

1 露出型偏光撮像器HOWPolの 観測システムの開発と 偏光キャリブレーション

広島大学大学院 理学研究科 物理科学専攻
高エネルギー宇宙・可視赤外線天文学研究室

M094600 小松 智之

主査:川端 弘治 副査:杉立 徹

2011年 2月 10日

概要

ガンマ線バースト (GRB) は、天球上のある一点が一時的にガンマ線で明るく輝く現象である。GRB の中でも、初期宇宙やブラックホール誕生の手がかりとされる GRB の一部は、ガンマ線で光った後、X 線から電波に亘る幅広い波長帯で時間の冪乗で暗くなる残光を伴う。この残光の観測、特に偏光観測は GRB 放射の起源とされる相対論的ジェットの磁場構造を明らかにする上で重要であるが、残光は急速に暗くなるため、GRB 出現後速やかに観測できるかどうかは鍵となる。

そこで、我々は、高速駆動が可能なかなた望遠鏡と組み合わせて GRB の初期残光の偏光を得ることが出来る「1 露出型偏光撮像装置 (HOWPol)」を開発してきた。急速に減光し位置誤差が大きい GRB を捉える為に、HOWPol は広視野で且つ一回の露出で偏光データを得ることが出来るようになっている (図 1)。しかし、駆動制御システムの完成が遅れたため、当初は GRB 出現のアラートに即応した自動観測をすることが出来ない状況であった。また、ナスミス焦点の装置であるため器械偏光が大きく、キャリブレーション方法の確立が遅れていた。

そこで、私は GRB 自動観測を始めとした様々な観測モードの完成を目指して、HOWPol の制御システム全般の整備を行った。具体的には駆動系を制御する PC 上でサーバープロセスを立ち上げ、クライアント側 PC の命令に応じて駆動系を制御するソフトウェアを構築した。これによって、検出器を制御する PC から望遠鏡も含めたすべての制御を統括することが可能となり、GRB の自動観測が可能となった他、観測効率も大幅に向上した。また、様々な望遠鏡位置での標準星の試験観測を行い、得られたデータを用いて偏光のキャリブレーションを行った。ナスミス焦点の装置は、第 3 鏡による 90° 反射に伴い約 4% に達する器械偏光を伴う。GRB の観測では 0.5% 以下の偏光精度が要求されることから、この器械偏光の正確な補正が重要となる。ある無偏光標準星に対して観測から得られた直線偏光データ (Q, U) の HA (時角) 依存性に、第 3 鏡の反射のみを考慮した器械偏光モデル曲線をフィットした様子を図 1.3 に示す。また、器械偏光に加えて偏光方位角原点および器械消偏光についても調査した。これらを踏まえ、 $\Delta p \leq 0.4\%$ の精度で偏光測定を行うためのキャリブレーション方法を確立した。そして、GRB 091208B や MAXI J1659-152 といった天体に対する科学的な偏光データの取得にも成功した。

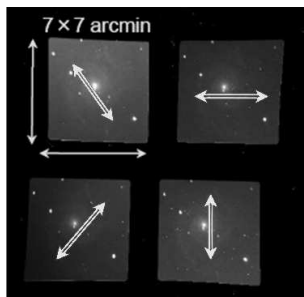


図 1: 広視野 1 露出モードでの取得画像

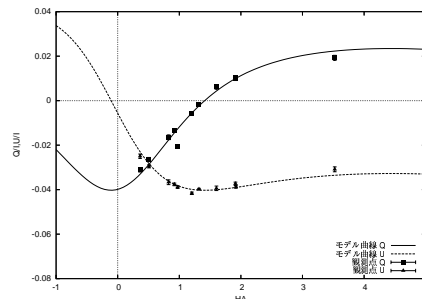


図 2: モデル曲線 (Q, U の HA 依存性)

目次

第1章 序章	7
1.1 東広島天文台	7
1.1.1 かなた望遠鏡	7
1.1.2 観測装置	9
1.1.3 ガンマ線バースト	11
1.1.4 ガンマ線バーストの偏光観測	12
1.1.5 本研究の目的	13
第2章 HOWPol	14
2.1 HOWPolの構成	14
2.1.1 筐体	14
2.1.2 光学系	15
2.1.3 検出器系	16
2.1.4 駆動制御系	17
2.2 HOWPolの観測モード	23
2.2.1 撮像観測モード	23
2.2.2 分光観測モード	25
2.2.3 偏光観測モード	26
2.2.4 GRB自動観測モード	28
第3章 HOWPolの偏光キャリブレーション	29
3.1 偏光観測とその目的	29
3.1.1 偏光とストークスパラメータ	29
3.1.2 偏光観測の目的	30
3.2 HOWPolによる偏光観測	31
3.2.1 ストークスパラメータの導出	31
3.3 偏光観測に特有なキャリブレーション	36
3.3.1 器械偏光	36
3.3.2 器械消偏光	37
3.3.3 方位角原点	37
3.4 器械偏光の測定	38

3.4.1	調査項目	38
3.4.2	観測	38
3.4.3	円によるフィット	39
3.4.4	ナスミス焦点における器械偏光の定式化	40
3.4.5	器械偏光の時角・赤緯依存性	43
3.4.6	最適アパーチャサイズの決定	46
3.5	方位角原点補正	48
3.5.1	観測	48
3.6	器械消偏光の測定	49
第4章	天体の観測	50
4.1	強偏光標準星	50
4.2	MAXI J1659-152	51
第5章	まとめ	54

目次

1	広視野1露出モードでの取得画像	1
2	モデル曲線(Q,UのHA依存性)	1
1.1	かなた望遠鏡	8
1.2	カセグレン焦点に設置されたTRISPEC	9
1.3	第一ナスミス焦点に設置されたHOWPol	10
1.4	第二ナスミス焦点に設置された高速分光器	10
1.5	GRBのモデル(上原 岳士 2008年度 修士論文)	11
1.6	X線の初期残光の振る舞い(上原 岳士 2008年度 修士論文)	12
2.1	HOWPolの断面図	14
2.2	光学系の概念図	15
2.3	デュワーに取り付けられたCCD	16
2.4	HOWPolで撮られたM16の三色合成画像	16
2.5	分光用マスク (スリット幅0.2mm = 2.3")	17
2.6	狭視野偏光用マスク	17
2.7	広視野偏光用マスク	17
2.8	格子状マスク	17
2.9	狭視野偏光撮像用ダブルウォラストンプリズム	18
2.10	広視野偏光撮像用ダブルウォラストンプリズム	18
2.11	ウォラストンプリズム	19
2.12	ダブルウォラストンプリズム	20
2.13	サーバープログラムの起動の様子	21
2.14	ストリームソケット通信の流れ	22
2.15	撮像モードで得られる生画像(M16、Vバンド)	24
2.16	分光モードで得られる生画像(3c454.3)	25
2.17	スカイの分光データ	25
2.18		26
2.19		26
2.20		27

2.21	27
2.22	半波長板の効果	27
3.1	Q-U 平面と偏光度、偏光方位角	30
3.2	偏光観測の概念図：半波長板使用	31
3.3	偏光観測の概念図：1 露出型	34
3.4	35
3.5	35
3.6	反射による偏光	37
3.7	ナスミス焦点	37
3.8	第3鏡による器械偏光が円を描く様子	39
3.9	円でフィットした様子	39
3.10	方位角方向	40
3.11	方位角方向の動き (a)	40
3.12	方位角方向の動き (b)	40
3.13	方位角方向の動き (c)	40
3.14	高度軸方向	41
3.15	高度軸方向の動き (a)	41
3.16	高度軸方向の動き (b)	41
3.17	高度軸方向の動き (c)	41
3.18	フィットした様子:	
	HD125184 R バンド chip0	43
3.19	フィットした様子:	
	HD125184 R バンド chip1	43
3.20	フィットした様子:	
	HD125184 R バンド oneshot	44
3.21	フィットした様子:	
	HD212311 R バンド chip1	44
3.22	フィットした様子:	
	HD212311 R バンド chip0	44
3.23	フィットした様子:	
	HD212311 R バンド oneshot	44
3.24	フィットした様子:	
	HD154892 R バンド chip1	44
3.25	フィットした様子:	
	HD154892 R バンド chip0	44
3.26	フィットした様子:	
	HD154892 R バンド oneshot	45

3.27	広視野偏光モードのアーチャーサイズの決定 (縦軸:総カウント数 横軸:アーチャーサイズ) HD157982 Rc バンド	47
3.28	広視野偏光モードのアーチャーサイズの決定 (縦軸:偏光度 横軸:アーチャーサイズ) HD157982 Rc バンド	47
4.1	MAXI J1659-152 のライトカーブ	52
4.2	V-R の変化	52
4.3	R-I の変化	52
4.4	V-I の変化	53
4.5	偏光:Q-U 平面	53
4.6	比較星の位置	53

表 目 次

1.1	かなた望遠鏡の仕様	8
2.1	光学系の仕様	15
2.2	分光用グリズムの仕様	19
2.3	wh1～wh7の割り当て	23
2.4	プリスキャン領域とオーバースキャン領域 (X座標、単位 pix)	25
3.1	試験観測	39
3.2	モデル曲線の各パラメータ	45
3.3	アパーチャーサイズの決定	47
3.4	強偏光標準星の試験観測	48
3.5	強偏光標準星のカタログ値	48
4.1	天体の観測	50

第1章 序章

1.1 東広島天文台

1.1.1 かなた望遠鏡

かなた望遠鏡は有効口径 1.5m の主鏡を持ち、可視光・近赤外線で観測することを目的とした光学望遠鏡である (図 1.1)。この望遠鏡は、もともと 1994 年に赤外シミュレータとして国立天文台の三鷹キャンパスに設置され、すばる望遠鏡に取り付ける観測装置や鏡の試験をするために使われてきた。そして 2004 年、その役割を終えたことを受けて天文研究への再利用を目的として広島大学が譲り受け、架台や制御系を更新させた上で、広島大学宇宙科学センター附属東広島天文台に移設した。それ以降、宇宙科学センターではガンマ線バーストのような高エネルギー宇宙現象の解明に向け、かなた望遠鏡を使い人工衛星の観測と連携した多波長観測を行っている。

かなた望遠鏡は国内最大級である 1.5m の主鏡を持ち、高い集光力を持つ。また、東広島天文台に移設する際に望遠鏡の架台部分を改造したことで、任意の天体に対して 50 秒以内で指向可能な高い駆動性能を持っている (表 1.1)。この駆動性能は口径 1.5m クラスの望遠鏡としては世界最高水準である。さらに、大学所有の望遠鏡であり、他の観測に縛られることなく突発天体に対してフレキシブルに対応できるという利点がある。これらの利点を活かし、東広島天文台では突発天体の即時観測を行っている。特に発生後から急激に減光してしまうガンマ線バーストの可視光観測では人工衛星からのアラート (GCN) を受けて自動的に偏光観測を開始するシステムになっている。

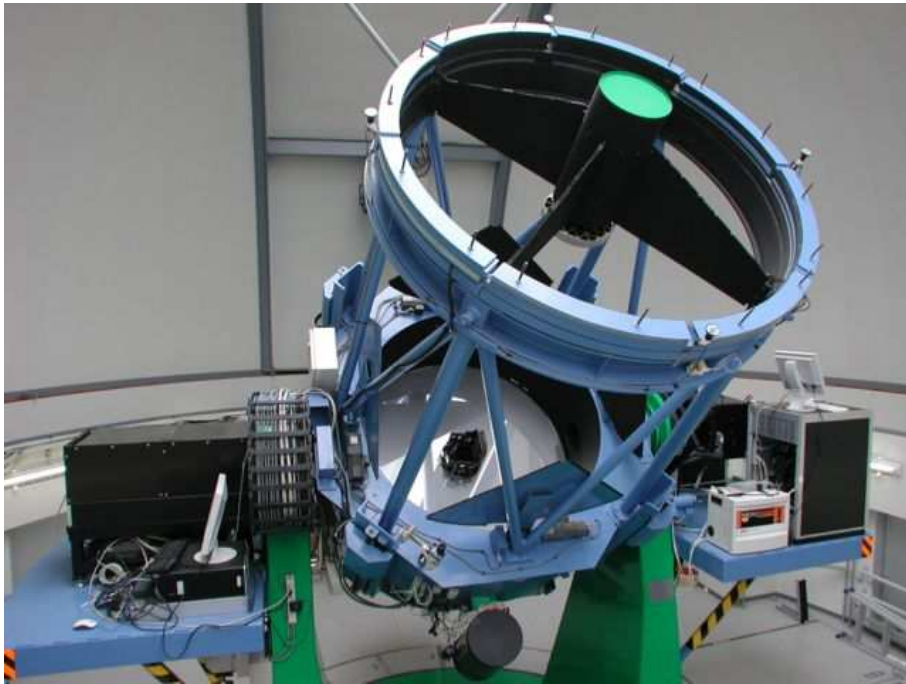


図 1.1: かなた望遠鏡

観測環境および利便性に優れた東広島市東南部の山頂部東広島天文台に設置された、かなた望遠鏡。青と緑の色は広島大学をイメージしたものである。写真中央に1.5m 主鏡、ナスミス台左に高速分光器、右に HOWPol が装着されている。

表 1.1: かなた望遠鏡の仕様

項目	仕様
光学系	リッチー・クレティアン光学系
主鏡	有効径 1540mm/主鏡の F 比=2.0
焦点モード	カセグレン焦点 (F/12.0)・ナスミス焦点 (F/12.0)
焦点面スケール	カセグレン 11.15 秒角/mm・ナスミス 11.15 秒角/mm
焦点距離	焦点距離 18,501.7mm
分解能	1" FWHM
視野	15 分角 ϕ
最大駆動速度	5°/sec (方位角) 2°/sec (高度軸)
最大化速度	1°/sec ²
架台	経緯台方式

1.1.2 観測装置

我々は高エネルギー天体の観測に特化した観測装置を開発し、かなた望遠鏡に取り付けて観測を行っている。現在、観測装置はカセグレン焦点に1つ、ナスミス焦点に2つ取り付けられている。カセグレン焦点には可視赤外線同時撮像分光装置「TRISPEC(Triple Range Imager and SPECTrograph)」(図 1.2) という名古屋大学で開発された装置が取り付けられている。この装置は可視1バンド赤外2バンドの同時撮像・分光・偏光観測を行うことができる。また TRISPEC の後継機として、可視赤外線同時撮像カメラ「HONIR(Hiroshima Optical Near-InfraRed camera)」を広島大学独自に開発中である。これは可視・赤外線同時3バンド撮像・分光・偏光観測が可能なので、完成すると TRISPEC に比べて大幅な感度・精度の向上が期待される。ナスミス焦点には高速分光器(図 1.4) と一露出型可視広視野偏光撮像器「HOWPol(Hiroshima One-shot Wide-field Polarimeter)」(図 1.3) が取り付けられている。高速分光器はブラックホール連星や激変星などの可視光の短時間変動の観測を目的としたもので、最速で 35.8 frame/sec の撮像観測と分光観測が可能である。HOWPol は広視野でかつ一回の露出で偏光情報を取得することができるものであり、本研究で取り扱う装置である。HOWPol については第 2 章以降で詳しく説明していく。



図 1.2: カセグレン焦点に設置された TRISPEC

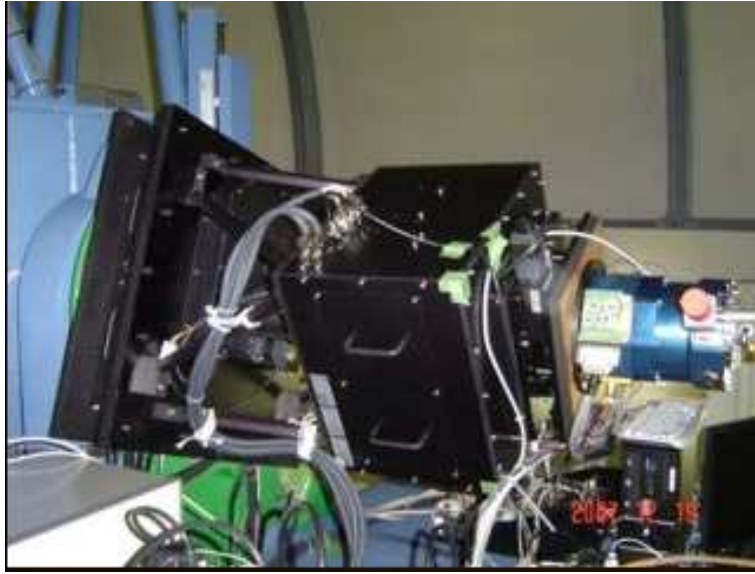


図 1.3: 第一ナスミス焦点に設置された HOWPol

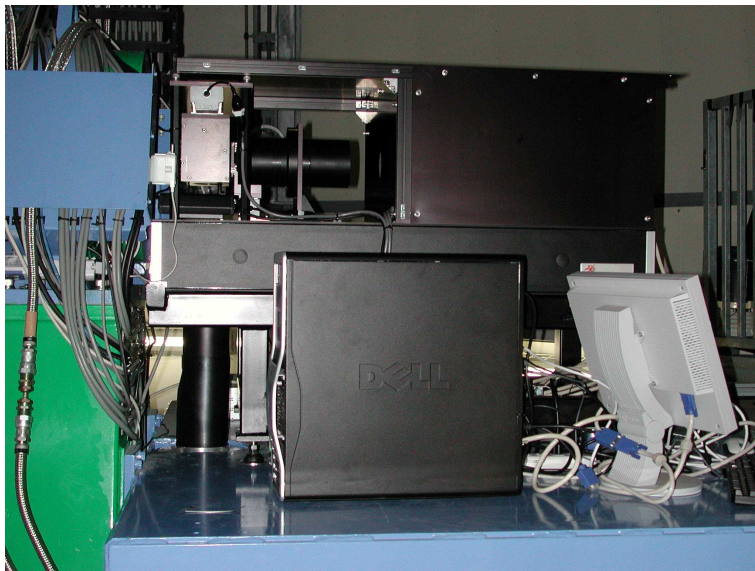


図 1.4: 第二ナスミス焦点に設置された高速分光器

1.1.3 ガンマ線バースト

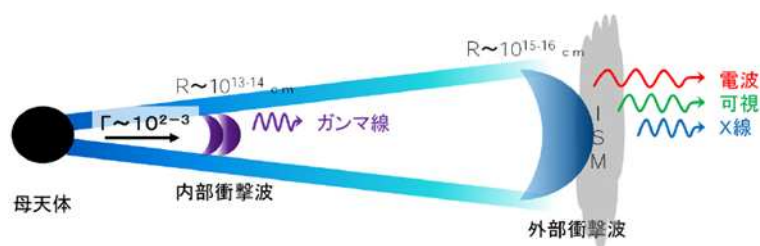


図 1.5: GRB のモデル (上原 岳士 2008 年度 修士論文)

ガンマ線バースト (Gamma-Ray Burst; GRB) とは約 250keV をピークに数 keV~数 MeV のエネルギーをもったガンマ線が天球上のある一点から地球上に多数のパルスとして到来する現象である。このガンマ線 (プロンプト放射) は 0.01~300 秒にわたって降り注ぎ、その Flux は最大 $10^{-6} \text{ erg s}^{-1} \text{ cm}^{-2}$ に達する。このガンマ線バーストは 1967 年にアメリカの軍事衛星 VERA によって発見され、そのスペクトルから非熱的な放射であることが分かった。そして、1991 年に打ち上げられた CGRO 衛星に搭載された BATSE によって、1 日に 1 個の頻度で天球面上で等方的に発生することがわかった。さらに、 T_{90} (ガンマ線バーストのフォトンの 90% が到来する時間) の個数分布が 2 つに分かれることを発見し、 $T_{90} \leq 2\text{s}$ のものを short GRB、 $T_{90} \geq 2\text{s}$ のものを long GRB と呼んでいる。また、short GRB のスペクトルは高エネルギー側でピークを、long GRB のスペクトルは低エネルギー側にピークを持つ傾向が見られることから、short GRB と long GRB では違う現象であると考えられている。

その後 GRB Coordinate Network (GCN) というシステムが構築された。これは人工衛星でガンマ線バーストを発見した時にインターネットを通じて全世界にアラートを送信するというものである。これによって 1997 年には、long GRB の一部にガンマ線で光った後に X 線から電波で光る残光が発見された。この残光は時間の冪乗で減光する。2003 年には、GRB030329 において、その残光を Very Large Telescope (VLT) やすばる望遠鏡などで分光観測した結果、一般的な残光成分を差し引いた後に残った成分のスペクトルが Ic 型超新星爆発と一致した。これによって一部の long GRB は宇宙論的な距離にある大質量星の重力崩壊と関係していることが明らかになったが、ただし、すべての long GRB が超新星起源であるかどうかは不明であり、現在も研究が進められている。ガンマ線 (プロンプト放射) を放出する物理機構として、ブラックホールの発生に伴ってできた火の玉から相対論的な速さの電子・陽電子プラズマのシェルが複数放出され、この速度差を持ったシェルが合体し内部衝撃波を起こしガンマ線を放射するというモデル (図 1.5) があり、さらに合体したシェルが星間物質に衝突し外部衝撃波を起こすことにより、X 線から電波にわたる残光を放射すると考えられている。Short GRB は中性子星同士の合体やブラックホールと中性子星の連星の合体によって起こると言う説がある。

2004年に打ち上げられたSwift衛星によって、残光のより早い観測が可能となった。これによって long GRB 初期の X 線残光の振る舞い (t^{-3} の急な減光と $t^{-0.5}$ の平らな減光) や X 線のフレア等 (図 1.6)、それまでの予想と違うものが発見された、この原因については現在もよくわかっていない。

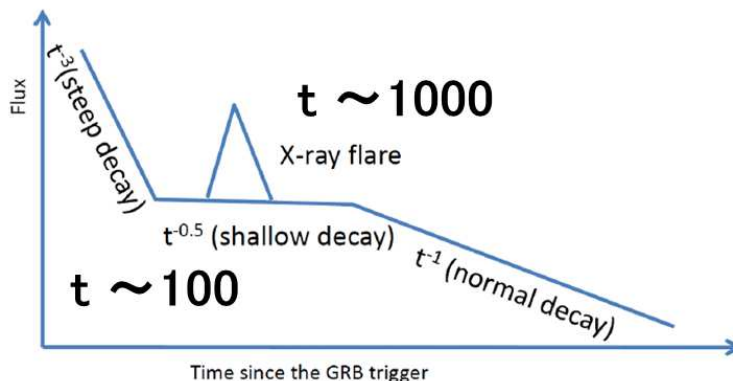


図 1.6: X 線の初期残光の振る舞い (上原 岳士 2008 年度 修士論文)

1.1.4 ガンマ線バーストの偏光観測

ガンマ線バーストの輻射機構は、プロンプト放射、残光のいずれもまだ解決しているとは言い難く、特にプロンプト放射については現在も精力的に議論が進められている。その解決の鍵を握るものとして、偏光観測が挙げられる。

プロンプト放射、残光ともにシンクロトロン放射が有力な発光機構であるので、偏光を測ることで輻射領域の磁場構造が推定可能となる。また、ジェット状の爆発をしている場合は、相対論的ビーミングの影響で見える領域が時間 (ジェットの減速) と共に変化することも予想される (D. Lazzati, New Journal of Physics, 9, 131 2006)。

ガンマ線バーストの残光の偏光観測例はいまだに稀少である。爆発から数時間経ってからの偏光観測では、GRB 021004 や GRB 020813、GRB 030329 で 1-5% 程度の偏光が捕えられた例があるが、光度曲線が特異なものが殆どで、爆発モデルの制限には至っていない。観測例が少ない理由は、偏光観測装置が取り付けられている望遠鏡が少ないことが根本的な原因であると考えられるが、通常の偏光装置では視野が狭く、位置誤差の大きい GRB に対する偏光観測が困難であることも要因に挙げられる。

そこで、我々のグループでは、爆発 100 秒後から 1000 秒程度までの、より明るい残光に対し、多数の GRB に対して偏光観測を行うことを計画している。かなた望遠鏡と HOWPol の組み合わせでこの観測が可能になる。第一目標は、残光の偏光観測による、残光輻射機構の解明であるが、long GRB の一部には、100 秒以上に亘りガンマ線のプロンプト放射

が継続するものがあり、そのような GRB に対しては、プロンプト放射の輻射機構の解明につながる可能性もある。

1.1.5 本研究の目的

Long GRB の物理的な特徴を詳しく知る為には、爆発後なるべく早い段階での ($t < 10^3$ 秒) 残光を観測する必要がある。例えば、この初期残光の偏光を観測することでジェット
の磁場構造や星周構造の情報を得ることが期待される。東広島天文台では、このガンマ線
バーストが発生してから数分から数十分の初期残光の偏光をとらえる為、GCN からのア
ラートを受けて即時観測が行われるシステムや、一回の露出で偏光パラメータを得るこ
とが出来る HOWPol を開発してきた。本研究の目的は、2008 年に観測に提供されていた
HOWPol において、偏光観測を行う為の制御システムの整備と、試験観測によって得ら
れたデータから偏光のキャリブレーションを行い、HOWPol で得たデータから正しい偏
光の値を導出することを可能なものにする事である。併せて、ハードウェアの立ち上
げ・維持を引き継ぎ、分光モードの実現などを目指した。

第2章 HOWPol

2.1 HOWPolの構成

2.1.1 筐体

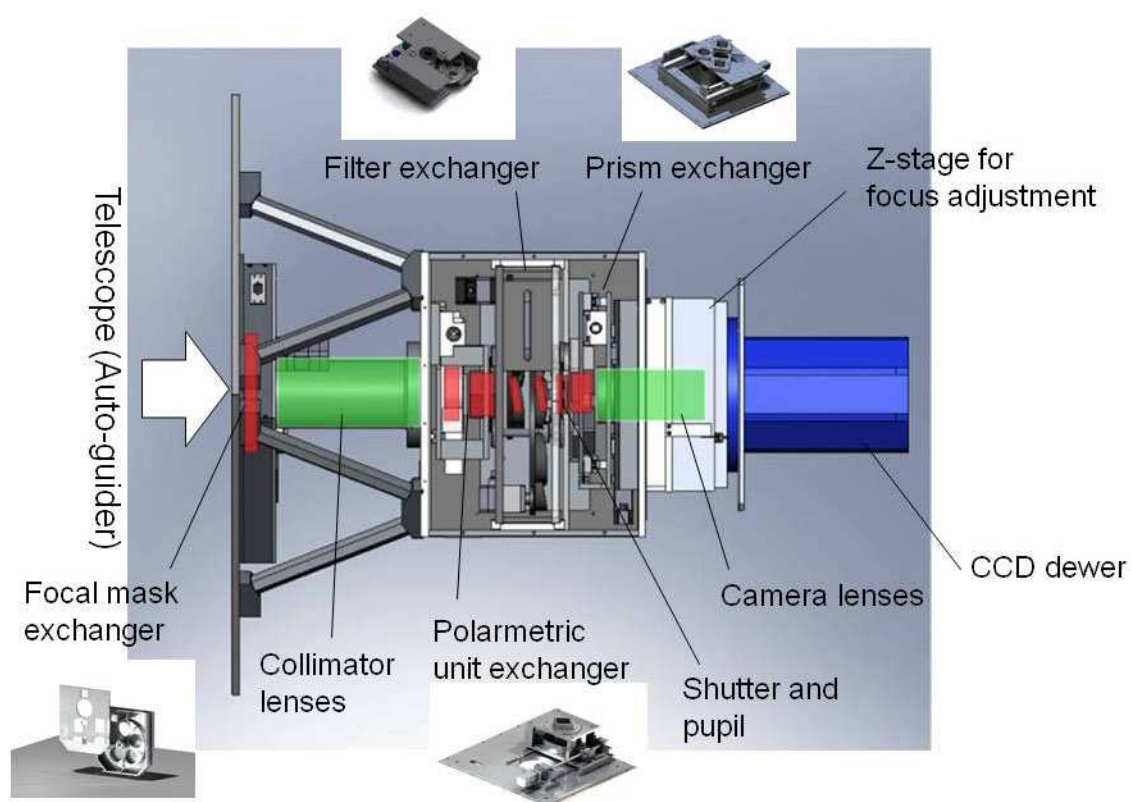


図 2.1: HOWPol の断面図

HOWPol の筐体について、横から見た断面図である図 2.1 の左から順番に説明していく。

図 2.1 の一番左が望遠鏡のナスミス焦点のローテータに取り付けられており、そこから光が入ってくる。次に撮像や偏光といった各観測モードに対応した焦点面マスクを交換できるターレットがあり、その右にコリメータレンズ系 (緑部分) がある。筐体中心の箱の部

分には左から偏光校正光学系用 X ステージ (半波長板)、フィルターターレット 2 つ、瞳マスク、シャッター機構、プリズム用 XY ステージがある (赤い部分)。そして、カメラレンズ系 (緑部分) があり、CCD デュワー (青い部分) がある。また、筐体中心の箱と CCD デュワーの間にはフォーカス合わせ用の Z ステージがあり、これを動かすことで CCD デュワーが動き焦点を調整出来る。

2.1.2 光学系

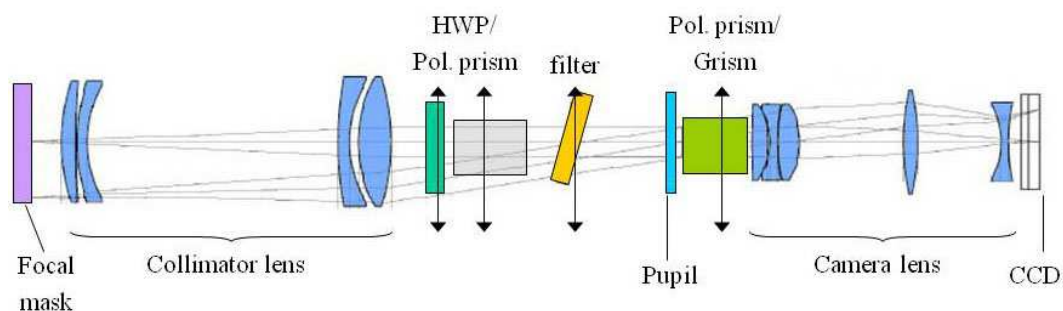


図 2.2: 光学系の概念図

項目	設計値
望遠鏡焦点から結像面までの距離	748.63mm
合成 F 値	6.9
平行光部長	266.1mm
瞳像直径	ϕ 23.9mm
560 < λ < 930nm 透過率	60 % 以上
400 < λ < 450nm 透過率	25 % 以上
450 < λ < 1100nm 透過率	38 % 以上
カメラレンズ焦点距離	148mm
80 % Encircled Energy (450 < λ < 1100nm)	0.6 秒角 (視野前面 プリズム無い時)
歪曲収差	+ 2.4 %
瞳像の収差	500 < λ < 1100nm で 52 μ m 600 < λ < 1000nm で 32 μ m

表 2.1: 光学系の仕様

HOWPolは(株)レンズ屋によって設計された瞳像の収差が極めて小さい再結像光学系を採用している。その概念図は図2.2に示してあり、光学系の仕様は表2.1となっている。図2.2の左(望遠鏡側)から入ってきた光をコリメータレンズ群で平行光にする。そして、その平行光をバンドパスフィルターやウォラストンプリズム等の様々な光学素子を通した後、カメラレンズ群でCCD上に再結合させるようになっている。

2.1.3 検出器系

HOWPolの検出器は $2k \times 4k$ の完全空乏型CCDを2枚並べて $4k \times 4k$ として使っている(図2.3)。2枚のCCDは、それぞれchip0(右)、chip1(左)と呼んでいる。図2.4はファーストライトとしてHOWPolの撮像モード(後述)で撮ったM16の画像を三色合成したものである。2枚のCCDの間には有効ピクセル間で1mm程のギャップがあり、図2.4の真ん中が暗くなっているのがそれに対応する。この検出器は空乏層を 200μ と通常の10倍以上厚くすることによって、波長 $1\mu\text{m}$ の近赤外線領域まで量子効率を伸ばしたものとなっている。限界等級は一番感度があるRバンドの撮像で19.2等(10分露出 $\Delta m = 0.02$ mag)、偏光撮像で16.0等(10分露出 $\Delta P = 0.2\%$)である。ピクセルスケールは0.294秒である。読み出しの制御には国立天文台で開発されたMessia5とM-front2(Nakaya et al.2006)を使っている。このCCDはデュワーに入れ、真空度 $10^{-5} \sim 10^{-6} \text{Torr}$ 、温度約173K(-100°C)に保たれている。これら真空度と温度はシリアル通信で常にモニターされ、実際に観測されて取得する画像データのヘッダーにかかれるようになっている。

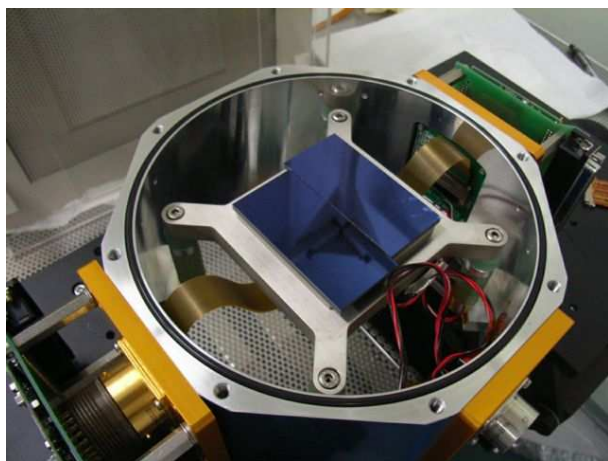


図 2.3: デュワーに取り付けられた CCD



図 2.4: HOWPolで撮られたM16の三色合成画像

2.1.4 駆動制御系

2.1.4.1 駆動系

HOWPolの駆動制御系には焦点マスクターレット、偏光校正光学系用Xステージ(半波長板)、フィルターターレット2つ、シャッター、プリズム用XYステージ、フォーカス合わせ用のZステージがある。これらのうちシャッターとフォーカス合わせ用のZステージ以外の機構には様々な光学素子が搭載されていて、それらの光学素子を選択的に光路上におくことによってHOWPolの観測モード(後述)を切替えるようになっている。以下、それぞれの駆動系機構で主な素子を示す。

焦点マスクターレット

焦点マスクは視野を絞る為のもので、使用する観測モードによって切替えて用いる。焦点マスクターレットには撮像用マスク、分光用マスク、狭視野偏光用マスク、広視野偏光用マスク、格子状マスクが入っており、ターレットが回転することでいずれかのマスクが光路上に入るようになっている。なお、この焦点マスクターレットは位置決め精度を向上させる為にプリロード機構が付いており、時計と逆方向に回すと機構が壊れてしまうので注意が必要となる。

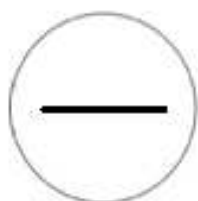


図 2.5: 分光用マスク
(スリット幅 $0.2\text{mm} = 2.3''$)



図 2.6: 狭視野偏光用マスク



図 2.7: 広視野偏光用マスク

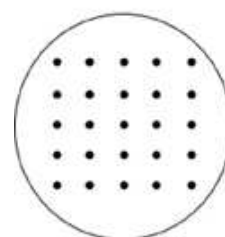


図 2.8: 格子状マスク

偏光校正光学系用 X ステージ

HOWPol の偏光観測は 1 露出型の他に、通常の偏光観測と同じように半波長板を入れて偏光データを取得する方法もある (詳しくは次の章で)。その為、半波長板を光路上に出入りさせる機構として偏光校正光学系用 X ステージがある。また、半波長板を使った偏光観測では半波長板を回転させる必要があるので、回転するステージがこの X ステージに取り付けられている。

フィルターターレット

フィルターターレットは、回転によって選択的に特定のフィルターを光路上に入れる為のものである。フィルターターレットにはフィルターを 5 枚 (+素通し) 搭載することができ、そのターレットが 2 つある。いずれも、光学素子間の多重反射による迷光を避けるため、フィルターを 10 度傾けて装着するようになっている。現在、一方にはバンドパスフィルターである B、V、Rc、Ic、z'+Y が入っており、もう一方にはハルトマン板が 2 枚、偏光フィルター (青)、波長校正用の時に使う青いフィルター、そして分光時に使う R64 というフィルターとが入っている。この R64 というフィルターは約 640nm 以下の波長をカットするのものである。

プリズム用 XY ステージ

プリズム用 XY ステージは X と Y の方向それぞれに動くことで観測に使用するプリズムを選ぶ事ができる。現在、撮像用の素通し以外に、狭視野偏光撮像用ダブルウォラストンプリズム (図 2.9)、広視野偏光撮像用ダブルウォラストンプリズム (図 2.10)、そして分光用グリズムが搭載されている。分光用グリズムの仕様は表 2.2 のようになっている。



図 2.9: 狭視野偏光撮像用ダブルウォラストンプリズム



図 2.10: 広視野偏光撮像用ダブルウォラストンプリズム

ウォラストンプリズムは複屈折性をもつプリズムを複数個、光学軸をずらして接合させたものである。ランダムな偏光を持った光をこのウォラストンプリズムに垂直に入射

素材	BK7 頂角 25.67°
格子数	420/mm 620mm フレーズ
波長域	450-1050nm(中心 620nm)
分散	平均 2.45Å /pix
分解能	18.8Å R~300

表 2.2: 分光用グリズムの仕様

させることで、互いに垂直な直線偏光の成分である常光 (ordinary ray ; o 光線) と異常光 (extraordinary ray ; e 光線) に分離することが出来る (図 2.11)。また、ダブルウォラストンプリズムは2つのウォラストンプリズムを光学軸を 45° 回転して接合させたものである。それぞれのウォラストンプリズムに半分づつ光が入るように、接合部分に光を入射することで 0°、45°、90°、135° の直線偏光成分に分割することができる (図 2.12)。

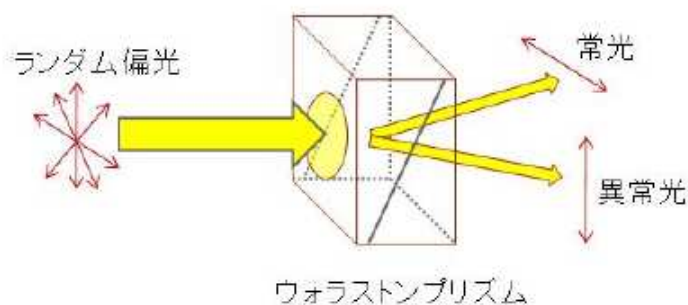


図 2.11: ウォラストンプリズム

フォーカス合わせ用の Z ステージ

フォーカス合わせ用の Z ステージはデューワーを光路軸方向に対して動かす為の機構である。このステージを前後させることで焦点をあわせることができる。

2.1.4.2 制御系

HOWPol は、ターレットや X ステージ・Y ステージといった、それぞれの駆動系にステップモータ (オリエンタルモータ製) を取り付けて制御し、観測モードの切替えを行う。モータは合計で 8 個あり、それぞれ独立のドライバーに取り付けられている。そし

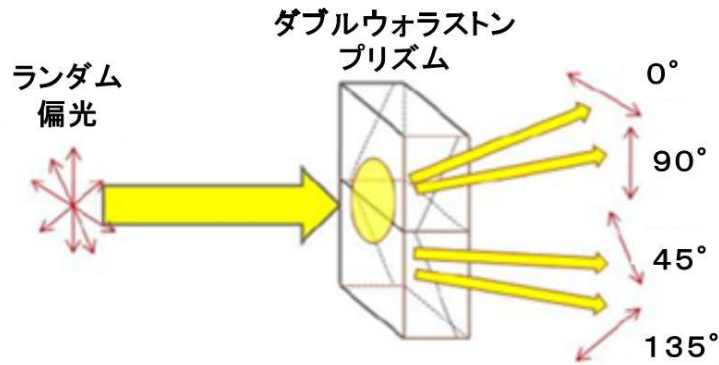


図 2.12: ダブルウォラストンプリズム

て、それらのドライバーは2つの4軸同時制御ボード(コンテック製)に取り付けられており、パソコンから8つのモータを独立で(同時に)制御することができる。また、それぞれの駆動系にはフォトセンサー(OMRON製)を取り付けることで、原点やリミットを検知することができる。検出器系とシャッターは別のPCで制御される。

上記の筐体、光学系、検出器系、および駆動系といったハードウェア部分は、HOWPol開発チームの前任者の人達が設計・組み立てを行い、2008年にはかなた望遠鏡に取り付けられて観測が開始された。2009年より開発チームに参加した私は、これらハード面の立ち上げを引き継ぎ、分光用グリズムとスリットの取り付け、シャッターの交換(経年劣化による為)、X(Y)ステージ機構のカップリングの交換などの機能維持及び更新に関する作業を行った。

駆動制御系についてはGUI(Graphical User Interface)操作をベースとした、観測モード切替えの為の制御ソフトウェアを、前任者の田中氏(2008年度修士論文)が作成した。これは駆動制御PC上で起動させ、制御室のPCからVNC(Virtual Network Computing)と言うリモートデスクトップソフトを使い遠隔操作するというものである。これは立ち上げ段階では便利に機能したが、実際の観測ではスクリプトによる自動制御を行う為、別のソフトウェアの作成が別途行われていた。それが、別駆動制御PC上にサーバープロセスを立ち上げ、制御室のPCからコマンドライン上で命令を行うCUI(character user interface)操作をベースとした制御ソフトである。これについても田中氏が取り掛かっていたが、未完成のままだった。そこで、私はこれら制御に関するサーバー・クライアントソフトウェアの開発を行った。さらにそれを使い、実際に観測を行うときに使う為の、望遠鏡操作を含めたプログラム(コマンド)を作成した。以下、簡単にそれらのソフトウェアについて記述する。

サーバーとクライアント

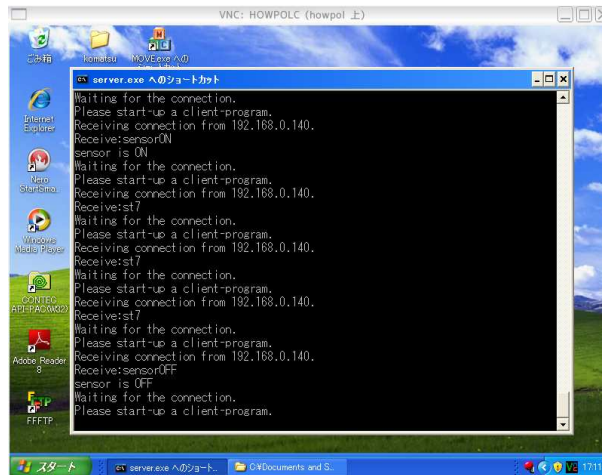


図 2.13: サーバプログラム起動の様子

サーバプログラムは駆動制御 PC(OS:Windows) 上にコンソールアプリケーションとして作成した(図 2.13)。コンテック社製のモータドライバを制御するライブラリとして Visual C++用のものしか提供されなかったため、開発には Visual C++を用いた。このサーバはクライアント側から命令(文字列)を受け取ると、その命令に応じてモータドライバに指示を送る。また、このサーバは一度プロセスを起動するとループ関数でクライアント側からの命令を待つようになっており、普段の観測では立ち上げた状態のままよい。ドライバのデバイス名(Axis_0~Axis_7)と駆動系の対応は表 2.3 のようになっている。なお、詳しいプログラムの内容はこの論文の最後に補遺として付録する。

クライアントはサーバ側に文字列で命令を送信するプログラムである。これは linux の PC 上に C 言語で作成した。クライアントプログラム(「howpolmv」)では、ターレットや XY ステージに wh1 から wh7 までの番号を付けてある(表 2.3)。また、現時点でのクライアントプログラムの主な機能は以下のようにになっている。

- wh1~wh7 の内 1 つを任意のパルス数分動かす
- wh1~wh7 の内 1 つを選択した素子が光路上に入るように動かす
- wh1~wh6 全てをそれぞれ選択した素子が光路上に入るように動かす
- 撮像モード(撮像用マスク、バンドパスフィルター)にする
- GRB モード(広視野偏光用プリズ、Rc フィルター、広視野偏光用マスク、半波長板無し)にする
- 分光モード(分光用グリズム、R64 もしくは素通し、分光用スリット)にする

- 狭視野偏光モード (狭視野偏光用プリズム、バンドパスフィルター、狭視野偏光用マスク、半波長板無しもしくは有あり)にする
- 広視野偏光モード (広視野偏光用プリズム、バンドパスフィルター、広視野偏光用マスク、半波長板無しもしくは有あり)にする
- wh1～wh6 を同時に原点 (init) に動かす
- 現在のステータス (各ホイールの位置情報) を取得する
- フォトセンサーを ON/OFF する
- アラームをリセットする

撮像モードや GRB モード等に切替える時には、時間効率良くモード切替えが行われるようにモータの動きを考えた。

また、これらのサーバーとクライアント間の通信には、ストリームソケットを使った通信 (TCP/IP) を用いている。ストリームソケットは連続的なデータを境界のない形式で送る。ソケット通信の流れは図 2.14 のようになっている。

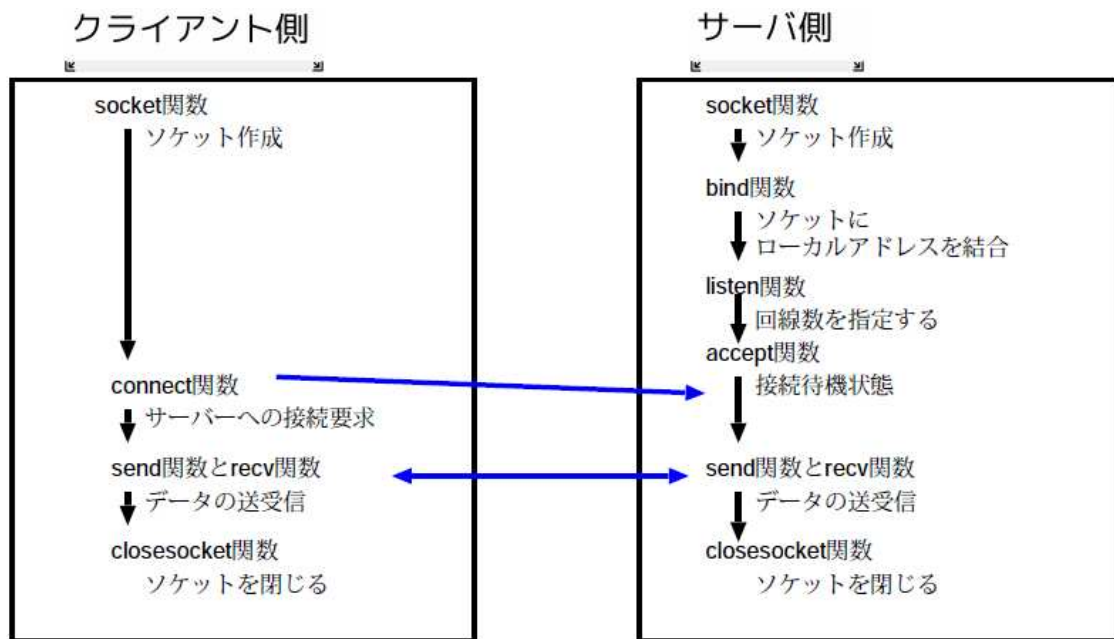


図 2.14: ストリームソケット通信の流れ

これらのプログラムを整備したことで、観測室にあるパソコンからの一元制御が可能となり、GRB の即時自動観測が可能となった。また、コマンドライン上で使うことのできる様々な観測用コマンドを作ることができ、観測効率は大幅に向上した。

ドライバー デバイス名	駆動機構	hwpolmv での名前	搭載素子
AXIS_0	プリズム用 X ステージ	wh1	狭視野偏光用プリズム 分光用グリズム 素通し 広視野偏光用プリズム
AXIS_1	プリズム用 Y ステージ		
AXIS_2	フィルターホイール 1	wh2	R64 ハルトマン板 1 波長校正用板 ハルトマン板 2 偏光素子 (青) 素通し
AXIS_3	フィルターホイール 2	wh3	B V Rc Ic z'+Y 素通し
AXIS_4	焦点マスクホイール	wh4	狭視野偏光用マスク 広視野偏光用マスク 撮像用マスク 分光用スリット 格子状マスク
AXIS_5	偏光校正光学系用 X ステージ	wh5	半波長板 (IN/OUT)
AXIS_6	半波長板回転ステージ フォトセンサー ON/OFF	wh6	半波長板 (0°-360°)
AXIS_7	フォーカス合わせ用 Z ステージ	wh7	

表 2.3: wh1~wh7 の割り当て

2.2 HOWPolの観測モード

2.2.1 撮像観測モード

HOWPolで撮像観測を行うモードである。光路上に撮像用マスクとバンドパスフィルター(もしくは素通し)を置くことで、望遠鏡の全視野15分角 ϕ を撮ることができる。撮像モードで実際に撮った生画像を図 2.15 に示す。ここで、左の画像が chip1(X座標左が正、Y座標上が正)、右の画像が chip0(X座標右が正、Y座標下が正)である。また、Y座

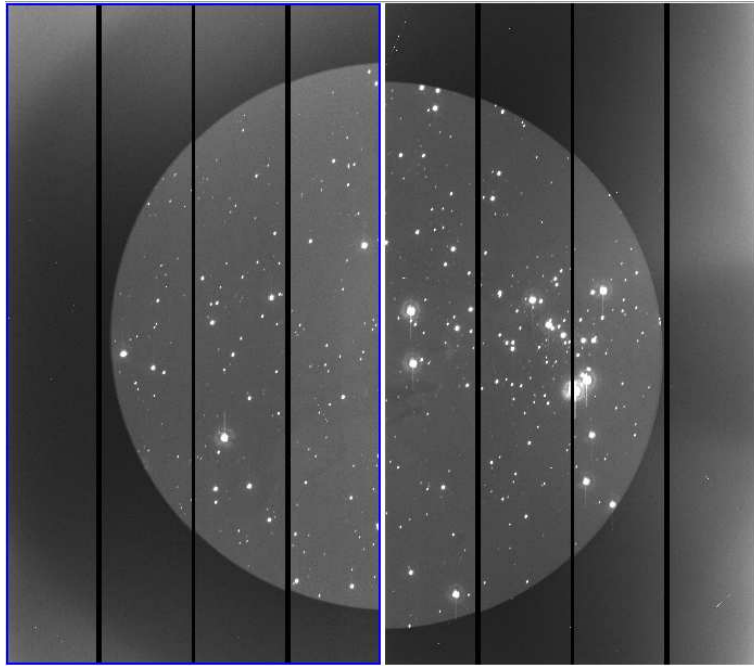


図 2.15: 撮像モードで得られる生画像 (M16、Vバンド)

標方向に黒い線が短冊の様になっている部分はプリスキャン領域、オーバースキャン領域と呼ばれる部分である (表 2.4)。HOWPol の CCD は 1 つの chip で 4 つのポートから電荷を読み出している。そして電荷を読み出した後、それぞれのポートで空読み出しを行ったものがこの領域である。この領域の値はバイアスと呼ばれ、得られるカウントのゼロレベルを与えてくれる。得られた画像から正確に測光を行う場合には、このバイアスの値を引く必要がある。HOWPol の場合、画像処理のソフト「IRAF」用に書かれた「howossub.cl」という c 言語スクリプトを使い、生画像からオーバースキャン領域を差し引き、この領域部分をカットした「オブジェクトフレーム」を作成する。また、CCD はポートごとや 1 ピクセルごとに感度むらが生じ、これらも補正してやる必要がある。その為には、ドームに付いているフラット板に均等な明るさのライトを当て、それを撮像した画像を用意する。その画像からバイアスを差し引いた後、規格化した「フラットフレーム」を作る。そして、先のオブジェクトフレームをフラットフレームで割ることで CCD の感度むらを補正することができる。この生画像からバイアスを差し引き、CCD の感度むらを補正するという一連の作業をリダクション (一次処理) と呼んでいる。これは撮像モードだけでなく、後述する分光モードや偏光モードにも必要な操作である。

ポート	1	2	3	4
プリスキャン領域	1-8	1065-1072	1073-1080	2137-2144
データ領域	9-520	553-1064	1081-1592	1625-2136
オーバースキャン領域	521-536	537-552	1593-1608	1609-1624

表 2.4: プリスキャン領域とオーバースキャン領域 (X 座標、単位 pix)

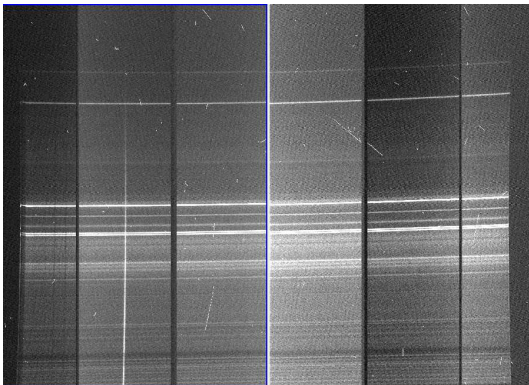


図 2.16: 分光モードで得られる生画像 (3c454.3)

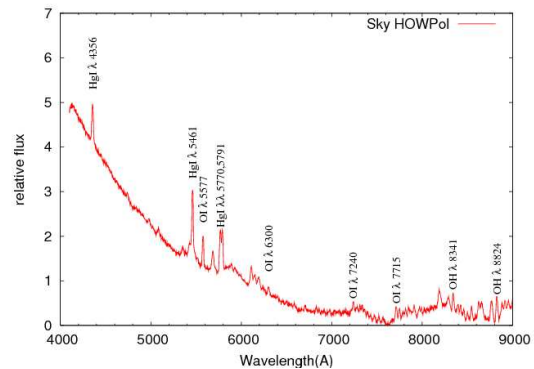


図 2.17: スカイの分光データ

2.2.2 分光観測モード

HOWPol で分光観測を行うモードである。このモードにする為には、分光用スリット、分光用グリズム、そしてオーダーカットフィルター (もしくは素通し) を光路上に置く。HOWPol での分光観測の手順は以下ようになる。

1. 撮像モードで撮り、分光したい天体の CCD 上の位置を確認する
2. 上の状態からマスクのみを分光用スリットと交換し撮像、スリット位置を確認する
3. 望遠鏡を動かし、スリット位置に天体がかかるようにする
4. 撮像し、分光出来る位置に天体が入っているかを確認する
5. 分光モードに切替えて露出を開始する

これらの手順を踏むことで分光データを得ることができる。ただし、HOWPol の駆動系の制御がコマンドライン上で出来るようになったものの、これらの動作をひとつずつしては、露出を開始するまでに時間がかかってしまう。望遠鏡の追尾性能には誤差があるため、長時間露出すると星がスリット位置から逃げてしまうという問題もあり、この作業をいかに効率良く行うかが重要である。そこで私は、望遠鏡と検出器への命令も含めた、駆動系の制御をインタラクティブに行うことのできるプログラム (howspec) を開発した。

これによって、上の1から5の手順をスムーズに行うことが出来、観測効率が大幅に向上した。さらに、2010年度佐藤卒論によって手がけられているオートガイダーが完成すれば、長時間の露出が可能となる見込みである。

分光モードで得られた生画像が図2.16である。X方向がスリット方向で、Y方向が波長方向である。データを解析する際には、まず撮像と同様にリダクションを行う。その後、波長同定が必要となる。これには、波長校正ランプがないしはスカイ(図2.17)を利用する。校正ランプはオートガイダーの完成によって利用できるようになる見込みである。

2.2.3 偏光観測モード

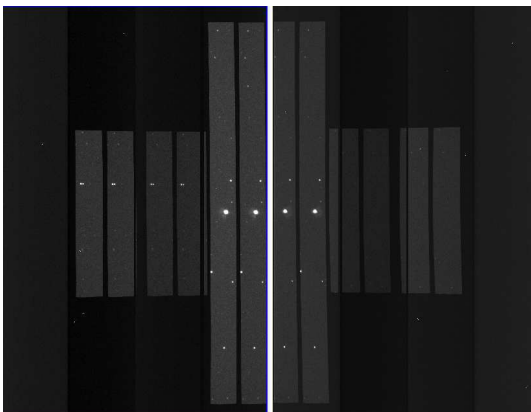


図 2.18:
広視野偏光モードで
得られる生画像 (Rc バンド)

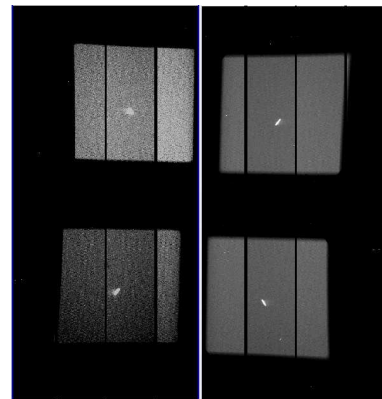


図 2.19:
狭視野偏光モードで
得られる生画像 (Rc バンド)

HOWPol で偏光観測を行うモードである。このモードには広視野偏光モードと狭視野偏光モードがある。広視野偏光モードは、ウォラストンプリズムで分けられる常光と異常光の分離角が大きく、7分角×7分角という広い視野の偏光観測をすることができる(図2.18)。この視野は後述する位置誤差の大きなGRBの観測に使われる。しかしその半面、色収差が大きいので図2.20や図2.21の様に星像が伸びてしまう。伸び方は図の様に短波長側の方が伸びが大きく(斜め上方向)、X方向にも薄く光が伸びている。その為、測光方法に注意する必要がある。一方、狭視野偏光モードは1分角×15分角と細い短冊状の視野(図2.19)ではあるが、色収差が小さく簡単に精度の良い偏光データを得ることができる。その為、位置の分かっている天体や、広がった天体に対してこのモードを使う。

広視野、狭視野モードともに、1露出型と半波長板を用いた2つの観測方法がある。1露出型は1回の露出で偏光データを得ることが出来る観測方法で、広視野用もしくは狭視野用の焦点マスクとダブルウォラストンプリズム、そしてバンドパスフィルターを使用する。この観測方法は素早い観測が可能で、且つ同時性のとれたデータを得ることができる



図 2.20:
広視野偏光モード
の色収差 (Rc バンド)

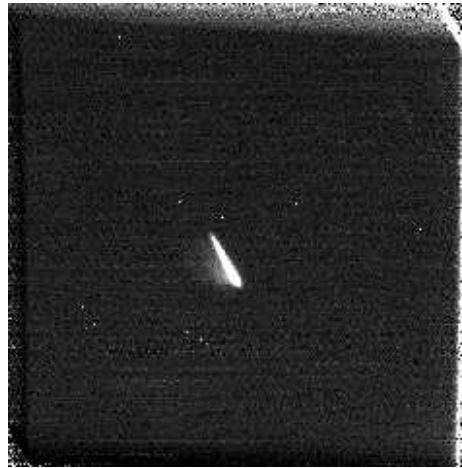


図 2.21:
広視野偏光モード
の色収差 (B バンド)

HOWPol 特有のものである。これによって、GRB の初期残光の観測を行う。半波長板を用いた観測方法は、1 露出型の光学素子のセッティングに加えて、半波長板を光路上に入れる。そして、その半波長板を 0° 、 22.5° 、 45° 、 67.5° と回転させながら 4 枚撮像することで偏光データを得る。ここで半波長板とは、直交する 2 つの偏光成分である常光と異常光のうち、片方の成分の位相を半波長分ずらすものである (図 2.22)。これにより、偏光度を変える事なく、偏光ベクトルの向きを回転させることが可能となる。

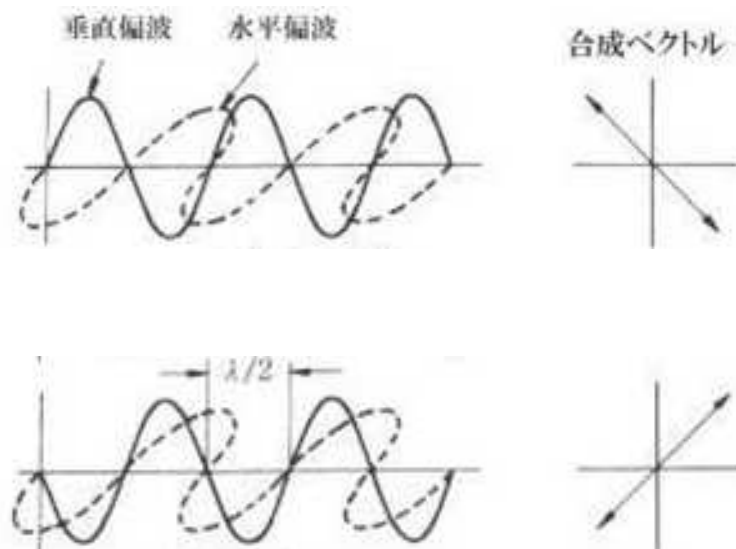


図 2.22: 半波長板の効果

それぞれのモードの偏光データを得る詳しい解析方法は次の章で紹介する。

2.2.4 GRB 自動観測モード

GCNによってGRB発生のアラートを受けると、HOWPolで即座に偏光観測を開始するモード(システム)である。GCNを受けてかなた望遠鏡をGRBの方向に向けるというシステムは、前任者の上原氏が2006年度卒論で開発した。GRB観測専用衛星であるSwiftのBAT(Burst Alert Telescope)は、バーストを検出するとすぐにその位置を約3分角の精度で自動的に決定し、GCNにアラートを流す(爆発から20-100秒後)。そのアラートを東広島天文台のサーバーが受け取り、GRBが現在観測可能な位置かどうかを判断する。もし、観測可能であれば現在行われている観測を中断、望遠鏡をその方向に向ける。同時に、私が開発した駆動制御系のサーバー・クライアントプログラムによって、GRBの観測モードとなるようにHOWPolの駆動系を制御する。この時、Swiftの位置誤差の3分角をカバーし、且つ時間分解能よく観測するために、GRBの観測モードは1露出型の広視野偏光撮像モードを使う。これによってアラートから最短1分程度でGRBの観測が出来るようになった。

なお、Swiftは、BATでGRBを検出した後、20~70秒以内にXRT(X-ray Telescope)とUVOT(Ultra-Violet Optical Telescope)によってX線、紫外線、可視光で観測する。XRTでの観測により約90秒以内に0.3-2.5秒角の精度で位置を決定することが出来るため、その後であれば狭視野偏光モードでの観測も可能である。

HOWPolは、GRBが出ていない晩には、その高い量子効率を活かして様々な天体の撮像・分光観測に活躍しており、極めて明るい超新星SN 2009dcの ^{56}Ni 生成量を導出するなど、成果も出つつある(Yamanaka et al. 2009)。

第3章 HOWPolの偏光キャリブレーション

3.1 偏光観測とその目的

3.1.1 偏光とストークスパラメータ

偏光 (polarization) とは、光の進行方向に垂直な面の方向に振動している電場または磁場が、ある方向に強く振動しているもの (ないしはその現象) を指す。偏光の方向は、通常は電場のベクトルの向きで表す。振動数を ν 、X 方向の振幅を ϵ_x 、Y 方向の振幅を ϵ_y 、そして X 方向と Y 方向の波の位相差を δ とおくと、一般に電磁波の電場 \vec{E} は、

$$\begin{aligned} E_x &= \epsilon_x(t)\cos(2\pi\nu t) \\ E_y &= \epsilon_y(t)\cos(2\pi\nu t + \delta) \end{aligned}$$

と表すことが出来る。位相差 δ が無いか、もしくは ϵ_x か ϵ_y が 0 の時は直線偏光、位相差 $\delta = -90^\circ, +90^\circ$ の時は円偏光となる。それ以外の場合、つまり一般的には楕円偏光となり、直線偏光や円偏光は楕円偏光の特別な場合とみなせる。HOWPol で観測できるのは直線偏光であるので、以降は主として直線偏光について記述する。

通常、偏光を観測したりデータを解析する際には、ストークスパラメータと呼ばれる、1852年にストークスが導入した4つの値 (I, Q, U, V) を使用する。偏光は元来、ベクトルであり、その計算はベクトル的に扱わなければならないが、この加算可能なパラメータを用いると、その計算が容易となる。また、光がどれだけ偏光しているかという偏光の強さを表す「偏光度」と、どの方向に偏光しているかを表す「偏光方位角」という、人間が認知しやすい量もこのパラメータを用いて簡単にあらわす事ができる。

ある値の時間平均を $\langle a \rangle$ と表すとすると、それぞれのパラメータは以下のように表される。

$$\begin{aligned} I &= \langle \epsilon_x^2 \rangle + \langle \epsilon_y^2 \rangle \\ Q &= \langle \epsilon_x^2 \rangle - \langle \epsilon_y^2 \rangle \\ U &= 2 \langle \epsilon_x \epsilon_y \cos \delta \rangle \\ V &= 2 \langle \epsilon_x \epsilon_y \sin \delta \rangle \end{aligned}$$

ここで、 I は総光量、 Q と U は直線偏光に関するパラメータである。また、 V は円偏光に関するパラメータである。直線偏光のみを取り出す場合は、 V を I に組み入れて $V=0$ とみなして良い。これらのパラメータを使って、偏光度 P と偏光方位角 θ は以下のように表すことができる。

$$P = \frac{\sqrt{Q^2 + U^2}}{I} \quad (3.1)$$

$$\theta = \frac{1}{2} \arctan\left(\frac{U}{Q}\right) \quad (3.2)$$

また、ある観測した天体から得られた Q, U の値を、 Q を横軸、 U を縦軸とした $Q-U$ 平面上にプロットした図が図3.1であり、 $Q-U$ 平面ないし $Q-U$ ダイヤグラムと呼んでいる。 Q と U はこのように互いに直交する座標系のパラメータとみなすことができる。式3.1と3.2より、この図の線分 RO の長さが偏光度 P となり、その線分と Q 軸とのなす角の半分が偏光方位角 θ となる。

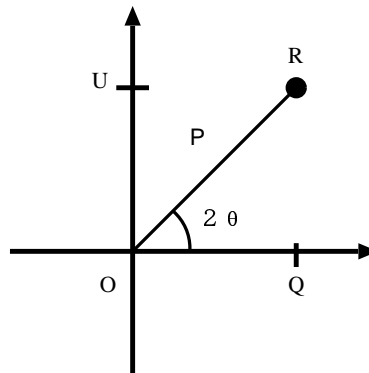


図 3.1: $Q-U$ 平面と偏光度、偏光方位角

Q と U 、そして偏光度と偏光方位角の詳しい求め方は次のセクションに記述する。

3.1.2 偏光観測の目的

可視光による偏光観測で主に、磁場構造や星周の幾何構造が分かる。揃った磁場中や、シンクロトロン放射からは偏光した光が放射されると考えられ、その偏光を観測することで磁場構造がわかる。また、偏光は光が散乱・反射した時にも起こる。よって、ある天体の光がその周りのガスや降着円盤に反射されている場合は、偏光した光を観測することでその天体周りの幾何構造を知ることができる。また、偏光はガンマ線やX線による観測では難しく、可視光による観測は重要である。

3.2 HOWPolによる偏光観測

3.2.1 ストークスパラメータの導出

観測したデータから偏光度 P と偏光方位角 θ を求める為に、まずストークスパラメータ Q, U を求める。HOWPol のデータから Q, U を求める方法は、半波長板を使った場合と1露出型の場合で多少異なっている。以下で解説する。

3.2.1.1 半波長板使用の場合

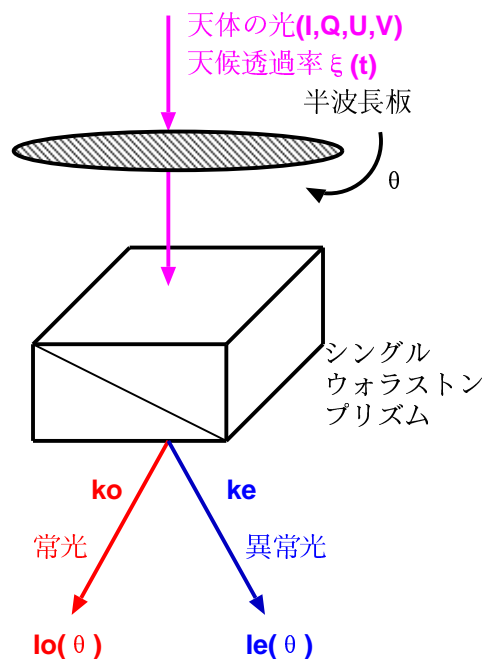


図 3.2: 偏光観測の概念図：半波長板使用

HOWPol で半波長板を使って観測する場合のストークスパラメータの求め方を説明する。図 3.2 はその時の観測の概念図である。図中のウォラストンプリズムがシングルウォラストンプリズムとしているのは、ダブルウォラストンプリズムの半分側つまり、chip1 もしくは chip0 だけを考慮しているとする (実際の観測された画像との対応は後述)。

まず、あるストークスパラメータ (I, Q, U, V) を持った天体の光が HOWPol に入ってきた時を考える。この光は時間依存の天候による大気透過率 $\xi(t)$ にしたがって透過光強度が変化することを考慮する。次に、その光がウォラストンプリズムの光軸の向きに対して θ だけ回転した半波長板を通過し、ウォラストンプリズムに入る。そして、ウォラストンプリズムで光が常光と異常光に分けられて出てくる (実際にはダブルウォラストンプリズ

ムなので chip1 と chip0 にそれぞれ常光、異常光が届く。)。ここで、添字の「o」と「e」は常光、異常光を表すものとする。このプリズムから出てきた光はそれぞれ、プリズムによってある効率 k_o, k_e で減光している。最終的に、半波長板を θ だけ回転させた時に観測される光の強度の値を $I_o(\theta)$ と $I_e(\theta)$ と置く。

以上の事を踏まえて、 $I_o(\theta)$ と $I_e(\theta)$ は下のよう書ける [9]。

$$I_o(\theta) = \frac{I}{2} \left\{ 1 + \frac{Q}{I} \cos 4\theta + \frac{U}{I} \sin 4\theta \right\} k_o \xi(t) \quad (3.3)$$

$$I_e(\theta) = \frac{I}{2} \left\{ 1 - \frac{Q}{I} \cos 4\theta - \frac{U}{I} \sin 4\theta \right\} k_e \xi(t) \quad (3.4)$$

ここで、半波長板は $\theta = 0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ$ の4つの位置角にセットするので、全てを書き下すと、下の様な8つの式になる。

$$I_o(0^\circ) = \frac{I}{2} \left\{ 1 + \frac{Q}{I} \right\} k_o \xi(t_{0^\circ}) \quad (3.5)$$

$$I_e(0^\circ) = \frac{I}{2} \left\{ 1 - \frac{Q}{I} \right\} k_e \xi(t_{0^\circ}) \quad (3.6)$$

$$I_o(22.5^\circ) = \frac{I}{2} \left\{ 1 + \frac{U}{I} \right\} k_o \xi(t_{22.5^\circ}) \quad (3.7)$$

$$I_e(22.5^\circ) = \frac{I}{2} \left\{ 1 - \frac{U}{I} \right\} k_e \xi(t_{22.5^\circ}) \quad (3.8)$$

$$I_o(45^\circ) = \frac{I}{2} \left\{ 1 - \frac{Q}{I} \right\} k_o \xi(t_{45^\circ}) \quad (3.9)$$

$$I_e(45^\circ) = \frac{I}{2} \left\{ 1 + \frac{Q}{I} \right\} k_e \xi(t_{45^\circ}) \quad (3.10)$$

$$I_o(67.5^\circ) = \frac{I}{2} \left\{ 1 - \frac{U}{I} \right\} k_o \xi(t_{67.5^\circ}) \quad (3.11)$$

$$I_e(67.5^\circ) = \frac{I}{2} \left\{ 1 + \frac{U}{I} \right\} k_e \xi(t_{67.5^\circ}) \quad (3.12)$$

上の式から $\theta = 0^\circ, 45^\circ$ の式 (3.5),(3.6),(3.9),(3.9) より、天候に関する効率 ξ を消して、

$$\frac{I_e(0^\circ)}{I_o(0^\circ)} = \left(\frac{1 - Q/I}{1 + Q/I} \right) \frac{k_e}{k_o} \quad (3.13)$$

$$\frac{I_e(45^\circ)}{I_o(45^\circ)} = \left(\frac{1 + Q/I}{1 - Q/I} \right) \frac{k_e}{k_o} \quad (3.14)$$

同じように、 $\theta = 22.5^\circ, 67.5^\circ$ の式 (3.7),(3.8),(3.11),(3.11) より、

$$\frac{I_e(22.5^\circ)}{I_o(22.5^\circ)} = \left(\frac{1 - U/I}{1 + U/I} \right) \frac{k_e}{k_o} \quad (3.15)$$

$$\frac{I_e(67.5^\circ)}{I_o(67.5^\circ)} = \left(\frac{1 + U/I}{1 - U/I} \right) \frac{k_e}{k_o} \quad (3.16)$$

これらより、

$$a_1 = \sqrt{\frac{I_e(0^\circ)/I_e(45^\circ)}{I_o(0^\circ)/I_o(45^\circ)}} \quad (3.17)$$

$$a_2 = \sqrt{\frac{I_e(22.5^\circ)/I_e(67.5^\circ)}{I_o(22.5^\circ)/I_o(67.5^\circ)}} \quad (3.18)$$

a_1 と a_2 をそれぞれ下のように定義すると、 k_o, k_e が消せて、

$$\begin{aligned} a_1 &= \frac{1 - Q/I}{1 + Q/I} \\ a_2 &= \frac{1 - U/I}{1 + U/I} \end{aligned}$$

以上より、

$$\frac{Q}{I} = \frac{1 - a_1}{1 + a_1} \quad \left(a_1 = \sqrt{\frac{I_e(0^\circ)/I_e(45^\circ)}{I_o(0^\circ)/I_o(45^\circ)}} \text{の時} \right) \quad (3.19)$$

$$\frac{U}{I} = \frac{1 - a_2}{1 + a_2} \quad \left(a_2 = \sqrt{\frac{I_e(22.5^\circ)/I_e(67.5^\circ)}{I_o(22.5^\circ)/I_o(67.5^\circ)}} \text{の時} \right) \quad (3.20)$$

$$I = \Sigma(I_o(\theta) + I_e(\theta)) \quad (\theta = 0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ) \quad (3.21)$$

となり、ストークスパラメータ Q, U は、装置の非完全性である k_o, k_e 、及び大気の変動性による効果である $\xi(t)$ に依らずに求めることができる。ただし、 0° - 67.5° の4フレームの取得が必要であり、4フレームの取得にかかる時間内の偏光の変化については求めることができないことに注意を要する。

プリズムの効率に関する値 k_e と k_o の比は、以下の様にして求めることができる。

$$\frac{I_e(0^\circ) I_e(22.5^\circ) I_e(45^\circ) I_e(67.5^\circ)}{I_o(0^\circ) I_o(22.5^\circ) I_o(45^\circ) I_o(67.5^\circ)} = \left(\frac{k_e}{k_o} \right)^4 \quad (3.22)$$

よって、

$$\frac{k_e}{k_o} = \left(\frac{I_e(0^\circ) I_e(22.5^\circ) I_e(45^\circ) I_e(67.5^\circ)}{I_o(0^\circ) I_o(22.5^\circ) I_o(45^\circ) I_o(67.5^\circ)} \right)^{\frac{1}{4}} \quad (3.23)$$

この値は、後述する1露出型でのリダクションに必要である。

3.2.1.2 1 露出型の場合

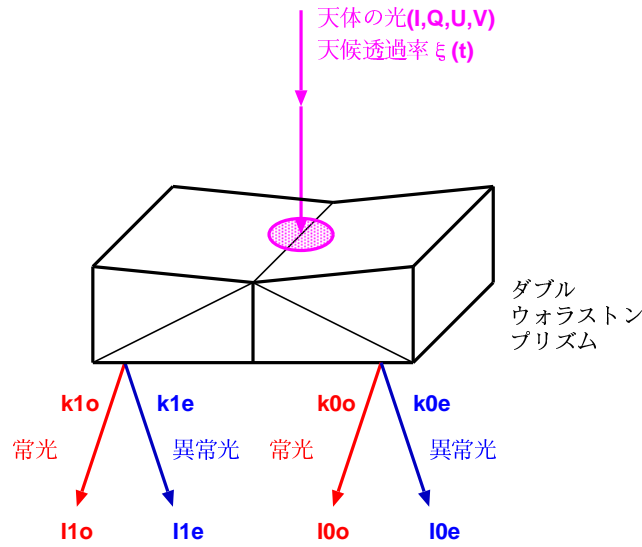


図 3.3: 偏光観測の概念図 : 1 露出型

次に、1 露出型の偏光観測をした場合のストークスパラメータの求め方を説明する。図 3.3はその時の観測の概念図である。1 露出型ではダブルウォラストンプリズムを使う。つまり、chip1 と chip0 両方を考える (実際の観測された画像との対応は後述)。

まず、半波長板を用いた場合と同様に、あるストークスパラメータ (I, Q, U, V) を持った天体の光が HOWPol に入ってきた時を考え、時間依存する天候による透過効率 $\xi(t)$ で変化することも考慮する。その光はダブルウォラストンプリズムに入る。そして、光が4つに分けられて出てくる。それぞれ、図の左から $I_{1o}, I_{1e}, I_{0o}, I_{0e}$ (観測量) と置く。ここで、添字の「1」「0」はウォラストンの左側右側 (chip の番号)、「o」「e」は常光、異常光に対応する。また、これらの光はそれぞれ、プリズムによってある効率 $k_{1o}, k_{1e}, k_{0o}, k_{0e}$ で減光している。

ダブルウォラストンプリズムは図の左半分から出てくる光と、右半分から出てくる光は互いに 45° 回転した成分を取り出すように作られている。また、常光と異常光は、光の電場成分の振幅方向のベクトルが直交しているので、 I_{1o} を基準 0° とすると、 I_{1o} は $0^\circ, I_{1e}$ は $90^\circ, I_{0o}$ は $45^\circ, I_{0e}$ は 135° の偏光成分を見ていることになる。入射前の光の θ 方向の偏光成分を $I(\theta)$ と書くとする、 ξ と k を考慮して、

$$I_{1o} = I(0^\circ)k_{1o}\xi(t) \quad (3.24)$$

$$I_{1e} = I(90^\circ)k_{1e}\xi(t) \quad (3.25)$$

$$I_{0o} = I(45^\circ)k_{0o}\xi(t) \quad (3.26)$$

$$I_{0e} = I(135^\circ)k_{0e}\xi(t) \quad (3.27)$$

と表すことができる。一方で、QとUは、

$$\frac{Q}{I} = \frac{I(0^\circ) - I(90^\circ)}{I(0^\circ) + I(90^\circ)}$$

$$\frac{U}{I} = \frac{I(45^\circ) - I(135^\circ)}{I(45^\circ) + I(135^\circ)}$$

であるので、式(3.24),(3.25),(3.26),(3.27)より代入して計算すると、 ξ を消去されて

$$\frac{Q}{I} = \frac{I_{1o} \left(\frac{k_{1e}}{k_{1o}} \right) - I_{1e}}{I_{1o} \left(\frac{k_{1e}}{k_{1o}} \right) + I_{1e}} \quad (3.28)$$

$$\frac{U}{I} = \frac{I_{0o} \left(\frac{k_{0e}}{k_{0o}} \right) - I_{0e}}{I_{0o} \left(\frac{k_{0e}}{k_{0o}} \right) + I_{0e}} \quad (3.29)$$

$$I = I_{1o} + I_{1e} + I_{0o} + I_{0e} \quad (3.30)$$

となる。 k_{1e}/k_{1o} や k_{0e}/k_{0o} は、半波長板を使った観測から求めた値(式3.23)を定数として使う。

3.2.1.3 HOWPol から得られた画像データとの対応

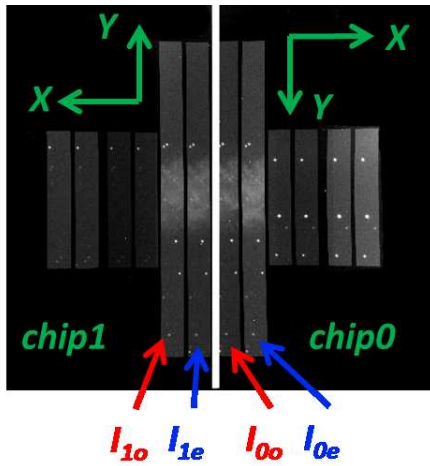


図 3.4:
狭視野偏光モードの画像

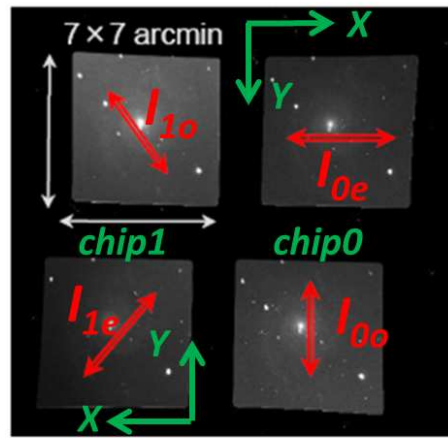


図 3.5:
広視野偏光モードの画像

狭視野偏光モード

観測で得られた撮像データは、通常図 3.4 のように左に chip1、右に chip0 の画像を表示させるようになっている。こうすると、縦方向が赤緯方向(上が北)に、横方向が赤経方

向(左が東)に対応し、他の星図との比較が容易になるためである。ピクセル数の座標が chip1 は X 座標が左が正、Y 座標が上が正に chip0 は X 座標が右が正、Y 座標は下が正になるように表示させる。1 露出型の場合、この状態で左の短冊から $I_{1o}, I_{1e}, I_{0o}, I_{0e}$ において、式 (3.28),(3.29),(3.30) により、ストークスパラメータを求める。半波長板を使用した場合、この画像の表示の状態、chip1 も chip0 も左の短冊が常光の $I_o(\theta)$ 、右が異常光の $I_e(\theta)$ とする。そして、半波長板を $0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ$ と回して得られた 4 枚の画像から、式 (3.19),(3.20),(3.21) を使いストークスパラメータを求める。偏光を求める時は同じ chip 同士のデータを使う。つまり、4 回の露出で chip1 から 1 組、chip0 から 1 組のストークスパラメータを求めることができる。

広視野偏光モード

広視野偏光モードの場合も、狭視野偏光モードと同じように左に chip1、右に chip0 の画像を表示させ、ピクセル数の座標が chip1 は X 座標が左が正、Y 座標が上が正に chip0 は X 座標が右が正、Y 座標は下が正になるように表示させる (図 3.5)。1 露出型の場合、図 3.5 の左上、左下、右下、右上の順番で $I_{1o}, I_{1e}, I_{0o}, I_{0e}$ となっており、式 (3.28),(3.29),(3.30) により、ストークスパラメータを求める。半波長板を使用の場合、chip1 は上が常光の $I_o(\theta)$ 、下が異常光の $I_e(\theta)$ とし、chip0 は下が常光の $I_o(\theta)$ 、上が異常光の $I_e(\theta)$ とする。そして、半波長板を使って得られた 4 枚の画像から、式 (3.19),(3.20),(3.21) を使い、それぞれの chip で 1 組ずつのストークスパラメータを求める。

3.3 偏光観測に特有なキャリブレーション

偏光観測で得られた画像から正確な偏光データを導き、天文学で議論を行う為には、偏光観測に特有のキャリブレーションを行う必要がある。それは主に、器械偏光の補正、器械消偏光の補正、そして方位角原点の補正の 3 つが挙げられる。以下にそれらを説明していく。

3.3.1 器械偏光

本来無偏光の光を観測しても、観測装置の光学系の不完全性によって、ある偏光度を持って観測される。この、望遠鏡も含めた装置内部で起こる偏光のことを器械偏光と呼ぶ。

光は反射すると一般に偏光する (図 3.6)。かなた望遠鏡で観測した天体の光が CCD に届くまでに反射されるのは主に、望遠鏡の主鏡、副鏡である。また、HOWPol はナスミス焦点 (図 3.7) に取り付けられているため第 3 鏡で 90° 反射される。通常、主鏡と副鏡はその軸対称性の為、それらの反射によって起こる偏光は互いに打ち消されて、カセグレン焦点での最終的な器械偏光は 0.1% 以下になることがほとんどである。ガンマ線バースト

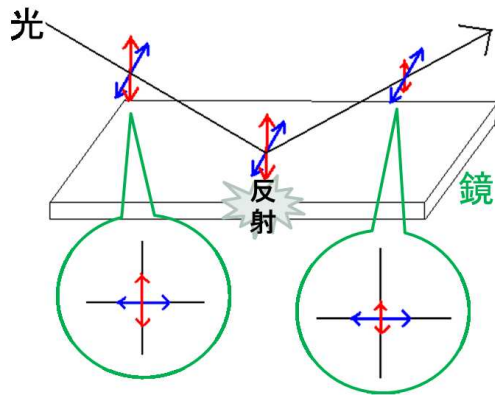


図 3.6: 反射による偏光

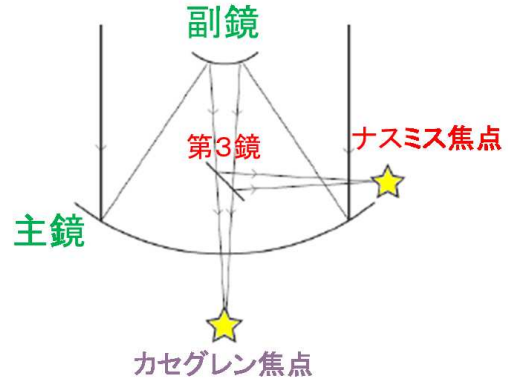


図 3.7: ナスミス焦点

の残光の偏光観測で求められる精度は0.5%以下であるので、このレベルの器械偏光は全く問題にならない。しかし、第3鏡による90°反射は約4%の器械偏光を生じる。よって、ナスミス焦点に取り付けられているHOWPolでは器械偏光の見積りが必須となる。これは、無偏光であることがあらかじめ分かっている、無偏光標準星を観測することで見積もることができる。

3.3.2 器械消偏光

器械消偏光とは、直線偏光の光の一部がレンズ等の光学系によって円偏光となるなどして、結果として観測される直線偏光の偏光度が減少するというものである。この器械消偏光は、ほぼ100%の偏光を生じさせるフィルターやプリズムを通した観測を行うことで見積もることができる。

3.3.3 方位角原点

偏光方位角は、通常偏光している向きを天球上で天の北極の方向を原点に左回りで表す。一方、器械系の方位角は、半波長板の回転の方向や原点、常光・異常光の採り方で変わってくる為、天球上の原点と回転方向になるように補正する必要がある。強偏光標準星を2つ観測することで、偏光方位角の原点を補正し、回転方向を決めることができる。

3.4 器械偏光の測定

3.4.1 調査項目

第3鏡由来の器械偏光を見積もる為に、以下の観点で調査を行う必要があると考えられる。

1. 光学的性質

- (i) 波長依存性
- (ii) 視野内依存性

2. 安定性

- (iii) 方向に対する安定性 (様々な天体に対して)
- (iv) 再現性 (第3鏡の安定性、日変化)

(i) は観測する波長による器械偏光の変化についてである。GRBの観測は主としてRcバンドの単バンドによる観測であるが、他の多くの天体では複数のバンドで観測することから、それぞれのバンドでの器械偏光の導出が必要となる。

(ii) はHOWPol(望遠鏡)の視野の中心付近と外側とでは、各レンズへの入射角の違いにより、器械偏光に違いが出ることを指す。これは、星団などを観測することで効率良く求めることができる。

(iii) は望遠鏡が向いている方向、つまり第3鏡の向きに依る器械偏光の変化についてである。これは、GRBを含む様々な天体を観測するためには必須となる。

(iv) はカセグレン焦点とナスミス焦点での観測を切替える為に、第3鏡を動かしたり、メンテナンスの為に取り外したりした時の器械偏光の再現性についてである。

私は、この中で任意の天体に対する器械偏光を推定する上で重要な(iii)について主に調査し、キャリブレーションを行った。

3.4.2 観測

前セクションで挙げた調査項目を満たすように、無偏光標準星 [10] の試験観測を行った。(iii) 方向に対する安定性や (iv) 再現性の観点から、違う赤緯 (Dec.) の3天体を長い夜数にかけて観測した。また、(i) 波長依存性の観点から1つの天体についてはB,V,Rc,Ic,zバンドで観測をした。また、狭視野偏光モードと広視野偏光モード両方で調査を行った。試験観測について表4.1にまとめる。

標準星	赤経 (RA.)	赤緯 (Dec)	V 等級	観測期間 (狭)	観測期間 (広)	フィルタ
HD125184	14 18 00.7	-07 32 32.6	6.5	2009.7/27-8/31	2010.6/3-7/23	Rc
HD154892	17 07 41.3	+15 12 37.6	7.8	2009.7/27-8/31	2010.6/3-7/23	B,V,Rc,Ic,z
HD212311	22 21 58.6	+56 31 53.0	8.1	2009.7/27-8/31	2010.6/3-7/23	Rc

表 3.1: 試験観測

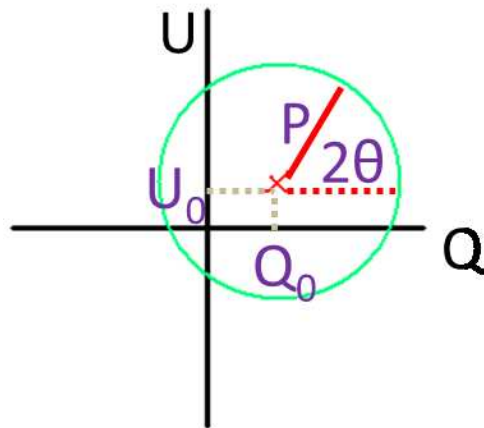


図 3.8: 第 3 鏡による器械偏光が円を描く様子

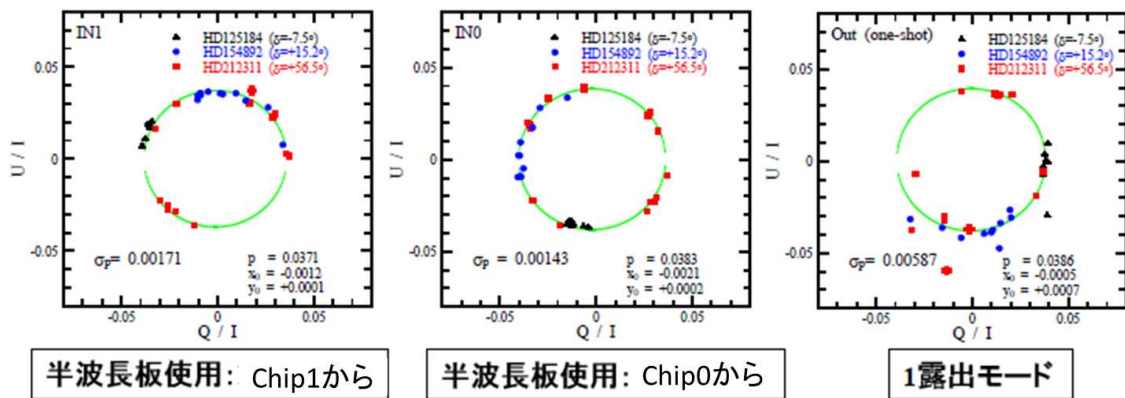


図 3.9: 円でフィットした様子

3.4.3 円によるフィット

器械偏光を探る手がかりとして、狭視野偏光モードで観測したデータを解析し、左から chip1, chip0, 1 露出型のそれぞれの結果を Q-U 平面上にプロットした結果が図 3.9 である。

ある点(Q_0, U_0)を中心に円を描いているのが分かる。これは、第3鏡由来のある一定な偏光度を持ちつつ、望遠鏡の向きによって偏光方位角が変化しているためであると考えられる(簡略化したものが図3.8)。このグラフを円でフィットした結果、偏光度(円の半径)は半波長板使用で chip1 から $P = 3.7 \pm 0.2\%$ 、chip0 から $P = 3.8 \pm 0.1\%$ 、1露出型から $P = 3.9 \pm 0.6\%$ と見積もった。円の中心である(Q_0, U_0)は第3鏡以外を由来とする装置固有の偏光であると考えられるが、 Q_0 も U_0 も ± 0.001 程度であり、ほとんど無視できる。

3.4.4 ナスミス焦点における器械偏光の定式化

第3鏡由来の器械偏光が望遠鏡の方向によってどう変わるのかを考える。まず、望遠鏡の動きを方位角方向と高度軸方向に分けて考える(図3.10,3.14)。



図 3.10: 方位角方向

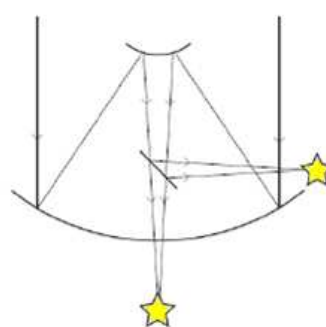


図 3.11: 方位角方向の動き (a)

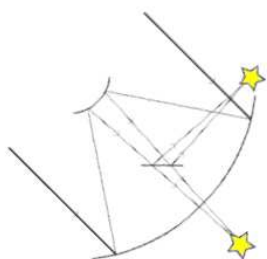


図 3.12: 方位角方向の動き (b)

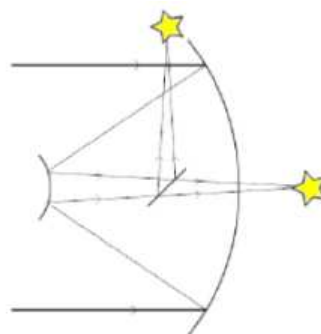


図 3.13: 方位角方向の動き (c)

方位角方向の動きを図3.11,3.12,3.13に表してある。図は鏡筒の断面図を簡略化したものである。図の見方を図(a)で説明すると、望遠鏡を上空から俯瞰で見ているとし、図において上から天体の光が入って来ているとする。下の方の曲面が望遠鏡の主鏡、上の曲面が副鏡、そして真ん中の斜めの面が第3鏡を表している。そして、主鏡側の星マークがカセグレン焦点、右側の星マークがナスミス焦点の位置を表している。(a)の状態、望遠

鏡がある方向 (例えば真北) を向いているとしよう。そこから、反時計回りに方向を変えていき、(b) の位置を通過して (c) の位置 (西) を向いたとする。この時、ナスミス焦点も第3鏡も同時に回転し、鏡筒に対して常に同じ位置・角度を保つ。よって、第3鏡とナスミス焦点の位置関係は変わらず、常に同じように反射した光がナスミス焦点で像を結ぶ。



図 3.14: 高度軸方向

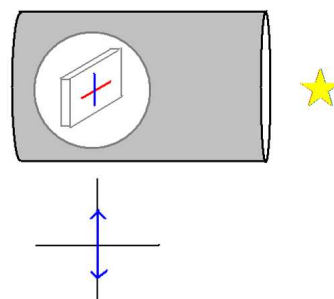


図 3.15: 高度軸方向の動き (a)

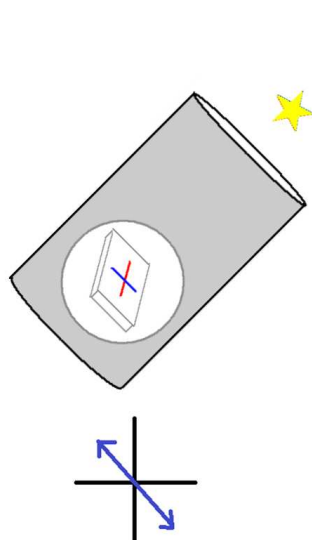


図 3.16: 高度軸方向の動き (b)

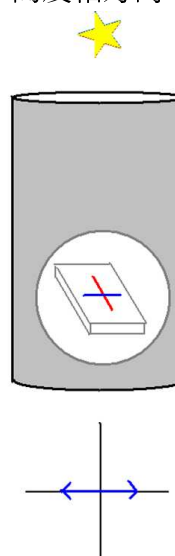


図 3.17: 高度軸方向の動き (c)

一方、高度軸方向の動きに関して表した図が図 3.15, 3.16, 3.17 である。これは鏡筒に対して真横から、つまりナスミス焦点に立って鏡筒を見ている状態である。星マークは観測する天体を表し、天体の光が入ってくる方向を表している。また、鏡筒がくりぬかれた中には第3鏡の向きが示してあり、図の下半分の矢印は光がナスミス焦点に入ってきた時の偏光状態を示してある。まず、鏡筒は図 (a) のように (実際にはありえないが) 高度 0° の星を見ているとする。この時、第3鏡の向きが、地面に対して鉛直な振幅を持つ偏光成分 (図の青色) は減衰することなくナスミス焦点に達する。しかし、地面に対して平行な振幅を持つ成分 (図の赤色) は反射によって減衰する。簡単の為、この成分が完全に減衰した (偏光の方向の基準) とすると、図 (a) の下半分で表されてるように鉛直方向に偏光し

た光がナスミス焦点へ届く。次に、(a)の状態から鏡筒を上方向に向けていき、高度45°の向き(b)を通過して(c)の様に高度90°(天頂)を向いたとする。同じように考えると、図(b)(c)のように、ナスミス焦点に届く光の偏光方位角は回転していく。

よって、第3鏡由来の器械偏光は方位角の動きには依らず、高度軸方向の動きだけに依ると言える。

このことから、ナスミス焦点における器械偏光を定式化していく。まず、第3鏡由来の器械偏光は高度軸方向の動きだけに依るということ、そして、図3.15(a)～図3.17(c)までで高度軸方向が90°動くと偏光の方位角が90°回転するということから、偏光方位角 θ と高度 h の関係は

$$\theta = h \quad (\theta = 0 \text{ で } h = 0 \text{ とする}) \quad (3.31)$$

となる。実際には、かなた望遠鏡と HOWPol の間にローテータという、写野回転を補正するための装置が付いているので、その写野回転角を θ_{nrot} と置くと、

$$\theta = h - \theta_{nrot} \quad (3.32)$$

ここで、高度 h は天頂距離 z を使って、 $h = \pi/2 - z$ と書ける。また、HOWPolの置かれている側のナスミス焦点ではカセグレン焦点での写野回転角を p と置くと、 $\theta_{nrot} = \pi - z + p$ となる[6][7]ので、式3.32は、

$$\begin{aligned} \theta &= h - (\pi - z + p) \\ &= \frac{\pi}{2} - z - (\pi - z + p) \\ &= -\frac{\pi}{2} - p \end{aligned} \quad (3.33)$$

となる。カセグレン焦点での写野回転角 p は時角 HA 、観測地の緯度 ϕ 、天体の赤緯 δ を使って表すことができ、

$$\theta = -\frac{\pi}{2} - \arctan(\sin HA / (\tan \phi \cos \delta - \sin \delta \cos HA)) \quad (3.34)$$

である。

なお、ストークスパラメータの定義から、器械偏光の Q, U 成分は P, θ より、

$$\begin{aligned} Q &= P \cos \{2(\theta - \theta_0)\} + Q_0 \\ U &= P \sin \{2(\theta - \theta_0)\} + U_0 \end{aligned} \quad (3.35)$$

として導くことが出来る。

3.4.5 器械偏光の時角・赤緯依存性

$$\theta = -\frac{\pi}{2} - \arctan(\sin HA / (\tan \phi \cos \delta - \sin \delta \cos HA))$$

$$Q = P \cos \{2(\theta - \theta_0)\} + Q_0$$

$$U = P \sin \{2(\theta - \theta_0)\} + U_0$$

式 3.34 より、ある観測地である天体を観測しているとき、 ϕ や δ は定数となり、 θ は時角 HA だけの関数 $\theta(HA)$ となる。そして、この θ を式 3.35 に代入することで、 Q と U も、ある観測地である決まった天体を観測している場合には時角 HA のみの関数となる。違う天体を観測する場合には、 δ が異なることから時角依存の仕方が異なるが、係数が一部変わる程度である。よっていくつかの赤緯での時角依存をうまく再現するモデル式が得られれば、その式を用いてどの位置の天体でも器械偏光を推定することができるようになる。

そこで、そのようなモデル曲線の式を求めた。まず、縦軸 Q, U で横軸 HA のグラフ上に観測で得られたデータ (半波長板使用 chip1, chip0, 露出型) をプロットした。そして、式 3.35 でフィッティングし、各パラメータ P, θ_0 を求めた (Q_0, U_0 は小さいので 0 とした)。フィッティングの様子 (例: HD125184, R バンド) を図 3.18, 図 3.19, 図 3.20 に示す。図の青の点がある HA での Q/I 、赤の点がある HA での U/I の値を示してあり、緑の線と赤の線は Q/I 、 U/I それぞれに対してフィッティングした曲線である。3 天体を観測したデータ点から求めた P と θ_0 はそれぞれ表 (3.2) の様になった。

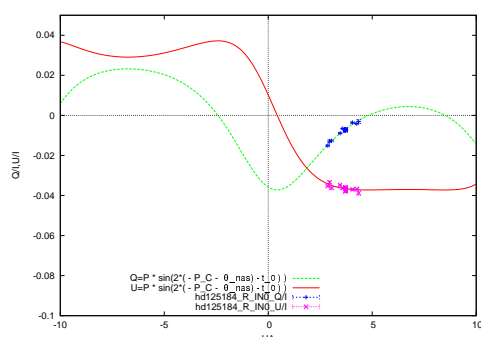


図 3.18: フィットした様子:
HD125184 R バンド chip0

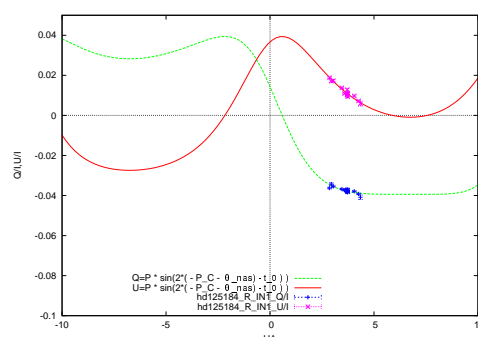


図 3.19: フィットした様子:
HD125184 R バンド chip1

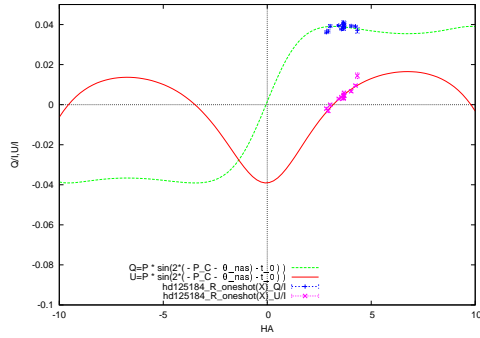


図 3.20: フィットした様子:
HD125184 R バンド oneshot

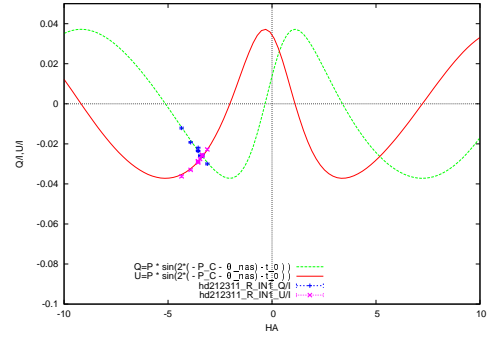


図 3.21: フィットした様子:
HD212311 R バンド chip1

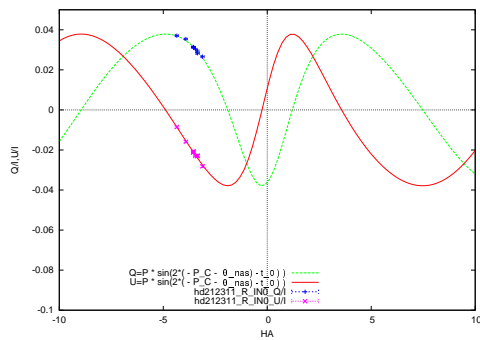


図 3.22: フィットした様子:
HD212311 R バンド chip0

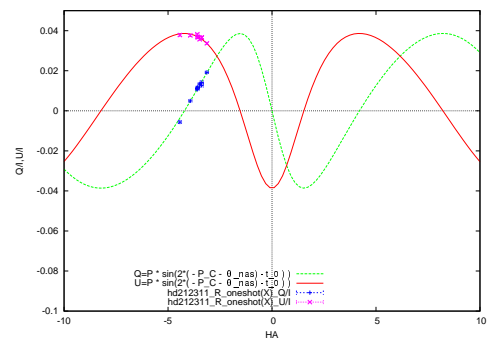


図 3.23: フィットした様子:
HD212311 R バンド oneshot

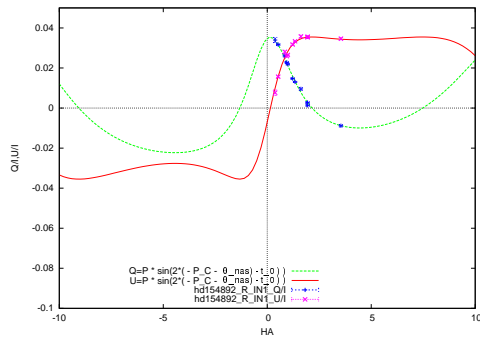


図 3.24: フィットした様子:
HD154892 R バンド chip1

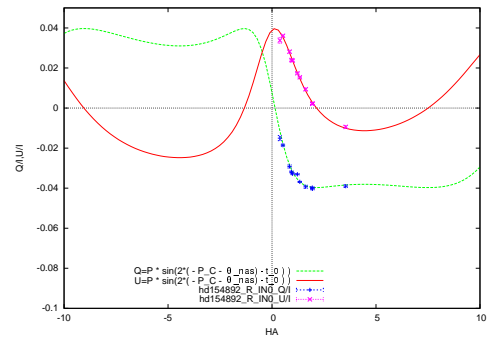


図 3.25: フィットした様子:
HD154892 R バンド chip0

ここで、この調査を行ったときはナスミスローテータのオフセットの原点 (CCD 上で北が上になる値) は -48.8° であった。この値を θ_N とする。表 3.2 より、 θ_0 はナスミスローテータのオフセットの角度 $\theta_{Noffset}$ に依存していることがわかり、

$$\theta_0 = \theta_{Noffset} - \theta_N + C$$

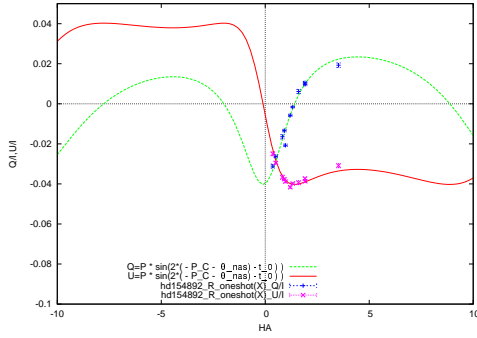


図 3.26: フィットした様子:
HD154892 R バンド oneshot

天体名	HD125184	HD154892	HD212311
観測時のナスミス ローテータのオフセット	-48.8°	90.4°	131.2°
P(oneshot)	0.0391 ± 0.0002	0.0403 ± 0.0002	0.0386 ± 0.0002
θ_0 [度](oneshot)	1.1 ± 0.1	-41.0 ± 0.1	-0.05 ± 0.1
$C_{oneshot}$	1.129	-0.2	-0.0461
P(chip1)	0.0394 ± 0.0001	0.0355 ± 0.0001	0.0372 ± 0.0001
θ_0 [度](chip1)	79.30 ± 0.09	39.3 ± 0.1	79.1 ± 0.1
C_{in1}	79.3	80.14	79.06
P(chip0)	0.0372 ± 0.0001	0.0397 ± 0.0001	0.0379 ± 0.0001
θ_0 [度](chip0)	-53.0 ± 0.10	84.5 ± 0.1	-53.29 ± 0.09
C_{in0}	-52.96	-54.71	-53.29

表 3.2: モデル曲線の各パラメータ

という定数 C があるとする。1 露出型、半波長板使用:chip1、半波長板使用:chip0 から求めたそれぞれのこの定数を $C_{oneshot}, C_{in1}, C_{in0}$ とする。3 天体それぞれから求めたこれらの定数の平均は、 $C_{oneshot}=0.3 \pm 0.4$ 、 $C_{in1}=79.5 \pm 0.3$ 、 $C_{in0}=-53.7 \pm 0.5$ となった。また、1 露出型、半波長板使用:chip1、半波長板使用:chip0 から求めたそれぞれの P を $P_{oneshot}, P_{in1}, P_{in0}$ とすると、3 天体それぞれから求めたこれらの値の平均は、 $P_{oneshot}=3.93 \pm 0.07 \%$ 、 $P_{in1}=3.74 \pm 0.16 \%$ 、 $P_{in0}=3.83 \pm 0.11 \%$ となった。

以上より、R バンドの Q, U - HA 依存性のモデル曲線が求まり、以下のようにになった。

$$\begin{aligned} \theta &= -\frac{\pi}{2} - \arctan(\sin HA / (\tan \phi \cos \delta - \sin \delta \cos HA)) \\ Q &= P \sin \{2(\theta - \theta_{Noffset} - \theta_N + C)\} \\ U &= P \cos \{2(\theta - \theta_{Noffset} - \theta_N + C)\} \end{aligned}$$

$\phi =$ 観測地の緯度 = 34.377222°(東広島天文台)

$\delta =$ 天体の Dec.

$\theta_{Offset} =$ 観測した時のナスミスローテータのオフセット

$\theta_N =$ CCD 上で上が北になるナスミスローテータのオフセット

$P_{onshot}=3.93\%$ 、 $P_{in1}=3.74\%$ 、 $P_{in0}=3.83\%$

$C_{onshot}=0.294$ 、 $C_{in1}=79.5$ 、 $C_{in0}=-53.7$

3.4.6 最適アパーチャサイズの決定

先に説明したように、広視野偏光モードでは色収差が大きいので星像が伸びてしまう。このようなデータでは、カウントを測るアパーチャサイズが小さすぎると、常光と異常光とで実質的に同じアパーチャーのカウントを得るのが難しく、逆に大きすぎるとスカイノイズの悪影響を受けるため、最適なサイズを求めなくてはならない。そこで、星像の中心からどれだけのアパーチャサイズ(半径)で測光すればよいかを調査した。調査には 2010.07/25 の HD154892 の観測データを用いた。

まず、各バンドごとに星像の中心付近からアパーチャサイズを大きくしていき測光を行った。そして、横軸をアパーチャサイズ、縦軸を総フラックスとしたグラフを作成した。そのグラフが図 3.27 である。緑が半波長使用:chip1、赤が半波長使用:chip0、青が 1 露出型である。これをみるとアパーチャサイズを大きくしていくと、総カウント数が大きくなっていくのがわかる。あるアパーチャサイズ(Rcバンドの場合約 60 くらい)でカウント数の変化が小さくなるのは、伸びた星像の内の明るい部分がアパーチャーの中に大体収まるからである。それ以降、アパーチャサイズがあまり大きすぎると値がバラつくのは、スカイの寄与が大きくなって来るからだと考えられる。また、半波長使用と 1 露出型で総カウント数の差があるのは、半波長板使用の場合 4 枚の画像を使う為、総露出時間が多くなるからである。そして、半波長板を使用した場合に chip1 と chip0 で総カウント数の差があるのは、ダブルウォラストンプリズムで光を分ける時に、まったく同じだけ分けられるわけではない為だと考えられる。

次に、アパーチャサイズに対する偏光度 P の変化を表したものが図 3.28 である。青が半波長使用:chip1、赤が半波長使用:chip0、茶色が 1 露出型であるこれをみると、アパーチャーが星像十分にカバーしてからしばらく(アパーチャサイズ 50~100)は偏光度の値のバラツキは小さいが、アパーチャサイズが大きくなると偏光度の値のバラツキが大きくなってしまふことがわかる。そこで、各バンドで偏光度の値のバラツキが小さくなり、且つ総カウントの 90% 以上を得ているアパーチャサイズの範囲を調べたものが表 3.3 である。

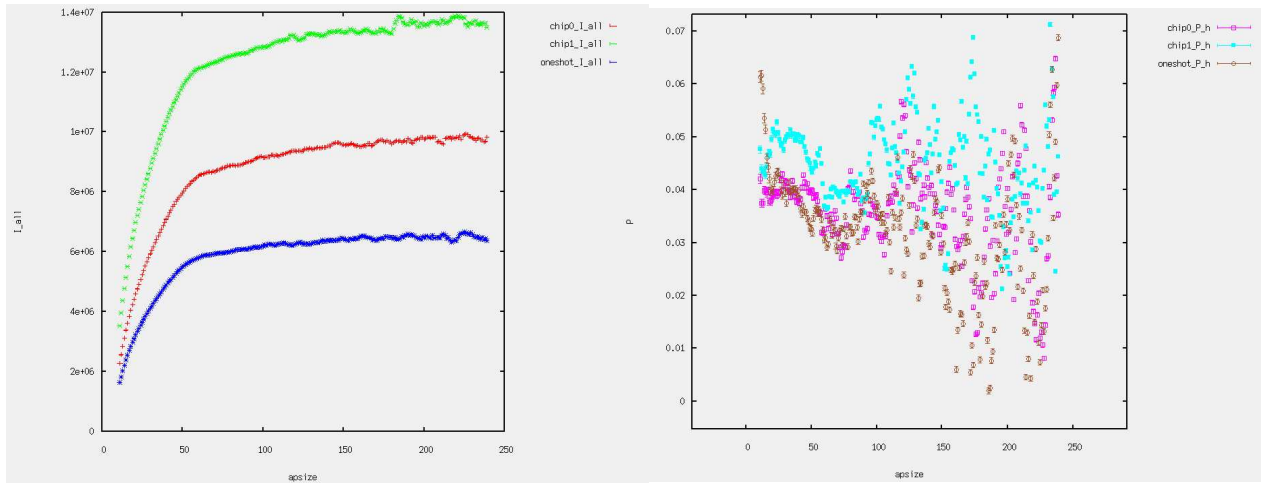


図 3.27: 広視野偏光モードのアーチャーサイズの決定 (縦軸:総カウント数 横軸:アーチャーサイズ) HD157982 Rc バンド
 図 3.28: 広視野偏光モードのアーチャーサイズの決定 (縦軸:偏光度 横軸:アーチャーサイズ) HD157982 Rc バンド

フィルター	モード (半波長板使用、1 露出型)	最適アーチャーサイズ (pix 数)
B	半波長板使用:chip1	150 - 230
B	半波長板使用:chip0	150 - 230
B	1 露出型	140 - 250
V	半波長板使用:chip1	80 - 150
V	半波長板使用:chip0	80 - 150
V	1 露出型	80 - 100
Rc	半波長板使用:chip1	80 - 150
Rc	半波長板使用:chip0	80 - 150
Rc	1 露出型	80 - 100
Ic	半波長板使用:chip1	50 - 140
Ic	半波長板使用:chip0	50 - 140
Ic	1 露出型	50 - 140
z	半波長板使用:chip1	50 - 150
z	半波長板使用:chip0	50 - 150
z	1 露出型	50 - 150

表 3.3: アパーチャーサイズの決定

3.5 方位角原点補正

3.5.1 観測

偏光方位角の原点を見積もるために、以下(表3.4)のように強偏光標準星4つ観測した。HD183143[10]とHD161056の観測から半波長板を使用したとき、HD19820とHD43384[11]の観測から1露出型の補正を行った。

名前	赤経 (RA.)	赤緯 (Dec)	V 等級	観測期間	観測モード	フィルタ
HD183143	19 27 26.6	+18 17 45.2	6.9	2009.3/11	狭偏光、半波長使用	Rc
HD161056	17 43 47.0	-07 04 46.9	6.3	2009.3/11(3枚)	狭偏光、半波長使用	Rc
HD19820	03 14 05.35	+59 33 47.7	7.11	2009.11/17(3枚)	狭偏光、1露出型	Rc
HD43384	06 16 58.7	+23 44 27	6.3	2009.11/03(5枚)	狭偏光、1露出型	Rc

表 3.4: 強偏光標準星の試験観測

	カタログ値		観測値 (器械座標)	
名前	偏光度 P	偏光方位角 PA	偏光度 P	偏光方位角 PA
HD183143	5.80%	179.7°	5.853 ± 0.22%	79.199
HD161056	4.012%	67.33°	4.016 ± 0.24%	-34.125
HD19820	4.526%	114.46°	4.695 ± 0.102%	72.552
HD43384	2.86%	170.7°	2.815 ± 0.076%	-53.195

表 3.5: 強偏光標準星のカタログ値

偏光方位角の原点を補正するためには、どの方向にどれだけ回転させればよいかの2つを補正する必要がある。その為、2つの標準星の観測が必要となる。補正する方法はカタログ値を θ_c 、Q,Uの器械偏光を補正した値から出した偏光方位角を θ_{obs} とすると、補正すべき値 θ_0 は、

$$\theta_0 = \theta_c - \theta_{obs} \quad (3.36)$$

もしくは、

$$\theta_0 = \theta_c + \theta_{obs} \quad (3.37)$$

となる。値を両方の式に当てはめてみて、2つの標準星から出た値が近くなる式を選べば良い。結果、HOWPolの狭偏光モードは半波長板使用時、1露出型ともに式3.36の方となり、その値は半波長板使用時:chip1で $149.9 \pm 0.76^\circ$ 、半波長板使用時:chip0で $102.2 \pm 0.83^\circ$ 、1露出型で $42.902 \pm 0.994^\circ$ となった。

3.6 器械消偏光の測定

上の強偏光標準星の観測結果から、器械消偏光を見積もった。まず、HA 依存性のモデル曲線を用いて観測した Q,U の器械偏光を補正する。次に、偏光方位角の原点を補正した結果、半波長板を使用した場合 (chip1 と chip0 の平均をとる) は、HD183143 が $P = 5.853 \pm 0.22\%$ 、HD161056 が $P = 4.016 \pm 0.24\%$ となる。そして、1 露出型の場合は HD43384 が $P = 2.815 \pm 0.076\%$ 、HD19820 が $P = 4.695 \pm 0.102\%$ となる。これはカタログ値と比べても誤差の範囲であり、器械消偏光は無視出来る程小さいといえる。

第4章 天体の観測

前章で求めた HA 依存性のモデル曲線から Rc バンドの器械偏光を求めることが出来るようになった。また、偏光方位角原点も求め、器械消偏光は無視出来るくらい小さいことが分かった。これらの偏光キャリブレーションの方法を使い、実際に観測した天体から偏光データを求めた。観測した天体は強偏光星である HD19820 と HD43384 と、X 線トランジェントである MAXI J1659-152 である (表 4.1)。

天体名	赤経 (R.A.)	赤緯 (Dec)	観測期間	フィルタ
HD19820	03 14 05.35	+59 33 47.7	2009.11/17(3 枚)	Rc
HD43384	06 16 58.7	+23 44 27	2009.11/03(5 枚)	Rc
MAXI J1659-152	16 59 10.0	-15 16 05	2010.9/28-10/26	V,Rc,Ic,J

表 4.1: 天体の観測

4.1 強偏光標準星

HOWPol の狭偏光撮像モードを用いて、HD19820 と HD43384 の観測を行った。それぞれ、Rc バンドを用いて半波長板を使用した場合と、1 露出型での場合の両方で観測をした。

まず、半波長板を使用した場合、HD19820 は 4 枚×3 組、HD43384 は 4 枚×5 組から chip1 と chip0 それぞれの値を出し、平均をとった。誤差はそれぞれの値の分散をとったものとした。結果は、HD19820 が $P = 4.50 \pm 0.15\%$ (chip1: $4.609 \pm 0.135\%$, chip0: $4.389 \pm 0.042\%$)、 $PA = 120.3 \pm 0.4^\circ$ (chip1: $120.72 \pm 0.20^\circ$, chip0: $119.93 \pm 0.13^\circ$) となり、HD43383 が $P = 2.89 \pm 0.25\%$ (chip1: $3.136 \pm 0.022\%$, chip0: $2.644 \pm 0.034\%$)、 $PA = 176.0 \pm 1.0^\circ$ (chip1: $176.74 \pm 0.58^\circ$, chip0: $175.16 \pm 0.87^\circ$) となった。これはカタログ値 (表 3.5) と比較すると、偏光度は 0.3% 以内の精度で求められている。偏光方位角については測光精度は 1° 程ではあるが、カタログ値から $3\sim 6^\circ$ ズれていた。

次に、1 露出型についての結果は、HD19820 が $P = 4.70 \pm 0.10\%$ $PA = 115.45 \pm 0.45^\circ$ 、HD43384 が $P = 2.82 \pm 0.08\%$ $PA = 169.71 \pm 1.14^\circ$ となった。1 露出型は HD19820 は 3 枚、HD43384 は 5 枚のフレームから求め、誤差はそれぞれの値の分散をとった。これによ

ると、P も PA も各フレームのバラツキはあまりない。これは、1 露出型が1 枚のフレームから偏光を求める事ができるためだと考えられる。

1 露出型の偏光を求める際に、ウォラストンプリズムによる効率 k_e/k_o は上の半波長板を使用した観測で求めた値を使用した。HOWPol の狭視野偏光撮像モードが使われるようになって以降、徐々にこの値は変化していることが分かった (付録参照)。そして、この値がズレたもので1 露出型の値を出してしまうと、偏光度や偏光方位角が精度良く求まらない。よって、1 露出型で観測する際にはなるべく近い日の値を使うのがいいと思われるが、同じ日の内でも変化が見られ、単純には考えられない。よって、GRB 等の突発天体が現れ、1 露出型だけで観測した時も、同じ視野内の天体を半波長板を使って観測し k_e/k_o の値を導出するのが良い方法だと考えられる。

4.2 MAXI J1659-152

我々は、東広島天文台にて MAXI J1659-152 (R.A., Dec = 16h59m10s, -15d16m05s J2000) の撮像及び、偏光観測に成功した。この天体は「きぼう」日本実験棟の船外実験プラットフォームに設置された全天 X 線監視装置「MAXI」によって発見された。初めは Swift/BAT で GRB 100925A として検出されたが、後に X 線トランジェントということが分かり、電波、近赤外、そして可視光での観測も行われた。MAXI J1659-152 は 2010.09/28 に hard-intermediate state になり、その後 2010.10/12 に soft-intermediate state、2010.10/17 に high-soft state、そして 2010.10/31 には hard-intermediate state になった。

観測は HOWPol と TRISPEC を用いた。

観測期間は 2010.09/28 (MJD 55467) ~ 2010.10/26 (MJD 55495) UT であり、B,V,Rc,Ic,J での撮像を行った。また、その内 2010.10/05 (MJD 55474) に HOWPol で Rc バンドでの偏光撮像に成功した。等級の較正には The Tycho-2 Catalogue (V)、The USNO-B1.0 Catalogue (Rc,Ic)、NOMAD Catalogue (V,J)、2MASS Catalogue (V,J) を用いた。

図 4.1 が観測期間のライトカーブである。横軸時間、縦軸等級としている。Rc と Ic の等級が徐々に暗くなっており、V に関しては誤差の範囲で変化が無い。図 4.2, 4.3, 4.4 は V-R、R-I、V-I の色の変化を表したものである。赤い線が MAXI J1659-152 の変化を表していて、緑の線は近くの比較星の変化である。誤差に依る不定性が大きな為、色の変化はあまり見られないが、観測期間中に若干青くなっているようにも見える。

そして、図 4.5 は MAXI J1659-152 の偏光観測から得られた Q,U の値を Q-U 平面上にプロットしたものである。赤い十字 MAXI J1659-152 から得た値で、緑が近くの比較星の偏光データである。画像上でどの星に相当するかを図 4.6 に表してある。今回は天体が低い高度であったり、天候が悪いなど観測条件がよくなかったので偏光のデータ点が 1 点しかとれなかった。偏光は HOWPol の狭視野偏光モードで撮ったもので、半波長板を使用して chip1 側から得られたデータである。また、偏光観測するには暗くなっていた為に、アパーチャーサイズが大きすぎるとスカイの寄与が大きくなり、偏光の値がばらついてしまう。そこで、適当と思われる大きさの付近でアパーチャーサイズを変えていくつか測光

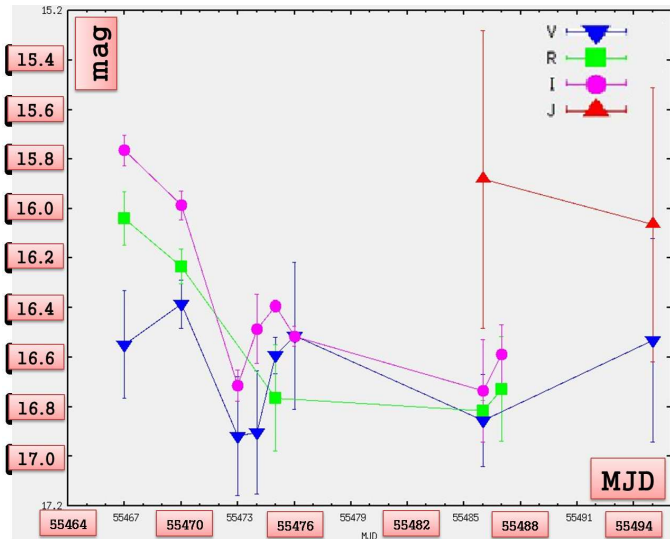


図 4.1: MAXI J1659-152 のライトカーブ

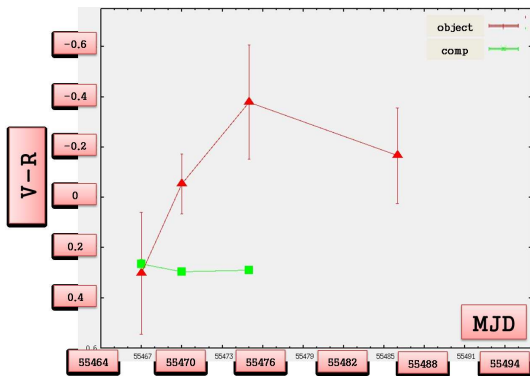


図 4.2: V-R の変化

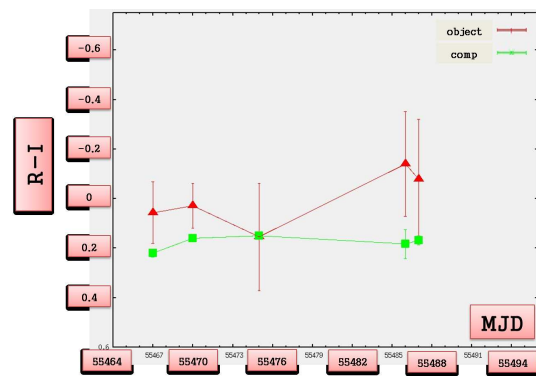


図 4.3: R-I の変化

し、そのバラツキを誤差とした。その後、器械偏光と偏光方位角原点の補正をし、Q-U 平面上にプロットした。

結果、偏光度は $P = 2.4 \pm 2.5\%$ であり、偏光方位角は $PA = 62 \pm 31^\circ$ となった。同じフレームに写りこんでいる近傍星の比較星が固有の偏光を持っていないとすると、その値は星間偏光の成分であると考えられる。今回のデータでは不定性が大きいですが、星間偏光の成分以外には偏光が見られないように考えられる。

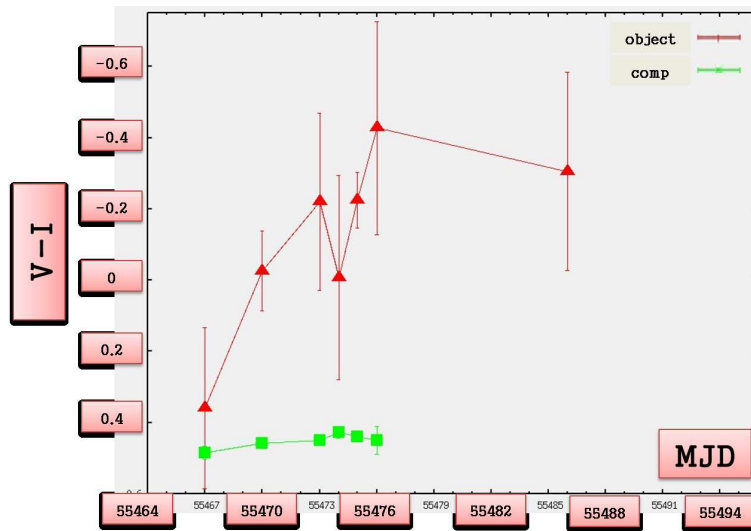


図 4.4: V-I の変化

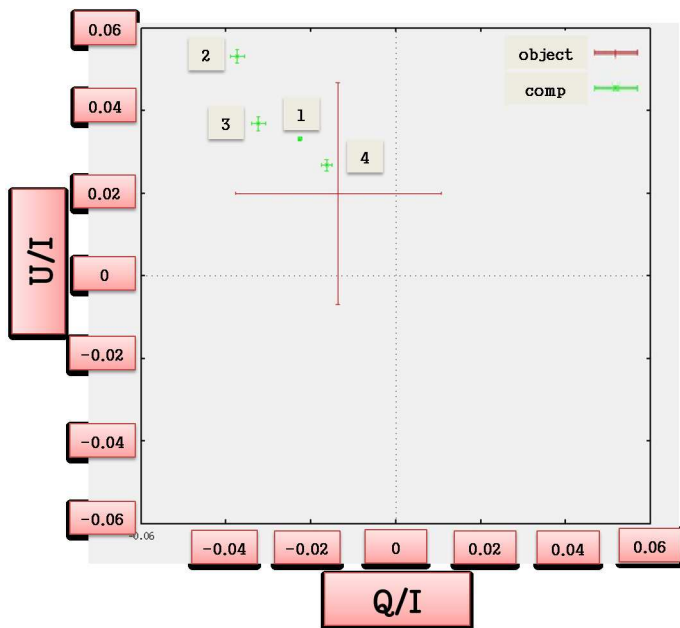


図 4.5: 偏光:Q-U 平面



図 4.6: 比較星の位置

第5章 まとめ

我々は、未開拓領域といえる GRB 初期残光の偏光を捕えるべく、1 露出型偏光撮像装置 HOWPol を開発してきた。

私は、2009 年 4 月以降の HOWPol の立ち上げの主力メンバーとして、GRB 自動観測を初めとした様々な観測モードの完成を目指して、HOWPol の制御システム全般の整備を行った。これによって、検出器を制御する 1 台の PC から、望遠鏡を含めた HOWPol 観測に関わるすべての制御を統括的に行うことが可能となり、GRB の自動・即応観測が初めて可能となった。また、分光観測など観測セットアップに時間を要するプロセスに掛かる時間が短縮され、観測効率を全般に向上させることができた。その結果、HOWPol は、TRISPEC に続くかなた望遠鏡の主力装置として活躍しており、超新星の測光・分光観測など、可視域の高効率の観測装置として成果を出し始めている (e.g., Yamanaka et al. 2009)。

また、標準星に対する様々な試験観測を行い、偏光のキャリブレーションを行った。HOWPol ではナスミス焦点特有の大きな器械偏光を持つため、同視野内に偏光標準星が写り込むなど特殊なケースを除いて、それまでは天体の偏光観測を行うことが困難な状況にあった。得られたデータから、器械偏光は第 3 鏡の反射由来の成分のみで表せることを突き止め、器械偏光の時角と赤緯の依存性のモデル曲線を求めて、偏光キャリブレーションに必要な器械偏光の導出を容易にした。また、器械消偏光が無視可能なほどに小さいことや、偏光方位角の原点補正の方法も見出した。そして、偏光が既知の標準星の観測にこれらのキャリブレーション方法を適用し、偏光度において約 0.3%、偏光方位角において約 1° という十分な精度で偏光データを得られることが判った。これで HOWPol の任意の天体の偏光観測を行うことが可能となった。

そして、MAXI J1659-152 や GRB 091208B などの天体に対する科学的な偏光データの取得にも成功した。また、GRB に対する 1 露出型偏光観測モードの自動即応観測も行っており、今後、GRB の初期残光に対する偏光観測例は着実に増えていくであろう。

なお、より幅広い観測、あるいは高精度の観測を行うためには、今回完了できなかった、器械偏光の波長依存性 (R バンド以外でも補正できるようにする) や、視野内依存性などの調査も継続して行う必要があるだろう。これらは、今後の課題である。また、今回の研究において、1 露出型偏光の測定精度には、ウォラストンプリズムに依る効率 k_e/k_o の安定性が影響することが分かったので、普段のキャリブレーション観測に含めて継続的にデータ取得することを提案する。

補遺 1

ウォラストンプリズムの透過率 ke/ko

date	ke/ko(chip1)	ke/ko_σ(chip1)	ke/ko(chip0)	ke/ko_σ(chip0)	set_number
20090727	0.995	0.0029	1.00738	0.00091	3
20090807	0.994	00000.38	1.01184	0.00049	2
20090814	0.992	0.0044	1.0103	0.0012	4
20090817	0.997	0.0026	1.0121	0.00082	8
20090818	0.990	0.0070	1.0117	0.0028	2
20090819	0.995	0.00020	1.008	0.00033	2
20090824	0.995	0.0036	1.0138	0.0030	5
20090826	0.997	0.0017	1.0111	0.00059	2
20090827	0.995	0.0034	1.0105	0.0015	3
20090830	0.999	0.00072	1.010	0.00010	2
20090831	0.993	0.0048	1.0089	0.00022	2
20100311	0.980	0.0015	1.0814	0.0022	4
20100921	0.974	0.0066	1.0719	0.014	6
20101005	0.9964	0.0086	nodata	nodata	5

補遺2

HOWPoI制御用 サーバープログラムの開発

```
// server.cpp : R\[ AvP[VGg |Cg'B

#include "stdafx.h"
#include <stdio.h>
#include <winsock2.h>
#include <ws2tcpip.h>
#include "Apismc.h"
#include "Smcdef.h"
#include "string.h"
#include "MOVEDlg.h"

#pragma comment(lib, "WSock32.lib")

#define PORT 1234 //NCAgv0|[g
#define PROH_X1 490000
#define PROH_X2 30000
#define PROH_Y1_S 8000
#define PROH_Y2_S 17000
#define PROH_Y1_B 47000
#define PROH_Y2_B 47000
#define NONE 0
#define IMAGE_x 260000 //XYstage
#define WFWol_x 497800 //
#define re_WFWol_x 470000 //
#define NFWol_x 30000 //
#define re_NFWol_x 50000 //
#define PupLens_x 30000 //
#define GRISM_x 506300 //
#define re_GRISM_x 470000 //
#define re_MIGISHITA_x 50000
#define IMAGE_y 24000 //
#define WFWol_y 3000 //
#define re_WFWol_y 3000 //
#define NFWol_y 3700 //
#define re_NFWol_y 3700 //
#define PupLens_y 52000 //
#define GRISM_y 50900 //
#define re_GRISM_y 50900 //
#define re_MIGISHITA_y 50000
#define FT1_ND1 0 //FT1(3555_old)
#define FT1_R64 1185 //
#define FT1_Hartmann1 2375 //
#define FT1_Blue 3555 //
#define FT1_Hartmann2 4740 //
#define FT1_PolBlue 5925 // (OPTPol)
#define FT1_PolRed 0 // (4740_old_IRRed)
```

```

#define NONE_AX3 400 //FT2
#define FT2_B 6325 //
#define FT2_V 5140 //
#define FT2_R 3955 //
#define FT2_I 2770 //
#define FT2_z 1585 //
#define NFPol 5000 //Mask
#define WFPol 10200 //
#define Imag 15500 //
#define LSPol 21000 //
#define Lattice 26250 //
//#define HWPIN -28922 //HWX
#define HWPIN -28873 //HWX
#define HWPOUT 0 //
#define HWrot_0 0 //HWrot
#define HWrot_1 4500 //225
#define HWrot_2 9000 //450
#define HWrot_3 13500 //675
#define HWrot_4 18000 //900
#define HWrot_5 22500 //1125
#define HWrot_6 27000 //1350
#define HWrot_7 31500 //1575
#define HWrot_8 36000 //1800
#define HWrot_9 40500 //2025
#define HWrot_A 45000 //2250
#define HWrot_B 49500 //2475
#define HWrot_C 54000 //2700
#define HWrot_D 58500 //2925
#define HWrot_E 63000 //3150
#define HWrot_F 67500 //3375
#define BESTFOCUS -110000
#define PULSE_HERROR 10
#define PULSE_LERROR 10
#define PULSE_GHERROR 1
#define PULSE_GLERROR 1
#define PULSE_HWPERROR 150

char grbuf[100];
char hoge[100];
char puni[100];
LONG CMOVEDlg::GetStatus(UINT wParam, LONG lParam)
{

dwRet = SmcGetDigitalOut("AXIS_06", &bOutData);
if ((bOutData & SMC_OUT2) == SMC_OUT2) strcpy_s(szBuffer, "ON");
else strcpy_s(szBuffer, "OFF");

//-----
// Get OutPulse from Driver
//-----
dwRet = SmcGetOutPulse(szDeviceName, &lOutPulse);
if(strcmp("O K",szComment)){
return ERROR;
}

if(strcmp(szDeviceName, "AXIS_02")==0){
while(lOutPulse >= 7111){
lOutPulse = lOutPulse - 7111;
}
}

if(strcmp(szDeviceName, "AXIS_03")==0){
while(lOutPulse >= 7111){

```

```

lOutPulse = lOutPulse - 7111;
}
}

if(strcmp(szDeviceName, "AXIS_04")==0){
while(lOutPulse >= 26583){
lOutPulse = lOutPulse - 26586;
}
}

sprintf_s(szBuffer, "%ld", lOutPulse);

//-----
// Get Pulse Status from Driver
//-----
dwRet = SmcGetPulseSts(szDeviceName, &bPulseSts);
if(strcmp("O K",szComment)){
return ERROR;
}
switch (bPulseSts) {
case SMC_PLS_STOP :
strcpy_s(szBuffer, "Pulse Stop");
break;
case SMC_PLS_READY :
strcpy_s(szBuffer, "Pulse Ready");
break;
case SMC_PLS_TIMER :
strcpy_s(szBuffer, "Pulse Timer");
break;
case SMC_PLS_ACCEL :
strcpy_s(szBuffer, "Pulse Accel");
break;
case SMC_PLS_CONST :
strcpy_s(szBuffer, "Pulse Constant");
break;
case SMC_PLS_DECEL :
strcpy_s(szBuffer, "Pulse Decel");
break;
}
//-----
// Get Move Status from Driver
//-----
dwRet = SmcGetMoveSts(szDeviceName, &bMoveSts);

switch (bMoveSts) {
case SMC_STOP :
strcpy_s(szBuffer, "Stop Motion");
break;
case SMC_PTP :
strcpy_s(szBuffer, "PTP Motion");
break;
case SMC_JOG :
strcpy_s(szBuffer, "JOG Motion");
break;
case SMC_ORG :
strcpy_s(szBuffer, "ORG Motion");
break;
}
//-----
// Get Limit Status from Driver
//-----
dwRet = SmcGetLimitSts(szDeviceName, &bLimitSts);

```

```

if ((bLimitSts & SMC_ORGLIM) == SMC_ORGLIM) strcpy_s(szBuffer, "ON");
else strcpy_s(szBuffer, "OFF");

if ((bLimitSts & SMC_ALM) == SMC_ALM) strcpy_s(szBuffer, "ON");
else strcpy_s(szBuffer, "OFF");

if ((bLimitSts & SMC_MLIM) == SMC_MLIM) strcpy_s(szBuffer, "ON");
else strcpy_s(szBuffer, "OFF");

if ((bLimitSts & SMC_PLIM) == SMC_PLIM) strcpy_s(szBuffer, "ON");
else strcpy_s(szBuffer, "OFF");

return ERROR_SUCCESS;

```

```

}

```

```

BOOL CMOVEDlg::SetMoveParam()
{
//-----
// Set axis to Master
//-----
dwRet = SmcSetMasterSlave(szDeviceName, SMC_MASTER);
//-----
// Set Distance to Driver
//-----
lDistance = atol(szBuffer);
if(bMotionType == SMC_PTP){
lDistance = labs(lDistance);
if( bStartDir == SMC_CCW){
lDistance = -(lDistance);
}
}
dwRet = SmcSetStopPosition(szDeviceName, SMC_INC, lDistance);
return TRUE;
}

```

```

BOOL CMOVEDlg::GetMoveParam()
{
//-----
// Get Distance from Driver
//-----
dwRet = SmcGetStopPosition(szDeviceName, SMC_ABS, &lDistance);
// ErrorCheck (dwRet, szComment, "SmcGetStopPosition");
// m_Comment.SetWindowText(szComment);
sprintf(szBuffer, "%ld", lDistance);
// m_Distance.SetWindowText(szBuffer);

return TRUE;
}

```

```

void CMOVEDlg::CW(){

dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
bStartDir = SMC_CW;
//-----
// Set parameters to Driver
//-----
dwRet = SmcSetMasterSlave(szDeviceName, SMC_MASTER);
lDistance = atol(szBuffer);
lDistance = labs(lDistance);

```

```

dwRet = SmcSetStopPosition(szDeviceName, SMC_INC, lDistance);
if (bRet == FALSE){
return;
}
dwRet = SmcSetReady(szDeviceName, SMC_PTP, bStartDir);
dwRet = SmcMotionStart(szDeviceName);
dwRet = SmcGetStopPosition(szDeviceName,SMC_INC,&lDistance);
sprintf(szBuffer, "%ld", lDistance);

}
void CMOVEDlg::CCW(){

if(strcmp(szDeviceName,"AXIS_04")==0){
printf("CCW rotation is prohibited!!\n ");
return;

}
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
bStartDir = SMC_CCW;
//-----
// Set parameters to Driver
//-----
dwRet = SmcSetMasterSlave(szDeviceName, SMC_MASTER);
lDistance = atol(szBuffer);
lDistance = labs(lDistance);
lDistance = -(lDistance);

dwRet = SmcSetStopPosition(szDeviceName, SMC_INC, lDistance);
if (bRet == FALSE){
return;
}
dwRet = SmcSetReady(szDeviceName, SMC_PTP, bStartDir);
dwRet = SmcMotionStart(szDeviceName);
dwRet = SmcGetStopPosition(szDeviceName,SMC_INC,&lDistance);
sprintf(szBuffer, "%ld", lDistance);

}
void CMOVEDlg::SensorOFF()
{
dwRet = SmcSetDigitalOut("AXIS_06", 0, SMC_OUT2);
}
void CMOVEDlg::SensorON()
{
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
}

void CMOVEDlg::InitAxis00()
{
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet = SmcGetLimitSts("AXIS_01", &bLimitSts);
// dwRet =SmcSetTargetSpeed("Axis_00",1500);

if ((bLimitSts & SMC_MLIM) != SMC_MLIM) {
dwRet = SmcGetLimitSts("AXIS_00", &bLimitSts);
strcpy_s(szDeviceName,"AXIS_01");
if((bLimitSts & SMC_ORGLIM) == SMC_ORGLIM){
dwRet = SmcSetMasterSlave("AXIS_01", SMC_MASTER);
bStartDir = SMC_CCW;
dwRet = SmcSetReady("AXIS_01", SMC_JOG, bStartDir);
dwRet = SmcMotionStart("AXIS_01");
waitmotor( "AXIS_01" );
}
else{
strcpy_s(szDeviceName,"AXIS_00");
}
}

```

```

dwRet = SmcSetMasterSlave("AXIS_00",SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_00",SMC_INC,-40000);
dwRet = SmcSetReady("AXIS_00", SMC_PTP, SMC_CCW);
dwRet = SmcMotionStart("AXIS_00");
waitmotor("AXIS_00");

dwRet = SmcGetLimitSts("AXIS_00", &bLimitSts);
if((bLimitSts & SMC_MLIM) == SMC_MLIM){
strcpy_s(szDeviceName,"AXIS_00");
dwRet = SmcSetMasterSlave("AXIS_00",SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_00",SMC_INC,80000);
dwRet = SmcSetReady("AXIS_00", SMC_PTP, SMC_CW);
dwRet = SmcMotionStart("AXIS_00");
waitmotor("AXIS_00");
strcpy_s(szDeviceName,"AXIS_01");
dwRet = SmcSetMasterSlave("AXIS_01",SMC_MASTER);
dwRet = SmcSetReady("AXIS_01", SMC_JOG, SMC_CCW);
dwRet = SmcMotionStart("AXIS_01");
waitmotor("AXIS_01");
}

else{
dwRet = SmcSetAlarmCLR("AXIS_00");
dwRet = SmcSetAlarmCLR("AXIS_01");
dwRet = SmcGetLimitSts("AXIS_00", &bLimitSts);
strcpy_s(szDeviceName,"AXIS_00");
dwRet = SmcSetMasterSlave("AXIS_00",SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_00",SMC_INC,10000);
dwRet = SmcSetReady("AXIS_00", SMC_PTP, SMC_CW);
dwRet = SmcMotionStart("AXIS_00");
waitmotor("AXIS_00");
strcpy_s(szDeviceName,"AXIS_01");
dwRet = SmcSetMasterSlave("AXIS_01",SMC_MASTER);
dwRet = SmcSetReady("AXIS_01", SMC_JOG, SMC_CCW);
dwRet = SmcMotionStart("AXIS_01");
waitmotor("AXIS_01");

}

}

}

dwRet = SmcGetLimitSts("AXIS_00",&bLimitSts);
if((bLimitSts & SMC_MLIM) != SMC_MLIM){
strcpy_s(szDeviceName,"AXIS_00");
dwRet = SmcSetMasterSlave("AXIS_00",SMC_MASTER);
dwRet = SmcSetReady("AXIS_00", SMC_JOG, SMC_CCW);
dwRet = SmcMotionStart("AXIS_00");
waitmotor("AXIS_00" );
}

strcpy_s(szDeviceName,"AXIS_00");
dwRet = SmcSetMasterSlave("AXIS_00",SMC_MASTER);
dwRet = SmcSetTargetSpeed("Axis_00",500);
dwRet = SmcGetLimitSts("AXIS_00", &bLimitSts);
dwRet = SmcSetStopPosition("AXIS_00", SMC_INC, 1);
dwRet = SmcSetReady("AXIS_00", SMC_PTP, SMC_CW);
while((bLimitSts & SMC_MLIM) == SMC_MLIM){
dwRet = SmcMotionStart("AXIS_00");
waitmotor("AXIS_00");
dwRet = SmcGetLimitSts("AXIS_00", &bLimitSts);
}
dwRet = SmcMotionStop("AXIS_00");
dwRet = SmcSetOutPulse("AXIS_00",0);

```



```

strcpy_s(szDeviceName,"AXIS_01");
dwRet = SmcSetMasterSlave("AXIS_01",SMC_MASTER);
dwRet = SmcGetLimitSts("AXIS_01", &bLimitSts);
dwRet = SmcSetStopPosition("AXIS_01", SMC_INC, 1);
dwRet = SmcSetReady("AXIS_01", SMC_PTP, SMC_CW);
while((bLimitSts & SMC_MLIM) == SMC_MLIM){
dwRet = SmcMotionStart("AXIS_01");
waitmotor("AXIS_01");
dwRet = SmcGetLimitSts("AXIS_01", &bLimitSts);
}

dwRet = SmcMotionStop("AXIS_01");
dwRet = SmcSetOutPulse("AXIS_01",0);
// dwRet = SmcSetDigitalOut("AXIS_06", 0, SMC_OUT2);
}

void CMOVEDlg::InitAxis02()
{
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);

bRet = SetMoveParam();
dwRet = SmcSetReady("AXIS_02", SMC_JOG, SMC_CW);
dwRet = SmcMotionStart("AXIS_02");
bRet = GetMoveParam();
dwRet = SmcGetLimitSts("AXIS_02", &bLimitSts);
while((bLimitSts & SMC_ORGLIM) != SMC_ORGLIM){
dwRet = SmcGetLimitSts("AXIS_02", &bLimitSts);
}
dwRet = SmcMotionStop("AXIS_02");
dwRet = SmcSetStopPosition("AXIS_02", SMC_INC, 1);
dwRet = SmcSetReady("AXIS_02", SMC_PTP, SMC_CW);
dwRet = SmcGetLimitSts("AXIS_02", &bLimitSts);
dwRet = SmcSetMasterSlave("AXIS_02",SMC_MASTER);
while((bLimitSts & SMC_ORGLIM) == SMC_ORGLIM){
dwRet = SmcMotionStart("AXIS_02");
waitmotor("AXIS_02");
dwRet = SmcGetLimitSts("AXIS_02", &bLimitSts);
}
dwRet = SmcMotionStop("AXIS_02");
dwRet = SmcSetOutPulse("AXIS_02",0);

// dwRet = SmcSetDigitalOut("AXIS_06", 0, SMC_OUT2);
}

void CMOVEDlg::InitAxis03()
{
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);

bRet = SetMoveParam();
dwRet = SmcSetReady("AXIS_03", SMC_JOG, SMC_CW);
dwRet = SmcMotionStart("AXIS_03");
bRet = GetMoveParam();
dwRet = SmcGetLimitSts("AXIS_03", &bLimitSts);
while((bLimitSts & SMC_ORGLIM) != SMC_ORGLIM){
dwRet = SmcGetLimitSts("AXIS_03", &bLimitSts);
}
dwRet = SmcMotionStop("AXIS_03");
dwRet = SmcSetMasterSlave("AXIS_03",SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_03", SMC_INC, 1);
dwRet = SmcSetReady("AXIS_03", SMC_PTP, SMC_CW);
}

```

```

dwRet = SmcGetLimitSts("AXIS_03", &bLimitSts);
while((bLimitSts & SMC_ORGLIM) == SMC_ORGLIM){
dwRet = SmcMotionStart("AXIS_03");
waitmotor("AXIS_03");
dwRet = SmcGetLimitSts("AXIS_03", &bLimitSts);
}
dwRet = SmcMotionStop("AXIS_03");
    dwRet = SmcSetOutPulse("AXIS_03",0);
// dwRet = SmcSetDigitalOut("AXIS_06", 0, SMC_OUT2);
}

void CMOVEDlg::InitAxis04()
{

dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
bRet = SetMoveParam();
dwRet = SmcSetReady("AXIS_04", SMC_JOG, SMC_CW);
dwRet = SmcMotionStart("AXIS_04");
bRet = GetMoveParam();
dwRet = SmcGetLimitSts("AXIS_04", &bLimitSts);
while((bLimitSts & SMC_ORGLIM) != SMC_ORGLIM){
dwRet = SmcGetLimitSts("AXIS_04", &bLimitSts);
}
dwRet = SmcMotionStop("AXIS_04");
dwRet = SmcSetMasterSlave("AXIS_04",SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_04", SMC_INC, 1);
dwRet = SmcSetReady("AXIS_04", SMC_PTP, SMC_CW);
dwRet = SmcGetLimitSts("AXIS_04", &bLimitSts);
while((bLimitSts & SMC_ORGLIM) == SMC_ORGLIM){
dwRet = SmcMotionStart("AXIS_04");
waitmotor("AXIS_04");
dwRet = SmcGetLimitSts("AXIS_04", &bLimitSts);
}
dwRet = SmcMotionStop("AXIS_04");
    dwRet = SmcSetOutPulse("AXIS_04",0);

//dwRet = SmcSetDigitalOut("AXIS_06", 0, SMC_OUT2);
}

void CMOVEDlg::InitAxis05()
{

dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet = SmcGetLimitSts("AXIS_05", &bLimitSts);
    bRet = SetMoveParam();
dwRet = SmcSetReady("AXIS_05", SMC_JOG, SMC_CW);
dwRet = SmcMotionStart("AXIS_05");
bRet = GetMoveParam();
waitmotor("AXIS_05");
dwRet = SmcSetOutPulse("AXIS_05",0);
//dwRet = SmcSetDigitalOut("AXIS_06", 0, SMC_OUT2);

}

void CMOVEDlg::InitAxis06()
{
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse >=0 && lOutPulse <= 32000){
bRet = SetMoveParam();
dwRet = SmcSetReady("AXIS_06", SMC_JOG, SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
dwRet = SmcGetLimitSts("AXIS_06", &bLimitSts);
while((bLimitSts & SMC_ORGLIM) != SMC_ORGLIM){

```

```

dwRet = SmcGetLimitSts("AXIS_06", &bLimitSts);
}
dwRet = SmcMotionStop("AXIS_06");

}
else{
bRet = SetMoveParam();
dwRet = SmcSetReady("AXIS_06", SMC_JOG, SMC_CW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
dwRet = SmcGetLimitSts("AXIS_06", &bLimitSts);
while((bLimitSts & SMC_ORGLIM) != SMC_ORGLIM){
dwRet = SmcGetLimitSts("AXIS_06", &bLimitSts);
}
dwRet = SmcMotionStop("AXIS_06");
}

dwRet = SmcSetMasterSlave("AXIS_06",SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC, 1);
dwRet = SmcSetReady("AXIS_06", SMC_PTP, SMC_CW);
dwRet = SmcGetLimitSts("AXIS_06", &bLimitSts);
while((bLimitSts & SMC_ORGLIM) == SMC_ORGLIM){
dwRet = SmcMotionStart("AXIS_06");
waitmotor("AXIS_06");
dwRet = SmcGetLimitSts("AXIS_06", &bLimitSts);
}
dwRet = SmcMotionStop("AXIS_06");
dwRet = SmcSetOutPulse("AXIS_06",0);

//dwRet = SmcSetDigitalOut("AXIS_06", 0, SMC_OUT2);

}

void CMOVEDlg::InitAxis07()
{
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);

dwRet = SmcGetLimitSts("AXIS_07", &bLimitSts);
bRet = SetMoveParam();
dwRet = SmcSetReady("AXIS_07", SMC_JOG, SMC_CW);
dwRet = SmcMotionStart("AXIS_07");
waitmotor("AXIS_07");
dwRet = SmcSetMasterSlave("AXIS_07",SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_07", SMC_INC, -1);
dwRet = SmcSetReady("AXIS_07", SMC_PTP, SMC_CW);
dwRet = SmcGetLimitSts("AXIS_07", &bLimitSts);
while((bLimitSts & SMC_PLIM) == SMC_PLIM){
dwRet = SmcMotionStart("AXIS_07");
waitmotor("AXIS_07");
dwRet = SmcGetLimitSts("AXIS_07", &bLimitSts);
}
dwRet = SmcSetOutPulse("AXIS_07",0);
//dwRet = SmcSetDigitalOut("AXIS_06", 0, SMC_OUT2);

}

void CMOVEDlg::Relay(){
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet = SmcGetOutPulse("AXIS_00", &lOutPulse);
xOutPulse = lOutPulse;
dwRet = SmcGetOutPulse("AXIS_01", &lOutPulse);
yOutPulse = lOutPulse;
if (PROH_X1 <= xOutPulse && yOutPulse <= PROH_Y1_S) //H
{
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet = SmcSetMasterSlave("AXIS_00",SMC_MASTER);
}
}

```

```

dwRet =SmcSetMasterSlave("AXIS_01",SMC_SLAVE);
dwRet =SmcSetStopPosition("AXIS_00",SMC_ABS,re_WFwol_x);
dwRet =SmcSetStopPosition("AXIS_01",SMC_ABS,re_WFwol_y);
dwRet =SmcSetReady("AXIS_00",SMC_PTP,SMC_CW);
dwRet =SmcSetReady("AXIS_01",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_00");
waitmotor("AXIS_00");
waitmotor("AXIS_01");
}
else if( xOutPulse <= PROH_X2 && yOutPulse <= PROH_Y2_S) //H
{
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetMasterSlave("AXIS_00",SMC_MASTER);
dwRet =SmcSetMasterSlave("AXIS_01",SMC_SLAVE);
dwRet =SmcSetStopPosition("AXIS_00",SMC_ABS,re_NFwol_x);
dwRet =SmcSetStopPosition("AXIS_01",SMC_ABS,re_NFwol_y);
dwRet =SmcSetReady("AXIS_00",SMC_PTP,SMC_CW);
dwRet =SmcSetReady("AXIS_01",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_00");
waitmotor("AXIS_00");
waitmotor("AXIS_01");
}
else if(PROH_X1 <= xOutPulse && PROH_Y2_B <= yOutPulse) //HE
{
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetMasterSlave("AXIS_00",SMC_MASTER);
dwRet =SmcSetMasterSlave("AXIS_01",SMC_SLAVE);
dwRet =SmcSetStopPosition("AXIS_00",SMC_ABS,re_GRISM_x);
dwRet =SmcSetStopPosition("AXIS_01",SMC_ABS,re_GRISM_y);
dwRet =SmcSetReady("AXIS_00",SMC_PTP,SMC_CW);
dwRet =SmcSetReady("AXIS_01",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_00");
waitmotor("AXIS_00");
waitmotor("AXIS_01");
}
else if( xOutPulse <= PROH_X2 && PROH_Y2_B <= yOutPulse) //HE
{
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetMasterSlave("AXIS_00",SMC_MASTER);
dwRet =SmcSetMasterSlave("AXIS_01",SMC_SLAVE);
dwRet =SmcSetStopPosition("AXIS_00",SMC_ABS,re_MIGISHITA_x);
dwRet =SmcSetStopPosition("AXIS_01",SMC_ABS,re_MIGISHITA_y);
dwRet =SmcSetReady("AXIS_00",SMC_PTP,SMC_CW);
dwRet =SmcSetReady("AXIS_01",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_00");
waitmotor("AXIS_00");
waitmotor("AXIS_01");
}
}
}

void CMOVEDlg::NWolPrism(){
Relay();

dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetMasterSlave("AXIS_00",SMC_MASTER);
dwRet =SmcSetMasterSlave("AXIS_01",SMC_SLAVE);
dwRet =SmcSetStopPosition("AXIS_00",SMC_ABS,re_NFwol_x);
dwRet =SmcSetStopPosition("AXIS_01",SMC_ABS,re_NFwol_y);
dwRet =SmcSetReady("AXIS_00",SMC_PTP,SMC_CW);
dwRet =SmcSetReady("AXIS_01",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_00");
waitmotor("AXIS_00");
}

```

```

waitmotor("AXIS_01");

dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetMasterSlave("AXIS_00",SMC_MASTER);
dwRet =SmcSetMasterSlave("AXIS_01",SMC_SLAVE);
dwRet =SmcSetStopPosition("AXIS_00",SMC_ABS,NFWol_x);
dwRet =SmcSetStopPosition("AXIS_01",SMC_ABS,NFWol_y);
dwRet =SmcSetReady("AXIS_00",SMC_PTP,SMC_CW);
dwRet =SmcSetReady("AXIS_01",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_00");

}

void CMOVEDlg::WWolPrism(){
Relay();

dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetMasterSlave("AXIS_00",SMC_MASTER);
dwRet =SmcSetMasterSlave("AXIS_01",SMC_SLAVE);
dwRet =SmcSetStopPosition("AXIS_00",SMC_ABS,re_WFWol_x);
dwRet =SmcSetStopPosition("AXIS_01",SMC_ABS,re_WFWol_y);
dwRet =SmcSetReady("AXIS_00",SMC_PTP,SMC_CW);
dwRet =SmcSetReady("AXIS_01",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_00");
waitmotor("AXIS_00");
waitmotor("AXIS_01");

dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetMasterSlave("AXIS_00",SMC_MASTER);
dwRet =SmcSetMasterSlave("AXIS_01",SMC_SLAVE);
dwRet =SmcSetStopPosition("AXIS_00",SMC_ABS,WFWol_x);
dwRet =SmcSetStopPosition("AXIS_01",SMC_ABS,WFWol_y);
dwRet =SmcSetReady("AXIS_00",SMC_PTP,SMC_CW);
dwRet =SmcSetReady("AXIS_01",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_00");

}

void CMOVEDlg::Grism(){
//none1();
Relay();

dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetMasterSlave("AXIS_00",SMC_MASTER);
dwRet =SmcSetMasterSlave("AXIS_01",SMC_SLAVE);
dwRet =SmcSetStopPosition("AXIS_01",SMC_ABS,re_GRISM_y);
dwRet =SmcSetStopPosition("AXIS_00",SMC_ABS,re_GRISM_x);
dwRet =SmcSetReady("AXIS_01",SMC_PTP,SMC_CW);
dwRet =SmcSetReady("AXIS_00",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_00");
waitmotor("AXIS_00");
waitmotor("AXIS_01");

dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetMasterSlave("AXIS_00",SMC_MASTER);
dwRet =SmcSetMasterSlave("AXIS_01",SMC_SLAVE);
dwRet =SmcSetStopPosition("AXIS_01",SMC_ABS,GRISM_y);
dwRet =SmcSetStopPosition("AXIS_00",SMC_ABS,GRISM_x);
dwRet =SmcSetReady("AXIS_01",SMC_PTP,SMC_CW);
dwRet =SmcSetReady("AXIS_00",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_00");

}

void CMOVEDlg::ND1(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetStopPosition("AXIS_02",SMC_ABS,FT1_ND1);

```

```

dwRet =SmcSetReady("AXIS_02",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_02");
bRet = GetMoveParam();

}
void CMOVEDlg::IRPol(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetStopPosition("AXIS_02",SMC_ABS,FT1_PolRed);
dwRet =SmcSetReady("AXIS_02",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_02");
bRet = GetMoveParam();

}
void CMOVEDlg::R64(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetStopPosition("AXIS_02",SMC_ABS,FT1_R64);
dwRet =SmcSetReady("AXIS_02",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_02");
bRet = GetMoveParam();

}
void CMOVEDlg::HART1(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetStopPosition("AXIS_02",SMC_ABS,FT1_Hartmann1);
dwRet =SmcSetReady("AXIS_02",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_02");
bRet = GetMoveParam();

}
void CMOVEDlg::FT1Blue(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetStopPosition("AXIS_02",SMC_ABS,FT1_Blue);
dwRet =SmcSetReady("AXIS_02",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_02");
bRet = GetMoveParam();

}
void CMOVEDlg::HART2(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetStopPosition("AXIS_02",SMC_ABS,FT1_Hartmann2);
dwRet =SmcSetReady("AXIS_02",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_02");
bRet = GetMoveParam();

}
void CMOVEDlg::OPTPol(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetStopPosition("AXIS_02",SMC_ABS,FT1_PolBlue);
dwRet =SmcSetReady("AXIS_02",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_02");
bRet = GetMoveParam();

}
void CMOVEDlg::Bfilter(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet =SmcSetStopPosition("AXIS_03",SMC_ABS,FT2_B);
dwRet =SmcSetReady("AXIS_03",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_03");

```

```

bRet = GetMoveParam();

}

void CMOVEDlg::Vfilter(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet = SmcSetStopPosition("AXIS_03",SMC_ABS,FT2_V);
dwRet = SmcSetReady("AXIS_03",SMC_PTP,SMC_CW);
dwRet = SmcMotionStart("AXIS_03");
bRet = GetMoveParam();

}

void CMOVEDlg::Rfilter(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet = SmcSetStopPosition("AXIS_03",SMC_ABS,FT2_R);
dwRet = SmcSetReady("AXIS_03",SMC_PTP,SMC_CW);
dwRet = SmcMotionStart("AXIS_03");
bRet = GetMoveParam();
}

void CMOVEDlg::Ifilter(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet = SmcSetStopPosition("AXIS_03",SMC_ABS,FT2_I);
dwRet = SmcSetReady("AXIS_03",SMC_PTP,SMC_CW);
dwRet = SmcMotionStart("AXIS_03");
bRet = GetMoveParam();
}

void CMOVEDlg::zfilter(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet = SmcSetStopPosition("AXIS_03",SMC_ABS,FT2_z);
dwRet = SmcSetReady("AXIS_03",SMC_PTP,SMC_CW);
dwRet = SmcMotionStart("AXIS_03");
bRet = GetMoveParam();
}

void CMOVEDlg::NFImpolM(){
dwRet = SmcGetOutPulse("AXIS_04", &lOutPulse);
if(lOutPulse <= 5000){
lDistance = 5000 - lOutPulse;
dwRet = SmcSetMasterSlave("AXIS_04", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_04", SMC_INC,lDistance);
dwRet = SmcSetReady("AXIS_04",SMC_PTP,SMC_CW);
dwRet = SmcMotionStart("AXIS_04");
bRet = GetMoveParam();
}
else{
CMOVEDlg::InitAxis04();
dwRet = SmcSetMasterSlave("AXIS_04", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_04", SMC_INC,NFPol);
dwRet = SmcSetReady("AXIS_04",SMC_PTP,SMC_CW);
dwRet = SmcMotionStart("AXIS_04");
bRet = GetMoveParam();
}
}

void CMOVEDlg::WFImpolM(){
dwRet = SmcGetOutPulse("AXIS_04", &lOutPulse);
if(lOutPulse <= 10200){
lDistance = 10200 - lOutPulse;
dwRet = SmcSetMasterSlave("AXIS_04", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_04", SMC_INC,lDistance);
dwRet = SmcSetReady("AXIS_04",SMC_PTP,SMC_CW);
}
}

```

```

dwRet = SmcMotionStart("AXIS_04");
bRet = GetMoveParam();
}
else{
CMOVEDlg::InitAxis04();
dwRet = SmcSetMasterSlave("AXIS_04", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_04", SMC_INC,WFPol);
dwRet = SmcSetReady("AXIS_04",SMC_PTP,SMC_CW);
dwRet = SmcMotionStart("AXIS_04");
bRet = GetMoveParam();
}
}
void CMOVEDlg::ImageM(){
dwRet = SmcGetOutPulse("AXIS_04", &lOutPulse);
if(lOutPulse <= 15500){
lDistance = 15500 - lOutPulse;
dwRet = SmcSetMasterSlave("AXIS_04", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_04", SMC_INC,lDistance);
dwRet = SmcSetReady("AXIS_04",SMC_PTP,SMC_CW);
dwRet = SmcMotionStart("AXIS_04");
bRet = GetMoveParam();
}
else{
CMOVEDlg::InitAxis04();
dwRet = SmcSetMasterSlave("AXIS_04", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_04", SMC_INC,Imag);
dwRet = SmcSetReady("AXIS_04",SMC_PTP,SMC_CW);
dwRet = SmcMotionStart("AXIS_04");
bRet = GetMoveParam();
}
}
void CMOVEDlg::SpecM(){
dwRet = SmcGetOutPulse("AXIS_04", &lOutPulse);
if(lOutPulse <= 21000){
lDistance = 21000 - lOutPulse;
dwRet = SmcSetMasterSlave("AXIS_04", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_04", SMC_INC,lDistance);
dwRet = SmcSetReady("AXIS_04",SMC_PTP,SMC_CW);
dwRet = SmcMotionStart("AXIS_04");
bRet = GetMoveParam();
}
else{
CMOVEDlg::InitAxis04();
dwRet = SmcSetMasterSlave("AXIS_04", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_04", SMC_INC,LSPol);
dwRet = SmcSetReady("AXIS_04",SMC_PTP,SMC_CW);
dwRet = SmcMotionStart("AXIS_04");
bRet = GetMoveParam();
}
}
void CMOVEDlg::latmask(){
dwRet = SmcGetOutPulse("AXIS_04", &lOutPulse);
if(lOutPulse <= 26250){
lDistance = 26250 - lOutPulse;
dwRet = SmcSetMasterSlave("AXIS_04", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_04", SMC_INC,lDistance);
dwRet = SmcSetReady("AXIS_04",SMC_PTP,SMC_CW);
dwRet = SmcMotionStart("AXIS_04");
bRet = GetMoveParam();
}
else{
CMOVEDlg::InitAxis04();
dwRet = SmcSetMasterSlave("AXIS_04", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_04", SMC_INC,Lattice);
}
}

```



```

dwRet = SmcSetReady("AXIS_04",SMC_PTP,SMC_CW);
dwRet = SmcMotionStart("AXIS_04");
bRet = GetMoveParam();
}
}
void CMOVEDlg::HWPIn(){
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet = SmcGetLimitSts("AXIS_05", &bLimitSts);
    bRet = SetMoveParam();
dwRet = SmcSetReady("AXIS_05", SMC_JOG, SMC_CCW);
if((bLimitSts & SMC_MLIM) != SMC_MLIM){
dwRet = SmcMotionStart("AXIS_05");
bRet = GetMoveParam();
}
}
void CMOVEDlg::HWPO(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse >=0 && lOutPulse <= 32000){
lDistance = -(lOutPulse + 1000);
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
waitmotor("AXIS_06");
bRet = GetMoveParam();
CMOVEDlg::InitAxis06();
}
else{
CMOVEDlg::InitAxis06();
}
}

void CMOVEDlg::HWP225(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > 4500 && lOutPulse <= 36500){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >= 0 && lOutPulse <= 4500){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,4500);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::HWP450(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > 9000 && lOutPulse <= 41000){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
}

```

```

waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >= 0 && lOutPulse <= 9000){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,9000);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::HWP675(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > 13500 && lOutPulse <= 46500){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >=0 && lOutPulse <= 13500){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,13500);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::HWP4(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > HWrot_4 && lOutPulse <= 46500){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >=0 && lOutPulse <= HWrot_4){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,HWrot_4);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::HWP5(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > HWrot_5 && lOutPulse <= 46500){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
}
}

```

```

bRet = GetMoveParam();
waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >=0 && lOutPulse <= HWrot_5){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,HWrot_5);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::HWP6(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > HWrot_6 && lOutPulse <= 46500){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >=0 && lOutPulse <= HWrot_6){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,HWrot_6);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::HWP7(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > HWrot_7 && lOutPulse <= 46500){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >=0 && lOutPulse <= HWrot_7){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,HWrot_7);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::HWP8(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > HWrot_8 && lOutPulse <= 46500){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
}
}

```

```

dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >=0 && lOutPulse <= HWrot_8){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,HWrot_8);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::HWP9(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > HWrot_9 && lOutPulse <= 46500){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >=0 && lOutPulse <= HWrot_9){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,HWrot_9);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::HWPB(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > HWrot_A && lOutPulse <= 46500){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >=0 && lOutPulse <= HWrot_A){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,HWrot_A);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::HWPB(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > HWrot_B && lOutPulse <= 46500){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);

```

```

dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >=0 && lOutPulse <= HWrot_B){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,HWrot_B);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::HWPC(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > HWrot_C && lOutPulse <= 46500){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >=0 && lOutPulse <= HWrot_C){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,HWrot_C);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::HWPD(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > HWrot_D && lOutPulse <= 46500){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >=0 && lOutPulse <= HWrot_D){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,HWrot_D);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::HWPE(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > HWrot_E && lOutPulse <= 46500){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);

```

```

dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >=0 && lOutPulse <= HWrot_E){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,HWrot_E);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::HWPFF(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if(lOutPulse > HWrot_F && lOutPulse <= 46500){
lDistance = lOutPulse + 1000;
dwRet = SmcSetMasterSlave("AXIS_06", SMC_MASTER);
dwRet = SmcSetStopPosition("AXIS_06", SMC_INC,-lDistance);
dwRet = SmcSetReady("AXIS_06",SMC_PTP,SMC_CCW);
dwRet = SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
waitmotor("AXIS_06");
CMOVEDlg::InitAxis06();
}
else if(lOutPulse >=0 && lOutPulse <= HWrot_F){
}
else{
CMOVEDlg::InitAxis06();
}
dwRet =SmcSetStopPosition("AXIS_06",SMC_ABS,HWrot_F);
dwRet =SmcSetReady("AXIS_06",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_06");
bRet = GetMoveParam();
}
void CMOVEDlg::Bestfocus(){
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet = SmcSetMasterSlave("AXIS_07", SMC_MASTER);
dwRet =SmcSetStopPosition("AXIS_07",SMC_ABS,-110000);
dwRet =SmcSetReady("AXIS_07",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_07");
bRet = GetMoveParam();

}
void CMOVEDlg::none1(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet = SmcSetMasterSlave("AXIS_00", SMC_MASTER);
dwRet =SmcSetStopPosition("AXIS_00",SMC_ABS,260000);
dwRet =SmcSetReady("AXIS_00",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_00");
bRet = GetMoveParam();

bRet = SetMoveParam();
dwRet = SmcSetMasterSlave("AXIS_01", SMC_MASTER);
dwRet =SmcSetStopPosition("AXIS_01",SMC_ABS,24000);
dwRet =SmcSetReady("AXIS_01",SMC_PTP,SMC_CW);
dwRet =SmcMotionStart("AXIS_01");
bRet = GetMoveParam();

```

```

}
void CMOVEDlg::none2(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet = SmcSetStopPosition("AXIS_02", SMC_ABS, 0);
dwRet = SmcSetReady("AXIS_02", SMC_PTP, SMC_CW);
dwRet = SmcMotionStart("AXIS_02");
bRet = GetMoveParam();

}

void CMOVEDlg::none3(){
bRet = SetMoveParam();
dwRet = SmcSetDigitalOut("AXIS_06", SMC_OUT2, SMC_OUT2);
dwRet = SmcSetStopPosition("AXIS_03", SMC_ABS, 400);
dwRet = SmcSetReady("AXIS_03", SMC_PTP, SMC_CW);
dwRet = SmcMotionStart("AXIS_03");
bRet = GetMoveParam();

}

void CMOVEDlg::Status0(){
dwRet = SmcGetOutPulse("AXIS_00", &lOutPulse);
xOutPulse = lOutPulse;
dwRet = SmcGetOutPulse("AXIS_01", &lOutPulse);
yOutPulse = lOutPulse;
if(WFWol_x - PULSE_LERROR <= xOutPulse && xOutPulse <= WFWol_x + PULSE_HERROR
&& WFWol_y - PULSE_LERROR <= yOutPulse && yOutPulse <= WFWol_y + PULSE_HERROR) strcpy_s(name, "1"); //wide field wol prism
else if(PupLens_x - PULSE_LERROR <= xOutPulse && xOutPulse <= PupLens_x + PULSE_HERROR
&& PupLens_y - PULSE_LERROR <= yOutPulse && yOutPulse <= PupLens_y + PULSE_HERROR) strcpy_s(name, "2"); //Pupil Lense
else if(NFWol_x - PULSE_LERROR <= xOutPulse && xOutPulse <= NFWol_x + PULSE_HERROR
&& NFWol_y - PULSE_LERROR <= yOutPulse && yOutPulse <= NFWol_y + PULSE_HERROR) strcpy_s(name, "3"); //Narrow field wol prism
else if(IMAGE_x - PULSE_LERROR <= xOutPulse && xOutPulse <= IMAGE_x + PULSE_HERROR
&& IMAGE_y - PULSE_LERROR <= yOutPulse && yOutPulse <= IMAGE_y + PULSE_HERROR) strcpy_s(name, "4"); //Image
else if(NONE - PULSE_LERROR <= xOutPulse && xOutPulse <= NONE + PULSE_HERROR
&& NONE - PULSE_LERROR <= yOutPulse && yOutPulse <= NONE + PULSE_HERROR) strcpy_s(name, "0"); //Init
else if(GRISM_x - PULSE_GLERROR <= xOutPulse && xOutPulse <= GRISM_x + PULSE_GHERROR
&& GRISM_y - PULSE_GLERROR <= yOutPulse && yOutPulse <= GRISM_y + PULSE_GHERROR) strcpy_s(name, "6"); //Grism
else strcpy_s(name, "e"); //else

dwRet = SmcGetDigitalOut("AXIS_06", &bOutData);
dwRet = SmcGetLimitSts("AXIS_00", &bLimitSts);
if ((bOutData & SMC_OUT2) == SMC_OUT2){
strcpy_s(Sensor, "1");
if ((bLimitSts & SMC_ORGLIM) == SMC_ORGLIM) strcpy_s(Org, "1");
else strcpy_s(Org, "0");
if ((bLimitSts & SMC_ALM) == SMC_ALM) strcpy_s(Alm, "1");
else strcpy_s(Alm, "0");
if ((bLimitSts & SMC_MLIM) == SMC_MLIM) strcpy_s(Mlim, "1");
else strcpy_s(Mlim, "0");
if ((bLimitSts & SMC_PLIM) == SMC_PLIM) strcpy_s(Plim, "1");
else strcpy_s(Plim, "0");
}
else{
strcpy_s(Sensor, "0");
strcpy_s(Org, "-");
strcpy_s(Alm, "-");
strcpy_s(Mlim, "-");
strcpy_s(Plim, "-");
}

dwRet = SmcGetMoveSts("AXIS_00", &bMoveSts);
if(bMoveSts == SMC_STOP) strcpy_s(Move, "0");

```

```

else strcpy_s(Move,"1");

}

void CMOVEDlg::Status1(){
dwRet = SmcGetOutPulse("AXIS_00", &lOutPulse);
xOutPulse = lOutPulse;
dwRet = SmcGetOutPulse("AXIS_01", &lOutPulse);
yOutPulse = lOutPulse;
if(WFwol_x - PULSE_LERROR <= xOutPulse && xOutPulse <= WFwol_x + PULSE_HERROR
&& WFwol_y - PULSE_LERROR <= yOutPulse && yOutPulse <= WFwol_y + PULSE_HERROR) strcpy_s(name,"1"); //wide field wol prism
else if(PupLens_x - PULSE_LERROR <= xOutPulse && xOutPulse <= PupLens_x + PULSE_HERROR
&& PupLens_y - PULSE_LERROR <= yOutPulse && yOutPulse <= PupLens_y + PULSE_HERROR) strcpy_s(name,"2"); //Pupil Lense
else if(NFWol_x - PULSE_LERROR <= xOutPulse && xOutPulse <= NFWol_x + PULSE_HERROR
&& NFWol_y - PULSE_LERROR <= yOutPulse && yOutPulse <= NFWol_y + PULSE_HERROR) strcpy_s(name,"3"); //Narrow field wol prism
else if(IMAGE_x - PULSE_LERROR <= xOutPulse && xOutPulse <= IMAGE_x + PULSE_HERROR
&& IMAGE_y - PULSE_LERROR <= yOutPulse && yOutPulse <= IMAGE_y + PULSE_HERROR) strcpy_s(name,"4"); //Image
else if(NONE - PULSE_LERROR <= xOutPulse && xOutPulse <= NONE + PULSE_HERROR
&& NONE - PULSE_LERROR <= yOutPulse && yOutPulse <= NONE + PULSE_HERROR) strcpy_s(name,"0"); //Init
else if(GRISM_x - PULSE_GLERROR <= xOutPulse && xOutPulse <= GRISM_x + PULSE_GHERROR
&& GRISM_y - PULSE_GLERROR <= yOutPulse && yOutPulse <= GRISM_y + PULSE_GHERROR) strcpy_s(name,"6"); //Grism
else strcpy_s(name,"e"); //else

dwRet = SmcGetDigitalOut("AXIS_06", &bOutData);
// printf("wh1X %s %d %s %s %s %s %s %s\n",name,xOutPulse,Move,Mlim,Org,Plim,Alm,Sensor);

dwRet = SmcGetLimitSts("AXIS_01", &bLimitSts);
if ((bOutData & SMC_OUT2) == SMC_OUT2){
strcpy_s(Sensor, "1");
if ((bLimitSts & SMC_ORGLIM) == SMC_ORGLIM) strcpy_s(Org, "1");
else strcpy_s(Org, "0");
if ((bLimitSts & SMC_ALM) == SMC_ALM) strcpy_s(Alm, "1");
else strcpy_s(Alm, "0");
if ((bLimitSts & SMC_MLIM) == SMC_MLIM) strcpy_s(Mlim, "1");
else strcpy_s(Mlim, "0");
if ((bLimitSts & SMC_PLIM) == SMC_PLIM) strcpy_s(Plim, "1");
else strcpy_s(Plim, "0");
}
else{
strcpy_s(Sensor, "0");
strcpy_s(Org, "-");
strcpy_s(Alm, "-");
strcpy_s(Mlim, "-");
strcpy_s(Plim, "-");
}

dwRet = SmcGetMoveSts("AXIS_01", &bMoveSts);
if(bMoveSts == SMC_STOP) strcpy_s(Move,"0");
else strcpy_s(Move,"1");
// printf("wh1Y %s %d %s %s %s %s %s %s\n",name,yOutPulse,Move,Mlim,Org,Plim,Alm,Sensor);

}

void CMOVEDlg::Status2(){
dwRet = SmcGetOutPulse("AXIS_02", &lOutPulse);
if(FT1_R64 - PULSE_LERROR <= lOutPulse && lOutPulse <= FT1_R64 + PULSE_HERROR) strcpy_s(name,"4"); //R64 filter
// if(FT1_ND1 - PULSE_LERROR <= lOutPulse && lOutPulse <= FT1_ND1 + PULSE_HERROR) strcpy_s(name,"1"); //ND1 filter
// else if(FT1_PolRed - PULSE_LERROR <= lOutPulse && lOutPulse <= FT1_PolRed + PULSE_HERROR) strcpy_s(name,"2"); // IRPol fi
else if(FT1_PolBlue - PULSE_LERROR <= lOutPulse && lOutPulse <= FT1_PolBlue + PULSE_HERROR) strcpy_s(name,"3"); // OPTPol fi
else if(FT1_Hartmann1 - PULSE_LERROR <= lOutPulse && lOutPulse <= FT1_Hartmann1 + PULSE_HERROR) strcpy_s(name,"5"); // Hartm
else if(FT1_Hartmann2 - PULSE_LERROR <= lOutPulse && lOutPulse <= FT1_Hartmann2 + PULSE_HERROR) strcpy_s(name,"6"); // Hartm
else if(FT1_Blue - PULSE_LERROR <= lOutPulse && lOutPulse <= FT1_Blue + PULSE_HERROR) strcpy_s(name,"7"); // Blue filter
else if(NONE - PULSE_LERROR <= lOutPulse && lOutPulse <= NONE + PULSE_HERROR) strcpy_s(name,"0"); // NONE
else strcpy_s(name,"e"); // else
dwRet = SmcGetDigitalOut("AXIS_06", &bOutData);
dwRet = SmcGetLimitSts("AXIS_02",&bLimitSts);
if ((bOutData & SMC_OUT2) == SMC_OUT2){

```



```

strcpy_s(Sensor, "1");
if ((bLimitSts & SMC_ORGLIM) == SMC_ORGLIM) strcpy_s(Org, "1");
else strcpy_s(Org, "0");
if ((bLimitSts & SMC_ALM) == SMC_ALM) strcpy_s(Alm, "1");
else strcpy_s(Alm, "0");
if ((bLimitSts & SMC_MLIM) == SMC_MLIM) strcpy_s(Mlim, "1");
else strcpy_s(Mlim, "0");
if ((bLimitSts & SMC_PLIM) == SMC_PLIM) strcpy_s(Plim, "1");
else strcpy_s(Plim, "0");
}
else{
strcpy_s(Sensor, "0");
strcpy_s(Org, "-");
strcpy_s(Alm, "-");
strcpy_s(Mlim, "-");
strcpy_s(Plim, "-");
}
}
dwRet = SmcGetMoveSts("AXIS_02", &bMoveSts);
if(bMoveSts == SMC_STOP) strcpy_s(Move, "0");
else strcpy_s(Move, "1");
// printf("wh2 %s %d %s %s %s %s %s %s %s\n",name,lOutPulse,Move,Mlim,Org,Plim,Alm,Sensor);
}
void CMOVEDlg::Status3(){
dwRet = SmcGetOutPulse("AXIS_03", &lOutPulse);
if(FT2_B - PULSE_LERROR <= lOutPulse && lOutPulse <= FT2_B + PULSE_HERROR) strcpy_s(name,"1"); //B band filter
else if(FT2_V - PULSE_LERROR <= lOutPulse && lOutPulse <= FT2_V + PULSE_HERROR) strcpy_s(name,"2"); //V band filter
else if(FT2_R - PULSE_LERROR <= lOutPulse && lOutPulse <= FT2_R + PULSE_HERROR) strcpy_s(name,"3"); //R band filter
else if(FT2_I - PULSE_LERROR <= lOutPulse && lOutPulse <= FT2_I + PULSE_HERROR) strcpy_s(name,"4"); //I band filter
else if(FT2_z - PULSE_LERROR <= lOutPulse && lOutPulse <= FT2_z + PULSE_HERROR) strcpy_s(name,"5"); //z band filter
else if(NONE_AX3 - PULSE_LERROR <= lOutPulse && lOutPulse <= NONE_AX3 + PULSE_HERROR) strcpy_s(name,"6"); //NONE
else if(NONE - PULSE_LERROR <= lOutPulse && lOutPulse <= NONE + PULSE_HERROR) strcpy_s(name,"0"); //init
else strcpy_s(name,"e"); //else
dwRet = SmcGetDigitalOut("AXIS_06", &bOutData);

dwRet = SmcGetLimitSts("AXIS_03", &bLimitSts);
if ((bOutData & SMC_OUT2) == SMC_OUT2){
strcpy_s(Sensor, "1");
if ((bLimitSts & SMC_ORGLIM) == SMC_ORGLIM) strcpy_s(Org, "1");
else strcpy_s(Org, "0");
if ((bLimitSts & SMC_ALM) == SMC_ALM) strcpy_s(Alm, "1");
else strcpy_s(Alm, "0");
if ((bLimitSts & SMC_MLIM) == SMC_MLIM) strcpy_s(Mlim, "1");
else strcpy_s(Mlim, "0");
if ((bLimitSts & SMC_PLIM) == SMC_PLIM) strcpy_s(Plim, "1");
else strcpy_s(Plim, "0");
}
else{
strcpy_s(Sensor, "0");
strcpy_s(Org, "-");
strcpy_s(Alm, "-");
strcpy_s(Mlim, "-");
strcpy_s(Plim, "-");
}
}
dwRet = SmcGetMoveSts("AXIS_03", &bMoveSts);
if(bMoveSts == SMC_STOP) strcpy_s(Move, "0");
else strcpy_s(Move, "1");
// printf("wh3 %s %d %s %s %s %s %s %s %s\n",name,lOutPulse,Move,Mlim,Org,Plim,Alm,Sensor);
}
void CMOVEDlg::Status4(){
dwRet = SmcGetOutPulse("AXIS_04", &lOutPulse);
if(NFPol - PULSE_GLERROR <= lOutPulse && lOutPulse <= NFPol + PULSE_GHERROR) strcpy_s(name,"1"); //Narrow Impol Mask

```

```

else if(WFPol - PULSE_GLERROR <= lOutPulse && lOutPulse <= WFPol + PULSE_GHERROR) strcpy_s(name,"2"); //Wide Impol Mask
else if(Imag - PULSE_GLERROR <= lOutPulse && lOutPulse <= Imag + PULSE_GHERROR) strcpy_s(name,"3"); //Image Mask
else if(LSPol - PULSE_GLERROR <= lOutPulse && lOutPulse <= LSPol + PULSE_GHERROR) strcpy_s(name,"4"); //Spectol Mask
else if(Lattice - PULSE_GLERROR <= lOutPulse && lOutPulse <= Lattice + PULSE_GHERROR) strcpy_s(name,"5"); //latticeMask
else if(NONE - PULSE_GLERROR <= lOutPulse && lOutPulse <= NONE + PULSE_GHERROR) strcpy_s(name,"0"); //init
else strcpy_s(name,"e"); //else
dwRet = SmcGetDigitalOut("AXIS_06", &bOutData);
dwRet = SmcGetLimitSts("AXIS_04",&bLimitSts);
if ((bOutData & SMC_OUT2) == SMC_OUT2){
strcpy_s(Sensor, "1");
if ((bLimitSts & SMC_ORGLIM) == SMC_ORGLIM) strcpy_s(Org, "1");
else strcpy_s(Org, "0");
if ((bLimitSts & SMC_ALM) == SMC_ALM) strcpy_s(Alm, "1");
else strcpy_s(Alm, "0");
if ((bLimitSts & SMC_MLIM) == SMC_MLIM) strcpy_s(Mlim, "1");
else strcpy_s(Mlim, "0");
if ((bLimitSts & SMC_PLIM) == SMC_PLIM) strcpy_s(Plim, "1");
else strcpy_s(Plim, "0");
}
else{
strcpy_s(Sensor, "0");
strcpy_s(Org,"-");
strcpy_s(Alm,"-");
strcpy_s(Mlim,"-");
strcpy_s(Plim,"-");
}
}
dwRet = SmcGetMoveSts("AXIS_04", &bMoveSts);
if(bMoveSts == SMC_STOP) strcpy_s(Move,"0");
else strcpy_s(Move,"1");
// printf("wh4 %s %d %s %s %s %s %s %s\n",name,lOutPulse,Move,Mlim,Org,Plim,Alm,Sensor);
}

void CMOVEDlg::Status5(){
dwRet = SmcGetOutPulse("AXIS_05", &lOutPulse);
if(HWPOUT - PULSE_LERROR <= lOutPulse && lOutPulse <= HWPOUT + PULSE_HERROR) strcpy_s(name,"0"); //HWP OUT
else if(lOutPulse <= HWPIN + PULSE_HWPERROR) strcpy_s(name,"1"); //HWP IN
// else if(HWPIN - PULSE_HWPERROR <= lOutPulse && lOutPulse <= HWPIN + PULSE_HWPERROR) strcpy_s(name,"1"); //HWP IN
// else if(lOutPulse <= HWPIN + PULSE_ERROR) strcpy_s(name,"1"); //HWP IN(-27000)
else strcpy_s(name,"e"); //else
dwRet = SmcGetDigitalOut("AXIS_06", &bOutData);
dwRet = SmcGetLimitSts("AXIS_05",&bLimitSts);
if ((bOutData & SMC_OUT2) == SMC_OUT2){
strcpy_s(Sensor, "1");
if ((bLimitSts & SMC_ORGLIM) == SMC_ORGLIM) strcpy_s(Org, "1");
else strcpy_s(Org, "0");
if ((bLimitSts & SMC_ALM) == SMC_ALM) strcpy_s(Alm, "1");
else strcpy_s(Alm, "0");
if ((bLimitSts & SMC_MLIM) == SMC_MLIM) strcpy_s(Mlim, "1");
else strcpy_s(Mlim, "0");
if ((bLimitSts & SMC_PLIM) == SMC_PLIM) strcpy_s(Plim, "1");
else strcpy_s(Plim, "0");
}
else{
strcpy_s(Sensor, "0");
strcpy_s(Org,"-");
strcpy_s(Alm,"-");
strcpy_s(Mlim,"-");
strcpy_s(Plim,"-");
}
}
dwRet = SmcGetMoveSts("AXIS_05", &bMoveSts);
if(bMoveSts == SMC_STOP) strcpy_s(Move,"0");
else strcpy_s(Move,"1");
// printf("wh5 %s %d %s %s %s %s %s %s\n",name,lOutPulse,Move,Mlim,Org,Plim,Alm,Sensor);
}

```

```

}
void CMOVEDlg::Status6(){
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
if (HWrot_0 - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_0 + PULSE_GHERROR) strcpy_s(name,"0"); //HWP 000.0
else if(HWrot_1 - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_1 + PULSE_GHERROR) strcpy_s(name,"1"); //HWP 022.5
else if(HWrot_2 - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_2 + PULSE_GHERROR) strcpy_s(name,"2"); //HWP 045.0
else if(HWrot_3 - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_3 + PULSE_GHERROR) strcpy_s(name,"3"); //HWP 067.5
else if(HWrot_4 - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_4 + PULSE_GHERROR) strcpy_s(name,"4"); //HWP 090.0
else if(HWrot_5 - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_5 + PULSE_GHERROR) strcpy_s(name,"5"); //HWP 112.5
else if(HWrot_6 - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_6 + PULSE_GHERROR) strcpy_s(name,"6"); //HWP 135.0
else if(HWrot_7 - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_7 + PULSE_GHERROR) strcpy_s(name,"7"); //HWP 157.5
else if(HWrot_8 - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_8 + PULSE_GHERROR) strcpy_s(name,"8"); //HWP 180.0
else if(HWrot_9 - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_9 + PULSE_GHERROR) strcpy_s(name,"9"); //HWP 202.5
else if(HWrot_A - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_A + PULSE_GHERROR) strcpy_s(name,"A"); //HWP 225.0
else if(HWrot_B - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_B + PULSE_GHERROR) strcpy_s(name,"B"); //HWP 247.5
else if(HWrot_C - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_C + PULSE_GHERROR) strcpy_s(name,"C"); //HWP 270.0
else if(HWrot_D - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_D + PULSE_GHERROR) strcpy_s(name,"D"); //HWP 292.5
else if(HWrot_E - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_E + PULSE_GHERROR) strcpy_s(name,"E"); //HWP 315.0
else if(HWrot_F - PULSE_GLERROR <= lOutPulse && lOutPulse <= HWrot_F + PULSE_GHERROR) strcpy_s(name,"F"); //HWP 337.5
else strcpy_s(name,"e"); //else
dwRet = SmcGetDigitalOut("AXIS_06", &bOutData);
dwRet = SmcGetLimitSts("AXIS_06",&bLimitSts);
if ((bOutData & SMC_OUT2) == SMC_OUT2){
strcpy_s(Sensor, "1");
if ((bLimitSts & SMC_ORGLIM) == SMC_ORGLIM) strcpy_s(Org, "1");
else strcpy_s(Org, "0");
if ((bLimitSts & SMC_ALM) == SMC_ALM) strcpy_s(Alm, "1");
else strcpy_s(Alm, "0");
if ((bLimitSts & SMC_MLIM) == SMC_MLIM) strcpy_s(Mlim, "1");
else strcpy_s(Mlim, "0");
if ((bLimitSts & SMC_PLIM) == SMC_PLIM) strcpy_s(Plim, "1");
else strcpy_s(Plim, "0");
}
else{
strcpy_s(Sensor, "0");
strcpy_s(Org, "-");
strcpy_s(Alm, "-");
strcpy_s(Mlim, "-");
strcpy_s(Plim, "-");
}
}
dwRet = SmcGetMoveSts("AXIS_06", &bMoveSts);
if(bMoveSts == SMC_STOP) strcpy_s(Move,"0");
else strcpy_s(Move,"1");
// printf("wh6 %s %d %s %s %s %s %s %s\n",name,lOutPulse,Move,Mlim,Org,Plim,Alm,Sensor);
}
void CMOVEDlg::Status7(){
dwRet = SmcGetOutPulse("AXIS_07", &lOutPulse);
if(lOutPulse == 0) strcpy_s(name,"0"); //init
else if(lOutPulse == BESTFOCUS) strcpy_s(name,"1"); //bestfocus
else strcpy_s(name,"e"); //else
dwRet = SmcGetDigitalOut("AXIS_06", &bOutData);
dwRet = SmcGetLimitSts("AXIS_07",&bLimitSts);
if ((bOutData & SMC_OUT2) == SMC_OUT2){
strcpy_s(Sensor, "1");
if ((bLimitSts & SMC_ORGLIM) == SMC_ORGLIM) strcpy_s(Org, "1");
else strcpy_s(Org, "0");
if ((bLimitSts & SMC_ALM) == SMC_ALM) strcpy_s(Alm, "1");
else strcpy_s(Alm, "0");
if ((bLimitSts & SMC_MLIM) == SMC_MLIM) strcpy_s(Mlim, "1");
else strcpy_s(Mlim, "0");
if ((bLimitSts & SMC_PLIM) == SMC_PLIM) strcpy_s(Plim, "1");
else strcpy_s(Plim, "0");
}
}

```

```

}
else{
strcpy_s(Sensor, "0");
strcpy_s(Org, "-");
strcpy_s(Alm, "-");
strcpy_s(Mlim, "-");
strcpy_s(Plim, "-");

}

dwRet = SmcGetMoveSts("AXIS_07", &bMoveSts);
if(bMoveSts == SMC_STOP) strcpy_s(Move, "0");
else strcpy_s(Move, "1");
// printf("wh7 %s %d %s %s %s %s %s %s\n",name,lOutPulse,Move,Mlim,Org,Plim,Alm,Sensor);

}

void CMOVEDlg::Save(){
FILE *outpulsefile;
outpulsefile = fopen("loutpulse.txt", "w");
dwRet = SmcGetOutPulse("AXIS_00", &lOutPulse);
fprintf(outpulsefile, "%d\n",lOutPulse);
dwRet = SmcGetOutPulse("AXIS_01", &lOutPulse);
fprintf(outpulsefile, "%d\n",lOutPulse);
dwRet = SmcGetOutPulse("AXIS_02", &lOutPulse);
fprintf(outpulsefile, "%d\n",lOutPulse);
dwRet = SmcGetOutPulse("AXIS_03", &lOutPulse);
fprintf(outpulsefile, "%d\n",lOutPulse);
dwRet = SmcGetOutPulse("AXIS_04", &lOutPulse);
fprintf(outpulsefile, "%d\n",lOutPulse);
dwRet = SmcGetOutPulse("AXIS_05", &lOutPulse);
fprintf(outpulsefile, "%d\n",lOutPulse);
dwRet = SmcGetOutPulse("AXIS_06", &lOutPulse);
fprintf(outpulsefile, "%d\n",lOutPulse);
dwRet = SmcGetOutPulse("AXIS_07", &lOutPulse);
fprintf(outpulsefile, "%d\n",lOutPulse);
fclose(outpulsefile);

}

void CMOVEDlg::Initstatus(){
dwRet = SmcInitAll();
FILE *fp;
int i=1;
char buffer[128];
if(( fp = fopen("loutpulse.txt", "r")) == NULL){
exit(2);
}
while ((fgets(buffer, 128, fp)) != NULL) {
buffer[strlen(buffer) - 1] = '\0';
lOutPulse = atol(buffer);
switch (i) {
case 1 :
strcpy_s(szDeviceName, "AXIS_00");
break;
case 2 :
strcpy_s(szDeviceName, "AXIS_01");
break;
case 3 :
strcpy_s(szDeviceName, "AXIS_02");
break;
case 4 :
strcpy_s(szDeviceName, "AXIS_03");
break;
case 5 :
strcpy_s(szDeviceName, "AXIS_04");
break;
case 6 :

```

```

strcpy_s(szDeviceName, "AXIS_05");
break;
case 7 :
strcpy_s(szDeviceName, "AXIS_06");
break;
case 8:
strcpy_s(szDeviceName, "AXIS_07");
break;
}

dwRet = SmcSetOutPulse(szDeviceName,lOutPulse);

i++;
}
fclose(fp);

}
void CMOVEDlg::waitmotor( char *dntmp)
{
while( 1 ){
GetStatus(1,1);
dwRet = SmcGetMoveSts( dntmp, &bMoveSts );
if( bMoveSts == SMC_STOP )
break;
}
}
void CMOVEDlg::ALMClear()
{
strcpy(szDeviceName,"AXIS_00");
dwRet = SmcSetAlarmCLR(szDeviceName);
strcpy(szDeviceName,"AXIS_01");
dwRet = SmcSetAlarmCLR(szDeviceName);
strcpy(szDeviceName,"AXIS_02");
dwRet = SmcSetAlarmCLR(szDeviceName);
strcpy(szDeviceName,"AXIS_03");
dwRet = SmcSetAlarmCLR(szDeviceName);
strcpy(szDeviceName,"AXIS_04");
dwRet = SmcSetAlarmCLR(szDeviceName);
strcpy(szDeviceName,"AXIS_05");
dwRet = SmcSetAlarmCLR(szDeviceName);
strcpy(szDeviceName,"AXIS_06");
dwRet = SmcSetAlarmCLR(szDeviceName);
strcpy(szDeviceName,"AXIS_07");
dwRet = SmcSetAlarmCLR(szDeviceName);
}

char *mid( char *buf, int sb, int eb){

int i,c=0;
strcpy_s( grbuf, "" );
for(i=sb-1; i<sb+eb-1;i++){
grbuf[c] = buf[i];
c++;
}
grbuf[c]='\0';
return(grbuf);
}

int main() {

CMOVEDlg howpol;
// |[gC\Pbg
int srcSocket; //

```

```

int dstSocket; //

// sockaddr_in \
struct sockaddr_in srcAddr;
struct sockaddr_in dstAddr;
int dstAddrSize = sizeof(dstAddr);
int status;
// ep[^\
int numrcv;
char buffer[1024]="";

howpol.Initstatus();
// Windows
WSADATA data;
WSAStartup(MAKEWORD(2,0), &data);
// sockaddr_in \Zbg
memset(&srcAddr, 0, sizeof(srcAddr));
srcAddr.sin_port = htons(PORT);
srcAddr.sin_family = AF_INET;
srcAddr.sin_addr.s_addr = htonl(INADDR_ANY);

// \PbgiXg[^\j
srcSocket = socket(AF_INET, SOCK_STREAM, 0);
// \PbgoCh
bind(srcSocket, (struct sockaddr *) &srcAddr, sizeof(srcAddr));
//
listen(srcSocket, 1);

while(1){ //[vxNCAg

// t
printf("Waiting for the connection.\nPlease start-up a client-program.\n");
dstSocket = accept(srcSocket, (struct sockaddr *) &dstAddr, &dstAddrSize);
printf("Receiving connection from %s.\n", inet_ntoa(dstAddr.sin_addr));

while(strcmp(buffer, "quit\n") != 0){

//pPbgM
numrcv = recv(dstSocket, buffer, sizeof(char)*1024, 0);
if(numrcv ==0 || numrcv ==-1 ){
status = closesocket(dstSocket); break;
}
printf("Receive:%s\n",buffer);
strcpy_s(hoge,buffer);
// sprintf( hoge, "%s-----"

// fgets(hoge,100,stdin);

strcpy_s(puni,mid(hoge,1,8));
if(strcmp(puni,"almclear")==0){
howpol.ALMclear();
}

strcpy_s(puni,mid(hoge,1,4));
if(strcmp(puni,"save")==0){
howpol.Save();
}

strcpy_s(puni,mid(hoge,1,9));
if(strcmp(puni,"sensorOFF")==0 || strcmp(puni,"sensoroff")==0){
howpol.SensorOFF();
printf("sensor is OFF\n");
}
}

```

```

strcpy_s(puni,mid(hoge,1,8));
if(strcmp(puni,"sensorON")==0 || strcmp(puni,"sensoron")==0){
howpol.SensorON();
printf("sensor is ON\n");
}

strcpy_s(puni,mid(hoge,1,2));
if(strcmp(puni,"wh") == 0){
if (strlen(hoge) < 4){
sprintf( buffer, "Too short wh command" );
goto sendbuffer;
}
strcpy_s( puni, mid(hoge,3,1));
if(strcmp(puni,"1")==0){
// printf("wh1 is XY stage\n");
strcpy_s(howpol.szBuffer,mid(hoge,6,30));
if(strcmp(mid(hoge,5,1),"x")==0){
strcpy_s(howpol.szDeviceName,"AXIS_00");
}
if(strcmp(mid(hoge,5,1),"y")==0){
strcpy_s(howpol.szDeviceName,"AXIS_01");
}
if(strcmp(mid(hoge,4,5),"Grism")==0){
howpol.Grism();
strcpy_s(buffer,"Narrow fieldwollaston Grism in\n");
}
if(strcmp(mid(hoge,4,6),"NPrism")==0){
howpol.NWolPrism();
strcpy_s(buffer,"Narrow fieldwollaston Prism in\n");
}
if(strcmp(mid(hoge,4,6),"WPrism")==0){
howpol.WWolPrism();
strcpy_s(buffer,"Wide field wollaston Prism in\n");
}
if(strcmp(mid(hoge,4,4),"init")==0){
howpol.InitAxis00();
howpol.Save();
strcpy_s(buffer,"XY stage init\n");
}
if(strcmp(mid(hoge,4,4),"None")==0){
howpol.none1();
strcpy_s(buffer,"XY stage image\n");
}
if(strcmp(mid(hoge,4,1),"p")==0){
howpol.CW();
sprintf_s(buffer,"+%d pulse move",howpol.lDistance);
}
if(strcmp(mid(hoge,4,1),"m")==0){
howpol.CCW();
sprintf_s(buffer,"-%d pulse move",howpol.lDistance);
}
}

if(strcmp(puni,"2")==0){
// printf("wh2 is filter terlet1\n");
strcpy_s(howpol.szDeviceName,"AXIS_02");
strcpy_s(howpol.szBuffer,mid(hoge,5,30));
if(strcmp(mid(hoge,4,4),"None")==0){
howpol.none2();
strcpy_s(buffer,"None\n");
}
if(strcmp(mid(hoge,4,3),"ND1")==0){
howpol.ND1();
strcpy_s(buffer,"ND1 filater in\n");
}
}

```

```

if(strcmp(mid(hoge,4,5),"IRPol")==0){
howpol.IRPol();
strcpy_s(buffer,"IRpol filter in\n");
}
if(strcmp(mid(hoge,4,6),"OPTPol")==0){
howpol.OPTPol();
strcpy_s(buffer,"OPTpol filter in\n");
}
if(strcmp(mid(hoge,4,3),"R64")==0){
howpol.R64();
strcpy_s(buffer,"R64 filater in\n");
}
if(strcmp(mid(hoge,4,3),"HP1")==0){
howpol.HART1();
strcpy_s(buffer,"Hartmann plate1 filater in\n");
}
if(strcmp(mid(hoge,4,3),"HP2")==0){
howpol.HART2();
strcpy_s(buffer,"Hartmann plate2 filater in\n");
}
if(strcmp(mid(hoge,4,4),"Blue")==0){
howpol.FT1Blue();
strcpy_s(buffer,"Blue filater in\n");
}
if(strcmp(mid(hoge,4,4),"init")==0){
howpol.InitAxis02();
howpol.Save();
strcpy_s(buffer,"filter1 init\n");
}
if(strcmp(mid(hoge,4,1),"p")==0){
howpol.CW();
}
if(strcmp(mid(hoge,4,1),"m")==0){
howpol.CCW();
}
}

if(strcmp(puni,"3")==0){
// printf("wh3 is filter terlet2\n");
strcpy_s(howpol.szDeviceName,"AXIS_03");
strcpy_s(howpol.szBuffer,mid(hoge,5,30));
if(strcmp(mid(hoge,4,1),"B")==0){
howpol.Bfilter();
strcpy_s(buffer,"B band filter in\n");
}
if(strcmp(mid(hoge,4,1),"V")==0){
howpol.Vfilter();
strcpy_s(buffer,"V band filter in\n");
}
if(strcmp(mid(hoge,4,1),"R")==0){
howpol.Rfilter();
strcpy_s(buffer,"R band filter in\n");
}
if(strcmp(mid(hoge,4,1),"I")==0){
howpol.Ifilter();
strcpy_s(buffer,"I band filter in\n");
}
if(strcmp(mid(hoge,4,1),"Z")==0){
howpol.zfilter();
strcpy_s(buffer,"Z band filter in\n");
}
if(strcmp(mid(hoge,4,4),"init")==0){
howpol.InitAxis03();
howpol.Save();
strcpy_s(buffer,"filter2 init\n");
}

```



```

}
if(strcmp(mid(hoge,4,4),"None")==0){
howpol.none3();
strcpy_s(buffer,"None\n");
}
if(strcmp(mid(hoge,4,1),"p")==0){
howpol.CW();
}
if(strcmp(mid(hoge,4,1),"m")==0){
howpol.CCW();
}
}

if(strcmp(puni,"4")==0){
// printf("wh4 is Mask Terlet\n");
strcpy_s(howpol.szDeviceName,"AXIS_04");
strcpy_s(howpol.szBuffer,mid(hoge,5,30));
if(strcmp(mid(hoge,4,6),"NImpol")==0){
howpol.NFImpolM();
strcpy_s(buffer,"Narrow impol mask in\n");
}
if(strcmp(mid(hoge,4,6),"WImpol")==0){
howpol.WFImpolM();
strcpy_s(buffer,"Wide impol mask in\n");
}
if(strcmp(mid(hoge,4,5),"Image")==0){
howpol.ImageM();
strcpy_s(buffer,"Wide image mask in\n");
}
if(strcmp(mid(hoge,4,7),"Spectol")==0){
howpol.SpecM();
strcpy_s(buffer,"spectol mode in\n");
}
if(strcmp(mid(hoge,4,7),"latmask")==0){
howpol.latmask();
strcpy_s(buffer,"lattice mask in\n");
}
}
if(strcmp(mid(hoge,4,4),"init")==0){
howpol.InitAxis04();
howpol.Save();
strcpy_s(buffer,"Mask init");
}
if(strcmp(mid(hoge,4,1),"p")==0){
howpol.CW();
}
if(strcmp(mid(hoge,4,1),"m")==0){
howpol.CCW();
}
}

if(strcmp(puni,"5")==0){
// printf("wh5 is HWP Xstage\n");
strcpy_s(howpol.szDeviceName,"AXIS_05");
strcpy_s(howpol.szBuffer,mid(hoge,5,30));
if(strcmp(mid(hoge,4,5),"hwpin")==0){
howpol.HWPIIn();
strcpy_s(buffer,"HWP in\n");
}
if(strcmp(mid(hoge,4,6),"hwput")==0){
howpol.InitAxis05();
strcpy_s(buffer,"HWP out\n");
}
if(strcmp(mid(hoge,4,4),"init")==0){
howpol.InitAxis05();
howpol.Save();
}
}

```

```

strcpy_s(buffer,"HWP init\n");
}
if(strcmp(mid(hoge,4,1),"p")==0){
howpol.CW();
}
if(strcmp(mid(hoge,4,1),"m")==0){
howpol.CCW();
}
}

if(strcmp(puni,"6")==0){
// printf("wh6 is HWP rot\n");
strcpy_s(howpol.szDeviceName,"AXIS_06");
strcpy_s(howpol.szBuffer,mid(hoge,5,30));
if(strcmp(mid(hoge,4,1),"0")==0){
howpol.HWP0();
strcpy_s(buffer,"HWP rotation angle 0\n");
}
if(strcmp(mid(hoge,4,1),"1")==0){
howpol.HWP225();
strcpy_s(buffer,"HWP rotation angle 22.5\n");
}
if(strcmp(mid(hoge,4,1),"2")==0){
howpol.HWP450();
strcpy_s(buffer,"HWP rotation angle 45.0\n");
}
if(strcmp(mid(hoge,4,1),"3")==0){
howpol.HWP675();
strcpy_s(buffer,"HWP rotation angle 67.5\n");
}
if(strcmp(mid(hoge,4,1),"4")==0){
howpol.HWP4();
strcpy_s(buffer,"HWP rotation angle 90.0\n");
}
if(strcmp(mid(hoge,4,1),"5")==0){
howpol.HWP5();
strcpy_s(buffer,"HWP rotation angle 112.5\n");
}
if(strcmp(mid(hoge,4,1),"6")==0){
howpol.HWP6();
strcpy_s(buffer,"HWP rotation angle 135.0\n");
}
if(strcmp(mid(hoge,4,1),"7")==0){
howpol.HWP7();
strcpy_s(buffer,"HWP rotation angle 157.5\n");
}
if(strcmp(mid(hoge,4,1),"8")==0){
howpol.HWP8();
strcpy_s(buffer,"HWP rotation angle 180.0\n");
}
if(strcmp(mid(hoge,4,1),"9")==0){
howpol.HWP9();
strcpy_s(buffer,"HWP rotation angle 202.5\n");
}
if(strcmp(mid(hoge,4,1),"A")==0 || strcmp(mid(hoge,4,1),"a")==0){
howpol.HWPA();
strcpy_s(buffer,"HWP rotation angle 225.0\n");
}
if(strcmp(mid(hoge,4,1),"B")==0 || strcmp(mid(hoge,4,1),"b")==0){
howpol.HWPB();
strcpy_s(buffer,"HWP rotation angle 247.5\n");
}
if(strcmp(mid(hoge,4,1),"C")==0 || strcmp(mid(hoge,4,1),"c")==0){
howpol.HWPC();
strcpy_s(buffer,"HWP rotation angle 270.0\n");
}

```

```

}
if(strcmp(mid(hoge,4,1),"D")==0 || strcmp(mid(hoge,4,1),"d")==0){
howpol.HWPD();
strcpy_s(buffer,"HWP rotation angle 292.5\n");
}
if(strcmp(mid(hoge,4,1),"E")==0 || strcmp(mid(hoge,4,1),"e")==0){
howpol.HWPE();
strcpy_s(buffer,"HWP rotation angle 315.0\n");
}
if(strcmp(mid(hoge,4,1),"F")==0 || strcmp(mid(hoge,4,1),"f")==0){
howpol.HWPF();
strcpy_s(buffer,"HWP rotation angle 337.5\n");
}
if(strcmp(mid(hoge,4,4),"init")==0){
howpol.InitAxis06();
howpol.Save();
strcpy_s(buffer,"HWP Init\n");
}
if(strcmp(mid(hoge,4,1),"p")==0){
howpol.CW();
}
if(strcmp(mid(hoge,4,1),"m")==0){
howpol.CCW();
}
}

if(strcmp(puni,"7")==0){
// printf("wh7 is Z stage\n");
strcpy_s(howpol.szDeviceName,"AXIS_07");
strcpy_s(howpol.szBuffer,mid(hoge,5,30));
if(strcmp(mid(hoge,4,9),"Bestfocus")==0){
howpol.Bestfocus();
strcpy_s(buffer,"Best focus\n");
}
if(strcmp(mid(hoge,4,4),"init")==0){
howpol.InitAxis07();
howpol.Save();
strcpy_s(buffer,"Z stage init\n");
}
}

if(strcmp(puni,"st") == 0){
if (strlen(hoge) < 3){
sprintf( buffer, "Too short st command" );
goto sendbuffer;
}

// strcpy_s(buffer,"wheel name OutPulse MoveStatus -Lim Org +Lim Alm Sensor\n");
if(strcmp(mid(hoge,3,1),"0")==0){
howpol.Status0();
sprintf_s(buffer,"%s%s%s%s%s%s%07d",\
howpol.name,howpol.Move,howpol.Mlim,howpol.Org,howpol.Plim,howpol.Alm,howpol.Sensor,howpol.xOutPulse);
}
if(strcmp(mid(hoge,3,1),"1")==0){
howpol.Status1();
sprintf_s(buffer,"%s%s%s%s%s%s%07d",\
howpol.name,howpol.Move,howpol.Mlim,howpol.Org,howpol.Plim,howpol.Alm,howpol.Sensor,howpol.yOutPulse);
}
if(strcmp(mid(hoge,3,1),"2")==0){
howpol.Status2();
sprintf_s(buffer,"%s%s%s%s%s%s%07d",\
howpol.name,howpol.Move,howpol.Mlim,howpol.Org,howpol.Plim,howpol.Alm,howpol.Sensor,howpol.lOutPulse);
}
}

```

```

    if(strcmp(mid(hoge,3,1),"3")==0){
        howpol.Status3();
        sprintf_s(buffer,"%s%s%s%s%s%s%07d",\
howpol.name,howpol.Move,howpol.Mlim,howpol.Org,howpol.Plim,howpol.Alm,howpol.Sensor,howpol.lOutPulse);

    }
    if(strcmp(mid(hoge,3,1),"4")==0){
        howpol.Status4();
        sprintf_s(buffer,"%s%s%s%s%s%s%07d",\
howpol.name,howpol.Move,howpol.Mlim,howpol.Org,howpol.Plim,howpol.Alm,howpol.Sensor,howpol.lOutPulse);
    }
    if(strcmp(mid(hoge,3,1),"5")==0){
        howpol.Status5();
        sprintf_s(buffer,"%s%s%s%s%s%s%07d",\
howpol.name,howpol.Move,howpol.Mlim,howpol.Org,howpol.Plim,howpol.Alm,howpol.Sensor,howpol.lOutPulse);
    }
    if(strcmp(mid(hoge,3,1),"6")==0){
        howpol.Status6();
        sprintf_s(buffer,"%s%s%s%s%s%s%07d",\
howpol.name,howpol.Move,howpol.Mlim,howpol.Org,howpol.Plim,howpol.Alm,howpol.Sensor,howpol.lOutPulse);
    }
    if(strcmp(mid(hoge,3,1),"7")==0){
        howpol.Status7();
        sprintf_s(buffer,"%s%s%s%s%s%s%07d",\
howpol.name,howpol.Move,howpol.Mlim,howpol.Org,howpol.Plim,howpol.Alm,howpol.Sensor,howpol.lOutPulse);
    }
    }
    strcpy_s(puni,mid(hoge,1,4));
    if(strcmp(puni,"exit") == 0){
        howpol.Save();
        SmcExitAll();
        //return(0);
    }

sendbuffer:
    //pPbgM
    send(dstSocket, buffer, sizeof(char)*1024, 0);
}

howpol.Save();

}

// Windows I
WSACleanup();
return(0);
}

```

謝辞

はじめに、この3年間研究面・生活面ともに指導して下さった川端先生、ありがとうございました。本研究を行うにあたって、HOWPol やかなた望遠鏡等の装置のことだけでなく、研究に対する姿勢や装置開発のチームとしての取り組み方など様々なことを教えてもらいました。僕の仕事量は先生の期待するところではなかったかもしれませんが、それでも、いつも指導して下さり、先生からたくさんの事を学びました。ほんとうに感謝しています。そして、研究室で特に深沢先生にはたいへん御世話になりました。学校生活の面では御迷惑をお掛けしましたし、就職活動などの大変な時期にも理解をしてくださいました。僕が修士論文を書くまでこれたのは先生のおかげです。また、研究室、宇宙科学センターの大杉先生、吉田先生、水野先生、片桐先生、植村先生、ひろたかさん、みなさんやさしく指導して下さったことに感謝いたします。

次に、研究室のみなさんに感謝します。もう卒業された先輩も含む、研究室の先輩方、勉強面では知識不足な僕にわかりやすく教えてくださいました。生活面でも適当な僕は迷惑をかけたとおもいます。また、後輩のみんなには先輩らしいことはできませんでした。同級生のみんなとも飲み会とかあまり参加できなくてごめんなさい。それでも、みんな楽しく接してくれたことに感謝しています。夜な夜な、大部屋でみんなと話したり、僕の変な話につきあってくれたり、天文台生活や、全部が楽しかったです。

この研究室での生活はとても良い経験になりました。関わったみんなに感謝します。この先もがんばります。

参考文献

- [1] 「Early Phase Observations of Extremely Luminous Type Ia Supernova 2009dc」
Yamanaka, M., Kawabata, K. S., Kinugasa, K., Tanaka, M., Imada, A., Maeda, K., Nomoto, K., Arai, A., Chiyonobu, S., Fukazawa, Y., Hashimoto, O., Honda, S., Ikejiri, Y., Itoh, R., Kamata, Y., Kawai, N., Komatsu, T., Konishi, K., Kuroda, D., Miyamoto, H., Miyazaki, S., Nagae, O., Nakaya, H., Ohsugi, T., Omodaka, T., Sakai, N., Sasada, M., Suzuki, M., Taguchi, H., Takahashi, H., Tanaka, H., Uemura, M., Yamashita, T., Yanagisawa, K., Yoshida, M. 2009 *Astrophysical Journal*, 707, L118-L122
- [2] 千代延 真吾 2006 年度 修士論文 「かなた望遠鏡用 1 露出型偏光撮像装置 HOWPol の筐体及び駆動機構の開発」 (広島大学)
- [3] 田中 祐行 2008 年度 修士論文 「かなた望遠鏡用 1 露出型偏光撮像装置 HOWPol の開発と性能評価」 (広島大学)
- [4] 上原 岳士 2008 年度 修士論文 「かなた望遠鏡を用いた X 線フレアを伴うガンマ線バースト残光の時間変動スペクトルの研究」 (広島大学)
- [5] 上原 岳士 2006 年度 卒業論文 「ガンマ線バーストの多波長即時解析システムの構築と GRB061121 の観測」 (広島大学)
- [6] 田中 済 1992 国立天文台報 「高精度望遠鏡ポインティングに必要な天体位置の補正」
- [7] 山下 泰正 1993 国立天文台報 「経緯台望遠鏡の写野回転補正に関する諸問題」
- [8] 永江 修 2005 年度 修士論文 「可視直線偏光分光観測によるマイクロクエーサー LSI+61° 303 の星周構造に関する研究」 (広島大学)
- [9] Koji S. Kawabata et al. 1999 *PASP* 「A New Spectropolarimeter at the Dodaira Observatory」
- [10] 「Planets, Stars & Nebulae studied with photopolarimetry」, 1974, ed. T. Gehrels.
- [11] Oudmaiijer et al. (2005), *MNRAS*