

7 Resultados Obtidos

Medidas de qualidade objetivas e subjetivas foram extraídas do algoritmo de ajuste de áudio e comparadas com os resultados obtidos com o algoritmo *Music 01* do *Sound Forge 8.0*. O *Sound Forge 8.0* foi escolhido como ferramenta de comparação dentre as enumeradas nos trabalhos relacionados pelos seguintes motivos: a aplicação dos seus algoritmos de ajuste gera um arquivo com o áudio ajustado e comprimido, trata-se de uma ferramenta bastante utilizada atualmente, inclusive em aplicações comerciais de processamento de áudio, e foi acessível durante a realização da pesquisa. Dentre os algoritmos do *Sound Forge*, alguns são específicos para manter determinadas características do áudio. O algoritmo *Music 01* foi escolhido por ter propósito mais geral (preservar a qualidade do áudio como um todo) e ser opção de ajuste *default* da ferramenta.

Todos os áudios utilizados nos testes deste capítulo foram codificados no formato MPEG-1 MP3 *stereo*, 44.1kHz e 128kbps. Esses parâmetros foram escolhidos por serem atualmente tipicamente utilizados durante codificação de áudio. Foram utilizados 5 tipos de áudio de curta duração (com duração de aproximadamente 8 segundos) e 2 de longa duração (cerca de 8 minutos), listados na Tabela 8.

Nome	Conteúdo	Duração (em seg)
Castanholas	Som de instrumento de percussão.	7.05
Flauta	Som de três notas de instrumento melódico tocadas em seqüência. Cada nota dura cerca de um terço da duração total do arquivo.	7.37
Voz	Voz feminina em Português.	8.70
Música	Música de nome “Águas de Março” de Tom Jobim, contendo voz feminina e masculina.	7.37
Filme	Trecho do áudio do filme “Matrix” com som de fundo e duas vozes.	8.93
Voz	Voz masculina em Português.	484

Música	Música “Gabriela” de Tom Jobim, contendo voz masculina e feminina.	482
---------------	--	-----

Tabela 8 - Tipos de áudio utilizados nos teste de qualidade¹⁵.

O Apêndice D ilustra uma ferramenta com interface gráfica que pode ser utilizada para solicitar processamento da ferramenta de ajuste. Essa aplicação foi utilizada para processar os áudios listados na Tabela 8 utilizando o algoritmo de ajuste proposto para áudio neste trabalho, que chamaremos de HyperAudioScale.

As seções a seguir detalham as medidas subjetivas e objetivas do algoritmo proposto. Por fim, também é apresentada uma análise da implementação da ferramenta de ajuste.

7.1. Medidas de Qualidade Subjetiva

As subseções a seguir descrevem primeiramente como o experimento foi realizado e depois apresenta os resultados obtidos.

7.1.1. Descrição do experimento

A elaboração e realização do teste para medição da qualidade subjetiva foram baseadas no padrão ITU-T P.911 (ITU, 1998) e no trabalho de Winkler e Faller (Winkler & Faller, 2005). Foram utilizados os 5 áudios de curta duração da Tabela 8. A escolha dos áudios foi baseada nas categorias recomendadas pela ITU-T, ilustradas na Tabela 9.

Categoria	Descrição
I	Voz de um locutor
II	Voz de mais de um locutor
III	Voz + música de fundo
IV	Som de um único instrumento musical
V	Som de vários instrumentos musicais

Tabela 9 - Categorias de tipos de áudio recomendadas para testes subjetivos, definidas pela ITU-T - fonte:(ITU, 1998).

Durante o teste, cada áudio foi reproduzido 16 vezes: 8 vezes o original, 4 fluxos processados utilizando o algoritmo proposto, com fatores de ajuste de 0.90, 0.95, 1.05 e 1.10, e 4 fluxos processados com o algoritmo *Music 01* do *Sound Forge 8.0* com os mesmos fatores. Portanto, cada participante ouviu 80 áudios. Os

¹⁵ Alguns áudios da tabela foram extraídos de uma seleção da *Audio Engineering Society*.

áudios foram apresentados sem interrupção, em pares (o original seguido de um áudio processado a partir deste), em ordem aleatória, sendo determinístico apenas o fato de que o mesmo tipo de áudio não seria reproduzido novamente antes que dois outros tipos o fossem. Entre dois áudios diferentes foi introduzido um tempo de 5 segundos de silêncio para o ouvinte avaliar o áudio que acabou de ouvir. O tempo total do teste para cada ouvinte foi, então, de pouco mais de 17 minutos ($80*8 + 80*5$).

Segundo a ITU-T (ITU, 1998), os testes subjetivos devem envolver entre 6 e 40 pessoas, dependendo da complexidade do problema. Neste trabalho, 10 pessoas participaram do teste. Todas elas afirmaram possuir audição normal. Os testes foram realizados com fone de ouvido *Philips SBC HP250* e software de reprodução *Windows Media Player versão 9* (Microsoft, 2005).

A qualidade de cada áudio foi atribuída relativamente, considerando que a qualidade do áudio original era excelente. A escala de avaliação utilizada possui 9 níveis (ITU, 1998), sendo cinco principais (Péssimo, Ruim, Razoável, Bom e Excelente) e quatro intermediários. O ouvinte deveria marcar um 'X' na qualidade julgada em um dos 9 níveis. A Figura 70 ilustra a escala adotada e a Tabela 10 resume a interpretação dos níveis principais de qualidade. Os níveis intermediários deveriam ser utilizados quando o ouvinte estivesse em dúvida entre dois níveis principais.



Figura 70 - Escala de votação utilizada para avaliar cada áudio no teste subjetivo.

Nível de qualidade	Nível de degradação
Excelente (muito boa)	Imperceptível
Bom	Pouco perceptível, mas não incômoda
Razoável	Perceptível e levemente incômoda
Ruim	Incômoda, mas não intolerável
Péssimo (muito ruim)	Muito incômoda (intolerável)

Tabela 10 - Interpretação de cada nível de qualidade.

Os ouvintes foram instruídos que alguns áudios poderiam ter a duração acelerada ou retardada, mas que a nota de qualidade por causa desse fator só deveria ser diminuída caso a velocidade de apresentação fosse incômoda. Além disso, os ouvintes foram treinados antes da realização dos testes, com objetivo de: tomar conhecimento de alguns defeitos comuns a áudios ajustados; serem apresentados aos áudios utilizados no teste; e serem expostos a uma simulação do teste para esclarecerem quaisquer dúvidas. Durante o treinamento, cada pessoa ouviu um conjunto de 5 pares de áudios. Sendo um de cada tipo da Tabela 8, apresentados em ordem aleatória.

7.1.2. Resultados obtidos

A média das notas obtidas para cada tipo de áudio, fator e algoritmo de ajuste são apresentadas na Tabela 11. As duas primeiras linhas da tabela apresentam os títulos dos dados apresentados. As próximas quatro linhas da tabela apresentam as médias das notas atribuídas, chamada de MOS (*Mean Opinion Score*), para diferentes fatores de ajuste processados utilizando o algoritmo proposto. As linhas seguintes são interpretadas de maneira similar só que o algoritmo de ajuste utilizado é o do *Sound Forge*.

Algoritmo de ajuste	Fator de ajuste	Tipo de áudio					Média por fator e algoritmo
		Castanholas	Flauta	Voz	Música	Filme	
Proposto	0.90	7.1	7.3	7.9	8.3	8.1	7.7
	0.95	6.8	8.0	8.7	8.4	8.4	8.1
	1.05	8.1	8.6	6.2	8.1	7.3	7.7
	1.10	7.4	7.6	5.8	7.1	7.0	7.0
Sound Forge	0.90	8.3	8.8	8.9	8.1	8.6	8.5
	0.95	8.6	8.9	8.9	8.6	8.2	8.6
	1.05	7.8	8.7	8.5	7.8	8.2	8.2
	1.10	7.4	8.5	8.0	8.0	8.4	8.1

Tabela 11 - MOS obtido para cada tipo de áudio e fator de ajuste.

A Figura 71 e a Figura 72 ilustram graficamente as notas da Tabela 11, juntamente com os intervalos de confiança das notas com coeficiente de 95% (Jain, 1991). Nas figuras, o segmento perpendicular a cada intervalo de confiança é o MOS. A Figura 71 ilustra os valores para os áudios processados com fatores 0.90 e 0.95 e a Figura 72 com os fatores 1.05 e 1.10. Os áudios processados pelo HyperAudioScale e pelo *Sound Forge* são legendados,

respectivamente, pelas iniciais “h” e “s”, seguidos do fator de ajuste (em %) utilizado.

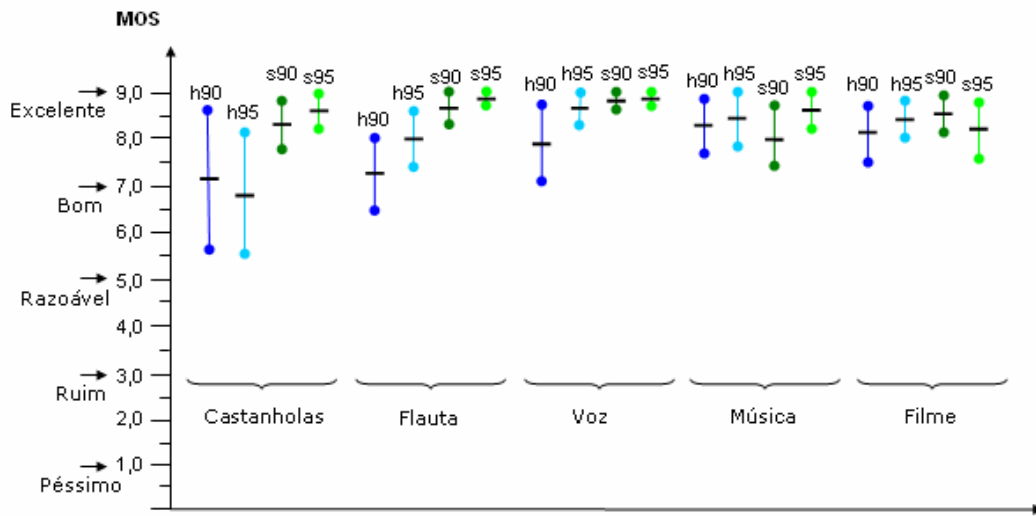


Figura 71 - MOS obtido para cada tipo de áudio com fator de ajuste 0.90 e 0.95 com coeficiente de confiança 95%.

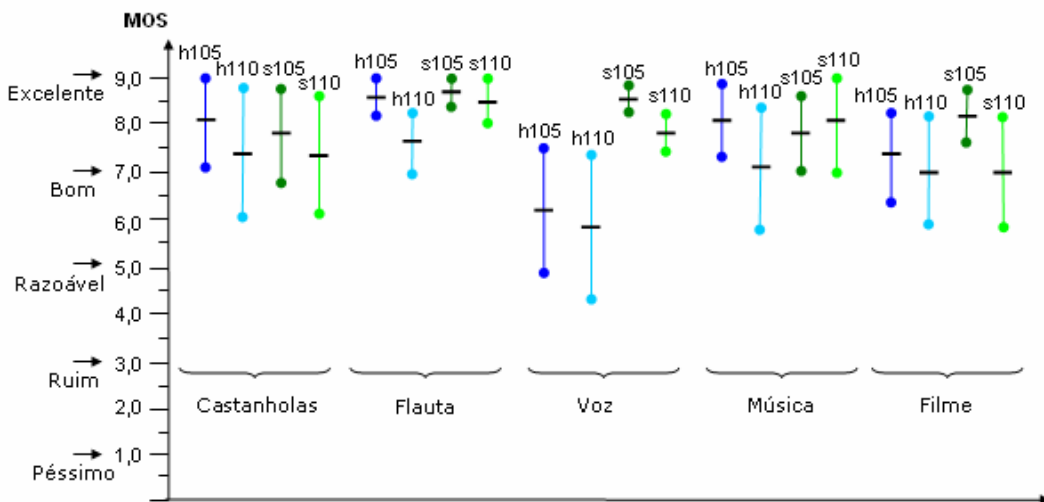


Figura 72 - MOS obtido para cada tipo de áudio com fator de ajuste 1.05 e 1.10 com coeficiente de confiança 95%.

Analisando os dados, pode-se notar que o algoritmo proposto gerou resultados bastante similares aos resultados do *Sound Forge* para a voz, música e filme utilizando fatores de 0.90 e 0.95, e para castanholas utilizando fatores de 1.05 e 1.10, sendo que seu pior resultado foi o áudio de voz com fator 1.05 e 1.1. Mesmo para esses casos, o nível de aceitação está na faixa do razoável. De modo geral, pelo menos para o resultado deste teste, o algoritmo proposto se aproximou

do resultado do *Sound Forge*, tendo algumas vezes até obtido resultados melhores.

Levando-se em consideração as diferenças entre os algoritmos comparados (conforme discutido no Capítulo 2), como, por exemplo, o fato do algoritmo proposto funcionar em tempo de execução, e também as comparações de eficiência e precisão do algoritmo proposto, conforme será apresentado na próxima seção, o resultado do algoritmo proposto é bastante promissor, especialmente para aplicações em que o ajuste é aplicado em partes pequenas de áudio, como é o caso daquelas que usam o ajuste visando apenas a manutenção das relações de sincronismo.

Uma segunda análise dos dados visa comparar a qualidade atingida por diferentes tipos de áudio. Nesse momento, é importante considerar quais áudios utilizados no teste são mais bem-comportados, que significa conter um pequeno intervalo de frequências num dado instante de tempo e possuir uma variação suave nas mudanças de frequências. Os áudios de voz humana e flauta são mais bem-comportados do que os demais. O áudio das castanholas, embora tenha apenas o som de um instrumento musical, possui muito tempo com ruído branco, o que o torna mal comportado. Os áudios da música e filmes também possuem várias frequências num dado momento.

A Tabela 11 ilustra as médias obtidas dos diferentes fatores de ajuste para cada tipo de arquivo. É esperado que a qualidade do algoritmo do *Sound Forge* seja melhor para áudios bem-comportados, pois ele pode tirar proveito das características do sinal para aplicar o ajuste. Desse modo, as maiores diferenças entre o algoritmo proposto e o *Sound Forge* ocorrem com o áudio da voz humana e flauta. Por outro lado, a diferença entre os resultados dos demais áudios foram menores.

Uma última análise dos dados mostra que a qualidade atribuída aos áudios mais acelerados foram ligeiramente maiores do que a dos menos acelerados utilizando a mesma variação do fator de ajuste. Ou seja, as médias para o fator *0.95* são maiores do que as do fator *1.05* e as médias para o fator *0.90* são maiores do que as do fator *1.10*. A primeira possibilidade para explicar essa afirmativa é que os algoritmos de ajuste funcionam melhor para acelerar do que retardar. No caso do algoritmo proposto, cada remoção de quadro acarreta uma descontinuidade e supressão de parte do sinal enquanto uma duplicação acarreta

duas descontinuidades e duplicação de parte do sinal. Outra possibilidade é que o ser humano prefere assistir uma apresentação com exibição mais rápida do que mais devagar, resultado esse também demonstrado por alguns estudos (Omoigui et al., 1999; Amir et al., 2000).

7.2. Medidas de Qualidade Objetivas

Algumas medidas objetivas de qualidade também foram extraídas do algoritmo proposto. Os testes realizados utilizaram os áudios da Tabela 8. Esta seção analisa a eficiência de processamento do algoritmo, a precisão da duração atingida no fluxo de áudio processado e a linearidade na escolha dos quadros a processar em fluxos de áudio com reservatório de bits. As medidas de eficiência e precisão são comparadas com o algoritmo *Music 01* do *Sound Forge*.

7.2.1. Eficiência

A ferramenta de ajuste proposta demora cerca de 10 segundos para processar qualquer um dos áudios longos (de aproximadamente 8 minutos) apresentados na Tabela 8 com fator 1.10, utilizando uma máquina *Pentium 4 2.4GHz 1GB-RAM*, sem nenhum outro processo computacionalmente custoso sendo executado paralelamente. O *Sound Forge*, por sua vez, demora cerca de 54 segundos para as mesmas condições, sendo cerca de 27 segundos para abrir o arquivo, de 12 segundos para processá-lo e de 15 segundos para salvar o fluxo modificado em formato MP3. Vale destacar que os tempos medidos do *Sound Forge* não consideram o tempo que o usuário da ferramenta precisa para acessar as funcionalidades da ferramenta, somente o tempo utilizado pela ferramenta.

Sabe-se que a duração dos quadros de áudio pode ser calculada pela Figura 73. Desse modo, no caso deste teste, tem-se que a duração é de 26.12ms $[=(1152/44.1)*10^{-3}]$.

$$Duração_quadro = \frac{Total\ amostras\ do\ quadro}{taxa_de_amostragem\ do\ quadro}$$

Figura 73 - Cálculo da duração de um quadro do fluxo elementar.

Efetuando cálculos com base nos dados acima, o algoritmo proposto conseguiu atingir a taxa de 1849 quadros/segundo enquanto que para exibir o áudio é utilizada uma taxa de apenas 38.3 quadros/segundo.

Um último teste de eficiência foi realizado acrescentando metadados de sistemas MPEG-2 nos áudios MP3 anteriormente mencionados utilizando o algoritmo proposto. Com isso, o tempo total de processamento para os arquivos de 8 minutos de duração subiu de cerca de 10 para cerca de 20 segundos. Esse aumento do tempo de processamento se deve a necessidade de processar os dados do fluxo de sistema e instanciar *threads* que processam o áudio do fluxo elementar de trecho em trecho.

7.2.2. Precisão

A precisão dos algoritmos de ajuste elástico foi obtida através do cálculo da esperança da diferença entre a duração ótima e a atingida do fluxo, ajustado com fator 0.9 e 1.1, utilizando os arquivos da Tabela 8. Para analisar esse parâmetro, é válido ressaltar que a duração dos fluxos de áudios divididos em quadros é obtida através da multiplicação do número de quadros no fluxo pela duração dos quadros. Desse modo, nenhum algoritmo de ajuste é capaz de sempre gerar a duração ótima, uma vez que a duração dos fluxos divididos em quadros é sempre múltiplo da duração dos quadros. No caso desse teste, os quadros possuem duração igual a 26.12ms.

A Tabela 12 ilustra os resultados obtidos. Nos testes realizados, a precisão obtida em arquivos curtos com o HyperAudioScale foi menor do que um quadro, que é de 26.12ms (com exceção de um caso por menos de 1ms), e de cerca de 2 quadros utilizando o *Sound Forge*. Para arquivos longos, os algoritmos obtiveram precisões iguais, menor do que um quadro.

Fator de ajuste aplicado	Arquivo	E (Duração (ótima - atingida))	
		Algoritmo proposto (ms)	Algoritmo <i>Sound Forge</i> (ms)
0.90	Castanholas	-5	-55
1.10	Castanholas	-5	-55
0.90	Flauta	23	-57
1.10	Flauta	-13	-43
0.90	Voz curta	20	-60
1.10	Voz curta	-20	-40
0.90	Música curta	23	-57
1.10	Música curta	7	-43
0.90	Filme	17	-63
1.10	Filme	-27	-47
0.90	(MÉDIA CURTOS)	15.6	-58.4
1.10	(MÉDIA CURTOS)	-11.6	-45.6
0.90	Voz longa	-14	-14
1.10	Voz longa	-16	-16

0.90	Música longa	-14	-14
1.10	Música longa	-16	-16
0.90	(MÉDIA LONGOS)	-14	-14
1.10	(MÉDIA LONGOS)	-16	-16

Tabela 12 - Precisão do processamento de arquivos MP3 (em milissegundos).

7.2.3. Linearidade

Conforme comentado na Seção 3.3, nem todos os quadros do fluxo de áudio podem ser escolhidos para processamento quando existe reservatório de bits. Isso significa que o algoritmo de ajuste elástico proposto pode ser não-linear em relação à escolha dos quadros a processar. Para medir a não-linearidade do algoritmo proposto para áudios com reservatório de bits, foi definida uma variável aleatória X que conta o número de quadros entre dois quadros processados.

A Tabela 13 ilustra a esperança e o desvio padrão dessa variável aleatória estimados com base nos arquivos da Tabela 8. É esperado que essa medida tenha piores resultados quando o número de quadros a serem processados cresce. Sendo assim, os fatores escolhidos para esse teste foram 0.9 e 1.1 . Com esses fatores, espera-se obter $X=10$.

Fator de ajuste aplicado	Arquivo	X= Número de quadros entre dois quadros processados	
		Esperança	Desvio padrão
0.90	Castanholas	9.96	5.34
1.10	Castanholas	10.00	0.54
0.90	Flauta	9.96	0.32
1.10	Flauta	10.0	0.21
0.90	Voz curta	9.96	5.34
1.10	Voz curta	10.00	0.53
0.90	Música curta	10.00	0.50
1.10	Música curta	10.00	0.52
0.90	Filme	9.70	5.56
1.10	Filme	10.0	0.24
0.90	Voz longa	10.0	2.21
1.10	Voz longa	10.0	0.44
0.90	Música longa	10.0	0.30
1.10	Música longa	10.0	0.25

Tabela 13 - Medida de não-linearidade do processamento de arquivos MP3.

Analisando a Tabela 13, verifica-se que a esperança medida de X de fato encontra-se bem próxima (ou igual) a 10 . Alguns arquivos de curta duração não conseguiram obter a média de 10.00 com precisão de duas casas decimais, o que se refletiu na precisão medida na subseção anterior, entretanto o valor foi atingido por arquivos maiores.

Os valores de desvio padrão são pequenos. O maior valor atingido foi 5.56, no caso do áudio do filme e ainda esse valor não é ruim, pois é aproximadamente metade do valor da média. Isso significa que não existem muitos casos em que dois quadros simultâneos ou muito próximos são processados. Conclui-se, assim, que a não-linearidade do algoritmo proposto não é muito significativa, pelo menos nos áudios testados. Isso é interessante porque implica que não ocorrem mudanças significativas do fator de ajuste aplicado em diferentes partes do fluxo de áudio.

Vale observar que o desvio padrão é, em geral, maior para fatores de 0.9 do que de 1.1. Esse resultado é esperado uma vez que, em fluxos de áudio com reservatório de bits, devem existir mais quadros com unidades lógicas menores do que o campo físico do que o contrário.

7.3. Análise da Implementação

A implementação atual da ferramenta de ajuste está quase finalizada, sendo sua principal pendência o fato de não estar plenamente integrada com o algoritmo de ajuste de vídeo, como mencionado na Seção 4.5. Esta subseção analisa a extensibilidade do código fonte para novos formatos de mídia e o retardo introduzido pelo algoritmo de ajuste da ferramenta.

7.3.1. Extensibilidade

O subsistema de ajuste de áudio atualmente oferece suporte a vários formatos. Considerando que o processamento é realizado nas classes de formato genérico de áudio, a inclusão de um novo formato de áudio pode ser realizada simplesmente definindo um conjunto de classes que contêm as particularidades do formato, mais um *assembler* para construir tais classes a partir de dados binários, adicionado a um meio de reconhecer o novo formato para instanciar as classes corretas.

Por outro lado, os subsistemas de vídeo e sistemas foram desenvolvidos apenas para formatos MPEG-2, embora algum esforço tenha sido feito para abstrair as particularidades do formato. Para efetivamente generalizar tais subsistemas, é necessário estudar outros formatos de modo a coletar o que

realmente pode ser independente e o que deve ser instanciado para cada formato de mídia específico.

Ao total, a ferramenta de ajuste contém 172 classes, sendo que 49 são do algoritmo de áudio, 41 do de vídeo e 20 do de sistemas. 37 classes são genéricas para serem utilizadas por algoritmos de ajuste de qualquer mídia e 25 realizam a comunicação com exibidores de conteúdo.

7.3.2. Retardo Introduzido

O retardo inicial introduzido pela ferramenta de ajuste é definido como o tempo mínimo que é necessário acrescentar antes de iniciar a reprodução de um fluxo de mídia. Esse tempo deve ser suficiente para que a exibição da mídia possa ocorrer sem interrupção, precedida pela realização do ajuste elástico, que pode continuar ocorrendo em paralelo à exibição.

Sabe-se que o tempo total de processamento para realizar ajuste elástico em fluxos de sistemas é expresso pela fórmula abaixo (ver também Figura 59).

$$T_{Total} = T_{IniDemux} + \sum_{trechos} T_{Controlador} + \sum_{trechos} Max\{AjusteFE_i\} + T_{FinalMux}$$

Figura 74 - Cálculo do tempo total para ajustar fluxo de sistemas.

O $T_{IniDemux}$ é o tempo necessário para o demultiplexador encontrar e montar o primeiro PACKET do fluxo de sistemas. A partir daí, o demultiplexador sinaliza que o controlador pode efetivamente começar sua execução.

Os tempos $T_{Controlador}$ e o $Max\{AjusteFE_i\}$ devem ser contabilizados para cada trecho da mídia original. $T_{Controlador}$ é o tempo que o controlador de sistemas precisa para percorrer todos os fluxos elementares e configurar parâmetros para execução de seus processadores. Em cada trecho, os tempos necessários para realizar ajuste elástico em cada fluxo elementar podem ser diferentes. Uma vez que o controlador deverá esperar a finalização do processamento de todos os fluxos, o tempo a ser contabilizado é o do processamento do fluxo elementar mais demorado, dado por $Max\{AjusteFE_i\}$.

O multiplexador começa a executar logo que o controlador processa o primeiro PACKET do fluxo de mídia e continua até terminarem todos os PACKETS do fluxo de sistemas. O $T_{FinalMux}$ indica o tempo decorrente quando o controlador já terminou sua execução, juntamente com os processadores de

fluxos elementares, e o multiplexador ainda precisa dar saída aos últimos PACKETs do fluxo de sistemas.

Considerando, entretanto, que o processamento pode ser realizado durante a exibição da mídia e as medidas apresentadas da Subseção 7.2.1, pode-se afirmar que os principais fatores de retardo inicial a serem considerados são a montagem de um PACKET inicial e retardo introduzido por dependências de bits nos fluxos elementares. Na realidade, o retardo inicial só é realmente um problema se o ajuste estiver sendo realizado em tempo de exibição e não for possível acessar os dados da mídia com velocidade superior à de exibição.

A montagem do PACKET inicial é imprescindível, uma vez que não é possível que a ferramenta de ajuste dê saída a um PACKET do fluxo de sistemas enquanto todos os seus bits não tenham sido processados. Isso acontece porque as estruturas PACKET do fluxo MPEG-2 possuem campos que devem ser atualizados somente depois do processamento de todos os quadros de fluxos elementares nele contidos.

Para dar uma idéia do retardo que pode ser introduzido por este fator, pode-se considerar o pior caso, que é o PACKET de áudio. Sabendo que o tamanho máximo de um PACKET é de 2^{16} bytes e que o tamanho típico de um pacote de áudio é de 417 bytes, é possível codificar até 157 quadros de áudio em um único PACKET. No entanto, dificilmente o codificador irá deixar tantos quadros de áudio juntos, sendo seu comportamento padrão intercalá-los entre quadros de vídeos. Nos exemplos testados, tem-se um valor típico de 20 quadros de áudio em um único PACKET. Ainda com número reduzido de quadros, é necessário acrescentar um retardo cerca de $20 * 26.12 * 10^{-3} \approx 0.5$ segundos para receber todos os dados e começar a realizar o ajuste.

Além disso, deve-se considerar o retardo introduzido por dependências de bits nos fluxos elementares. O algoritmo de ajuste de áudio introduz retardo em formatos que possuem dependência entre quadros, pois é necessário armazenar em memória todos os quadros dos quais o quadro processado depende. Por exemplo, como mencionado na Subseção 3.4.1, em fluxos MP3 é possível que um quadro possa depender de até 10 quadros anteriores. Assim, é possível ter que introduzir um retardo de até $10 * 36 * 10^{-3} = 0.36$ segundos para receber os dados de mídia em fluxos MP3. No caso de fluxos de sistemas a situação é ainda pior, pois um quadro de áudio que está em um PACKET pode depender de um quadro que está

em um PACKET anterior. Como esses dois PACKETS ainda podem estar intercalados por PACKETS de fluxos de outras mídias, o retardo a ser introduzido pode ser muito grande.

O algoritmo de MPEG-2 vídeo também introduz um retardo devido à ordem de codificação dos quadros de vídeo ser diferente da ordem de apresentação (Cavendish, 2005). Esse problema introduz retardo de um único quadro de vídeo e, portanto, é menos sério do que o retardo do áudio.

Outro retardo a se observar é o tempo que a ferramenta de ajuste precisa para modificar algum parâmetro de processamento, por exemplo, o fator de ajuste a ser aplicado. Nesse caso, é importante observar o tamanho dos *buffers* existentes para a realização do ajuste (ver Figura 58) e para comunicação com o exibidor de conteúdo. Grandes *buffers* podem introduzir retardo se alguma mudança for solicitada. Por outro lado, pequenos *buffers* podem diminuir desempenho da ferramenta e até ocasionar falhas na exibição da mídia.

O retardo introduzido pela ferramenta de ajuste pode impactar em cenários em que existe interatividade do usuário. Soares (Soares, 2005) fornece alguns dados interessantes para ilustrar como o retardo prejudica a interatividade de uma ligação telefônica. A tabela abaixo ilustra como o aumento do valor de retardo pode confundir os participantes da conversa, aumentando tanto o tempo de silêncio de uma conversa quanto o tempo em que os dois participantes falam.

Estado	Valor do Retardo		
	40ms ¹⁶	600ms	1200ms
Um lado fala	40.5%	40.0%	38.4%
Dois lados falam	4.2%	5.9%	6,2%
Mútuo silêncio	23.2%	25.9%	29.4%

Tabela 14 - Efeitos do aumento do retardo em conversa telefônica - fonte: (Soares, 2005).

Uma opção para resolver ou diminuir o problema do retardo é permitir que a ferramenta de ajuste tenha acesso aos dados audiovisuais com velocidade superior à de exibição. Por exemplo, quando a ferramenta está integrada a um formatador hipermídia, mecanismos de *prefetch* dos objetos no ambiente de execução podem ajudar a solucionar o problema.

¹⁶ O valor de 40ms é o tempo máximo permitido em uma ligação fixa percorrendo distâncias continentais (menos de 5000Km).