

Universidade de São Paulo
Instituto de Astronomia, Geofísica e Ciências Atmosféricas
Departamento de Astronomia

Bruno Corrêa Quint

Estudo e caracterização do instrumento
Brazilian Tunable Filter Imager

São Paulo

2015

Bruno Corrêa Quint

Estudo e caracterização do instrumento
Brazilian Tunable Filter Imager

Tese apresentada ao Departamento de Astronomia do Instituto de Astronomia, Geofísica e Ciências Atmosféricas da Universidade de São Paulo como requisito parcial para a obtenção do título de Doutor em Ciências.

Área de Concentração: Astronomia

Orientadora: Prof.^a Dr.^a Cláudia Lúcia Mendes de Oliveira

São Paulo

2015

Ao meu pai e guardião, Carlos Alberto Quint.

Agradecimentos

Eu quero agradecer, primeiramente, à minha orientadora Cláudia Lúcia Mendes de Oliveira pela oportunidade de ter trabalhado neste projeto, as portas abertas, por todo o suporte dado em momentos bastante difíceis deste período em que estive fazendo este doutorado, pelas risadas e puxões de orelha, pela motivação e energia que me deu para que eu pudesse ter terminado esta obra e, principalmente, por ter acreditado em mim em tantos momentos em que outros duvidavam.

Quero agradecer também ao Denis Andrade e ao Keith Taylor pela paciência e por permanecerem firmes no trabalho junto a este instrumento, fazendo o possível e, muitas vezes, o impossível para tentar fazer o instrumento funcionar.

Agradeço ao pessoal do telescópio SOAR e do CTIO pelo suporte e colaboração em tantas das vezes que precisamos deles para ajudar-me nos mais diversos problemas. Deste pessoal, quero direcionar meus agradecimentos especialmente a Gerardo Gomez, Eduardo Aguirre, Guillermo Dubo, Ian Ordenes e ao Juan Eduardo Serrano pelo suporte na montagem. Agradeço de coração também à Marcela Urquieta, a quem eu tanto incomodei nas minhas viagens com mudanças de planos de última hora, pelas conversas que tivemos, as palavras doces e que me tanto me deram força enquanto eu estava no Chile, e pela ajuda em resolver todo o tipo de problema de logística e acomodação que enfrentei nestes últimos anos. Agradeço também ao Roberto Tighe por ceder-me seu laboratório de óptica no CTIO e pelas lições dadas. Quero agradecer também a Steve Heathcote por ter me aceito como seu aluno orientado no período em que fiquei no Chile.

No IAG, eu gostaria de agradecer todos os colegas com que dividi algum dos tantos momentos especiais que vivi nestes últimos anos. Quero agradecer especialmente ao Felipe Navarete por ter me aguentado por alguns meses no Chile. Agradeço também ao

Fellipy Silva que me aguentou aqui em São Paulo mesmo e que tanto me ensinou sobre a manutenção de elementos ópticos. Agradeço pela companhia e amizade ao Paulo Jackson, Tiago Ricci, Juliana Motter e ao Pedro Paulo Bonetti Beaklini. Por estes motivos e pela companhia nos cafés pós almoço, quero agradecer à Diana Gama, Camile Mendes Castilho, Maria Luiza Dantas, Marcus Duarte e Márcio B. de Avellar.

Quero agradecer aos meus amigos de São Paulo por todos os momentos que tivemos juntos. Cafés, jantãs, almoços, baladas, noites jogando vídeo-game. Meus agradecimentos, em especial, vão para o Rafael Piccollo, o Fábio Saitta, o Marcelo Feal, o Guilherme Peniche e o Diego Santos. Agradeço imensamente e especialmente também à Talita Di Giorgio Torres, à Sandra Di Giorgio e ao Antônio Torres por terem me “adotado” neste período em que estive aqui. Agradeço aos meus atuais *roomies* Ana Paola Salviano, Marielly Malagoli e Pedro Blumberg Moreira aturarem o meu mau humor quando chego cansado em casa, pelos bolinhos e tensas partidas de Bomberman e Mario Kart.

Ainda em São Paulo, quero agradecer imensamente aos amigos que fiz na Bujinkan Budo Taijutsu. Sem os treinos com eles, eu não acredito que teria equilíbrio para superar as dificuldades que passei aqui em São Paulo. Os meus agradecimentos especiais vão para o Giuliano di Sevo (Horo), por ter me aceito em seu dojo, para o Vinícius Barbosa, pelos papos via Facebook enquanto eu estava no Chile, e para o Eduardo Ferrara e o Fernando Silveira que foram meus professores também por um curto tempo.

No Chile, quero agradecer pelos momentos maravilhosos que passei ao Sebastián Araya León, ao Cristian Gallardo Latrille, à Catherine Kaleira (Katie) e ao Felipe Aliprandi. Agradeço também aos meus amigos da Bujinkan Budo Taijutsu de La Serena, em especial ao Marcelo Tapia Mazuela e ao Juan Gerardo Olavarria Roa pela amizade dentro e fora de dojo.

Em Floripa, quero agradecer imensamente aos meus amigos da época da graduação Ian Jordy Lopez Dias, Junior Koch, Rafael Seemann, Diego Grams, Gustavo Tontini, Marcelo Pagani, Lucas Arrigoni Iervolino pelos anos de amizade, indispensáveis para que eu possa recarregar minhas baterias quando vou para Florianópolis.

Quero agradecer também à minha família, em Florianópolis, também pelo forte carinho que recebo sempre que os visito. Quero agradecer também à Mary Stella Assis Costa por ter me ajudado direta e indiretamente nos meus últimos anos de doutorado.

Quero agradecer especialmente à minha mãe, Elisa Aparecida Corrêa, que, hora general,

hora extremamente carinhosa, sempre me apoiou em todas as minhas decisões e sempre esteve torcendo por mim. Ao meu irmão, André Corrêa Quint, por estar do meu lado e por assumir a dianteira em tantas situações em que eu estava por fraquejar. Agradeço imensamente ao meu pai, Carlos Alberto Quint, pelas lições dadas, pelas conversas, por sempre me aceitar incondicionalmente e por mostrar o quão importante devem ser os laços entre as pessoas queridas.

Agradeço ao PROEX e à CAPES pelo financiamento do meu doutorado no Brasil, ao CNPq pelo financiamento durante a minha estadia no Chile através do programa Ciência Sem Fronteiras pelo processo nº 238887/2012-0. Agradeço ao INCT-A, pelo financiamento das viagens ao Chile e pela colaboração com o financiamento de parte do projeto do BTFI em si pelo processo nº 2008/57807-5 e à FAPESP que também colaborou financiando este projeto (processo nº 2006/56213-9). Agradeço, também, ao LNA pela colaboração e financiamento de parte do projeto.

Por fim, quero agradecer a todos aqueles que, já citados aqui ou não, colaboraram para que esta tese pudesse ser concluída com qualidade, fazendo correções, dando sugestões ou mesmo me dando um simples apoio moral.

“O que tu ganhas ao conquistar teus objetivos não é tão importante quanto o que te tornas para conquistá-los.”

Henry David Thoreau

Resumo

Apresentamos neste trabalho um estudo sobre o instrumento astronômico chamado *Brazilian Tunable Filter Imager*, ou BTFI, desenhado e construído para trabalhar junto ao telescópio SOAR (***S**Outhern **A**strophysical **R**esearch Telescope*), no Chile. Como o seu próprio nome indica, o BTFI é um dispositivo que captura imagens utilizando um filtro espectral ajustável. O conceito de “filtro ajustável” envolve o fato de que podemos obter imagens variando continuamente o comprimento de onda selecionado. Para isto, o BTFI utiliza dois módulos: um Fabry-Perot, para dados com alta resolução espectral, e um novo tipo de filtro espectral ajustável chamado *imaging Bragg Tunable Filter* (iBTF), para dados com baixa resolução espectral.

O iBTF consiste em um dispositivo que utiliza duas redes de difração holográficas paralelas em si de modo que o feixe colimado proveniente do telescópio atravesse as duas sequencialmente. Ao passar pela primeira, a luz sofre difração. Este efeito é cancelado ao passar pela segunda rede recuperando, assim, a informação angular para um certo comprimento de onda. Para um determinado par de redes, a seleção do comprimento de onda que chega aos detectores é feita através do ângulo de incidência da luz com elas.

Já o FP consiste em duas placas de vidro cilíndricas e coaxiais dispostas de modo a formarem uma cavidade óptica entre elas. A luz colimada que vem do telescópio passa pela primeira placa, sofre múltiplas reflexões nesta cavidade e, devido ao efeito de interferência ondulatória que ali ocorre, somente alguns comprimentos de ondas são transmitidos. Aqui, a seleção do comprimento de onda é dada através da variação da largura desta cavidade óptica.

Com estes dois filtros, o BTFI fornece cubos de dados com resolução espectral média de 17 no intervalo de 3800 Å e 6200 Å e de 2000 no intervalo de 6364 Å e 7044 Å com o

iBTF. O BTFI conta com dois FPs: um com resolução espectral de 9500 dentro de um intervalo espectral livre de 11,20 Å e outro com resolução espectral de 4400 dentro de um intervalo espectral livre de 49,6 Å.

Além disso, o BTFI faz uso de dois detectores do tipo *Electron Multiplier CCD* (EMCCD) que permitem rápida leitura das imagens com baixo ruído. O campo de observação com o BTFI é de 3 x 3 minutos de arco com uma amostragem de 0,12 segundos de arco por píxel. A largura da *point-spread-function* (PSF) instrumental foi medida abaixo de 0,3 segundos de arco.

Após a descrição e a caracterização da parte opto-mecânica do instrumento, apresentaremos também alguns dados observacionais obtidos com o BTFI bem como uma descrição do processo envolvido na redução destes dados. Por fim, teremos uma análise dos pontos críticos necessários para a entrega do BTFI à comunidade astronômica brasileira.

Abstract

We present a study about the astronomical instrument called Brazilian Tunable Filter Imager, or BTFI. It was designed and built to work with the SOAR Telescope (***S**Outhern **A**strophysical **R**esearch Telescope*), in Chile. As its name indicates, BTFI is an instrument used to acquire a sequence of images using tunable spectral filters. The concept of a “tunable filter” in astronomy relates to the fact that we can acquire images at a set of selected wavelengths. For that, BTFI uses two technologies: a Fabry-Perot, for relatively high spectral resolution data, and a new type of tunable filter known as an imaging Bragg Tunable Filter (iBTF), that provides relatively low spectral resolution data.

The iBTF consists of a device that uses two parallel volume-phase holographic gratings such that collimated light coming from the source intersects each in turn. When the light passes the first grating, it is diffracted. The diffraction is then canceled when the light passes through the second grating and the angular information at wavelengths within the blaze profile is then recovered. For a given pair of gratings, the wavelengths that hit the detector depends on the incident angle of the light onto the gratings.

The FP, on the other hand, consists of two parallel glass plates forming an optical cavity. Collimated light coming from a point in the telescope image suffers multiple reflections in this optical cavity and, through interference, only a select set of wavelengths are transmitted. Here, the wavelength selection is defined by the size of this optical gap.

With these two tunable filters, BTFI provides data-cubes with an average spectral resolution of 17 between 3800 Å and 6200 Å and 1500 between 6364Å and 7044Å with the iBTF. Also, BTFI has, now, two FPs: one with spectral resolution of 9500, with a free spectral range of 11.20 Å and another with a resolution of about 4400 within a free spectral range of 49.6 Å at 6598 Å.

BTFI uses two Electron Multiplier CCDs as detectors that allow fast image readouts with low noise. BTFI's field-of-view is 3 x 3 arcsec with a pixel sampling of 0.12 arcsec. The angular point-spread-function has a full-width-at-half-maximum of less than 0.3 arcsec.

After a description and characterization of the instrument we present some observational data taken with the instrument. We analyze its problems and describe the next steps to be followed in order to deliver it to the Brazilian Astronomical community.

Sumário

1. <i>Introdução</i>	27
1.1 BTFI como um novo instrumento para o SOAR	29
1.2 Objetivos e estrutura desta tese	34
2. <i>Descrição</i>	37
2.1 Sistema de imageamento	41
2.2 Fabry-Perot	42
2.2.1 Fabry-Perot operando em modo clássico	46
2.3 iBTF	52
2.3.1 Redes de difração holográficas	52
2.3.2 Montagem opto-mecânica do iBTF	58
2.4 Detectores	59
2.5 Sistema de controle	62
3. <i>Operação e Caracterização</i>	67
3.1 Sistema de imageamento simples	67
3.1.1 Alinhamento óptico	68
3.1.2 Foco Instrumental	76
3.1.3 Foco nominal versus foco relativo	76
3.1.4 Degradação da PSF em função do modo de operação do CCD	78
3.1.5 Eficiência operacional do sistema de imageamento simples	79
3.2 Fabry-Perot	81
3.2.1 Alinhamento das placas do étalon	82
3.2.2 <i>Bump</i> espectral - O problema e a solução	84

3.2.3	Eficiência do sistema de controle do FP	91
3.3	iBTF	96
3.3.1	Limitações na fabricação das redes VPH	96
3.3.2	Redes de difração Wasatch	97
3.3.3	Redes de difração Ondax	102
3.3.4	Eficiência operacional	111
4.	Observações	113
4.1	Aquisição de dados	114
4.2	Imagens BIAS	117
4.3	Imagens DARK	121
4.4	Imagens FLAT	123
4.5	Cubos de calibração	124
4.6	Correção de fase	127
4.6.1	Correção de fase de cubos do iBTF	129
4.6.2	Correção de fase de cubos de FPs	133
4.7	Calibração em comprimento de onda	138
4.7.1	Calibrando cubos do iBTF	138
4.7.2	Calibrando cubos do Fabry-Perot	140
4.8	Calibração astrométrica e Ângulo de Posição	149
4.9	Alguns resultados observacionais	152
4.9.1	Observação com o iBTF - NGC2440	152
4.9.2	Observação com o LR FP - NGC7009	153
4.9.3	Observação com o HR FP - 30Dor	154
5.	Conclusões	157
	Referências	163
	Apêndice	167
	Apêndice	169
A.	Ligando e estabilizando o controlador CS100	171

B. Código fonte: <i>mkcube3</i>	175
C. Código fonte: <i>phmextractor.py</i>	181
D. Código fonte: <i>phmfit.py</i>	195
E. Código fonte: <i>phmapply.py</i>	203

Lista de Figuras

2.1	BTFI - Visão frontal.	38
2.2	BTFI - Visão da parte de trás do instrumento.	38
2.3	Desenho óptico simplificado do BTFI.	39
2.4	Módulos internos ao BTFI.	40
2.5	Desenho óptico original do BTFI	41
2.6	Foto do painel frontal do controlador CS100.	45
2.7	Princípio de funcionamento de um Fabry-Perot	46
2.8	Intensidade relativa da luz transmitida através das placas do FP em função da diferença de fase δ e para diferentes reflectâncias R	47
2.9	Exemplo de espectro obtido com um Fabry-Perot.	48
2.10	À esquerda: representação simplificada de duas redes de difração em duas diferentes configurações. À direita, fotos obtidas em laboratório mostrando um feixe de luz proveniente de um cabo de fibras ópticas iluminado por uma lâmpada incandescente passando através do iBTF com as redes a 45° (figuras superiores) e 35° (figuras inferiores). Fotos e figuras extraídas da dissertação de Quint (2010).	52
2.11	Representação geométrica simplificada de uma rede de difração holográfica. Figura baseada em Kogelnik (1969).	53
2.12	Lei de Bragg	54
2.13	Modos de operação das redes de difração holográficas de acordo com o <i>slant angle</i> ϕ . À esquerda, rede de transmissão. À direita, rede de reflexão.	55
2.14	Rede VPH de transmissão simulada	57
2.15	Rede VPH de reflexão simulada	57

2.16	Modelo tridimensional do iBTF.	58
2.17	Funcionamento de um CCD - Analogia dos baldes.	60
2.18	EMCCD - Conceito básico	61
2.19	Tela principal do KAFKA – <i>Operation Mode</i>	63
2.20	As três abas disponíveis no lado esquerdo do Modo de Operação do KAFKA.	64
3.1	Centro do campo do BTFI.	69
3.2	Montagem com laser utilizada para o alinhamento óptico do BTFI.	70
3.3	Distância nominal entre a lente de campo e a última superfície da roda secundária.	71
3.4	<i>Folding mirrors</i> em diferentes ângulos. (a) Localização aproximada dos <i>folding mirrors</i> . (b) Caixa que abriga os <i>folding mirrors</i> . (c) Parafusos de ajuste <i>push</i> , em verde, e <i>pull</i> , em azul	72
3.5	O colimador do BTFI.	73
3.6	Detalhe do sistema <i>push-pull</i> para alinhamento dos espelhos e redes de difração do iBTF. (a) Suporte de redes de difração e espelhos do iBTF (b) Detalhes dos parafusos que empurram e puxam um dos lados do suporte fazendo com que a superfície do espelho ou rede de difração se incline para um lado ou para outro.	74
3.7	Sistema <i>push-pull</i> no último espelho que dobra o feixe óptico logo após as últimas lentes imageadoras. Este sistema é idêntico em ambas as câmeras.	75
3.8	Espelho que dobra o caminho óptico que segue em direção à camera 2 (no centro da imagem). Na parte direita da imagem é possível ver a lente de campo da câmera 1 e, ao fundo, a terceira roda de filtros	75
3.9	Imagem do pinhole obtida usando o iBTF com espelhos no lugar das redes de difração. O FWHM é de 0,3 segundos de arco. (a) Imagem do pinhole na Câmera 1 (iBTF). (b) Perfil radial da imagem do pinhole medido com <i>imexam</i> - abscissa em número de píxeis e ordenada em número de contagens.	77

3.10 Imagem do pinhole obtida usando a Câmera 2 com o Fabry-Perot fora do caminho óptico. O FWHM é de 0,26 segundos de arco. (a) Imagem do pinhole na Câmera 2 (Fabry-Perot). (b) Perfil radial da imagem do pinhole medido com imexam - abscissa em número de píxeis e ordenada em número de contagens.	77
3.11 Largura à meia altura em função da posição relativa do M3.	80
3.12 Aglomerado IC2602 observado na configuração iBTF-MX.	81
3.13 Fabry-Perot sobre bancada.	83
3.14 Fabry-Perot não alinhado	83
3.15 Orientação dos eixos do FP	84
3.16 Fabry-Perot alinhado	85
3.17 <i>Bump</i> espectral no FP de baixa resolução	86
3.18 <i>Bump</i> espectral no FP de alta resolução	86
3.19 Análise do <i>bump</i> espectral em função do ângulo entre o feixe incidente e a superfície do filtro interferométrico. De cima para baixo temos os seguintes ângulos: sem inclinação nenhuma (0°), 1° , 2° , 5° e 10°	87
3.20 Espectro extraído de um cubo de dados obtido com o HR FP iluminado por uma lâmpada de Neônio e usando o filtro 6600.5A-19.3A próximo ao plano focal.	87
3.21 Espectro extraído de um cubo de dados obtido com o HR FP iluminado por um laser de HeNe sem a utilização de filtro algum.	88
3.22 Espectro HR FP com laser HeNe, sem filtro e com máscara na pupila. . . .	88
3.26 Cubo de calibração obtido com o HR FP dentro do SAM.	91
3.23 Espectros extraídos do pixel central de um cubo de dados obtidos com o HR FP usando um laser de HeNe, sem filtro interferencial e cobrindo 50% do feixe colimado em quatro diferentes direções. (a) Espectro obtido ao cobrir a metade superior do FP. (b) Espectro obtido ao cobrir a metade inferior FP. (c) Espectro obtido ao cobrir a metade à direita do FP. (d) Espectro obtido ao cobrir a metade à esquerda do FP.	94

3.24	Comparação entre os anéis usando o HR FP e o controlador UofH CS100 com loop aberto e loop fechado. Com o loop aberto, vemos somente um conjunto de anéis. Com o loop fechado, além do conjunto principal de anéis, vemos o conjunto secundário de anéis responsável pelo <i>bump</i> espectral. Os anéis mais fracos que aparecem de fundo são reflexos causados pelas lentes da câmera fotográfica. (a) Loop aberto. (b) Loop fechado.	95
3.25	HR-FP controlado pelo AAO1 CS100 - Loop fechado. Os anéis formam um único conjunto mostrando a ausência do <i>bump</i> espectral. O conjunto de anéis mais fraco que aparece completamente fora de centro com o conjunto principal é fruto de reflexos nas lentes da câmera fotográfica.	95
3.27	Rede Wasatch 1A - Eficiência espectral	98
3.28	Rede Wasatch 1B - Eficiência espectral	98
3.29	Rede Wasatch 2A - Eficiência espectral	98
3.30	Rede Wasatch 2B - Eficiência espectral	99
3.31	PSF no céu do iBTF usando o par Wasatch 1A e 1B.	100
3.32	PSF no céu do iBTF usando espelhos	101
3.33	Espectro extraído do píxel central do cubo obtido usando uma lâmpada de HgAr e com o iBTF em modo de transmissão.	102
3.34	Redes Ondax Mosaico - Fantasmas	104
3.35	Redes Ondax 16809 - Fantasmas	105
3.36	Ângulos dentro de uma VPH.	106
3.37	PSF usando pinhole no iBTF.	107
3.38	Rede Ondax 22709 - PSF	107
3.39	Interferograma - Rede clássica de difração	108
3.40	Interferograma - Rede Ondax 22709	108
3.41	Rede Ondax 24142 - PSF	109
3.42	Rede Ondax 24147 - PSF	109
3.43	Problema de ajuste no modo <i>double-pass</i>	110
3.44	Corte vertical nas imagens 3.43a (em preto) e 3.43b (em azul).	111
4.1	Estrutura de um cubo de dados com observações astronômicas: das três dimensões, duas delas são espaciais enquanto a terceira é a dimensão espectral	113

4.2	iBTF em diferentes configurações.	115
4.3	FP em diferentes configurações.	116
4.4	BIAS no modo clássico	118
4.5	BIAS no modo clássico recortado.	119
4.6	BIAS combinado no modo clássico.	119
4.7	BIAS no modo EM	120
4.8	DARK combinado 20 x 5 s no modo EM.	122
4.9	FLAT com a câmera 2	124
4.10	HR FP - Exemplo de espectro	126
4.11	Representação artística de como a informação relacionada a um ou mais comprimentos de onda se distribui em um cubo de dados obtido com um iBTF – adaptado de Quint (2010).	128
4.12	Representação artística de como a informação relacionada a um ou mais comprimentos de onda se distribui em um cubo de dados obtido com um Fabry-Perot. À esquerda, temos somente um comprimento de onda λ_1 . À direita, temos dois comprimentos de onda. O caráter periódico também é evidenciado nesta última figura – adaptado de Quint (2010).	128
4.13	Exemplo de extração de mapa de fase com o iBTF.	130
4.14	Exemplo de extração de mapa de fase com o iBTF usando cubo de correlação.	131
4.15	Mapas de fase extraídos do cubo obtido com o iBTF.	131
4.16	Exemplo de ajuste do mapa de fases com o iBTF.	132
4.17	Resultado após correção de fase	133
4.18	Efeitos causados pela correção de fase em um cubo do iBTF.	134
4.19	Encontrando o FSR para dados do FP	135
4.20	Encontrando o centro dos anéis do FP.	136
4.21	Perfil radial do mapa de fase de um FP.	137
4.22	Mapas de fase extraído do cubo obtido com o FP.	137
4.23	Resultado da correção de fase de um cubo de FP.	138
4.24	Espectro da linha de Ne usando HR FP e filtro 6600/20	142
4.25	Espectro da linha de Ne usando HR FP e filtro 6745/40	143
4.26	Gráfico para encontrar as ordens de interferência.	145
4.27	NGC2440 observado com o iBTF.	153

4.28	Mapa de estrutura de NGC2440.	154
4.29	NGC7009 observada com o LR FP.	155
4.30	30Dor observada com o HR FP.	155
5.1	30 Dor observado com o HR FP com o SAM usando filtro 6563/20 2x2".	160
A.1	Desenho esquemático do painel frontal do controlador CS100 extraído da página do telescópio CFHT, em outubro de 2014.	172

Lista de Tabelas

1.1	Parâmetros instrumentais de interferômetros FP.	32
2.1	Resumo das prescrições do sistema óptico do BTFI (Unidades em mm). . .	43
2.2	BTFI - Principais parâmetros ópticos	44
2.3	Parâmetros nominais para o LowRes-FP	45
2.4	Parâmetros nominais para o HighRes FP	45
2.5	Filtros interferenciais disponíveis.	49
2.6	EMCCDs no BTFI - Informações técnicas.	62
3.1	Largura à meia altura medida no processo de busca pela posição do foco relativo do M3	79
3.2	Valores usados na estimativa dos tempos gastos no controle do FP.	92
3.3	Redes Wasatch - Parâmetros espectrais encontrados.	99
3.4	Comparação na FWHM da imagem do pinhole obtida na Câmera 1.	101
4.1	Níveis de DARK medidos em cada uma das câmeras.	122
4.2	Estimativas dos níveis de ruído de DARK.	122
4.3	Tabela usada para encontrar as ordens de interferência.	145
4.4	HR FP - Parâmetros experimentais	149
4.5	LR FP - Parâmetros experimentais	149
4.6	Orientação e Ângulo de Posição	150
4.7	Orientação das imagens em função do ângulo de posição do telescópio. . . .	151
A.1	Configuração a ser ajustada no CS100 ao usar o FP R20000	173

Introdução

Há pouco mais de duas décadas, o Brasil avançava na astronomia nacional ao entrar em um consórcio internacional para a construção de um novo telescópio. Naquela época, a comunidade astronômica brasileira já tinha acesso aos telescópios Perkin-Elmer de 1,60 m, em operação desde 1981; Boller & Chivens de 0,60 m, em operação desde 1992; e Zeiss de 0,60 m, que foi obtido através de um acordo entre o Brasil e a ex-Alemanha Oriental em troca de café e em operação desde 1983. Todos os três telescópios estão localizados no Observatório do Pico dos Dias (OPD), próximo à cidade de Brazópolis - MG e estão em operação desde abril de 1980¹.

Este consórcio resultou na construção do telescópio SOAR cuja sigla, em inglês, significa Observatório Astronômico de Pesquisas do Hemisfério Sul — ***SO**outhern **A**strophysical **R**esearch Telescope*. Os parceiros envolvidos neste consórcio são o Brasil, representado pelo LNA/MCTI (Laboratório Nacional de Astrofísica/Ministério da Ciência, Tecnologia e Inovação), o *National Optical Astronomy Observatory* (NOAO), a Universidade da Carolina do Norte (UNC) e a Universidade Estadual de Michigan (MSU). O Chile também consta como parceiro por ceder parte do seu território para a construção deste telescópio.

Usado pela primeira vez em noites de observação astronômica em março de 2004 (Schwarz et al., 2004) e inaugurado oficialmente no dia 17 de abril deste mesmo ano, o SOAR está situado em *Cerro Pachón*, uma montanha dos Andes Chilenos com altitude de 2.700 metros acima do nível do mar.

Um dos compromissos assumidos pelo Brasil dentro do consórcio para a construção

¹ <http://www.lna.br/opd/opd.html>

do telescópio SOAR foi o desenvolvimento de dois novos instrumentos para serem usados neste telescópio. Ou seja, entrar para o ramo de instrumentação astronômica foi muito mais do que aproveitar uma oportunidade, foi honrar um compromisso. E, assim, dois novos instrumentos começaram a ser desenvolvidos em território nacional.

O primeiro deles é chamado SIFS – SOAR Integral-Field Unit Spectrograph (Lepine et al., 2003), desenvolvido pelo Laboratório Nacional de Astrofísica (LNA), localizado em Itajubá, Minas Gerais. Em resumo, este instrumento fornecerá 1300 espectros através de uma matriz contendo 26 x 50 microlentes. Cada microlente irá amostrar 0,30 segundos de arco, totalizando um campo de 8 x 15 segundos de arco. Estas microlentes alimentam um cabo de fibras ópticas com 13 metros de extensão que levam a luz recebida do telescópio a um espectrógrafo de bancada.

A luz recebida pelo espectrógrafo sofre difração ao passar por uma das redes de difração holográficas². Esta rede está disponível em um conjunto interno ao instrumento e é selecionada através de um braço robótico de acordo com região e a resolução espectral desejadas. O instrumento já foi construído e encontra-se atualmente instalado no SOAR. Entretanto, devido às baixas temperaturas em que o instrumento tem que operar, o cimento óptico do triplete da câmera rompeu-se. A câmera foi enviada de volta para o fabricante para ser cimentada novamente em uma temperatura semelhante a que temos no Cerro Pachón, para que as variações térmicas não tenham efeitos tão grandes. Antes de voltar para dentro do instrumento, a cola que une as lentes do triplete precisa ser testada em ambientes simulados em laboratório para que este tipo de acidente não ocorra novamente.

O segundo é o *SOAR Telescope Echelle Spectrograph* ou STELES (Castilho et al., 2004), também desenvolvido no LNA. Este instrumento possui dois canais espectrais, um vermelho e outro azul, separados por uma rede de difração holográfica. Cada canal utiliza uma rede de difração do tipo *echelle* (ver o artigo do instrumento para mais detalhes a respeito deste tipo de rede de difração) para fornecer espectros com altíssima resolução espectral ($R = 50.000$) no intervalo espectral de 3000 à 5500 Å no braço azul e de 5300 à 8900 Å no braço vermelho. A abertura da fenda cobrirá 0,8 segundos de arco do céu.

Além do SIFS e do STELES, a proposta de construção de um novo instrumento foi realizada no final de 2006 para atender a demanda da comunidade científica brasileira em ter, junto ao telescópio SOAR, um instrumento que pudesse ser usado para casos científicos

² Para uma descrição sobre o que é uma rede de difração holográfica § 2.3.1

que exigem boa resolução angular e espectral em um campo considerável de 3 x 3 minutos de arco, como núcleos ativos de galáxias, interação entre galáxias, cinemática e composição de nebulosas planetárias e meio interestelar.

Este novo instrumento, chamado BTFI — *Brazilian Tunable Filter Imager* ou Imageador com Filtro Ajustável Brasileiro (Mendes Mendes de Oliveira et al., 2013), foi desenhado e construído para trabalhar junto a uma das saídas no foco Nasmyth do telescópio ou sobre o módulo de óptica adaptativa do SOAR — *SOAR Adaptive Module*, ou SAM (Tokovinin et al., 2008).

1.1 BTFI como um novo instrumento para o SOAR

Por ser um imageador, o BTFI, a priori, nos fornece dados em formato matricial com duas dimensões. Cada elemento destas matrizes, chamado pixel (px), contém informação sobre a quantidade de luz que provém de um determinado ponto no céu. Chamamos estas matrizes de imagens e, com elas, obtemos a distribuição angular de um determinado objeto dentro do campo de visão do telescópio. Como os detectores usados em astronomia para a obtenção de imagens não armazenam informação de cor, utilizamos filtros espectrais para obter a distribuição angular de um determinado objeto em uma determinada banda espectral. Obtendo imagens em diferentes comprimentos de onda, podemos, por exemplo, estimar a temperatura de uma estrela, estimar a relação massa-luminosidade de uma galáxia (Bell e de Jong, 2000) ou classificar uma galáxia em *early type* ou *late type* (Bruzual e Charlot, 2003).

Quando o BTFI é acoplado diretamente ao SOAR, fornece imagens cuja resolução angular é delimitada pelos efeitos de turbulência atmosférica. Em astronomia, dizemos que a resolução angular é limitada pelo *seeing*, ou a largura à meia altura do perfil radial de uma fonte pontual medida em segundos de arco. No caso, o *seeing* típico obtido no telescópio SOAR pode variar entre 0,6 e 2,0 segundos de arco, dependendo das condições atmosféricas. Obviamente, numa noite com muito mau tempo, estes valores podem ser ainda maiores.

Ao ser usado em conjunto com o SAM, o BTFI pode fornecer imagens com resolução angular entre 0,3 e 0,5 segundos de arco em noites com boas condições climáticas. Como o SAM possui um campo de 3 minutos de arco, o BTFI foi desenhado para obter imagens com

um mesmo campo e com uma amostragem de 0,12 segundos de arco. O campo pequeno e a alta resolução foram também motivados pela complementaridade com o instrumento similar a ser montado no ESO (*European Southern Observatory*) NTT (*New Technology Telescope*), o 3DNNTT, que tem um campo maior de 17 minutos de arco.

As imagens do BTFI são obtidas através de um novo tipo de detector chamado EMCCD (*Electron Multiplier Coupled-Charge Device*). Na prática, a principal diferença entre este tipo de detector e os CCDs (*Coupled-Charge Device*) reside no fato dos EMCCDs fornecerem imagens com um nível de ruído térmico muito mais baixo do que estes últimos. O nível de ruído eletrônico é também muito mais baixo do que os de CCDs comuns, mesmo operando com altas taxas de leitura. Os EMCCDs também são muito mais sensíveis que os CCDs clássicos, o que faz com que o tempo de exposição gasto para obter um bom sinal proveniente de um determinado astro seja muito menor.

Por outro lado, os EMCCDs possuem um custo muito mais elevado e necessitam de um controlador eletrônico muito mais complexo. O ajuste fino do nível de ganho é um dos outros fatores que dificultam a aplicação deste tipo de detector. Outro problema associado ao uso dos EMCCDs, é que poucos grupos o utilizam e, com isso, fica mais difícil encontrar referências para ajuste dos controladores deste tipo de detector, para técnicas de observação e para a redução dos dados. Além disso, por serem mais sensíveis, podem ser danificados se expostos a fontes luminosas intensas.

Aliando a habilidade que o BTFI tem de selecionar o comprimento de onda que provém do objeto de estudo de modo contínuo, o que justifica o termo “filtro ajustável” utilizado em seu nome, com a capacidade que os EMCCDs possuem de fornecer imagens rapidamente, é possível fazer uma varredura no domínio espectral em cima de um determinado objeto astronômico. Em outras palavras, cada imagem desta varredura é obtida em um comprimento de onda diferente.

Organizando as imagens obtidas nesta varredura espectral, é possível colocar todas elas em um único arquivo chamado cubo de dados (*data-cubes*). Esta técnica também é conhecida como espectroscopia 3D e o termo para varredura que costumamos usar é *scan*.

Para obter dados com diferentes resoluções espectrais, o BTFI conta com dois módulos: o primeiro deles é chamado de *imaging Bragg Tunable Filter* (iBTF) ou *double Bragg Tunable Filter* (dBTF). Este módulo faz uso de pares de redes de difração holográficas para selecionar o comprimento de onda da luz que chega em uma determinada região do detec-

tor. Dependendo das redes utilizadas, podemos obter cubos com baixa resolução espectral entre 5 e 20 dentro do intervalo espectral entre 4000 Å e 6500 Å aproximadamente, ou cubos com resolução espectral intermediária entre 1000 e 2000 dentro de um intervalo espectral de aproximadamente 500 Å ao redor de $H\alpha$. O conceito do iBTF foi apresentado pela primeira vez em Blais-Ouellette et al. (2004) e, hoje, é patenteado por uma empresa canadense chamada *Photon Etc.*³. Seguindo informações no próprio site, o iBTF é utilizado em diversas áreas como mineralogia, operações com óleos e gases, dermatologia e em agricultura. Entretanto, é a primeira vez que um iBTF é utilizado para fins astronômicos. Daremos mais detalhes sobre este módulo na seção 2.3 do capítulo 2.

O segundo módulo utilizado dentro do BTFI para a realização de varreduras espectrais é chamada de *Fabry-Perot* (FP), um interferômetro inventado em 1897 pelos franceses Charles Fabry e Alfred Perot (Mulligan, 1998). Entretanto, mesmo sendo um interferômetro cuja física é tão bem conhecida, somente a partir do início da década de 80 que o FP passou a ser utilizado como filtro ajustável para espectroscopia 3D.

O primeiro FP usado para aquisição de imagens com varredura em comprimento de onda foi montado no instrumento TAURUS (Taylor e Atherton, 1980; Atherton et al., 1982), desenvolvido para trabalhar no *Anglo-Australian Telescope* (AAT). TAURUS esteve em operação entre 1981 e 1983. Posteriormente, este instrumento foi redesenhado e uma nova versão chamada *Taurus-II Tunable Filter* ou TTF foi utilizada de 1996 até 2003 no AAT. Uma réplica deste instrumento, também chamada TAURUS, foi utilizada no *William Herschel Telescope* (WHT), na Espanha, entre 1996 e 2000.

O segundo dispositivo de aquisição de imagens que fez uso de um FP em seu conceito instrumental foi o instrumento francês CIGALE (*CINematics of GALaxiEs* – Boulesteix et al. (1984)). Construído em Marselha, no sul da França, este instrumento foi utilizado em diversos telescópios: no *Canada-France-Hawaii Telescope* (CFHT); no telescópio Byurakian de 2,6 metros, localizado na Armênia; no telescópio russo Zelenchuk de 6,0 metros e no telescópio ESO de 3,6 metros de La Silla, Chile. Este instrumento era composto por um redutor focal, um Fabry-Perot, e um sistema de imageamento de contagens de fótons (*Image Photon Counting System* - IPCS). O IPCS permitia obter imagens com resolução temporal de até 1/50 segundos e com baixíssimo ruído de leitura. Com isso, era possível realizar uma sequência de várias varreduras em comprimento de onda com o FP para mi-

³ <http://www.photonetc.com/>

nimizar possíveis efeitos de variações na imagem devido a efeitos atmosféricos, massa de ar e transparência do céu. Esta técnica é utilizada ainda hoje.

Outros exemplos de instrumentos que utilizam ou utilizaram Fabry-Perot são: HIFI (*Hawaii Imaging Fabry-Perot Interferometer* - Bland e Tully (1989)), usado tanto no telescópio CFHT quanto no telescópio de 2,2 metros da Universidade do Havaí; PALILA (Boulesteix e Grundseth, 1987), construído também em Marselha, usado no CFHT entre 1990 e 1994 até ser doado para o Observatório do Monte Mégantic, no Canadá; o MOS-FP/MOS-SIS (Crampton et al., 1992), que substituiu o PALILA em 1994 e foi utilizado no CFHT até 2006; o GriF (Clenet et al., 2000), desenhado para trabalhar também no CFHT na região do infra-vermelho; o GH α SP (*Gassendi H α survey of SPirals* - Garrido et al. (2002)), usado até hoje no foco Cassegrain do telescópio de 1,93 metros no Observatório de *Haute-Provence*, no sul da França; o GH α Fas (*Galaxy H α Fabry-Perot System* - Cargnan et al. (2008)), disponível desde 2007 no telescópio de 4,2 metros do WHT. Outros instrumentos que fazem uso de Fabry Perot são o FaNTOmM (Hernandez et al., 2003), o PUMA (Rosado et al., 1995), SCORPIO (Afanasiev e Moiseev, 2005), MMTF (Veilleux et al., 2010), RSS (Rangwala et al., 2008), OSIRIS (Cepa et al., 2000) e o TFI (Ingraham et al., 2010). Um resumo das principais propriedades destes instrumentos é apresentado na tabela 1.1, extraída de Mendes Mendes de Oliveira et al. (2013).

Instrumento	Situação	Telescópio	Intervalo Espectral [Å]	Resolução Espectral [-]	Campo [min de arco]	Amostragem [seg de arco]
TAURUS	Fora de operação	AAT (3,89m), WHT (4,20m)	3700 – 9500	100 – 60000	9,87	0,37
HIFI	Fora de operação	CFHT (3,58m), UH (2,2m)	4000 – 7500	4000 – 16000	10	0,43 – 0,69
CIGALE	Fora de operação	ESO (3,60m)	6563 – 6782	15000	5,0	0,45
PALILA	Fora de operação	CFHT (3,58m)	6563 – 6782	15000	5,8	0,34
MOS-FP	Fora de operação	CFHT (3,58m)	3650 – 10000	5000 – 15000	10	0,8
GriF	Fora de operação	CFHT (3,58m)	Bandas H e K	2000	0,60	0,12
GHASP	Em operação	OHP (1,92m)	6563 – 6782	15000	5,8	0,68
CHaFaS	Em operação	WHT (4,2m)	6563 – 6782	5000 – 15000	4,0	0,45
FaNTOmM	Em operação	Mégantic (1,60m)	6560 – 6780	5000 – 15000	19,4	1,61
PUMA	Em operação	San Pedro (2,1m)	3650 – 8650	10650	10	0,67
SCORPIO	Em operação	SAO (6,0m)	5000 – 9000	3000 – 10000	6,1	0,40
MMTF	Em operação	Magellan (6,5m)	5000 – 9200	200 - 1840	27 / 10	0,60
RSS/FP	Em operação	SALT (11m)	4300 – 8600	300 – 9000	8	0,25
OSIRIS	Futuro	GRANTECAN (10,4m)	3650 – 10500	300 – 5000	7,8	0,13
TFI	Futuro	JWST (6,5m)	1500 – 5000	75 – 120	2,2	0,60

Tabela 1.1 - Parâmetros instrumentais de interferômetros FP.

O BTFI foi desenhado para trabalhar usando dois Fabry-Perots para fornecer cubos de dados com alta resolução, entre 6000 e 25000. O primeiro é posicionado logo após o plano focal do telescópio, no feixe divergente. O segundo é posicionado no espaço colimado,

próximo à pupila do instrumento. Entretanto, a fabricação destes dois FPs ainda não foi concluída. Até que estes FPs estejam prontos, o BTFI está usando alternadamente dois FPs antigos obtidos com colaboradores: um com resolução espectral nominal de 4500 e outro com resolução espectral nominal de 20000. Por hora, chamaremos o primeiro de *Low-Resolution Fabry-Perot* (LR FP) e o segundo de *High-Resolution Fabry-Perot* (HR FP).

A construção do BTFI foi liderada pela Profa. Dra. Cláudia Lucia Mendes de Oliveira, professora do Instituto de Astronomia, Geofísica e Ciências Atmosféricas (IAG) da Universidade de São Paulo (USP) e pelo Prof. Dr. Keith Taylor, instrumentalista e gerente de vários projetos, incluindo o próprio TAURUS. Os demais parceiros neste projeto estão, em sua maioria, também vinculados ao IAG/USP. Entretanto, o projeto só pôde ser executado com colaboração do Instituto Nacional de Pesquisas Espaciais (INPE) e de René Laporte, responsável pelo desenho mecânico e pela montagem do instrumento. Outros parceiros nacionais foram a Universidade Estadual de Santa Cruz (UESC), em Ilhéus - BA; a Universidade Federal do Rio Grande, localizada na cidade de Rio Grande - RS; a Universidade Federal de Santa Catarina, em Florianópolis - SC e o LNA (este último financiou cerca de 10% do projeto). Houve colaboração também entre outros institutos dentro da própria USP, como foi o caso da Escola Politécnica e do Instituto de Física. O BTFI também contou com colaborações internacionais sem as quais não teria sido construído, o *Laboratoire d'Astrophisique du Marseille*, na França, e a *Université du Montréal*, no Canadá. A contribuição do LAM foi fundamental uma vez que eles estão construindo o instrumento 3DNTT e o desenvolvimento dos novos FPs e do sistema de controle destes FPs estão sendo feitos por eles.

Atualmente, o BTFI possui um status de Restricted Use Instrument (RUI). Isto significa que o instrumento pertence ao Brasil e, a priori, estará disponível, quando estiver funcional, somente para a comunidade astronômica brasileira. Dependendo do seu desempenho, o BTFI poderá ser transformado em *Facility Instruments*, quando o SOAR passará também a ter responsabilidades sobre ele. Outros *Facility Instruments* instalados no telescópio SOAR são os espectrógrafos GOODMAN e OSIRIS e os imageadores SPARTAN e SOI.

Nos últimos anos temos realizado testes para avaliar diversos pontos do funcionamento do BTFI. Os testes foram feitos tanto em laboratório quanto em noites de observação. O capítulo 3 desta tese inclui uma compilação dos resultados obtidos nestes testes e uma

avaliação do que pode e deve ainda ser melhorado no BTFI.

1.2 *Objetivos e estrutura desta tese*

O trabalho apresentado aqui é uma continuação do trabalho que apresentamos na dissertação de mestrado quando o BTFI ainda não estava operacional e onde fizemos a modelagem de redes de difração holográficas utilizando o modelo de Kogelnik (1969) e modelagem de cubos de dados obtidos com Fabry-Pérot com diferentes fontes (Quint et al., 2010).

Conforme já mencionamos, o objetivo final desta tese é realizar uma avaliação do estado atual do BTFI levando em consideração todo o seu processo de construção e aprimoramento na parte opto-mecânica nos últimos anos. Além disto, mostraremos que tipo de estudos científicos poderão ser realizados com o instrumento levando em consideração suas características atuais. Esta caracterização consiste em uma análise da qualidade das imagens e dos espectros obtidos com o BTFI nos seus diversos modos de operação. Além disso, faz parte desta caracterização medir a eficiência operacional dos sistemas em cada um dos modos de operação do instrumento.

No capítulo a seguir, faremos uma revisão teórica dos fundamentos físicos necessários para a melhor compreensão do instrumento, uma revisão bibliográfica para apresentar o que já se conhece em relação a cada módulo que compõe o BTFI e uma revisão dos requisitos básicos determinados no projeto inicial deste instrumento.

Apresentaremos no terceiro capítulo o trabalho que realizamos em laboratório e no céu para estudar o comportamento do instrumento focando na sua parte opto-mecânica nos últimos anos. Além do desempenho do sistema de controle e a sua eficiência operacional. Apresentaremos também as características dos diversos elementos ópticos bem como suas peculiaridades e limitações práticas juntamente com uma descrição dos nossos esforços na tentativa de aprimorar o desempenho do BTFI como um todo, especialmente na parte em que envolve a utilização dos Fabry-Pérot e do iBTF.

O processo de caracterização descrito aqui será limitado aos testes opto-mecânicos do sistema, com foco no iBTF, justamente pelo fato de ser um sistema novo dentro do contexto da astronomia, e no Fabry-Pérot. Além de nos fornecer informações cruciais sobre o funcionamento do instrumento, o processo de caracterização ajuda a entender o

que precisa ser melhorado e quais problemas precisam ser eliminados.

Após a caracterização do instrumento, utilizaremos as bibliotecas computacionais que estão sendo desenvolvidas pela equipe que integra o BTFI para remover as diversas formas de ruídos existentes e realizar a correção de fase (seção 4.6) dos cubos de dados. A compreensão plena do funcionamento do instrumento aliada à experiência em programação e redução de dados ajudará a analisar a eficiência das novas rotinas de modo a aprimorá-las e torná-las robustas e de fácil uso para o usuário final.

No quarto capítulo apresentaremos resultados obtidos em observações com o BTFI durante diversas jornadas de testes chamadas de **comissionamentos**. Apresentaremos também todo o procedimento necessário para tratar as imagens obtidas e quais tipos de dados de calibração devem ser tomados antes e depois de uma noite de observação. Além disso, daremos uma explicação do procedimento necessário para unificar estas imagens em um cubo de dados, como reordenar a informação espectral dentro de cada cubo e, finalmente, como realizar a calibração em comprimento de onda de modo que os dados possam ser utilizados para análises científicas.

Por fim, terminaremos este trabalho com o capítulo 5 apresentando nossas conclusões. Faremos ademais uma compilação de todas as limitações atuais do instrumento bem como propostas de melhorias e propostas de casos científicos em que ele já pode ser utilizado com as características e limitações atuais.

Descrição

O BTFI foi montado pela primeira vez no fim do primeiro semestre de 2010 nas dependências do Instituto de Astronomia, Geofísica e Ciências Atmosféricas (IAG), na Universidade de São Paulo (USP), onde foram feitos os primeiros testes funcionais com o equipamento. Em junho de 2010, ele foi completamente desmontado, enviado para o Chile e montado novamente já nas dependências do observatório SOAR sobre um carro construído especialmente para facilitar o armazenamento e o transporte do instrumento.

Sobre este carro, a posição natural do instrumento é na vertical, tomando como referência o eixo óptico na entrada do instrumento. A figura 2.1 mostra a parte frontal do BTFI montado sobre o carro de transporte e posicionado na vertical. A caixa proeminente na parte inferior do instrumento é a caixa eletrônica (flexa vermelha), onde chegam os cabos que alimentam o instrumento e onde ficam os disjuntores e servo-terminais que transformam sinais *Ethernet* em sinais serial. Na parte superior do instrumento, a tampa que o cobre foi removida, o que nos permite ver a chamada Câmera 1 do BTFI (marcada com um círculo amarelo), usada quando o instrumento opera com o iBTF. Mais ao centro, a tampa também foi removida, dando acesso à parte de trás do iBTF (destacado com um quadrado azul claro).

Quando instalado no telescópio, o instrumento é posicionado na horizontal, novamente tomando como referência a direção do eixo óptico na sua entrada. Neste caso, a caixa eletrônica é posicionada na parte superior do instrumento e a Câmera 1 na parte de trás dele.

Um outro ângulo do BTFI é mostrado na figura 2.2. Nela, com o instrumento também

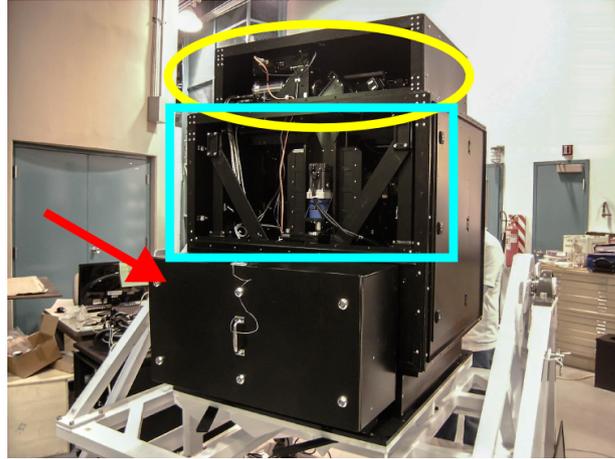


Figura 2.1: BTFI sobre o carro de transporte – visão frontal do instrumento.

na posição vertical, podemos ver a caixa proeminente na parte superior onde temos acesso à Câmera 1 e a caixa proeminente na região posterior, onde temos acesso à Câmera 2.

Nesta posição, medindo a partir da placa que conecta o BTFI ao telescópio, o instrumento tem, aproximadamente, 1,50 m de altura, 1,00 m de largura e 1,20 m de profundidade. A massa medida durante a instalação foi de 333,39 kg.

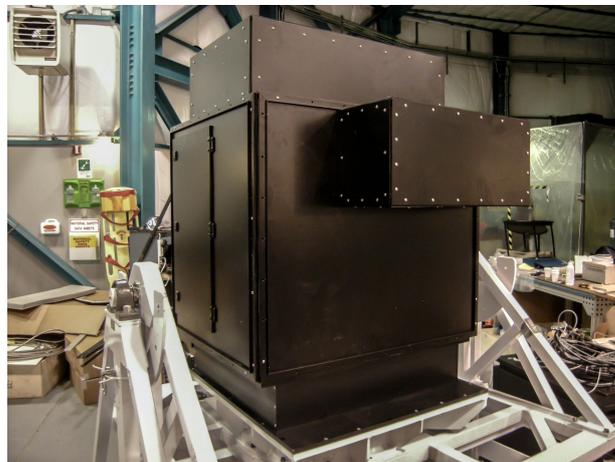


Figura 2.2: BTFI sobre o carro de transporte – visão traseira do instrumento.

Imaginamos agora o instrumento posicionado na horizontal, a luz vindo do telescópio da esquerda para a direita e formando uma imagem do céu no plano focal do instrumento, representado por **Im** na figura 2.3, com uma razão focal $F/\# = 16,625$. Logo em seguida, a luz volta a divergir e encontra a lente de campo **FL**, um dubleto com 96 mm de diâmetro, e o primeiro Fabry-Perot que, por estar próximo à imagem formada pelo telescópio, é

chamado de FP_{Im} . Este FP é utilizado como um seletor de ordens de interferência para o segundo FP que será descrito posteriormente. Quando este não está em uso, é substituído por uma placa de vidro de 50 mm de espessura para manter o foco instrumental.

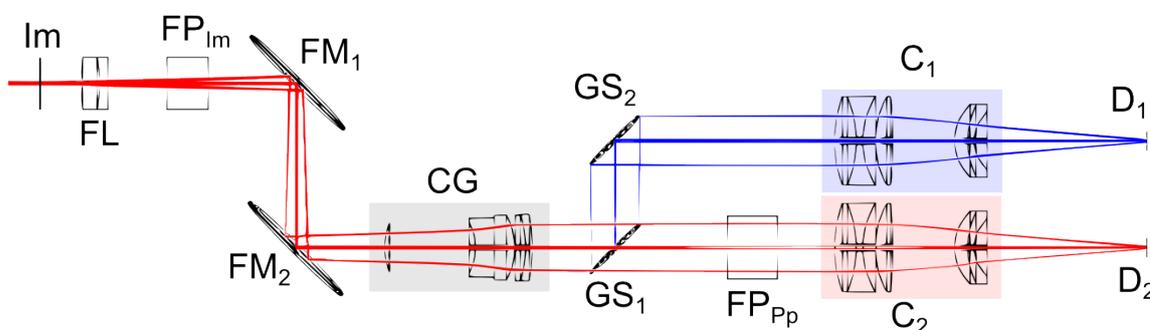


Figura 2.3: Desenho óptico simplificado do BTFI.

Depois deste primeiro FP, o caminho óptico é “dobrado” em 90° para poder cumprir as restrições de espaço disponíveis para o instrumento. Para isto, são usados dois espelhos FM_1 e FM_2 – *Folding Mirror #1* e *Folding Mirror #2*. Em seguida, a luz encontra um grupo de lentes que compõem o colimador, CG – *Colimator Group*.

Agora, no espaço colimado é onde estão o iBTF e o FP de alta resolução espectral. Conforme já explicado anteriormente, toda a luz colimada incide sobre a primeira rede de difração holográfica GS_1 . Parte desta luz sofre difração (em azul), é direcionada para a segunda rede de difração holográfica GS_2 e é re-imageada pela câmera 1, C_1 , sobre o detector D_1 . O funcionamento das redes GS_1 e GS_2 será apresentado na seção 2.3.1.

A outra porção, em vermelho, segue reto no caminho óptico sem sofrer nenhum efeito da rede GS_1 (ordem zero de difração) e encontra o segundo Fabry-Perot, localizado próximo à pupila do instrumento – FP_{Pp} . Os detalhes de como o FP funciona e quais as diferenças entre usar o FP no feixe divergente ou próximo à pupila serão explicados posteriormente ainda neste capítulo, na seção 2.2. Logo após o FP, a luz, então, encontra a câmera 2, C_2 , que forma uma imagem sobre o detector D_2 .

O BTFI também possui três rodas de filtros. A primeira delas localiza-se exatamente sobre o plano focal do instrumento que, por sua vez, coincide com o plano focal do telescópio. Esta roda de filtros possui espaço para alocar quatro filtros. Por estar no plano focal do instrumento, ela recebe máscaras usadas para centragem de campo e para ajustes de foco instrumental. Por este motivo, a chamamos de *Mask Wheel* ou roda de

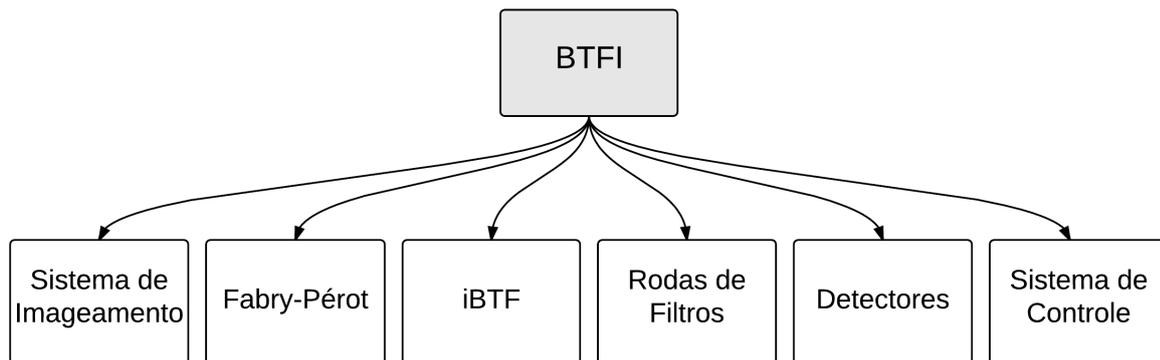


Figura 2.4: Módulos internos ao BTFI.

máscaras. Ela também é, algumas vezes, chamada de *Primary Wheel*, roda primária, ou de *4-Position Wheel*, roda de quatro posições.

Após a roda primária está a chamada *Image Wheel*, com espaço disponível para alocar oito filtros. Por estar no feixe divergente, esta roda foi construída para alocar filtros espectrais de banda larga. É comum chamarmos esta roda de *Secondary Wheel* ou *8-Position Wheel*.

A terceira roda de filtros é chamada de *Pupil Wheel*, por estar próxima à pupila do instrumento, logo após o FP_{Pp} . Com espaço disponível para alocar seis filtros, esta é a roda que recebe filtros interferenciais usados para selecionar uma pequena região do espectro em que o FP irá atuar. Os filtros são instalados nesta roda através de adaptadores que permitem que eles sejam inclinados em relação ao eixo óptico e, com isso, variando o comprimento de onda central transmitido em alguns ângstroms.

Devido a sua complexidade, precisamos ter claro como o processo de caracterização deve ser realizado. Para tal, dividimos o instrumento em diferentes módulos de acordo com o diagrama mostrado na figura 2.4.

Em tese, cada componente dentro do BTFI deveria ter sido testado individualmente mas tivemos que lidar com a falta de equipamento disponível na época de cada teste. Sendo assim, nos limitamos em avaliar o sistema de imageamento simples, o Fabry-Perot e o iBTF em quatro pontos principais:

- Qualidade da imagem;
- Qualidade espectral;

- Níveis e fontes de ruído;
- Eficiência operacional.

Neste capítulo, nos focaremos em descrever cada um dos itens apresentados no diagrama da figura 2.4. No próximo capítulo, apresentaremos os métodos utilizados para avaliá-los e, também, apresentaremos os resultados dos testes realizados.

2.1 Sistema de imageamento

O projeto do desenho óptico do BTFI foi feito pela empresa australiana *Prime Optics*¹ utilizando o software *Zemax*². A figura 2.5 mostra o desenho óptico original considerando somente as superfícies ópticas existentes a partir do plano focal do telescópio/instrumento.

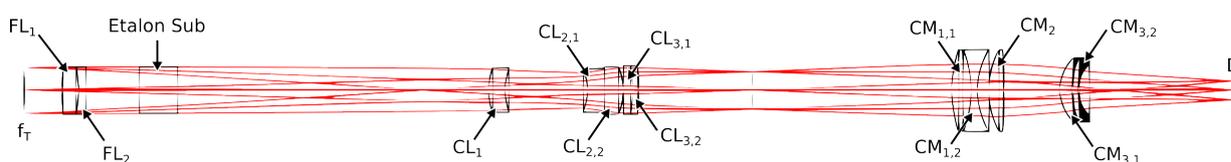


Figura 2.5: Desenho óptico original do BTFI

Neste desenho, estão representadas somente as lentes existentes dentro do instrumento a partir do plano focal do telescópio. A notação utilizada nele segue a notação utilizada pela *Prime Optics* no documento original que contém toda a informação óptica³. FL_1 e FL_2 são as lentes de campo (*Field Lens*); CL_1 , $CL_{2,1}$, $CL_{2,2}$, $CL_{3,1}$ e $CL_{3,2}$ são as lentes que compõem o colimador (*Collimator Lenses*) e $CM_{1,1}$, $CM_{1,2}$, CM_2 , $CM_{3,1}$ e $CM_{3,2}$ são as lentes que compõem as lentes da câmera imageadora (*CaMera lenses*). Os três conjuntos de raios representam os raios que provêm do centro e dos dois extremos do campo. De acordo com o documento original, o instrumento BTFI deve ter um meio campo de observação (*half field-of-view*) de 0.025° ou 1,5 minutos de arco (campo total de 3 minutos de arco). Este desenho se mantém tanto no caso em que o BTFI é usado com o módulo de óptica adaptativa do SOAR quanto quando acoplado diretamente ao eixo Nasmyth.

A tabela 2.1 contém os principais parâmetros ópticos para a fabricação destes elementos. Nela, a primeira coluna indica qual a identificação da superfície. A segunda, RoC , é o raio

¹ <http://www.primeoptics.com.au/>

² <http://www.radiantzemax.com>

³ <http://btfi.iag.usp.br/>

de curvatura da superfície em milímetros. Valores positivos indicam que o centro do raio de curvatura na figura 2.5 está à esquerda da superfície (ou que aquela superfície é côncava) e valores negativos, à direita (ou que aquela superfície é convexa). A terceira coluna indica a distância entre a superfície atual e a próxima superfície. A quarta coluna contém o material entre a superfície atual e a próxima superfície. A última coluna contém o diâmetro das superfícies em milímetros.

A identificação da superfície normalmente segue a notação “ID_{X,Y,Z}”, onde “ID”, contendo duas letras, denomina qual é o grupo do caminho óptico correspondente (CL para colimador, por exemplo), “X” indica o número do componente óptico pertencente a um determinado grupo, “Y” indica qual o elemento dentro deste componente e “Z” indica qual a superfície relacionada a este elemento. Por exemplo “CM3,1,1” indica a primeira superfície da primeira lente do terceiro grupo de lentes da câmera óptica.

Neste desenho, o Fabry-Perot presente no feixe divergente, logo após as lentes de campo pode ser substituído por uma placa de vidro chamada de *dummy plate*. O único papel desta placa é manter a distância óptica entre as lentes de campo e as lentes do colimador caso o primeiro FP não seja utilizado. Caso ela não esteja presente, a posição do foco instrumental é alterada.

Este desenho não leva em consideração o fato de haver dois espelhos que dobram o caminho óptico na região entre o primeiro FP e o colimador nem os espelhos que dobram o caminho óptico após a última lente da câmera imageadora e antes do detector.

O sistema de imageamento do BTFI consiste basicamente em um redutor focal com uma razão focal de entrada de $(F/\#)_{In} = 16,625$ e razão focal de saída de $(F/\#)_{Out} = 7,10$, de acordo com os relatórios do desenho óptico gerado pelo Zemax do desenho óptico. Os demais parâmetros ópticos do sistema como um todo são apresentados na tabela 2.2.

2.2 Fabry-Perot

O BTFI foi desenhado para trabalhar com dois Fabry-Perot (FP), dispositivos ópticos compostos por duas placas de vidro cilíndricas e concêntricas, chamadas de *étalon*. Estas duas placas são dispostas a uma pequena distância uma da outra formando uma cavidade óptica chamada de *gap*. Elas são posicionadas de modo que o eixo de simetria delas coincida com o eixo óptico do sistema. A luz que incide sobre elas atravessa a primeira placa de

Tabela 2.1 - Resumo das prescrições do sistema óptico do BTFI
(Unidades em mm).

I.D.	RoC	Espessura	Material	Diâmetro
Foco NASMYTH do SOAR	-966.959	0		86
Foco AO do SOAR	796.850	0		86
Até a lente de campo		50		
FL1,1	332.5536	22	S-FPL51Y	96
FL1,2/FL2,1	-162.9689	10	N-BAK2	96
FL2,2	∞	71		96
ETALON 1	∞	50	SILICA	96
ETALON 2	∞	412.977		96
CL1,1	82.754	22	N-BAK2	74
CL1,2	79.321	110.023		62
CL2,1,1	-61.017	22	N-BAK2	56
CL2,1,2/CL2,2,1	1373.475	25	S-FPL51Y	74
CL2,2,2	-82.754	1		74
CL3,1,1	-865.097	12	S-FPL51Y	74
CL3,1,2/CL3,2,1	-138.202	8	SILICA	74
CL3,2,2	-228.686	150		74
Pupila	∞	265		50
CM1,1,1	164.681	26	S-FPL51Y	108
CM1,1,2/CM1,2,1	-141.571	8	N-BAK2	108
CM1,2,2	103.990	15		108
CM2,1	114.106	21	S-FPL51Y	108
CM2,2	-621.324	73.345		108
CM3,1,1	60.646	16	S-FPL51Y	88
CM3,1,2/CM3,2,1	100.5579	8	SILICA	88
CM3,2,2	46.843	208.246		70
Detector	∞			36.2

Tabela 2.2 - Principais parâmetros ópticos do sistema de imageamento do BTFI.

Razão focal de entrada	16,625
Razão focal de saída	7,10
Distância focal efetiva	350 mm
Diâmetro da pupila	50 mm
Magnificência angular	2,338
Amostragem	0,12 " por pixel
Campo	3 x 3 '

vidro e sofre múltiplas reflexões na cavidade óptica. Dependendo do tamanho do *gap*, do comprimento de onda da luz incidente e do ângulo com que ela incide sobre o segundo étalon, ela pode ser transmitida ou pode ser refletida de volta.

O primeiro dos dois FPs, localizado logo após o plano focal, atua como um filtro de baixa resolução espectral ou como um seletor de ordens de interferência. O segundo, localizado sobre a pupila do instrumento no espaço colimado, atua como um Fabry-Perot clássico, conforme será descrito na seção 2.2.1. Ambos os Fabry-Perot foram desenhados para ter uma resolução espectral nominal variável entre $R = 500$ e $R = 2000$ no modo *tunable filter*, e $R = 6000$ e $R = 25000$, no modo clássico. Entretanto, ambos ainda estão em processo final de fabricação e desenvolvimento faltando a parte de aplicação de filmes finos nas superfícies de vidro dos *étalons*, pela empresa francesa Thales SESO⁴, e o desenvolvimento de um novo sistema de controle que, atualmente, está sendo liderado pelo *Laboratoire d'Astrophisique du Marseille*.

Por este motivo, o BTFI está utilizando Fabry-Perots Queensgate com resolução espectral nominal $R = 4500$ e $R = 20000$, alternadamente no plano da pupila. O primeiro deles, o LR FP ou FP de baixa resolução espectral, pertence à Universidade de Maryland e foi emprestado ao BTFI até que os FP originais estejam prontos. Os parâmetros nominais deste FP são listados na tabela 2.3. Nela, *gap* é a distância entre as placas do FP, *Free – Spectral – Range* é o intervalo espectral livre e *FWHM* é a largura à meia altura de uma linha (*Full-Width-at-Half-Maximum*).

O segundo FP, chamado de High Resolution (HR) FP, ou FP de alta resolução espectral, pertence ao Observatório Astronômico Australiano (*Australian Astronomical Observatory*

⁴ <http://seso.com/>

Tabela 2.3 - Parâmetros nominais para o LowRes-FP

<i>Gap</i>	$44\mu m$	
<i>Free-Spectral-Range</i>	$48,7 \text{ \AA} @ 6590,4 \text{ \AA}$	$2224 \text{ km/s @ H}\alpha$
<i>FWHM</i>	$1,4 \text{ \AA}$	$63,9 \text{ km/s @ H}\alpha$

– AAO). De acordo com o AAO, este FP opera com uma *Finesse* nominal (ver seção 2.2.1), parâmetro que está relacionado à nitidez das linhas espectrais observadas, igual a 30 em 5000 \AA e com uma resolução espectral de 24000 neste comprimento de onda. Baseado nestes valores, calculamos os parâmetros nominais deste FP usando as equações que apresentaremos na seção 2.2.1.

Tabela 2.4 - Parâmetros nominais para o HighRes FP

<i>Gap</i>	$200\mu m$	
<i>Free-Spectral-Range</i>	$6,25 \text{ \AA}$	$245,5 \text{ km/s @ H}\alpha$
<i>FWHM</i>	$0,208 \text{ \AA}$	$9,32 \text{ km/s @ H}\alpha$

Os demais parâmetros serão explicados posteriormente nesta seção. Os parâmetros obtidos experimentalmente para ambos FP são apresentados na seção 4.7.2.

Estes FPs são controlados por um dispositivo de controle chamado CS100 (Figura 2.6). O CS100, por sua vez, é controlado via computador através de uma placa eletrônica da *National Instruments*® e através de um software escrito em *LabView*.

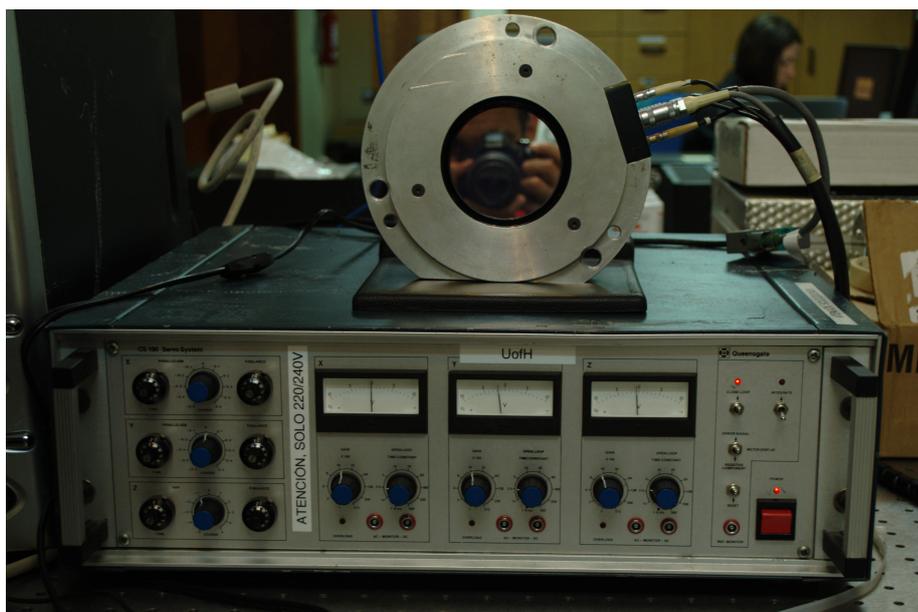


Figura 2.6: Foto do painel frontal do controlador CS100.

Para melhor compreender o funcionamento de um FP e os parâmetros envolvidos, eis abaixo uma breve explicação destes parâmetros, adaptada dos livros *Handbook of Optics* (Bass et al., 1995) e *Principles of Optics Electromagnetic Theory of Propagation* (Born e Wolf, 1980) e da dissertação de mestrado de Quint (2010).

2.2.1 Fabry-Perot operando em modo clássico

O modo clássico de operação de um Fabry-Perot consiste em posicioná-lo o mais próximo possível da pupila de um sistema óptico. Como mencionado anteriormente, um FP é um dispositivo óptico composto por duas placas de vidro cujas superfícies são planas e paralelas entre si. Normalmente, estas duas placas são dispostas de modo em que o vetor normal a estas superfícies coincida com o eixo óptico de um sistema (figura 2.7).

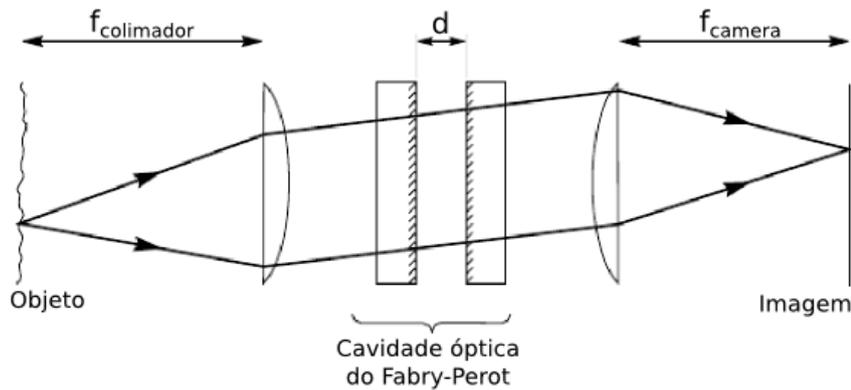


Figura 2.7: Princípio de funcionamento de um Fabry-Perot. Figura adaptada do livro Bass et al. (1995)

Estas placas também são posicionadas a uma distância d uma da outra de modo a formar uma pequena cavidade óptica. Esta distância d deve ser pequena o suficiente para que as ondas incidentes sejam ainda coerentes entre si e para que, assim, a interferência construtiva da luz possa ocorrer em alguns comprimentos de onda e interferência destrutiva possa ocorrer nos demais. O critério de interferência construtiva para um determinado comprimento de onda λ ocorre caso a diferença de caminho óptico entre dois feixes subsequentes na cavidade óptica seja igual a um múltiplo inteiro do comprimento de onda.

Esta distância d normalmente possui um valor nominal e pode variar dentro de certos limites, de acordo com a construção do Fabry-Perot. A variação de d é feita através de um software que se comunica com o sistema de controle do FP. O valor real de d é algo que precisa ser determinado utilizando dados de calibração. No sistema que usamos no BTFI,

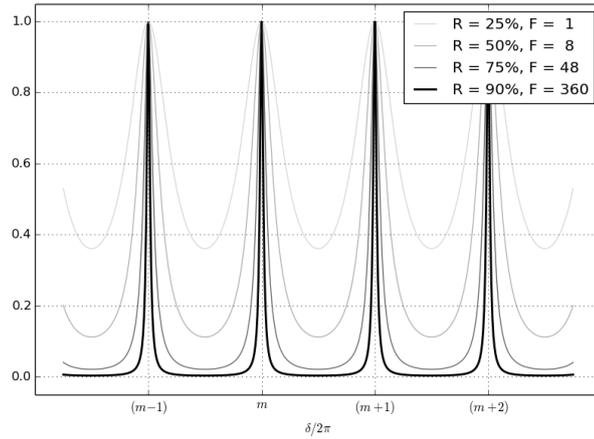


Figura 2.8: Intensidade relativa da luz transmitida através das placas do FP em função da diferença de fase δ e para diferentes reflectâncias R .

os valores passados ao controlador são dados em *binary control values* (valores binários de controle) ou *bcv* e a variável que utilizamos para representar estes valores é z . A relação entre d e z normalmente é linear mas polinômios de maiores ordens podem ser utilizados para relacioná-las com maior precisão se for necessário.

As superfícies das placas de vidro que estão voltadas para esta cavidade possuem um índice de reflexão, ou reflectância, R , definido pela razão entre a intensidade da luz incidente e a intensidade da luz refletida.

No caso de luz incidente não polarizada, a função que descreve a quantidade de luz transmitida através do Fabry-Perot é descrita por:

$$\frac{I^{(t)}}{I^{(i)}} = \frac{1}{1 + F \sin^2(\delta/2)} \quad (2.1)$$

Nesta equação, $I^{(t)}/I^{(i)}$ denota a razão entre a intensidade de luz transmitida e a intensidade da luz incidente. O parâmetro F é definido em função da reflectância:

$$F = \frac{4R}{(1 - R)^2} \quad (2.2)$$

A variável δ é a diferença de fase introduzida nos raios de luz. Podemos escrever esta diferença de fase em função do comprimento de onda λ do raio luminoso com ângulo de incidência θ e do índice de refração n na cavidade óptica com d de espessura:

$$\delta = \frac{4\pi n d \cos \theta}{\lambda} \quad (2.3)$$

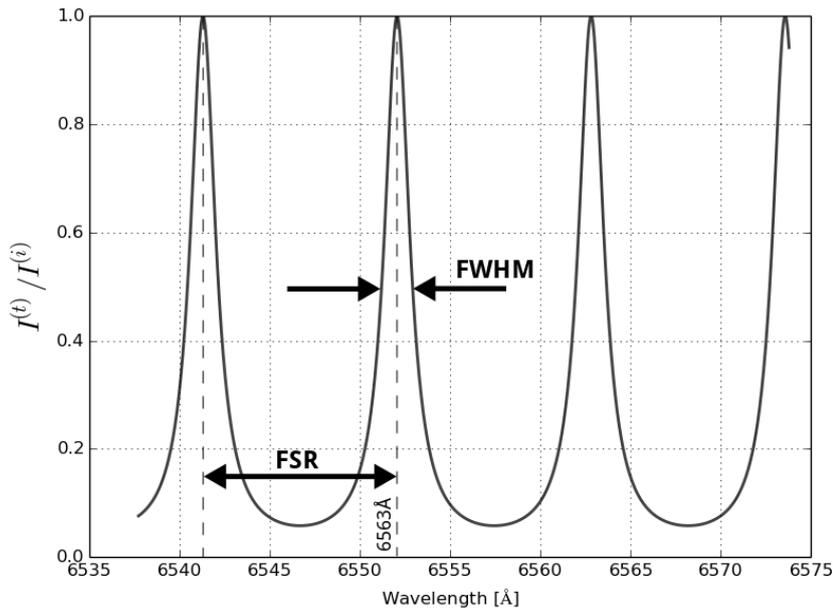


Figura 2.9: Exemplo de espectro obtido com um Fabry-Perot.

O gráfico apresentado na figura 2.8 mostra a razão $I^{(t)}/I^{(i)}$ em função de δ para diferentes valores de F . Um pico de interferência ocorre toda vez que a diferença de fase δ é um múltiplo inteiro de m ou, matematicamente:

$$\frac{\delta}{2\pi} = m = \frac{2nd \cos \theta}{\lambda} \quad (2.4)$$

Conforme descreve a equação 2.1, a diferença de fase depende de diversas variáveis. Para melhor compreensão do funcionamento de um FP temos que fixar algumas delas enquanto variamos as outras.

Por exemplo, consideremos um objeto no centro do campo observado. Por estar no centro, todos os raios luminosos emitidos por este objeto, ao serem colimados, ficam paralelos ao eixo óptico ($\theta = 0$). Consideremos também que ele seja um objeto pontual monocromático emitindo luz em H_{α} ($\lambda = 6562.8 \text{ \AA}$) e que existe ar entre as placas de vidro do FP de modo que o índice de refração seja $n = 1$. Nestas condições, podemos obter o espectro deste objeto variando o *gap* do FP entre 199,4 e 200,4 μm . Este espectro é apresentado na figura 2.9.

Neste gráfico, mesmo com uma fonte monocromática, temos vários picos de transmissão. Cada pico é uma ordem de interferência deste comprimento de onda. A distância entre dois picos em comprimento de onda nos dá o intervalo espectral livre ou, em inglês, o *Free-*

Spectral-Range, denotado por FSR. Este parâmetro é importante para determinar qual o intervalo máximo em que podemos obter um espectro em uma observação sem que as linhas comecem a se sobrepor.

Podemos quantificá-lo se considerarmos variações somente no domínio espectral, mantendo os outros parâmetros fixos e considerando que as ordens de interferência são muito maiores do que 1. Nessas condições, temos que:

$$\begin{aligned}
 \delta(m+1) - \delta(m) &= 2\pi \\
 \frac{4\pi nd \cos \theta}{\lambda_0} - \frac{4\pi nd \cos \theta}{(\lambda_0 + FSR_\lambda)} &= 2\pi \\
 \frac{1}{\lambda_0} - \frac{1}{\lambda_0 + FSR_\lambda} &= \frac{1}{2nd \cos \theta} \\
 FSR_\lambda &= \frac{\lambda_0(\lambda_0 + FSR_\lambda)}{2nd \cos \theta} \\
 FSR_\lambda &= \frac{\lambda_0 + FSR_\lambda}{m} \\
 FSR_\lambda - \frac{FSR_\lambda}{m} &= \frac{\lambda_0}{m} \\
 FSR_\lambda &\approx \frac{\lambda_0}{m}
 \end{aligned} \tag{2.5}$$

Para poder selecionar que região do espectro será utilizada, precisamos usar filtros interferenciais com largura de banda próxima a do FSR_λ utilizado na observação. No caso, o BTFI conta com um conjunto de quatro filtros interferenciais com 76,2 mm x 76,2 mm x 7,0 mm que atuam sobre luz não polarizada. Os filtros disponíveis são listados na tabela 2.5.

Tabela 2.5 - Filtros interferenciais disponíveis.

Código	Comprimento de onda central (Å)	Largura de Banda (Å)	Eficiência máxima (%)
6578/20	6578,8	19,9	75
6600/20	6600,0	19,3	75
5020/20	5020,9	16,5	65
6745/40	6745,5	38,6	75

O nome *Free-Spectral-Range* muitas vezes causa confusão visto que podemos iluminar o FP com uma fonte monocromática e calcular a distância entre dois picos de transmissão variando outros parâmetros. Por exemplo: se iluminarmos o FP com um feixe de luz com comprimento de onda λ_0 e incidindo com um ângulo θ , podemos determinar a distância

entre dois picos em função do tamanho do *gap* d . Desta forma, a diferença de fase varia somente com d e o FSR é determinado da seguinte forma:

$$\begin{aligned} \delta(m+1) - \delta(m) &= 2\pi \\ \frac{4\pi n(d + FSR_d) \cos \theta}{\lambda_0} - \frac{4\pi n d \cos \theta}{\lambda_0} &= 2\pi \\ FSR_d &= \frac{\lambda_0}{2n \cos \theta} \end{aligned} \quad (2.6)$$

Quando iluminamos um FP com uma fonte de luz monocromática e obtemos uma imagem utilizando somente um comprimento de onda, o padrão de interferência resultante é um conjunto de anéis concêntricos. A diferença entre os raios dos anéis define o FSR em função do ângulo θ ou FSR_θ . Para transformar o raio dos anéis em número de píxeis r_{pix} para um dado ângulo de incidência, utilizamos a equação 2.7. Nela, p é o tamanho de um píxel e f_{cam} é a distância focal da câmera. Estas duas variáveis precisam estar nas mesmas unidades.

$$\begin{aligned} \theta &= \arctan\left(\frac{r_{pix} \cdot p}{f_{cam}}\right) \\ \theta &\approx \frac{r_{pix} \cdot p}{f_{cam}} \end{aligned} \quad (2.7)$$

No centro destes anéis, $\theta = 0^\circ$ e a FSR_d iguala-se a $\lambda_0/2n$. O FSR em função de θ pode ser calculado se considerarmos d , n e λ_0 constantes. Nesta situação, a diferença entre o raio de dois anéis consecutivos:

$$\begin{aligned} \delta(m+1) - \delta(m) &= 2\pi \\ \frac{4\pi n d \cos(\theta + FSR_\theta)}{\lambda_0} - \frac{4\pi n d \cos \theta}{\lambda_0} &= 2\pi \\ \cos(\theta) \cos(FSR_\theta) - \sin(\theta) \sin(FSR_\theta) - \cos(\theta) &\approx \theta \cdot 1 - \theta FSR_\theta - \theta = \frac{\lambda_0}{2nd} \\ FSR_\theta &\approx \frac{\lambda_0}{2nd\theta} \approx \frac{\lambda_0}{2nd} \frac{f_{cam}}{r_{pix} \cdot p} \end{aligned} \quad (2.8)$$

Onde foi considerado que θ é um ângulo muito pequeno e não nulo.

É interessante notar, porém, que a espessura destes anéis ou a nitidez do espectro depende somente do parâmetro F definido na equação 2.2 e que, por sua vez, depende

somente da reflectância entre as placas. Isto pode ser comprovado se considerarmos a largura-à-meia-altura das linhas de interferência ϵ ou FWHM, mostrado na Figura 2.9. Para uma franja de ordem m , os pontos onde a intensidade vale a metade do máximo são $\delta = 2m\pi \pm \epsilon/2$. Igualando $I^{(t)}/I^{(i)}$ à $1/2$, temos:

$$\frac{1}{2} = \frac{1}{1 + F \sin^2 \frac{\epsilon}{2}} \quad (2.9)$$

Quando F é suficientemente grande, ϵ é pequeno e podemos aproximar $\sin \epsilon/4 \approx \epsilon/4$. Com isso, a equação 2.9 torna-se:

$$\epsilon = \frac{4}{\sqrt{F}} \quad (2.10)$$

Aqui, definimos um novo parâmetro bastante utilizado quando falamos de Fabry-Perot chamado de *Finesse* como a razão entre a diferença de fase entre duas franjas de interferência de ordens adjacentes $\delta(m+1) - \delta(m) = 2\pi$ e a largura-à-meia-altura:

$$\mathcal{F} = \frac{2\pi}{\epsilon} = \frac{\pi\sqrt{F}}{2} \quad (2.11)$$

De novo, a *Finesse* de um FP depende essencialmente da reflectância das placas. Na prática, podemos determinar a *Finesse* de um FP medindo o FSR radial ou em z usando uma fonte monocromática e podemos dividir este valor pela FWHM no mesmo domínio em que ela foi medida.

$$\mathcal{F} = \frac{FSR_d}{FWHM_d} = \frac{FSR_\theta}{FWHM_\theta} \quad (2.12)$$

Numa observação, ajustamos o valor de z e obtemos uma imagem. Esta imagem conterá anéis relacionados com as ordens de interferência construtiva. Conforme aumentamos ou diminuimos z , a distância entre as placas d também aumenta ou diminui e os raios dos anéis encontrados nas imagens também aumentam ou diminuem. Uma varredura completa com o FP consiste numa sequência de imagens variando z de modo a cobrir um FSR. A cada imagem obtida, o incremento a ser dado em z precisa ser, no máximo, a metade de um $FWHM_z$. Cada incremento dado é chamado de passo. Para cobrir um FSR _{z} completo, o número de passos a ser dado é o dobro da *Finesse* de um FP. Mais detalhes sobre a determinação dos parâmetros utilizados em uma observação com o FP podem ser encontrados no capítulo 4.

2.3 iBTF

O iBTF (*imaging Bragg Tunable Filter*) é o segundo módulo utilizado dentro do BTFI para a obtenção de cubos de dados, responsável por fornecer cubos de baixa resolução espectral (entre 5 e 20) e de resolução espectral intermediária (entre 1000 e 2000). Ele funciona simplesmente posicionando as duas redes de difração holográficas paralelas entre si de modo que o feixe colimado que vem do telescópio atravessa a primeira, sofrendo dispersão por difração, que é então cancelada assim que este mesmo feixe atravessa a segunda rede (figura 2.10).

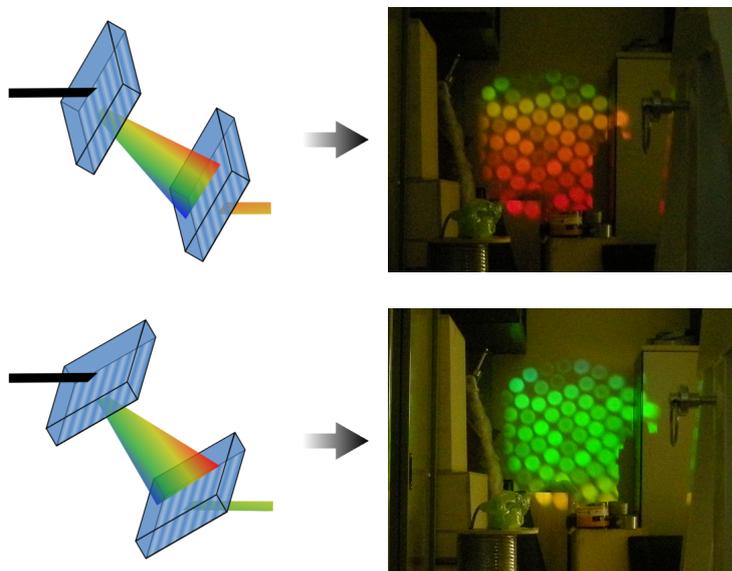


Figura 2.10: À esquerda: representação simplificada de duas redes de difração em duas diferentes configurações. À direita, fotos obtidas em laboratório mostrando um feixe de luz proveniente de um cabo de fibras ópticas iluminado por uma lâmpada incandescente passando através do iBTF com as redes a 45° (figuras superiores) e 35° (figuras inferiores).

Fotos e figuras extraídas da dissertação de Quint (2010).

Apresentamos, aqui, uma breve descrição sobre o funcionamento das redes de difração holográficas seguida de uma descrição do iBTF em si, focando no seu sistema mecânico e de controle.

2.3.1 Redes de difração holográficas

As redes de difração holográficas ou, em inglês, *Volume-Phase Holographic Gratings* — VPHG ou redes VPH são elementos ópticos dispersivos utilizados para separar os comprimentos de onda da luz que incidem sobre elas. A grosso modo, elas operam da mesma

forma que as redes de difração clássicas. A principal diferença entre as redes VPH e as redes clássicas de difração está no modo em que a diferença de caminho óptico entre os raios de luz é introduzida. Nas redes clássicas de difração, a diferença de caminho óptico é introduzida por obstáculos físicos feitos na superfície das redes, como a sequência de fendas ou a sequência de superfícies refletoras.

Nas redes VPH, a diferença de caminho óptico é introduzida através da variação periódica ou modular do índice de refração de um material holográfico que é depositado sobre um substrato, como o vidro óptico. Conforme a direção, a frequência e a amplitude desta modulação do índice de refração, a rede de difração possui diferentes propriedades.

Uma forma bastante simples de representar uma rede de difração holográfica é através de uma região do espaço limitada por dois planos infinitos, conforme descrito em Kogelnik (1969). Por simplicidade, dizemos que estes planos são paralelos ao plano XY do sistema cartesiano de coordenadas. Dizemos também que um deles passa pela origem e o outro cruza o eixo Z na coordenada $z = d$. Definimos nosso eixo de coordenadas de modo que Z e X estejam no plano do papel e apontem, respectivamente, para a direita e para cima e que Y seja perpendicular a este plano e esteja apontando para fora do papel (figura 2.11).

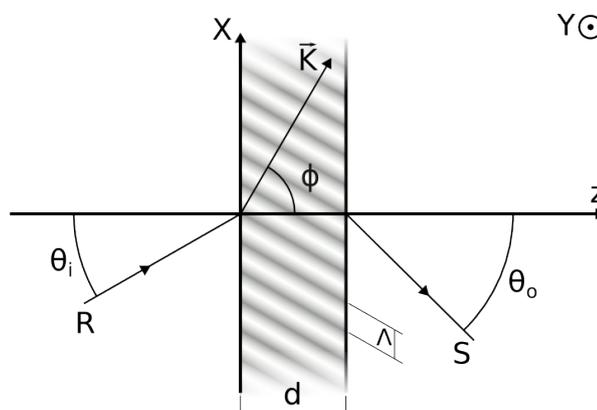


Figura 2.11: Representação geométrica simplificada de uma rede de difração holográfica. Figura baseada em Kogelnik (1969).

Consideramos, agora, que este filme holográfico possua um índice de refração que varia periodicamente seguindo uma função seno ou cosseno e que o vetor \vec{K} , que determina a direção em que ele varia, está contido no plano ZX. Este vetor, chamado de *grating vector* ou vetor da rede faz um ângulo ϕ com o eixo Z e sua amplitude é definida como $\vec{K} = 2\pi/\Lambda$, onde Λ é o período de modulação da variação do índice de refração. A constante dielétrica dentro e fora do filme holográfico são iguais entre si.

Usamos, agora, um raio de luz com comprimento de onda λ_0 incide sobre esta rede e que faça um ângulo θ_i com o eixo Z. Consideramos que o plano de incidência que contém este raio também esteja contido no plano ZX.

De acordo com a teoria de Kogelnik, caso o raio incidente S satisfaça a Lei de Bragg, difração ocorre e ele emerge do outro lado do filme holográfico com $\theta_o = \theta_i = \theta$. A Lei de Bragg é uma equação utilizada no estudo de raios X para determinar a distância entre planos atômicos em estruturas cristalinas. Seu formato original é apresentado na equação 2.13. Nela, m é um número inteiro indicando a ordem de interferência, λ é o comprimento de onda da radiação incidente, d é a distância entre dois planos cristalinos e θ é a metade do ângulo que separa o raio incidente e o raio emergente (figura 2.12).

$$m\lambda = 2d \sin \theta \quad (2.13)$$

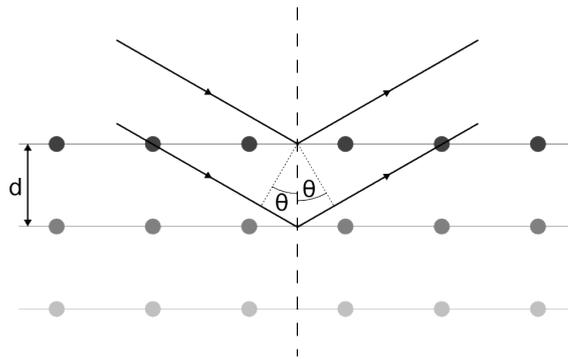


Figura 2.12: Lei de Bragg para estruturas cristalinas - figura baseada em Myers (2009, figura 3.2).

A forma adaptada da Lei de Bragg para o caso de redes VPH é apresentada na equação 2.14. Caso o raio incidente satisfaça essa lei, temos difração. Caso contrário, a luz segue seu caminho através do filme holográfico. Os parâmetros utilizados nesta equação são os mesmo utilizados na figura 2.11.

$$m\lambda_0 = 2\Lambda n \cos(\phi - \theta_0) \quad (2.14)$$

Onde θ_0 é o ângulo em que a difração é máxima para um determinado comprimento de onda λ_0 . Conforme definimos anteriormente, Λ é o período da modulação do índice de refração ou, em outras palavras, a distância entre dois planos com mesmo índice de

refração. O ângulo ϕ , chamado de *slant angle*, possui um papel importante na determinação do modo de operação das redes holográficas. Dependendo do valor que ele assume, a rede opera como uma rede de transmissão ou como uma rede de reflexão. A influência de ϕ nas equações de onda da radiação incidente e emergente é descrita com detalhes em Kogelnik (1969, seção 2.2). Para simplificar, podemos dizer que, quando $\phi = 0$, a equação 2.14 transforma-se na equação 2.15 e a rede atua como uma rede de reflexão. Quando $\phi = \pi/2$, a equação 2.14 transforma-se na equação 2.16 e a rede atua como uma rede de transmissão (figura 2.13). Para outros casos, uma análise aprofundada dos parâmetros do sistema envolvendo a rede, o ângulo de incidência e o comprimento de onda precisa ser feita antes que possamos dizer qual o modo em que a rede opera.

$$m\lambda_0 = 2\Lambda n \cos(\theta_0) \quad (2.15)$$

$$m\lambda_0 = 2\Lambda n \sin(\theta_0) \quad (2.16)$$

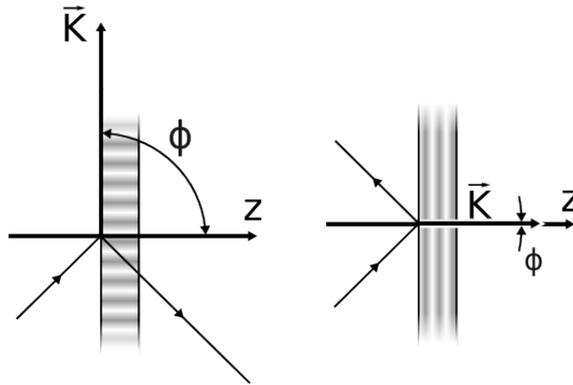


Figura 2.13: Modos de operação das redes de difração holográficas de acordo com o *slant angle* ϕ . À esquerda, rede de transmissão. À direita, rede de reflexão.

Para redes de difração holográficas operando em modo de transmissão, a eficiência η_t é dada pela equação 2.17.

$$\eta_t = \frac{\sin^2 \sqrt{\nu^2 + \xi^2}}{(1 + \xi^2/\nu^2)} \quad (2.17)$$

Estes dois parâmetros são definidos como:

$$\nu = \frac{\pi \Delta n d}{\lambda \cos \theta} \quad (2.18)$$

$$\xi = \frac{\vartheta d}{2 \cos \theta} \quad (2.19)$$

Onde Δn é a amplitude da modulação do índice de refração e ϑ é um fator chamado de *dephasing measure*. Esta última variável mede o quão distante da condição de Bragg o nosso sistema está. Ela é definida, de modo geral, pela equação 2.20.

$$\vartheta \equiv \frac{2\pi}{\Lambda} \cos(\theta - \phi) - \frac{\pi}{n\Lambda^2} \lambda \quad (2.20)$$

Para melhor compreendê-la, consideramos o caso em que $\theta = \theta_0 + \Delta\theta$ e $\lambda = \lambda_0 + \Delta\lambda$ estão bem próximos da condição de Bragg. Em outras palavras, consideramos o caso em que $\Delta\theta$ e $\Delta\lambda$ são pequenos. Com esta hipótese, podemos, então, expandir a equação 2.20 em série de Taylor e obtemos:

$$\vartheta \approx \Delta\theta \frac{2\pi}{\Lambda} \sin(\theta - \phi) - \Delta\lambda \frac{\pi}{n\Lambda^2} \quad (2.21)$$

No caso de redes que operam no modo de reflexão, a eficiência é dada por

$$\eta_r = \left[1 + \frac{(1 - \xi^2/\nu^2)}{\sinh^2 \sqrt{\nu^2 - \xi^2}} \right]^{-1} \quad (2.22)$$

A forma inicial dos parâmetros ν e ξ no artigo de Kogelik é diferente nos casos de redes que operam no modo de transmissão ou de reflexão. Entretanto, ao aplicar as equações nas definições do artigo e manipulá-las devidamente, concluímos que ν e ξ são os mesmos em ambos os casos. Além disso, podemos repetir a análise para casos em que a condição de Bragg é satisfeita e a equação 2.22 torna-se:

$$\lim_{\vartheta \rightarrow 0} \eta_r = \lim_{\xi \rightarrow 0} \eta_r = \tanh^2 \left(\frac{\pi d \Delta n}{\lambda \cos \theta_0} \right) \quad (2.23)$$

A distribuição de energia da luz que sofre difração em função do comprimento de onda e dos demais parâmetros de construção de uma determinada rede de difração clássica, seja ela de transmissão ou de reflexão, é chamada de *blaze function*. Em redes de difração clássicas, a *blaze function* é uma propriedade intrínseca da rede e precisa ser otimizada conforme a sua aplicação (Palmer e Loewen, 2005).

Já no caso de redes de difração holográficas, para cada ângulo de incidência, a distribuição de energia é diferente e possui uma eficiência máxima no comprimento de onda que

satisfaz a equação de Bragg. Em outras palavras, a *blaze function* de uma rede de difração holográfica varia com o ângulo de incidência. As figuras 2.14 e 2.15 mostram exemplos de simulações de VPHs com diferentes *blaze functions* para diferentes ângulos de incidência.

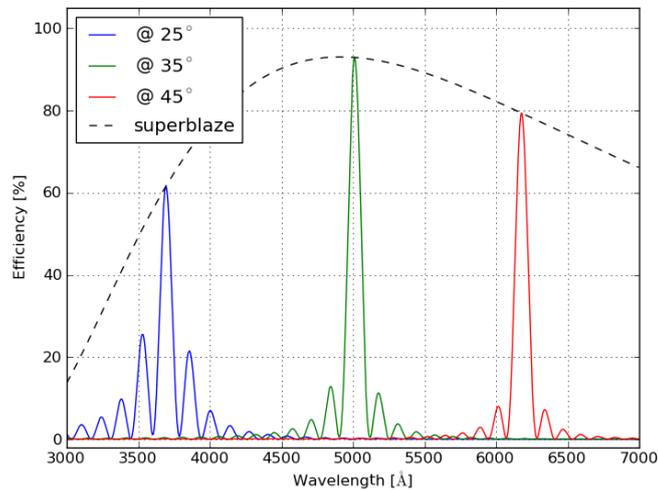


Figura 2.14: Rede VPH de transmissão simulada usando modelo de Kogelink. Figura extraída de Quint (2010).

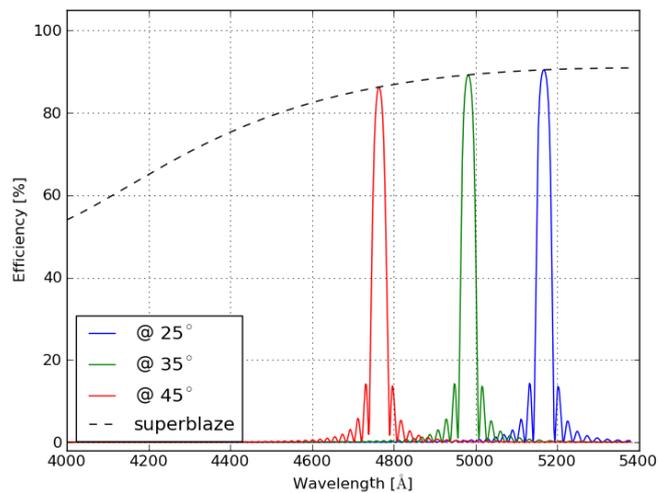


Figura 2.15: Rede VPH de reflexão simulada usando modelo de Kogelink. Figura extraída de Quint (2010).

Estas duas figuras mostram a distribuição de energia em função do comprimento de onda para para 25° , 35° e 45° . A curva tracejada que liga o máximo de eficiência nestes três ângulos é chamada de *super-blaze*. Ela descreve a eficiência máxima para um determinado λ - θ que obedeça a condição de Bragg.

Além de ter a função *blaze* ajustável e de poderem ser fabricadas para trabalhar em modo de reflexão ou de transmissão, as redes VPH apresentam muitas outras vantagens sobre as redes de difração clássica (Barden et al., 1998). Entre elas, está o fato de que uma rede VPH pode ter eficiências muito mais altas que as redes de difração clássicas. Além disto, elas podem ser utilizadas aos pares, de modo que a dispersão da luz que ocorre na primeira rede de difração VPH possa ser cancelada pela segunda, recuperando, assim, a informação angular em um determinado comprimento de onda. Sobre as redes holográficas fabricadas especialmente para o BTFI, daremos mais detalhes na seção 3.3.

2.3.2 Montagem opto-mecânica do iBTF

No modelo construído para o BTFI, uma das redes de difração é instalada em um suporte com movimento de rotação e a outra é instalada em um suporte com movimento de rotação e translação. Desta forma, a segunda rede pode deslocar-se para buscar a luz que sai da primeira rede. Por convenção nossa, chamamos o primeiro estágio de rotação de M2, o segundo estágio de rotação de M1 e o estágio de translação de M3.

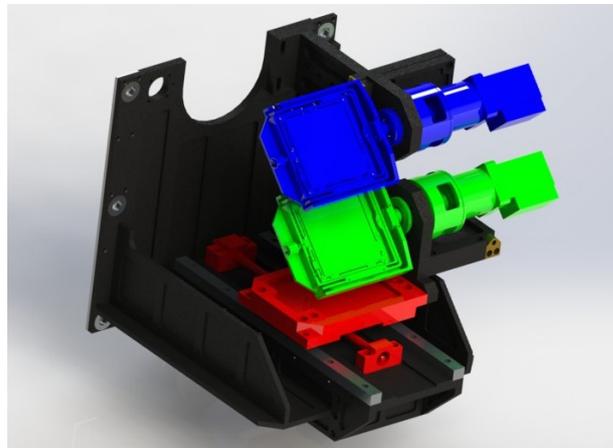


Figura 2.16: Modelo tridimensional feito em SolidWorks mostrando os dois estágios de rotação (M2 em azul e M1 em verde) e o estágio de translação (M3 em vermelho).

Todos os movimentos são controlados através de um software escrito em VisualBasic 6 que faz a interface entre o sistema de controle do BTFI, que será descrito na seção 2.5 deste capítulo, e um dispositivo PLC (do inglês, *Programmable logic controller*) Mitsubishi QD75MH que se comunica com os três servo-motores Mitsubishi HF KP43K através de um *driver* Mitsubishi MR-J3-40B (1).

2.4 Detectores

Para a captação e armazenamento de luz, o BTFI conta com um tipo de detector chamado *Electron Multiplier Coupled-Charge Devices*, ou EMCCDs. Estes detectores são baseados nos CCDs clássicos, já bastante utilizados na astronomia. A descrição de CCDs e EMCCDs a seguir foi baseada no capítulo 2 da dissertação de mestrado de Andrade (2011).

Um CCD consiste basicamente em uma superfície de material semicondutor sensível à luz, como o silício. Ao expor esta superfície à luz, os fótons que incidem sobre ela podem liberar elétrons que estão na banda de valência, com menor nível de energia deste material, e os jogam para a banda de condução, com maior nível de energia. Ao ocupar a banda de condução, estes elétrons ficam livres para percorrer a superfície do CCD. Entretanto, estes elétrons livres são acoplados a um campo elétrico aplicado localmente numa região próxima à superfície. Chamamos esta pequena região de píxel.

No caso, um CCD é dividido em uma matriz com N linhas e M colunas de píxeis. O tamanho de um pixel define o limite inferior da amostragem de uma imagem. Considerando um elemento de resolução com tamanho fixo, quanto menor o tamanho de um pixel, mais píxeis serão utilizados por elemento de resolução. O número $N \times M$ de píxeis de um CCD define a área de coleta do CCD que, por sua vez, está relacionada com o campo de observação e com a amostragem angular por píxel.

A quantidade de elétrons aprisionada em cada píxel está relacionada com o fluxo de fótons que incidem sobre o CCD e com a sua eficiência (probabilidade de transformar um fóton em elétron). Após expor o CCD por um determinado tempo, os elétrons ou cargas liberadas durante a exposição são transportados até um amplificador eletrônico que os converte em sinal digital. Esta conversão é chamada de leitura do CCD. O transporte destas cargas é feito píxel a píxel através da variação espacial do campo elétrico ao longo das colunas do CCD (registros paralelos) até última linha do CCD (registro serial). Depois, o transporte destas cargas é feito píxel a píxel ao longo do registro serial até o amplificador eletrônico.

A figura 2.17 também foi extraída da dissertação de Andrade (2011). Ela nos mostra uma analogia entre o CCD e as esteiras com baldes que recebem água da chuva.

Os CCDs clássicos possuem duas fontes principais de ruído. A primeira delas é de

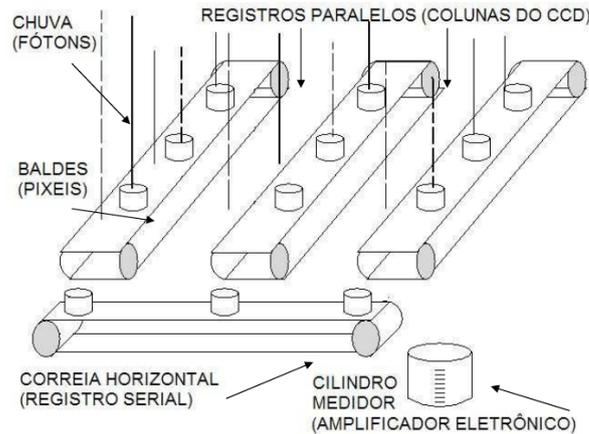


Figura 2.17: Funcionamento de um CCD - Analogia dos baldes.
Extraída de Andrade (2011)

natureza térmica e este ruído é chamado de *dark noise* ou ruído de escuro. A radiação térmica que permeia o meio em que o CCD está imerso também pode fornecer energia o suficiente para gerar elétrons livres. Por este motivo, é bastante comum resfriar os CCDs a baixas temperaturas utilizando nitrogênio líquido ou sistemas de refrigeração a gás. No caso do BTFI, por exemplo, os detectores operam a -100°C .

A segunda fonte de ruído, chamada de ruído de leitura, ou *readout noise*, é gerada no momento da leitura das cargas livres sobre um detector. Podemos diminuir o ruído de leitura de um CCD se diminuirmos também a taxa de leitura. Entretanto, para fontes muito fracas, o tempo perdido com a leitura de uma imagem pode ser muito grande, podendo tornar a observação impraticável.

Apesar de existirem várias técnicas para minimizar o ruído de leitura e o ruído de escuro, conforme será descrito no capítulo 4, eles ainda se tornam dominantes quando vamos observar fontes muito fracas. E é aí que o EMCCD entra.

Um EMCCD funciona praticamente da mesma maneira que um CCD convencional. A diferença está na inclusão de um registro serial prolongado na saída do detector e nas tensões aplicadas no transporte das cargas (figura 2.18). Nesta figura, o detector possui uma região que é exposta à luz (*image area*) e uma área “reserva” (*storage area*) onde ela armazena a imagem que será lida. As tensões aplicadas nos eletrodos do registro multiplicador são muito maiores do que as tensões convencionais. Com estas altas tensões, os elétrons são acelerados ao passar de uma célula a outra no registro multiplicador e, com isso, há uma pequena chance de liberarem um outro elétron. Apesar da probabilidade ser

pequena (Djazovski et al., 2013), após passar por vários estágios de multiplicação o ganho final pode ser de centenas ou milhares de vezes.

O ruído de leitura causado pelo amplificador é efetivamente dividido por este ganho final resultando em ruídos de leitura desprezíveis (inferiores a um elétron por leitura). O ganho efetivo do registro multiplicador de um EMCCD pode ser ajustado variando a amplitude da alta tensão aplicada no transporte das cargas. Com isso, um EMCCD pode operar com um ganho efetivo de 1 e se comportar como um CCD convencional. Assim, para fontes mais fracas, o EMCCD possui um melhor desempenho que os CCDs convencionais (Djazovski et al., 2013; Daigle et al., 2009). Entretanto, conforme já mencionado na introdução deste trabalho, os EMCCDs possuem um custo mais elevado em comparação ao CCD convencional e um sistema eletrônico mais delicado e difícil de construir, ajustar e controlar.

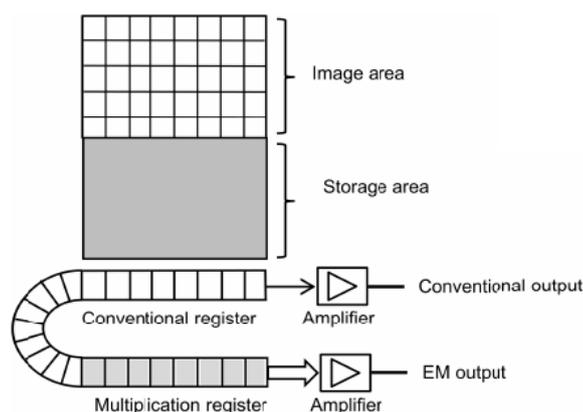


Figura 2.18: Figura extraída de Djazovski et al. (2013) mostrando o funcionamento de um EMCCD.

O BTFI conta com dois detectores $e2v^5$ CCD207 EMCCD controlados por um dispositivo chamado CCCP (*CCD Controller for Counting Photons*), desenvolvido na Universidade de Montreal e comercializado atualmente pela empresa NuVu Câmeras⁶ (Daigle et al., 2009). As informações técnicas dos EMCCDs utilizados no BTFI são reunidas na tabela 2.6. Os níveis de ruídos foram extraídos de Andrade et al. (2012), que apresenta uma descrição técnica detalhada da construção dos EMCCDs usado no instrumento.

Apesar dos detectores EMCCD utilizados no BTFI serem descritos aqui, sua a caracterização não faz parte desta tese.

⁵ <http://www.e2v.com/>

⁶ www.nuvucameras.com/

Tabela 2.6 - EMCCDs no BTFI - Informações técnicas.

Modelo	e2v CCD207 EMCCD
Número de pixels	1696 x 1616
Tamanho de cada píxel	19 μ m
Taxa de leitura	10MHz
Ruído de leitura	0,044 $e^- .px^{-1}$
Ruído de escuro	0,00121 $e^- .px^{-1} .s^{-1}$
Ruído CIC	0,009 $e^- .px^{-1}$

2.5 Sistema de controle

O BTFI é controlado, essencialmente, por um software escrito em LabView 2011 chamado KAFKA. Este software foi escrito pelo engenheiro de controle Álvaro Calasans e modificado pelos engenheiros Renato Severo e Denis Andrade e pelo astrônomo Rodrigo Massera (2012, comunicação privada). Ele centraliza o controle de todos os componentes através de uma interface gráfica que traduz os comandos dados pelo usuário em mensagens em formato ASCII. A comunicação entre o KAFKA e os drivers de cada componente ocorre via TCP/IP. Todos os comandos são enviados a partir do KAFKA para um determinado endereço de IP e uma determinada porta de comunicação.

Os motores responsáveis por controlar as três rodas de filtros são controlados através de um *Microsoft® Terminal Server*. Este dispositivo recebe os comandos via rede Ethernet e os repassa para cada um dos motores de acordo com a porta de comunicação utilizada.

As câmeras são controladas por um software escrito em LabView e C++ chamado PanView pelo Eng. de software Marco Bonati (CTIO - comunicação privada). O PanView é instalado em um computador dedicado rodando Ubuntu 10.04. A comunicação entre o PanView e as câmeras é dada através de fibras ópticas que saem deste computador e chegam até as duas câmeras.

Um software escrito pelo engenheiro de controle Fernando Fontes e aprimorado pelo Eng. Renato Severo (Unipampa, 2010 - comunicação privada) e pelo atual astrônomo Rodrigo Massera (USP, 2013 - comunicação privada) em VisualBasic®6 é o responsável por comandar o iBTF. Este software está hospedado em um computador com sistema operacional Windows®XP. A comunicação entre ele e o iBTF é feita através de um driver PLC (*Programmable logic controller* ou Controlador Lógico Programável) via porta serial.

Conforme já descrito na seção 2.2, os atuais FPs são comandados por controladores

CS100 que, por sua vez, se comunicam com o KAFKA através de um driver escrito em LabView e uma placa lógica da *National Instruments* instalada em um computador rodando Windows®XP.

A figura 2.19 mostra a interface principal do KAFKA, o modo de operação. Os campos apresentam-se apagados pois esta imagem foi obtida com o KAFKA operando sem estar conectado aos demais sistemas.

No canto superior esquerdo desta tela, temos duas abas: o modo de configuração (*Configuration Mode*) e o modo de operação (*Operation Mode*). Os comandos utilizados numa noite normal de observação estão todos na aba do modo de operação. No lado direito, temos informações de data e hora, da posição em que a montagem do telescópio está apontando e informações do ambiente como temperatura, umidade e velocidade do vento. Estas informações são todas lidas diretamente do sistema interno do telescópio.

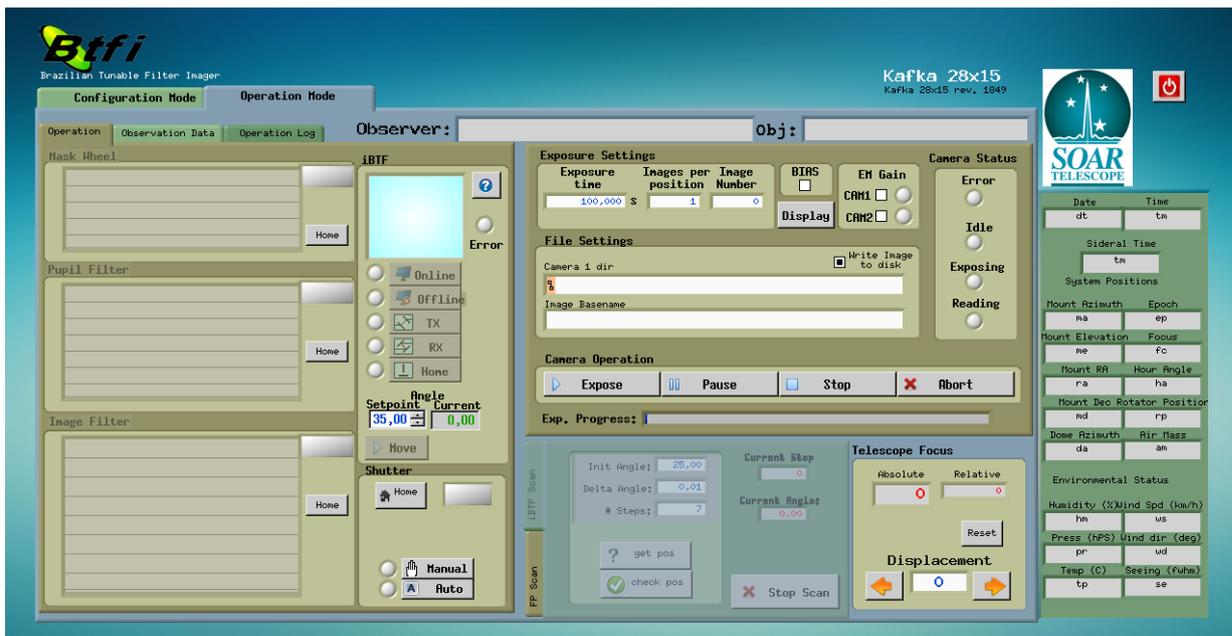
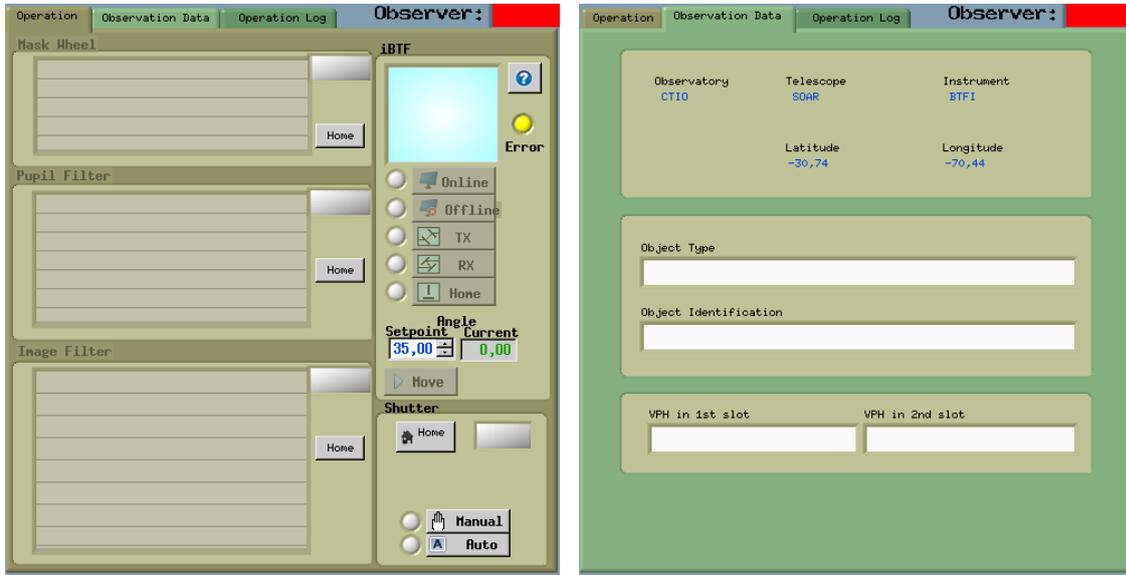


Figura 2.19: Tela principal do KAFKA – *Operation Mode*.

No lado esquerdo desta janela, temos mais três abas. A primeira delas é a aba de operação (figura 2.20a). Ela armazena as possíveis configurações das rodas de filtro e do iBTF. Ela é também usada para controlar estas configurações, i.e., trocar o filtro que está sendo utilizado, ajustar o ângulo de operação do iBTF ou mesmo trocar o modo de operação, entre reflexão e transmissão.

A aba *Observation Data* (figura 2.20b) é utilizada somente para armazenar informações



(a) Operation

(b) Observation



(c) Log

Figura 2.20: As três abas disponíveis no lado esquerdo do Modo de Operação do KAFKA.

a respeito da observação em si. Ela possui campos onde o usuário pode entrar com o nome do objeto observado, o tipo de aquisição de imagens (BIAS, DARK, calibração, observação, etc) e o nome das redes de difração holográficas instaladas no iBTF. Todas estas informações são armazenadas no cabeçalho das imagens obtidas pelo BTFI.

A terceira aba possui um registro de todos os comandos fornecidos pelo usuário ao KAFKA (figura 2.20c). Ela é bastante útil para solucionarmos eventuais problemas que venham a acontecer durante uma noite de observação.

Na parte central e superior, temos um campo que controla o sistema de imageamento do BTFI. É nela onde podemos colocar o tempo de exposição de cada imagem, o número de imagens em cada posição no caso de uma varredura com o iBTF ou com o FP e o número do sufixo a ser atribuído à próxima imagem. Esta parte também apresenta os campos onde é armazenado o caminho das imagens no computador de aquisição do BTFI e o nome de base destas imagens. Neste campo também temos caixas de seleção usadas para escolher o modo de operação do detector: modo EM ou modo clássico. Por fim, temos os botões utilizados para expor uma ou mais imagens, para pausar uma exposição, para parar uma sequência de imagens ou para abortar no meio de uma exposição.

Na parte central e inferior, temos um campo com duas abas responsável pela varredura usando o iBTF ou o FP. No modo atual, temos que escolher um modo ou o outro. Estas abas possuem campos para colocarmos a posição inicial onde nossa varredura iniciará, *Init Angle* para o iBTF ou *Initial Z* para o FP; qual o incremento dado em cada passo da varredura, *Delta Angle* para o iBTF ou *Delta Z* para o FP; e o número de passos dados, *# Steps* para ambos. No caso de uma varredura com o FP, ainda temos a possibilidade de realizar a varredura diversas vezes e em diferentes sentidos: sempre incrementando a posição (*One Direction*) para cada passo ou incrementando e depois diminuindo (*Up and Down*). O campo responsável por determinar quantas vezes a varredura será feita nas mesmas configurações para o FP é o *# Sweeps*. As imagens obtidas nesta varredura são utilizadas posteriormente para montar um cubo de dados.

Ainda podemos ver no modo de operação, ao lado da área responsável pela aquisição de um cubo de dados, um campo responsável por alterar a posição do espelho terciário do telescópio. Com isso, podemos compensar a diferença entre o foco do telescópio e o foco instrumental (seção 3.1.3). Entretanto, os botões deste campo ainda não estão completamente funcionais e o ajuste do foco relativo entre o telescópio e o instrumento

deve ser feito pelo próprio operador do telescópio durante uma noite de observação.

A aba para o modo de configuração é utilizada para alterar o nome dos filtros instalados no instrumento e para ter um controle de mais baixo nível dos drivers que controlam os motores destes mesmos filtros.

No próximo capítulo, usamos o que já foi exposto até aqui para proceder à caracterização do instrumento e descrição sobre sua operação.

Operação e Caracterização

Em essência, o BTFI é um sistema de imageamento simples com um colimador e uma câmera. A luz proveniente do telescópio entra no BTFI, convergindo até o plano focal e, depois, passa a divergir. Em seguida, ela encontra as lentes do colimador, que fazem com que todos os raios luminosos provenientes de um mesmo ponto fiquem paralelos entre si. Por fim, estes raios encontram as lentes da câmera óptica e fazem com que o feixe luminoso foque sobre o detector. Cada um destes elementos possui papel fundamental para que possamos adquirir imagens de alta qualidade com o BTFI.

Neste capítulo, faremos a caracterização instrumental do BTFI baseando-nos no que foi proposto durante a primeira parte do capítulo 2. Iniciamos nosso processo apresentando a qualidade óptica sem considerar a presença do iBTF ou do FP justamente para podermos estudar a influência destes dois.

Já a resposta espectral é algo que depende explicitamente de cada um dos modos utilizados e das diversas opções de operação. Por fim, faremos uma análise da eficiência operacional do sistema, como um todo.

3.1 Sistema de imageamento simples

Antes de partirmos para uma análise sobre o desempenho do BTFI utilizando o FP ou o iBTF, daremos uma breve explicação sobre o procedimento empregado no alinhamento óptico-mecânico dos componentes. Logo em seguida, faremos um estudo do foco instrumental nominal, seguindo para a sua relação com o foco do telescópio. Temos também uma comparação entre a qualidade de imagens obtidas no modo clássico e no modo EM dos detectores do BTFI. Por fim, faremos uma estimativa da eficiência operacional do sistema

de imageamento medindo o tempo que é gasto com a parte de controle do instrumento.

3.1.1 Alinhamento óptico

O alinhamento dos componentes optomecânicos do instrumento BTFI devem ser realizados com o instrumento fora do telescópio. Para tal procedimento, posicionamos o BTFI na vertical. Todo alinhamento óptico deve ser feito baseado em alguma referência absoluta. No caso do BTFI, temos que nos aproximar o máximo possível do eixo óptico do telescópio. Podemos assumir que a placa de interface entre o BTFI e o telescópio é perfeitamente plana e perpendicular ao eixo óptico do telescópio. Assumimos também que o centro do círculo formado pelos parafusos de acoplamento coincide com o eixo óptico do telescópio.

Para definir um eixo óptico dentro do instrumento BTFI próximo ao eixo óptico do telescópio, vamos utilizar uma técnica de alinhamento chamada de autorreflexão e centragem de campo. A primeira técnica consiste em fazer um feixe de laser colimado percorrer parte do sistema óptico do BTFI até o elemento óptico a ser alinhado. Logo em seguida, este laser deve ser refletido com um espelho plano mecanicamente acoplado ao sistema óptico em processo de alinhamento e deve voltar pelo mesmo caminho que percorreu até chegar neste espelho. Caso o laser volte exatamente sobre o mesmo caminho, dizemos que o sistema está alinhado opticamente em relação ao laser. Caso contrário, é necessário ajustar o componente óptico até que essa condição seja satisfeita.

Assim sendo, como todos os elementos ópticos dentro do BTFI são alinhados com a ajuda deste laser, o primeiro passo foi fazer com que ele coincida com o eixo óptico do telescópio dentro do BTFI. Conforme já foi dito, o eixo óptico passa exatamente pelo centro do círculo formado pelos parafusos que acoplam o BTFI ao telescópio. Para encontrá-lo, amarramos um fio de linha em dois pares de parafusos opostos. O centro do círculo é definido como o ponto de intersecção entre estas duas linhas (figura 3.1).

Em seguida, tivemos que garantir que o feixe de laser seja perfeitamente paralelo ao eixo óptico do telescópio. Isso foi alcançado colocando um espelho plano logo na entrada do instrumento. Assim, para que o laser coincida com o eixo óptico, ele precisa refletir neste espelho e voltar pelo mesmo caminho e, também, passar pelo ponto em que as duas linhas descritas anteriormente se cruzam.

Existem diversos suportes para laser no mercado que possuem ajustes lineares e angu-

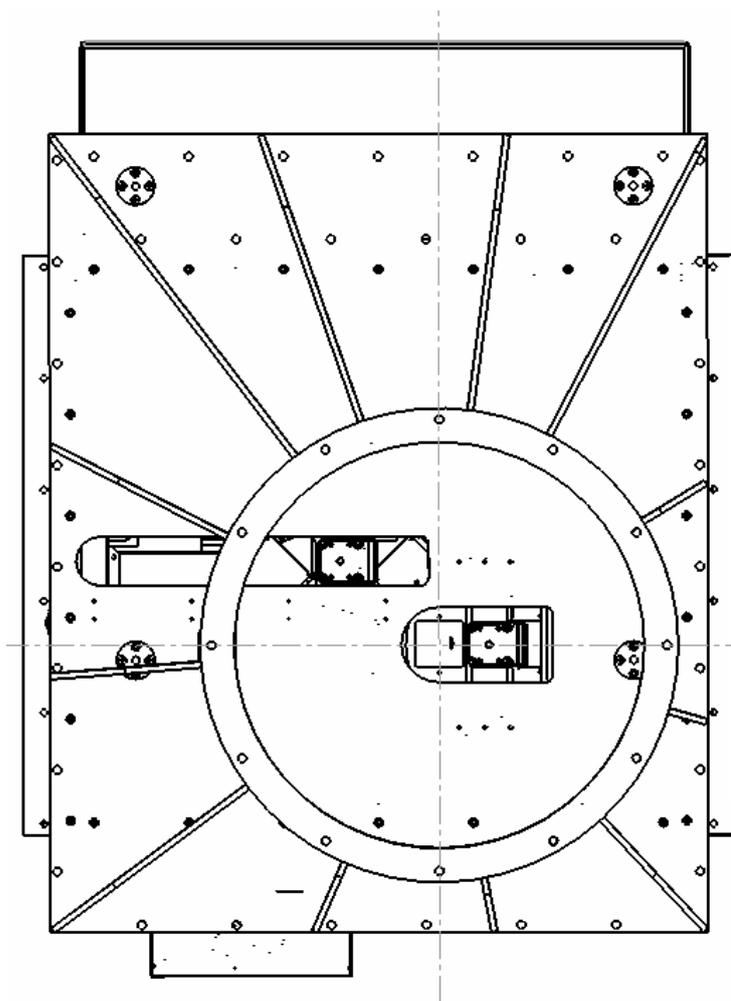


Figura 3.1: Ponto que define o centro do campo do BTFI na entrada do instrumento marcado pelo cruzamento das linhas tracejadas em cinza.

lares que permitem que o laser seja alinhado em relação à estrutura mecânica do BTFI.

Um trilho foi acoplado horizontalmente à estrutura mecânica do BTFI, logo abaixo dele. Sobre este trilho, acoplamos um laser colimado apontando na direção horizontal seguindo quase paralelamente ao trilho. Ainda sobre ele, acoplamos um espelho próximo à saída do laser refletindo-o paralelamente ao trilho mas no sentido contrário. Qualquer ajuste na inclinação deste espelho altera a inclinação do laser, visto que ele está bem próximo da fonte. Um segundo espelho é colocado numa distância mínima de um metro. Qualquer ajuste na inclinação deste segundo espelho terá um efeito menor na inclinação do laser mas o deslocará linearmente. Um terceiro espelho fixo a 45° foi colocado no caminho do laser para jogar a luz para dentro do instrumento BTFI.

Ajustamos o segundo espelho até que o laser cruze o ponto de intersecção entre as duas linhas que definem o centro do círculo formado pelos parafusos que acoplam o BTFI ao

telescópio. Se o sistema não está alinhado, o laser, ao ser refletido no espelho acoplado à entrada do BTFI, volta por um caminho diferente, indicando que ele não estava perpendicular à placa de interface BTFI-telescópio. Se este for o caso, seguimos, então, jogando com os ajustes linear e angular do feixe de luz (ajustes de inclinação do primeiro e segundo espelho) até que o feixe passasse pelo centro do campo do BTFI e até que ele fosse refletido voltando pelo mesmo caminho que percorreu até chegar ao espelho. A figura 3.2 mostra uma foto da montagem utilizada em uma das vezes em que precisamos verificar o alinhamento do instrumento.

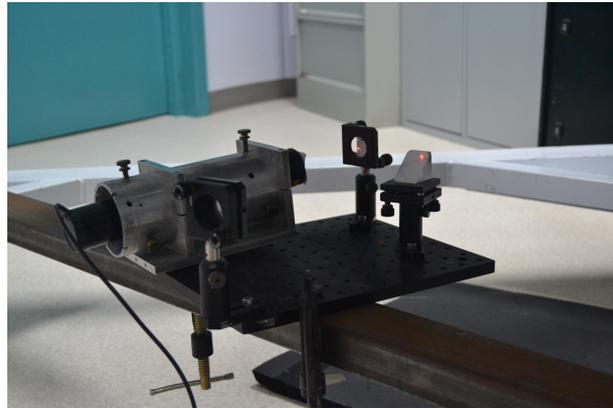


Figura 3.2: Montagem com laser utilizada para o alinhamento óptico do BTFI.

O primeiro elemento óptico que a luz encontra no caminho óptico é a lente de campo. Esta lente é fixa em um barril com um fuso externo que permite que ela seja deslocada para frente ou para trás sobre o eixo óptico (mais perto ou mais longe do laser). Este é o único movimento permitido a ela. De acordo com a tabela 2.1, a distância nominal entre o plano focal e o ponto central da superfície da lente de campo deve ser de 50 mm. Este ponto se sobressai para fora do barril em menos de 1 mm (o valor nominal é de 0,78 mm). Considerando que o foco está exatamente na metade da roda de filtros primária (roda de máscaras), então a distância nominal entre a última superfície da roda secundária e a borda do barril da lente de campo deve ser de 1,59 mm, conforme a figura 3.3.

Depois da lente de campo, temos o primeiro Fabry-Perot mas ele não possui ajuste algum de inclinação. Em tese, o acoplamento mecânico deve ser o suficiente para mantê-lo alinhado perpendicularmente ao eixo óptico. Os efeitos do mal alinhamento deste FP em relação ao eixo óptico definido anteriormente não foram estudados. Estes estudos serão realizados quando os novos FPs que estão em processo de fabricação na França estiverem

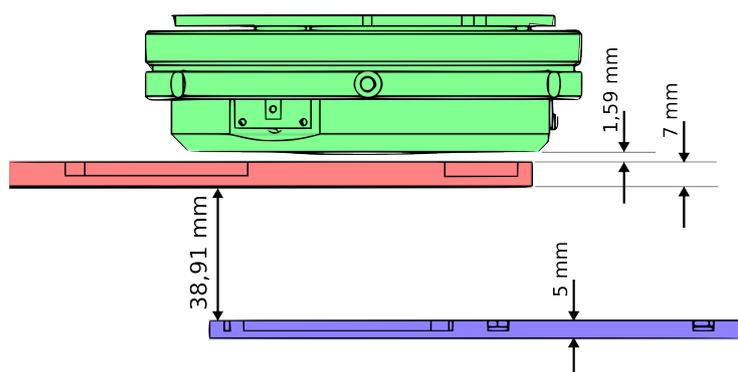
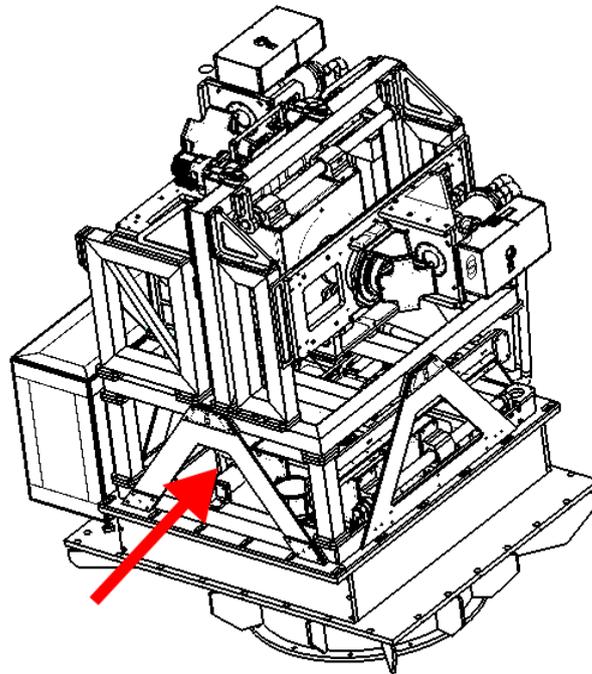


Figura 3.3: Distância nominal entre a lente de campo e a última superfície da roda secundária. Representação simplificada da roda de filtro primária (em azul) e na parte inferior do desenho, da roda de filtro secundária (em vermelho) e no centro, e o barril (em verde) na parte superior que abriga as primeiras lentes de campo. Nesta orientação, a luz que entra no instrumento se propaga na vertical de baixo para cima.

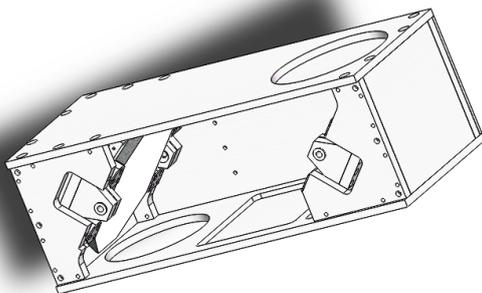
prontos. Quando este Fabry-Perot não está em uso, o que é o caso atual, uma placa de vidro óptico, chamada de *dummy plate*, deve ser colocada em seu lugar para poder manter o caminho óptico constante dentro do instrumento. As superfícies dessa placa são planas e paralelas entre si e ela é espessa o suficiente para minimizar o número de reflexões internas e os testes em laboratórios mostraram que efeito do mal alinhamento desta placa é desprezível.

Após a *dummy plate*, temos os dois espelhos, chamados *foldng mirrors* que dobram o caminho óptico para deixar o instrumento mais compacto (figura 3.4). Cada espelho possui um sistema de ajuste chamado *push-pull*, ou seja, dois parafusos que o puxam em um sentido enquanto outro parafuso o empurra no sentido contrário (figura 3.4c). Assim, com a tensão aplicada nos dois sentidos opostos, o espelho fica firme. Estes dois espelhos são montados dentro de uma caixa fechada no interior do instrumento. Não há espaço disponível entre os *foldng mirrors* para colocar um espelho plano para alinhamento por autorreflexão. Além disso, estes dois espelhos internos atuam em conjunto. Por essas razões, o espelho plano utilizado no alinhamento do BTFI é instalado na saída desta caixa que abriga estes dois espelhos internos. Como o BTFI está posicionado na vertical, a própria gravidade se encarrega de manter o espelho de alinhamento perpendicular ao eixo óptico. Novamente, utilizamos a técnica que separa o deslocamento linear e angular do laser. O *foldng mirror* mais próximo ao espelho de alinhamento é ajustado para alterar a

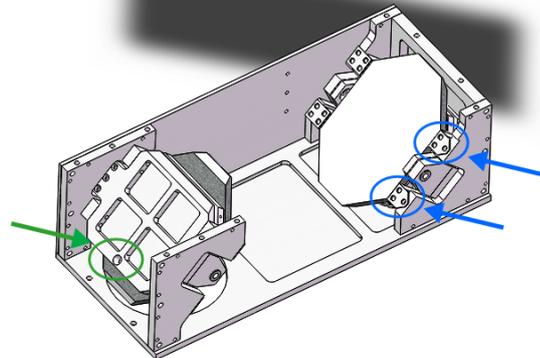
inclinação do laser. O *folding mirror* mais próximo à entrada do instrumento é ajustado para alterar o deslocamento linear do laser.



(a)



(b)



(c)

Figura 3.4: *Folding mirrors* em diferentes ângulos. (a) Localização aproximada dos *folding mirrors*. (b) Caixa que abriga os *folding mirrors*. (c) Parafusos de ajuste *push*, em verde, e *pull*, em azul

Em seguida, temos o colimador que não possui ajustes de inclinação, somente deslocamento linear transversal, conforme destacado nos círculos amarelo e azul-ciano na foto 3.5. A mesma técnica é empregada aqui. Como não há ajuste de inclinação, uma máscara circular contendo um furo no centro foi o suficiente. Entretanto, para uma última verificação, utilizamos a autorreflexão com o espelho de alinhamento apoiado sobre o colimador. Esta

foto também nos mostra os parafusos que seguram o tambor do colimador ao resto da estrutura mecânica. Eles podem ser usados para trazer o colimador mais próximo ao plano focal ou para deixá-lo mais distante.

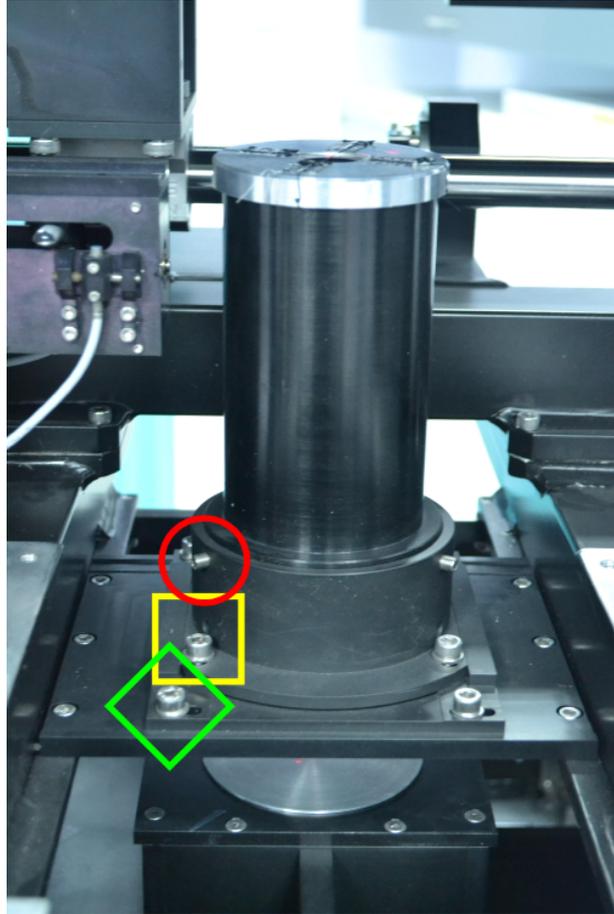


Figura 3.5: Foto mostrando o colimador (o tubo negro no centro da imagem) e os parafusos que o prendem à estrutura metálica do BTFI, destacado com um círculo vermelho, e os ajustes nas direções X e Y, destacados com um losango verde e um quadrado amarelo. Sobre o colimador, está uma das máscaras utilizadas para o alinhamento.

Após o colimador, o caminho óptico do BTFI é bifurcado. A luz pode seguir diretamente do colimador para a Câmera 2 ou pode ser refletida em espelhos acoplados ao iBTF e ir para a Câmera 1. Como o alinhamento das duas câmeras é idêntico, descrevo primeiramente o alinhamento dos espelhos do iBTF. Assumindo que os eixos de rotação do iBTF estão devidamente alinhados, os espelhos do iBTF podem ser alinhados por autorreflexão. Primeiramente o espelho acoplado ao motor M2 deve ser posicionado em 0° (aproximadamente perpendicular ao eixo óptico).

A inclinação dos espelhos acoplados ao iBTF podem ser ajustados de duas formas: a

primeira delas é através da própria rotação dos motores, alterando, assim, a calibração angular deste módulo; o segundo ajuste está disponível na lateral dos suportes das redes de difração (figura 3.6). Este ajuste é um sistema do tipo *push-pull*, semelhante àquele utilizado nos espelhos que dobram o caminho óptico: dois parafusos empurram uma das extremidades do suporte das redes em uma direção enquanto outro parafuso a empurra na outra direção. Após ajustar o espelho acoplado a M2, o deixamos posicionado em 45° para que o laser chegue ao M1.

Para alinhar M1, adotamos a mesma estratégia aplicada em M2: usamos o sistema *push-pull* para alinhar a superfície em uma direção e a calibração angular do iBTF desse motor para alinhar na outra direção.

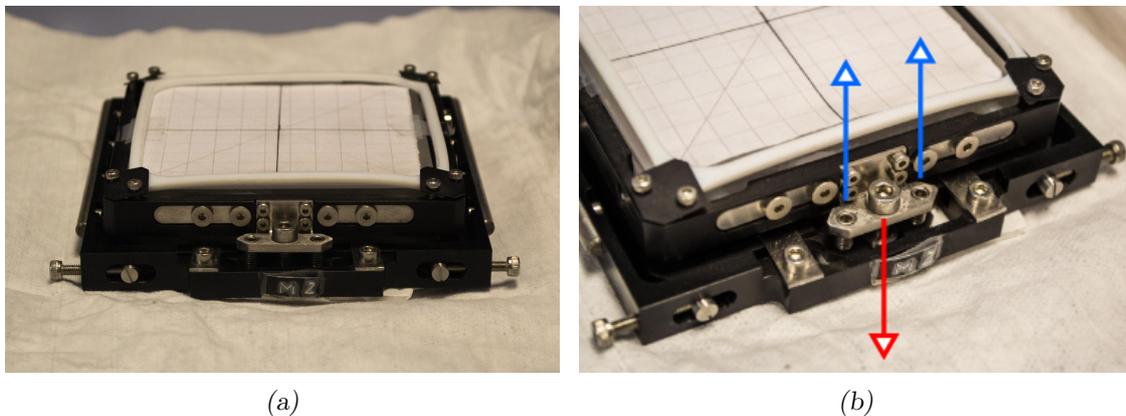


Figura 3.6: Detalhe do sistema *push-pull* para alinhamento dos espelhos e redes de difração do iBTF. (a) Suporte de redes de difração e espelhos do iBTF (b) Detalhes dos parafusos que empurram e puxam um dos lados do suporte fazendo com que a superfície do espelho ou rede de difração se incline para um lado ou para outro.

Logo em seguida, tanto na câmera 1 quanto na câmera 2, temos um sistema óptico de imageamento composto pelas lentes responsáveis pela formação da imagem sobre os detectores e um espelho responsável por dobrar o feixe óptico de modo que ele incida sobre os detectores. O alinhamento destes espelhos exige que os detectores sejam desinstalados. Assim, temos acesso para apoiar o espelho plano de alinhamento na extremidade oposta aos espelhos do sistema imageador. Outra opção seria apoiar este espelho de alinhamento na entrada dos detectores. Entretanto, os detectores também possuem um ajuste de inclinação e não haveria controle sobre as possíveis fontes de desalinhamento.

Estes últimos espelhos do sistema de imageamento também contam com um sistema *push-pull* para o ajuste de inclinação em relação ao eixo óptico, como mostra a imagem

3.7. Uma vez devidamente posicionados, instalamos os detectores de volta às suas posições originais dentro do instrumento e utilizamos o espelho de alinhamento para ajustarmos a inclinação deles em relação ao eixo óptico. Aqui, não foi necessário um ajuste muito preciso conforme pudemos perceber após a verificação final do alinhamento óptico através da aquisição de imagens.

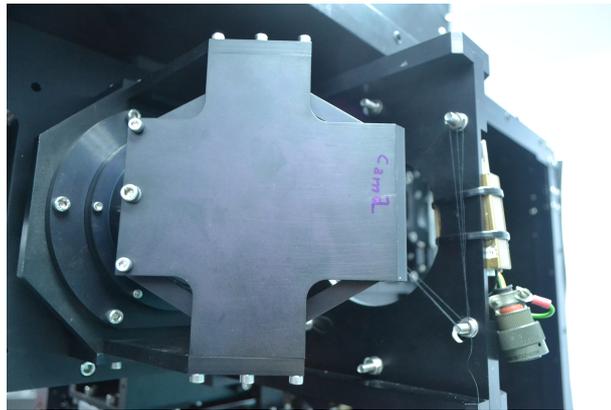


Figura 3.7: Sistema *push-pull* no último espelho que dobra o feixe óptico logo após as últimas lentes imageadoras. Este sistema é idêntico em ambas as câmeras.

A câmera 2 possui um espelho extra antes da *Pupil Wheel* e antes do Fabry-Perot para dobrar uma vez mais o feixe óptico. Este espelho deve ser alinhado logo após o alinhamento dos espelhos do iBTF. Para tal, apoiamos um espelho plano sobre as lentes que compõem o sistema imageador que precede a câmera 1. Este espelho também possui um sistema *push-pull* semelhante aos empregados nos suportes das redes de difração do iBTF (figura 3.8).

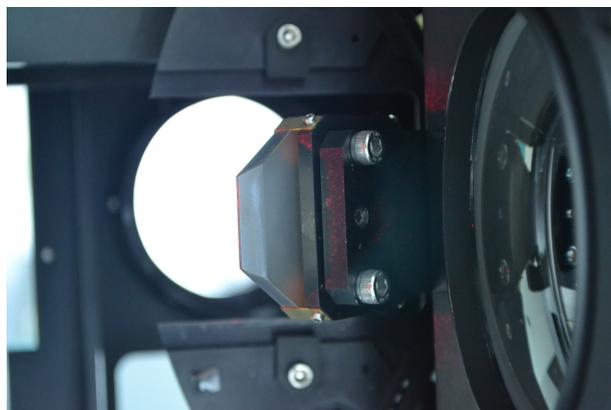


Figura 3.8: Espelho que dobra o caminho óptico que segue em direção à camera 2 (no centro da imagem). Na parte direita da imagem é possível ver a lente de campo da câmera 1 e, ao fundo, a terceira roda de filtros

O alinhamento do sistema como um todo é uma operação que deve ser feita somente nos casos em que algum componente óptico seja removido ou trocado ou caso seja necessário por algum outro motivo. De acordo com nossa experiência, uma vez alinhada, a parte óptica do BTFI permanece estável.

3.1.2 Foco Instrumental

Uma vez que os componentes ópticos dentro do instrumento estão alinhados entre si e em relação ao eixo óptico do telescópio, posicionamos os detectores sobre o plano focal na saída do instrumento e alinhá-los perpendicularmente ao eixo óptico.

O critério utilizado para determinar que os detectores estão bem posicionados foi o tamanho e o formato da *point-spread-function* (PSF) instrumental. No caso, o tamanho da PSF está relacionada com a distância entre a câmera e o plano focal nominal e a formato (simetria) da PSF mostra qual o ângulo entre os detectores e o eixo óptico.

Para poder medir a PSF, instalamos uma máscara na roda primária de filtros com um pequeno orifício de 50 μm de diâmetro chamado *pinhole*. Vale lembrar que esta roda primária está localizada no plano focal do instrumento.

A largura a meia altura medida na câmera 1 foi de 2,48 píxeis, correspondendo a 0,30 segundo de arco no céu, enquanto a largura a meia altura medida na câmera 2 foi de 2,14 píxeis, correspondendo a 0,26 segundo de arco no céu. Este valor foi adotado como aceitável devido ao fato de que a largura-à-meia-altura de uma observação seja limitada pela turbulência atmosférica (*seeing*). Para o SOAR, o *seeing* costuma ser maior que 0,5 segundo de arco. Em outras palavras, a largura da *point-spread-function*, ou PSF, será limitada pela turbulência atmosférica e não pelo instrumento (sem considerar elementos dispersivos como o Fabry-Perot ou as redes de difração). As figuras 3.9 e 3.10 mostram o perfil radial da imagem do pinhole nas câmeras do iBTF (Câmera 1) e do Fabry-Perot (Câmera 2), respectivamente, utilizando a opção 'r' da rotina *imexam* do pacote IRAF de redução de dados. Os valores medidos da FWHM são os apresentados na última coluna na base destas figuras.

3.1.3 Foco nominal versus foco relativo

Numa noite comum de observação, a primeira coisa a ser feita é a calibração do sistema óptico interno do telescópio, realizado pelo operador do SOAR. Ele aponta o telescópio

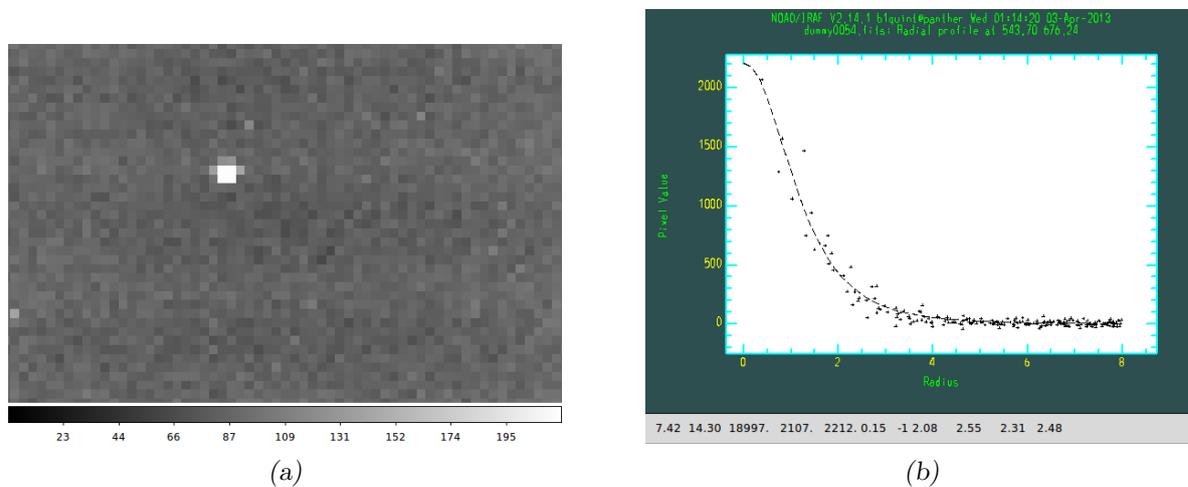


Figura 3.9: Imagem do pinhole obtida usando o iBTF com espelhos no lugar das redes de difração. O FWHM é de 0,3 segundos de arco. (a) Imagem do pinhole na Câmera 1 (iBTF). (b) Perfil radial da imagem do pinhole medido com `imexam` - abscissa em número de píxeis e ordenada em número de contagens.

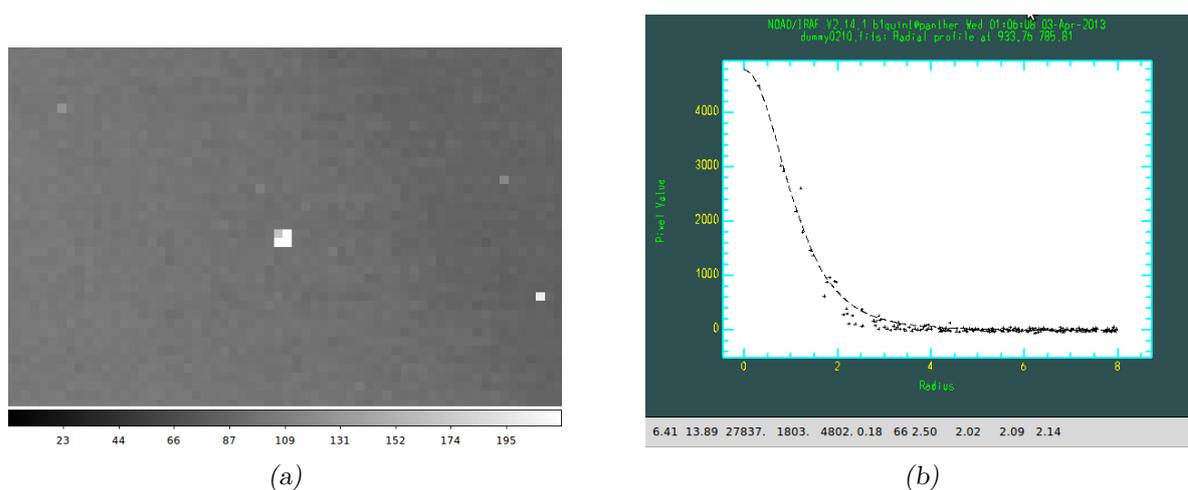


Figura 3.10: Imagem do pinhole obtida usando a Câmera 2 com o Fabry-Perot fora do caminho óptico. O FWHM é de 0,26 segundos de arco. (a) Imagem do pinhole na Câmera 2 (Fabry-Perot). (b) Perfil radial da imagem do pinhole medido com `imexam` - abscissa em número de píxeis e ordenada em número de contagens.

para uma estrela próxima ao objeto de interesse e move o espelho terciário até encontrar a posição com o melhor foco, definido como a posição em que a FWHM do perfil radial de fontes pontuais é a menor obtida. Se o módulo de óptica adaptativa não estiver em uso, a FWHM é limitada pela atmosfera e, normalmente, possui valores acima de 0,6 segundo de arco. Chamamos este de *foco nominal* do telescópio.

De acordo com as medidas que realizamos nas várias observações com o BTFI, sabemos que a posição do espelho terciário que fornece melhor foco instrumental é diferente da posição do foco nominal. A posição relativa em que o espelho terciário deve ser posto em cada observação foi medida diversas vezes e sabemos que esta posição se mantém de uma noite para outra. O valor mais atual foi medido na noite de 2 de abril de 2014.

Nesta noite, o *seeing* medido pelo telescópio era de 1,3 segundos de arco. O objeto alvo utilizado para esta medição foi a região central da nebulosa de 30 Doradus. Normalmente, para este tipo de operação, são escolhidas estrelas isoladas ou aglomerados abertos de estrelas. Entretanto, 30 Doradus era o primeiro objeto na lista de observação daquela noite e optamos por realizar o foco nele para evitar perda de tempo movimentando o telescópio de um ponto a outro.

O foco relativo foi, então, medido enquanto variávamos a posição do espelho terciário. Estas medidas foram feitas utilizando o script `profile-measure.py` e `hocus-focus.py`, ambos escritos pelo Prof. Dr. Fabrício Ferrari (Universidade Federal do Rio Grande, 2012 - comunicação privada). Os valores medidos são apresentados na tabela 3.1 e na figura 3.11. De acordo com eles, a melhor posição encontrada para o espelho terciário foi $+80 \mu\text{m}$. Ajustando estes dados a uma parábola (representada por linhas cheias nesta figura), a melhor posição para o M3 seria de $+60 \mu\text{m}$, aproximadamente.

Os procedimentos descritos nesta seção são repetidos sempre que há alteração na posição de algum dos componentes ópticos internos do instrumento ou quando eles são removidos e recolocados. No caso, os dados apresentados aqui serviram para mostrar que o instrumento pode ser configurado para fornecer imagens com a qualidade para o qual foi projetado.

3.1.4 Degradação da PSF em função do modo de operação do CCD

O primeiro fator a ser considerado ao observarmos com o BTFI é o modo de operação dos detectores. A priori, não sabíamos se alternar entre o modo clássico e o modo EM

Tabela 3.1 - Largura à meia altura medida no processo de busca pela posição do foco relativo do M3

Posição do M3	σ_x	σ_y
-80	27.51	26.14
-60	22.75	21.82
-40	19.70	18.98
-20	15.49	14.89
0	13.65	12.29
20	13.24	11.96
40	10.45	10.40
60	12.64	11.19
80	9.72	9.24
100	11.76	11.00

poderia ou não degradar a PSF das imagens. Sendo assim, na noite de 26 de fevereiro de 2013, obtivemos imagens com o iBTF utilizando dois espelhos nos lugares das redes. Esta configuração é chamada de iBTF-MX. Com ela, capturamos imagens que contém em si a resposta instrumental angular intrínseca ao BTFI e ao telescópio.

O primeiro objeto observado foi o aglomerado aberto IC2602 (Theta Carinae ou Plêiades do Sul) (figura 3.12). A princípio, observamos o aglomerado com o detector operando no modo clássico. A FWHM média medida no campo foi de 7.38 pixels (0.88 arcseg) com desvio padrão de 1.16 pixels (0.14 arcseg). Observando o mesmo campo com o ganho EM ligado obtivemos um FWHM de 11.37 pixels (1.36 arcseg) com desvio padrão ao longo do campo de 4.73 pixels (0.56 arcseg).

Com estas observações, tivemos uma indicação de que existe um efeito de degradação na qualidade da imagem ao usar o modo EM, possivelmente pelo fato do modo EM apresentar melhor desempenho em regiões com menor fluxo de luz. Para poder quantificarmos este efeito, estas comparações deveriam ter sido feitas em laboratório, utilizando pinholes e realizando o tratamento das imagens. Estes testes serão feitos após as câmeras estarem em funcionamento perfeito, o que ainda não é o caso neste momento.

3.1.5 Eficiência operacional do sistema de imageamento simples

Além da qualidade da imagem e da eficiência espectral, um instrumento também precisa ser caracterizado quanto às perdas no tempo de operação. Estas perdas, normalmente

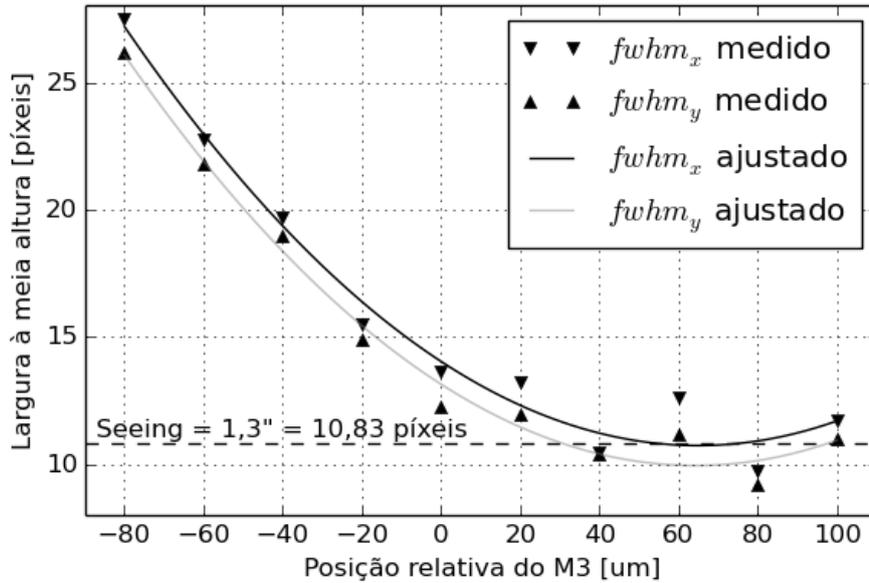


Figura 3.11: Largura à meia altura em função da posição relativa do M3.

chamadas de *duty-cycle-loss*, não devem exceder mais do que 10% .

Além do mais, como o BTFI pode operar como um sistema de imageamento simples ou com o FP/iBTF, este tipo de levantamento da eficiência operacional deve ser realizado em cada um dos modos de operação.

No caso do imageamento simples, o *duty-cycle-loss* está relacionado somente com o tempo de comando de uma exposição e com o tempo de leitura, que pode ser estimado usando uma sequência de imagens BIAS (imagens com tempo de exposição zero), visto que, nelas, o detector não chega a ser aberto.

Sendo assim, nós tomamos uma sequência 60 de imagens e calculamos a diferença média através da diferença entre a hora em que a primeira e a última imagem foram tomadas e o número de imagens. No modo clássico, chegamos a medir 1,1 segundos enquanto no modo EM, chegamos a medir 0,41 s. Este tópico é retomado nas seções 3.2.3 e 3.3.4, onde realizamos o estudo sobre a eficiência operacional do sistema usando o FP e o iBTF, respectivamente.

A partir de agora, trabalharemos com cada um dos módulos avaliando os efeitos sobre a qualidade da imagem e fazendo uma análise da performance espectral de cada um deles. Para que este capítulo seja consistente com o capítulo anterior, iniciaremos nossa análise pelo Fabry-Perot e, em seguida, faremos um estudo sobre o iBTF.

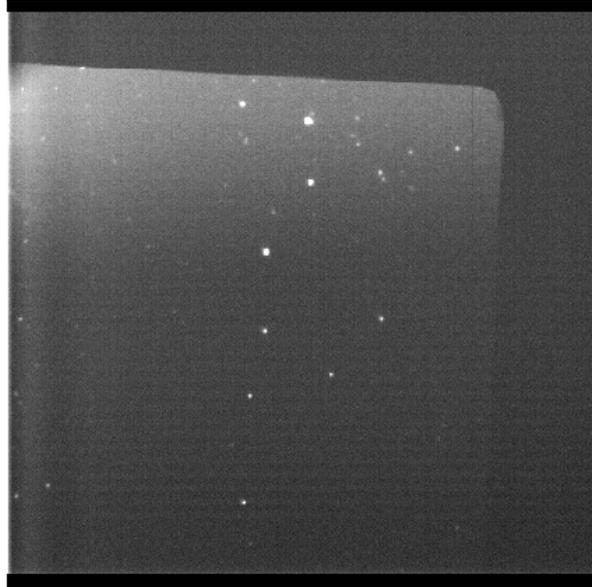


Figura 3.12: Aglomerado IC2602 observado na configuração iBTF-MX e com detector operando em modo clássico. Nenhum tratamento foi feito nesta imagem.

3.2 Fabry-Perot

O BTFI foi desenhado para trabalhar com dois Fabry-Perots de nova geração que estão sendo fabricados no momento na França. Estes Fabry-Perot possuem como diferencial um *gap* nominal variável. Com isso, estimamos que a resolução espectral fornecida por eles varie entre 6000 e 25000.

Entretanto, estes novos FPs ficarão prontos somente no segundo semestre de 2015. Com isso, o BTFI tem utilizado dois FP emprestados: um com *gap* nominal de $44\mu\text{m}$ e resolução espectral nominal de 4500, emprestado da Universidade de Maryland, e outro com *gap* nominal de $200\mu\text{m}$ e com resolução espectral nominal de 20000, emprestado do Observatório Nacional Anglo-Australiano (AAO - *Anglo-Australian Observatory*). Neste texto, chamaremos o primeiro de LR FP (*Low Resolution Fabry-Perot*, ou Fabry-Perot de baixa resolução espectral) e o segundo de HR FP (*High Resolution Fabry-Perot*).

Estes dois FPs são controlados por um dispositivo chamado CS100, fabricado pela empresa Queensgate. No caso, o BTFI também conta com dois controladores emprestados: um deles do AAO e outro da Universidade do Havá (UofH - *University of Hawaii*). Após a instalação de um FP junto a um controlador, o sistema precisa ser balanceado e as placas de vidro devem estar paralelas entre si.

O procedimento utilizado para balancear o sistema é descrito no próprio manual de

instruções do controlador Queensgate CS100. O balanceamento do sistema consiste em encontrar uma condição próxima o suficiente ao paralelismo para que o controlador consiga fechar o ciclo de controle. Este balanceamento é descrito no apêndice A e foi extraído do próprio manual de instruções do controlador CS100. Já o alinhamento das placas é descrito logo a seguir.

3.2.1 Alinhamento das placas do *étalon*

Atualmente, o processo de alinhamento entre as duas placas do *étalon* é inteiramente manual. Por ser um componente antigo, o controlador CS-100 deve ser ligado, pelo menos, uma hora antes do uso do FP para a estabilização da parte eletrônica. É de extrema importância balancear as pontes entre os capacitores que medem a distância entre as placas e os piezos que controlam esta mesma distância nos eixos X, Y e Z (apêndice A).

Para alinhar o FP, precisamos ter acesso visual a ele. Isso pode ser feito removendo o FP de dentro do instrumento ou simplesmente removendo-o do caminho óptico e abrindo a tampa que cobre o BTFI localizada logo acima da caixa eletrônica (instrumento na vertical).

Em seguida, posicionamos alguma lâmpada de calibração ou fonte monocromática atrás do Fabry-Perot, tentando mantê-la próxima ao centro do *étalon*. Normalmente as lâmpadas de Neônio e de Mercúrio são as mais indicadas. Ao olhar a olho nu diretamente para o FP, veremos uma grande quantidade de anéis concêntricos (figura 3.13). É comum colocarmos uma folha de papel ou outro difusor qualquer entre a lâmpada e a superfície do FP para facilitar a visualização dos anéis.

Cada anel corresponde a uma ordem de interferência construtiva para um diferente comprimento de onda. Os anéis sempre estarão centrados no eixo óptico de qualquer sistema. No caso de estarmos olhando diretamente para o FP, o eixo óptico é definido pela linha que sai dos nossos olhos. Em outras palavras, os anéis se deslocam conforme movemos a cabeça de um lado para o outro enquanto olhamos o *étalon* de modo que o seu centro está sempre próximo ao reflexo dos nossos olhos. Este efeito é mostrado usando uma câmera fotográfica na figura em questão onde o eixo óptico é definido pela linha que passa pelo centro das lentes da câmera e incide sobre o FP.

Deslocando a câmera fotográfica de um lado para o outro, os anéis se movem junto com ela. Caso as placas do Fabry-Perot não estejam devidamente alinhadas, os anéis mudarão



Figura 3.13: Fabry-Perot instalado sobre uma mesa e iluminado por uma lâmpada de mercúrio.

de tamanho ao movermos nossa cabeça (ou a câmera) de um lado para outro. Este efeito é representado na figura 3.14.

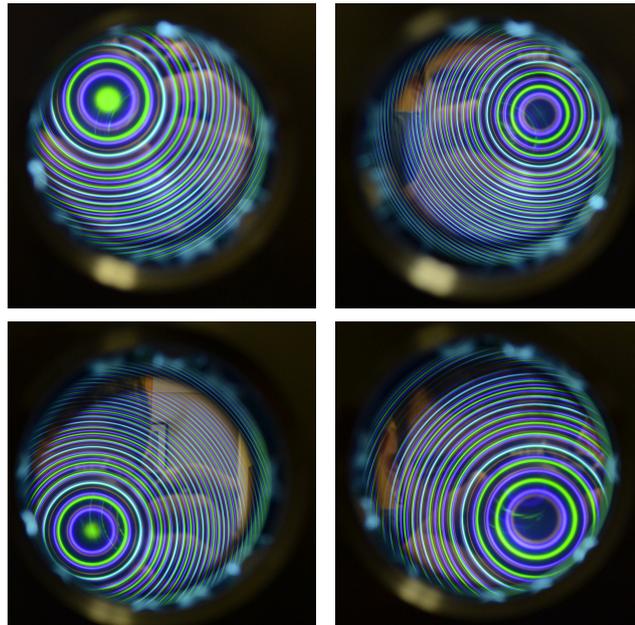


Figura 3.14: Variação no raio dos anéis de interferência devido ao não paralelismo das placas de vidro do Fabry-Perot ao deslocar a câmera de um lado a outro. Note que a região central que era clara na figura à esquerda em cima é completamente escura na figura à direita embaixo.

O alinhamento é realizado através de ajustes nos botões FINE/COARSE do controlador CS100 nas direções X e Y. A direção Y é dada pela reta vertical que passa pela saída

dos cabos do FP e a direção X é perpendicular à direção Y (figura 3.15). Os botões FINE/COARSE na direção Z podem ser utilizados para ajustarem o tamanho dos anéis para facilitar o processo de alinhamento das placas do FP.

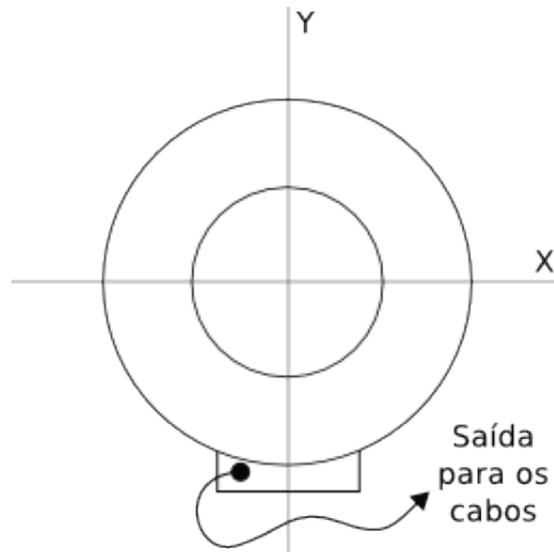


Figura 3.15: Desenho simplificado mostrando o FP e a orientação dos eixos de ajuste do paralelismo do FP em relação à caixa de saída de cabos.

O Fabry-Perot está alinhado quando o observador posiciona-se de modo a ver os anéis dentro do *étalon*, move a cabeça de um lado a outro sempre mantendo o olhar sobre o FP e o raio dos anéis que aparecem lá dentro se mantém constante (figura 3.16). Este é o método empregado também para alinhar Fabry-Perots usados em outros instrumentos como o TAURUS, o CIGALE e o GHASP.

3.2.2 *Bump* espectral - O problema e a solução

Durante observações feitas com ambos os FP's, notamos que a assinatura espectral ao longo do eixo Z dos cubos de dados obtidos com lâmpadas de calibração possuíam uma forma diferente do esperado. De acordo com a teoria apresentada no capítulo 2, seção 2.2.1, o perfil de um espectro obtido com um Fabry-Perot deveria ser simétrico e obedecer à expressão 2.1 que descreve a intensidade luminosa transmitida através de um FP em função da diferença de caminho óptico dentro do *gap* entre as duas placas de vidro. Esta função é mostrada na figura 2.8. Normalmente ela é aproximada por uma função Lorentziana convoluída por uma função gaussiana.

Entretanto, o perfil espectral extraído de cubos de dados obtidos com ambos os FPs

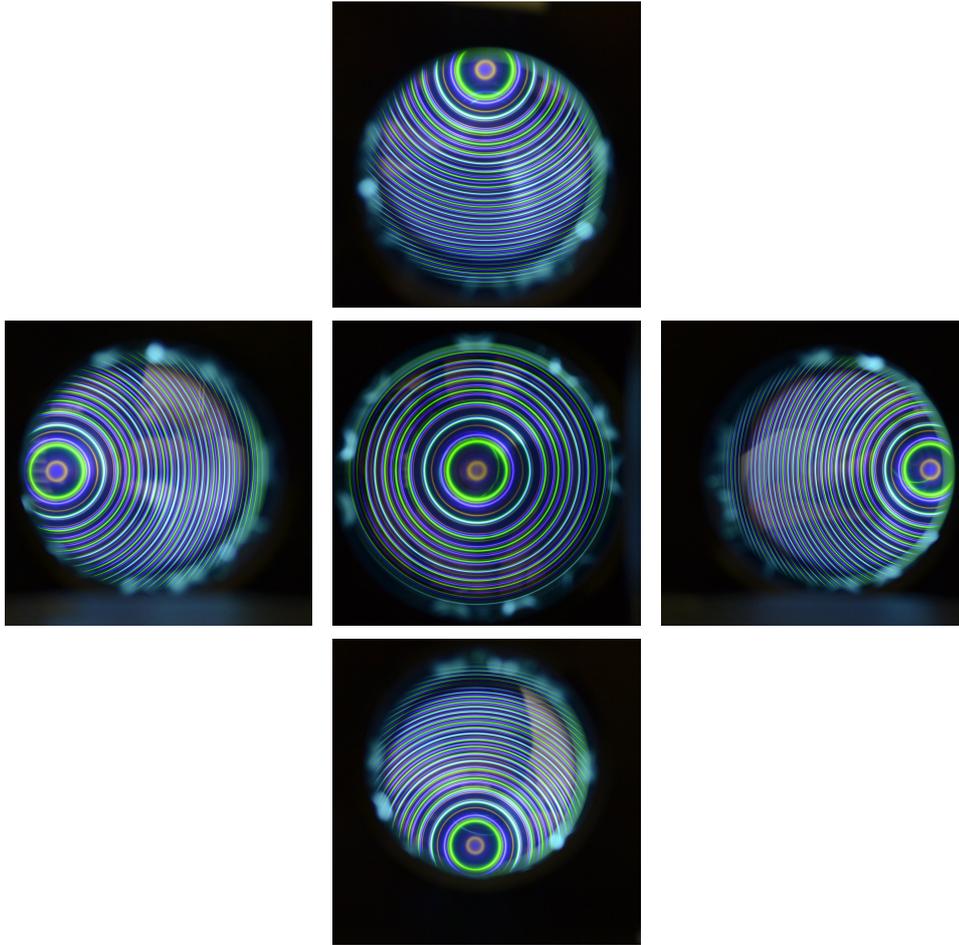


Figura 3.16: Fotos em diferentes regiões do FP mostrando como o raio dos anéis se mantém idênticos em todas as posições.

não era simétrico e possuía uma saliência (*bump*). As figuras 3.17 e 3.18 mostram exemplos de espectros extraídos do centro de cubos de dados obtidos com lâmpadas de calibração usando ambos os FPs no feixe colimado do BTFI. Os *bumps* nestas figuras são indicados com setas.

A causa da existência deste *bump* espectral era, a priori, desconhecida. O fato dele estar presente em dados obtidos com ambos os Fabry-Perots nos levou a acreditar que este efeito poderia estar ligado ao possível mal alinhamento óptico do BTFI. Havia também a hipótese que este *bump* estivesse relacionado às múltiplas reflexões geradas entre os FP e os filtros interferenciais que os sucediam no caminho óptico ou mesmo por luz refletida nas bordas do Fabry-Perot. Uma última hipótese era a de que os dois FP estivessem muito velhos e que o efeito que estávamos vendo era causado pela degradação do filme refletor na cavidade óptica entre as duas placas de vidro de ambos.

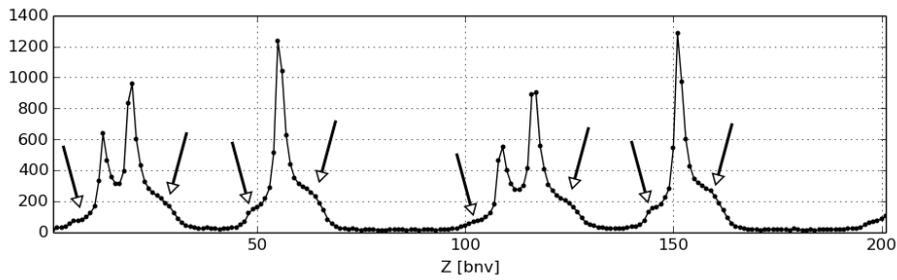


Figura 3.17: Espectro extraído de um cubo de dados obtido com o FP de baixa resolução iluminado por uma lâmpada de Neônio e usando o filtro 6600A-75A no espaço colimado. Dados obtido no dia 24 de agosto de 2013. As setas indicam os supostos *bumps*.

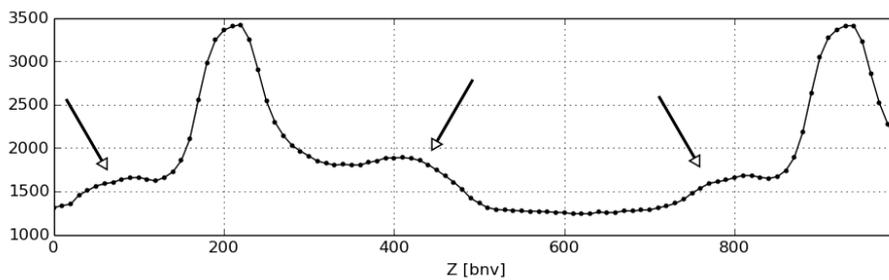


Figura 3.18: Espectro extraído de um cubo de dados obtido com o FP de alta resolução espectral iluminado por uma lâmpada de Neônio e usando o filtro 6600A-20A no espaço colimado. Dados obtido no dia 2 de abril de 2014. As setas indicam os supostos *bumps*.

Reparamos que dados obtidos em diferentes épocas possuíam perfis espectrais diferentes entre si, o que nos levou a crer que este *bump* teria alguma relação com o ângulo de inclinação do filtro em relação ao feixe incidente.

Como a montagem que acopla os filtros à roda de filtros do BTFI não possui um ajuste angular preciso, colocamos o filtro 6600/20 sobre o primeiro suporte de redes de difração holográficas do iBTF, localizado logo após o colimador. Com isto, pudemos obter vários cubos de dados variando o ângulo entre o feixe incidente e a superfície do filtro com precisão. Os resultados são apresentados na figura 3.19.

Esta figura nos mostra que o único efeito gerado nos espectros extraídos dos cubos de dados obtidos com o Fabry-Perot foi o de diminuir a intensidade da linha. Isso, na realidade, era o mais esperado, visto que ao inclinar um filtro interferencial qualquer, o comprimento de onda central do filtro desloca-se para o azul (Löfdahl et al., 2011).

Normalmente, os filtros interferenciais ficam alocados no espaço colimado, próximo à pupila do instrumento. Entretanto, nós os instalamos na roda de filtros secundária,

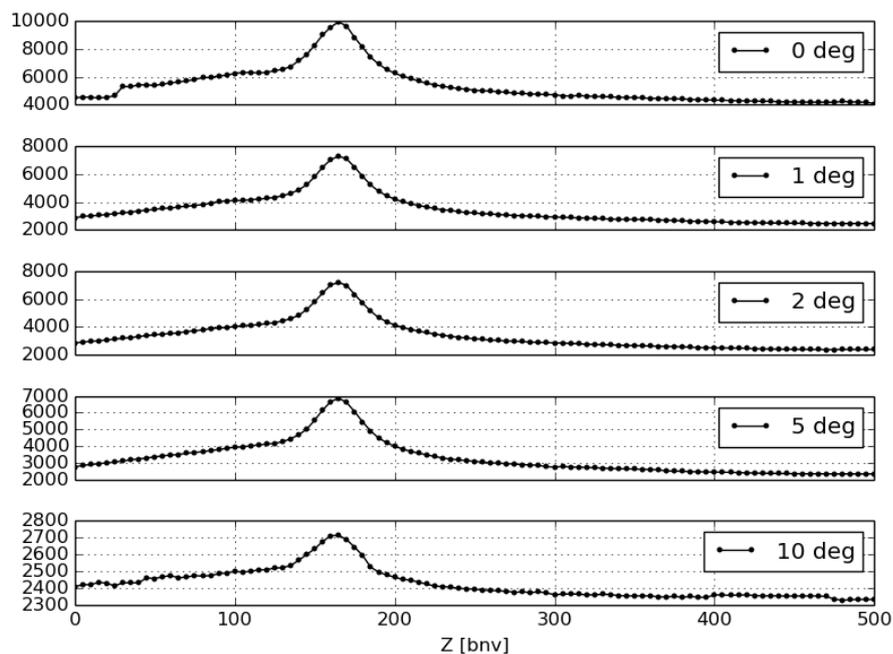


Figura 3.19: Análise do *bump* espectral em função do ângulo entre o feixe incidente e a superfície do filtro interferométrico. De cima para baixo temos os seguintes ângulos: sem inclinação nenhuma (0°), 1° , 2° , 5° e 10° .

localizada logo após o plano focal. A figura 3.20 mostra o resultado. Nela, podemos ver claramente que o *bump* espectral ainda está presente. Após estes dois últimos testes, começamos a suspeitar que a origem do *bump* pudesse não estar relacionada com a presença do filtro interferencial.

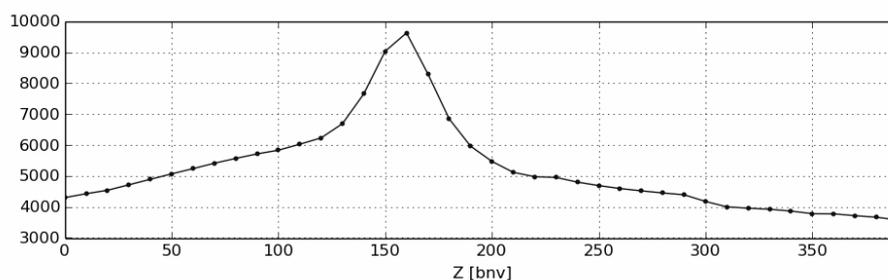


Figura 3.20: Espectro extraído de um cubo de dados obtido com o HR FP iluminado por uma lâmpada de Neônio e usando o filtro 6600.5A-19.3A próximo ao plano focal.

Para comprovar a relação ou não com o filtro espectral, trocamos a lâmpada de Neônio que estava iluminando o interior do instrumento por um feixe de laser de HeNe (tipicamente $\lambda = 6328 \text{ \AA}$) passando por uma lente divergente e por várias camadas de material difusor

(papel vegetal) de modo que a entrada do instrumento fosse iluminada uniformemente. Como o laser é emitido em somente um comprimento de onda dentro de uma banda muito estreita, não foi necessário utilizar nenhum filtro. Novamente detectamos a presença do *bump* (figura 3.21). Isto elimina por completo a hipótese de que o *bump* possa ser causado por filtros interferenciais.

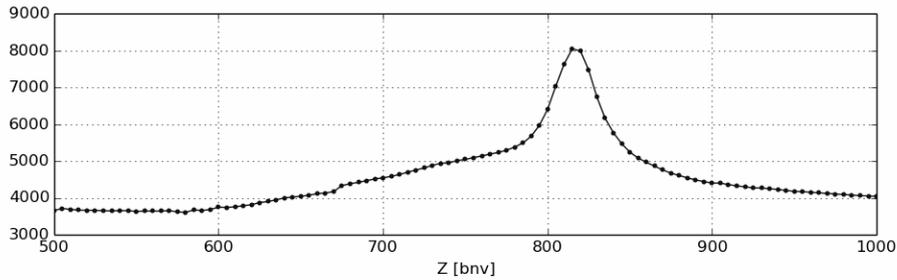


Figura 3.21: Espectro extraído de um cubo de dados obtido com o HR FP iluminado por um laser de HeNe sem a utilização de filtro algum.

Descartada a possibilidade dos filtros interferenciais estarem exercendo alguma influência sobre os dados neste sentido, fizemos vários outros testes até que descobrimos que, ao colocar uma máscara próxima à pupila para cobrir parte do feixe colimado, a assinatura espectral mudava de forma. O primeiro exemplo deste efeito é apresentado na figura 3.22.

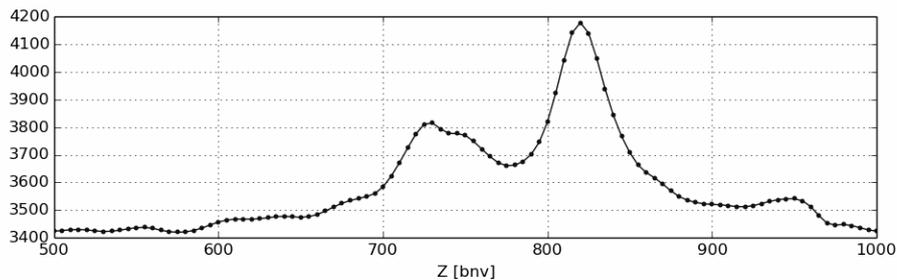


Figura 3.22: Espectro extraído de um cubo de dados obtido com o HR FP iluminado por um laser de HeNe, sem filtro interferométrico com 80% da pupila coberta por uma superfície opaca.

Seguindo com este teste, resolvemos cobrir parte do feixe colimado colocando uma máscara que cobria metade do feixe próximo à pupila em quatro diferentes direções. As figuras 3.23a, 3.23b, 3.23c e 3.23d mostram os espectros extraídos de cubos de dados obtidos com o HR-FP cobrindo a metade inferior, superior, à direita e à esquerda do FP,

considerando o instrumento na vertical e considerando que, ao cruzar o FP, o eixo óptico está na horizontal.

Nestas figuras, podemos perceber que diferentes regiões do FP possuem diferentes efeitos sobre o *bump*. Em alguns casos, a assinatura espectral é minimizada mas não chega a ser completamente eliminada. Em outros, ela fica bem mais evidente, insinuando até a presença de uma outra linha de emissão. Entretanto, sabemos que a fonte é monocromática e que esta outra linha é um fantasma, um defeito cuja origem, até então, era desconhecida.

Neste ponto, tínhamos certeza de que a causa do *bump* não estava relacionada com os filtros interferenciais e nem com comprimentos de onda parasitas. Com isto, acreditávamos que a origem deste *bump* estava relacionada com a degradação dos filmes reflectivos dos FPs ou com problemas de alinhamento óptico do BTFI.

Enviamos, então, um dos Fabry-Perots para a empresa ICOS¹ para que ela fizesse uma análise do estado dos filmes do FP. Segundo a empresa, estes filmes estão em bom estado, apesar dos FPs serem bastante antigos. Este FP também foi testado pelo grupo de instrumentação astronômica de Marseille dentro do instrumento GHASP, onde o *bump* não foi detectado. Neste ponto, estávamos prontos a concluir que a causa do *bump* estava relacionada com a óptica do BTFI.

Outro fato importante não mencionado foi que reparamos, ao realizar o alinhamento dos FPs, que tínhamos dois conjuntos de anéis: um com maior intensidade e outro de menor intensidade. Enquanto o conjunto de anéis mais intensos se comportava exatamente conforme descrito na seção anterior no caso de um FP alinhado, o conjunto de anéis secundário se comportava como um FP não alinhado. A priori, este efeito foi tido como desprezível visto que acreditávamos que este segundo conjunto de anéis era causado por reflexões nas bordas do FP, pois este efeito também foi visto com o FP fora do instrumento.

Percebemos que, nos casos em que o *bump* e a linha principal estavam bem separados, os anéis secundários também mostravam-se presentes. Como se não bastasse, estes anéis não eram perfeitamente concêntricos aos anéis principais fazendo com que acreditássemos que fossem os mesmos anéis vistos a olho nu no processo de alinhamento. Se isto fosse confirmado, estaríamos provando que a causa deste problema não estaria relacionada com a óptica do BTFI.

Este efeito foi confirmado ao obtermos imagens utilizando uma câmera fotográfica Ni-

¹ www.icopticalsystems.com/

kon D3100 com abertura F/5,0 e distância focal de 18 mm e utilizando o sistema de imageamento do módulo de óptica adaptativa do SOAR, respectivamente chamados de SAMI e SAM. Porque o *bump* também estava presente em dados obtidos com estes outros dois sistemas, confirmamos que este efeito não era causado pela óptica do BTFL.

Seguindo a lógica, o único componente comum em todos os casos em que tivemos presente o *bump* espectral, foi o controlador CS100. Seguindo o manual de instruções fornecido pela empresa Queensgate, alteramos os ajustes do ganho e da constante de tempo (GAIN e TIME CONSTANT) no controlador e percebemos que os anéis secundários mudavam de posição. Ademais, percebemos que os anéis secundários desapareciam quando o controlador estava desligado ou quando o ciclo de controle estava aberto. Ao fechar o ciclo de controle, os anéis secundários reapareciam.

No caso, estávamos utilizando o controlador emprestado da UofH. Ajustando o ganho e a constante temporal deste CS100, conseguimos fazer com que os anéis secundários coincidisse com os anéis principais. A figura 3.24a nos mostra o comportamento dos anéis quando o controlador CS100 da UofH estava desligado: os anéis são singulares e bem definidos. Esta figura é idêntica no caso em que este mesmo controlador é ligado mas o *loop* é mantido aberto. Ao fechar o ciclo de controle (ativar o *close loop*), os anéis secundários reapareciam (figura 3.24b).

Mesmo assim, sempre que fechávamos o ciclo de controle do FP com o controlador (*close loop*), os anéis ficavam mais largos do que quando o controlador estava desligado ou com *loop* aberto.

A solução para este problema foi substituir o controlador CS100 havaiano pelo controlador CS100 australiano. Nós o testamos e constatamos que os anéis secundários desapareceram, mesmo com o *loop* fechado (figura 3.25). Montamos um cubo de dados com imagens obtidas com a câmera fotográfica Nikon D3100 para tentarmos verificar a presença do *bump* espectral mas os reflexos entre a primeira lente da câmera fotográfica e as placas do FP atrapalhou o processo.

Pudemos distinguir entre as reflexões entre a lente da câmera e o FP e o *bump* causado pelo controlador ao mover a câmera em frente ao FP. No caso de reflexões parasitas causadas pelas lentes da câmera e pelo FP, o raio dos anéis secundário se mantém constante ao movermos a câmera em frente ao FP enquanto o raio dos anéis secundários gerados pelo controlador muda conforme executamos o mesmo movimento.

Verificamos a solução deste problema utilizando o HR FP dentro do SAM, conforme apresentamos na figura 3.26b. Somente assim, tivemos prova para afirmar definitivamente que o problema do *bump* espectral foi resolvido e confirmamos que o *bump* era um problema causado pelo controlador CS100 da UofH. Acreditamos que estes anéis secundários gerados pelo controlador surgem devido ao mal ajuste no ganho em um ou mais piezos. As tensões aplicadas aos piezos variam idealmente com uma função quadrada de 100 Hz. Um mal ajuste pode acrescentar perturbações oscilantes com uma frequência característica, semelhante a um batimento. Como produto disto, temos os anéis principais, gerados pelo ciclo de controle fechado, e os anéis secundários, gerados pelo batimento em torno da posição nominal. O controlador CS100 do UofH será enviado para conserto enquanto usaremos durante nossas próximas observações o CS100 do AAO.

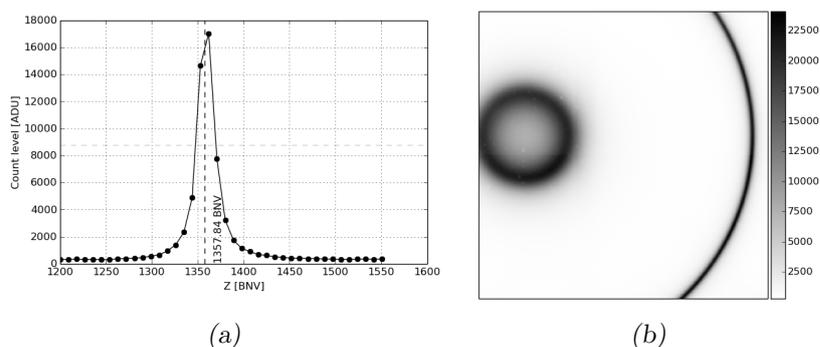


Figura 3.26: Diferentes representações do cubo de calibração obtido com o HR FP usando o SAM, lâmpada de Ne e o filtro 6600/20. Em ambas as representações não vemos a presença do *bump* espectral. (a) Espectro central extraído do cubo de calibração. (b) Canal central extraído do cubo de calibração.

3.2.3 Eficiência do sistema de controle do FP

Mencionamos anteriormente, na seção 3.1.5, sobre a necessidade de avaliarmos a eficiência operacional do instrumento. Por hora, sabemos que o tempo gasto por imagem é de 1,1 segundos para o modo clássico e 0,41 segundo para o modo EM.

Por isso, utilizamos estes dados para estimar a eficiência do sistema de controle do FP. Vale lembrar que os FP são controlados através de rede *Ethernet* via TCP/IP (comunicação entre o computador do KAFKA até o computador com os drivers do FP em LabView) e via porta paralela (comunicação entre o computador com os drivers do FP até o controlador CS100).

Para estimar o tempo gasto com a leitura t_r e o tempo gasto para mudar de um canal

para o outro t_c , tomamos os dados da última noite de observação feita com um dos Fabry-Perots, realizada em abril de 2014, e criamos uma tabela com o número de varreduras por cubo N_s , o número de canais por cubo N_c , o número de imagens por canal N_i , o tempo de exposição por imagem t_{exp} e o tempo total gasto t_{tot} (tabela 3.2). Dada esta tabela, utilizamos a equação 3.1 para fazer um ajuste utilizando o Método dos Mínimos Quadrados implementado em Python usando a biblioteca Scipy deixando os parâmetros t_r e t_c como livres.

Tabela 3.2 - Valores usados na estimativa dos tempos gastos no controle do FP. N_s é o número de *sweeps*, N_c é o número de canais, N_i é o número de imagens por canal, t_{exp} é o tempo de exposição por imagem e DCL é o *duty-cycle-loss*.

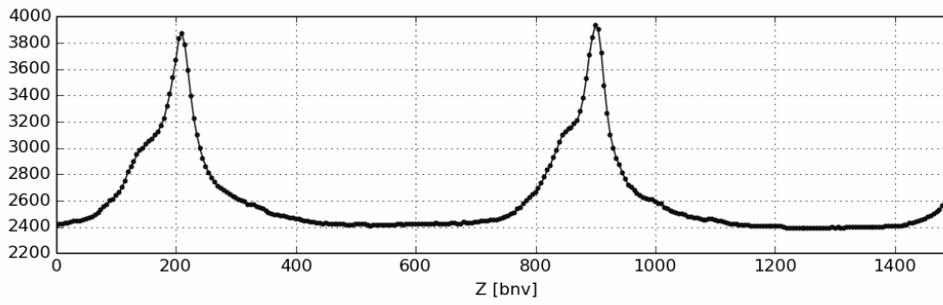
N_s	N_c	N_i	t_{exp}	t_{tot}	DCL
1	101	1	0,5 s	205,791 s	75,5%
1	101	1	5,0 s	627,859 s	19,6%
1	101	1	5,0 s	617,555 s	18,2%
1	101	5	5,0 s	2886,612 s	12,5%
1	101	1	0,5 s	173,186 s	70,8%
1	80	5	5,0 s	2286,596 s	12,5%
1	80	5	5,0 s	2285,519 s	12,5%
1	80	5	2,0 s	1068,496 s	25,1%
1	80	5	2,0 s	1079,087 s	25,8%
1	80	5	5,0 s	2291,542 s	12,7%

$$t_{tot} = N_s N_c N_i t_{exp} + N_s N_c N_i t_r + N_s (N_c - 1) t_s \quad (3.1)$$

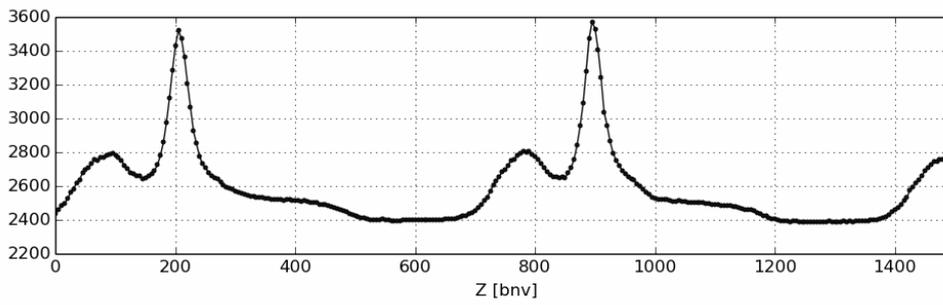
Estimamos, então, que o valor médio perdido a cada troca de canais é de 0,71 s e o tempo gasto por imagem é de 0,57 s. Este resultado é interessante por ser coerente com os valores encontrados na seção 3.1.5 considerando que, nesta tabela, misturamos dados com o modo Clássico e o modo EM. Da mesma forma que o tempo de leitura, este tempo pode variar de acordo com as condições da rede *Ethernet* e do computador usado para controle do BTFI. Neste caso, os dados foram obtidos usando um computador com processador Inter Core i7, 32 Gb de RAM, dois discos rígidos de 1 Tb e rodando Ubuntu 10.04 32 bits (na rede do BTFI este computador é chamado de BTFI-CTRL).

Os dados utilizados desta tabela são dados observacionais e não foram obtidos com o intuito de medir a eficiência operacional do sistema envolvendo o FP. Um resultado

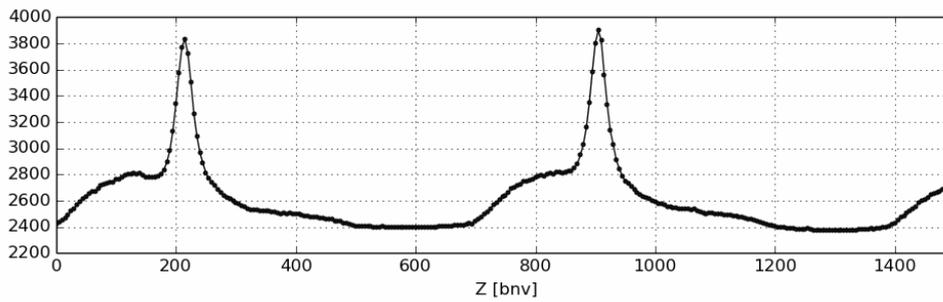
com maior confiabilidade poderia ser obtido caso tivéssemos a oportunidade de realizar estes testes mas, no momento da aquisição, nossas prioridades eram outras. A tabela utilizada mostra que, mesmo os que tempos de leitura e de mudança de canal sejam curtos, eles possuem um efeito cumulativo que compromete bastante a eficiência operacional do sistema. Do ponto de vista operacional, é melhor obter várias imagens tentando maximizar o tempo de exposição e fazer uma só varredura a realizar várias varreduras. Entretanto, várias varreduras sobre um mesmo objeto ajudam a diminuir os efeitos de variação de *seeing* e transparência da noite, o que é importante em observações com o Fabry-Perot, que normalmente são longas (para galáxias, aproximadamente duas horas, no total).



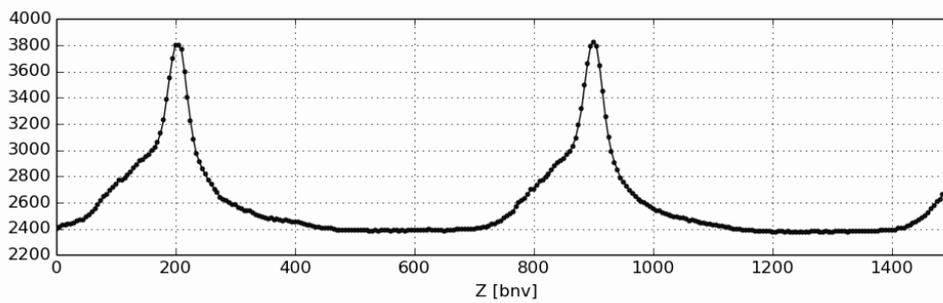
(a)



(b)



(c)



(d)

Figura 3.23: Espectros extraídos do pixel central de um cubo de dados obtidos com o HR FP usando um laser de HeNe, sem filtro interferencial e cobrindo 50% do feixe colimado em quatro diferentes direções. (a) Espectro obtido ao cobrir a metade superior do FP. (b) Espectro obtido ao cobrir a metade inferior FP. (c) Espectro obtido ao cobrir a metade à direita do FP. (d) Espectro obtido ao cobrir a metade à esquerda do FP.

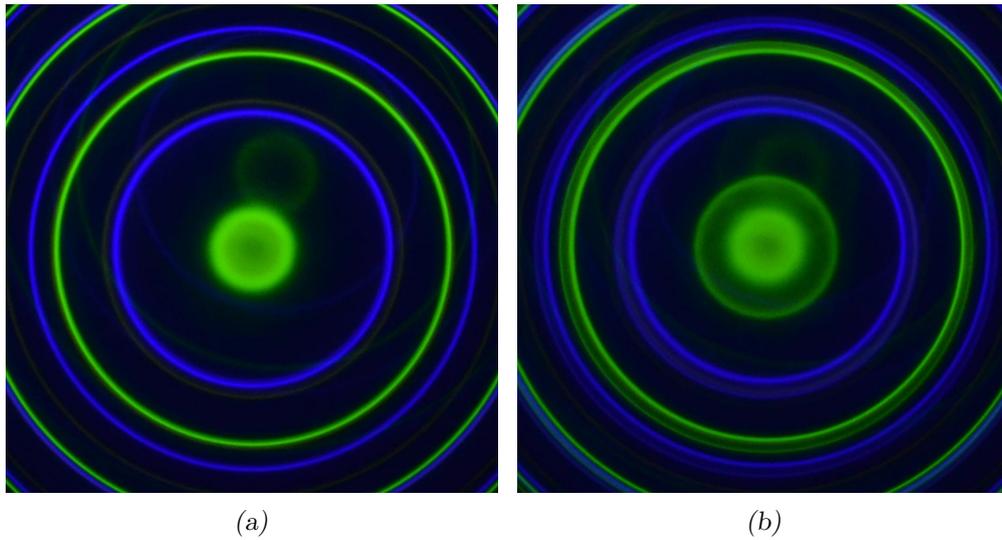


Figura 3.24: Comparação entre os anéis usando o HR FP e o controlador UofH CS100 com loop aberto e loop fechado. Com o loop aberto, vemos somente um conjunto de anéis. Com o loop fechado, além do conjunto principal de anéis, vemos o conjunto secundário de anéis responsável pelo *bump* espectral. Os anéis mais fracos que aparecem de fundo são reflexos causados pelas lentes da câmera fotográfica. (a) Loop aberto. (b) Loop fechado.

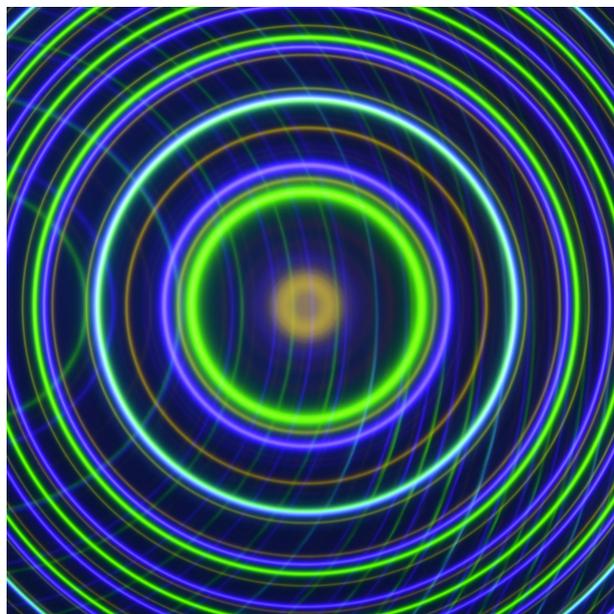


Figura 3.25: HR-FP controlado pelo AAO1 CS100 - Loop fechado. Os anéis formam um único conjunto mostrando a ausência do *bump* espectral. O conjunto de anéis mais fraco que aparece completamente fora de centro com o conjunto principal é fruto de reflexos nas lentes da câmera fotográfica.

3.3 *iBTF*

Conforme explicado no capítulo 2, os principais componentes que compõem o *iBTF* são as redes de difração holográficas. Sendo assim, iniciamos a caracterização do *iBTF* fazendo uma análise das redes adquiridas justificando a escolha dos parâmetros de fabricação e levantando os problemas que cada uma delas apresentou.

Em seguida, apresentaremos uma análise da montagem mecânica e do sistema de controle visando avaliar sua confiabilidade e eficiência operacional.

3.3.1 *Limitações na fabricação das redes VPH*

Apesar de ser uma tecnologia relativamente recente, as redes de difração holográficas já são produzidas por diversas empresas especializadas em fabricar componentes ópticos. Além disso, elas são utilizadas em diversos instrumentos como o espectrógrafo Goodman, instalado no SOAR, entre outros. De acordo o modelo de Kogelik (Kogelnik, 1969), a resolução espectral das redes VPH e a dispersão angular são diretamente proporcionais à espessura d do filme holográfico e inversamente proporcional à distância Λ entre dois planos com mesmo índice de refração, conforme as equações 3.2 e 3.3. Nelas, $\Delta\lambda_{\frac{1}{2}}$ denota a largura à meia altura no domínio espectral e $\Delta\theta_{\frac{1}{2}}$ denota a largura à meia altura no domínio angular.

$$R = \frac{\lambda}{2\Delta\lambda_{\frac{1}{2}}} \approx \frac{d}{\Lambda \cot \theta} \quad (3.2)$$

$$2\Delta\theta_{\frac{1}{2}} = \frac{\Lambda}{d} \quad (3.3)$$

Devido às restrições mecânicas e operacionais, os ângulos de incidência sobre as VPH existentes no interior do *iBTF* estão limitados entre 25° e 45° e isso limita os possíveis valores de Λ . Os parâmetros de entrada que fornecemos para uma determinada empresa desenvolver uma rede de difração englobam o comprimento de onda λ cujo o máximo de eficiência da difração deve estar localizado quando o ângulo de incidência θ é de 35° , o ângulo operacional médio. Analisando cuidadosamente a equação de Bragg para redes holográficas (equação 2.14), temos que estes dois parâmetros são fixos. O índice de refração n depende do material que o VPH é feito e, então, também é considerado um parâmetro fixo. O ângulo ϕ , que define a direção de modulação do índice de refração, também é fixo

no momento da fabricação da rede. Além do mais, sempre trabalharemos com a primeira ordem m de difração e, então, este parâmetro também é fixo.

No final, com todos estes parâmetros fixos, a frequência linear f da modulação do filme holográfico acaba por ser, também, um parâmetro fixo e, conseqüentemente, Λ também o é, visto que um parâmetro é o inverso do outro. Com isso, a única forma que encontramos de obter resoluções espectrais mais elevadas para as redes que trabalham dentro do BTFI foi elevando a espessura do filme holográfico. De todas as empresas que fabricam redes de difração atualmente, somente duas delas se colocaram à disposição para tentar confeccionar redes de difração com tais especificações: Ondax Inc e Wasatch Photonics.

Normalmente a caracterização das redes de difração é feita somente no domínio espectral. Entretanto, como o BTFI as usa para fornecer imagens, temos que avaliar também a influência destas redes no domínio angular. As redes de difração obtidas com a Ondax e com a Wasatch são bastante diferentes entre si. Cada uma delas tem suas peculiaridades e, por isso, fizemos uma análise da qualidade das redes de acordo com a empresa que as forneceu.

3.3.2 Redes de difração Wasatch

Vamos começar nosso estudo pelas redes obtidas com a Wasatch. Esta empresa foi responsável por fabricar redes de difração de baixa resolução. As VPHs da Wasatch são feitas usando *dichromated gelatin* (um tipo especial de gelatina usada em holografia). Atualmente o BTFI possui quatro redes de difração operando em modo de transmissão. Todas as redes possuem frequência linear nominal (Λ) de 2300 mm^{-1} e espessura de $60 \mu\text{m}$. Visto que elas são utilizadas aos pares, nós as chamamos de redes 1A, 1B, 2A e 2B.

Assim que conseguimos estas redes, tivemos a oportunidade de ir ao LNA para medir a eficiência espectral delas. Estas medidas foram feitas usando um sistema desenvolvido por Flavio Ribeiro em seu trabalho de mestrado (Ribeiro, 2010). Os dados para a determinação da eficiência das redes foram obtidos em 25° , 30° , 35° e 45° para cada uma das redes em diferentes comprimentos de onda. Entretanto, ao realizar a redução dos dados, percebemos que nem todas as varreduras puderam ser aproveitadas. Os dados que conseguimos reduzir são apresentados nas figuras 3.27, 3.28, 3.29 e 3.30. Nelas, os pontos representam a eficiência medida em cada comprimento de onda para um determinado ângulo fixo e as linhas representam um ajuste para estimar os principais parâmetros espectrais apresenta-

dos na tabela 3.3. A tabela e as figuras nos mostram que a eficiência destas redes não é tão alta quanto o desejado. Isto foi discutido com a empresa e novos pares de redes estão em fabricação.

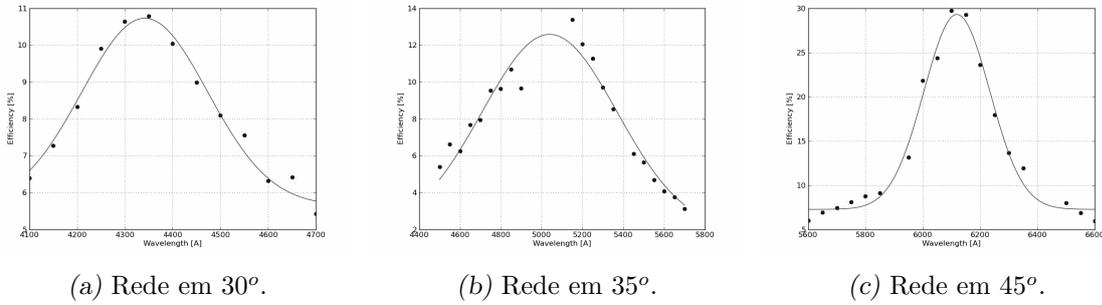


Figura 3.27: Eficiência espectral da rede Wasatch 1A em diferentes ângulos. (a) (b) (c)

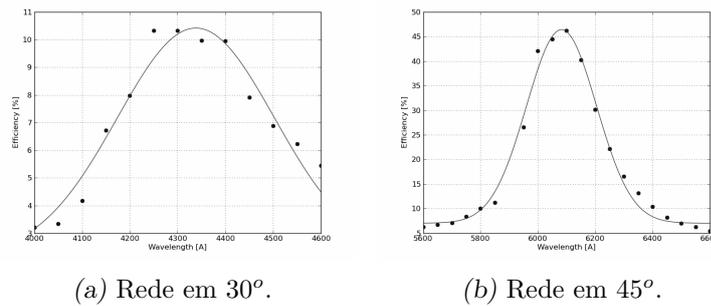


Figura 3.28: Eficiência espectral da rede Wasatch 1B em diferentes ângulos.

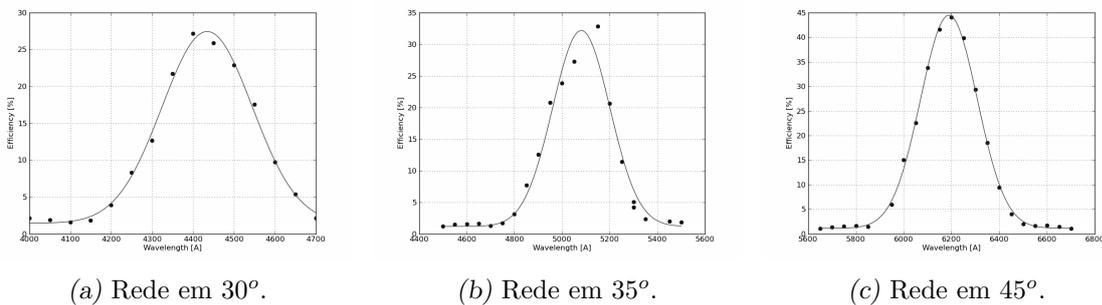


Figura 3.29: Eficiência espectral da rede Wasatch 2A em diferentes ângulos.

Além disto, essas redes foram feitas usando vidro comum como substrato. Isso, é claro, faz com que a qualidade da imagem do BTFI seja degradada².

² ver maiores explicações sobre a qualidade de imagem na seção 3.1.3

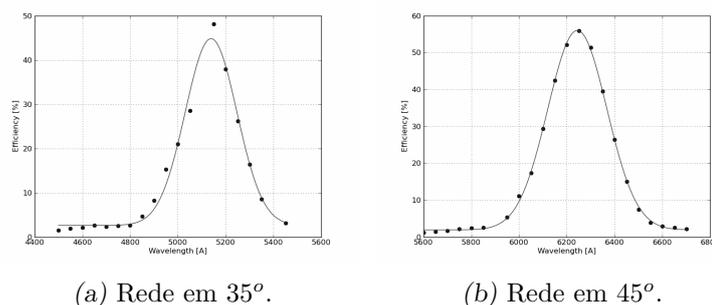


Figura 3.30: Eficiência espectral da rede Wasatch 2B em diferentes ângulos.

Tabela 3.3 - Redes Wasatch - Parâmetros espectrais encontrados. λ_{peak} é o comprimento de onda cuja eficiência é máxima para cada ângulo de incidência e λ_{min} e λ_{max} denotam os comprimentos de onda mínimo e máximo a serem usados no iBTF encontrados através de extrapolação linear.

Rede	Ângulo	λ_{peak}	Eff _{max}	FWHM	$\lambda_{min} - \lambda_{max}$
Rede 1A	30°	4341,80 Å	5,088%	310,154 Å	3801,96 Å– 6141,16 Å
	45°	6118,68 Å	22,040%	270,398 Å	
Rede 1B	30°	4338,24 Å	8,259%	387,211 Å	3756,81 Å– 6082,51 Å
	45°	6082,51 Å	39,393%	287,078 Å	
Rede 2A	30°	4434,33 Å	26,024%	259,841 Å	3880,035 Å– 6202,23 Å
	45°	6189,10 Å	43,375%	279,035 Å	
Rede 2B	35°	5138,74 Å	42,249%	254,577 Å	4033,52 Å– 6243,96 Å
	45°	6243,96 Å	54,131%	291,204 Å	

Ao obter uma imagem de uma estrela com o iBTF, usando as redes de difração Wasatch e o modo clássico CCD (sem EM) da câmera 1, não obtivemos dado algum. Isso porque uma estrela, apesar de ser uma fonte forte, emite em vários comprimentos de onda. Quando observamos com as redes de difração, filtramos grande parte desta luz e a intensidade de luz que chega no detector é muito menor. Devido a isso, esta comparação deveria ter sido feita em laboratório usando lâmpadas de calibração e máscaras na roda primária. Entretanto, no momento em que realizamos estes testes, não dispúnhamos as máscaras apropriadas. Por isso, esta comparação teve que ser feita no céu utilizando o modo EM da câmera 1.

Os resultados obtidos em observações usando redes e espelhos no iBTF são apresentados nas figuras 3.31 e 3.32. As duas imagens foram obtidas em dias consecutivos e em diferentes

objetos, o que poderia justificar um possível alargamento na PSF. Entretanto, a variação de *seeing* não justifica mudanças no formato da PSF, principalmente o alargamento na direção horizontal. Esta forma alongada da PSF é vista com todas as redes de transmissão. Sabemos que este alongamento não está relacionado com a dispersão da luz, visto que ela acontece na vertical enquanto o alongamento é predominantemente horizontal. Apesar de sabermos que a deformação da PSF está relacionada com o uso das redes Wasatch, não chegamos a realizar uma investigação mais a fundo para descobrir o porquê dela ter este efeito. Acreditamos não valer a pena continuar com estes testes com estas redes de vidro comum. Novas redes Wasatch com o vidro apropriado estão em confecção.

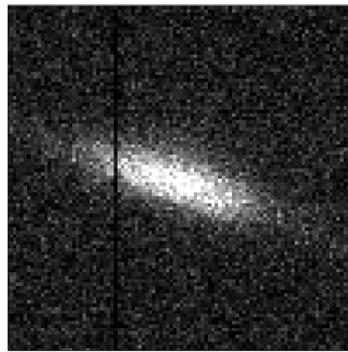


Figura 3.31: PSF no céu do iBTF usando as redes Wasatch 1A e 1B. Esta é uma ampliação da imagem de uma das estrelas observadas (100 x 100 pixels). A largura à meia altura encontrada usando `imexam` do IRAF foi de 15.85 pixels para um perfil gaussiano e uma elipsidade de 0.62. O perfil radial não é mostrado devido à alta deformação na PSF da estrela (o programa de medição de FWHM não converge).

Além das redes com $60\mu\text{m}$, também adquirimos um par de redes com $42\mu\text{m}$ de espessura. Utilizando novamente um *pinhole* com $50\mu\text{m}$ de diâmetro, medimos o efeito das redes de difração na PSF. Os resultados das medidas são apresentados na tabela 3.4. Nela, comparamos a FWHM obtida usando espelhos (iBTF-MX) e usando redes de transmissão (iBTF-TX). Além disso, também comparamos a degradação na PSF devido ao fato de usarmos o detector no modo clássico (CCD) ou com o ganho EM ligado (EMCCD).

Anteriormente, o *pinhole* tinha um FWHM de 2.48 píxeis ($0,3''$). Agora, a PSF possui uma largura de 3.30 píxeis ($0,4''$), um aumento de 33%. Este alargamento da PSF provavelmente está associado ao fato de haver uma maior variação nas contagens devido ao ganho que cada píxel recebe.

Como não conseguimos dados válidos no LNA para caracterizar esta rede espectral-

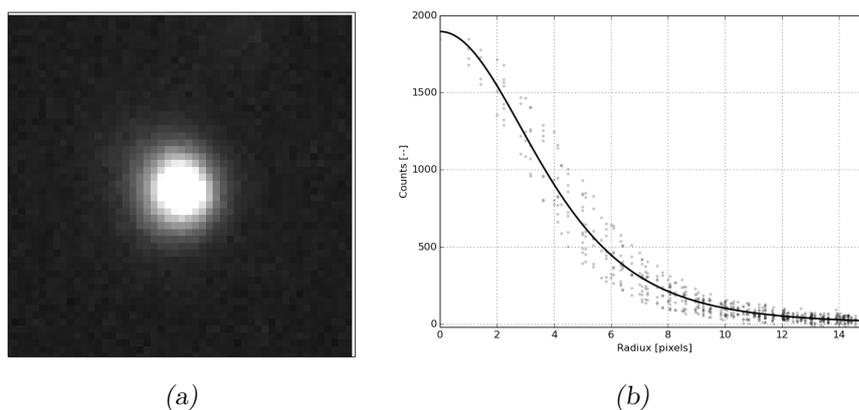


Figura 3.32: PSF no céu do iBTF usando espelhos. À esquerda, temos uma ampliação de uma das estrelas observadas (50 x 50 pixels). À direita, temos o perfil radial desta estrela e uma linha representando um ajuste com o perfil de Moffat. A largura-à-meia-altura encontrada foi neste ajuste foi de 7,013 pixels ou 0,84". (a) Imagem da estrela em que a PSF foi medida. (b) Perfil radial da PSF ajustado por um perfil Moffat.

Tabela 3.4 - Comparação na FWHM da imagem do pinhole obtida na Câmera 1.

	FWHM (píxeis)	FWHM (arcseg)	Aumento
iBTF-MX / CCD	2.48	0.30	
iBTF-TX / CCD	2.62	0.32	8%
iBTF-TX / EMCCD	3.30	0.40	33%

mente, obtivemos com o BTFI um cubo de calibração utilizando uma lâmpada de HgAr. O espectro extraído do centro do cubo é apresentado na figura 3.33. Acreditamos, a princípio, que as duas linhas mais fortes à direita correspondam à 5460,735 Å e 5677,105 Å. Se este for o caso, a largura à meia altura, em ângstroms, é de 135 Å (2.8 z-píxeis), correspondendo a uma resolução espectral de aproximadamente $R = 40$.

A existência das duas linhas à direita do espectro colocaram em dúvida a identificação dos comprimentos de onda apresentados. Não sabemos ainda de onde estas duas linhas vêm e se elas possuem alguma relação com o mesmo problema de ajuste angular entre as duas redes, problema que acontece ao usar as redes Ondax em modo *double-pass* e que explicaremos na seção a seguir.

No momento, novas redes com vidro óptico foram encomendadas e esperamos que elas tenham um melhor comportamento devido à troca do substrato da rede. As redes de transmissão que temos em posse apresentam um rendimento muito baixo para poderem ser usadas em observações.

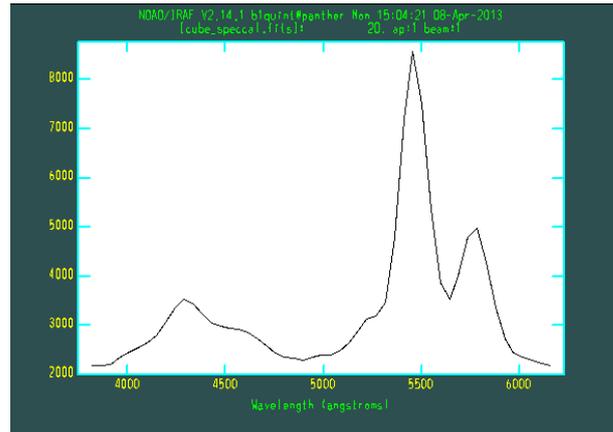


Figura 3.33: Espectro extraído do píxel central do cubo obtido usando uma lâmpada de HgAr e com o iBTF em modo de transmissão.

3.3.3 Redes de difração Ondax

Conforme mencionamos mais acima, a segunda empresa que forneceu as redes de difração VPH para o BTFI é chamada Ondax. As redes obtidas com esta empresa são feitas de um material chamado *doped-glass* (vidro dopado). Os hologramas feitos desse material podem ser mais espessos. No caso, as redes fabricadas com a Ondax possuem uma espessura nominal de 0,6 mm o que, em teoria, nos permite alcançar uma resolução espectral de quase 2000.

Como o intervalo espectral em que cada rede atua é definido pelo comprimento de onda selecionado em 25° e 45° e como isto depende somente da frequência linear das redes e não da espessura delas, uma rede com uma resolução espectral muito alta exige um alto nível de controle mecânico para controlar o ângulo de incidência. Os motores responsáveis por mover os dois estágios rotacionais que seguram as redes têm codificadores de 18 bits (262144 passos por volta). Isso dá uma resolução angular de menos de $0,0014^\circ$ por passo. Entretanto, nossa experiência é que o iBTF opera com uma resolução angular aproximada de $0,05^\circ$, devido ao sistema de controle atual.

Apesar de ser um valor aparentemente baixo, se considerarmos uma rede de difração cobrindo de 4000 \AA até 7000 \AA com resolução espectral de 1000 ($\text{FWHM} = 5 \text{ \AA}$), precisaríamos de uma resolução angular menor do que $0,03^\circ$.

Por causa desta limitação, as redes da Ondax deveriam operar no modo de reflexão, restringindo, assim, o intervalo espectral. Desta forma, a resolução angular atual dos motores do iBTF é suficiente para escanear uma linha através do campo usando a amos-

tagem de Nyquist. No caso, o valor nominal para o comprimento de onda central (ângulo de incidência de 35° no ar) foi de 6750 \AA . Usando a lei de Snell e a lei de Bragg junto com o índice de refração do substrato fornecido pela Ondax ($n = 1,495$), calculamos que a frequência linear deveria ser de $1359,52 \text{ mm}^{-1}$ e que o intervalo espectral no centro do detector deveria ir de 6364 \AA (45°) à 7044 \AA (25°).

A primeira restrição das redes da Ondax é que, por causa da sua espessura, ainda não há tecnologia disponível para se fazer um filme holográfico estável e constante sobre uma grande área. Por causa disso, decidimos tentar fazer um mosaico de redes de difração. A ideia foi simplesmente depositar nove filmes holográficos sobre um único substrato, cobrindo a pupila toda do BTFI e reduzir o vignetting. E foi assim que o primeiro lote de VPHs foi feito em regime “*shared risk*” com a empresa Ondax.

O problema é que cada filme tem um vetor de rede levemente desalinhado um com o outro. Graças à magnificação da câmera imagadora do BTFI, até mesmo a menor diferença no alinhamento dos filmes holográficos nas várias redes que compõem o mosaico faz com que a luz foque em lugares diferentes. Por causa disso, quando o mosaico de redes foi usado, ao invés de obter uma única imagem de um único orifício no plano focal, obtivemos uma imagem acompanhada de fantasmas muito próximos um do outro, conforme mostrado na figura 3.34a.

Para verificar se a origem destes fantasmas estava mesmo relacionada com o mosaico de redes, construímos uma máscara que deixa a luz passar apenas em um único elemento do mosaico. Assim, todos os fantasmas relacionados ao mosaico da rede sumiram, conforme podemos ver na figura 3.34b. Entretanto, um par de fantasmas relacionado pelas reflexões nas superfícies da rede ainda permaneceu. Com isto, as redes seguintes obtidas com a empresa Ondax eram redes monolíticas, compostas por somente um elemento. Identificamos estas redes pelos números de série 16809 e 16819.

Essas novas redes de difração tinham os mesmos parâmetros de construção. Porém, devido às restrições tecnológicas na sua fabricação, seu tamanho era limitado a 37×30 mm. Isso reduziu a abertura em mais de 80% e, portanto, a eficiência do instrumento.

Entretanto, diminuir o tamanho das redes não eliminou os fantasmas causados pelas reflexões nas superfícies das redes. Em teoria, ambas as redes possuem uma camada anti-reflexo que faz com que menos que 0,5% da luz seja refletida. No caso de uma fonte monocromática, estes fantasmas não são muito fortes. Entretanto, no caso de uma fonte

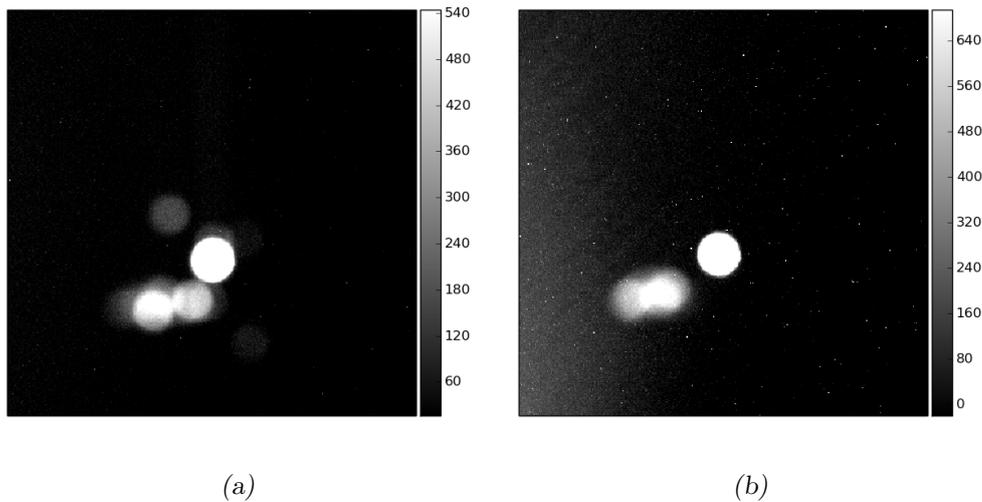


Figura 3.34: Comparação das imagens obtidas usando (à direita) ou não (à esquerda) uma máscara junto ao mosaico de redes de difração. Os fantasmas na imagem à esquerda surgem devido ao mal alinhamento dos filmes holográficos das primeiras redes VPH fabricadas pela empresa Ondax. Já os fantasmas na imagem à direita provavelmente são causados devido às reflexões na primeira e última superfície da rede não coberta pelas máscaras. (a) Sem máscara. (b) Com máscara que só deixa passar luz sobre o pedaço central do mosaico.

com uma componente contínua, estes fantasmas passam a ter uma contribuição significativa e podem prejudicar as observações. No caso, se analisarmos uma única imagem onde um determinado objeto com linhas de emissão esteja justamente na região do detector onde há um pico de reflexão por difração da rede, quase não podemos perceber a presença do fantasma (figura 3.35a).

Contudo, colapsando o cubo, podemos ver como a contribuição da reflexão das superfícies das redes pode ser significativa ao considerarmos o espectro integrado (figura 3.35b).

Para solucionarmos este problema, teríamos que alinhar a direção do vetor da rede de difração \vec{K} , que define a direção com que ocorre a modulação do índice de refração do filme holográfico (ver seção 2.3), com o vetor normal à superfície dos substratos da rede. Entretanto, o processo de fabricação das redes de difração não permite que estes dois vetores sejam alinhados com a precisão necessária para que os fantasmas e as imagens geradas por difração coincidam.

Sendo assim, tentar explorar o outro extremo. Pedimos para a empresa Ondax fabricar redes de difração de modo em que o desalinhamento entre estes dois vetores fosse grande o suficiente para que, se a luz relacionada com a difração iluminasse todo o campo do

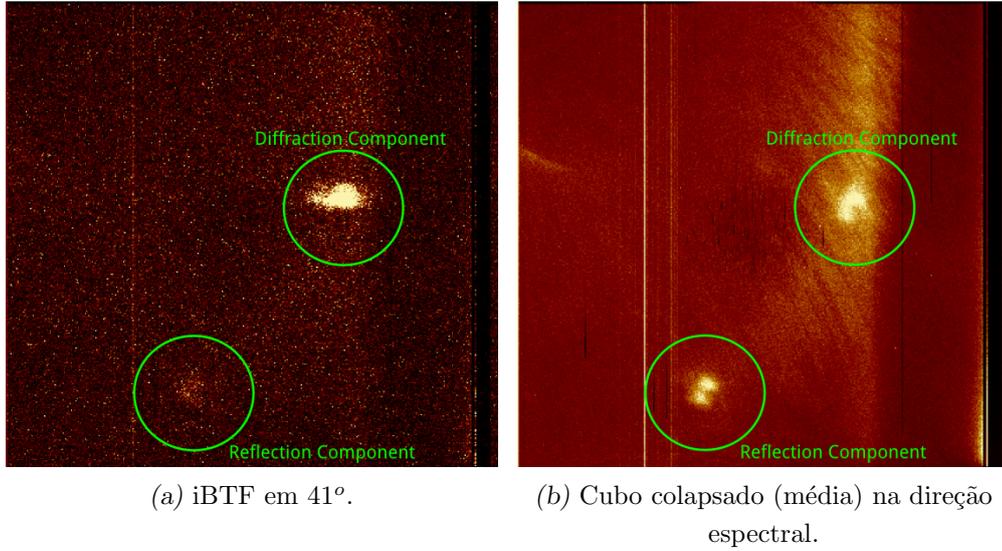


Figura 3.35: Imagens extraídas da observação da nebulosa NGC2440 em dezembro de 2012. Nelas podemos ver as componentes relacionadas com a difração da luz (*diffraction component*) e com as reflexões nas superfícies da rede (*reflection component*).

detector, a luz relacionada com a reflexão nas superfícies do substrato cairia totalmente fora do CCD.

Para determinar qual o ângulo mínimo entre a direção de modulação do filme holográfico e a superfície do substrato, utilizamos óptica geométrica simples, considerando que não existem deformações ópticas nas lentes da câmera, temos que

$$\tan \theta = \frac{h}{f}, \quad (3.4)$$

onde f é a distância focal das lentes imageadoras e h é a dimensão de maior valor do detector. No caso do BTFI, a distância focal nominal é de $f = 355\text{mm}$ e as dimensões do detector são $26,11\text{mm} \times 25,73\text{mm}$, correspondendo a um campo total de $4,2^\circ$ no detector.

Entretanto, este valor ainda não está relacionado com a rede. É uma propriedade pura do sistema de imageamento do BTFI. Para saber qual o ângulo mínimo entre a normal da superfície do substrato e a direção da modulação, devemos considerar a figura 3.36, cujo θ_d é o ângulo de emergência do feixe de luz difratado e θ_i é o ângulo de incidência.

Através desta figura podemos ver que a relação entre o ângulo de incidência dentro do substrato e o ângulo de difração dentro dele é dada por

$$\theta_d^{vidro} = \theta_i^{vidro} - 2\phi. \quad (3.5)$$

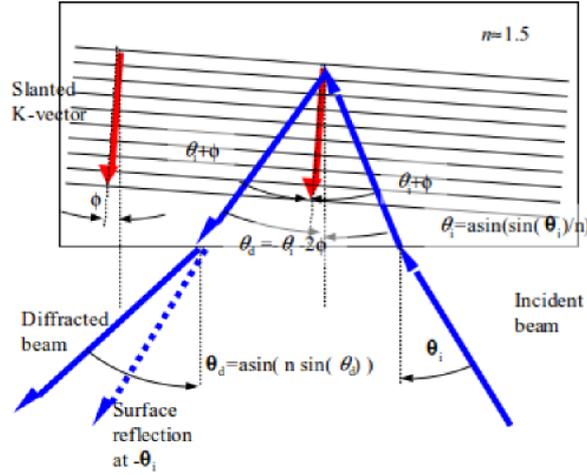


Figura 3.36: Representação geométrica dos ângulos de incidência e de difração em uma VPH de reflexão. Cortesia de Frank Havermeyer, Ondax.

Em seguida, utilizando esta equação e aplicando a lei de Snell com aproximação de primeira ordem tanto para θ_i quanto para θ_d (equações 3.6 e 3.7), temos finalmente que

$$\theta_d^{ar} = \theta_i^{ar} - 2n\phi.$$

$$\theta_i^{vidro} = \arcsin\left(\sin\frac{\theta^{ar}}{n}\right) \approx -\frac{\theta^{ar}}{n} \quad (3.6)$$

$$\theta_d^{vidro} = -\theta_d^{vidro}n \quad (3.7)$$

Ou seja, a distância angular entre o ângulo de incidência e o ângulo de difração é de $2n\phi$. No caso, como a distância angular entre o ângulo de incidência e o ângulo de difração devem ser maiores que $4,2^\circ$, temos que ϕ deve ser maior do que $4,2^\circ/2n$. Para um $n \approx 1,5$, temos que ϕ deve ser maior do que $1,4^\circ$. Segundo a Ondax, é possível fazer redes de difração com ϕ até aproximadamente 2° . Sendo assim, um novo lote de redes foi feito, as redes 22679 e 22709.

No caso, as redes foram feitas de modo que ϕ estivesse na direção perpendicular à direção da dispersão. Caso ϕ fosse somado na direção da dispersão, o único efeito que teríamos seria uma alteração na calibração em comprimento de onda e o fantasma ainda estaria presente. Sendo assim, estas duas redes apresentam componentes de difração e de reflexão do substrato bem separadas.

Entretanto, notamos que as redes Ondax degradam a PSF angular de modo que uma

observação com elas é impraticável. Para exemplificar, mostramos, na figura 3.37, uma imagem obtida com um *pinhole* de $200\mu\text{m}$ iluminado por uma fonte contínua usando dois espelhos nos lugares das redes. Agora, comparamos esta imagem com a PSF obtida com a rede 22709, conforme mostrado na figura 3.38. Esta forte degradação na largura e na forma da PSF foi observada ao usar todas as redes Ondax fabricadas até então.

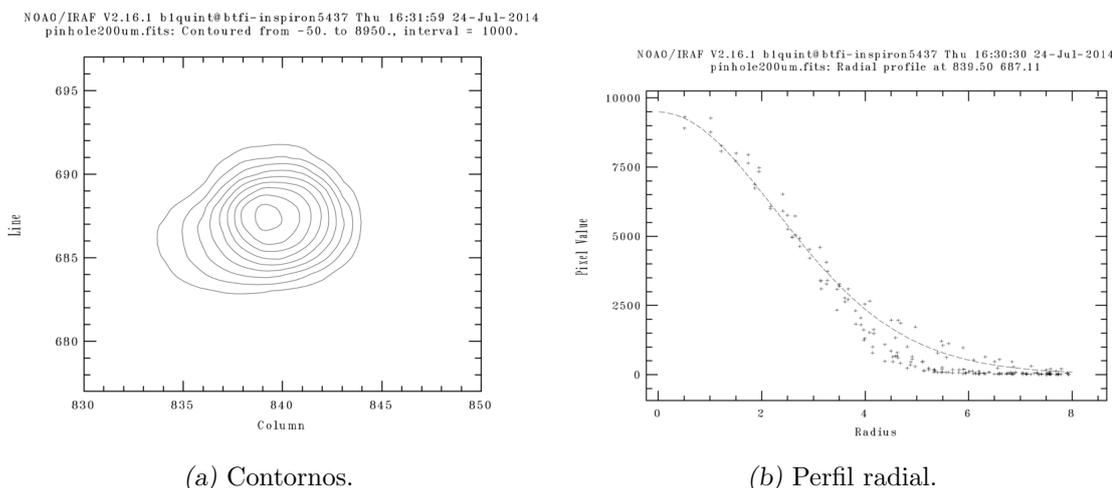


Figura 3.37: Diferentes representações da *point-spread-function* obtida usando o iBTF, dois espelhos, o *pinhole* de $200\mu\text{m}$ e uma fonte contínua. A largura à meia altura medida foi de 5,55 píxeis, com elipsidade de 0,108 e ângulo de posição de $25,9^\circ$.

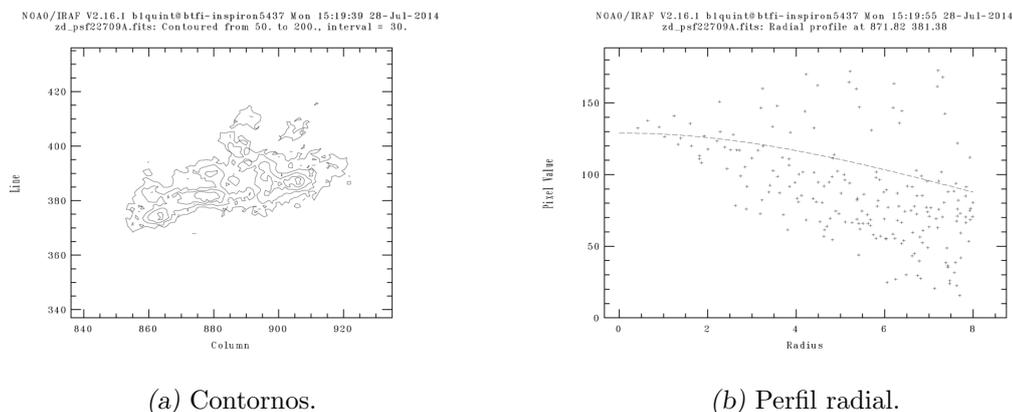


Figura 3.38: Diferentes representações da *point-spread-function* obtida usando o iBTF, um espelho conjugado com a rede Ondax 22709, o *pinhole* de $200\mu\text{m}$ e a lâmpada de Neônio. A PSF foi medida em mais de 30 píxeis.

Uma investigação mais profunda nos mostrou que o motivo pelo qual estas redes degradam tanto a imagem é que o filme holográfico não é uniforme em toda a sua extensão. Isto foi comprovado através de imagens obtidas com interferômetros no LNA. Na figura 3.39b, a imagem obtida do interferograma de uma rede de difração clássica em 0° (ordem

0 de difração) e 35° (1ª ordem de difração) e, na figura 3.40, os interferogramas obtidos com a rede 22709 em 0° (ordem 0 de difração) e 45° (1ª ordem de difração).

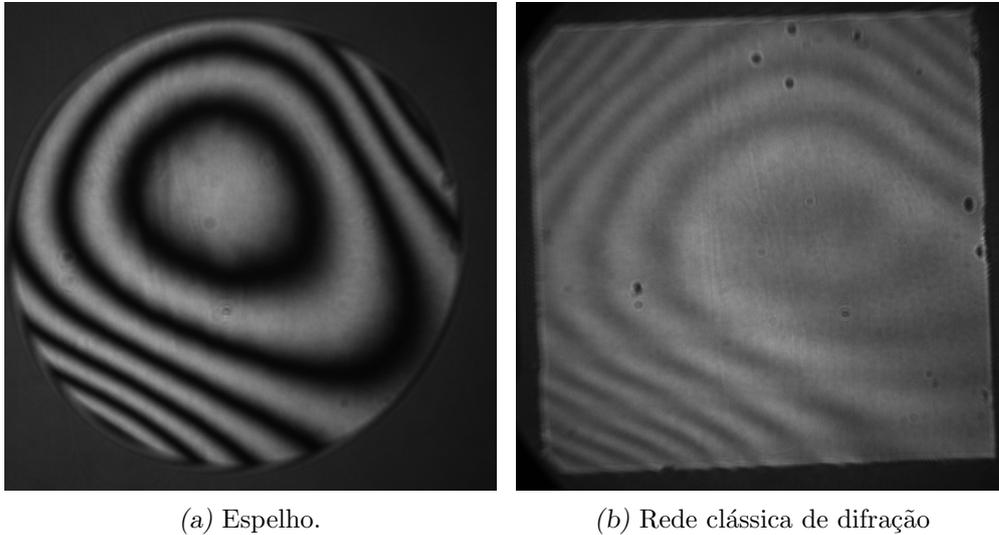


Figura 3.39: Interferograma de um espelho e de uma rede clássica de difração para servir como referência para ordem 0 e ordem 1 de difração. Elas nos mostram que uma superfície uniforme possui poucas e largas franjas.

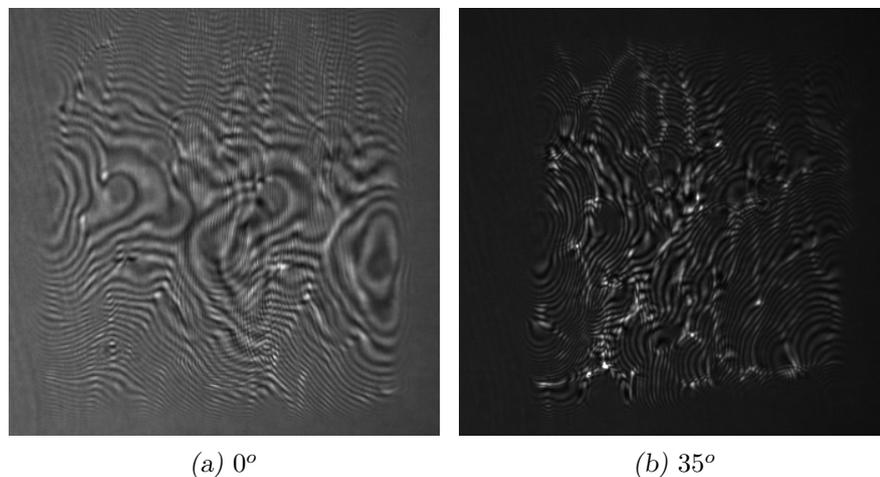


Figura 3.40: Interferograma da rede Ondax 22709 mostrando as irregularidades na reflexão simples (ordem 0 de difração) e na primeira ordem de difração. As franjas estreitas e bastante numerosas mostram irregularidades nas superfícies da rede e do substrato.

Apresentamos nossos resultados para a Ondax e pedimos que eles fizessem novas redes prestando atenção, também, na PSF angular do instrumento. Eles nos forneceram as redes com número de série 24142 e 24147 com a mesma frequência linear, mesma espessura, mesmo tamanho e com o mesmo ângulo de inclinação da modulação do índice de refração.

Os vidros para produção destas redes foram medidos para identificação de franjas não desejadas no laboratório de óptica da Universidade de São Paulo por cortesia do Prof. Zílio. Estas redes deveriam apresentar melhores resultados em termos de qualidade de imagem. Entretanto, conforme podemos ver nas imagens 3.41 e 3.42, os resultados são melhores, mas não bons o suficiente para observações astronômicas.

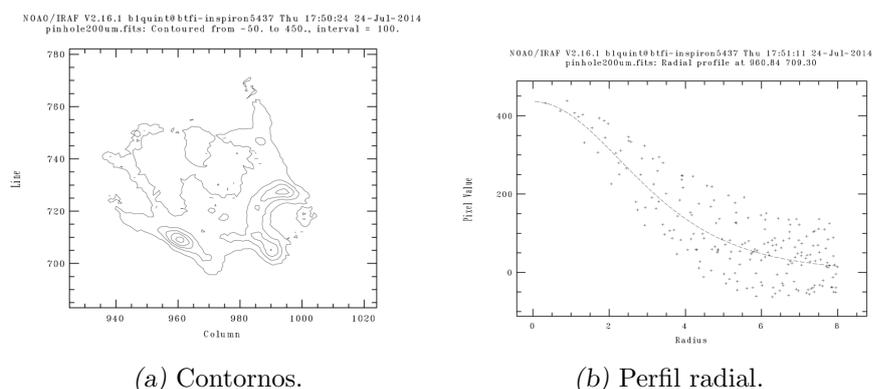


Figura 3.41: Diferentes representações da *point-spread-function* obtida usando o iBTF, um espelho conjugado com a rede Ondax 24142, o pinhole de 200 μm e a lâmpada de Neônio

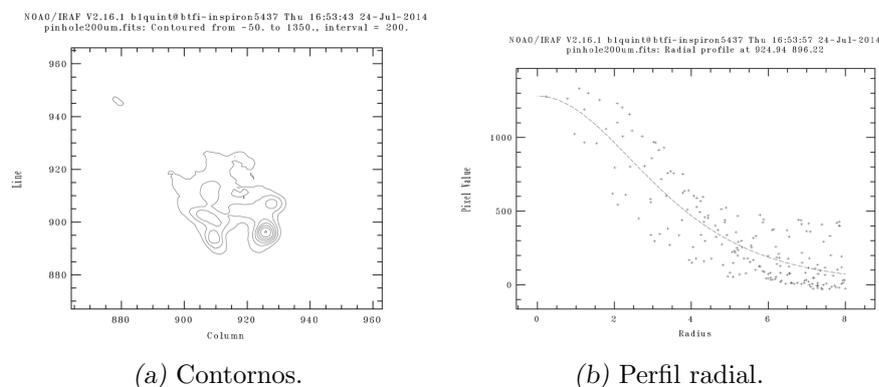


Figura 3.42: Diferentes representações da *point-spread-function* obtida usando o iBTF, um espelho conjugado com a rede Ondax 24147, o pinhole de 200 μm e a lâmpada de Neônio. O FWHM medido é de 6,17 e a elipsidade é 0,113.

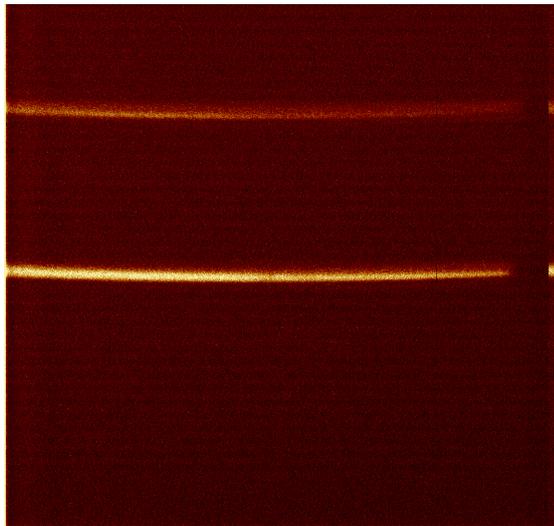
Além de todos os problemas no domínio angular, vimos que as redes de difração da empresa Ondax também não se comportaram conforme esperado no domínio espectral. Para descrever este problema, utilizamos dados das observações realizadas nas datas 6 de dezembro de 2012. Devido à sua alta resolução espectral, as redes de difração holográficas que atuam em modo de reflexão oferecem a possibilidade de operarem individualmente, configuração *single-pass*, ou em pares, configuração *double-pass*. Mas, durante parte desta noite, obtivemos dados usando o iBTF na configuração *double pass* usando as redes 16809

e 16819.

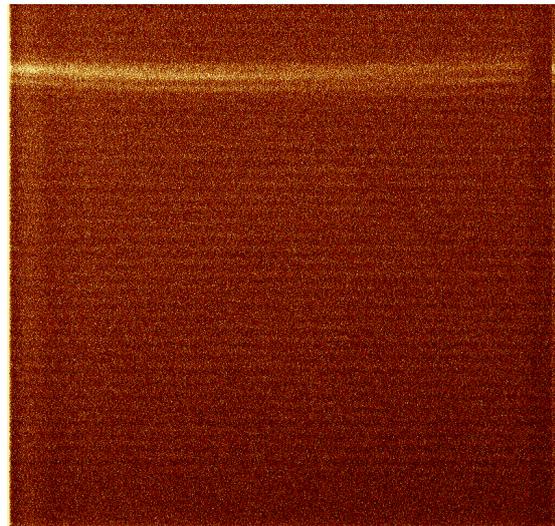
A principal vantagem de utilizar a configuração *double-pass* é um ganho na resolução espectral de um fator aproximado de $\sqrt{2}$. Além disto, os fantasmas que aparecem devido à reflexão na interface ar-substrato também ficam muito mais fracos.

Por outro lado, usar o iBTF em modo *double-pass* faz com que menos luz chegue ao detector. Como se isso não bastasse, é necessário fazer um ajuste fino para que o pico de difração da primeira rede de difração coincida perfeitamente com o pico de difração da segunda.

Devido à restrições no processo de fabricação das redes, este ajuste é válido somente em um determinado intervalo angular. A figura 3.43a mostra uma imagem obtida com o iBTF em modo *double-pass* iluminado por uma lâmpada de calibração de Ne localizada no interior do telescópio. Nela, a dispersão ocorre na direção vertical da imagem e é possível ver as linhas de difração bem definidas. Lembrando que a dispersão ocorre na direção vertical da imagem. Isto porque o ajuste das duas redes foi feito exatamente neste ângulo.



(a) iBTF
posicionado em
37,5°.



(b) iBTF
posicionado em
30,0°.

Figura 3.43: iBTF posicionado em dois ângulos diferentes para mostrar os efeitos da perda no ajuste entre as duas redes.

Já a figura 3.43b mostra uma estrutura alargada e muito mais fraca. Isto porque ela foi obtida com o iBTF posicionado em 30°, um ângulo muito distante do ângulo de ajuste. Para melhor visualizar este efeito, um gráfico é apresentado na figura 3.44 onde um corte

é feito nas figuras 3.43a e 3.43b na direção da dispersão. Neste gráfico é possível ver que, na curva correspondente à imagem obtida em 30° , há uma segunda componente no pico de difração onde deveria haver somente uma. Esta segunda componente nos mostra que os picos de difração das duas redes não estão devidamente ajustados.

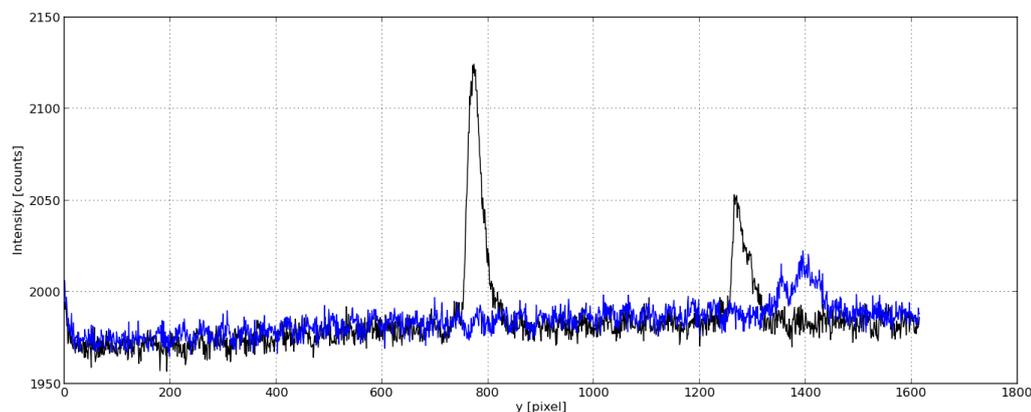


Figura 3.44: Corte vertical nas imagens 3.43a (em preto) e 3.43b (em azul).

Entramos em contato com a Ondax para relatar o problema. Segundo ela, este tipo de problema já é esperado visto que é impossível fabricar redes com frequências lineares idênticas. Mesmo um erro menos do que 1% já é capaz de provocar este efeito. Por este motivo, a partir destas observações utilizamos as redes Ondax somente em *single mode*.

3.3.4 Eficiência operacional

Um cubo de dados contendo 50 canais, com 5 imagens obtidas em cada canal (posição) e com tempo de exposição de 1,0 segundo, tem um tempo efetivo de exposição “*on-object*” de 4 minutos e 10 segundos (ou 250s). Entretanto, medimos que o tempo total gasto para obtê-lo é de 17 minutos e 10 segundos (1030s), representando uma perda de 13 minutos ou uma eficiência operacional de 24.26%. Entre as possíveis fontes de atraso no sistema do BTFI, temos o tempo de comunicação entre o software de controle do instrumento, chamado KAFKA, e o software de controle do iBTF, escrito em *Visual Basic 6*[®]. Temos também o tempo de comunicação software de controle do iBTF e o controlador PLC dos seus três motores e a velocidade destes motores.

Para investigar a fundo sobre a perda causada pelo sistema do iBTF, o aluno de Iniciação Científica, Rodrigo Massera, foi ao Chile para trabalhar sob nossa orientação du-

rante uma semana. Juntos, fizemos uma análise aprofundada no código fonte do software de controle do iBTF buscando possíveis fontes de atraso.

Conforme suspeitávamos, foi possível elevar a velocidade dos motores do iBTF de $2.06^\circ/\text{s}$ para $4.12^\circ/\text{s}$, no caso dos motores de rotação, e de 3.05 mm/s para 6.10 mm/s . Esta alteração para esta configuração, na qual tínhamos um passo bem pequeno entre uma posição e outra, acabou apresentando ganho desprezível.

Conseguimos também alterar a velocidade de comunicação entre o PLC e o software de controle através de uma variável chamada BAUDRATE. Ela foi alterada de 9600 bits/s para 115200 bits/s , apresentando um ganho na eficiência de operação do sistema de 6.74% .

Por fim, localizamos, dentro do código fonte, diversas fontes de atraso deixadas possivelmente para fins de debug do código. Eliminando estas fontes, a eficiência de operação melhorou em 1% .

Após todas estas alterações, um cubo de dados contendo 50 canais, com 5 imagens obtidas em cada canal e com tempo de exposição de 1.0 segundos, levou 11 minutos e 55 segundos para ser concluído. Comparando este valor com o tempo de exposição efetivo de 4 minutos e 10 segundos, temos que a perda no tempo de ciclo de trabalho com o iBTF diminuiu de 75.74% para 65.06% . No futuro, faremos testes para exposições mais longas que serão típicas para uso do instrumento pois uma perda tão grande com controle e comunicação é inaceitável.

Observações

O produto final de observações com o BTFI é um cubo de dados contendo informação angular nas direções X e Y e informação espectral na direção Z. O cubo pode ser visto tanto como um conjunto de imagens obtidas em diferentes comprimentos de onda quanto como um conjunto de espectros (figura 4.1). Descreveremos, neste capítulo, os procedimentos necessários para obter dados neste formato e para transformar dados brutos em cubos livres de assinatura instrumental e calibrados em comprimento de onda.

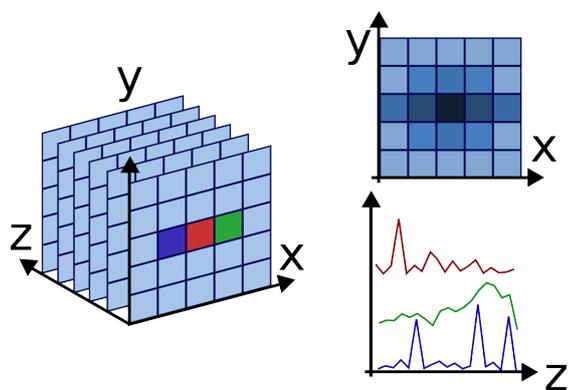


Figura 4.1: Estrutura de um cubo de dados com observações astronômicas: das três dimensões, duas delas são espaciais enquanto a terceira é a dimensão espectral

A matéria prima necessária para montar um destes cubos é um conjunto de imagens no formato FITS (Wells et al., 1981) obtido durante a observação de um determinado objeto utilizando o FP ou o iBTF em diferentes configurações. Estas imagens possuem ruído eletrônico e térmico, além da variação pixel a pixel de alta e baixa frequência. O tratamento destas imagens é feito, inicialmente, da mesma forma que faríamos com um sistema de imageamento comum, ou seja, através da remoção de BIAS, DARK e normalização por

FLAT. Chamamos esta etapa de pré-redução de dados ou tratamento das imagens.

Logo em seguida, montamos um cubo de dados considerando os parâmetros utilizados durante a observação. Estes parâmetros ficam armazenados no cabeçalho (*header*) do arquivo FITS correspondente. Para manipular estes cubos no formato FITS, utilizamos o software QFitsView (Ott, 2012), o pacote de rotinas astronômicas IRAF (Tody, 1986) e scripts escritos em Python que estão no apêndice.

Após montarmos um cubo de dados, precisamos reorganizar a informação espectral dentro dele. Este procedimento deve ser feito tanto para o iBTF quanto para o FP.

Por fim, após termos o nosso cubo de dados devidamente reordenado, entramos no processo de **calibração em comprimento de onda** onde identificamos os comprimentos de onda de cada um dos quadros do cubo de dados. Para o iBTF, este processo é trivial, visto que a relação entre o ângulo de incidência da luz e o comprimento de onda que atinge a região do espectro de referência é direta através da equação de Bragg (Kogelnik, 1969). Entretanto, no caso do Fabry-Perot, este processo pode ser mais complicado devido ao seu caráter periódico por usar altas ordens de interferência.

Durante este capítulo, descrevemos cada um dos passos detalhadamente mostrando os parâmetros usados e suas limitações.

4.1 Aquisição de dados

Como já dito algumas vezes, o BTFI é, em essência, um sistema de aquisição de imagens. Seu diferencial é utilizar filtros cujos comprimentos de onda selecionados são ajustáveis continuamente. Estas imagens, ao serem amontoadas umas sobre as outras, formam um cubo de dados. Em uma noite de observação, as imagens que compõem um cubo de dados precisam ser obtidas de forma ordenada, alternando entre a captura em si e a mudança de configurações internas do instrumento.

No iBTF, a configuração que altera o comprimento de onda que chega ao detector é o ângulo de operação, definido como o ângulo de incidência sobre as redes de difração no centro do campo, em graus. Já para o FP, a configuração ajustável é a tensão aplicada aos piezos que definem a distância entre as placas do FP, dado em bcv (*Binary Control Value*). A relação entre os ângulos de operação do iBTF ou o valor em bcv do FP e o comprimento de onda é algo que deve ser encontrado posteriormente durante o processo

chamado **calibração em comprimento de onda**.

Entretanto, se iluminarmos todo o plano focal do BTFI com uma fonte monocromática e fizermos uma exposição com o iBTF ou com o FP, a luz desta fonte não cobre todo o detector.

Um exemplo é mostrado nas figuras 4.2a, 4.2b e 4.2c, que nos mostram duas das linhas de emissão de uma lâmpada de Neônio. Esta lâmpada ilumina todo o plano focal do BTFI. Conforme variamos o ângulo de operação do iBTF, estas linhas se deslocam para cima ou para baixo. As demais linhas de emissão do Ne caem fora do detector nas configurações apresentadas mas podem reaparecer nas imagens dependendo do ângulo de operação do iBTF.

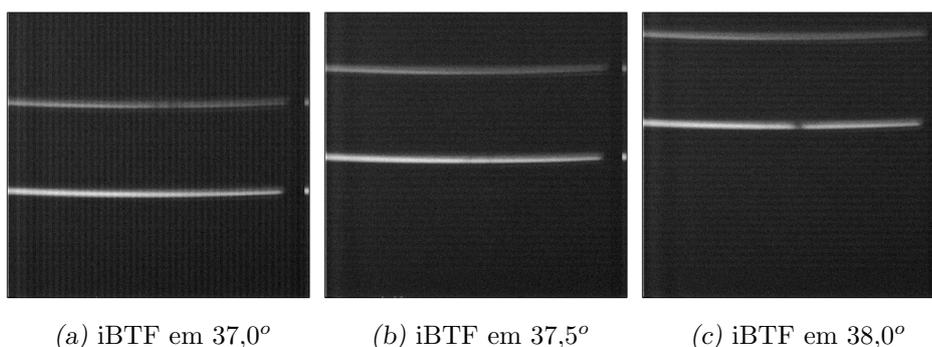
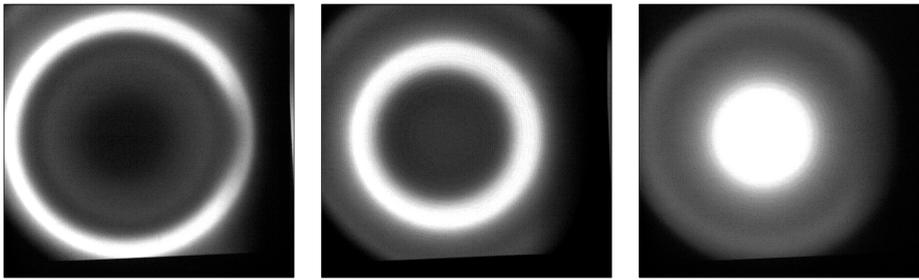


Figura 4.2: Imagens obtidas iluminando o plano focal com uma lâmpada de Neônio e usando o iBTF com a rede Ondax 16809 em diferentes ângulos de operação. Dados de 6 de dezembro de 2012. As duas linhas presentes nesta imagem correspondem às linhas em 6678 Å e 6652 Å.

Um outro exemplo é mostrado nas figuras 4.3a, 4.3b e 4.3c. Aqui, a mesma lâmpada de Neônio é utilizada para iluminar todo o plano focal do BTFI, mas as imagens são obtidas com um Fabry-Perot de alta resolução. Nelas, podemos ver anéis concêntricos cujos raios diminuem conforme aumentamos o valor de Z em bcv. O filtro interferencial 6600/20 foi utilizado para que somente uma das linhas de emissão do Ne passasse pelo FP. Caso ele não fosse usado, veríamos uma coleção de anéis com diferentes intensidades e raios superpostos, dificultando a compreensão do fenômeno e a posterior identificação das linhas.

Desta forma, podemos ver que a luz relacionada a um determinado comprimento de onda cai em diferentes regiões do detector para diferentes configurações. Sendo assim, para que possamos reconstruir uma imagem em um comprimento de onda específico, precisamos fazer uma varredura (também chamado aqui de *scan*) variando as configurações do iBTF ou do FP de modo que possamos garantir que esta linha percorra todo o detector.



(a) HR FP em $z = 0$ *bnv* (b) HR FP em $z = 100$ *bnv* (c) HR FP em $z = 200$ *bnv*

Figura 4.3: Imagens obtidas iluminando o plano focal com uma lâmpada de Neônio, usando o FP, o filtro 6600/20 e variando a distância entre as placas. Dados de 2 de abril de 2014. As duas linhas presentes nesta imagem correspondem à 6678 Å e 6652 Å.

O KAFKA, apresentado na seção 2.5 do capítulo 2, foi escrito de forma a fazer esta variação continuamente sem que o usuário precise interagir com o BTFI a cada nova configuração e exposição. Este recurso pode ser encontrado na aba *Operation Mode*, nos campos *iBTF Scan* ou *FP Scan*, dependendo do modo em que for operado.

Em ambos os casos, o usuário deve fornecer o ângulo inicial (*iBTF*) ou a posição inicial (FP) desejada, o passo a ser dado em cada interação do *scan* e o número de passos a serem dados. No caso do Fabry-Perot, o usuário ainda pode fornecer o número de *sweeps* (número de varreduras em um único cubo de dados) a serem dados e o sentido deles. Logo, no caso de um FP, podemos optar por realizar diversas varreduras em uma certa configuração ao realizarmos a observação de um objeto com apenas um clique, resultando em um único conjunto de dados que será transformado posteriormente em um cubo de dados. A direção de um *sweep* indica se, ao terminar uma varredura, o FP deve voltar à posição inicial e começar outra no mesmo sentido ou se, ao chegar no ponto final, ela deve mudar de direção mudando o sinal do incremento. Este recurso foi implementado pois, em determinados casos, as várias varreduras em diferentes direções ajudam a suavizar os efeitos da variação de brilho atmosférico em diferentes comprimentos de onda.

De qualquer forma, como a matéria prima destes cubos de dados são imagens, precisamos, antes de uma noite de observação, obter imagens de referência para a posterior extração dos ruídos de BIAS, DARK e FLAT, conforme já é feito tradicionalmente em observações astronômicas.

4.2 Imagens BIAS

A primeira fonte de ruído de um detector CCD ou EM-CCD é de natureza eletrônica. Por maior que seja a qualidade do detector, cada pixel sempre terá um valor de contagens mínimo inerente ao sistema. Este mínimo eleva o valor de todos os píxeis da imagem em um determinado valor chamado, em astronomia, de BIAS. Além disto, cada pixel se comporta de maneira diferente. Isso faz com que este mínimo varie pixel a pixel.

Na prática, este ruído é removido através da captura de imagens com tempo de exposição de zero segundos, ou seja, com o obturador fechado. Assim, os valores em cada pixel corresponderão apenas ao ruído pertencente à eletrônica inerente ao detector. Como existe uma certa flutuação estatística também inerente ao sistema, é bastante comum obter várias imagens e combiná-las utilizando a média ou a mediana de cada pixel, dependendo do caso.

No caso do BTFI, os detectores podem ser operados no modo clássico ou no modo EM (electron multiplier - ver seção 2.4). O BIAS de cada um desses modos também terá diferentes peculiaridades. A figura 4.4a mostra um exemplo de imagem de BIAS com o modo clássico e a figura 4.4b uma região aumentada mostrando a flutuação pixel a pixel devido ao ruído eletrônico.

A figura 4.4c apresenta o histograma de um dos BIAS obtidos no modo clássico. Nesta figura, podemos ver que a grande maioria dos píxeis possui uma contagem em torno de 1100 contagens. Este é o valor do BIAS para estas imagens. Entretanto, podemos perceber que existem dois outros picos neste histograma: um em torno de 4590 contagens e outro em torno de 8730 contagens.

Com uma análise mais profunda, descobrimos que estes dois picos são originados por um defeito no detector: estas altas contagens pertencem às quatro primeiras colunas do lado esquerdo das imagens. Ao remover estas colunas, os dois picos desaparecem. Esperávamos que o histograma se aproximasse a uma distribuição normal centrada no valor efetivo do BIAS mas encontramos algo próximo à superposição de duas gaussianas (figura 4.5).

Nossa principal suspeita é que este segundo pico esteja relacionado com a estrutura do ruído de fundo eletrônico, presente em todas as imagens, e com o ruído causado por píxeis quentes (com contagens mais elevadas). Este é um dos argumentos utilizados para combinarmos várias imagens de BIAS.

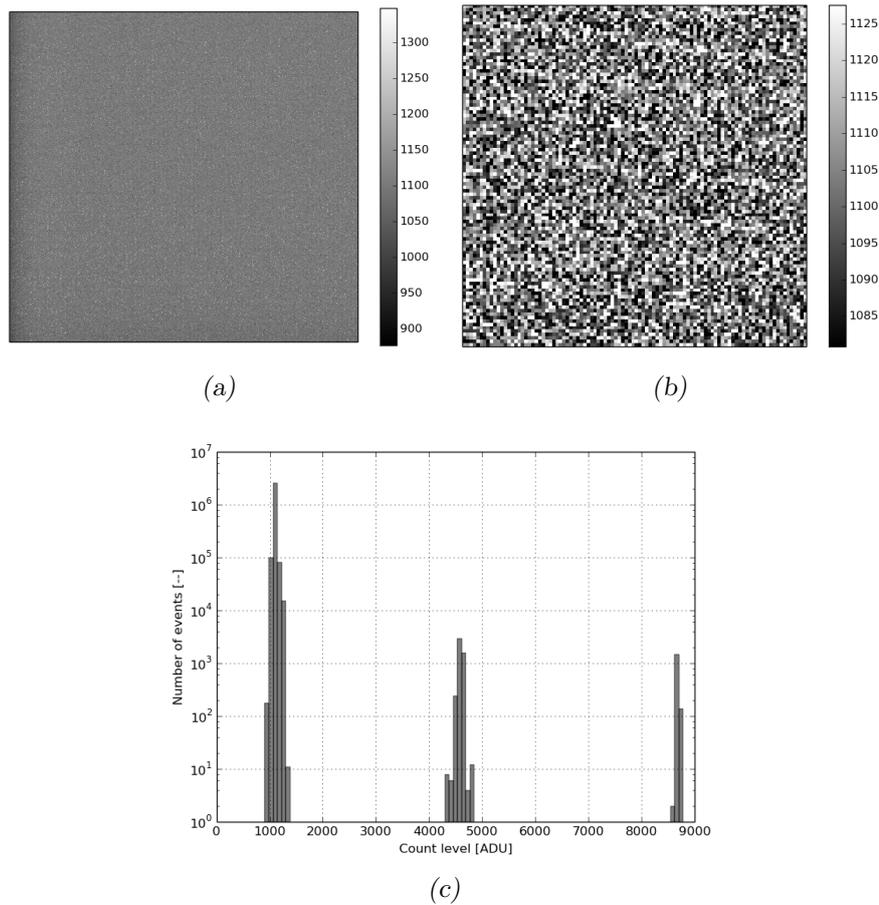


Figura 4.4: Diferentes representações de uma imagem de BIAS obtida no modo clássico. (a) Imagem completa. (b) Ampliação de 100 x 100 píxeis no centro da imagem. (c) Histograma da imagem completa.

Entretanto, durante uma noite de observação podemos ter uma variação na temperatura ambiente, o que faz com que os níveis médios do ruído de fundo variem de imagem pra imagem.

Sendo assim, as imagens de BIAS são utilizadas somente para estimar a estrutura do ruído de fundo e não o nível médio. No caso das imagens obtidas com o BTFI, tanto no modo Clássico quanto no modo EM-CCD, a determinação do nível de fundo é feita utilizando uma parte do detector que nunca é exposta à radiação, chamada de região de OVERSCAN. No caso do BTFI, essa região é definida por todos os píxeis localizados entre as colunas 1603 e 1668 do detector. Para isso, utilizamos a rotina `ccdproc` do IRAF. Essa rotina faz uma estimativa do nível de fundo na região de OVERSCAN e o subtrai de todos os píxeis da imagem.

Logo em seguida, combinamos estas imagens de BIAS usando a rotina `ccdproc`. Esta combinação das imagens pode ser feita utilizando a média ou a mediana de cada píxel. A

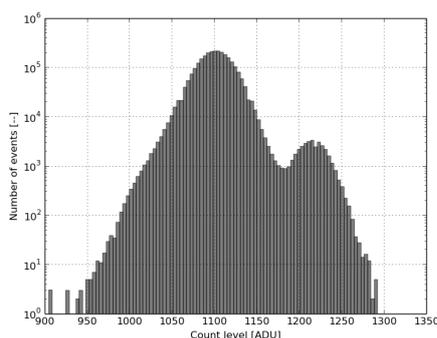


Figura 4.5: Histograma extraído de uma única imagem de BIAS no modo clássico após a remoção das quatro primeiras colunas.

imagem resultante é apresentada na figura 4.6 sob várias representações.

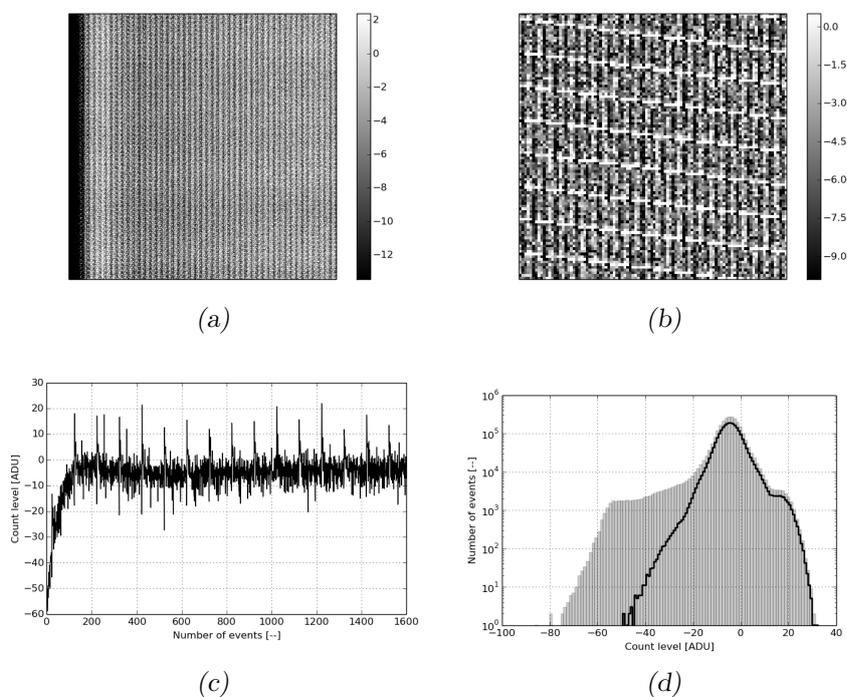


Figura 4.6: Diferentes representações de uma imagem de BIAS obtida no modo clássico após combinar 60 imagens. (a) Imagem completa. (b) Zoom de 100 x 100 no centro da imagem. (c) Gráfico da linha central. (d) Histograma da figura completa (em cinza) e da seção excluindo as primeiras 100 colunas (contorno em preto).

É interessante notar o gradiente no lado esquerdo da imagem 4.6a. Confirmamos a presença e o nível deste gradiente fazendo um gráfico das contagens dos píxeis e suas coordenadas em X da linha central da imagem, mostrado na figura 4.6c. A figura 4.6d mostra a diferença entre considerar esta região (histograma com fundo em cinza) ou não (linha contínua). A cauda à direita do histograma está relacionada ao ruído, aparentemente

de natureza eletrônica, devido à sua periodicidade, que pode ser percebido na imagem 4.6b, que mostra uma região de 100 x 100 píxeis extraída da imagem de BIAS combinada.

Repetimos exatamente o mesmo procedimento com as imagens de BIAS obtidas com o modo EM ligado. Primeiramente usamos o CCDPROC para remover o nível de fundo das imagens BIAS e recortar a seção útil do CCD. Os resultados são mostrados na figura 4.7 onde apresentamos um exemplo de uma imagem de BIAS completa, uma região de 100 x 100 píxeis extraída do centro desta imagem e um histograma desta mesma imagem já tratada.

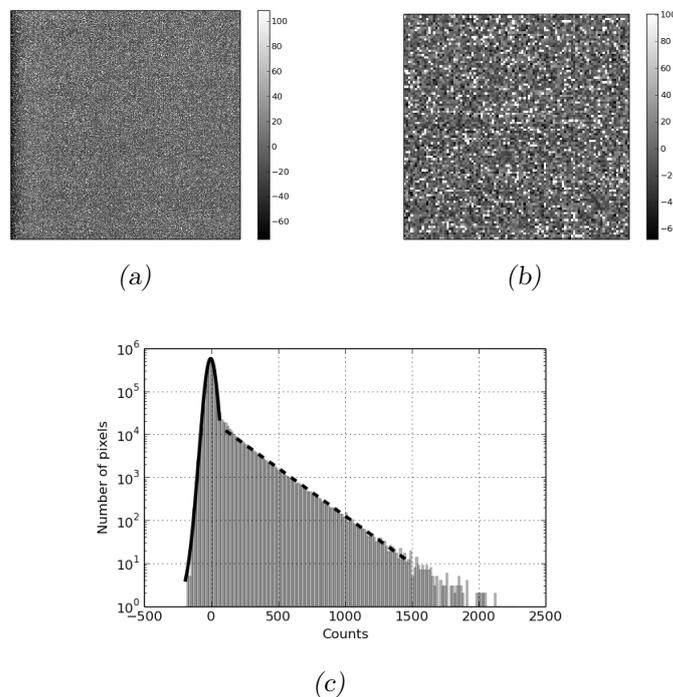


Figura 4.7: Imagem de BIAS obtida com o modo EM. Nelas, o nível de fundo medido na região OVERSCAN já foi extraído e as colunas à esquerda removidas. A linha contínua mostra o ruído aleatório em torno de zero e a linha tracejada está relacionada com o ganho EM descrito na seção 2.4 do capítulo 2. (a) Imagem de BIAS completa. (b) Região de 100 x 100 extraída do centro da imagem. (c) Histograma da imagem já tratada.

Nela, notamos que o histograma possui dois componentes. Primeiramente uma componente gaussiana representada por uma linha contínua centrada em zero contagens. Esta componente está relacionada com o ruído de leitura intrínseco ao detector. A segunda é uma componente linear que aparece em todas as imagens de BIAS obtidas no modo EM. De acordo com Tulloch e Dhillon (2010), o ângulo desta reta ajustada pode ser utilizada para calcular o ganho do modo EM, entretanto, não nos aprofundamos neste ponto.

Atualmente, sabemos que as câmeras CCDs do BTFI apresentam vários problemas no momento. Entretanto, a descrição destes problemas e o tratamento necessário para solucioná-los está fora dos objetivos desta tese.

4.3 Imagens DARK

Outra fonte de ruído presente em todo detector CCD é o ruído térmico. Esta é a principal razão pelo qual os detectores do BTFI trabalham a -100°C . Imagens DARK são obtidas para se avaliar a importância deste ruído. Uma imagem de DARK é uma imagem obtida com o tempo de exposição equivalente ao tempo de exposição utilizado em observações mas com o obturador fechado. O processo de redução das imagens de DARK é feito seguindo o procedimento tradicional:

- Remover o nível de fundo estimado através da região de OVERSCAN;
- Remover a estrutura do nível de fundo obtido ao combinar as imagens de BIAS já processadas;
- Combinar as imagens de DARK com mesmo tempo de exposição.

Normalmente, realizamos os dois primeiros passos através da rotina `ccdproc` do IRAF e o terceiro passo usando a rotina `imcombine`. Ao combinar as imagens, recomendamos utilizar os parâmetros `zero` e `scalesec` desta última rotina. No caso, ajustamos o valor de `zero` para `mode`, indicando que iremos elevar ou diminuir o valor das contagens de cada píxel pela moda estatística de uma região que deverá ser colocada como valor do parâmetro `scalesec`.

Como exemplo, apresentamos a figura 4.8 que mostra uma imagem completa de DARK obtido com 5 segundos de exposição com o modo EM após todo o processo de remoção de ruído de fundo e após combinar 20 imagens. Ela mostra também uma região ampliada com 100×100 píxeis e um histograma cujo pico nos dá uma estimativa do nível de ruído de DARK.

A tabela 4.1 mostra uma estimativa do nível de ruído de DARK obtido com cada uma das câmeras operando como um detector no modo clássico ou no modo EM. Essa estimativa foi feita através da moda estatística da imagem.

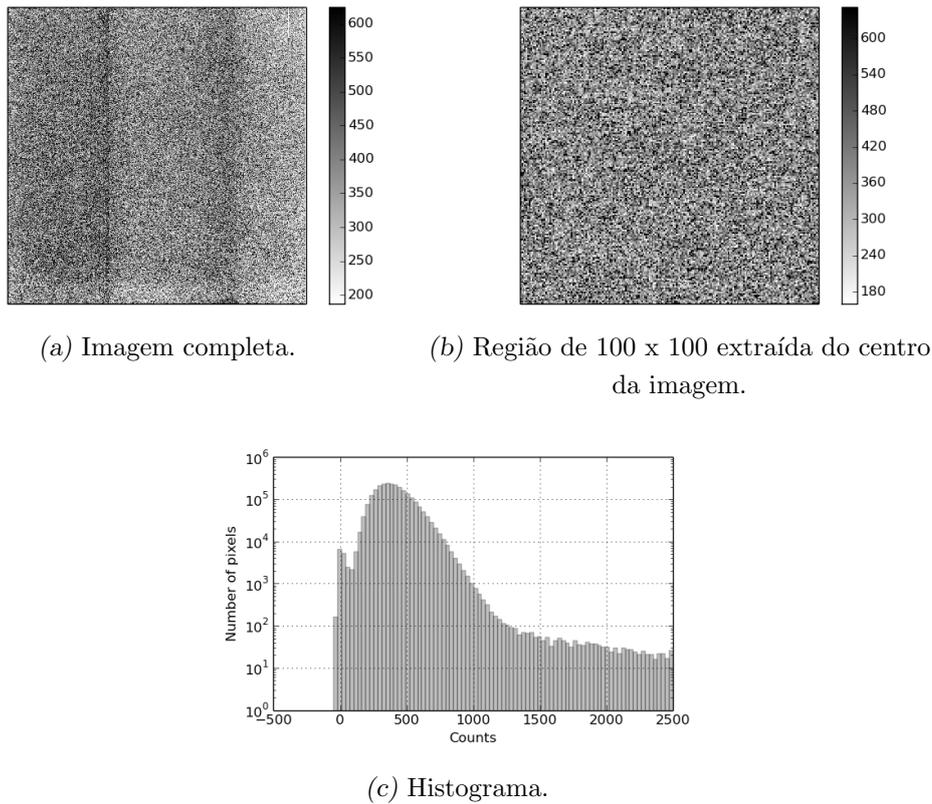


Figura 4.8: Diferentes representações de uma image de DARK produzida após processar e combinar 20 imagens de DARK com tempo de exposição de 5 s com o modo EM.

Tabela 4.1 - Níveis de DARK medidos em cada uma das câmeras.

Câmera	Modo de Operação	Tempo de Exposição	Nível de DARK estimado
Câmera 2	Modo Clássico	20,0 s	3,046 adu
Câmera 2	Modo Clássico	30,0 s	5,399 adu
Câmera 2	Modo EM	2,0 s	154,7 adu
Câmera 2	Modo EM	5,0 s	364,5 adu

Com os dados desta tabela, estimamos o nível de ruído de DARK por segundo para cada uma das câmeras em cada um dos modos disponíveis usando o ganho no modo clássico de 5,79 elétrons / adu e o ganho EM de 3087 elétrons / elétron (Andrade et al., 2012). Os resultados são apresentados na tabela 4.2.

Tabela 4.2 - Estimativas dos níveis de ruído de DARK.

	Modo Clássico	Modo EM
Câmera 2	$1,46 \text{ e}^- \cdot \text{s}^{-1} \cdot \text{px}^{-1}$	$0,13 \text{ e}^- \cdot \text{s}^{-1} \cdot \text{px}^{-1}$

Com isto, descobrimos que precisamos trabalhar para melhorar o isolamento contra luz do instrumento pois a corrente de DARK foi maior do que os valores estimados em laboratório, $0,00121 \text{ e}^- \cdot \text{s}^{-1} \cdot \text{px}^{-1}$ para o modo EM (Andrade et al., 2012).

4.4 Imagens FLAT

Imagens de FLAT são imagens obtidas iluminando todo o detector de maneira uniforme. Normalmente, elas são imagens capturadas apontando o telescópio para um anteparo iluminado dentro da cúpula do observatório (*dome flat*) ou apontando o telescópio para uma região do céu durante o entardecer ou o amanhecer (*sky flat*).

O objetivo das imagens de FLAT é compensar a variação da sensibilidade pixel a pixel do detector e, também, compensar eventuais falhas instrumentais que fazem com que uma determinada região do detector receba mais luz do que outra.

No caso de observações com um FP, devemos removê-lo do caminho óptico mas obter imagens de FLAT para cada filtro interferencial utilizado. No caso do iBTF, as redes devem ser substituídas por espelhos.

A imagem 4.9 mostra um exemplo de FLAT obtido no dia 2 de abril de 2014 com a câmera 2 com o FP fora do caminho óptico. Ela é fruto de um conjunto de imagens de *dome flats* processadas e combinadas usando as rotinas `ccdproc` e `imcombine` respectivamente. Ao combinar as imagens, utilizamos os parâmetros `scaling` e `scalesec`, atribuindo novamente o valor `mode` para o primeiro e uma seção no centro da imagem onde a moda deve ser calculada no segundo. A ideia de usar este parâmetro é normalizar as imagens de modo que os pixels estejam normalizados pela moda desta região definida pelo parâmetro `scalesec`.

Esta imagem nos mostra três sombras que limitam o campo. A parte curva no canto inferior direito da imagem, indicado por uma flecha com fundo branco, é uma sombra causada pela *dummy plate*, que substitui o FP. A sombra à direita da imagem, indicada por uma flecha curva, está relacionada com a roda de filtros secundária, que não está bem centrada ao campo. Já a sombra na parte inferior, indicada por uma flecha preenchida, está relacionada com a roda de filtros primária, que também não está bem centrada. Para termos uma imagem livre destas três sombras, precisamos realizar modificações optomecânicas como recentragem das rodas de filtros e a possível obtenção de uma *dummy*

plate com maior abertura.

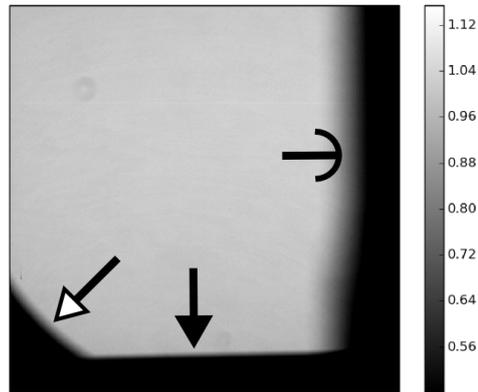


Figura 4.9: Imagem de FLAT obtida com a câmera 2. As flechas indicam a presença de *vignetting* na imagem, restringindo o campo de observação.

Com base nesta imagem, criamos uma máscara conforme os valores de cada píxel. Esta máscara é uma imagem com as mesmas dimensões que as imagens que estão sendo reduzidas e cujos píxeis possuem valores 0 ou 1. Utilizando a imagem de FLAT em questão, definimos píxeis com valores acima de 0,85 passam a ter valor 1 e píxeis com valores abaixo deste número passam a ter valor 0. Criamos esta máscara utilizando a rotina `imexpr` do IRAF. Esta máscara é utilizada posteriormente para definir a região útil do detector e é empregada através de multiplicação simples.

Usando a rotina `imexpr`, também selecionamos os píxeis da imagem de FLAT produzida com valores entre 0,85 e 1,15. Píxeis cujos valores estejam fora deste intervalo são substituídos pelo valor 1. Isto porque a imagem de FLAT é utilizada como divisor no tratamento das imagens obtidas em observação para a nivelção píxel a píxel. Caso os valores destes píxeis sejam muito diferentes de 1, a imagem de FLAT pode acrescentar ruído às imagens de observação.

4.5 Cubos de calibração

Os cubos de calibração são dados que possuem dois propósitos principais: o primeiro deles é servir de referência para a extração do mapa de fase, conforme será explicado mais adiante na seção 4.6. O segundo objetivo dos cubos de calibração é ter uma referência espectral para a calibração em comprimento de onda dos dados observacionais, conforme descrito na seção 4.7 deste capítulo.

Um cubo de calibração normalmente precisa ter uma razão Sinal-Ruído alta o suficiente para a identificação de duas ou mais linhas. Quanto mais linhas, melhor. Como estes dados são obtidos utilizando lâmpadas de calibração, podemos obtê-los idealmente antes da noite de observação, durante o dia. Assim, já teremos uma ideia de quais serão as configurações necessárias para as observações durante a noite.

Caso o usuário esteja usando pela primeira vez o instrumento, recomendamos que ele obtenha um cubo de calibração fazendo uma varredura explorando os extremos das configurações disponíveis. No caso do iBTF, para evitar que houvesse qualquer colisão entre os componentes ópticos internos devido a algum erro de configuração, o instrumento foi limitado a operar entre 25 e 45 graus, tanto no modo de transmissão quanto no modo de reflexão.

Podemos usar como exemplo as figuras 4.2a, 4.2b e 4.2c. Nelas, o valor médio da largura à meia altura da linha mais intensa no centro da imagem é de 23,05 px. A posição desta mesma linha nas três imagens é 565,31, 753,542 e 951,526 px. Sabemos, com estes números que esta linha anda 386,216 px por grau. Com isto, a largura a meia altura destas linhas em graus é de $0,059^\circ$. De acordo com o critério de Nyquist, a amostragem mínima deveria ser de $0,03^\circ$ mas, como já foi discutido anteriormente, a resolução angular dos motores do iBTF nos permite somente observações com passos de $0,05^\circ$. Assim, estes é o incremento mínimo em uma observação usando estas redes e, numa observação, teríamos que escolher qual o ângulo inicial e o ângulo final de operação. Com a amostragem limitada pelo sistema de controle, o número de passos já estaria definido. Normalmente, a definição dos ângulos está relacionada com a calibração em comprimento de onda, então deixaremos isto para a parte em que trataremos desta etapa.

Já no caso do Fabry-Perot, a limitação é dada pelo número de bits que carrega a informação do espaçamento em que as placas do étalon devem estar separadas (z). No caso, este valor vai de 0 à 4095. Entretanto, é suficiente que um cubo de calibração cubra um FSR. Se tivermos um cubo superamostrado, o FSR é calculado durante o processo de extração de fase com a rotina escrita em Python chamada `phmxttractor`, conforme será explicado com detalhes mais adiante na seção 4.6. A largura à meia altura na direção espectral, a Finesse e o passo também são fornecidos por esta rotina. Entretanto, estes valores também podem ser obtidos manualmente através da análise de um cubo de dados obtido com o FP.

Tomando como exemplo o cubo usado para criar as figuras 4.3a, 4.3b e 4.3c, podemos prosseguir da mesma forma para determinar qual o passo em bcv entre um canal e outro. O centro dos anéis está aproximadamente em $(x,y) = (712 \text{ px}, 836 \text{ px})$. Para $z = 100 \text{ bcv}$, o raio do anel é aproximadamente 483 px e a largura à meia altura é de aproximadamente 132 px. Para $z = 150 \text{ bcv}$, o raio do anel é aproximadamente 354 px e a largura à meia altura é de aproximadamente 161 px. Isso confirma que, conforme previsto pela teoria, anéis com maior raio são mais estreitos. Numa aproximação grosseira, poderíamos dizer que 1 bcv corresponde a uma variação de 2,58 px e a largura a meia altura em bcv é de 51,16 bcv. O passo mínimo para uma amostragem adequada deveria ser de 25,58 bcv, de acordo com o critério de Nyquist.

Fazendo um gráfico do espectro do píxel no centro dos anéis, podemos medir o FSR em bcvs e determinar o número mínimo de passos que este cubo deveria ter. No caso, este espectro é apresentado na figura 4.10. Nela, temos que os dois picos ocorrem em $Z = 198,71 \text{ bcv}$ e $Z = 909,19 \text{ bcv}$, um FSR de 710 bcv. O número mínimo de passos a ser dado, considerando um passo de 25 bcv por canal é 28.

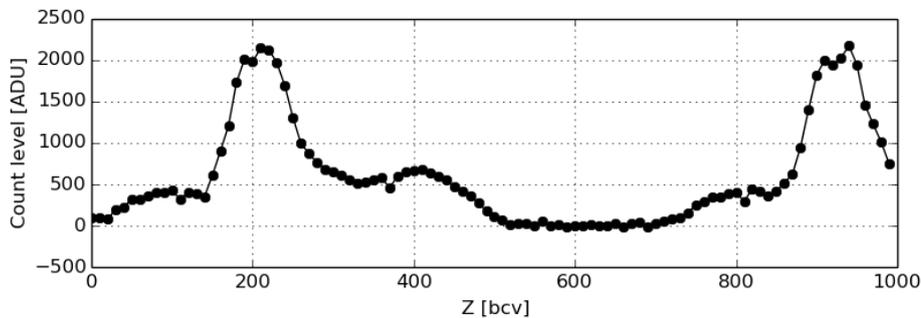


Figura 4.10: Espectro extraído do centro dos anéis de um cubo obtido com o HR FP, lâmpada de Neônio e filtro 6600/20.

Levando em consideração as lâmpadas de calibração disponíveis dentro do ISB (*Instrument Support Box*) e os filtros interferenciais dentro do BTFI, diferentes cubos de dados devem ser obtidos.

No caso do HR FP, recomendamos usar a lâmpada de Neônio juntamente com os filtros 6600/20 e 6758/38,6.

No caso do LR FP, recomendamos o uso da lâmpada de Neônio juntamente com o filtro 6600/75, que está disponível no CTIO. Por ter uma resolução espectral mais baixa, a FSR deste FP é maior e, por isso, três linhas podem ser identificadas usando um único filtro.

Também recomendamos o uso da lâmpada de Neônio para o iBTF operando com as redes Ondax, visto que todas elas foram desenhadas para trabalhar entre 6500 Å e 6900 Å. No caso das redes Wasatch, recomendamos o uso da lâmpada de Hg(Ar) (mercúrio e argônio).

Devido à possíveis instabilidades no paralelismo e na calibração do FP, recomendamos que um cubo de calibração deve ser obtido antes de cada observação com o FP. Este cubo deve ser grande o suficiente, em termos de número de canais, para que se possa construir um novo mapa de fase para cada observação. Em outras palavras, basta que um anel percorra o detector.

No caso do iBTF, nossa experiência nos mostra que tomar um cubo de calibração a cada troca de redes de difração é o suficiente.

Para a montagem dos cubos, utiliza-se um script chamado **mkcube3** cujo código é apresentado no apêndice B. Este script é um programa escrito em Python que roda por linha de comando. Ele lê o cabeçalho (*header*) de todas as imagens fornecidas a ele para determinar se elas foram obtidas em uma varredura com o iBTF ou o Fabry-Perot. Em seguida, ele cria uma lista agrupando todas as imagens que foram obtidas com o mesmo ângulo, no caso do iBTF, ou com o mesmo z , no caso do FP. Somente então, ele combina as imagens com mesma configuração e as coloca dentro de um cubo de dados.

Como dito anteriormente, tendo os cubos montados, podemos usá-los para o processo de correção de fase, que será explicado logo a seguir, e de calibração em comprimento de onda.

4.6 Correção de fase

Quando um único comprimento de onda é observado através de uma varredura usando tanto o iBTF quanto o Fabry-Perot, o que obtemos é um cubo de dados com uma linha que percorre o campo de diferentes formas. No caso do iBTF, temos uma linha aproximadamente horizontal que se desloca para cima e para baixo conforme os ângulos de operação do iBTF. No caso do Fabry-Perot, temos anéis concêntricos que aumentam e diminuem de raio conforme as placas do étalon se afastam ou se aproximam.

No caso, se considerarmos a posição relacionada a esta luz monocromática dentro de um cubo de dados ainda não corrigido, temos que ela descreve aproximadamente um plano

ao longo do cubo (figura 4.11).

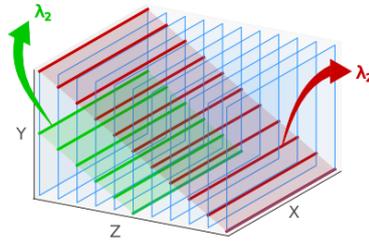


Figura 4.11: Representação artística de como a informação relacionada a um ou mais comprimentos de onda se distribui em um cubo de dados obtido com um iBTF – adaptado de Quint (2010).

Já no caso do FP, a informação relacionada a um determinado comprimento de onda descreve paraboloides concêntricos e periódicos ao longo do cubo (figura 4.12).

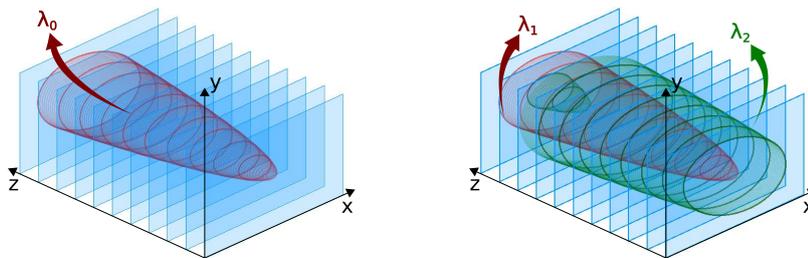


Figura 4.12: Representação artística de como a informação relacionada a um ou mais comprimentos de onda se distribui em um cubo de dados obtido com um Fabry-Perot. À esquerda, temos somente um comprimento de onda λ_1 . À direita, temos dois comprimentos de onda. O caráter periódico também é evidenciado nesta última figura – adaptado de Quint (2010).

Para que possamos ter um cubo de dados cuja informação em um determinado comprimento de onda esteja em um único quadro do cubo, precisamos realizar a chamada Correção de Fase.

Correção de fase é o nome do procedimento em que os espectros dentro de um cubo de dados são reorganizados de modo que cada quadro (*frame*) representa uma imagem obtida em um determinado comprimento de onda. Este procedimento precisa ser realizado tanto para dados obtidos com o Fabry-Perot quanto para dados obtidos com o iBTF. Ao final da correção de fase, um cubo terá então a informação angular em X e Y e a informação espectral em Z. De outro ponto de vista, cada canal Z, terá uma imagem correspondente a um único comprimento de onda.

Este procedimento é dividido em três grandes etapas: extração, ajuste e correção do mapa de fase. Cada etapa será explicada com mais detalhes nas seções a seguir.

4.6.1 Correção de fase de cubos do iBTF

Conforme já explicado anteriormente, mapa de fase é uma imagem 2D que contém o valor da posição dos espectros de cada píxel do cubo de dados em relação a um determinado píxel de referência. Para este processo, escrevemos, como parte do trabalho desta tese, um *script* em Python que faz a extração do mapa de fase chamado **phmxttractor** (apêndice C). Este script foi baseado em um software chamado BatFly, escrito pelo Prof. Dr. Fabrício Ferrari (2012, comunicação privada). Este e todos os outros pacotes escritos para esta tese precisam de Python 2.7, Matplotlib 1.3.1, NumPy 1.9.1, SciPy 0.13.3 e AstroPy 0.4.4.

O modo com que o `phmxttractor` calcula o mapa de fase de um cubo obtido com o iBTF difere um pouco do modo com que ele o faz para um cubo obtido com o FP. Como todos os passos que envolvem a correção de fase com o iBTF são mais simples, iniciaremos explicando como ela é feita usando-o como exemplo.

Em Linux, este script deve ser copiado para a pasta `$HOME/bin` do usuário. Em seguida, devemos dar permissão para executá-lo através do comando:

```
$ chmod a+x $HOME/bin/phmxttractor
```

Para executá-lo, o usuário precisa somente abrir um terminal e digitar o seguinte comando:

```
$ phmxttractor [cubo-de-calibracao.fits]
```

E responder às perguntas do *prompt*.

A primeira coisa que o `phmxttractor` faz é procurar por um píxel de referência no cabeçalho do cubo. Este píxel de referência é utilizado para calcular o quanto o espectro de um determinado píxel do cubo de dados precisa ser deslocado para que ele fique alinhado com o espectro do píxel de referência. Este parâmetro foi deixado em aberto pois nem sempre o objeto de estudo está no centro do campo e, como veremos posteriormente, após a correção de fase, parte do cubo de dados obtido com o iBTF é descartada.

Caso o píxel de referência não seja encontrado no cabeçalho, o programa perguntará se ele deve usar o píxel no centro do campo como referência. Caso negativo, ele perguntará pelas coordenadas do píxel de referência ao usuário.

Logo em seguida, ele determina o canal com máxima intensidade e armazena os resultados em uma imagem com mesma largura e altura que o cubo de dados. Em seguida, ele subtrai o valor de todos os pixels do valor armazenado no pixel de referência e, por fim, multiplica o resultado pela amostragem utilizada.

O principal problema deste método surge quando há mais de um máximo com intensidades parecidas. Isso faz com que o ruído seja o suficiente para confundir o método, conforme apresenta a figura 4.13. Nela, o programa consegue identificar os máximos dos espectros azul e verde sem problemas. Entretanto, no caso do espectro vermelho, o programa acaba encontrando um outro máximo de difração e insere um valor possivelmente errado no mapa de fase.

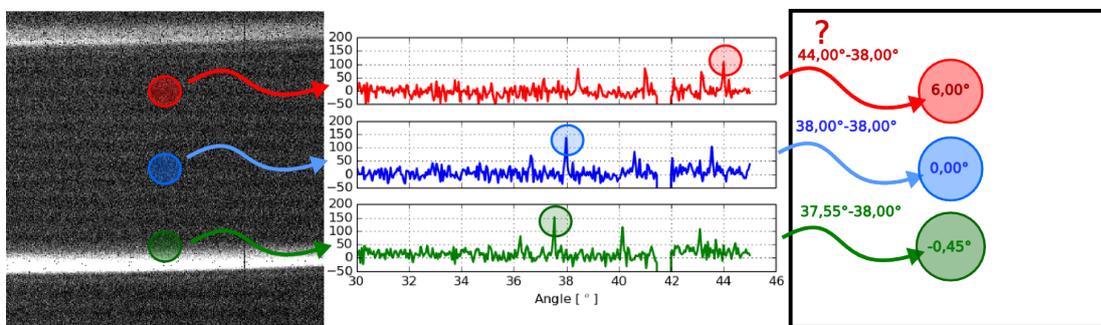


Figura 4.13: Exemplo didático de extração do mapa de fase com o iBTF. À esquerda, temos uma imagem do mesmo cubo usado com o iBTF anteriormente, obtido em $37,5^\circ$, iluminado com uma lâmpada de Ne e usando as redes 16809 e 16819. No centro, temos o espectro destas três regiões. Temos ali também destacados os picos de cada espectro. Estes valores são armazenados no mapa de fase fictício, à direita.

Para minimizar os casos em que este tipo de engano acontece, calculamos o que chamamos de **cubo de correlação**. Neste cubo, antes de pegarmos o máximo do espectro em cada pixel, calculamos a correlação de cada pixel com o espectro do pixel de referência. Antes de correlacionar o espectro de um determinado pixel com o espectro de referência, nós o normalizamos pelo seu valor máximo e o subtraímos do seu valor médio para evitar que a função de correlação tenha um formato triangular. A ideia do cubo de correlação foi usada originalmente no programa BatFly, mencionado anteriormente.

A figura 4.14 mostra o resultado desta operação sobre o cubo em que estamos usando como exemplo. Nela, podemos ver um dos quadros extraídos do cubo de correlação onde facilmente vemos somente uma linha intensa ao invés de várias linhas mais fracas.

O resultado da extração do mapa de fases do iBTF é apresentado na figura 4.15a. Este mapa de fase contém pixels isolados cujos valores não têm nenhuma relação com o valor

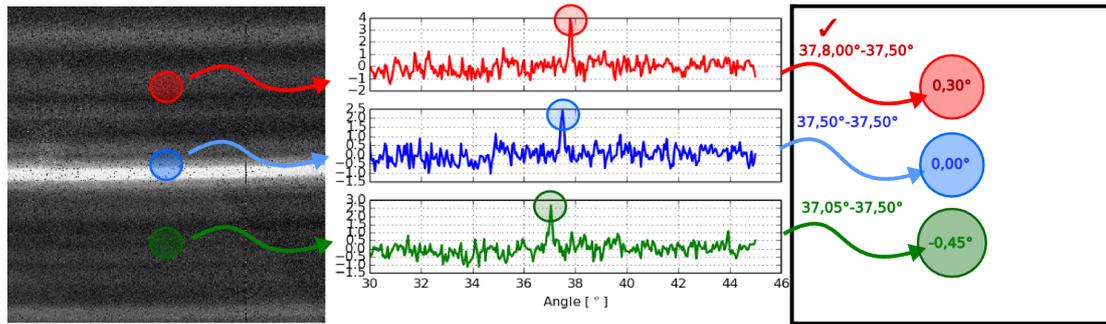


Figura 4.14: Exemplo de extração de mapa de fase com o iBTF usando cubo de correlação. À esquerda temos o canal em $37,50^\circ$ do cubo de correlação construído a partir do píxel central do cubo. Ao centro, temos as funções de correlação do píxel central com cada um dos três píxeis considerados. À direita, temos um mapa de fase fictício que armazena o quanto que teremos que deslocar o espectro de cada um destes píxeis para que eles fiquem alinhados com o píxel de referência.

dos seus vizinhos. Isto é causado pois, em alguns píxeis, o programa falha em encontrar a posição relativa entre os espectros dentro do cubo de dados e isto acontece, principalmente, em regiões onde a razão sinal-ruído é baixa. Para minimizar este efeito, podemos aplicar um filtro angular usando uma mediana 3×3 , escolhido arbitrariamente. O resultado é apresentado na figura 4.15b. Este filtro foi aplicado “manualmente” utilizando a biblioteca `ndimage` do SciPy.

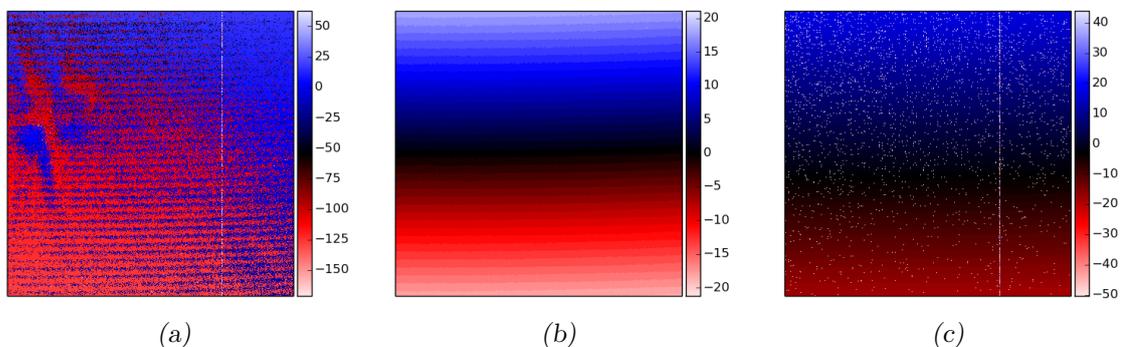


Figura 4.15: Mapas de fase extraídos do cubo obtido com o iBTF. As cores destes mapas indica o número de canais em que precisamos deslocar o espectro ao invés do ângulo, como vínhamos fazendo anteriormente. Nas regiões em azul, deslocamos o espectro em uma direção; em vermelho, na direção oposta. (a) Mapa de fase simples. (b) Mapa de fase obtido com cubo de correlação após mediana espacial 3×3 . (c) Mapa de fase obtido com cubo de correlação.

Mesmo aplicando este filtro, uma análise mais detalhada nos mostra que a variação do mapa de fase ao longo do campo é discreta. Isto é causado pois um cubo de dados é construído através de uma varredura com passos discretos e é um efeito esperado. Para obter um mapa de fases que varie continuamente, usamos um outro script, escrito também como

parte do trabalho que aqui apresentamos, chamado `phmfit`. O código fonte deste script é apresentado no apêndice D. O processo de instalação dele é idêntico ao do `phmxttractor`. Para executá-lo, o usuário precisa somente digitar:

```
$ phmfit mapa-de-fase.fits
```

E responder às perguntas do *prompt*.

Este script, no caso do iBTF, ajusta um plano inclinado ao mapa de fase. Com este mapa ajustado, evitamos possíveis descontinuidades que podem resultar numa correção de fase indevida. Para este ajuste, ao invés de utilizar todos os píxeis do mapa de fase, o script faz uma subamostragem. O número de pontos a ser usado pode ser fornecido pelo usuário através do argumento `-n` ou `--npoints`. O mapa de fase ajustado e os resíduos (mapa ajustado subtraído do mapa observado), são apresentados na figura 4.16. Os resíduos variam entre -0.5 e $+0.5$ pois o mapa de fase observado varia com passos da mesma amplitude dos passos dados entre um canal e outro na aquisição de dados enquanto o mapa ajustado varia continuamente.

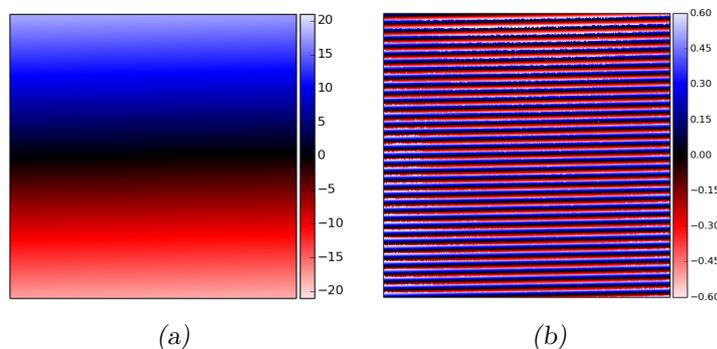


Figura 4.16: Exemplo de ajuste do mapa de fases com o iBTF. À esquerda temos o mapa ajustado e à direita os resíduos da subtração entre o mapa ajustado e o mapa observado. A unidade das escalas de cores é número de canais. (a) Mapa de fase ajustado. (b) Resíduos obtidos subtraindo o mapa de fase ajustado do mapa de fase observado após aplicar o filtro mediana.

Por fim, aplicamos o mapa de fase usando o último script chamado `phmapply`, instalado da mesma forma que os anteriores e executado com a seguinte linha de comando:

```
$ phmapply data-cube.fits phase-map.fits
```

Esta é a etapa que mais consome tempo computacional. Dependendo do tamanho do cubo de dados, ela pode levar até algumas horas. O tempo de processamento também

depende do número de pontos usado ao superamostrar o espectro de cada píxel antes de deslocá-lo. Este parâmetro pode ser ajustado através da opção `-n` do script.

Após a correção de fase, podemos medir o resultado em todo o campo calculando um novo mapa de fase do cubo corrigido. Neste novo mapa de fase, os valores absolutos de cada píxel não devem ser maiores do que o passo dado na aquisição dos cubos. No caso de um cubo sem calibração no eixo Z, estes valores não devem ser maiores do que 1 canal. O resultado é apresentado na figura 4.17.

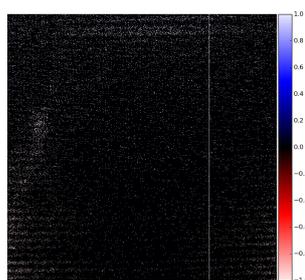


Figura 4.17: Exemplo do mapa de desvio dos espectros após a correção de fase - mapa de cores em número de canais.

A maior limitação envolvida no processo de correção de fase de cubos do iBTF é que parte da informação precisa ser descartada. Isto porque o processo “inclina” o cubo na direção de Z. Como, em computação, só podemos lidar com matrizes tridimensionais ortogonais, completamos as regiões que ficaram vazias ao “inclinarmos” o cubo com zeros. Com isto, após a correção de fase, precisamos realizar um corte na direção espectral do cubo para eliminar estas regiões que foram preenchidas com zeros. Neste corte, parte da informação vai embora também. A figura 4.18 nos mostra que o erro na posição do espectro na maioria dos pixels é nulo, conforme esperado. Nela, os pixels que apresentam altos valores estão relacionados a pixels ruidosos ou defeituosos que este cubo de dados apresenta.

Mostramos, agora, como os mesmos scripts se comportam no caso de dados obtidos com FPs e quais os algoritmos utilizados no processo de correção de fase de dados desta natureza.

4.6.2 Correção de fase de cubos de FPs

O processo de correção de fase de cubos obtidos com um FP segue os três mesmos passos: a extração do mapa de fase, usando o script `phmxttractor`, o ajuste do mapa de

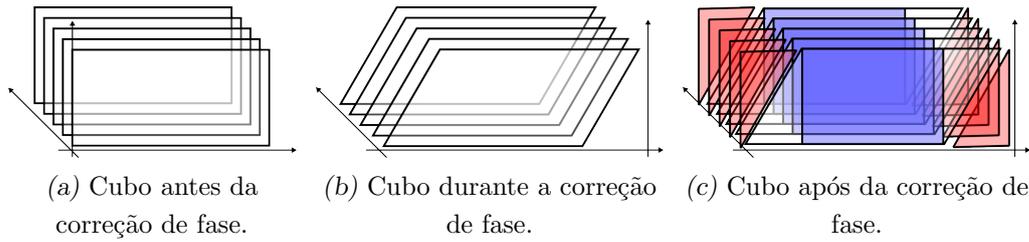


Figura 4.18: Efeitos causados pela correção de fase em um cubo do iBTF. A primeira figura mostra a geometria original, com Z apontando na horizontal e a direção Y na vertical. A segunda figura mostra o que acontece com a geometria do cubo durante o processo de correção de fase. A terceira e última figura mostra, em vermelho, as regiões preenchidas com zeros para que os dados possam ser manipulados no computador e, em azul, a região útil do cubo que contém informação em todos os píxeis. O restante é descartado.

fase extraído, usando o script `phmfit`, e a aplicação do mapa de fase ajustado, usando o script `phmapply`. Para a extração, precisamos de um cubo de calibração com uma ou mais linhas de emissão iluminando todo o plano focal. Para os exemplos apresentados aqui, usamos os dados obtidos em 2 de abril de 2014 com uma lâmpada de Ne, o FP de alta resolução e o filtro 6600/20.

De modo semelhante ao processo de extração de mapa de fase com o iBTF, caso tenhamos mais de uma linha de emissão no nosso cubo e tenhamos a suspeita de que estas linhas possam confundir o script, podemos usar a opção `-c` para usar um cubo de correlação.

Ao rodar, a primeira informação que o `phmextractor` busca é o FSR. Este parâmetro é utilizado mais adiante para encontrar o centro dos anéis e para o ajuste do mapa de fase. O procedimento padrão consiste em subtrair todos os canais $\Phi(x, y, z)$ pelo primeiro canal $\Phi(x, y, 1)$, onde deve haver pelo menos um anel, tomar o valor absoluto do cubo resultante e colapsá-lo (somá-lo) nas direções espaciais (equação 4.1).

$$FSR = \left(\sum_{x,y} |\Phi(x, y, z) - \Phi(x, y, 1)| \right)^{z \min} \quad (4.1)$$

No nosso exemplo, o resultado é apresentado pelos círculos na figura 4.3. O `phmextractor` mostra os gráficos que apresentamos aqui sempre que a opção `-s` ou `--show` for dada na linha de comando.

Em seguida, este resultado é ajustado por uma função *spline* de terceira ordem (linha cheia), desconsiderando os quatro primeiros canais. O FSR (linha tracejada) é, então, o ponto em que esta função ajustada tem seu mínimo. O ajuste é necessário pois um

FSR pode estar entre dois canais e os quatro primeiros quadros são ignorados para que o programa não se confunda ao buscar pelo valor mínimo.

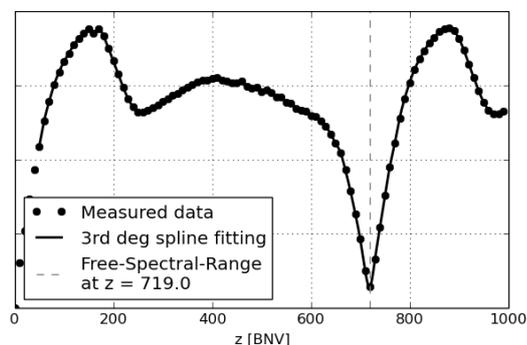


Figura 4.19: Encontrando o FSR para dados do FP. Os pontos representam a diferença de cada quadro em relação ao primeiro, a linha cheia representa um ajuste dos pontos encontrados por uma função *spline* de terceira ordem e a linha tracejada mostra onde foi encontrado o FSR.

Caso o programa falhe em encontrar o FSR, seja porque o cubo possui menos do que um FSR ou seja por qualquer outra razão, ele pede ao usuário qual deve ser o FSR em *bev* e em números de canais.

O segundo passo é encontrar o píxel de referência caso não seja fornecido pelo usuário. Tradicionalmente, busca-se pelo píxel que corresponde ao centro dos anéis pois este ponto coincide com o centro óptico do sistema e, neste ponto, $\theta = 0^\circ$ na equação 2.4, o que facilita cálculos posteriores.

Para encontrar o píxel de referência, fizemos com que o script faça um corte em XZ e em YZ. Isto por duas razões: primeiramente, por sabermos que o mapa de fase de um FP pode ser aproximado por uma superfície parabólica (ver equação 4.20). Segundo, como o parabolóide possui um eixo de simetria paralelo a Z, a intersecção dele com qualquer plano que também seja paralelo a Z resultará numa parábola e uma curva em duas dimensões é muito mais simples de ser ajustada por um polinômio do que uma superfície em três dimensões. Não somente isso, mas no caso de uma parábola descrita por $f(x) = ax^2 + bx + c$, se soubermos a e b , sabemos que seu ponto de inflexão (mínimo ou máximo) se encontra em $x = -b/2a$. Tendo o ponto de inflexão na direção X e na direção Y, temos o centro dos nossos anéis (figura 4.20). Testes indicam que este algoritmo possui suas limitações e funciona somente no caso em que os anéis estejam próximos ao centro. Entretanto, ele é rápido e simples de compreender e funciona para os dados obtidos com o BTFI.

Caso a opção `-s` seja fornecida ao script, ele mostrará o espectro de referência e mos-

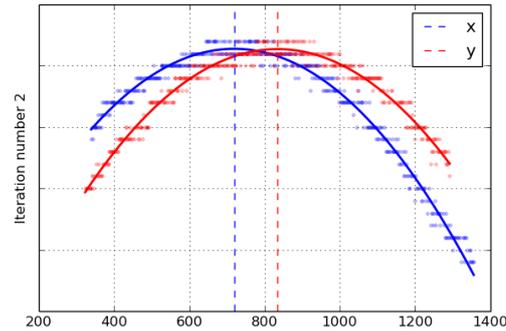


Figura 4.20: Encontrando o centro dos anéis do FP. Os pontos correspondem aos valores encontrados pela função argmax nos cortes dos planos YZ e XZ, as linhas contínuas correspondem às parábolas ajustadas na direção X (em azul) e na direção Y (em vermelho). As linhas tracejadas correspondem ao centro encontrado.

trará as medidas da largura de uma das linhas segundo a definição de *full-width-at-half-maximum* e segundo o ajuste da linha por uma função gaussiana. A versão atual nos mostra as duas opções de FWHM pois, apesar de usar FWHM calculada através do ajuste de uma gaussiana, em alguns casos a outra opção pode ser mais confiável na hora de calcularmos qual o tamanho ideal de um cubo durante uma observação em termos de números de canais.

De acordo com os testes que realizamos, este método funciona muito bem no caso de haver poucos anéis no cubo de dados. Para cubos com mais anéis, ele ainda precisa ser aprimorado.

A partir daqui, o processo de extração do mapa de fase de um cubo obtido com FP é idêntico ao processo que utilizamos para extrair um mapa de fase de um cubo do iBTF. Apresentamos o mapa de fase observado usando o cubo de correlação na figura 4.22a. Para construir este mapa, o programa utiliza somente os píxeis cujo valor seja maior do que um desvio padrão sobre a média do cubo inteiro. Neste mapa também podemos ver a presença do FSR, mostrada no salto que é dado de -270 bcv à 440 bcv.

Em seguida, usamos o script `phmfit` para realizar o ajuste do mapa de fase e corrigir os efeitos do FSR. Novamente o programa usa uma grade de píxeis para realizar o ajuste, aumentando a eficiência computacional. Logo em seguida, ele calcula (e mostra) o perfil radial do mapa de fase centrado no píxel de referência. Em FPs convencionais, o mapa de fase é um parabolóide com a concavidade voltada para cima. No caso do BTFI, o FP foi programado acidentalmente de uma forma diferente e o parabolóide é voltado para

baixo. Isto é determinado pelo script que, após organizar o perfil radial do mapa de fase observado, pega o sinal da mediana da derivada de primeira ordem.

Depois, ele ajusta uma parábola ao perfil radial para todos os pontos que estão dentro de um FSR (pontos em vermelho na figura 4.21a). Em seguida, é calculado o erro deste ajuste. Para pontos muito distantes da parábola ajustada, o programa entende que é necessário compensar o FSR. Ele realiza esta compensação e refaz o ajuste. O resultado final é apresentado na figura 4.21b. As figuras 4.22b e 4.22c apresentam o mapa de fase ajustado e o resíduo, respectivamente.

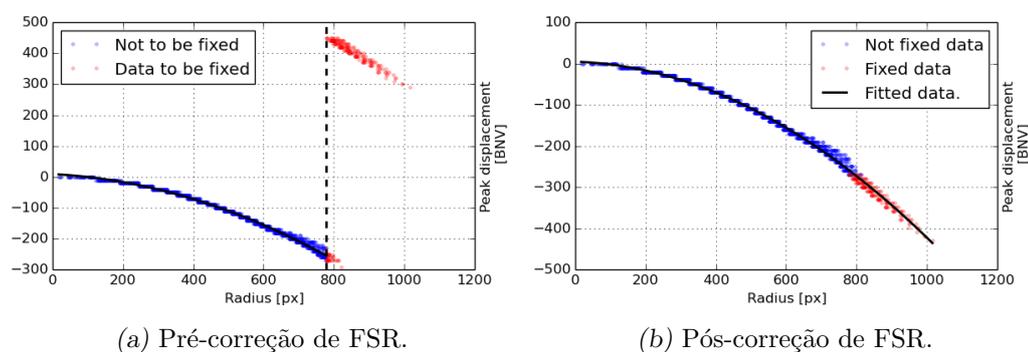


Figura 4.21: Perfil radial do mapa de fase de um FP. Na figura à esquerda, os pontos em azul são os pontos considerados no ajuste da parábola e os em vermelho são excluídos. Na figura à direita, os pontos em vermelho são compensados pelo FSR antes do reajuste da parábola.

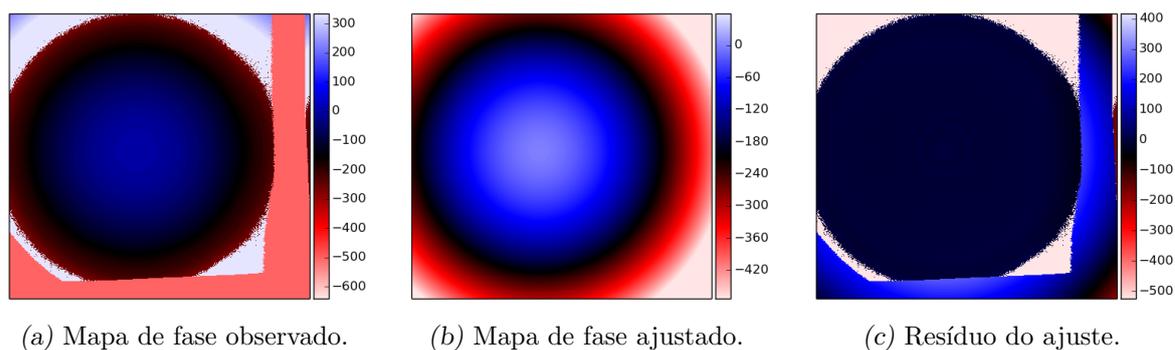


Figura 4.22: Mapas de fase extraído do cubo obtido com o FP. Os mapas de cores estão em bcvs.

O resíduo nos mostra que as faixas na lateral direita e na parte inferior do mapa são problemáticas. Mas isto já era esperado pois são regiões onde quase não há incidência de luz. Ele também mostra um erro elevado nas bordas, seguindo o formato do anel, mas isto está relacionado com o FSR.

O processo de aplicar o mapa de fase em um cubo de dados obtido com o FP também

tem suas diferenças em relação aos cubos obtidos com o iBTF. A grande diferença está no fato de que cada espectro do FP se repete a cada FSR. Neste caso, os cubos do FP têm a vantagem pois a informação, mesmo que restrita a um FSR, não é jogada fora. Durante a aplicação da correção de fase, cortamos o cubo em um FSR. Logo em seguida, deslocamos o espectro em uma direção, de acordo com o valor do píxel correspondente no mapa de fase. A porção do espectro que ficaria para fora de uma das extremidades do cubo de dados é destacada e colocada na extremidade oposta. O resultado é um deslocamento cíclico ao longo do eixo espectral. Como se houvesse infinitos cubos colados um no outro na direção espectral, fizéssemos o deslocamento dos espectros e, depois cortasse o cubo em um FSR.

Aplicamos a correção de fase no cubo utilizado para extrair os mapas de fase aqui apresentados e fizemos um mapa com a posição da linha após a correção de fase. Usando 10 pontos por canal na superamostragem, o erro foi de 20 bcvs, ou dois canais. Este erro está relacionado a um mal ajuste do mapa de fase nas bordas do cubo.

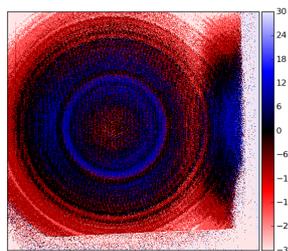


Figura 4.23: Resultado da correção de fase de um cubo de FP usando diferentes números de pontos no momento da superamostragem que precede o deslocamento dos espectros ao longo do cubo. Regiões mais claras indicam regiões com maior problema na correção de fase. O mapa de cores está em bcvs.

4.7 Calibração em comprimento de onda

Após aplicada a correção de fase, o próximo passo a ser realizado é a calibração em comprimento de onda dos cubos obtidos com o iBTF e com o FP. Ainda não trabalhamos em um script que faça este tipo de operação para os cubos do BTFI mas explicaremos quais passos temos dado para realizá-la.

4.7.1 Calibrando cubos do iBTF

A calibração em comprimento de onda dos cubos de dados do iBTF é, de certa forma, trivial e semelhante à calibração de um espectro unidimensional. Tradicionalmente, a

identificação das linhas e a calibração em comprimento de onda é feita utilizando o IRAF junto com as rotinas `identify` e `dispcor`.

O procedimento a ser realizado é o mesmo mas as utilizamos no espectro de referência que é produzido durante a correção de fase dos cubos do iBTF. O espectro de referência é o espectro do píxel de referência, conforme explicamos anteriormente. Seja um cubo de dados nomeado `nome_do_arquivo.fits`, o arquivo que contém o espectro de referência recebe um sufixo `--ref_spec` logo antes da extensão (`.fits`) do arquivo (resultado: `nome_do_arquivo--ref_spec.fits`).

No caso de um espectro muito ruidoso, o usuário ainda pode abrir o cubo de dados corrigido no QFitsView e usar a seguinte linha de comando para extrair o espectro médio de uma região:

```
buffer2 = avg(buffer1[x_inicial:x_final,y_inicial:y_final,*], /z)
```

E salvar o espectro resultante acessando o menu *File, Save as FITS*. No IRAF, o usuário pode chamar a rotina `identify` simplesmente digitando:

```
identify nome_do_arquivo.fits
```

Na janela que abrir, o usuário deve posicionar o cursor próximo a uma das linhas de emissão e pressionar a tecla `m`, digitar o comprimento de onda correspondente àquela linha em ângstroms e apertar a tecla `ENTER`. Este procedimento deve ser feito para todas as linhas. Após fazer isso com todas as linhas, o usuário deve apertar a tecla `f` para fazer um ajuste das posições e a tecla `q` para aceitar o ajuste. Então, o IRAF mostrará o espectro calibrado em comprimento de onda. Ao apertar `q` novamente, o foco da tela voltará para o terminal onde o IRAF pergunta se o usuário deseja que as informações fiquem armazenadas no banco de dados local. Apertando a tecla `ENTER`, o usuário aceita o armazenamento. Para mais informações sobre a rotina `identify`, o usuário pode apertar a tecla `?` ou buscar pelo manual na página do IRAF na Internet¹.

Para aplicar a correção do espectro nele mesmo, usamos a rotina `dispcor` simplesmente digitando:

```
dispcor espectro_de_entrada.fits espectro_de_saida.fits
```

¹ <http://iraf.noao.edu/>

Após este comando, o `dispcor` retornará diversos parâmetros. Os parâmetros que nos interessam são do comprimento de onda do primeiro píxel do espectro `w1` e a incremento em comprimento de onda entre píxeis `dw`. Para aplicá-los ao cubo de dados, usamos o `QFitsView`.

Estes parâmetros devem ser acrescentados ao cabeçalho do arquivo FITS que contém o cubo. O campo `CRPIX3` (número do canal de referência), deve ser 1. O comprimento de onda inicial `w1` deve ser armazenado no campo `CRVAL3` (valor do canal de referência). O passo em comprimento de onda `dw` deve ser armazenado nos campos `CDEL3` e `C3_3` (incremento entre canais). Caso o campo `DISPAXIS` não esteja presente, ele precisa ser acrescentado também e o valor atribuído a ele deve ser 3 (o que corresponde ao terceiro eixo ou eixo Z).

Com isto temos a garantia de que a calibração espectral do cubo de dados funciona tanto no IRAF quanto no `QFitsView`.

4.7.2 Calibrando cubos do Fabry-Perot

O processo de calibração em comprimento de onda em cubos de dados obtidos com Fabry-Perots costuma ser muito mais complexo que em cubos obtidos com o iBTF. A principal razão reside no fato de que os espectros obtidos com os FPs serem limitados ao *free-spectral-range*, conforme já definimos anteriormente, e que linhas cuja distância seja maior do que $\Delta\lambda$ podem se sobrepor de tal forma que fica impossível distinguí-las ou identificá-las.

A calibração exige que tenhamos dois cubos de dados obtidos com linhas espectrais diferentes e identificadas. Idealmente, deveríamos ter dois cubos com uma fonte monocromática cada um. Este tem sido um dos maiores problemas na calibração em comprimento de onda no BTFI pois, atualmente, temos somente uma combinação usando as lâmpadas de calibração internas ao SOAR com os filtros do BTFI que fornecem uma única linha de emissão. No caso, a lâmpada de Neônio com o filtro 6600/20.

Todos as outras combinações entre filtros e lâmpadas acabam por fornecer mais de uma linha ou nenhuma linha. Entretanto, depois de vários testes, descobrimos que, ao obter um cubo de calibração com o filtro 6745/40, juntamente com a lâmpada de Neônio, podemos ver uma outra linha. Analisando o espectro do Ne, a linha mais próxima é 6717 Å. Acreditamos que, mesmo estando fora da banda espectral em que o filtro permite a

passagem da luz, a sua proximidade com ela faz com que uma pequena porção de luz seja transmitida, o que é comprovado pois o cubo é construído usando um tempo de exposição muito maior que o de um cubo obtido com o filtro 6600/20.

Para tornar mais fácil a explicação, vamos retomar a equação que determina o critério de transmissão de luz pelo FP:

$$m\lambda = 2nd \cos \theta \quad (4.2)$$

Quando fazemos uma varredura com o FP, o parâmetro que estamos mudando é o d . Entretanto, quando controlamos o FP pelo computador, não temos acesso ao valor real de d . O que temos, é um parâmetro z binário que possui alguma relação com d . Em primeira aproximação, podemos dizer que a relação entre z e d é linear. Matematicamente falando, consideramos a seguinte relação:

$$d = Az + B \quad (4.3)$$

Substituindo d da equação 4.3 na equação 4.2, temos:

$$m\lambda = 2n \cos(\theta) (Az + B) \quad (4.4)$$

O ângulo θ é o ângulo que um raio de luz faz com o eixo óptico. No caso, se estivermos fazendo esta análise no centro dos anéis, podemos considerar que $\theta = 0^\circ$ e, com isso, $\cos(\theta) = 1$. Como normalmente o meio que permeia a cavidade óptica entre as duas placas de vidro do FP é o ar, podemos considerar que o índice de refração é uniforme e aproximadamente 1. Com estas duas hipóteses, a equação 4.4 toma uma forma mais simples:

$$\frac{m\lambda}{2} = a + bz \quad (4.5)$$

Ou ainda

$$m\lambda = A + Bz \quad (4.6)$$

Ou seja, sabendo o valor dos coeficientes A e B e a ordem de interferência m em que estamos trabalhando, sabemos qual o comprimento de onda de uma linha que apareça no centro dos anéis dos cubos de um FP.

No caso do BTFI, uma outra informação a ser considerada é a direção em que o raio dos anéis aumenta. O raio de um anel está relacionado a θ por:

$$\theta = \arctan\left(\frac{rp}{f}\right) \quad (4.7)$$

Onde r é o raio dos anéis em píxeis, p é o tamanho de um píxel em milímetros e $f = 355\text{mm}$ é a distância focal da lente imageadora em milímetros. Analisando a equação 4.4, se pudermos obter uma imagem com vários anéis e tivermos um anel com ordem m , o primeiro anel externo a ele terá ordem $m - 1$ e o primeiro anel interno a ele terá ordem $m + 1$. Isto é importante pois, considerando o caso de uma fonte monocromática e o caso em que o raio dos anéis cresce com z dentro de um cubo de dados, se extrairmos um espectro do centro destes anéis, a primeira linha que aparecer terá ordem m , a segunda $m + 1$, a terceira $m + 2$, etc. Se os anéis diminuïrem com z , as ordens de interferência das linhas deste espectro são m , $m - 1$, $m - 2$, etc.

Juntando todas estas informações, podemos, agora, começar a descobrir o valor dos parâmetros da equação 4.6. Vamos tomar o espectro do centro de um cubo de dados obtido com uma lâmpada de Neônio usando o filtro 6600/20 (figura 4.24). Nela podemos ver as duas ordens m_1 e $m_1 - 1$ da linha $\lambda_1 = 6598 \text{ \AA}$. Podemos também ver o problema do *bump* espectral descrito na seção 3.2.2.

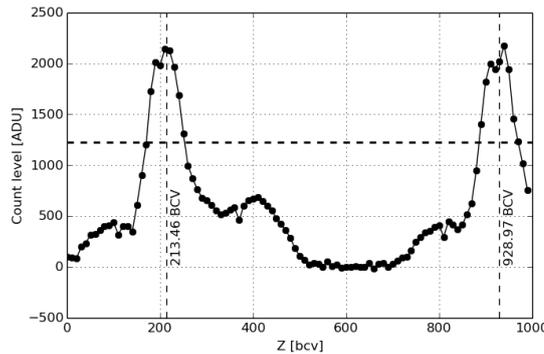


Figura 4.24: Espectro da linha de Ne usando HR FP e filtro 6600/20 extraída do centro dos anéis do cubo de dados.

Segundo esta figura, temos dois picos: $z_1(m_1) = z_1 = 209,93bcv$ e $z_1(m_1 - 1) = z'_1 = 929,38bcv$. Estes picos foram calculados usando o baricentro das linhas considerando somente os dados acima da média do espectro somada ao desvio padrão do espectro (linha tracejada na horizontal). Com estes dois valores podemos aplicar a equação 4.6 para as

duas ordens de interferência:

$$m\lambda_1 = A + Bz_1 \quad (4.8)$$

$$(m - 1)\lambda_1 = A + Bz'_1 \quad (4.9)$$

Subtraindo a equação 4.9 da equação 4.8, temos:

$$\begin{aligned} m_1\lambda_1 - (m_1 - 1)\lambda_1 &= A + Bz_1 - A - Bz'_1 \\ \lambda_1 &= B(z_1 - z'_1) \\ \lambda_1 &= B\Delta z_1 \\ B &= \frac{\lambda_1}{\Delta z_1} \\ B &= \frac{6598\text{\AA}}{209,93bcv - 929,38bcv} B = -9,17\text{\AA}.bcv^{-1} \quad (4.10) \end{aligned}$$

Onde definimos $\Delta z_1 = z_1 - z'_1$ como o FSR para o comprimento de onda 1, no caso 6598 Å. O parâmetro B nos dá uma relação direta da variação em ângstroms para cada BVC. O sinal negativo em B é uma particularidade dos dados do BTFI devido ao modo com que o programa que controla o FP foi programado. Em outros FP, B normalmente é positivo. Uma vez que, para determinar A , precisamos de mais informação, vamos utilizar o cubo obtido com a lâmpada de Ne com o filtro 6745/40, onde a linha $\lambda_2 = 6717$ Å aparece (figura 4.25).

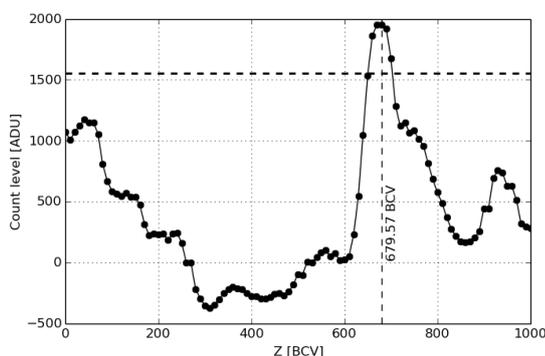


Figura 4.25: Espectro da linha de Ne usando HR FP e filtro 6745/40 extraída do centro dos anéis do cubo de dados.

A posição da linha z_2 é novamente calculada sobre o baricentro considerando os pontos

sobre duas vezes o desvio padrão somado à média do espectro (linha tracejada preta na horizontal). Desta forma obtemos que $z_2 = 679,62bcv$.

Usando a equação 4.6 escrita para os dois comprimentos de onda, temos:

$$m_1\lambda_1 = A + Bz_1 \quad (4.11)$$

$$m_2\lambda_2 = A + Bz_2 \quad (4.12)$$

Se soubéssemos as ordens de interferência m_1 e m_2 , encontrar A seria trivial. Entretanto, nos falta ainda informação. Trabalhando estas duas equações, podemos, inclusive, encontrar uma relação entre as ordens de interferência m_1 e m_2 . Usando estas duas equações, temos:

$$\begin{aligned} m_1\lambda_1 - m_2\lambda_2 &= A + Bz_1 - A - Bz_2 \\ m_1\lambda_1 - m_2\lambda_2 &= Bz_1 - Bz_2 \\ m_1\lambda_1 - m_2\lambda_2 &= -\frac{\lambda_1}{\Delta z_1}z_1 + \frac{\lambda_1}{\Delta z_1}Bz_2 \\ m_1\lambda_1 - m_2\lambda_2 &= \frac{\lambda_1}{\Delta z_1}(z_2 - z_1) \\ -m_2\lambda_2 &= -m_1\lambda_1 + \frac{\lambda_1}{\Delta z_1}(z_2 - z_1) \\ m_2 &= m_1\frac{\lambda_1}{\lambda_2} + \frac{z_2^2 - z_1^2}{\Delta z_1}\frac{\lambda_1}{\lambda_2} \\ m_2 &= \frac{\lambda_1}{\lambda_2}\left(m_1 + \frac{z_2 - z_1}{\Delta z_1}\right) \end{aligned} \quad (4.13)$$

Para encontrar m_1 e m_2 , usamos também a restrição de que ambos devem ser números inteiros (Atherton et al., 1982). Mesmo tendo uma relação entre as ordens de interferência, ainda temos informação faltando. Precisamos descobrir d , A , m_1 e m_2 . Para prosseguir, usamos a informação do fabricante de que o valor nominal do *gap* do FP é de $200\mu\text{m}$ (discutiremos o caso em que este parâmetro não é conhecido posteriormente). Com isto, temos que:

$$\begin{aligned} m_1 &= \frac{2d}{\lambda_1} \\ m_1 &= 606,244 \end{aligned} \quad (4.14)$$

Tabela 4.3 - Tabela usada para encontrar as ordens de interferência.

Nela, são indicados também os valores onde começamos a atribuir valores para m_1 e qual o valor de m_1 encontrado de modo que m_2 esteja o mais próximo possível de ser um número inteiro.

m_1	m_2	Distância de um número inteiro	
608	596.6714	0.3286	
607	595.6889	0.3111	
606	594.7065	0.2935	Chute inicial
605	593.7241	0.2759	
604	592.7417	0.2583	
603	591.7593	0.2407	
602	590.7768	0.2232	
601	589.7944	0.2056	
600	588.8120	0.1880	
599	587.8296	0.1704	
598	586.8471	0.1529	
597	585.8647	0.1353	
596	584.8823	0.1177	
595	583.8999	0.1001	
594	582.9175	0.0825	
593	581.9350	0.0650	
592	580.9526	0.0474	
591	579.9702	0.0298	
590	578.9878	0.0122	
589	578.0053	0.0053	Ordens escolhidas
588	577.0229	0.0229	
587	576.0405	0.0405	

Criamos a tabela 4.3 atribuindo valores inteiros para m_1 tendo como referência o valor inicial encontrado na equação 4.14 e assumindo a condição de que m_2 seja um número inteiro. Para facilitar a visualização, a figura 4.26 mostra um gráfico baseado nesta tabela onde temos o quanto m_2 está longe de ser um número inteiro em função de m_1 .

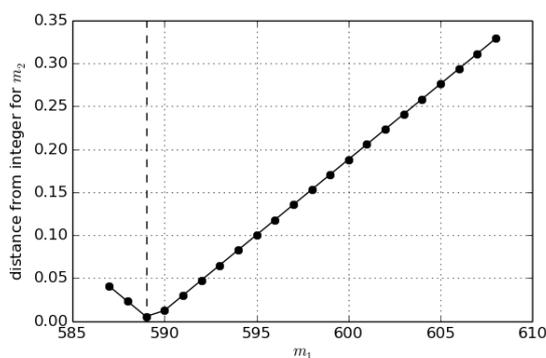


Figura 4.26: Gráfico para encontrar as ordens de interferência através da distância entre o valor encontrado para m_2 e um número inteiro.

Assim sendo, temos que $m_1 = 589$ e $m_2 = 578$. Com eles, podemos encontrar o valor de A :

$$A = m_1 \lambda_1 - Bz_1$$

$$A = 3888708,78 \text{ \AA} \quad (4.15)$$

Tendo A , o processo de calibração em comprimento de onda está quase completo. Para testá-lo, calculamos o comprimento de onda usando A e B encontrados considerando m_2 . Como resultado, tivemos 6717,10 Å, o comprimento de onda λ_2 , um erro de 0,06 Å.

O último passo, agora, consiste em aplicar a calibração a um cubo de dados obtido com o BTFL. Para começar, tomamos o valor em bcv do canal de referência. O campo correspondente ao canal de referência no cabeçalho de um cubo de dados é CRPIX3 e o campo correspondente ao seu valor é CRVAL3. Caso eles não existam no cabeçalho, ambos devem valer 1.

Mantendo o canal de referência CRPIX3, o novo valor que ele assume CRVAL3 é o comprimento de onda calculado através da equação 4.16.

$$\text{CRVAL3}(\lambda) = \frac{A + B \text{CRVAL3}(z)}{m} \quad (4.16)$$

O outro parâmetro importante na calibração de um cubo de dados é o campo CDELTA3 do cabeçalho do arquivo FITS. Ele nos diz qual o incremento a ser dado a cada canal. Dependendo do software utilizado, o campo lido também pode ser C3_3. Os dois são equivalentes e os dois precisam ser atualizados. No caso de termos CDELTA3 em unidades bcv, a transformação a ser feita é:

$$\text{CDELTA3}(\lambda) = \frac{B}{m} \text{CDELTA3}(z) \quad (4.17)$$

Com isso teremos qual o incremento em comprimento de onda para cada canal. Caso nem CDELTA3 nem C3_3 estejam no cabeçalho, devemos assumir que eles valem 1. Se quisermos trabalhar em um comprimento de onda muito distante do comprimento de onda usado na calibração, devemos ir variar m até encontrar na equação 4.16 até um intervalo espectral que cubra uma das linhas esperadas em uma observação.

Por exemplo, do processo de extração do mapa de fase, sabemos que o FSR deste cubo é de 719,45 bcv, equivalente a 11,20 Å. O comprimento de onda usado na calibração foi 6598 Å e queremos aplicar esta calibração em um cubo observado em H α 6563 Å. O FSR para H α é o FSR 11,20 Å dividido por 6598 Å e multiplicado por 6563 Å, resultando em 11,26 Å.

O comprimento de onda do primeiro canal do cubo de calibração é 6602,22 Å. Como H α está a aproximadamente cinco vezes o FSR abaixo deste valor, recalculamos CRVAL3

por volta de cinco ordens abaixo. Após jogar um pouco com os valores, encontramos $CRVAL3 = 6566,30 \text{ \AA}$ com seis ordens abaixo da ordem do comprimento de onda utilizado na calibração.

Agora, em casos onde o valor do *gap* é desconhecido, temos que recorrer a uma outra variável. No caso, utilizamos dois anéis de um mesmo comprimento de onda obtidos em uma única imagem. Devido à distância focal do BTFI e ao tamanho do CCD, nós não conseguimos um cubo com estas características. Para um determinado comprimento de onda, quando o anel de uma determinada ordem começa a surgir no centro da imagem, o anel da ordem seguinte já está desaparecendo.

Sendo assim, para poder ilustrar o exemplo da determinação do *gap* do LR FP em laboratório, utilizamos uma câmera Nikon D3100 com distância focal de $f = 18\text{mm}$. O detector desta câmera possui $23,1 \text{ mm} \times 15,4 \text{ mm}$. Para imagens com 2304×1536 pixels, o tamanho de cada pixel é de $0,01 \text{ mm}$. Escrevendo, agora, $\cos \theta$ em função da equação 4.7 para ângulos pequenos:

$$\begin{aligned}\theta &= \arctan\left(\frac{rp}{f}\right) \\ &\approx \arcsin\left(\frac{rp}{f}\right) \\ \theta &\approx \frac{rp}{f}\end{aligned}\tag{4.18}$$

$$\begin{aligned}\cos \theta &\approx 1 - \frac{\theta^2}{2} \\ \cos \theta &\approx 1 - \frac{r^2 p^2}{2f^2}\end{aligned}\tag{4.19}$$

Para $n = 1$, juntando as equações 4.6 e 4.19, temos:

$$m\lambda = 2d \left(1 - \frac{r^2 p^2}{2f^2}\right)\tag{4.20}$$

Esta equação é interessante pois está relacionada com o fato do mapa de fase de um FP ter um formato parabólico, devido à sua dependência com r^2 . Isolando d , temos:

$$d \approx \frac{m\lambda}{2 - r^2 p^2 / f^2}\tag{4.21}$$

Chamando o raio do anel de ordem m de r e o raio do próximo anel externo ao primeiro, de ordem $m - 1$, de r' e tomando uma única imagem, temos:

$$\begin{aligned}\frac{m\lambda}{2 - r^2p^2/f^2} &= \frac{(m - 1)\lambda}{2 - r'^2p^2/f^2} \\ \frac{m}{m - 1} &= \frac{2 - r^2p^2/f^2}{2 - r'^2p^2/f^2} = \beta \\ m &= \beta(m - 1) \\ m - \beta m &= -\beta \\ m(1 - \beta) &= -\beta \\ m &= \frac{\beta}{1 - \beta} \\ m &= \frac{\beta}{\beta - 1}\end{aligned}\tag{4.22}$$

Como fonte, usamos uma lâmpada de mercúrio por ter uma única linha bem proeminente na região verde do espectro. Usando esta linha, medimos os dois primeiros raios internos $r = 48 \text{ px}$ e $r' = 216 \text{ px}$.

Usando estes valores, temos que $\beta = 1,006894$ e $m = 146.05$. Tomando a parte inteira da ordem de difração, considerando $\lambda = 5460,735 \text{ \AA}$ e usando a equação 4.21, temos que $d = 39,9 \text{ }\mu\text{m}$. Apesar de não corresponder exatamente ao valor nominal de $44 \text{ }\mu\text{m}$, acreditamos que o valor esteja coerente e que o erro seja proveniente do modo rudimentar com que o raio dos anéis foi medido (por inspeção visual).

Usando os espectros de cubos de calibração já calibrados em comprimentos de onda, conseguimos, finalmente, obter os parâmetros experimentais para os dois Fabry-Perots. Estes parâmetros são mostrados nas tabelas 4.4 e 4.5. Os símbolos de maior e menor na tabela possuem relação com o *bump* que degradou o formato da linha. No caso, estes valores foram calculados com um espectro calibrado usando o comando `k-k` do `splot` no IRAF. Estes valores foram calculados tomando somente o topo das linhas. Caso considerássemos as bordas, a FWHM é maior, influenciando todos os outros valores derivados dela. No caso, estas tabelas também nos mostram que os efeitos do *bump* degradaram mais as linhas espectrais do HR FP do que do LR FP. Acreditamos que, com dados obtidos com o CS100 havaiano, possamos encontrar uma melhor resolução espectral.

Tabela 4.4 - HR FP - Parâmetros experimentais obtidos para o Fabry-Perot de alta resolução.

λ_{ref}	6598,9529 Å
FWHM	> 0,6945 Å
FSR	11,20 Å
R	< 9506
m	589
F	< 16.1

Tabela 4.5 - LR FP - Parâmetros experimentais obtidos para o Fabry-Perot de baixa resolução.

λ_{ref}	6598,9529 Å
FWHM	> 1,5 Å
FSR	49,6 Å
R	< 4399
m	97
F	33

4.8 Calibração astrométrica e Ângulo de Posição

A calibração astrométrica de uma imagem consiste em uma transformação de coordenadas de pixels para coordenadas astronômicas. Em geral, são usadas a ascensão reta e a declinação. Os parâmetros utilizados para esta transformação são armazenados no cabeçalho das imagens e cubos FITS.

No caso do BTFI, o PanView armazena no cabeçalho de cada arquivo a ascensão reta e a declinação do centro do campo observado nas palavras chave MOUNTRA, MOUNTDEC. Isto fornece ao usuário uma informação prévia de para onde o telescópio estava apontando no momento da aquisição dos dados. Entretanto, informações sobre a direção em que a imagem foi obtida ou quantos segundos de arco equivalem cada píxel não estão disponíveis no momento.

Também não ficam disponíveis nos *headers* a direção para onde o leste e o norte apontam em uma imagem ou cubo de dados e o ângulo de posição dos espelhos internos ao telescópio. O ângulo de posição (PA- *position angle*) é uma informação dada pelo operador do telescópio na hora da aquisição.

O método ideal para descobrirmos a relação entre o PA e a orientação seria obter ima-

gens de um campo com várias estrelas identificadas alterando o ângulo de posição. Isto não foi feito pois nosso foco, durante os comissionamentos, era problemas mais fundamentais do instrumento envolvendo a parte óptica e operacional. Mesmo assim, conseguimos reunir informações necessárias para identificar a relação entre o PA e a orientação das imagens. A tabela 4.6 nos mostra exemplos de objetos que foram obtidos na noite de 2 de setembro de 2013 com o HR FP (Câmera 2) de regiões com estrelas o suficiente para podermos fazer a astrometria.

Tabela 4.6 - Orientação e Ângulo de Posição

Objeto	RA-DEC	Ângulo de Posição	Norte	Leste
Hourglass	18:03:41.047 -24:22:39.994	0°	↓	←
NGC346	24:59:04.142 -72:10:48.094	270°	←	↑

Para poder determinar a relação do PA e da direção das imagens obtidas, realizamos a calibração astrométrica em cada um dos cubos acima utilizando o *Aladin Sky Atlas* (Bonnarel et al., 2000). Ao realizar este tipo de calibração, o *Aladin* nos fornece os campos e valores a serem incluído nos cabeçalhos dos arquivos FITS, conforme o exemplo dado na tabela 4.8.

```

COMMENT = 'General Information' / Aladin add
COMMENT This astrometrical calibration was computed via Aladin
CRPIX1  =   182.42767863420855
CRPIX2  =   266.55590649424425
EQUINOX =                   2000.0
CRVAL1  =   270.93791666666664
CRVAL2  =  -24.368194444444445
CTYPE1  = 'RA---TAN'
CTYPE2  = 'DEC--TAN'
RADECSYS= 'ICRS      '
CD1_1   = -6.448595426911539E-5
CD1_2   = -2.981199464014841E-6
CD2_1   =  2.981199464014841E-6
CD2_2   = -6.448595426911539E-5

```

Exemplo de cabeçalho 1: Exemplo de parâmetros a serem adicionados ao cabeçalho de um arquivo FITS para calibração astrométrica.

Com estes dados sabemos, então, a relação entre os ângulos de posição e a orientação das imagens, conforme apresentado na tabela 4.7.

Tabela 4.7 - Orientação das imagens em função do ângulo de posição do telescópio.

PA	Norte	Leste
0°	↓	←
90°	→	↓
180°	↑	→
270°	←	↑

Para o futuro, acreditamos que seja essencial a modificação do código fonte do KAFKA para que ele leia a informação do PA diretamente do telescópio podendo, assim, armazená-la no cabeçalho das imagens. Com isso, poderemos criar um novo script que encontre os parâmetros necessários para uma astrometria inicial para auxiliar na localização de objetos.

4.9 Alguns resultados observacionais

O BTFI tem obtido dados observacionais desde julho de 2010. Entretanto, nenhum destes dados possuía qualidade suficiente para compararmos com dados obtidos com outros instrumentos ou mesmo para publicarmos estes resultados em uma revista científica. Eis alguns exemplos.

4.9.1 Observação com o iBTF - NGC2440

Os dados apresentados nas figuras 4.27a e 4.27b foram obtidos na noite de 6 de dezembro de 2012 utilizando o iBTF no modo de reflexão utilizando as redes de difração holográficas 16809 e 16819 no modo *double-pass*. Nestes dados, fizemos correção de BIAS estimado pela região de OVERSCAN, correção de DARK, FLAT, correção de fase e calibração em comprimento de onda.

Este cubo de dados foi obtido através de uma varredura de $39,00^\circ$ até $43,15^\circ$ com passos de $0,05^\circ$ (83 canais). Em cada posição, foram obtidas três imagens de cinco segundos. A observação demorou 37 minutos e 26 segundos.

A figura 4.27a contém uma imagem obtida através do colapso em comprimento de onda do cubo de dados observado. Os três círculos coloridos indicam a posição onde os espectros da figura 4.27b foram extraídos. Por exemplo, o primeiro espectro (vermelho) foi retirado do centro da nebulosa (círculo vermelho). Os espectros foram normalizados de modo que a intensidade da linha de [NII] 6584 Å valesse 1.

Após a correção de fase, tivemos que cortar parte do cubo. A parte aproveitada contém um espectro que vai de 6532,36 Å até 6624,8 Å com passos de -1,54 Å por canal (decrecendo conforme o canal cresce) ou com -30,8 Å por grau. Usando o espectro representado em vermelho na figura 4.27b, obtivemos uma FWHM de 3,61 Å em 6563 Å e 5,05 Å em 6584 Å, o que nos dá uma resolução espectral aproximadamente de 1500.

De modo a ilustrar um pouco do que podemos fazer com um cubo de dados, apresentamos um mapa de estruturas em RGB na figura 4.28a. Para construí-lo, nós tiramos a média de diferentes regiões espectrais do cubo e as definimos como diferentes cores, conforme os espectros mostrados na figura 4.28b. Por exemplo, as duas linhas de [NII] 6548 Å e [NII] 6584 Å são mapeadas em verde, a linha de H α é mostrado em vermelho e o contínuo é mostrado em azul. O problema destas observações é a baixa qualidade da imagens obtidos

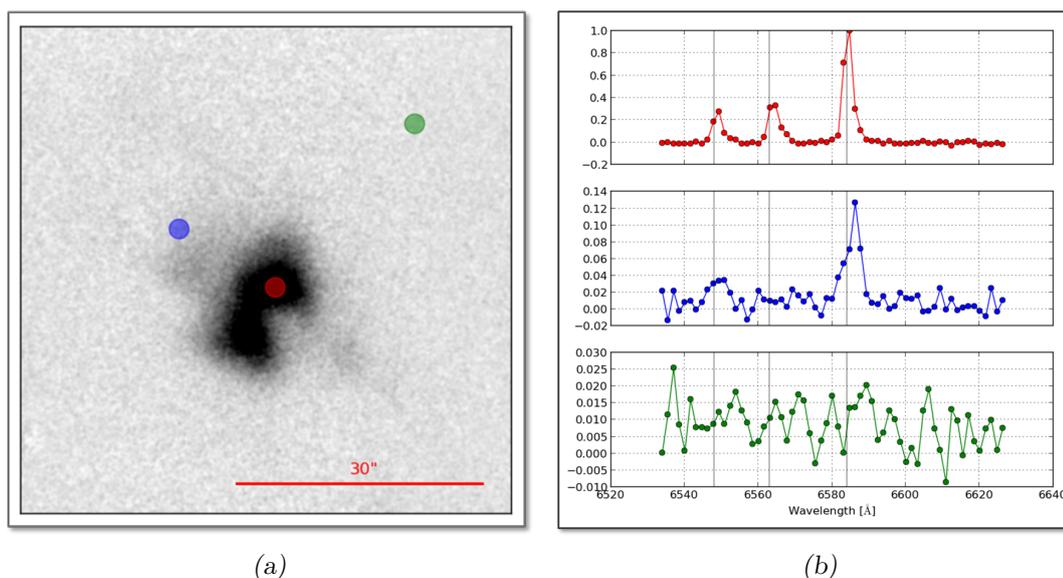


Figura 4.27: NGC2440 observado com o iBTF. (a) Cubo colapsado em λ . (b) Espectro normalizado de diferentes regiões

devido ao uso das redes ONDAX.

4.9.2 Observação com o LR FP - NGC7009

Os dados relacionados à nebulosa NGC7009, ou nebulosa Saturno, foram obtidos na noite de 24 de agosto de 2013 utilizando o Fabry-Perot Queensgate de baixa resolução e um filtro interferencial com largura de banda de 75 Å centrado na linha de H α 6563 Å. As figuras 4.29a e 4.29b representam o cubo colapsado em comprimento de onda e os espectros em diferentes regiões do cubo.

Para esta observação, foi feita uma varredura com 100 passos (ou 101 canais), com cinco imagens em cada um dos canais. Cada passo possui um incremento de 7 bcv sendo que o primeiro canal iniciou em 200 bcv. Cada imagem possuiu um tempo de exposição de 1.0 segundos. Uma análise parecida com a anterior foi feita aqui. O problema com estes dados é a presença do *bump* explicado na seção 3.2.2 do capítulo 3. O tempo total gasto com esta observação foi de 22 minutos e 48,13 segundos, indicando uma perda operacional de 63% com o computador antigo do BTFI.

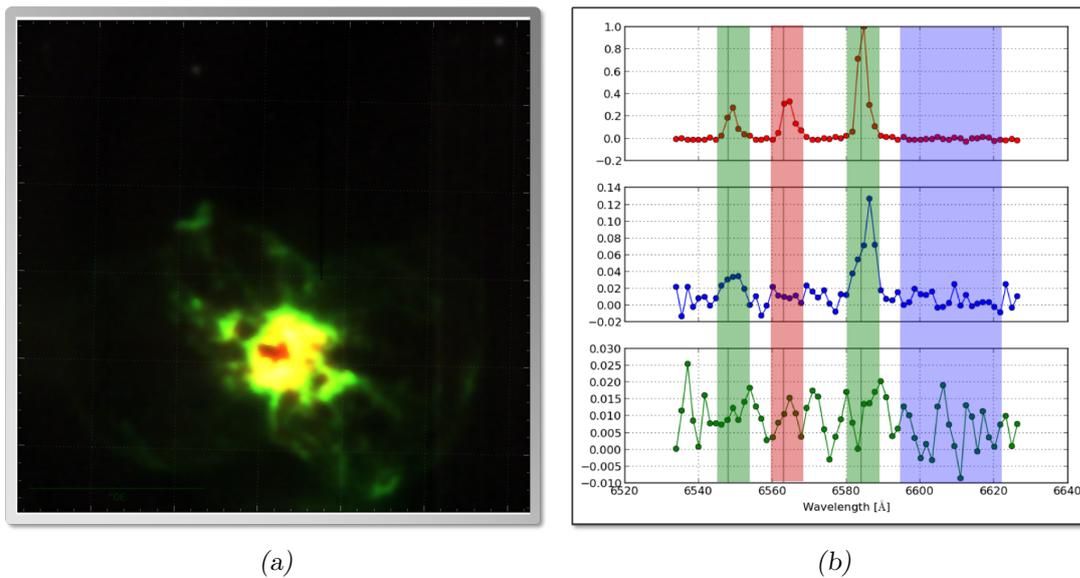


Figura 4.28: Mapa de estrutura de NGC2440. O mapa de cores à esquerda corresponde às regiões espectrais destacadas nos espectros à direita. (a) Mapa de estrutura em RGB. (b) Regiões espectrais correspondentes ao mapa de cores.

4.9.3 Observação com o HR FP - 30Dor

Os últimos dados que apresentamos correspondem a observações de 30Dor feitas no dia 2 de abril de 2014. Nela, utilizamos o FP de alta resolução espectral e o filtro estreito com largura de banda de 20 \AA centrado em 6568 \AA com 2×2 polegadas, emprestado do CTIO. A figura 4.30a nos mostra o cubo de dados com remoção de ruídos e os píxeis usados para extrair os espectros mostrados na figura 4.30b. Em vermelho, temos o espectro de uma região com emissão em $H\alpha$; em verde, temos o espectro de uma região contendo um objeto com emissão contínua e, em azul, temos o espectro de uma região sem emissão.

Para esta observação, foi feita uma varredura com 80 passos (ou 81 canais), com cinco imagens em cada uma dos canais. Cada passo possui um incremento de 10 bcv sendo que o primeiro canal iniciou em 0 bcv. Cada imagem possuiu um tempo de exposição de 5.0 segundos. Novamente o problema aqui foi a presença do *bump*, devido a um problema com um dos controladores do FP que agora já foi identificado.

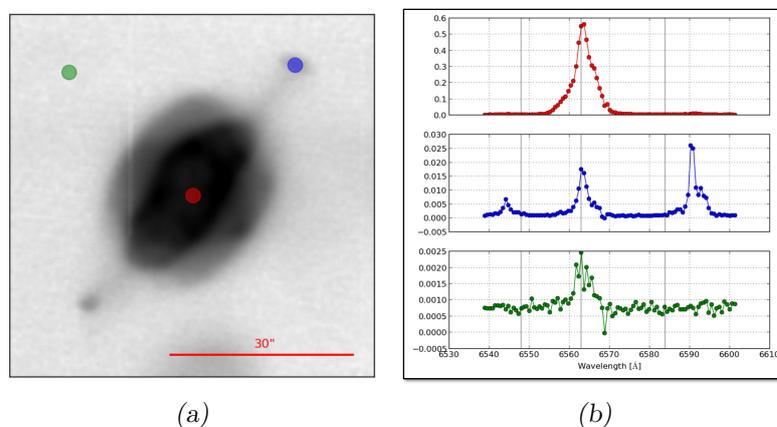


Figura 4.29: NGC7009 observada com o LR FP. (a) Cubo colapsado em λ . (b) Espectro de diferenças regiões normalizados pelo valor máximo da imagem.

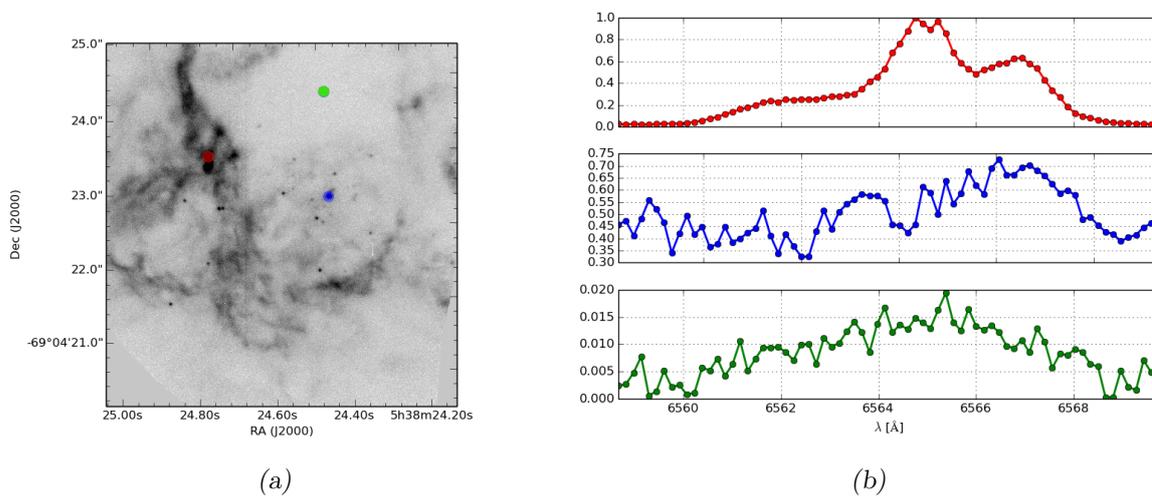


Figura 4.30: 30Dor observada com o HR FP. Os três espectros apresentados à direita correspondem aos três círculos marcados à esquerda. Em vermelho, temos uma região com emissão de $H\alpha$; em verde, uma fonte com emissão contínua e, em azul, o fundo. (a) Cubo colapsado em λ . (b) Espectro de diferenças regiões normalizados pelo valor máximo da imagem.

Conclusões

O resultado desta tese foi a análise completa dos diversos pontos do BTFI, dando suporte às modificações, aprimoramento e correções necessárias para o pleno funcionamento desse instrumento.

Um importante objetivo deste trabalho que apresentamos foi avaliar se o BTFI estava pronto para ser usado pela comunidade de astrônomos brasileiros ou não e, neste último caso, o que precisa ser melhorado no ponto de vista óptico e operacional. Para isto, apresentamos uma descrição completa da parte óptica do instrumento dando ênfase aos dois filtros ajustáveis utilizados dentro do instrumento: o Fabry-Perot e o iBTF.

Um dos pontos principais avaliados nesta tese foi a qualidade óptica instrumental no modo de imageamento simples (sem usarmos o FP ou o iBTF). Para avaliar este ponto, tivemos que fazer uma revisão do alinhamento óptico do sistema completo e fazer o foco instrumental considerando uma fonte artificial e um *pinhole* na entrada do instrumento. De acordo com os resultados apresentados na seção 3.1 do capítulo 3, as imagens obtidas com o BTFI apresentam uma PSF com FWHM de 0,30 segundos de arco e 0,26 segundos de arcos para as Câmeras 1 e 2, respectivamente. Estes valores mostram que a PSF é menor que a do *seeing* de uma noite típica de observação com o SOAR, normalmente com uma FWHM maior que 0,50 segundos de arcos, o que nos mostra que a qualidade de imagem do instrumento está dentro dos requisitos do projeto óptico inicial.

Apresentamos, também, a degradação da qualidade da imagem em função do modo de operação dos detectores do BTFI e estimamos a diferença entre o foco instrumental e o foco do telescópio a ser compensado a cada observação. De acordo com nossas medidas, precisamos realizar um deslocamento de $+60\mu m$ para compensar a diferença entre o foco instrumental e o foco do telescópio. Este deslocamento deve ser fornecido ao assistente

noturno no início de cada noite de observação.

Mostramos também que a PSF também pode ser degradada ao usar o modo EM. No caso, num campo observado com FWHM média de 0,88 segundo de arco no modo clássico, a FWHM média com o modo EM dos detectores foi de 1,36 segundo de arco. Além disso, mostramos que o tempo de leitura dos detectores do BTFI é muito menor do que para os detectores convencionais: 1,1 s para o modo clássico e 0,41 s para o modo EM, lembrando que os detectores possuem 1696 x 1616 pixels.

Já em relação ao iBTF, encontramos diversos problemas, tanto no modo de transmissão quanto no modo de reflexão. Conforme apresentamos na seção 3.3.2, ao usá-lo no modo de transmissão com as redes obtidas com a empresa Wasatch, nos deparamos com uma baixa eficiência espectral (abaixo de 50%, em média), com uma resolução espectral média de 17 e com um forte alongamento da PSF das imagens. Acreditamos que estes dois efeitos estejam relacionados ao substrato que foi utilizado nestas redes: vidro simples sem nenhum filme anti-reflexo.

No modo de reflexão, tivemos mais problemas (seção 3.3.3). O primeiro deles reside no fato de que, mesmo com um filme anti-reflexo aplicado sobre os substratos das redes, temos a presença de fantasmas que possuem um efeito maior no caso de fontes com emissão espectral contínua. Para solucionar este problema tivemos várias interações com a empresa Ondax. Como a maioria dos fantasmas que apareciam estavam relacionados ao mosaico de redes de difração construído, optamos por usar redes monolíticas com menores dimensões. Mesmo assim, tivemos ainda alguns fantasmas persistentes nas imagens gerados pelas reflexões nas superfícies do substrato da rede. Após várias interações com a Ondax, decidimos separar a ordem 1 e a ordem 0 de difração em diferentes direções mas acabamos nos deparando com uma forte degradação da PSF por todas as redes holográficas após esta mudança.

Apresentamos, também, uma análise do uso destas redes no modo *single-pass* e *double-pass*, que poderia servir como solução para alguns dos problemas. Entretanto, vimos que, no caso de redes de reflexão com alta resolução usadas dentro do BTFI, o modo *double-pass* é limitado: temos um efeito de alargamento e bifurcação das linhas espectrais causado pela diferença entre a frequência linear e pela diferença entre os ângulos de operação e ângulos de ajuste entre as duas redes.

Notamos que a resolução espectral destas redes é $R \approx 1500$, no modo *double-pass* e que

elas podem atuar de, aproximadamente, 6364 Å à 7044 Å.

Além disto, mostramos na seção 3.3.4 que a perda de tempo no sistema de controle do iBTF é de 65,06%, muito acima do ideal: por volta de 10%. Sabemos que o hardware do iBTF pode ser controlado por comandos via servidor *OPC (Object Linking and Embedding for Process Control)* através de programas escritos em LabView e acreditamos que uma nova versão do programa de interface entre o KAFKA e os PLCs escritos nesta linguagem possam elevar significativamente a resposta do sistema.

Para o futuro, esperamos que as novas redes encomendadas com a empresa Wasatch tenham eficiências mais altas e que o efeito de degradação da PSF seja menor. Já no caso das redes Ondax, não acreditamos que valha a pena continuar as interações com esta empresa visto que já esgotamos nossas idéias para melhorar as redes.

A possibilidade de uso dos FP é mais otimista. Conforme descrito na seção 3.2.2, o problema do *bump* espectral, solucionado recentemente, nos leva a crer que o BTFI possa ser oferecido neste modo. A prova do potencial do uso de FPs no SOAR foi obtida no começo deste ano quando instalamos o HR FP dentro do instrumento SAM usando o CS100 do AAO. Como exemplo, temos, nas figuras 3.26 e 5.1, exemplos de dados obtidos na noite de 17 de março de 2015.

Além disso, com o FP, mostramos, na seção 3.2.3, que, para cubos de dados obtidos através de curtas exposições, o instrumento apresentou-se com uma eficiência operacional pouco abaixo do desejado. Entretanto, essa eficiência aumenta com o tempo de exposição obtido em cada um dos canais e o tempo gasto com a observação de um objeto está dentro do que é considerado normal para observações de outras galáxias com FP (1 a 2 horas).

Outro problema comum em observações usando FPs é a redução de dados. Em especial os processos de correção de fase e calibração em comprimento de onda. Ambos os problemas foram atacados e descritos com detalhes nas seções 4.6 e 4.7. Estes dois processos foram implementados em scripts facilmente usados por astrônomos.

O processo de calibração astrométrica dos dados obtidos com o BTFI ainda é feito de forma completamente artesanal. Entretanto, fornecemos uma tabela obtida com os dados existentes relacionando o ângulo de posição do telescópio com a orientação do campo. Para o futuro, esperamos escrever um script que faça uma calibração astrométrica inicial das imagens obtidas com o instrumento. Para isto, precisaremos modificar o código fonte do KAFKA de modo que ele consiga ler esta informação do telescópio e a armazene no

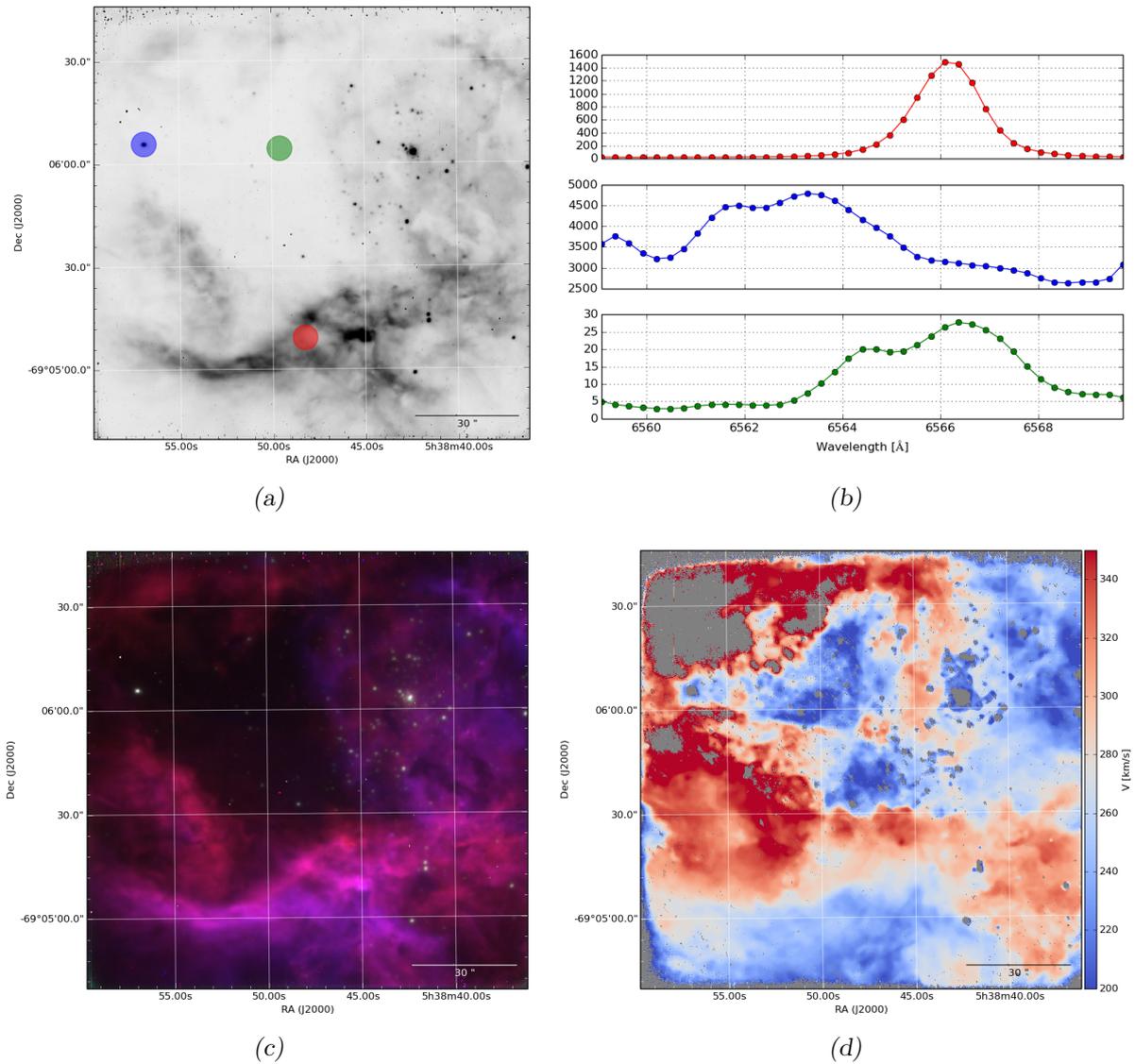


Figura 5.1: 30 Dor observado com o HR FP com o SAM usando filtro 6563/20 2x2" e o controlador CS100 do AAO. Nela, o leste aponta para a esquerda e o norte para baixo. (a) Cubo colapsado na direção espectral. (b) Espectro extraído de três diferentes regiões do cubo de dados. (c) Composição RGB do cubo de dados. Em verde, temos o contínuo. Em vermelho, a metade da linha de $H\alpha$ correspondente ao *redshift*. Em azul, a metade da linha de $H\alpha$ correspondente ao *blueshift*. (d) Mapa de velocidades calculado usando o baricentro da linha de $H\alpha$.

cabeçalho das imagens FITS para correlacioná-la, junto com as coordenadas de ascensão reta e declinação e com a amostragem de cada pixel, com os parâmetros armazenados no cabeçalho que fornecem a calibração astrométrica.

Deixamos pendente uma análise da deformação de campo e da variação da PSF ao longo do campo. Além disso, precisamos, ainda, fazer uma análise do deslocamento e deformação do campo causados por possíveis flexões na estrutura mecânica do instrumento. No seu estado atual, podemos dizer que faltam poucos problemas que precisam ser resolvidos antes que o BTFI possa ser entregue para a comunidade astronômica brasileira.

Os dados apresentados na seção 4.9 nos dão uma idéia da qualidade dos dados obtidos com o BTFI até o final de 2014, justificando a inexistência de algum artigo usando-os: estes dados não são publicáveis.

Um dos próximos passos que daremos será uma análise opto-mecânica sobre as rodas de filtros primária, secundária e sobre a *dummy-plate*. Isto pois temos indícios de que estes três componentes estão aparecendo nas imagens e limitando o campo útil do BTFI (capítulo 4, figura 4.9). O fato de estes três componentes estarem no caminho óptico, limitando o campo do BTFI, não está relacionado com o alinhamento óptico em si e, sim, com o alinhamento deles em relação ao eixo do sistema que está a desejar. Consertar este problema exige alterações mecânicas no instrumento e é necessária a intervenção de um engenheiro mecânico para realizar as possíveis soluções deste ponto.

Em segundo lugar, segundo as estimativas de corrente de escuro na seção 4.3, o isolamento contra luz do instrumento não é satisfatório. É necessário um estudo para buscar uma alternativa visando modificar a cobertura atual ou fabricar uma nova cobertura que não permita que nenhuma contaminação luminosa entre no instrumento.

O último fator que limita o uso do BTFI está relacionado com o comportamento das câmeras, ponto que não foi objeto de estudo desta tese. Testes liderados por Denis Andrade (IAG/USP) com colaboração de Peter Moore e Marco Bonati (CTIO/SOAR) indicam que as câmeras não possuem vácuo estável, ou seja, ao longo dos dias, o nível da pressão interna da câmera aumenta. Isso pode indicar vazamento ou contaminação interna. Essa instabilidade do vácuo apresenta-se como problema pois, de alguma forma, está influenciando no ganho EM dos detectores. Um trabalho de caracterização e investigação do problema está sendo realizado atualmente por Denis Andrade (IAG/USP) em colaboração com o Prof. Dani Guzman (UC/Chile).

Tendo solucionado estes pontos, acreditamos que o BTFI esteja pronto para ser com o Fabry-Perot usado pela comunidade astronômica brasileira em observações. Entretanto, no modo de baixa resolução, com o iBTF, a entrega do instrumento para a comunidade depende ainda da qualidade das imagens e dos espectros que serão obtidos com as novas redes holográficas encomendadas com a empresa Wasatch.

Referências Bibliográficas

- Afanasiev V. L., Moiseev A. V., The SCORPIO universal focal reducer of the 6-m telescope, *Astronomy Letters*, 2005, vol. 31, p. 194
- Andrade D., Daigle O., Quint B., Taylor K., Mendes de Oliveira C., Ramirez-Fernandez J., EMCCD camera noise performance for the Brazilian tunable filter imager, 2012, vol. 8453, p. 84531U
- Andrade D. F. D. E., Sistema Embarcado para Aquisição de Imagens Astronômicas, Universidade de São Paulo, 2011, Dissertação de mestrado
- Atherton P. D., Taylor K., Pike C. D., Harmer C. F. W., Parker N. W., Hook R. N., TAURUS: a wide-field imaging Fabry-Perot spectrometer for astronomy, *Monthly Notices of the Royal Astronomical Society*, 1982, vol. 201, p. 661
- Barden S. C., Arns J. A., Colburn W. S., Volume-phase holographic gratings and their potential for astronomical applications, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 1998, vol. 3355, p. 866
- Bass M., Stryland E. V., Williams D., Wolfe W., *Handbook of Optics*, volume I 2nd edn. vol. 1, McGraw-Hill, Inc., 1995
- Bell E. F., de Jong R. S., Stellar mass-to-light ratios and the Tully-Fisher relation, 2000, vol. 20, p. 17
- Blais-Ouellette S., Wishnow E. H., Shopbell P. L., van Breugel W., Taylor K., Smith R., Double Bragg Grating Tunable Filter, *Proceedings of SPIE*, 2004, vol. 5492, p. 779

- Bland J., Tully R. B., The Hawaii imaging Fabry-Perot interferometer (HIFI), *The Astronomical Journal*, 1989, vol. 98, p. 723
- Bonnarel F., Fernique P., Bienayme O., Egret D., Genova F., Louys M., Ochsenbein F., Wenger M., Bartlett J. G., The ALADIN Interactive Sky Atlas, *Astronomy and Astrophysics Supplement Series*, 2000, vol. 143, p. 8
- Born M., Wolf E., *Principles of Optics Electromagnetic Theory of Propagation, Interference and Diffraction of Light*, 1980
- Boulesteix J., Georgelin Y., Marcelin M., Monnet G., First results from CIGALE scanning Perot-Fabry interferometer, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 1984, vol. 445, p. 37
- Boulesteix J., Grundseth B., Observations tri-dimensionnelles par interférométrie Pérot-Fabry au TCFH, *CFHT Information Bulletin*, 1987, pp 2–3
- Bruzual G., Charlot S., Stellar population synthesis at the resolution of 2003, *Monthly Notices of the Royal Astronomical Society*, 2003, vol. 344, p. 1000
- Carignan C., Hernandez O., Beckman J. E., Fathi K., GHaFaS: Galaxy H alpha Fabry-Perot Spectrometer for the WHT, *Pathways Through an Eclectic Universe ASP Conference Series*, 2008, vol. 390
- Castilho B. V., Delabre B., Gneiding C. D., A new concept for echelle spectrographs: the SOAR Telescope Echelle Spectrograph, *Proc. SPIE*, 2004, vol. 5492, p. 433
- Cepa J., Aguiar M., Escalera V. G., Gonzalez-Serrano I., Joven-Alvarez E., Peraza L., Rasilla J. L., Rodriguez-Ramos L. F., Gonzalez J. J., Cobos Duenas F. J., Sanchez B., Tejada C., Bland-Hawthorn J., Militello C., Rosa F., OSIRIS tunable imager and spectrograph, *Proc. SPIE Vol. 4008*, 2000, vol. 4008, p. 623
- Clenet Y., Arsenault R., Beuzit J.-L., Chalabaev A., Delage C., Joncas G., Lacombe F., Lai O., Le Coarer E., Le Mignant D., Pau S., Rabou P., Rouan D., GriF: an infrared 3D spectroscopic mode for KIR/PUEO, *Proc. SPIE Vol. 4007*, 2000, vol. 4007, p. 942
- Crampton D., Grindmann W. A., Leckie B., Morbey C. L., Lemonnier J. P., Felenbok P., Marteau M., Vola P., Georgelin Y., Le Fèvre O., Grundseth B., Monnet G., Salmon

- D., Multi-Aperture and Subarcsecond Imaging Spectrograph for CFHT, *Progress in Telescope and Instrumentation Technologies*, 1992
- Daigle O., Carignan C., Gach J., Extreme faint flux imaging with an EMCCD, *Publications of the ...*, 2009, vol. 121, p. 866
- Djazovski O., Daigle O., Laurin D., Bedirian M., Ducharme M.-E., Artigau E., Doyon R., Electron-multiplying CCDs for future space instruments. In *Photonics North 2013*, International Society for Optics and Photonics, 2013, p. 89150Q
- Garrido O., Marcelin M., Amram P., Boulesteix J., GHASP: An H α kinematic survey of spiral and irregular galaxies, *Astronomy and Astrophysics*, 2002, vol. 387, p. 821
- Hernandez O., Gach J.-L., Carignan C., Boulesteix J., FaNTOMM: Fabry Perot of New Technology for the Observatoire du mont Megantic. In *Astronomical Telescopes and Instrumentation*, International Society for Optics and Photonics, 2003, p. 1472
- Ingraham P., Dooyon R., Beaulieu M., Rowlands N., Scott A., Speckle differential imagery performance using a JWST tunable filter etalon prototype. In *Space Telescopes and Instrumentation 2010: Optical*, vol. 7731, 2010, p. 77313R
- Kogelnik H., Coupled wave theory for thick hologram gratings, *Bell System Technical Journal*, 1969, vol. 48, p. 2909
- Lepine J. R. D., de Oliveira A. C., Figueredo M. V., Castilho B. V., Gneiding C., Mendes de Oliveira C., Barbuy B., Jones D. J., Kanaan A., Strauss C., Rodrigues F., Andrade C. R., de Oliveira L. S., de Oliveira J. B., SIFUS: SOAR integral field unit spectrograph, *Astronomical ...*, 2003, vol. 4841, p. 1086
- Löfdahl M. G., Henriques V. M. J., Kiselman D., Löfdahl, M. G. Henriques, V. M. J. Kiselman, D. A tilted interference filter in a converging beam, *A&A*, 2011, vol. 533, p. A82
- Mendes de Oliveira C., Taylor K., Quint B., Andrade D., Ferrari F., Others The Brazilian Tunable Filter Imager for the SOAR Telescope, *Publications of the Astronomical Society of the Pacific*, 2013, vol. 125, p. 396
- Mulligan J., Who were Fabry and Perot?, *American Journal of Physics*, 1998, pp 797–802

- Myers H. P., *Introductory Solid State Physics* 2 edn. UK Taylor & Francis Taylor & Francis Inc., 2009
- Ott T., *QFitsView: FITS file viewer*, Astrophysics Source Code Library, 2012
- Palmer C., Loewen E., *Diffraction grating handbook*, 2005
- Quint B. C., *Modelagem numérica de alto nível de filtros do tipo Fabry-Pérot e de redes de difração holográficas para o instrumento BTFI*, Universidade de São Paulo, 2010, Dissertação
- Quint B. C., Taylor K., Ferrari F., Mendes de Oliveira C., Muramatsu M., *Illusion - A Fabry-Perot Data-Cube Synthesizer*, *Astronomical Data Analysis Software and Systems XIX. Proceedings of a conference held October 4-8, 2010*, vol. 434
- Rangwala N., Williams T. B., Pietraszewski C., Joseph C. L., *AN IMAGING FABRY-PÉROT SYSTEM FOR THE ROBERT STOBIE SPECTROGRAPH ON THE SOUTHERN AFRICAN LARGE TELESCOPE*, *The Astronomical Journal*, 2008, vol. 135, p. 1825
- Ribeiro F., *Caracterização de Elementos Dispersivos*, UNIVERSIDADE FEDERAL DE ITAJUBÁ, 2010, Dissertação de Mestrado, 80
- Rosado M., Langarica R., Bernal A., Cobos F., Garfias F., Gutierrez L., Tejada C., Tinoco S., Le Coarer E., *The UNAM Scanning Fabry-Perot Interferometer (puma) for the Study of the Interstellar Medium*, *Revista Mexicana de Astronomia y Astrofisica Serie de Conferencias*, 1995, vol. 3
- Schwarz H. E., Ashe M. C., Boccas M., Bonati M., Delgado F., Gavez R., Martinez M., Schurter P., Schmidt R., Tighe R., Walker A. R., *The SOAR optical imager: status and first results*, vol. 5492, 2004, p. 564
- Taylor K. I., Atherton P. D., *Seeing-limited radial velocity field mapping of extended emission line sources using a new imaging Fabry-Perot system*, *Monthly Notices of the Royal Astronomical Society*, 1980, vol. 191, p. 675
- Tody D., *The Iraf Data Reduction And Analysis System*. In *1986 Astronomy Conferences*, International Society for Optics and Photonics, 1986, p. 733

-
- Tokovinin A., Tighe R., Schurter P., Cantarutti R., van der Blik N., Martinez M., Mondaca E., Montane A., SAM: a facility GLAO system for SOAR. In *SPIE Astronomical Telescopes + Instrumentation*, International Society for Optics and Photonics, 2008, p. 70154C
- Tulloch S. M., Dhillon V. S., , 2010 On the use of electron-multiplying CCDs for astronomical spectroscopy
- Veilleux S., Weiner B. J., Rupke D. S. N., McDonald M., Birk C., Bland-Hawthorn J., Dressler A., Hare T., Osip D., Pietraszewski C., Vogel S. N., MMTF: THE MARYLAND-MAGELLAN TUNABLE FILTER, *The Astronomical Journal*, 2010, vol. 139, p. 145
- Wells D. C., Greisen E. W., Harten R. H., FITS - a Flexible Image Transport System, *Astronomy and Astrophysics Supplement*, 1981, vol. 44

Apêndice

Apêndice A

Ligando e estabilizando o controlador CS100

Ambos os Fabry-Perots utilizados no instrumento BTFI são controlados através do *Queensgate Fabry-Perot Controller CS100*. Este controlador é o dispositivo eletrônico responsável por manter as placas de vidro do Fabry-Perot paralelas entre si e por regular a distância entre elas. Tanto os FPs quanto o controlador usados atualmente no BTFI são dispositivos bastante antigos, da década de 1980. Os atuadores que movem as placas de vidro são dispositivos piezo-eletrônicos e a distância entre as placas é monitorada em tempo real por discos metálicos que atuam como capacitores. Dispositivos decodificadores lêem a tensão armazenada nos capacitores para determinar o tamanho do espaço entre as placas.

Numa situação ideal, os operadores de telescópio devem ser os responsáveis por ligar o CS100 e estabelecer a situação ideal para que os FP possam ser controlados via software. Este procedimento foi extraído originalmente do manual de instruções do controlador CS100 e reescrito na página do telescópio CFHT¹ (*Canada-France-Hawaii Telescope*), na parte de documentação do instrumento MOS/OSIS, e é descrito novamente aqui.

A figura A.1 mostra o painel frontal do controlador CS100. Existem nove botões na parte esquerda do painel do CS100 responsáveis pelos ajustes, da esquerda para a direita, COARSE, FINE e RESISTIVE COMPONENT. De cima para baixo, estes nove botões são responsáveis pelos ajustes nos eixos X, Y e Z. Estes ajustes são configurados conforme o Fabry-Perot a ser utilizado.

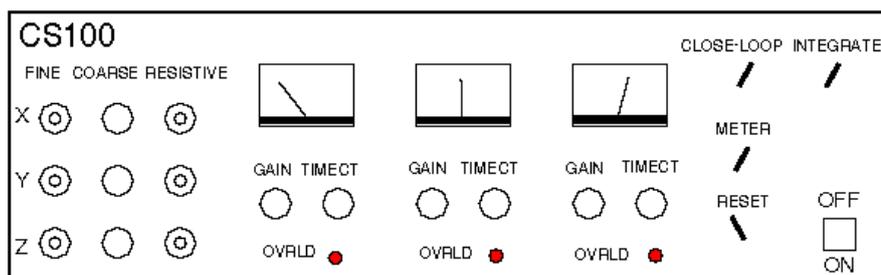


Figura A.1: Desenho esquemático do painel frontal do controlador CS100 extraído da página do telescópio CFHT, em outubro de 2014.

Antes mesmo do CS100 ser ligado, estes nove botões devem estar na configuração que corresponde à cada FP (figura A). O ganho GAIN e a constante temporal (TIME CONSTANT) devem ser ajustados para $G = 64$ e para $T = 25$. Em seguida, o CS100 pode ser ligado. Caso o LED que indica que o CLOSE LOOP estiver desligado, o interruptor

¹ <http://www.cfht.hawaii.edu/Instruments/Spectroscopy/Fabry-Perot/mosaicfp5.html>

correspondente deve ser ligado. Caso o contrário, o interruptor de RESET deve ser acionado com um movimento para baixo. Estes interruptores e LED podem ser encontrados na região à direita do painel do CS100 e estão devidamente identificados.

Tabela A.1 - Configuração a ser ajustada no CS100 ao usar o FP R20000

<i>Direction</i>	<i>Fine</i>	<i>Coarse</i>	R-Balance
X	19,22	0,0	2,90
Y	9,25	0,0	5,40
Z	2,70	-1,0	1,94

Caso a configuração inicial dos botões COARSE/FINE/RESISTIVE for desconhecida, o operador deverá seguir os seguintes passos:

1. Com o loop aberto, os valores do ganho e da constante temporal devem ser ajustados para um valor baixo: $G = 8$ e $T = 1.6ms$.
2. Neste momento, os interruptores de CLOSE LOOP e INTEGRATE devem estar desligados (para baixo), e o interruptor METER DISPLAY deve estar na posição ERROR SIGNAL.
3. Neste momento o CS100 pode ser ligado.
4. As agulhas dos displays X, Y e Z devem ser zerados utilizando os botões de controle X, Y e Z COARSE e FINE. Existe uma certa correlação entre os três canais então são necessárias algumas interações entre eles. Aqui, não é necessário que os displays estejam exatamente em 0. Mesmo com um ganho bastante baixo, os medidores de capacitância são bastante sensíveis. A escala completa corresponde a +/- 2 nm de movimento das placas com um ganho de 8x. Possivelmente, a agulha do display também ficará vibrando devido a ruídos eletrônicos. Vale lembrar que girar os botões COARSE/FINE no sentido horário fará com que as agulhas se desloquem da esquerda para a direita.
5. Aqui, o interruptor METER DISPLAY deve ser mudado para RESISTIVE COMPONENT e as leituras nos displays X, Y e Z devem ser zerados usando os botões R BALANCE à esquerda do painel. Isto pode ser feito com uma precisão um pouco maior mas vale lembrar que, aqui, girar os botões R BALANCE no sentido horário faz com que as agulhas se desloquem da direita para a esquerda.

6. O interruptor METER DISPLAY deve voltar, então, para a posição ERROR SIGNAL e os passos 4 e 5 devem ser repetidos até que o medidor esteja na posição 0 quando o METER DISPLAY estiver em RESISTIVE COMPONENT e dentro de escala quando ele estiver na posição ERROR SIGNAL.
7. Quando esta condição for alcançada, o ganho deve ser ajustado para 32x e a constante temporal deve ser ajustada para 250 ms. Logo em seguida, o interruptor CLOSE LOOP deve ser ligado (para cima). Após aproximadamente um segundo, o LED CLOSE LOOP deve acender indicando a operação correta de loop fechado do sistema de controle.

Cada FP possui um ganho e uma constante temporal ótima. Seja com a configuração sugerida aqui ou com a configuração obtida com os passos descritos acima, o FP está pronto para ser alinhado e, finalmente, operado em uma noite de observação.

Apêndice B

Código fonte: mkcube3

```
#!/usr/bin/python
# -*- coding: utf8 -*-
"""
    mkCube 3.0.0

    This file is part of the Butterfly Software System. It is used to organize
    images obtained through a BTFI scan and use them to create a data-cube.

    by Bruno Quint <bquint at astro.iag.usp.br>
"""

from __future__ import division, print_function

try:
    import astropy.io.fits as pyfits
except ImportError:
    import pyfits
    from sys import exit

    print("\n\tmkcube3 --")
    print("\tAstroPy module was not found in the system.")
    print("\tIt is required to run this program.")
    print("\tPlease, take a look at http://www.astropy.org/ for information")
    print("\tabout installing it on your system.")
    print("\n\tLeaving now.\n")
    exit()

def error(message):
    """
    Prints an error message in the terminal and leaves the program.
    :param message: The message that will be given for the user.
    :return: None
    """
    from sys import exit

    message = '\033[1;38 [!] %s \033[1;m' % message
    print(message)
    exit()

def warning(message):
    """
    Print a colour warning message in the screen.
    """
    message = ' \033[1;33m%s\033[1;m' % message
    print(message)

class MakeCube(object):
    def __init__(self):
        """
        Class constructor. It is used to initialize all the variables used in

```

```

the program silently.
"""
self.depth = None
self.height = None
self.input_files = None
self.input_table = None
self.mode = None
self.output_file = None
self.verbose = False
self.width = None
self.header = None
self.cube = None

@property
def create_input_table(self):
    """
    :return: None
    """
    from numpy import array
    from sys import stdout

    table = []
    header_field = None
    if self.verbose:
        print(" Reading files' headers to organize the data.")
    if self.mode.lower() in ['fp', 'fabry-perot']:
        header_field = 'FPZCURR'
    elif self.mode.lower() in ['ibt']:
        header_field = 'IBCURANG'
    else:
        error(" I am sorry but I could not find the BTFI mode" +
              " in the header.")

    i, number_of_files = 0.0, len(self.input_files)
    header_value = None
    for input_file in self.input_files:
        if self.verbose:
            i += 1.0
            stdout.write("\r %.2f%%" % (i * 100. / number_of_files))
            stdout.flush()
        try:
            header_value = pyfits.getheader(input_file)[header_field]
        except (IOError, KeyError):
            pass

        table.append([input_file, float(header_value)])

    print("")
    table = array(table, dtype=str)
    files = table[:, 0]
    poses = array(table[:, 1], dtype=float)
    indexes = poses.argsort()
    files = files[indexes]
    poses = poses[indexes]
    table = array([files, poses], dtype=str).transpose()

    return table

def get_cube_dimensions(self):
    """
    Determines what are the data-cube's dimensions.
    :return: None
    """
    self.width = self.get_width
    self.height = self.get_height
    self.depth = self.get_depth
    if self.verbose:
        print(" The data-cube will have the following dimensions:")
        print(" [%d, %d, %d]" % (self.width, self.height, self.depth))

```

```

@property
def get_depth(self):
    """
    :return: Returns data-cube depth.
    """
    from numpy import array, float32, unique
    pos = self.input_table[:, 1]
    pos = array(unique(pos), dtype=float32)
    return pos.size

@property
def get_height(self):
    """
    :return: the data-cube width based on the FITS header.
    """
    height = int(pyfits.getheader(self.input_files[0])['NAXIS2'])
    return height

@property
def get_instrument_mode(self):
    """
    Read the instrument mode from the header of a random input file.
    """
    filename = self.input_files[0]
    if self.verbose:
        print(" Reading instrument mode from file:\n %s" % filename)
    try:
        instrument_mode = pyfits.getheader(filename)['INSTRMOD']
    except KeyError:
        warning("Could not find instrument mode!")
        instrument_mode = 'unknown'
    if self.verbose:
        print(" Instrument mode: %s" % instrument_mode)
    return instrument_mode

def get_output_filename(self, overwrite=None, verbose=False):
    """
    This is a generic method used to check if a file called 'name' already
    exists. If so, it starts some interaction with the user.

    @keyword overwrite: if False, this method will interact with the user to
    ask if 'name' file shall be overwritten or if a new name will be given. If
    True, 'name' file is automatically overwritten.

    @keyword verbose: force verbose mode on even when overwrite is automatic.

    v1.0.1 - added 'overwrite' keyword.
            - added 'verbose' keyword.
    """
    import os
    import sys

    name = self.output_file
    v = False if (overwrite is True) else True
    if self.verbose:
        print("\n Writing to output file %s" % name)

    while os.path.exists(name):
        if overwrite in ['y', 'Y', True]:
            if v or verbose:
                print(" Deleting '%s' file now." % name)
                os.remove(name)

        elif overwrite in ['', 'n', 'N', False]:
            name = raw_input(" Please, enter a new filename:\n > ")

        elif overwrite in ['q']:
            if v:

```

```

        print(" Exiting program.")
        sys.exit()

    else:
        overwrite = \
            raw_input(" '%s' file exist. Overwrite? (y/[n])" % name)
        if v:
            print(" Writing data-cube to %s" % name)

    return name

@property
def get_width(self):
    """
    :return: the data-cube width based on the FITS header.
    """
    width = int(pyfits.getheader(self.input_files[0])['NAXIS1'])
    return width

@property
def make_cube(self):
    from numpy import array, empty, unique
    from sys import stdout

    cube = None
    table = self.input_table
    files = table[:, 0]
    poses = array(table[:, 1], dtype=float)
    unique_poses = unique(poses)

    if self.verbose:
        print(" Creating data-cube...")

    if self.verbose:
        print(" Allocating memory...")
    try:
        cube = empty((self.depth, self.height, self.width), dtype=float)
    except MemoryError:
        error(" Ops! The cube was too big for your computer.\n"
            " Try making a smaller one.\n"
            " Leaving now.")
    if self.verbose:
        print(" Ok.")
        print(" Filling data-cube:")

    i, number_of_positions = 0, unique_poses.size
    for pos in unique_poses:
        temp_files = files[poses == pos]
        frame = self.make_frame(temp_files)
        cube[i] = frame
        if self.verbose:
            i += 1
            stdout.write("\r %3.2f%%" % (i * 100 / number_of_positions))
            stdout.flush()
    print("\n Done.")
    return cube

def make_frame(self, files):
    from numpy import empty

    dummy_cube = empty((len(files), self.height, self.width))
    for i in range(len(files)):
        try:
            dummy_cube[i] = pyfits.getdata(files[i])
        except IOError:
            warning(" File %s may be corrupted." % files[i])

    return dummy_cube.mean(axis=0)

```

```

@property
def make_header(self):
    """
    Creates the cube's header.
    :return: header
    """
    header = pyfits.getheader(self.input_files[0])
    header[''] = "--- mkCube3 Calibration ---"

    if self.mode.lower() in ['fp', 'fabry-perot']:
        header['DISPAXIS'] = 3
        header['CRPIX3'] = 1
        header['CRVAL3'] = float(header['FPZINIT'])
        header['CDELTA3'] = float(header['FPZDELTA'])
        header['CTYPE3'] = 'LINEAR'
        header['CUNIT3'] = 'BNV'
        header['C3_3'] = float(header['FPZDELTA'])

    elif header['INSTRMOD'].lower() in ['ibt']:
        header['DISPAXIS'] = 3
        header['CRPIX3'] = 1
        header['CRVAL3'] = float(header['TFAINIT'])
        header['CDELTA3'] = float(header['TFADDELTA'])
        header['CTYPE3'] = 'LINEAR'
        header['CUNIT3'] = 'degrees'
        header['C3_3'] = float(header['TFADDELTA'])

    else:
        warning("Invalid BTFI Instrument Mode.")
        warning("Dummy calibration will be added to the data-cube.")
        header['DISPAXIS'] = 3
        header['CRPIX3'] = 1
        header['CRVAL3'] = 1
        header['CDELTA3'] = 1
        header['CTYPE3'] = 'LINEAR'
        header['CUNIT3'] = 'channel'
        header['C3_3'] = 1

    return header

def parse_arguments(self):
    """
    Parses the arguments given by the user and stores the information in
    the class properties.
    :rtype : none
    """
    import argparse
    import glob

    parser = argparse.ArgumentParser(
        description="Build a data-cube from image files.")

    parser.add_argument('-o', '--output', metavar='output', type=str,
                        default="cube.fits", help="Name of the output cube.")

    parser.add_argument('-q', '--quiet', action='store_true',
                        help="Run quietly.")

    parser.add_argument('files', metavar='files', type=str, nargs='+',
                        help="input filenames.")

    args = parser.parse_args()

    self.input_files = args.files
    if len(self.input_files) == 1:
        self.input_files = glob.glob(self.input_files[0])
    self.output_file = args.output
    self.verbose = not args.quiet

```

```
def print_header(self):
    """
    Print this program's header.
    :rtype : None
    """
    if not self.verbose:
        pass
    else:
        print(__doc__)
    return None

def run(self):
    """
    The main thread.
    """
    self.parse_arguments()
    self.print_header()
    self.mode = self.get_instrument_mode
    self.output_file = self.get_output_filename()
    self.input_table = self.create_input_table
    self.get_cube_dimensions()
    self.header = self.make_header
    self.cube = self.make_cube
    self.save_datacube()
    if self.verbose:
        print(" All done!\n")

def save_datacube(self):
    """
    Just save the data-cube in the disk.
    """
    if self.verbose:
        print(" Saving data-cube on the disk...")
    pyfits.writeto(self.output_file, self.cube, self.header, output_verify="fix")
    if self.verbose:
        print(" Done!")
    del self.cube
    return

if __name__ == '__main__':
    main = MakeCube()
    main.run()
```

Apêndice C

Código fonte: phmextractor.py

```
#!/usr/bin/python
# -*- coding: utf8 -*-
"""
    Phase-map Xtractor
    by Bruno C Quint, Fabrício Ferrari

    vla - Phase extraction for Fabry-Perot.
    2014.04.16 15:45 - Created an exception for errors while trying to access
                    'CRPIX%' cards on cube's header.
"""
from __future__ import division, print_function

import argparse
import astropy.io.fits as pyfits
import matplotlib.pyplot as pyplot
import numpy
import time
import scipy
import scipy.interpolate as interpolate
import scipy.ndimage as ndimage
import sys

def main():

    # Parse arguments -----
    parser = argparse.ArgumentParser(description="Extracts the phase-map" +
                                         "from a fits file containing a data" +
                                         "-cube.")

    parser.add_argument('-c', '--correlation', action='store_true',
                        help="Use correlation cube? true/[FALSE]")

    parser.add_argument('filename', type=str, help="Input data-cube name.")

    parser.add_argument('-o', '--output', type=str, default=None,
                        help="Name of the output phase-map file.")

    parser.add_argument('-q', '--quiet', action='store_true',
                        help="Run program quietly. true/[FALSE]")

    parser.add_argument('-s', '--show', action='store_true',
                        help="Show plots used in the process. true/[FALSE]")

    args = parser.parse_args()

    # Starting program -----
    v = not args.quiet
    start = time.time()

    if v:
        print("")
        print(" Phase-Map Extractor")
        print(" by Bruno Quint & Fabricio Ferrari")
```

```

    print(" version 0.1c - May 2014")
    print(" Extracting phase-map from file: %s" % args.filename)

# Checking input data -----
if v:
    print(" Checking data-cube for phase-correction.")
    check_dimensions(args.filename)
    check_instrument(args.filename)

# Selecting BTFI mode and extracting phase-map -----
mode = check_mode(args.filename)
if mode == 'ibtf':
    PhaseMap_ibTF(args.filename, correlation=args.correlation,
                  show=args.show, verbose=v)
elif mode == 'fp':
    PhaseMap_FP(args.filename, correlation=args.correlation,
                show=args.show, verbose=v)

# All done! -----
end = time.time() - start
if v:
    print("\n Total time ellapsed: %02d:%02d:%02d" %
          (end // 3600, end % 3600 // 60, end % 60))
    print(" All done!\n")

def check_dimensions(filename, dimensions=3, keyword='NAXIS'):
    """
    Method written to check the dimensions of the input fits data.
    """
    header = pyfits.getheader(filename)

    if not 'NAXIS' in header:
        data = pyfits.getdata(args.filename)
        ndim = data.ndim
    else:
        ndim = header['NAXIS']

    if ndim is not dimensions:
        print(" INVALID DATA: wrong number of dimensions")
        print(" Leaving now!")
        print("")
        sys.exit()
    else:
        return

def check_instrument(filename, instrument='btfi', keyword='INSTRUME'):
    """
    Method written to check the instrument.
    """
    header = pyfits.getheader(filename)

    # First check if the keyword exists
    if not keyword in header:
        print("")
        print(" Instrument type not recognized.")
        print(" Do you want to proceed? [Y or n]")

        answer = '.'
        while answer.lower() not in ' yn':
            answer = raw_input('? ')

        if answer.lower() == 'n':
            print(" Leaving now.\n")
            sys.exit()
        else:
            return

    # Then check if it is the right instrument
    if header[keyword].lower() is not instrument:
        print(" Wrong instrument. Proceed? [Y or n]")

        answer = '.'

```

```

    while answer.lower() not in ' yn':
        answer = raw_input('? ')

    if answer.lower() == 'n':
        print(" Leaving now.\n")
        sys.exit()

    return

def check_mode(filename, keyword='INSTRMOD'):
    """
    Return if BTFI was obtained with a Fabry-Perot or with the iBTF.
    """
    header = pyfits.getheader(filename)

    if keyword not in header:
        print("\n Instrument mode not found.")
        instrument_mode = ''
        while instrument_mode.lower() not in ['ibtff', 'fp']:
            instrument_mode = raw_input("? Enter 'ibtff' or 'fp': ")
    else:
        if header[keyword].upper() in ['IBTF']:
            instrument_mode = 'ibtff'

        if header[keyword].upper() in ['FP', 'FABRY-PEROT']:
            instrument_mode = 'fp'

    return instrument_mode

def safe_save(name, extension=None, overwrite=False, verbose=False):
    """
    This is a generic method used to check if a file called 'name' already
    exists. If so, it starts some interaction with the user.

    @param name: the name of the file that will be written in the future.

    @keyword extension: check if the file has the following extension. If not,
    it fills for the user. Defaults is None. An example would be
    extension='.fits'.

    @keyword overwrite: if False, this method will interact with the user to
    ask if 'name' file shall be overwritten or if a new name will be given. If
    True, 'name' file is automatically overwritten.

    @keyword verbose: force verbose mode on even when overwrite is automatic.

    v1.0.2 - added 'extension' keyword.
    v1.0.1 - added 'overwrite' keyword.
    - added 'verbose' keyword.
    """
    import os

    if os.path.splitext(name)[1] != extension and extension is not None:
        name = name + extension

    v = False if (overwrite is True) else True
    if v: print('\n Writing to output file "%s"' % name)

    while os.path.exists(name):

        if overwrite in ['y', 'Y', True]:
            if v or verbose:
                print(" Overwriting %s file." % name)
            os.remove(name)

        elif overwrite in ['', 'n', 'N', False]:
            name = raw_input(" Please, enter a new filename:\n > ")
            if os.path.splitext(name)[1] != extension and extension is not None:
                name = name + extension

        elif overwrite in ['q']:
            if v:

```

```

        print(" Exiting program.")
        sys.exit()

    else:
        overwrite = raw_input(" '%s' file exist. Overwrite? (y/[n])"%name)
        if v:
            print(" Writing data-cube to %s" %name)

    return name

#=====
class PhaseMap:

    def __init__(self, filename, **config):

        # Setting main configuration -----
        self.input_file = filename
        self.config = config
        self.verbose = config['verbose']

        if 'show' in config:
            self.show = config['show']
        else:
            self.show = False

        self.loading = [' ', '- ', '\\ ', '| ', '/ ']

        # Reading raw data -----
        self.print(" Loading data.")
        self.data = pyfits.getdata(filename)
        self.header = pyfits.getheader(filename)
        self.print(" Done.")

        # Reading data-cube configuration -----
        self.width = self.header['NAXIS1']
        self.height = self.header['NAXIS2']
        self.depth = self.header['NAXIS3']

        # Reading Z calibration for plotting -----
        self.z = self.get_calibration()

        try:
            self.units = self.header['CUNIT3']
        except KeyError:
            self.units = "channels"

        try:
            self.sample = self.header['C3_3']
        except KeyError:
            self.sample = 1.0

        return

    def extract_phase_map(self):
        """
        Extract the phase-map.
        """
        from astropy.io.fits import getdata
        from numpy import argmax, inf, where

        self.print("\n Starting phase-map extraction.")
        self.print(" Reading data from %s file" % self.extract_from)
        data = getdata(self.extract_from)
        #data = where(data > data.mean() + data.std(), data, -inf)

        phase_map = argmax(data, axis=0) * self.sample
        return phase_map

    def find_reference_pixel(self):
        """Read the reference pixel from header or find it."""

```

```

if ('PHMREFX' in self.header) and ('PHMREFY' in self.header):
    self.print("\n Found reference pixel in header.")
    ref_x = self.header['PHMREFX']
    ref_y = self.header['PHMREFY']
    self.print(" Using [%d, %d]" % (self.ref_x, self.ref_y))
else:
    self.print("\n Reference pixel NOT in header.")

    answer = '.'
    while answer.lower() not in ' yn':
        answer = raw_input("? Use central pixel? [Y, n]\n? ")

    if answer.lower() in ' y':
        ref_x = self.width // 2
        ref_y = self.height // 2
    else:
        ref_x = int(raw_input("? Please, enter reference X: "))
        ref_y = int(raw_input("? Please, enter reference Y: "))

    self.print(" Using [%d, %d]" % (ref_x, ref_y))

return ref_x, ref_y

def get_calibration(self):
    """
    Return an array with the current calibration.
    """
    z = numpy.arange(self.depth)
    try:
        # The "+ 1" change from fortran like to c like indexing
        z = z - self.header['CRPIX3'] + 1
        z = z * self.header['C3_3']
        z = z + self.header['CRVAL3']

    except KeyError:
        print("! Calibration in third axis not found.")
        print("! I will ignore this step.")

    return z

def get_fwhm(self):
    """
    Returns the full-width-at-half-maximum.
    """

    # TODO Add an option to choose wheter to use gauss FWHM or measured FWHM

    from scipy.optimize import leastsq
    from scipy.stats import mode

    fsr = self.free_spectral_range / self.header['C3_3']
    z = self.z[:fsr]
    s = self.ref_s[:fsr]
    s = s - mode(s)[0]

    zz = numpy.linspace(z[0], z[-1], 1000)
    ss = interpolate.interpld(z, s, kind='cubic')
    sss = ss(zz) - ss(zz).max() / 2

    fit_func = lambda p, x: p[0] * numpy.exp(-(x - p[1]) ** 2 / (2 * p[2] ** 2))
    err_func = lambda p, x, y: y - fit_func(p, x)
    p, _ = leastsq(err_func, p, args=(zz, sss))
    fwhm_gauss = 2.35482 * p[2]

    zzz = zz[sss > 0]
    fwhm_measured = zzz.ptp()

    if self.show:
        pyplot.figure()
        pyplot.title("Measure the FWHM")

```

```

    pyplot.plot(z, s, 'bo')
    pyplot.plot(z, ss(z), 'b-', lw=2)
    pyplot.plot(z, sss, 'r-', lw=2, alpha=0.3)
    pyplot.plot(z, fit_func(p, z), 'g-', lw=2, alpha=0.3)
    pyplot.axvline(p[1] - fwhm_gauss / 2, ls='--', c='green', lw=2)
    pyplot.axvline(p[1] + fwhm_gauss / 2, ls='--', c='green', lw=2,
        label='Gauss Fit = %.1f %s' % (fwhm_gauss, self.units))
    pyplot.axvline(p[1] + fwhm_measured / 2, ls='--', c='red', lw=2)
    pyplot.axvline(p[1] - fwhm_measured / 2, ls='--', c='red', lw=2,
        label='Definition = %.1f %s' % (fwhm_measured, self.units))
    pyplot.legend(loc='best')
    pyplot.grid()
    pyplot.tight_layout()
    pyplot.show()

if self.verbose:
    print("")
    print(" Measured FWHM = %.2f %s" % (fwhm_measured, self.units))
    print(" Gauss-fit FWHM = %.2f %s " % (fwhm_gauss, self.units))
    print(" Using the measured FWHM for further calculations.")

return fwhm_measured

def get_reference_spectrum(self):
    """
    Get the reference spectrum.
    """
    from scipy.stats import mode

    ref_s = pyfits.getdata(self.input_file)[: , self.ref_y, self.ref_x]
    ref_s = ref_s / ref_s.max() # Normalize
    ref_s = ref_s - ref_s.mean() # Remove mean to avoid triangular shape
    ref_s = ref_s - mode(ref_s)[0] # Try to put zero on zero

    if self.show:
        pyplot.figure()
        pyplot.title("Reference Spectrum")
        pyplot.plot(self.z, ref_s, 'ko-', label="Reference spectrum")
        pyplot.grid()
        pyplot.xlabel("z [%s]" % self.units)
        pyplot.tight_layout()
        pyplot.show()

    return ref_s

def get_refx_pixel(self):
    """
    Return the position of the reference X in pixels.
    """

    return

def print(self, string):
    """
    Print only in verbose mode.
    """
    if self.verbose: print(string)
    return

def use_correlation(self):
    """
    Use correlation data-cube.
    """
    import numpy

    from astropy.io.fits import getdata, getheader, writeto
    from glob import glob
    from os.path import splitext
    from sys import stdout

    self.print("\n A correlation cube will be used.")
    self.print(" Looking for an existing correlation data-cube in the current folder.")

```

```

candidates = glob("*.fits")

corr_cube = None
for candidate in candidates:
    if 'CORRFROM' in getheader(candidate):
        if getheader(candidate)['CORRFROM'] == self.input_file:
            self.print(" Correlation cube to be used: %s" % candidate)
            return candidate

if corr_cube == None:
    self.print(" Correlation cube not found. Creating a new one.")
    data = getdata(self.input_file)
    corr_cube = numpy.empty_like(data)

    x = numpy.arange(self.width)
    y = numpy.arange(self.height)
    X, Y = numpy.meshgrid(x, y)
    x, y = numpy.ravel(X), numpy.ravel(Y)

    for i in range(x.size):
        s = data[:,y[i],x[i]]
        s = s / s.max() # Normalize
        s = s - s.mean() # Remove mean to avoid triangular shape
        s = numpy.correlate(s, self.ref_s, mode='same')
        corr_cube[:,y[i],x[i]] = s

        temp = ((i + 1) * 100.00 / X.size)
        stdout.write('\r %2d%% ' % temp)
        stdout.write(self.loading[int(temp * 10 % 5)])
        stdout.flush()

    self.print(" Done.")
    corr_name = splitext(self.input_file)[0] + '--corrcube.fits'
    self.print(" Saving correlation cube to %s" % corr_name)

    corr_hdr = self.header.copy()
    corr_hdr.set('CORRFROM', self.input_file, 'Cube used for corrcube.')
    corr_hdr.set('', '', before='CORRFROM')
    corr_hdr.set('', '--- Correlation cube ---', before='CORRFROM')

    writeto(corr_name, corr_cube, corr_hdr, clobber=True)
    del corr_hdr
    del corr_cube

    return corr_name

def save(self):
    """
    Save files.
    """
    from astropy.io.fits import getdata, writeto
    from os.path import splitext

    v = self.verbose
    f = splitext(self.input_file)[0]
    h = self.header.copy()
    h.set('PHMREFX', self.ref_x)
    h.set('PHMREFY', self.ref_y)
    h.set('', '', before='PHMREFX')
    h.set('', '--- PHM Xtractor ---', before='PHMREFX')

    h.set('PHMREFF', self.input_file, 'Original file')
    h.set('PHMTYPE', 'observed')
    h.set('PHMUNIT', self.units)
    h.set('PHMSAMP', self.sample)

    filename = safe_save(f + "--obs_phmap.fits", overwrite=True, verbose=v)
    self.print(" Saving observed phase-map to file: %s" % filename)
    writeto(filename, self.phase_map, h, clobber=True)

    ## TODO Fix refspect file's header to keep calibration
    filename = safe_save(f + "--ref_spec.fits", overwrite=True, verbose=v)

```

```

self.print(" Saving reference spectrum to file: %s" % filename)
writeto(filename, self.ref_s, h, clobber=True)

return

#=====
class PhaseMap_FP(PhaseMap):

def __init__(self, filename, correlation=False, show=False, verbose=False):

    PhaseMap.__init__(self, filename, correlation=correlation,
                      show=show, verbose=verbose)

    # This is a Fabry-Perot data-cube. Let's make that clear to the user
    if self.verbose:
        print("\n Extracting phase-map from a Fabry-Perot data-cube.")

    # Measure the free-spectral-range
    self.free_spectral_range = self.get_free_spectral_range()

    # Getting reference spectrum
    self.ref_x, self.ref_y = self.find_reference_pixel()
    self.ref_s = self.get_reference_spectrum()

    # Calculate the FWHM
    self.fwhm = self.get_fwhm()

    # Calculate the finesse
    self.finesse = self.get_finesse()

    if self.verbose:
        print(" Ideal number of channels: %.1f channels"
              % round(2 * self.finesse))
        print(" Ideal sampling: %.1f %s / channel"
              % (self.free_spectral_range / round(2 * self.finesse),
                self.units))

    if correlation:
        self.extract_from = self.use_correlation()
    else:
        self.extract_from = self.input_file

    self.phase_map = self.extract_phase_map()
    self.save()

    return

def extract_phase_map(self):
    """
    Extract the phase-map.
    """

    now = time.time()

    try:
        sampling = self.header['C3_3']
    except KeyError:
        sampling = 1

    fsr = round(self.free_spectral_range / sampling)

    # Reading data
    if self.verbose:
        print("\n Starting phase-map extraction.")
        print(" Reading data from %s file" % self.extract_from)
    data = pyfits.getdata(self.extract_from)
    # data = data[0:fsr]

    # Extracting phase-map
    if self.verbose:
        print(" Extracting phase-map...")
    data = numpy.where(data > data.mean() + data.std(), data, -numpy.inf)

```

```

phase_map = numpy.argmax(data, axis=0) * sampling

if self.verbose:
    print(" Done in %.2f seconds" % (time.time() - now))
return phase_map

def find_reference_pixel(self):
    """
    Read the reference pixel from header or find it.
    """
    if self.verbose:
        print("\n Finding reference pixel.")

    if ('PHMREFX' in self.header) and ('PHMREFY' in self.header):

        if self.verbose:
            print(" Found reference pixel found in header.")

        ref_x = self.header['PHMREFX']
        ref_y = self.header['PHMREFY']

        if self.verbose:
            print(" Using [%d, %d]" % (ref_x, ref_y))

    else:
        if self.verbose:
            print(" Reference pixel NOT found in header.")
            print(" Trying to find the center of the rings.")
        ref_x, ref_y = self.find_rings_center()

    return ref_x, ref_y

def find_rings_center(self):
    """
    Method used to find the center of the rings inside a FP data-cube.
    """

    now = time.time()

    # Renaming some variables
    width = self.width
    height = self.height
    fsr = round(self.free_spectral_range / self.header['C3_3'])

    # Choosing the points
    x = (numpy.linspace(0.2, 0.8, 500) * width).astype(int)
    y = (numpy.linspace(0.2, 0.8, 500) * height).astype(int)

    ref_x = self.header['NAXIS1'] // 2
    ref_y = self.header['NAXIS2'] // 2

    self.print(" Start center finding.")
    old_ref_x = ref_x
    old_ref_y = ref_y

    if self.show:
        pyplot.figure()

    for i in range(6):

        ref_y = max(ref_y, 0)
        ref_y = min(ref_y, self.header['NAXIS2'])

        ref_x = max(ref_x, 0)
        ref_x = min(ref_x, self.header['NAXIS2'])

        temp_x = self.data[:fsr, ref_y, x]
        temp_y = self.data[:fsr, y, ref_x]

        temp_x = numpy.argmax(temp_x, axis=0)
        temp_y = numpy.argmax(temp_y, axis=0)

```

```

px = scipy.polyfit(x, temp_x, 2)
py = scipy.polyfit(y, temp_y, 2)

ref_x = round(- px[1] / (2.0 * px[0]))
ref_y = round(- py[1] / (2.0 * py[0]))

if self.show:
    pyplot.title("Finding center of the rings")
    pyplot.cla()
    pyplot.plot(x, temp_x, 'b.', alpha=0.25)
    pyplot.plot(x, scipy.polyval(px, x), 'b-', lw=2)
    pyplot.plot(y, temp_y, 'r.', alpha=0.25)
    pyplot.plot(y, scipy.polyval(py, y), 'r-', lw=2)
    pyplot.gca().yaxis.set_ticklabels([])
    pyplot.axvline(ref_x, ls='--', c='blue', label='x')
    pyplot.axvline(ref_y, ls='--', c='red', label='y')
    pyplot.legend(loc='best')
    pyplot.grid()
    pyplot.ylabel("Iteration number %d" % (i+1))

# Selecting valid data
error_x = numpy.abs(temp_x - scipy.polyval(px, x))
error_y = numpy.abs(temp_y - scipy.polyval(py, y))

cond_x = numpy.where(error_x <= 3 * error_x.std(), True, False)
cond_y = numpy.where(error_y <= 3 * error_y.std(), True, False)

x = x[cond_x]
y = y[cond_y]

# Choosing when to stop
if (abs(old_ref_x - ref_x) <= 2) and (abs(old_ref_y - ref_y) <= 2):

    try:
        # If the cube was binned this will be useful
        ref_x = (ref_x - self.header['CRPIX1'] + 1) \
            * self.header['CDELTA1'] + self.header['CRVAL1']

        # If the cube was binned this will be useful
        ref_y = (ref_y - self.header['CRPIX2']) \
            * self.header['CDELTA2'] + self.header['CRVAL2']
    except KeyError:
        pass

    if self.verbose:
        print("Rings center found at: [%d, %d]" % (ref_x, ref_y))
        print("Done in %.2f s" % (time.time() - now))

    if self.show:
        pyplot.tight_layout()
        pyplot.show()

    return ref_x, ref_y

else:
    old_ref_x = ref_x
    old_ref_y = ref_y

if self.show:
    pyplot.tight_layout()
    pyplot.show()

if self.verbose:
    print("Rings center NOT found.")

ref_x = self.header['NAXIS1'] // 2
ref_y = self.header['NAXIS2'] // 2

# If the cube was binned this will be useful
try:
    ref_x = (ref_x - self.header['CRPIX1']) \
        * self.header['CDELTA1'] + self.header['CRVAL1']

```

```

        ref_y = (ref_y - self.header['CRPIX2']) \
            * self.header['CDELTA2'] + self.header['CRVAL2']
    except:
        pass

    if self.verbose:
        print(" Done in %.2f s" % (time.time() - now))
        print(" Using [%d, %d]." % (ref_x, ref_y))
    return ref_x, ref_y

def get_finesse(self):
    """
    Assuming you have the Free-Spectral-Range in Z unit and that
    you have the FWHM in Z units as well, calculate the finesse.
    """

    finesse = self.free_spectral_range / self.fwhm

    if self.verbose:
        print(" Finesse = %.1f" % finesse)

    return finesse

def get_free_spectral_range(self):
    """
    A quick-and-dirty way to measure the free range in FP units.
    The method subtracts each frame of the data-cube from the
    first one. Then, it calculates the absolute value and collapse
    in X and Y. The FSR is where the resulting spectrum is minimum,
    excluding (of course), the first one.
    """

    if self.verbose:
        print(" Finding the free-spectral-range.")

    now = time.time()

    # First frame is the reference frame
    ref_frame = self.data[0,:,:]

    # Subtract all frames from the first frame
    data = self.data - ref_frame

    # Get the absolute value
    data = numpy.abs(data)

    # Sum over the spatial directions
    data = data.sum(axis=2)
    data = data.sum(axis=1)

    # Interpolate data
    s = interpolate.UnivariateSpline(self.z, data, k=3)
    z = numpy.linspace(self.z[5:].min(), self.z.max(), 1000)

    # Find the free-spectral-range in z units
    fsr = z[numpy.argmin(s(z))] - self.z[0]

    # Find the free-spectral-range in number of channels
    fsr_channel = numpy.argmin(numpy.abs(self.z - z[numpy.argmin(s(z))]))

    # Calculate the sampling
    sampling = fsr / fsr_channel

    if self.verbose:
        print(" FSR = %.1f %s" % (fsr, self.units))
        print("      = %d channels" % fsr_channel)
        print(" Sampling = %.1f %s / channel" % (sampling, self.units))
        print(" Done in %.2f s" % (time.time() - now))

    # Plot to see how it goes
    if self.show:

```

```

    pyplot.title("Finding the Free-Spectral-Range")
    pyplot.plot(self.z, data, 'bo', label='Measured data')
    pyplot.plot(z, s(z), 'r-', lw=2, label='3rd deg spline fitting')
    pyplot.xlabel("z [%s]" % self.units)
    pyplot.axvline(x=(fsr + self.z.min()), ls='--', c='red',
                   label='Free-Spectral-Range \nat z = %.1f' % fsr)
    pyplot.legend(loc='best')
    pyplot.gca().yaxis.set_ticklabels([])
    pyplot.grid()
    pyplot.tight_layout()
    pyplot.show()

    return fsr

def save(self):
    """
    Save files.
    """
    from os.path import splitext

    v = self.verbose
    f = splitext(self.input_file)[0]
    h = self.header.copy()
    h['PHMREFX'] = self.ref_x
    h['PHMREFY'] = self.ref_y
    h.set('', '', before='PHMREFX')
    h.set('', '--- PHM Xtractor ---', before='PHMREFX')

    # if 'PHMREFX' not in self.header:
    #     update = '.'
    #     while update.upper() not in 'YESNO':
    #         update = raw_input(" Update input file? [Y]/n \n ")
    #     if update.upper() in 'YES':
    #         self.print(" Updating input file %s" % self.input_file)
    #         data = getdata(self.input_file)
    #         writeto(self.input_file, data, h, clobber=True)

    fsr = self.free_spectral_range

    h['PHMREFF'] = (self.input_file, 'Original file')
    h['PHMTYPE'] = 'observed'
    h['PHMUNIT'] = self.header['CUNIT3']
    h['PHMFSR'] = (round(fsr, 2),
                  'Free-spectral-range in %s units' % self.units)
    h['PHMSAMP'] = (self.header['C3_3'], 'Used sample [%s / channel].',
                   % self.units)

    # TODO Remove 3rd axis calibration residuals
    filename = safe_save(f + "--obs_phmap.fits", overwrite=True, verbose=v)
    self.print(" Saving observed phase-map to file: %s" % filename)
    pyfits.writeto(filename, self.phase_map, h, clobber=True)

    filename = safe_save(f + "--ref_spec.fits", overwrite=True, verbose=v)
    self.print(" Saving reference spectrum to file: %s" % filename)
    pyfits.writeto(filename, self.ref_s, h, clobber=True)

    return

#####
class PhaseMap_iBTF(PhaseMap):

    def __init__(self, filename, correlation=False, show=False, verbose=False):

        PhaseMap.__init__(self, filename, correlation=correlation,
                          show=show, verbose=verbose)

        # This is an iBTF data-cube. Let's make that clear to the user
        if self.verbose:
            print("\n Extracting phase-map from a iBTF data-cube.")

        # Getting reference spectrum
        self.ref_x, self.ref_y = self.find_reference_pixel()

```

```
self.ref_s = self.get_reference_spectrum()

# Use correlation?
if correlation:
    self.extract_from = self.use_correlation()
else:
    self.extract_from = self.input_file

# Extract phase-map
self.phase_map = self.extract_phase_map()
self.save()

#=====
if __name__ == '__main__':
    main()
```


Apêndice D

Código fonte: phmfit.py

```
#!/usr/bin/python
#-*- codign: utf8 -*-
"""
    2014.04.16 15:51 - Fixed keyword to access phase-map sampling.
    2014.08.25 18:30 - Fixed ref_x/ref_y
                    - Show phmap, ref_x/ref_y, npoints
"""
from __future__ import division, print_function

def main():

    import argparse
    import matplotlib.pyplot as plt
    import numpy
    import os
    import scipy.optimize as optimize
    from astropy.io import fits as pyfits

    # Parsing arguments -----
    parser = argparse.ArgumentParser(description="Fits an existing phase-map.")

    parser.add_argument('filename',
                        type=str,
                        help="Input phase-map name.")

    parser.add_argument('-i', '--interactions', default=5, type=int,
                        help="Number of interactions in the process [5]")

    parser.add_argument('-n', '--npoints', default=2500, type=int,
                        help="Number of points that will be used to fit" +
                        "the phase-map [50]")

    parser.add_argument('-o', '--output', type=str, default=None,
                        help="Name of the output phase-map file.")

    parser.add_argument('-q', '--quiet', action='store_true',
                        help="Run program quietly.")

    parser.add_argument('-s', '--show_plots', action='store_true',
                        help="Show plots (good for checking quality of the" + \
                        "observed phase-map and the fitting.)")

    args = parser.parse_args()
    v = not args.quiet
    if v:
        print("\n Phase-Map Fitting for BTFI")
        print(" by Bruno Quint & Fabricio Ferrari")
        print(" version 0.0a - Jan 2014")
        print("")

    check_dimensions(args.filename, dimensions=2)

    # Loading observed map -----
```

```

if v:
    print(" Loading file: %s" % args.filename)
phase_map = pyfits.open(args.filename)[0]

# Check if file was obtained with BTFI instrument
header = phase_map.header
try:
    if header['INSTRUME'].upper() not in ['BTFI'] and v:
        if v:
            print(" [Warning]: %s file was not obtained"+\
                "with BTFI instrument." % args.filename)
except KeyError:
    warning("'INSTRUME' card was not found in the files' header.")

mode = check_mode(args.filename)

# Fitting Phase-Map for IBTF -----
if mode == 'ibtf':
    if v:
        print(" File obtained through an iBTF scan.")

    width = phase_map.header['naxis1']
    height = phase_map.header['naxis2']
    vmin = phase_map.data.mean() - 1.5 * phase_map.data.std()
    vmax = phase_map.data.mean() + 1.5 * phase_map.data.std()
    plt_config = {'origin': 'lower',
                  'cmap': get_colormap(),
                  'interpolation': 'nearest',
                  'vmin': vmin, 'vmax': vmax}

    if v:
        print(" Phase-map dimensions: [%d, %d]" % (width, height))
        print(" Done.")

    plt.subplot(131)

    plt.imshow(phase_map.data, **plt_config)
    plt.xticks([], plt.yticks([]))
    plt.xlabel('Observed Map')

# Starting fitting process -----
npoints = numpy.sqrt(args.npoints).astype(int)
if v:
    print("\n Starting phase-map fitting.")
    print(" %d x %d points will be used in the process." %
          (npoints, npoints))

x = (numpy.linspace(0.1, 0.9, npoints) * width).astype(int)
y = (numpy.linspace(0.1, 0.9, npoints) * height).astype(int)
x, y = numpy.meshgrid(x, y)
x = numpy.ravel(x)
y = numpy.ravel(y)
z = numpy.ravel(phase_map.data[y,x])

fit_func = lambda p, x, y: p[0] + p[1] * x + p[2] * y
err_func = lambda p, x, y, z: z - fit_func(p, x, y)
params = [z.mean(), 0, 0]

# Fitting Plane -----
X = numpy.arange(phase_map.header['naxis1'])
Y = numpy.arange(phase_map.header['naxis2'])
X, Y = numpy.meshgrid(X, Y)

if v: print("")
for i in range(args.interactions):
    if v:
        print(" Fitting plane - Interaction %d" % (i + 1))

    if i == 0: e = z
    condition = numpy.where(numpy.abs(e - e.mean()) <= e.std())
    xx = x[condition]
    yy = y[condition]
    zz = z[condition]

```

```

    params, _ = optimize.leastsq(err_func, params, args=(xx, yy, zz))

    Z = fit_func(params, X, Y)
    error = Z - phase_map.data
    e = numpy.ravel(error[y,x])

    if v:
        p = params
        print(" phi(x,y) = %.2f + %.2fx + %.2fy" % (p[0], p[1], p[2]))
        print(" Error abs min: %f" % numpy.abs(e).min())
        print(" Error avg: %f" % e.mean())
        print(" Error std: %f" % e.std())
        print(" Error rms: %f" % numpy.sqrt((e ** 2).mean()))

    plt.scatter(xx, yy, c=zz, cmap=get_colormap())
    plt.xlim(0, width), plt.ylim(0, height)

    plt.subplot(132)
    plt.imshow(error, **plt_config)
    plt.xticks([], plt.yticks([]))
    plt.xlabel("Residual")

    plt.subplot(133)
    plt.imshow(Z, **plt_config)
    plt.xticks([], plt.yticks([]))
    plt.xlabel("Fitted map")

    plt.show()

    ref_x = header['PHMREFX']
    ref_y = header['PHMREFY']
    fname = header['PHMREF']

    fname = os.path.splitext(fname)[0]
    pyfits.writeto(fname + '--fit_phmap.fits',
                  Z - Z[ref_y, ref_x], header, clobber=True)
    pyfits.writeto(fname + '--res_phmap.fits',
                  Z - phase_map.data, header, clobber=True)

    print("")

# Fitting phase-map for a Fabry-Perot Map -----
elif mode == 'fp':
    npoints = numpy.sqrt(args.npoints).astype(int)

    if v:
        print(" File obtained through a Fabry-Perot scan.")
        print(" Starting phase-map fitting.")
        print(" %d x %d points will be used in the process." %
              (npoints, npoints))

    # Read data
    width = header['NAXIS1']
    height = header['NAXIS2']
    ref_x = header['PHMREFX']
    ref_y = header['PHMREFY']
    unit = header['PHMUNIT']
    sampling = header['PHMSAMP']
    FSR = header['PHMFSR']
    phmap = phase_map.data

    # From coordinates to pixels
    ref_x = header['CRPIX1'] + (ref_x - header['CRVAL1']) / header['CDELTA1']
    ref_y = header['CRPIX2'] + (ref_y - header['CRVAL2']) / header['CDELTA2']

    phmap = phmap - phmap[ref_y, ref_x]

    x = (numpy.linspace(0.05, 0.95, npoints) * width).astype(int)
    y = (numpy.linspace(0.05, 0.95, npoints) * height).astype(int)
    X, Y = numpy.meshgrid(x, y)

```

```

R = numpy.sqrt((X - ref_x) ** 2 + (Y - ref_y) ** 2)
Z = phmap[Y, X]

if args.show_plots:
    phmap_figure = plt.figure()
    phmap_axes = phmap_figure.add_subplot(111)
    phmap_imshow = phmap_axes.imshow(phmap, origin='lower',
                                     interpolation='nearest',
                                     cmap='coolwarm')
    phmap_axes.scatter(ref_x, ref_y, c='orange', s=400, marker="+", lw=3)
    phmap_axes.scatter(X, Y, c='g', s=1, marker=".", alpha=0.7)
    phmap_axes.set_xlabel("X [px]")
    phmap_axes.set_ylabel("Y [px]")
    phmap_axes.set_xlim(0, width)
    phmap_axes.set_ylim(0, height)
    phmap_axes.grid()
    phmap_figure.colorbar(phmap_imshow)

x = numpy.ravel(X)
y = numpy.ravel(Y)
r = numpy.sqrt((x - ref_x) ** 2 + (y - ref_y) ** 2)
z = numpy.ravel(Z)

condition = numpy.where(z > z.min(), True, False) * \
            numpy.where(z < z.max(), True, False)

r = r[condition]
z = z[condition]

z = z[numpy.argsort(r)]
r = numpy.sort(r)

# Checking if parabola is up or down.
if v:
    print("\n Checking if parabola is up or down.")

dz = numpy.diff(z,1)
dz_abs = numpy.abs(dz)
dz_sign = numpy.sign(dz)
sign = numpy.median(dz_sign[(dz_sign != 0) * (dz_abs <= sampling)])

if v:
    print(" Parabola is %s" % ('up' if sign > 0 else 'down'))

# Tell me the limits to fit the first parabola
where = numpy.argmin(numpy.abs(r[dz_abs >= FSR / 2][0] - r))

# Plot the gradient
if args.show_plots:
    plt.figure(figsize=(16,7))
    plt.subplot(2,2,3)
    plt.plot(r[1:], dz, 'b-')
    plt.gca().yaxis.set_label_position("right")
    plt.axvline(r[where], color='black', lw=2, ls='--')
    plt.axhline(FSR / 2, color='red', ls='--', label="FSR")
    plt.axhline(- FSR / 2, color='red', ls='--')
    plt.xlabel('Radius [px]')
    plt.ylabel('Gradient \n [%s]' % unit)
    plt.legend(loc='best')
    plt.grid()

# This is the first fit
p = numpy.polyfit(r[:where], z[:where], 2)
rr = numpy.linspace(r[0], r[where], 1000)
zz = numpy.polyval(p, rr)

# Plot the data before correction
if args.show_plots:
    plt.subplot(2,2,1)
    plt.plot(r[:where], z[:where], 'b.', alpha=0.25,
             label='Not to be fixed')

```

```

plt.plot(r[where:], z[where:], 'r.', alpha=0.25,
        label='Data to be fixed')
plt.plot(rr, zz, 'k-', lw=2)
plt.axvline(r[where], color='black', lw=2, ls='--')
plt.gca().yaxis.set_label_position("right")
plt.xlabel('Radius [px]')
plt.ylabel('Peak displacement \n [%s]' % unit)
plt.legend(loc='best')
plt.grid()

# Displace the FSR
error = numpy.abs(z - numpy.polyval(p, r) + sign * FSR)

# Plot error
if args.show_plots:
    plt.subplot(2,2,4)
    plt.plot(r, error, 'k.', alpha=0.25)
    # plt.gca().yaxis.tick_right()
    plt.gca().yaxis.set_label_position("right")
    plt.xlabel('Radius [px]')
    plt.ylabel('Error \n [%s]' % unit)
    plt.ylim(ymin=-50, ymax=1.1*error.max())
    plt.grid()

condition = (error > 2 * sampling)

# Plot data after correction
if args.show_plots:
    plt.subplot(2,2,2)
    plt.plot(r[condition], z[condition], 'b.', alpha=0.25,
            label='Not fixed data')
    plt.plot(r[~condition], z[~condition] + sign * FSR, 'r.',
            alpha=0.25, label='Fixed data')
    plt.gca().yaxis.set_label_position("right")
    plt.xlabel('Radius [px]')
    plt.ylabel('Peak displacement \n [%s]' % unit)
    plt.grid()

# This is the second fit
z = numpy.where(error >= 2 * sampling, z, z + sign * FSR)
p = numpy.polyfit(r, z, 2)

if args.show_plots:
    rr = numpy.linspace(r[0], r[-1], 1000)
    zz = numpy.polyval(p, rr)
    plt.plot(rr, zz, 'k-', lw=2, label='Fitted data.')
    plt.legend(loc='best')

error = z - numpy.polyval(p, r)

if v:
    print(" phi(x,y) = %.2e x^2 + %.2e x + %.2e " % (p[0], p[1], p[2]))
    print(" Error abs min: %f" % numpy.abs(error).min())
    print(" Error avg: %f" % error.mean())
    print(" Error std: %f" % error.std())
    print(" Error rms: %f" % numpy.sqrt((error** 2).mean()))
    print(" Sampling in Z: %s" % phase_map.header['phmsamp'])
    print(" ")

x = numpy.arange(width)
y = numpy.arange(height)
X, Y = numpy.meshgrid(x, y)
R = numpy.sqrt((X - ref_x) ** 2 + (Y - ref_y) ** 2)
Z = numpy.polyval(p, R)
Z = Z - Z[ref_y, ref_x]

fname = header['PHMREFF']
fname = os.path.splitext(fname)[0]
pyfits.writeto(fname + '--fit_phmap.fits', Z, header, clobber=True)
pyfits.writeto(fname + '--res_phmap.fits',
               Z - phmap, header, clobber=True)

```

```

    if v:
        print(" All done.\n")

    if args.show_plots:
        plt.show()

    else:
        if v: print(" [Warning]: File was not obtained from FP or iBTF.")
        if v: print(" [Warning]: Don't know what to do. Leaving now.\n")
        from sys import exit
        exit()

def check_dimensions(filename, dimensions=3, keyword='NAXIS'):
    """
    Method written to check the dimensions of the input fits data.
    """
    from astropy.io import fits as pyfits
    import sys

    header = pyfits.getheader(filename)

    if not 'NAXIS' in header:
        data = pyfits.getdata(args.filename)
        ndim = data.ndim
    else:
        ndim = header['NAXIS']

    if ndim is not dimensions:
        error(" INVALID DATA: wrong number of dimensions")
        error(" Leaving now!")
        print("")
        sys.exit()
    else:
        return

def check_mode(filename, keyword='INSTRMOD'):
    """
    Return if BTFI was obtained with a Fabry-Perot or with the iBTF.
    """
    from astropy.io import fits as pyfits

    header = pyfits.getheader(filename)

    if keyword not in header:
        warning("Instrument mode not found.")
        instrument_mode = ''
        while instrument_mode.lower() not in ['ibtf', 'fp']:
            instrument_mode = raw_input(" Enter 'ibtf' or 'fp': ")
    else:
        if header[keyword].upper() in ['IBTF']:
            instrument_mode = 'ibtf'

        if header[keyword].upper() in ['FP', 'FABRY-PEROT']:
            instrument_mode = 'fp'

    return instrument_mode

def get_colormap():
    from matplotlib import colors

    cdict = {'red': ((0.0, 1.0, 1.0),
                    (0.25, 1.0, 1.0),
                    (0.5, 0.0, 0.0),
                    (0.75, 0.0, 0.0),
                    (1.0, 0.9, 0.9)),
            'green': ((0.0, 0.9, 0.9),
                     (0.25, 0.0, 0.0),
                     (0.5, 0.0, 0.0),
                     (0.75, 0.0, 0.0),
                     (1.0, 0.9, 0.9)),
            'blue': ((0.0, 0.9, 0.9),
                    (0.25, 0.0, 0.0),
                    (0.5, 0.0, 0.0),
                    (0.75, 0.0, 0.0),
                    (1.0, 0.9, 0.9))}

```

```
        (0.25, 0.0, 0.0),
        (0.5, 0.0, 0.0),
        (0.75, 1.0, 1.0),
        (1.0, 1.0, 1.0))}

    return colors.LinearSegmentedColormap('heaven_hell', cdict, 256)

def error(my_string):
    s = bcolors.FAIL + '[ERROR] ' + bcolors.ENDC
    s = s + str(my_string)
    print(s)
    return

def warning(my_string):
    s = bcolors.WARNING + '[!] ' + bcolors.ENDC
    s = s + str(my_string)
    print(s)
    return

class bcolors:

    HEADER = '\033[95m'
    OKBLUE = '\033[94m'
    OKGREEN = '\033[92m'
    WARNING = '\033[93m'
    FAIL = '\033[91m'
    ENDC = '\033[0m'

    def disable(self):
        self.HEADER = ''
        self.OKBLUE = ''
        self.OKGREEN = ''
        self.WARNING = ''
        self.FAIL = ''
        self.ENDC = ''

if __name__ == '__main__':
    main()
```


Apêndice E

Código fonte: phmapply.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
"""
    Phase-Map Apply - A script to apply a phase-map on a data-cube.
    by Bruno Quint (bquint@astro.iag.usp.br)
    and Fabricio Ferrari (fabricio@ferrari.pro.br)
    version 0.0 - Feb 2014
"""

from __future__ import division, print_function

import argparse
import astropy.io.fits as pyfits
import numpy
import os
import sys
import time

from scipy.interpolate import UnivariateSpline

def main():

    # Setting Options -----
    parser = argparse.ArgumentParser(description="Apply a phase-map on" + \
        "a data-cube.")

    parser.add_argument('-o', '--output', metavar='output', type=str,
        default=None, help="Name of the output corrected cube")

    parser.add_argument('-q', '--quiet', action='store_true',
        help="Run it quietly.")

    parser.add_argument('-n', '--npoints', type=int, default=10,
        help="Number of points in the re-sampling for channel [10].")

    parser.add_argument('cube_file', metavar='cube_file', type=str,
        help="Input calibration cube filename.")

    parser.add_argument('map_file', metavar='map_file', type=str,
        help="Input phase-map image filename.")

    args = parser.parse_args()
    v = not args.quiet
    loading = [' ', '-', '\\', '|', '/']

    # Printing program header -----
    if v:
        start = time.time()
        print("\n Phase-Map Apply")
        print(" by Bruno Quint & Fabricio Ferrari")
        print(" version 0.0 - Feb 2014")

    root_dir = os.path.dirname(args.cube_file)
```

```

cube_file = args.cube_file
map_file = args.map_file

if args.output is None:
    out_file = 'phc_' + os.path.split(args.cube_file)[-1]
else:
    out_file = args.output

if v:
    print(" \n Root dir: %s" % root_dir)
    print(" Cube to be corrected: %s" % cube_file)
    print(" Phase-map to be applied: %s" % map_file)
    print(" Output corrected cube: %s" % out_file)

# Reading input data -----
if v:
    print("\n Reading cube to be corrected.")

data_cube = pyfits.open(cube_file)[0]

if v:
    print(" Done.")
    print("\n Reading phase-map to be applied.")

phase_map = pyfits.open(map_file)[0]

if v:
    print(" Done.")

# Checking data -----
if data_cube.data[0].shape != phase_map.shape:
    print("[!] Cube and map does not have matching width and height.")
    print("[!] Leaving now.\n")
    sys.exit()

if data_cube.data.ndim != 3:
    print("[!] Cube file is not really a cube.")
    print("[!] Leaving now.\n")
    sys.exit()

if phase_map.data.ndim != 2:
    print("[!] Map file is not really an image.")
    print("[!] Leaving now.\n")
    sys.exit()

check_instrument(cube_file)
mode = check_mode(cube_file)

## Phase-Correction for iBTF data-cube -----
if mode.lower() in ['ibtf']:

    # Padding data-cube with zeros -----
    if v:
        print(" Padding data-cube with zeros")

    phm_max = round(abs(phase_map.data).max()) + 1
    L, M, N = data_cube.data.shape
    pad = numpy.zeros((phm_max, M, N))

    if v:
        print(" Cube shape before padding: %d x %d x %d" % (N, M, L))
        print(" %d frames will be added." % (2 * phm_max))

    data_cube.data = numpy.vstack((pad, data_cube.data))
    data_cube.data = numpy.vstack((data_cube.data, pad))
    L, M, N = data_cube.data.shape

    try:
        data_cube.header['CRPIX3'] = data_cube.header['CRPIX3'] + phm_max
    except KeyError:
        data_cube.header['CRPIX3'] = L

```

```

if v:
    print(" Cube shape after padding: %d x %d x %d" % (N, M, L));
    print(" Done.");

# Applying phase-map -----
if v:
    print("\n Applying phasemap")
for i in range(M):
    for j in range(N):
        if v:
            temp = ((i + 1) * 100.00 / M)
            sys.stdout.write('\r %2d%% ' % temp)
            sys.stdout.write(loading[int(temp * 10 % 5)])
            sys.stdout.flush()
            spec = data_cube.data[:,i,j]
            shift = phase_map.data[i,j]
            data_cube.data[:,i,j] = shiftSpectrum(spec, shift, args.npoints)
if v: print(" Done.")

## Phase-Correction for Fabry-Perot data-cube -----
elif mode.lower() in ['fabry-perot', 'fp']:

M = data_cube.header['NAXIS1']
N = data_cube.header['NAXIS2']

ref_x = phase_map.header['PHMREFX']
ref_y = phase_map.header['PHMREFY']
units = phase_map.header['PHMUNIT']
sample = float(phase_map.header['PHMSAMP'])

# Reading the Free-Spectral-Range -----
try:
    if v:
        print(" Reading free-spectral-range from cube header.")
        # TODO add an option to use the FSR found while extracting
        # TODO the phase-map or while fitting it.
        # TODO or even to give the option for the user to enter it.
        # FSR = phase_map.header['PHMFITSR']
        FSR = phase_map.header['PHMFSR']
        if v:
            print(" Free Spectral Range = %.2f %s" % (FSR, units))
except (KeyError):
    print(" Please, enter the free-spectral-range in %s units" % units)
    FSR = input(" > ")

FSR = round(FSR / sample) # From BCV to Channels
if v:
    print(" Free-Spectral-Range is %d channels" % FSR)

fsr = FSR * args.npoints # From Channels to nPoints
fsr = int(round(fsr))
if v:
    print(" Free-Spectral-Range is %d points" % fsr)

# Assure that the reference spectrum will not be moved -----
try:
    phase_map.data = phase_map.data - phase_map.data[ref_y, ref_x]
except IndexError:
    print("[!] Reference pixel out of field.")
    print("[!] Skipping reference pixel map subtraction.")
    pass
phase_map.data = -1 * phase_map.data

# Converting phase-map values to channels -----
phase_map.data = phase_map.data / sample

# Converting phase-map from channels to number of points -----
phase_map.data = phase_map.data * args.npoints

# Applying phase-map -----
if v:

```

```

    print("\n Applying phase-map:")

    z = numpy.arange(data_cube.header['NAXIS3'])
    new_z = numpy.arange(0, z.size, 1.0 / args.npoints)

    for i in range(M):
        for j in range(N):

            # Extracting a spectrum
            spec = data_cube.data[:,j,i]
            dz = phase_map.data[j,i]

            # Re-sample spectrum
            spline = UnivariateSpline(z, spec, s=0.0)
            new_spec = spline(new_z)

            # Cutting spectrum
            new_z = new_z[0:fsr+1]
            new_spec = new_spec[0:fsr+1]

            # Shifting spectrum
            new_spec = numpy.roll(new_spec, int(dz))

            # Under-sampling spectrum
            spline = UnivariateSpline(new_z, new_spec, s=0.0)
            spec = spline(z)

            # Storing new spectrum
            data_cube.data[:,j,i] = spec

            # Giving a feedback to the user
            if v:
                temp = ((i + 1) * 100.00 / M)
                sys.stdout.write('\r %2.2f%% ' % temp)
                sys.stdout.flush()

            end_of_cube = min(int(round(FSR)), data_cube.data.shape[0])
            data_cube.data = data_cube.data[0:end_of_cube, :, :]
            if v: print(" Done.")

    else:
        sys.exit()

    # Saving corrected data-cube -----
    if v:
        print("\n Writing output to file %s." % out_file);
        data_cube.writeto(out_file, clobber=True)
    if v:
        print(" Done.");
        end = time.time() - start
        print("\n Total time ellapsed: %02d:%02d:%02d" % \
              (end // 3600, end % 3600 // 60, end % 60));
        print(" All done!\n");

def check_instrument(filename, instrument='btfi', keyword='INSTRUME'):
    """
    Method written to check the instrument.
    """
    header = pyfits.getheader(filename)

    # First check if the keyword exists
    if not keyword in header:
        print("")
        print(" Instrument type not recognized.")
        print(" Do you want to proceed? [Y or n]")

        answer = '.'
        while answer.lower() not in ' yn':
            answer = raw_input('? ')

        if answer.lower() == 'n':
            print(" Leaving now.\n")

```

```

        sys.exit()
    else:
        return

def check_mode(filename, keyword='INSTRMOD'):
    """
    Return if BTFI was obtained with a Fabry-Perot or with the IBTF.
    """
    from astropy.io import fits as pyfits

    header = pyfits.getheader(filename)

    if keyword not in header:
        warning("Instrument mode not found.")
        instrument_mode = ''
        while instrument_mode.lower() not in ['ibtbf', 'fp']:
            instrument_mode = raw_input(" Enter 'ibtbf' or 'fp': ")
    else:
        if header[keyword].upper() in ['IBTF']:
            instrument_mode = 'ibtbf'

        if header[keyword].upper() in ['FP', 'FABRY-PEROT']:
            instrument_mode = 'fp'

    return instrument_mode

## Method shiftSpectrum =====
def shiftSpectrum(spec, dz, nPoints=100):

    dzSign = -numpy.sign(dz)
    dz = abs(dz)
    dzPoints = int(dz * nPoints)
    if dzPoints is 0: return spec

    # Get the spectrum from cube
    z = numpy.arange(spec.size)
    spline = UnivariateSpline(z, spec, s=0.0)

    # Re-sample spectrum
    newZ = numpy.linspace(z[0], z[-1], z.size * nPoints)
    newSpec = spline(newZ)

    # Add padded borders
    newSpec = numpy.append(numpy.zeros(dzPoints), newSpec)
    newSpec = numpy.append(newSpec, numpy.zeros(dzPoints))

    # Shifting spectrum
    newSpec = numpy.roll(newSpec, int(dzSign * dzPoints))

    # Cutting Spectrum
    newSpec = newSpec[dzPoints:-dzPoints]

    # Down-sampling
    spline = UnivariateSpline(newZ, newSpec, s=0.0)
    spec = spline(z)

    return spec

## Method shift_spectrum =====
def shift_spectrum(spec, dz, fsr=-1, sample=1.0, n_points=100):
    """
    @param spec: 1D numpy.array representing a spectrum to be shifted.
    @param dz: how big is the shifting.
    @keyword fsr: a float representing the free-spectra-range in sample units.
    @keyword sample: a float representing the increment between each channel.
    @keyword n_points: number of points that will be used for super-sampling.
    """

    dzSign = -numpy.sign(dz)
    dz = abs(dz) / sample # From cube units to channels
    dzPoints = int(dz * n_points) # From channels to new sample units

```

```

index = fsr / sample # From cube units to channels
index = index * n_points # From channels to new sample units

if dzPoints is 0:
    return spec

# Get the spectrum from cube
z = numpy.arange(spec.size)
spline = UnivariateSpline(z, spec, s=0.0)

# Re-sample spectrum
newZ = numpy.linspace(z[0], z[-1], z.size * n_points)
newSpec = spline(newZ)

# Cutting Spectrum
newSpec = newSpec[0:fsr]

# Shifting spectrum
newSpec = numpy.roll(newSpec, int(dzSign * dzPoints))

# Down-sampling
spline = UnivariateSpline(newZ, newSpec, s=0.0)
spec = spline(z)

return spec

def error(my_string):
    s = bcolors.FAIL + '[ERROR] ' + bcolors.ENDC
    s = s + my_string
    print(s)
    return

def warning(my_string):
    s = bcolors.WARNING + '[WARNING] ' + bcolors.ENDC
    s = s + my_string
    print(s)
    return

class bcolors:

    HEADER = '\033[95m'
    OKBLUE = '\033[94m'
    OKGREEN = '\033[92m'
    WARNING = '\033[93m'
    FAIL = '\033[91m'
    ENDC = '\033[0m'

    def disable(self):
        self.HEADER = ''
        self.OKBLUE = ''
        self.OKGREEN = ''
        self.WARNING = ''
        self.FAIL = ''
        self.ENDC = ''

#=====
if __name__ == '__main__':
    main()

```