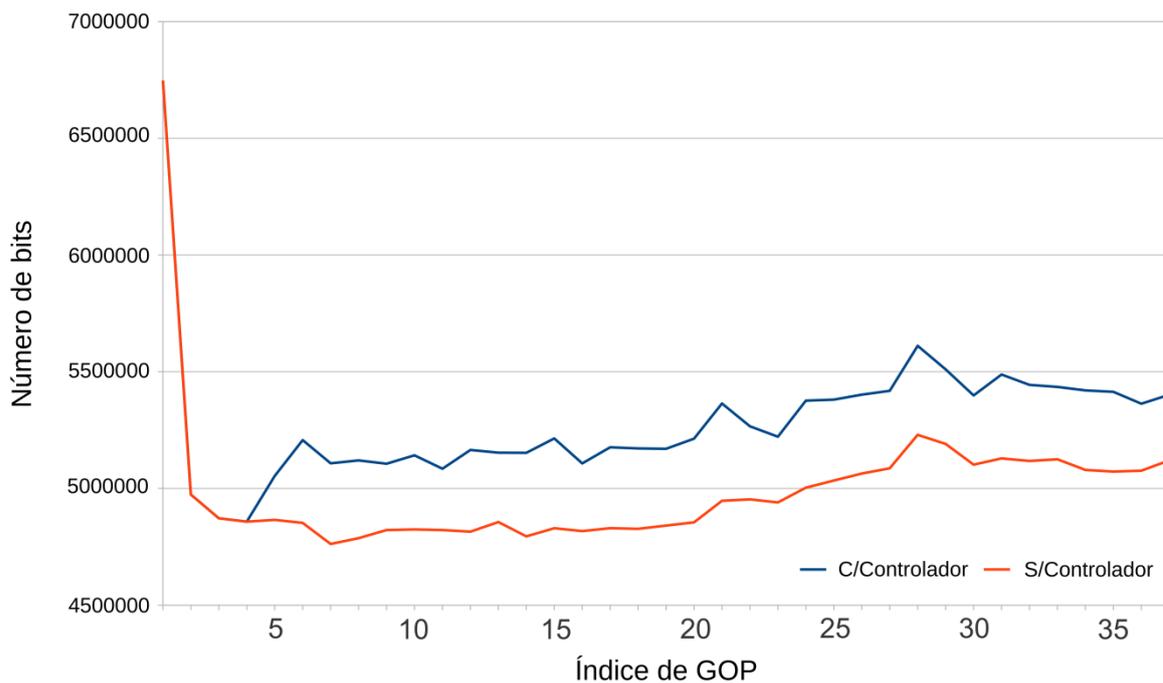


Figura 30 – Número de bits por quadro para a sequência *PeopleonStreet* (QP 22)

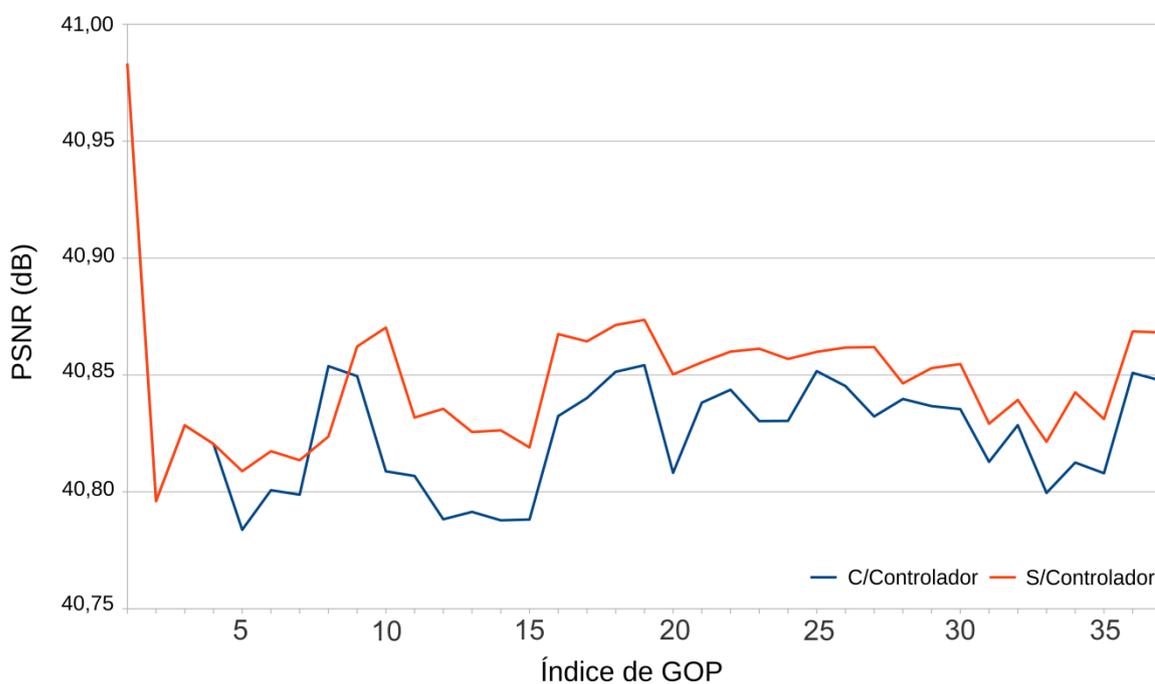


Figura 31 – PSNR por quadro para a sequência *PeopleonStreet* (QP 22)

Pode-se perceber que nos primeiros quatro GOPs não há acréscimo na quantidade de bits, entretanto, após esse ponto, há um acréscimo significativo. Dessa forma, o aumento na quantidade de bits faz com que o valor de BD-BR

também aumente. Além disso, também se pode perceber que a variação de PSNR é mínima, o que acarreta em menores perdas em BD-PSNR.

6.2 Resultados para adaptação à mudança de SP

Foi realizada também uma análise na qual é avaliada a controlabilidade e adaptabilidade do controlador, considerando a variação, em tempo de codificação, do SP. Esta análise é importante, pois, em aplicações reais, o cenário de energia disponível pode variar. As Figuras 32 e 33 apresentam o comportamento do controlador para uma mudança no *set point* de 30% para 15%, para as sequências *PeopleonStreet* e *BasketballDrill*, respectivamente.

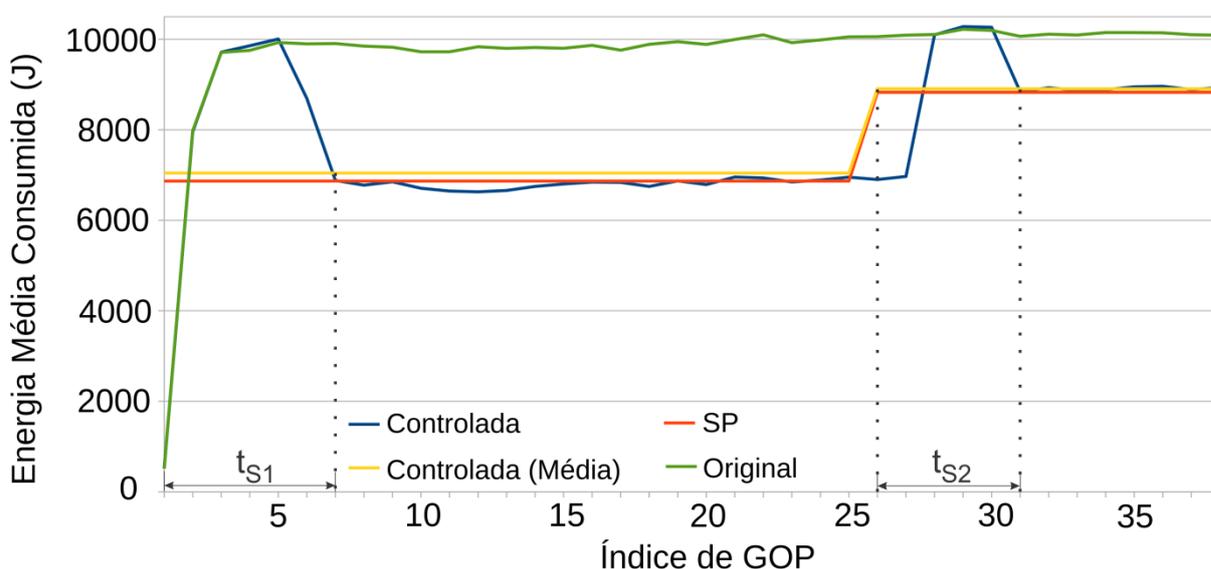


Figura 32 – Mudança no SP para a sequência *PeopleonStreet* (QP22)

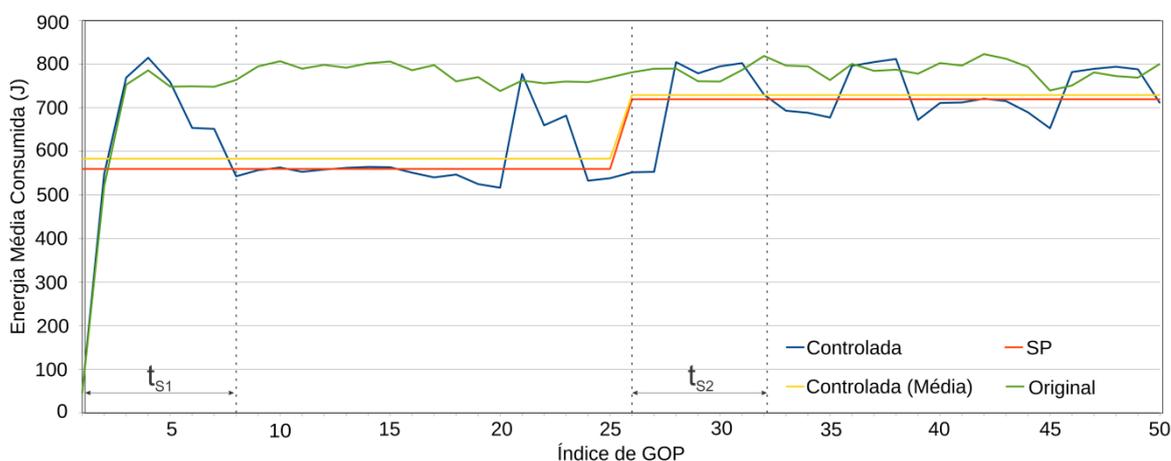


Figura 33 – Mudança no SP para a sequência *BasketballDrill* (QP22)

A análise realizada mostra a habilidade do controlador de energia em adaptar dinamicamente o consumo de energia a um novo valor de *set point*. A reação do controlador de energia apresenta um atraso, após a mudança do SP, devido à dependência dos dois últimos GOPs. Nas Figuras 32 e 33 também são apresentados os tempos de estabilização (tempo transcorrido até atingir o estado permanente) para o primeiro (t_{s1}) e segundo (t_{s2}) SP. Pode-se perceber, na Figura 32, que $t_{s1} = 6$ GOPs e $t_{s2} = 5$ GOPs, e que, na Figura 33, que $t_{s1} = 7$ GOPs e $t_{s2} = 5$ GOPs. O controlador é mais rápido para atingir o *set point* 2 porque nesse caso o mesmo já está ligado, enquanto que no começo da codificação o controle só atua após transcorridos três GOPs.

Em ambos os cenários acontece um aumento no *set point*, por exemplo, um dispositivo tem um aumento na disponibilidade de carga da bateria. O controlador consegue se adaptar à nova situação e aumentar o consumo de energia do codificador, oscilando em torno do novo consumo desejado, melhorando a eficiência de codificação. Os tempos de estabilização médios para as sequências testadas podem ser observados na Tabela 11.

Tabela 11 – Tempos de estabilização médios do controlador

Sequência	QP	ts (#GOP)	ts _{médio} (#GOP)
<i>People on Street</i>	22	6	6,5
	27	7	
	32	6	
	37	7	
<i>Kimono</i>	22	6	6
	27	6	
	32	6	
	37	6	
<i>China Speed</i>	22	6	6,75
	27	7	
	32	7	
	37	7	
<i>Blowing Bubbles</i>	22	6	6,75
	27	7	
	32	7	
	37	7	
<i>Basketball Drill</i>	22	7	6,75
	27	6	
	32	7	
	37	7	
t _s controlador		6,55	

6.3 Comparação com outros controladores

Infelizmente, até o momento não foram encontrados na literatura trabalhos que realizem controle do consumo de energia aplicado ao codificador HEVC. Para que possam ser realizadas comparações a fim de validar os resultados do presente trabalho foi implementado um controlador de energia mais simples, baseado apenas na alteração do QP, para servir como base de comparação. O sistema de controle funciona da seguinte maneira: inicialmente o vídeo começa sendo codificado no QP 22 (menor entre os recomendados nas CCTs) e, de acordo com o *set point* desejado, o sistema aumenta o valor do QP (uma unidade a cada GOP), até que o consumo de energia atinja o SP com erro mínimo. Na Tabela 12 são apresentados dados comparativos entre os dois controladores implementados, com *set point* fixo em 30%.

Tabela 12 – Comparação entre controladores com *set point* fixo em 30%

Sequência	Consumo de Energia S/Controlador (J)	Set Point (J)	Controlador de QP				Controlador Dinâmico			
			Consumo de Energia (J)	Economia de Energia (%)	Bit-rate (kbps)	PSNR (dB)	Consumo de Energia (J)	Economia de Energia (%)	Bit-rate (kbps)	PSNR (dB)
<i>People on Street</i>	9659,70	6740,10	7532,87	18,24	12838,37	35,41	6897,70	27,66	39753,18	41,65
<i>Kimono</i>	4001,16	3077,80	3296,81	22,88	2070,53	38,42	3080,59	29,91	5605,22	42,34
<i>China Speed</i>	1675,36	1142,51	1248,26	20,74	1428,17	35,70	1203,93	24,62	5113,77	43,91
<i>Blowing Bubbles</i>	177,34	143,88	134,08	36,81	472,03	34,27	147,12	27,75	1373,17	39,02
<i>Basketball Drill</i>	759,88	552,58	602,22	21,02	1100,97	36,25	567,42	27,32	3671,03	40,99
<i>Valores Médios</i>	3254,69	2331,37	2562,85	23,94	3582,01	36,01	2379,35	27,45	11103,27	41,58

Em média, o controlador dinâmico de energia proposto atinge valores de economia de energia superiores aos apresentados pelo controlador de QP, com menor erro percentual. Apesar de atingir 23,94% de economia de energia, o controlador de QP apresenta valores de PSNR inferiores aos apresentados pelo controlador dinâmico de energia, embora os valores de *bit-rate* também sejam menores. O controlador de QP deteriora a imagem em mais de 5dB, o que pode ser considerada uma perda extremamente elevada, a diminuição do *bit-rate* não compensa essa perda. Isto acontece porque o controlador, ao alterar apenas o QP, altera também, ao longo da codificação, os valores dos resíduos. Assim, conforme

aumenta o QP (a fim de atingir maior economia de energia), mais informação é descartada, logo, a taxa de bits reduz, reduzindo também a qualidade do vídeo.

Pode-se perceber que diante deste modelo mais simples de controlador de energia, o controlador dinâmico proposto neste trabalho apresenta melhor desempenho sob todos os aspectos: maior economia de energia, menor erro em relação ao *set point* e maior eficiência de codificação. Além disso, conforme exposto na Figura 34, o tempo de estabilização do controlador de QP é consideravelmente maior ($t_s = 15$ GOPs) do que o tempo de estabilização do controlador dinâmico, para a mesma sequência. A Tabela 13 apresenta os tempos de estabilização médios para as sequências testadas com o controlador de QP. Pode-se perceber que, comparado com o controlador de QP, o controlador dinâmico apresenta tempo de estabilização muito menor, 6,55 GOPs, quando comparado com o tempo de estabilização médio de 16,8 GOPs do controlador de QP. Embora atinja o SP, o controlador de QP não é indicado para trabalhar em aplicações reais, uma vez que não atinge os requisitos dinâmicos de operação por possuir tempo de estabilização muito elevado.

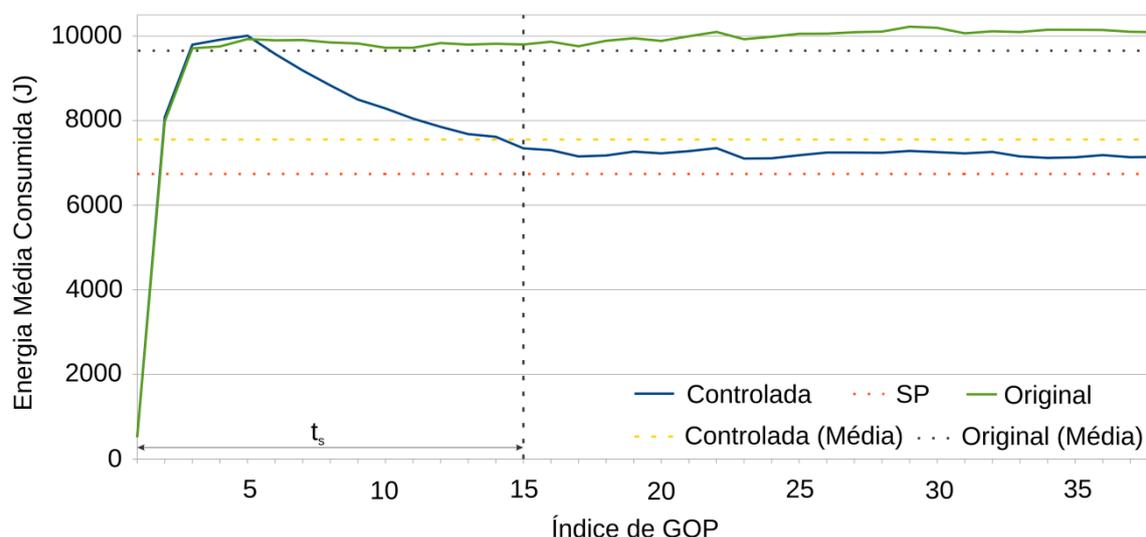


Figura 34 – Controlador de QP para a sequência *PeopleonStreet* para SP de 30%

Tabela 13 – Tempo de estabilização médio para o controlador de QP fixo em 30%

Sequência	ts (#GOP)	ts_{dinâmico} (#GOP)
<i>PeopleonStreet</i>	15	6,5
<i>Kimono</i>	17	6
<i>ChinaSpeed</i>	20	6,75
<i>BlowingBubbles</i>	14	6,75
<i>BasketballDrill</i>	18	6,75
<i>Valores Médios</i>	16,8	6,55

7 CONCLUSÃO E TRABALHOS FUTUROS

Esta dissertação apresentou o desenvolvimento de um controlador dinâmico de energia, projetado com base na eficiência de Pareto, a partir de uma análise de sensibilidade do consumo de energia do codificador HEVC com relação à variação de parâmetros de codificação. Todo este estudo envolveu inúmeras simulações e experimentos voltados para as análises desejadas. Este trabalho é o primeiro na literatura a propor um controlador de energia para o codificador HEVC.

A utilização do *software* de medição de energia RAPL em conjunto com o codificador HEVC é uma inovação. Em outros trabalhos encontrados na área, que atuam de forma mais similar, o RAPL é utilizado para medir a energia consumida pelo HM em uma análise *off-line*. O presente trabalho inova ao adaptar o RAPL dentro do código do codificador, tornando o funcionamento de ambos os *softwares* uma atividade paralela e conjunta.

Muitos trabalhos existentes na literatura desenvolveram controladores para o codificador HEVC, no entanto, propõem apenas sistemas de controle de complexidade para o codec HEVC. O que acontece, é que muitas vezes, a complexidade de um sistema nem sempre está associada de maneira linear com seu consumo de energia, isto se torna evidente principalmente em sistemas heterogêneos, nos quais um estudo do consumo de energia é fundamental, não bastando um estudo apenas da complexidade. No entanto, análises realizadas por estes trabalhos serviram como motivação, sendo desenvolvido neste trabalho um controlador dinâmico de energia.

O controlador de energia desenvolvido neste trabalho parte de uma análise *off-line* exaustiva, na qual são analisados os parâmetros de codificação que mais impactam no consumo de energia. A análise foi não determinística, ou seja, envolveu a manipulação de dados estatísticos; ferramentas de análise estatística, como o cálculo do desvio padrão, foram utilizadas para tornar a análise de dados o mais confiável possível.

A manipulação destes dados *off-line* envolveu, além da análise de consumo de energia, a análise de custo R-D de cada configuração. Com base nessa análise de sensibilidade foram desenvolvidas diversas configurações, as quais eram formadas por diferentes combinações de parâmetros de codificação. Para manipular estes dados, outra inovação foi justamente a determinação de pontos ótimos de

operação a partir destes conjuntos. Os pontos ótimos de operação foram determinados a partir da utilização de frentes de Pareto, recurso amplamente utilizado em problemas que envolvem otimização.

A partir da obtenção dos pontos ótimos de operação foi possível o desenvolvimento do controlador dinâmico de energia. O controlador dinâmico de energia funciona em nível de GOP, baseado num esquema que efetua o cálculo do fator de redução de consumo de energia, o qual é calculado com base em dados de consumo de energia de GOPs temporalmente anteriores. Após o cálculo deste fator, uma tabela de atualização de pontos de controle é percorrida, nesta tabela cada linha e coluna correspondem à determinada configuração pontos de controle, e um novo ponto de controle pode ser determinado.

Dessa forma, o trabalho inova ao introduzir uma metodologia completamente nova para o desenvolvimento de um controlador dinâmico de energia. A metodologia empregada pode ser facilmente adotada para estudos em qualquer plataforma que seja objetivo de estudo e que envolva a codificação de vídeo.

Foram analisados os resultados obtidos com o uso de controlador e seus efeitos amplamente discutidos. Para um *set point* fixo em 30% foi obtido erro médio de 4,69%, aumento médio em BD-BR de 5,09% e perda média em BD-PSNR de -0,188 dB, com tempo médio de convergência de 6,55 GOPs. Além disso, foi analisado também o comportamento dinâmico do controlador, quando ocorre mudança de SP, situação na qual o controlador apresentou comportamento satisfatório. Uma análise comparativa também foi realizada, entre o controlador dinâmico de energia proposto e um controlador de energia mais simples, baseado apenas na variação do QP. Verificou-se que o controlador dinâmico proposto apresentou melhor desempenho sob todos os aspectos, desde economia de energia proporcionada, erro em relação ao SP, eficiência de codificação e, também, tempo de estabilização do controlador, quando comparado com o controlador de QP.

7.1 Trabalhos Futuros

O estudo efetuado por este trabalho trás uma solução completa, apresentando um controlador inovador para o codificador HEVC. Apesar disso, vários experimentos e otimizações ainda poder ser realizados, dessa forma, o leque de possibilidades para trabalhos futuros é bastante grande. Primeiramente, tem-se o

fato deste trabalho ter sido executado em um sistema computacional de propósito geral, em virtude de recursos estruturais disponíveis para pesquisa. No período final do desenvolvimento deste trabalho foi adquirida pelo grupo de pesquisa uma placa ODROID-XU3 (ODROID, 2015). Esta placa representa um sistema embarcado, no qual toda análise efetuada neste trabalho pode ser adaptada. Isto proporcionaria uma análise mais próxima ao que existe hoje em termos de plataforma de *hardware* para dispositivos móveis, o que traria resultados ainda mais precisos quanto a aplicação do controlador. Além disso, mais parâmetros de codificação podem ser considerados, a análise de sensibilidade pode ser ainda maior, envolvendo maiores conjuntos de configurações.

O projeto do controlador pode ainda ser expandido, considerando ampliação na análise geral, uma vez que pode-se associar energia e complexidade num mesmo estudo.

REFERÊNCIAS

AFONSO, Vladimir. **Desenvolvimento de uma arquitetura para estimação de movimento fracionária segundo o padrão emergente HEVC**. Dissertação de Mestrado. Mestrado em Ciência da Computação, Universidade Federal de Pelotas, 2013.

AGOSTINI, Luciano Volcan. **Desenvolvimento de Arquiteturas de Alto Desempenho Dedicadas à Compressão de Vídeo Segundo o Padrão H.264/AVC**. Tese de Doutorado. Doutorado em Computação, Universidade Federal do Rio Grande do Sul. Porto Alegre, 2007.

BJØNTEGAARD, G. **Improvements of the BD-PSNR model**. ITU-T. Berlim, 2008.

BOSSEN, F. **Common Test Conditions and Software Reference Configurations (JCTVC-L1100)**, 2013.

BOYCE, J. **HM16: High Efficiency Video Coding Test Model (HM16) Encoder Discription**. JCTVC-R1002, Sapporo, Japão, 2014

CARRO, L.; WAGNER, F. R.; **Sistemas computacionais embarcados**. Jornadas de Atualização em Informática da SBC. 2003.

CHADHA, R.; Bhasker J.. **An ASIC Low Power Primer: Analysis, Techniques and Specification**. New York: Springer, 2013.

CHOI, Hyomin et. al. **Pixel-Wise Unified Rate-Quantization Model for Multi-Level Rate Control**. IEEE Journal of Selected Topics in Signal Processing, vol. 7, no.6, pp. 1112-1123, 2013.

CISCO. **Cisco Visual Networking Index: Forecast and Methodology**. CISCO, 2015. Disponível em: < http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html >. Acesso em: 10 nov. 2015.

CORRÊA, Guilherme Ribeiro et. al. **Complexity-Aware High Efficiency Video Coding**. New York: Springer, 2016.

CORRÊA, Guilherme Ribeiro. **Computational Complexity Reduction and Scaling for High Efficiency Video Encoders**. Tese de doutorado. Doutorado em Engenharia Eletrotécnica e de Computadores. Universidade de Coimbra. Coimbra, 2014.

CORRÊA, Marcel Moscarelli. **Desenvolvimento de Arquiteturas para interpolação de subamostras de luminância e crominância do padrão de codificação de vídeo emergente HEVC**. Monografia de Conclusão de Curso. Graduação em Ciência da Computação, Universidade Federal de Pelotas. Pelotas, 2013.

EBRAHIMI, Touradj. **A quick overview of HEVC - current status** - . Multimedia Signal Processing Group, Swiss Federal Institute of Technology. Disponível em : < https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=0ahUKEwipndTmxpbKAhUCipAKHchwCPQQFggTMAI&url=http%3A%2F%2Fvideosense.eu%2Findex.php%3Foption%3Dcom_phocadownload%26view%3Dcategory%26download%3D15%3Ahevc4videosense%26id%3D11%3Ateaching-material%26Itemid%3D58&usg=AFQjCNHLAwGU2giaiANcH095uZGvH56lbg&sig2=Rg1CU6_YHhIK6-3e_i3gwg&bvm=bv.110151844,d.Y2I >. Acesso em: 15 jan. 2016.

FISCHER, Leonardo M. **Avaliação de Consumo Energético para Plataformas Móveis**. Trabalho Acadêmico (Mestrado). Mestrado em Ciência da Computação, Universidade Federal de Pelotas. Pelotas, 2012.

GEM5. **Running gem5**. Disponível em: <http://www.gem5.org/Running_gem5#Experimenting_with_DVFS>. Acesso em: 02 dez. 2015.

GHANBARI, M. **Standard Codecs: Image Compression to Advanced Video Coding**. United Kingdom: The Institution of Electrical Engineers, 2003.

GONZALEZ, R.; WOODS, R. **Processamento de Imagens Digitais**. São Paulo: Edgar Blücher, 2003.

GOSH, A. et. al. **Evolutionary Algorithms for Multi-Criterion Optimization: A Survey**. International Journal of Computing & Information Sciences. 2004.

GRELLERT, Mateus. **Computational Effort Analysis and Control in High Efficiency Video Coding**. Dissertação de Mestrado. Mestrado em Ciência da Computação, Universidade Federal do Rio Grande do Sul. Porto Alegre, 2014.

GROIS, D., et. al. **Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders**. PCS, pp. 394-397, 2013.

HÄHNEL, Marcus; et. al.. **Measuring Energy Consumption for Short Code Paths Using RAPL**. Greenmetrics. Londres, 2012.

HASKELL, Barry; PURI, Atul; NETRAVALI, Arun. **Digital Video: An Introduction to MPEG-2**. New York: Kluwer, 2002.

HENNESSY, John; PATTERSON, David. **Computer Architecture: a quantitative approach**. San Francisco: Elsevier, 2007.

ITU-T. **ITU-T Recommendation H.265: High Efficiency Video Coding**. ITU-T. [S.I.]. 2015.

ITU-T. **Subjective Video Quality Assessment Methods for Multimedia Applications**. Geneva, 1999.

KALALI, Ercan et. al. **A Low Energy HEVC Inverse DCT Hardware**. IEEE Third International Conference on Consumer Electronics (ICCE), Berlim, 2013.

KHUN, P. **Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation**. Boston: Kluwer Academic Publishers, 1999.

LEE, Y.; KIM, J.; KYUNG, C. M.. **Energy-aware Video Encoding for Image Quality Improvement in Battery-Operated Surveillance Camera**. IEEE TVLSI, vol. 20, no. 2, pp. 1899-1909, 2012.

LI, P.; CHENG, Y.; JI, W.. **Rate-distortion-complexity analysis on AVS encoder**. PCM, pp. 73-83, Springer, 2011.

LI, Sheng et. al. **McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Manycore Architectures**. International Symposium on Microarchitecture, 2009.

LIKWID. **Likwid Powermeter**. Disponível em: <<https://code.google.com/p/likwid/wiki/LikwidPowermeter>>. Acesso em: 03 dez. 2015.

MA, Zhan; HU, Hao; WANG, Yao. **On Complexity Modeling of H.264/AVC Video Decoding and Its Applications for Energy Efficient Decoding**. IEEE TM, vol. 13, no. 6, pp. 1240-1255, 2011.

MARWEDEL, P. **Embedded Systems Design**. Dordrecht, The Netherlands: Springer, 2006.

MONTEIRO, Eduarda et. al. **Rate-Distortion and Energy Performance of HEVC Video Encoders**. Power and Timing Modeling Optimization and Simulation (PATMOS), 24th International Workshop, 2014.

MONTEIRO, Eduarda et. al. **Rate-Distortion and Energy Performance of HEVC and H.264/AVC Encoders: a Comparative Analysis**. International Symposium on Circuits and Systemas (ISCAS), 2015.

ODROID. **Odroid XU3**. Disponível em: <http://www.hardkernel.com/main/products/prdt_info.php?g_code=G140448267127>. Acesso em: 20 dez. 2015.

ORTEGA A.; RAMCHANDRAN, K.. **Rate-distortion methods for image and video Compression**. IEEE Signal Processing Magazine, vol. 15, pp. 23-50, 1998.

PADRUVA, Srinivas. Intel **Open Source Technology Center: Running Average Power Limit**. Disponível em <<https://01.org/blogs/tlcounts/2014/running-average-power-limit-%E2%80%93-rapl>>. Acesso em: 03 dez. 2015.

PAN, Jacob. **RAPL (Running Average Power Limit) Driver**. Disponível em: <<http://lwn.net/Articles/545745/>>. Acesso em: 02 dez. 2015.

PAPI. **Accessing the Intel RAPL Registers**. Disponível em: <http://icl.cs.utk.edu/projects/papi/wiki/PAPITopics:RAPL_Access>. Acesso em: 03 dez. 2015.

PORTO, Marcelo Schiavon. **Desenvolvimento Algorítmico e Arquitetural para a Estimação de Movimento na Compressão de Vídeo de Alta Resolução**. Tese de Doutorado. Doutorado em Ciência da Computação, Universidade Federal do Rio Grande do Sul. Porto Alegre, 2012.

RAPL. **Reading RAPL Energy Measurements from Linux**. Disponível em: <web.eece.maine.edu/~vweaver/projects/rapl/>. Acesso em: 02 dez. 2015.

RICHARDSON, I. **Video Codec Design: Developing Image and Video Compression Systems**. Chichester: John Wiley and Sons, 2002.

RICHARDSON, I. **H.264/AVC and MPEG-4 Video Compression – Video Coding for Next-Generation Multimedia**. Chichester: John Wiley and Sons, 2003.

ROSEWARNE, C. et. al. **HM16: High Efficiency Video Coding Test Model 16 Encoder Description** (JCTVC-K1002), Warsaw, 2015.

SAAB, Farah et. al. **Energy-Efficient HEVC Using Stochastic Processing**. 4th International Conference on Energy Aware Computing Systems and Applications (ICEAC), 2014.

SANBORN, S; MA, X. **Quantifying Information Content in Data Compression Using the Autocorrelation Function**. IEEE Signal Processing Letters, [S.I.], v.12, p.230-233, Mar. 2005.

SANCHEZ, Gustavo F. **Padrão Emergente 3DVC: Algoritmos e Novas Ferramentas para a Codificação de Vídeos 3D**. Trabalho Acadêmico (Mestrado). Mestrado em Ciência da Computação, Universidade Federal de Pelotas, Pelotas, 2013.

SILVA, Maíra Martins da Silva. **Técnicas de Otimização Clássica**. Disponível em: <http://disciplinas.stoa.usp.br/pluginfile.php/118220/mod_resource/content/1/OPT_classical_opt.pdf>. Acesso em 16 de dez. 2015.

SULLIVAN, G., MARPE, D., WIEGAND, T. **The H.264/MPEG4 Advanced Video Coding Standard and its Applications**. IEEE Communications Magazine, vol.44, pp. 134-143, 2006.

SULLIVAN, Gary J. et al. **Overview of the High Efficiency Video Coding (HEVC) Standard**. IEEE Transactions on Circuits and Systems for Video Technology, Vol, 22, NO. 12, 2012.

SULLIVAN, G.; WIEGAND T.. **Rate-distortion optimization for video Compression**. IEEE Signal Processing Magazine, vol. 15, pp. 74-90, 1998.

SUN, Lin et. al. **Rate Distortion Modeling and Adaptive Rate Control Scheme for High Efficiency Video Coding (HEVC)**. International Symposium on Circuits and Systemas (ISCAS), 2014.

SZE, Vivienne et. al. **High Efficiency Video Coding (HEVC): Algorithms and Architectures**. New York: Springer, 2014.

TOMUSK, Erik. **Practicalities of Design Space Exploration with gem5 and McPAT**. Institute for Computation Systems Architecture. University of Edinburgh, 2012.

VALLE, Marcos Eduardo. **Multiplicadores de Lagrange**. Disponível em: <<http://www.ime.unicamp.br/~valle/Disciplinas/MA211/Aula8.pdf>>. Acesso em 15 de dez. 2015.

VENKATESH, A.; et. al. **Evaluation of Energy Characteristics of MPI Communication Primitives with RAPL**. IEEE 27th International Symposium on Paralell & Distributed Processing Workshps and PhD Forum. 2013.

WIEGAND T.; SULLIVAN, G.; Bjontegaard G.; LUTHRA A.. **Overview of the H.264/AVC video coding standard**. IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, pp. 560-576, 2003.

WOLF, W. **Computer as components: principle of embedded computing system design**. Elsevier, 2005.

XU, Qian et. al. **A New Source Model and Accurate Rate Control Algorithm With QP and Rounding Offset Adaptation**. IEEE International Conference on Image Processing, 2008.

YOUTUBE. **Youtube Estatísticas**. Disponível em: <<https://www.youtube.com/yt/press/pt-BR/statistics.html>>. Acesso em 03 de fev. 2016.

ZATT, Bruno. **Energy-Efficient Algorithms and Architectures for Multiview Video Coding**. Tese de Doutorado. Doutorado em Microeletrônica, Universidade Federal do Rio Grande do Sul. Porto Alegre, 2012.

ZHU J.; LIU Z.; WANG D.. **Fully pipelined DCT/IDCT/Hadamard unified transform architecture for HEVC Codec**. International symposium on circuits and systems (ISCAS), 2013.

APÊNDICES

APÊNDICE A – Lista de Publicações

A) Trabalhos completos publicados em anais de congressos:

Evento: SBCCI 2015 – 28th Symposium on Integrated Circuits and Systems Desing

Título: *Real-Time Architecture for HEVC Motion Compensation Sample Interpolator for UHD Vídeos.*

Autores: Wagner Penny, Guilherme Paim, Marcelo Porto, Luciano Agostini e Bruno Zatt.

Qualis: B1

Evento: SIM 2015 – 30º Simpósio Sul de Microeletrônica

Título: *A Hardware Architecture for an HEVC Motion Compensation Luminance Interpolator.*

Autores: Wagner Penny, Marcelo Porto, Luciano Agostini e Bruno Zatt.

Observação: Prêmio de melhor artigo do congresso (*Best Paper Award*)

B) Resumos expandidos publicados em anais de congressos:

Evento: ENPOS 2015 – XVII Encontro de Pós-Graduação

Título: Avaliação do Consumo Energético do Codificador HEVC Avtravés da Variação de Parâmetros de Codificação.

Autores: Wagner Penny, Ítalo Machado, Marcelo Porto, Luciano Agostini e Bruno Zatt.

Evento: CIC 2015 – XXIV Congresso de Iniciação Científica

Título: Análise do Consumo de Energia em CPU do Codificador HEVC Utilizando a Ferramenta RAPL.

Autores: Ítalo Machado, Wagner Penny, Marcelo Porto, Luciano Agostini e Bruno Zatt.

ANEXOS

ANEXO A – Detalhamento da sequência *BlowingBubbles*

1



5



10



15



20



25



30



35



40



45



50